Poznámky z přednášek
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

Programování s omezujícími podmínkami
BONUS

Peter Černo, 2010
petercerno@gmail.com

Garant: doc. RNDr. Roman Barták, Ph.D.
E-mail: Roman.Bartak@mff.cuni.cz
Domácí stránka: `http://ktiml.ms.mff.cuni.cz/~bartak/`

Anotace: Přednáška podává přehled o technikách splňování omezujících podmínek. Zaměřena je na algoritmy splňování podmínek a to jak algoritmy prohledávací (prohledávání do hloubky, lokální prohledávání) tak algoritmy propagační (hranová konzistence, konzistence po cestě). Probíráno je také řešení příliš omezených problémů a různé modelovací techniky.

Sylabus:

1. Úvod: historické souvislosti, vazba na další obory, aplikační oblasti, definice problému splňování omezujících podmínek, binarizace problémů.
2. Algoritmy lokálního prohledávání: metoda největšího stoupání, minimalizace konfliktů, náhodná procházka, Tabu seznam, GSAT, Genet.
3. Algoritmy prohledávání do hloubky: backtracking, backjumping, dynamický backtracking, backmarking, prohledávání s diskrepancemi, neúplné prohledávání.
4. Konzistenční techniky: vrcholová a hranová konzistence, konzistence po cestě a algoritmy pro jejich dosažení, obecné konzistenční pojmy.
5. Kombinace propagace podmínek a prohledávání: kontrola dopředu, (častěný/úplný) pohled dopředu, heuristiky pro uspořádání proměnných a hodnot, prohledávání bez navracení.
6. Optimalizační problémy, metody větví a mezí. Příliš omezené problémy a jejich modely.
7. Globální podmínky.
8. Modelování problémů a praktické realizace.

Cíl předmětu: Naučit základní používané techniky používané při úlohách splňování podmínek.

Literatura:

1. R. Dechter: Constraint Processing, Elsevier Science, 2003

2. K. Marriott, P.J. Stuckey: Programming with Constraints: An Introduction, The MIT Press, 1998

3. E. Tsang: Foundations of Constraint Satisfaction, Academic Press, 1993

This page is intentionally left blank.

02. 02. 2009 Mo
PETER ČERNO

NOPT046 PROG. S OMEZ. PODMIENKAMI

PROBLÉM SPĽŇOVANIA PODMIENOK (CSP = CONSTRAINT SATISFACTION PROBLEM)

(1) KONEČNE VEĽA PREMENNÝCH
(2) DOMÉNY - KONEČNÉ MNOŽINY (SUPERDOMÉNA)
(3) KONEČNE VEĽA PODMIENOK (EXTENZIONÁLNE / INTENZIONÁL.)

PRÍPUSTNÉ RIEŠENIE = ÚPLNÉ + KONZISTENTNÉ RIEŠ.
OPTIMÁLNE RIEŠ. = PRÍP. RIEŠ., KT. MAX / MIN OBJEKTÍVNU FCIU.

PODMIENKY: ČIAST. INFO., LOKÁLNY POHĽAD, HETEROGÉNNE DOMÉNY, SMEROVOSŤ, DEKLARATÍVNE, ADITÍVNE, MÁLOKEDY NEZÁVISLÉ

NAČO CP? BLÍZKO REÁLNEMU PROBLÉMU, DEKLARATÍVNY CHAR., JEDNOTNÝ RÁMEC PRE INTEGRÁCIU RÔZNYCH RIEŠ. TECHNÍK, ČISTÉ A ELEGANTNÉ JAZYKY

AVŠAK: KOMBIN. EXPLÓZIA, NEPREDVÍDATEĽNÉ SPRÁVANIE, NOVÉ DATA = NOVÝ PROBLÉM, PRÍLIŠ LOKÁLNE PODMIENKY, ...

REPREZENTÁCIA CSP

VRCHOLY = PREMENNÉ, (HYPER)HRANY = PODMIENKY

BINÁRNE CSP = BINÁRNE PODMIENKY (UNÁRNE KÓDUJEME DOMÉNOU)

DVA CSP PROBLÉMY SÚ EKVIVALENTNÉ $\Leftrightarrow$ MAJÚ ROVNAKÚ MNOŽINU RIEŠENÍ

ROZŠÍRENIE: RIEŠENIA MOŽNO MEDZI SEBOU SYNTAKTICKY PREVÁDZAŤ

VETA: KAŽDÉ CSP MOŽNO TRANSFORMOVAŤ NA ROZŠÍRENE-EKVIVALENTNÉ BINÁRNE CSP.

DÔKAZ: PREHODENÍM PODMIENOK A PREMENNÝCH
DVOJNÉ KÓDOVANIE: (k-)c. $\rightsquigarrow$ PREMENNÁ V_c (DOMÉNA = KONZIST. k-TICE)
MEDZI V_c A $V_{c'}$, KDE c A c' ZDIEĽAJÚ PREM. $\rightarrow$ BINÁRNA PODM.

$c_1: x_1 + x_2 + x_6 = 1$

KÓDOVANIE SO SKRYTOU PREMENNOU

(k-)c PREVEDIEME NA v_c (DOMÉNA = KONZ. k-TICE) PRE $k>2$
PRE $\forall x \in c$ ZAVEDIEME BIN. PODMIENKU MEDZI x A v_c

PREVOD KÓDOVANIA SO SKRYTOU PREMENNOU NA DUÁLNE KÓDOV.:
(1) MEDZI KAŽDOU DVOJICOU DUÁLN. PREMENNÝCH NÁJDI CESTY DĽŽKY 2 ⇝ VYTVOR PRÍSLUŠNÚ BIN. PODMIENKU
(2) ODSTRÁŇ TIEŽ CESTY

HYBRIDNÉ KÓDOVANIE : KÓDOVANIE SO SKR. PR. + DUÁLNE KÓD.

(a) DUÁLNE KÓDOVANIE : LEPŠIA PROPAGÁCIA VS. EXTENZIONÁL. REPR.
(b) SO SKR. PREMENNOU : ZACHOVÁVA PÔVODNÉ PREM. VS. OSLABENIE PROP.

ZLOŽITEJŠIE PROPAGAČNÉ ALGORITMY PRE NEBIN. PODMIENKY

SYSTEMATICKÉ PREHĽADÁVANIE

+ ÚPLNÉ, − EFEKTIVITA
(a) PRECH. ÚPLNÝCH OHODNOTENÍ : GT (GEN. A TESTUJ), LOK. PREHĽ.
(b) ROZŠIRUJEME ČIAST. OHODN. : STROMOVÉ PREHĽ.

GT, BT, BJ, BM, DB, IB, LDS LOKÁLNE: HC, MC, RW, TABU, ...

GT (GENERUJ A TESTUJ)

PRECHÁDZA ÚPLNÉ, ALE NEKONZISTENTNÉ OHODNOTENIA

GENERUJE ZBYTOČNE VEĽA OHODNOTENÍ

AKO ZLEPŠIŤ GT?
(a) INTELIGENTNÉ PREHĽADÁVANIE − POUČÍ SA Z CHÝB ⇝ LOK. PR.
(b) GENEROVANIE SPOJENÉ S TESTOVANÍM → BACKTRACKING

02.02.2009 Mo
PETER ČERNO

NOPT046 PROG. S OBMEZ. PODMIENKAMI

LOKÁLNE PREHĽADÁVANIE

EVAL = POČET NESPL. PODM.
OKOLIE = OBLIŠNÁ HODNOTA 1 PR.

STAV, EVALUÁCIA, OKOLIE, LOK. OPTIMUM (STRIKTNÉ/NESTRIKTNÉ), PLATEAU, GLOBÁLNE OPTIMUM

ROBÍME LOK. ZMENY, STRÁCAME ÚPLNOSŤ

POZNÁMKA: OKOLIE HC JE: $n * (d-1)$

HC (HILL CLIMBING) NAJVÁČŠIE STÚPANIE

ZAČNE V NÁHODNOM STAVE s
HĽADÁ VŽDY NAJLEPŠÍ STAV V OKOLÍ (MAXMOVES-KRÁT)
AK JE V STRIKTNOM LOK. MINIME → RESTART GOTO
AK JE V GLOB. MINIME, TJ. EVAL.(s) = 0 then return s

MC (MIN-CONFLICTS) MINIMALIZÁCIA KONFLIKTOV

KONFLIKTNÁ PREMENNÁ = VYSKYTUJE SA V NESPLNENEJ PODM.
IBA ZMENA KONFL. PREMENNEJ MÔŽE PRINIESŤ ZLEPŠENIE

ZAČNE V NÁHODNOM STAVE s
POKIAĽ eval(s) > 0 A moves < MaxMoves
ZVOĽ NÁHODNE KONFLIKTNÚ PREMENNÚ
NASTAV JEJ HODNOTU TAK, ABY MINIMALIZOVALA eval
AK SA JEJ HODNOTA ZMENILA, ++moves ⇒ NEMUSÍ NIKDY SKONČIŤ
return s

NEDOKÁŽE VYSKOČIŤ Z LOK. MINIMA !

RW (RANDOM WALK) NÁHODNÁ PRECHÁDZKA

STAV Z OKOLIA JE VOLENÝ NÁHODNE, RESP. HODNOTA PREM. JE VOLENÁ NÁHODNE

POTREBUJEME ZACIELENIE !

p .. PRAVDEPODOBNOSŤ NÁHODNÉHO KROKU
1-p .. PRAVD. POUŽITIA SMEROVEJ HEURISTIKY

MCRW (MIN-CONFLICTS-RANDOM-WALK)

ZAČNI V NÁHODNOM STAVE s
DOKEDY eval(s) > 0 A moves < MaxMoves
ZVOĽ NÁHODNE KONFLIKTNÚ PREMENNÚ
NAVIAC OPROTI MC → S PRST. p JEJ NASTAV NÁHODNÚ HODNOTU
INAK JEJ NASTAV HODNOTU MINIMALIZUJÚCU eval
AK SA JEJ HODNOTA ZMENILA, ++ moves
return s

SDRW (STEEPEST-DESCENT-RANDOM-WALK)

ZAČNI V NÁHODNOM STAVE s
DOKEDY eval(s) > 0 A moves < MaxMoves
ROZDIEL OPROTI HC → S PRST. p ZVOĽ NÁHODNE KONFLIKTNÚ PREMENNÚ
A PRIRAĎ JEJ NÁHODNÚ HODNOTU
INAK ZVOĽ NAJLEPŠÍ STAV V OKOLÍ s
AK SA STAV ZMENIL, ++ moves
return s

(REALIZÁCIA POMOCOU VÝZNAČ. ATRIBÚTOV)

TABU ZOZNAM = ZOZNAM ZAKÁZANÝCH STAVOV, MÁ FIXNÚ DĹŽKU

ASPIRAČNÉ KRITÉRIUM = ODTABUIZOVANIE STAVU

TABU SEARCH - ZABRÁNIME KRÁTKEMU CYKLENIU

ZAČNI V NÁHODNOM STAVE s, TABU-LIST ← ∅
DOKEDY eval(s) > 0 A moves < MaxMoves
ZVOĽ NAJLEPŠÍ STAV V OKOLÍ s (AKCIA a)
ZAVEĎ AKCIU a DO TABU-LISTU
++ moves
return s

02.02.2009 Po
PETER ČERNO

NOPT046 PROG. S OBMEZ. PODMIENKAMI

SAT A LOK. PREMIEŇANIE

INTENZIFIKÁCIA - VÝBER STAVU Z OKOLIA (HC HEUR., MC HEUR.)
DIVERZIFIKÁCIA - ÚNIK Z LOK. OPTIMA (restart, RW, TABU)

GSAT (ϕ IN CNF, MaxTries, MaxMoves)

for i ← 1 to MaxTries
    ZVOĽ NÁHODNÉ OHODNOTENIE PREM. S
    for j ← 1 to MaxMoves
        AK ϕ PLATÍ V S then return S
        INAK FLIP 1 PREMENNEJ, KT. MAXIMAL. # SPLNENÝCH KLAUZULÍ

HEURISTIKY:
RANDOM-WALK, VÁŽENIE KLAUZULÍ, PRIEMEROVANIE RIEŠENIA

KONEKCIONISTICKÝ PRÍSTUP

SIEŤ JEDNODUCHÝCH PROCESOROV
CIEĽ: NÁJSŤ STABILNÝ STAV → REPREZ. RIEŠENIE
NEURÓNOVÉ SIETE, CELULÁRNE AUTOMATY

GENET

[PRE KAŽDÚ HODNOTU 1 NEURÓN]

PREMENNEJ S DOMÉNOU ZODPOVEDÁ CLUSTER "NEURÓNOV"
DVA NEURÓNY SÚ SPOJENÉ HRANOU (INHIBIČNOU) ⇔ ICH HODNOTY SÚ NEKOMPATIBILNÉ (BINÁRNE CSP)
NA ZAČIATKU JE V KAŽDOM CLUSTERE VYBRANÝ JEDEN AKTÍVNY NEURÓN
NEURÓNY MENIA SVOJE STAVY SYNCHRÓNNE
V KAŽDOM CLUSTERE SA VYBERIE NEURÓN S NAJV. VSTUPOM
VÝPOČET KONČÍ V STABILNOM STAVE

V LOKÁLNOM MINIME SA POSILNIA VÁHY VYBRANÝCH HRÁN

TZV. DYNAMICKÉ LOK. PREHĽADÁVANIE -
MENÍME EVALUAČNÚ FUNKCIU !

SA (SIMULATED ANNEALING) SIMULOVANÉ ŽÍHANIE

OCHLADZOV. KRIVKA

ZAČNI V NÁHODNOM STAVE s
POSTUPNE BER TEPLOTU $T \leftarrow time[t]$
[V CYKLE]
ZVOĽ NÁHODNÚ LOKÁLNU ZMENU s
AK JE ZLEPŠUJÚCA ⇒ OK
AK NIE, POMOC JU APLIKUJ S PRST. $e^{\Delta E / T}$

$\underbrace{eval(s) - eval(next\ s)}$

METROPOLISOVA HEURISTIKA

GENERICKÝ ALGORITMUS LOCAL SEARCH

```
LocalSearch (MaxTries, MaxMoves)
  ZAČNI V NÁHODNOM STAVE s
  for i ← 1 to MaxTries while GCONDITION do
      for j ← 1 to MaxMoves while LCONDITION do
          if eval(s) = 0 then return s
          SELECT n IN NEIGHBOURHOOD(s)
          if ACCEPTABLE(n) then s ← n
      s ← RESTART(s)
  return best s
```

03.02.2009 Tu
PETER ČERNO

NOPT046 PROG. S OBMEZ. PODMIENKAMI

BACKTRACKING REKURZÍVNE

```
BT(X: variables, V: assignments, C: constraints)
  if X = ∅ then return V
  x ← VIBER PREMENNÚ z X
  foreach h ∈ D_x do if C SÚ KONZISTENTNÉ S V ∪ {x/h}
     R ← BT(X - {x}, V ∪ {x/h}, C)
     if R ≠ fail then return R
  return fail
```

BACKTRACKING ITERATÍVNE

```
BT(X: variables, C: constraints)
  i ← 1, D'_i ← D_i
  while 1 ≤ i ≤ n do
     INSTANTIATE-AND-CHECK (i, C)
     if x_i = null then --i else ++i, D'_i ← D_i
  if i = 0 then return fail
  else return {x_1, ..., x_n}

INSTANTIATE-AND-CHECK (i, C)
  while D'_i ≠ ∅ do
     SELECT AND DELETE SOME h FROM D'_i
     x_i ← h
     if {x_1, ..., x_i} ARE CONSISTENT WITH C then return
  x_i ← null
```

ALGORITMUS PREHĽADÁVANIA S NAVRACANÍM POSTUPNE PRECHÁDZA <u>ČIASTOČNÉ KONZISTENTNÉ</u> OHODNOTENIA, DOKÝM NENÁJDE <u>ÚPLNÉ KONZISTENTNÉ</u> OHODNOTENIE

PROBLÉMY BACKTRACINGU

TRASHING : NEDOKÁŽE VYUŽIŤ INFORMÁCIE O DÔVODE (ZDROJI) KONFLIKTU. $A, B, C, D, E :: 1..10$, $A > E$

RIEŠENIE : BACKJUMPING (SKOK NA PÔVODU CHYBY)

REDUNDANTNÁ PRÁCA : NEPAMÄTÁ SI UŽ ODHALENÉ KONFLIKTY MEDZI PRIRADENIAMI PREMENNÝCH

$A, B, C, D, E :: 1..10$, $B + 8 < D$, $C = 5 * E$

RIEŠENIE : BACKMARKING

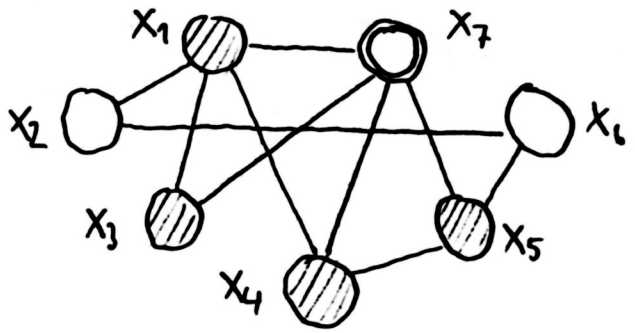
ONESKORENÉ ODHALENIE CHYBY

NESPLNENIE PODMIENKY ODHALÍ AŽ PRI OHODN. JEJ PREM.

$A, B, C, D, E :: 1..10$, $A = 3 * E$

RIEŠENIE : FORWARD CHECKING

BACKJUMPING (GRAFOM RIADENÝ BACKJUMPING / GRAPH-DIRECTED BACKJUMPING)



SLEPÝ UZOL = UZOL, KDE NEMOŽNO VYBRAŤ HODNOTU PREMENNEJ

PRI SKOKU SPÄŤ STAČÍ UVOLNIŤ NIEKTORÝ SLEPÝ UZOL

Z LISTU $X = x_7$ SKOČÍME NA NAJBLIŽŠIEHO PREDCHODCU

Z VNÚTORNÉHO VRCHOLU X SKOČÍME NA NAJBLIŽŠIEHO PREDCHODCU ALE VŠETKÝCH SLEPÝCH UZLOV

$(x_7 \to x_5,\ x_4, x_3, x_1,\ \to x_4,\ x_3, x_1,\ \to x_3,\ x_1)$

OZNAČME $anc(x)$ = PREDCHODCI PREM. X V GRAFE PODMIENOK USPORIADANOM PODĽA PORADIA PRI OHODNOCOVANÍ

03.02.2009 Tu
PETER ČERNO

NOPT046 PROG. S OBMEZ. PODMIENKAMI

NECH SME SA DO UZLU x VRÁTILI Z UZLOV $y_1, \dots, y_k$
AK PRE x NEMÁME DALŠIU HODNOTU, POTOM SKOČÍME NA NAJBLIŽŠIU PREMENNÚ Z:

$$\boxed{anc(x) \cup anc(y_1) \cup \dots \cup anc(y_k) - \{y_1, \dots, y_k\}}$$

BACKJUMPING REKURZÍVNE

```
GraphBJ(X: variables, V: assignments, C: constraints)
  if X = ∅ then return V
  x ← VYBER PREMENNÚ Z X
  conflict ← anc(x)
  foreach h∈D_x do if C SÚ KONZISTENTNÉS V∪{x/h}
    R ← GraphBJ(X-{x}, V∪{x/h}, C)
    if R = fail(JumpSet) then
      if x∉JumpSet then return R
      conflict ← ans(x) ∪ JumpSet - {x}
    else return R
  return fail(conflict)
```

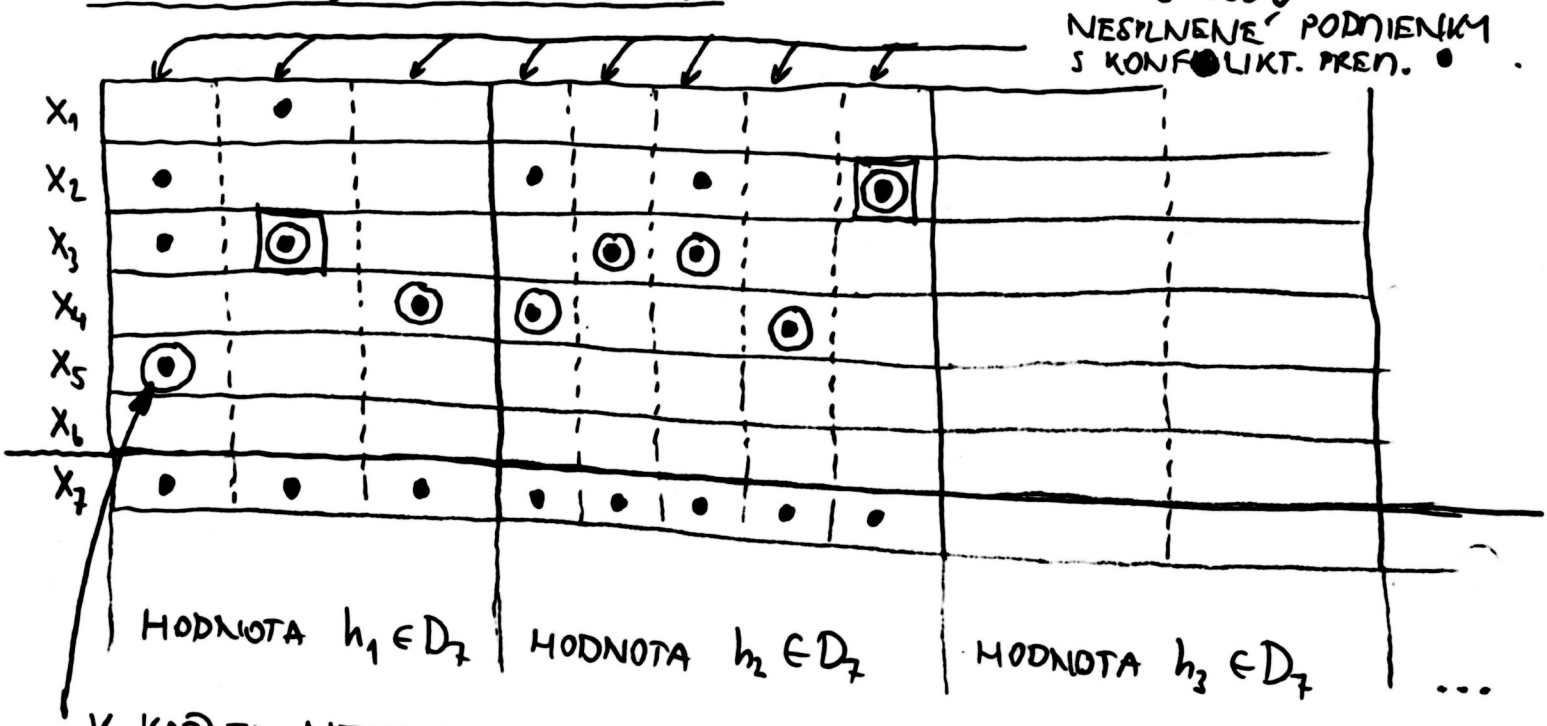
!

BACKJUMPING ITERATÍVNE

```
GraphBJ(X: variables, C: constraints)
  i ← 1, D'_i ← D_i, I_i ← anc(x_i)
  while 1 ≤ i ≤ n do
    INSTANTIATE-AND-CHECK(i, C)
    if x_i = null then   j ← LATEST INDEX IN I_i
      I_j ← I_j ∪ I_i - {x_j},   i ← j
    else ++i, D'_i ← D_i, I_i ← anc(x_i)
  if i=0 then return fail else return {x_1, ..., x_n}
```

!

GASCHNIGOV BACKJUMPING ZJEDNENIE SKOKOV



V KAŽDEJ NESPLNENEJ PODMIENKE SI VYZNAČÍME NAJBLIŽŠIU KONFLIKTNÚ PREMENNÚ ⊙

A PRE KAŽDÚ HODNOTU h_i VYZNAČÍME Z NICH TÚ NAJVZDIALENEJŠIU ⊡

NAPR. AK BY MALO BYŤ $x_7 = h_1$, NESTAČILO BY ZMENIŤ x_4, x_5, x_6 ... MUSÍME ZMENIŤ x_3 !!

AVŠAK KEĎŽE NEVIEME AKÉMU h_i SA BUDE ROVNAŤ x_7 SKOČÍME NA NAJBLIŽŠIE OZNAČENÚ ⊡ (CEZ VŠETKY h_i)

consistent (Labelled, Constraints, Level)

PREDPOKLADÁME, ŽE $x/h/Level \in Labelled$

METÓDA ZISTÍ, ČI SÚ Constraints KONZISTENTNÉ S Labelled, A AK NIE, TAK VRÁTI fail(J), KAM TREBA SKOČIŤ:

PRE KAŽDÚ NEKONZISTENTNÚ $C \in Constraints$ ZISTÍ NAJVÄČŠÍ LEVEL $L < Level$ NEJAKEJ JEJ $x \in C$ A Z TÝCHTO HODNÔT VRÁTI minimum J: ⊡

03.02.2009 Tu
PETER ČERNO

NOPT046 PROG. S OMEZ. PODMIENKAMI

```
consistent(Labelled, Constraints, Level)
  J ← Level, NoConflict ← true
  foreach C in Constraints do if vars(C) ⊆ Labelled
      if C is not satisfied by Labelled then
        NoConflict ← false
        J ← min(J, max{L | x∈vars(C), l<Level, x/v/l ∈ Labelled, })
  if NoConflict then return true else return fail(J)

GBJ(Unlabelled, Labelled, Constraints, PreviousLevel)
  if Unlabelled = ∅ then return Labelled
  x ← VYBER PREMENNÚ z Unlabelled
  Level ← PreviousLevel+1, Jump ← 0
  foreach h∈Dx do
    C ← consistent({x/h/Level} ∪ Labelled, Constr., Level)
    if C = fail(J) then Jump ← max(Jump, J)
    else
      Jump ← PreviousLevel   !!!
      R ← GBJ(Unlabelled − {x}, {x/h/Level} ∪
        ∪ Labelled, Constraints, Level)
      if R ≠ fail(Level) then return R
  return fail(Jump)
```

ITERATÍVNE:

GBJ (X: variables, C: Constraints)

$i \leftarrow 1$, $D'_i \leftarrow D_i$, $jump_i \leftarrow 0$

while $1 \leq i \leq n$ do

$x_i \leftarrow$ SELECT-VALUE (i, C)

V TOMTO MOMENTE JE $jump_i = i-1$

if x_i = null then $i \leftarrow jump_i$

else $++i$, $D'_i \leftarrow D_i$, $jump_i \leftarrow 0$

if $i = 0$ then return fail else return $\{x_1, \dots, x_n\}$

SELECT-VALUE (i, C)

while $D'_i \neq \emptyset$ do

SELECT AND DELETE SOME h FROM D'_i

consistent $\leftarrow$ true, $k \leftarrow 1$

while $k < i$ & consistent do

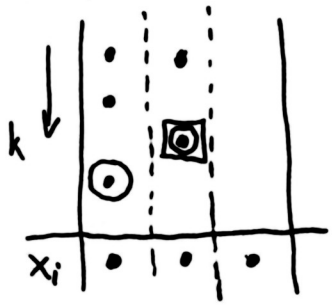
if $k > jump_i$ then $jump_i \leftarrow k$

if $x_i := h$ IS CONSIST. WITH $\{x_1, \dots, x_k\} \cap C$ then $++k$

else consistent $\leftarrow$ false

MAXIMÁLNA CEZ VŠETKY $h \in D_i$

AK NIE JE $x_i := h$ KONZIST. $\Rightarrow$ NÁJDE k AKO ⊙



if consistent then return h $\Longrightarrow$ V TOMTO MOMENTE JE $jump_i = i-1$ A UŽ NIKDY NEKLESNE !!

return null

GRAF. RIADENÝ BACKJUMPING:

RIADI SA IBA ŠTRUKTÚROU SIETE PODMIENOK

DOKÁŽE REALIZOVAŤ NIEKOĽKO SPÄTNÝCH KROKOV

GASCHNIGOV BACKJUMPING:

BERIE V ÚVAHU, KT. PODMIENKY SKUTOČNE SPÔSOBILI KONFLIKT

SKOČÍ IBA JEDEN SKOK !

KONFLIKTOM RIADENÝ BACKJUMPING (CBJ):

SPOJENIE VÝHOD OBOCH METÓD, VRACIANE KONFL. MNOŽINU

03.02.2009 Tu
PETER ČERNO

NOPT046 PROGR. S OBMEZ. PODMIENKAMI

PROBLÉM BACKJUMPINGU : ZAHADZUJE UŽ UROBENÚ PRÁCU !

DYNAMICKÝ BACKTRACKING =

BACKJUMPING + PAMÄTANIE SI DÔVOD KONFLIKTU +
PRENOS DÔVODU KONFLIKTU + ZMENA POREDIA PREMENNÝCH

DB (Variables, Constraints)
  Labelled ← ∅, Unlabelled ← Variables
  while Unlabelled ≠ ∅ do
    ZVOLÍME $x \in$ Unlabelled
    $Values_x \leftarrow D_x$ - NEKONZISTENTNÉ HODNOTY
    if $Values_x = \emptyset$ then
      NECH E JE VYSVETLENIE KONFLIKTU
      (MNOŽINA KONFLIKTNÝCH PREMENNÝCH)
      if $E = \emptyset$ then failure
      else NECH Y JE NAJVIAC NEDÁVNA PREM. V E
        ODSTRÁNIME Y Z Labelled S VYSVETLENÍM $E - \{Y\}$
        ODSTRÁNIME ∀ VYSVETLENIA, KT. OBSAHUJÚ Y
    else
      VYBERIEME $h \in Value_x$
      Unlabelled ← Unlabelled − {X}
      Labelled ← Labelled ∪ {X/h}
  return Labelled

REDUNDANTNÁ PRÁCA, NAPR.: A,B,C,D :: 1..10, A+8 < C, B=5*D
ZMENA B NEOVPLYVNÍ HODNOTU C

BACKMARKING

MARK (X, V) ... NAJVZDIALENEJŠÍ KONFLIKT S X := V

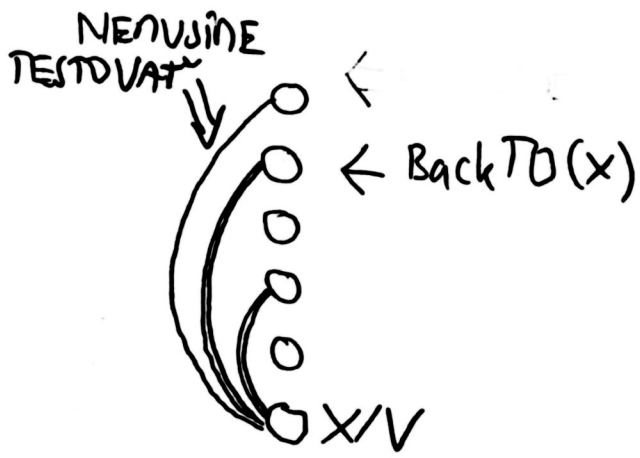
BACKTO (X) .. MIESTO NÁVRATU OD POSLEDNÉHO OHODNOT. X
PREMENNÉ NAD BACKTO (X) NEZMENILI HODNOTU
OD POSLEDNÉHO MOMENTU OHODNOCOVANIA X

AK MARK (X, h) < BACKTO (X),
POTOM X := h JE STÁLE NEKONZISTENTNÉ
S PREMENNOU MARK (X, h), LEBO TÁ SA NEZMENILA

POZNÁMKA: BACKMARKING MOŽNO KOMBINOVAŤ S BACKJUMPINGOM

TEST KONZISTENCIE (BACKMARKING)

consistent (X/V, Labelled, Constraints, Level)
  foreach Y/VY/LY such that LY ≥ BackTo(X)
    BERIEME IBA PREMENNÉ, KT. SA MOHLI ZMENIŤ
    IDEME V RASTÚCOM POREDÍ PODĽA LY
    if X/V is not compatible with Y/VY then
      Mark(X,V) ← LY
      return fail
  Mark(X,V) ← Level - 1
  return true



03.02.2009 Tu
PETER ČERNO

NOPTD46 PROG. S OBMEZ. PODMIENKAMI

```
BM (Unlabelled, Labelled, Constraints, Level)
  if Unlabelled = ∅ then return Labelled
  X ← VÝBER PREMENNÚ Z Unlabelled        (PEVNÉ PORADIE!)
  foreach h ∈ Dx do
    if Mark(X,h) ≥ BackTo(X) then
      if consistent(X/h, Labelled, Constraints, Level)
        R ← BM(Unlabelled - {X},
          Labelled ∪ {X/h/Level}, Constraints, Level+1)
        if R ≠ fail then return R        (MÁME RIEŠENIE)
  BackTo(X) ← Level - 1
  foreach Y in Unlabelled do
    BackTo(Y) ← min(Level-1, BackTo(Y))
  return fail
```

VRACIAME SA K PREDCH. PREMENNEJ, NECH VŠETCI VEDIA O NÁVRATE

04. 02. 2009 We
PETER ČERNO

NOPT046 PROG. S OBMEZ. PODMIENKAMI

KONZISTENČNÉ TECHNIKY

VRCHOL, VRCHOLOVÁ KONZISTENCIA (NODE CONSISTENCY)
KAŽDÁ HODNOTA Z AKT. DOMÉNY SPĽŇA $\forall$ UNÁRNE PODMIENKY
$\Rightarrow$ ALGORITMUS NC $\rightarrow$ SPRAVÍ CSP VRCHOLOVO KONZ.

HRANA (V_i, V_j) JE <u>HRANOVO KONZISTENTNÁ</u> (ARC CONSISTENT)
$\Leftrightarrow \forall x \in D_i \; \exists y \in D_j : V_i := x, V_j := y$ SPĽŇA $\forall$ BIN. PODM. NAD V_i, V_j.
CSP JE HRANOVO KONZISTENTNÝ $\Leftrightarrow$ KAŽDÁ (V_i, V_j) JE HR. KONZ.

<u>REVÍZIA HRANY</u>:

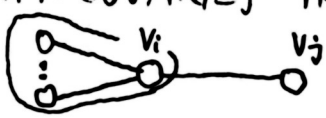
function REVISE(i,j) returns
- true $\Leftrightarrow$ DOŠLO K VYMAZANIU NEJAKEJ HODNOTY Z D_i
- false

AC-1(G)

```
repeat
   changed ← false
   foreach arc (i,j) ∈ G do
      changed ← changed ∨ REVISE(i,j)
      if D_i = ∅ then STOP WITH FAIL
until changed = false
```

NEFEKTÍVNY!

[PO REVISE(i,j)]
AKÉ HRANY SA MAJÚ ZREVIDOVAŤ PO ZMENŠENÍ DOMÉNY?
$\rightarrow$ HRANY VEDÚCE DO INKRIMINOVANEJ PREMENNEJ
$\rightarrow$ OKREM HRANY (V_j, V_i)



ZOVŠEOBECNENÝ WALTZOV OHODN. ALG.

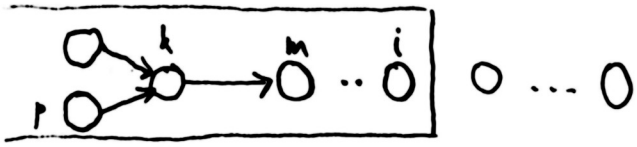
AC-2(G)

$\boxed{\text{for } i \leftarrow 1 \text{ to } n \text{ do}}$

$Q \leftarrow \{(i,j) \mid (i,j) \in arcs(G),\ j < i\}$

$Q' \leftarrow \{(j,i) \mid \text{—''—}\}$ PRE RE-REVÍZIU

while $Q \neq \emptyset$ do



while $Q \neq \emptyset$ do

SELECT AND DELETE (k,m) FROM Q

if REVISE(k,m) then

$Q' \leftarrow Q' \cup \{(p,k) \mid (p,k) \in arcs(G),\ p \leq i,\ p \neq m\}$

if $D_k = \emptyset$ then STOP WITH FAIL

$Q \leftarrow Q',\ Q' \leftarrow \emptyset$

AC-3(G)

$Q \leftarrow \{(i,j) \in arcs(G),\ i \neq j\}$

while $Q \neq \emptyset$ do

SELECT AND DELETE (k,m) FROM Q

if REVISE(k,m) then

$Q \leftarrow Q \cup \{(p,k) \in arcs(G),\ p \neq k, m\}$

if $D_k = \emptyset$ then STOP WITH FAIL

ASI NAJPOUŽÍVANEJŠÍ, ALE <u>NIE OPTIMÁLNY</u>!

PRI KAŽDEJ REVÍZII TESTUJEME OPAKOVANE MNOŽSTVO DVOJÍC HODNÔT NA KONZISTENCIU

04.02.2009 We
PETER ČERNO

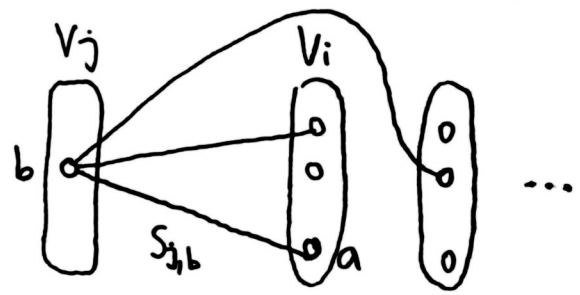
NOPT046 PROG. S OBMEZ. PODMIENKAMI

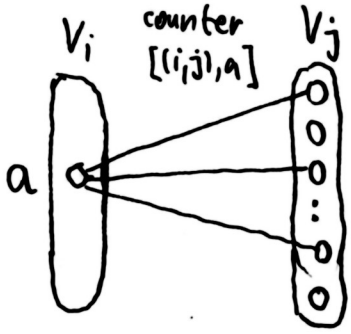
PODPORA PRE $a \in D_i$ JE $\{ \langle j, b \rangle \mid b \in D_j, (a,b) \in C_{ij} \}$

UDRŽUJEME ZOZNAM HODNÔT, KTORÉ PODPORUJEME A POČET VLASTNÝCH PODPÔR

$S_{j,b}$ = MNOŽINA DVOJÍC $\langle i, a \rangle$, PRE KTORÉ JE $\langle j, b \rangle$ PODP.

counter$[(i,j), a]$ = POČET PODPÔR, KTORÉ MÁ HODNOTA $a \in D_i$ U PREMENNEJ V_j






```
INITIALIZE (G)
  Q ← ∅, S ← ∅
  foreach (Vi,Vj) foreach a in Di
    total ← 0
    foreach b in Dj do if (a,b) ∈ Cij then
      ++total;   Sj,b ← Sj,b ∪ { <i,a> }
    counter[(i,j),a] ← total
    if counter[(i,j),a] = 0 then
      DELETE a FROM Di
      Q ← Q ∪ { <i,a> }
  return Q
```

$a \in D_i$ NEMÁ PODPORU V D_j PRI PODM. C_{ij}

Q .. FRONTA HODNÔT $\langle i, a \rangle$, KTORÉ BOLI ODSTRÁNENÉ Z DOMÉNY (D_i), PRETOŽE NEMAJÚ PODPORU V NEJAKEJ PODMIENKE C_{ij}

AC-4(G)

$Q \leftarrow$ INITIALIZE(G)

while $Q \neq \emptyset$ do

SELECT AND DELETE ANY $\langle j, b \rangle \in Q$

foreach $\langle i, a \rangle \in S_{j,b}$ do

-- counter[(i,j), a]

if counter[(i,j), a] = 0 & $a \in D_i$ then

DELETE a FROM D_i

if $D_i = \emptyset$ then STOP WITH FAIL

$Q \leftarrow Q \cup \{\langle i, a \rangle\}$

<u>OPTIMÁLNY V NAJHORŠOM PRÍPADE</u>, V PRIEMERE NIE JE AŽ TAKÝ DOBRÝ, VYSOKÉ PAMÄŤOVÉ NÁROKY

AC-5 - GENERICKÝ, SÉMANTIKA PODMIENOK

AC-6 - ZLEPŠUJE PRIEM. ČAS A PAM. NÁR. AC-4, DRŽÍ IBA 1 PODPORU

AC-7 - ZALOŽENÝ NA PODPORÁCH

PREČO JE AC-3 NEEFEKTÍVNY? KVÔLI PODMIENKE V REVISE:

IF THERE IS NO SUCH $Y \in D_j$ S.T. (X, Y) IS CONSISTENT

$\Rightarrow$ <u>AC-3.1</u> PRE KAŽDÚ HODNOTU SI PAMÄTÁM POSLEDNÚ NÁJDENÚ PODPORU

EXIST((i,x), j)

$Y \leftarrow$ last[(i,x), j]

if $Y \in D_j$ then return true $\leftarrow$ x NESTRATILA PODPORU Y

while $Y \leftarrow$ next$(Y, D_j) \neq$ NULL do

if $(x, Y) \in C_{i,j}$ then

last[(x,i), j] $\leftarrow Y$, return true

return false

04.02.2009 We
PETER ČERNO

NOPT046 PROG. S OMEZ. PODMIENKAMI

AC-2001 (G) (AC-8) OPTIMÁLNY AC-3

```
Q ← nodes(G)
while Q ≠ ∅ do
    SELECT AND DELETE j ∈ Q
    foreach (i,j) ∈ arcs(G) do
        if REVISE2001(i,j) then
            if Di = ∅ then return FAIL
            Q ← Q ∪ {i}
```

```
REVISE2001(i,j)
deleted ← false
foreach x ∈ Di do
    if ¬EXIST((i,x),j) then
        DELETE x FROM Di
        deleted ← true
return deleted
```

SMEROVÁ HRANOVÁ KONZISTENCIA

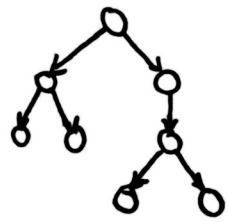
CSP JE SMEROVO HRANOVO KONZISTENTNÉ (DIRECTIONAL ARC CONSISTENT) PRI DANOM USP. PREMENNÝCH $\Leftrightarrow$ KAŽDÁ (i,j) $i<j$ JE HRANOVO KONZ.

```
DAC-1(G)
for j ← n downto 1 do
    foreach (i,j) ∈ arcs(G), i < j do
        REVISE(i,j)
        if Di = ∅ then STOP WITH FAIL
```

CSP AC $\Rightarrow$ CSP DAC

AKO USPORIADAŤ VRCHOLY ?

PRI STROME - OD KOREŇA. $\Rightarrow$ DAC GARANTUJE, ŽE VŽDY MOŽNO NÁJSŤ HODNOTU POTOMKA KOMPAT. S RODIČOM



◁ CSP JE <u>HRANOVO KONZIST.</u> ⇔ PRE DANÉ USP. PRED. JE <u>SMEROVO HRANOVO KONZ.</u> V <u>OBOCH SMEROCH</u>

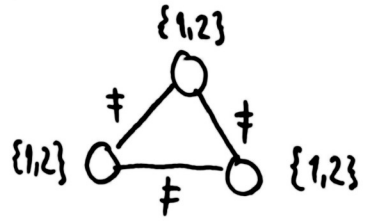
AK JE DAC APLIKOVANÉ NA STROMOVÉ CSP NAJSKÔR PRI USP. OD KOREŇA, A POTOM OPAČNE, POTOM ZÍSKAME HRANOVÚ KONZISTENCIU.

KAŽDÁ HODNOTA MÁ PODPORU VO VŠETKÝCH POTOMKOCH (1. BEH) I V RODIČOVI (2. BEH)

JE AC DOSTATOČNÉ?

(1) DOSTANEME RIEŠENIE? NIE!

(2) VIEME ASPOŇ, ŽE EXISTUJE? NIE!



AC <u>ZMENŠÍ PREHĽADÁVANÝ PRIESTOR</u>

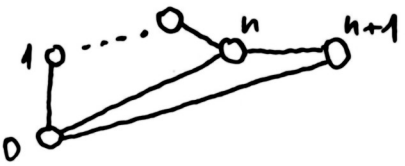
KONZISTENCIA PO CESTE

CESTA $(V_0, V_1, \dots, V_m)$ JE <u>KONZISTENTNÁ</u> (<u>PATH CONSISTENT</u>)
⇔ $\forall x_0 \in D_0\ x_m \in D_m,\ (x,y) \in C_{0,m}\ \exists x_i \in D_i\ \forall i \in \{1, \dots, m-1\}$:
$(x_i, x_{i+1}) \in C_{i,i+1}\ \forall i \in \{0, 1, \dots, m-1\}$.

CSP JE KONZISTENTNÉ PO CESTE ⇔ ∀CESTY SÚ KONZIST.

<u>VETA</u>: CSP JE PC ⇔ KAŽDÁ CESTA DĽŽKY 2 JE PC.

⇐)



CSP PC ⇒ CSP AC : (i,j) JE AC ⇔ (i,j,i) JE PC

$X, Y, Z :: 1..2,\ X \neq Y,\ X \neq Z,\ Y \neq Z$ JE AC, ALE NIE PC

AC VYRADÍME HODNOTY Z DOMÉN, PC DVOJICE HODNÔT!

PC ROBÍ VŠETKY REVÁCIE IMPLICITNE

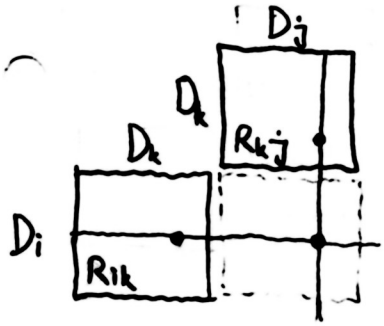
04.02.2009 We
PETER ČERNO

NOPT046 PROG. S OBMEZ. PODMIENKAMI

REPREZENTÁCIA PODMIENOK

PODMIENKA R_{ij} REPREZENTOVANÁ BIN. MATICOU $A \in \{0,1\}^{D_i \times D_j}$ $A[x_i, x_j] = 1 \Leftrightarrow (x_i, x_j) \in R_{ij}$

PRIENIK PODMIENOK R_{ij} A R'_{ij} : $R_{ij} \,\&\, R'_{ij}$
ZLOŽENIE PODMIENOK R_{ik} A R_{kj} : $R_{ik} * R_{kj}$



$(x_r, x_t) \in R_{ik} * R_{kj} \Leftrightarrow$
$\exists x_s \in D_k : (x_r, x_s) \in R_{ik} \,\&\, (x_s, x_t) \in R_{kj}$

$R_{ij} = R_{ji}^T$, R_{ii} DIAGONÁLA REPR. DOMÉNU D_i

$\text{REVISE}(i,j) \equiv R_{ii} \leftarrow R_{ii} \,\&\, (R_{ij} * R_{jj} * R_{ji})$

AKO UROBIŤ CESTU (i,j,k) KONZISTENTNÚ ?
$R_{ij} \leftarrow R_{ij} \,\&\, (R_{ik} * R_{kk} * R_{kj})$

```
PC-1(Vars, Constraints)
n ← |Vars|, Y^n ← Constraints
repeat
    Y^0 ← Y^n
    for k=1 to n for i=1 to n for j=1 to n do
        Y^k_ij ← Y^{k-1}_ij & (Y^{k-1}_ik * Y^{k-1}_kk * Y^{k-1}_kj)
until Y^n = Y^0
Constraints ← Y^0
```

ZÁSADNÝ PROBLÉM PC-1: PRI ZMENE DOMÉNY PODMIENKY SA ZNOVA REVIDUJÚ $\forall$ CESTY

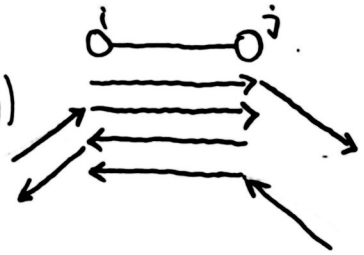
REVISE-PATH (i, k, j)
  $Y_{ij} \leftarrow Y_{ij} \,\&\, (Y_{ik} * Y_{kk} * Y_{kj})$
  return true $\Leftrightarrow$ Y_{ij} HAS CHANGED

PRENOŽE $Y_{ji} = Y_{ij}^T$ STAČÍ UVAŽOVAŤ IBA CESTY (i,k,j) $i \leq j$

NECH PRI REVÍZII (i,k,j) DOŠLO K ZMENE DOMÉNY Y_{ij}

SITUÁCIA a: $i < j$

$S_a(i,j)$



CESTY (i,j,j) ANI (i,i,j) NIE JE NUTNÉ REVIDOVAŤ

$|S_a(i,j)| = 2n-2$

SITUÁCIA b: $i = j$

$S_b(i,j)$ JE NUTNÉ ZREVIDOVAŤ CESTY OBSAHUJÚCE VO VNÚTRI i OKREM (i,i,i) A (k,i,k): $Y_{kk} \leftarrow Y_{kk} \,\&\, (Y_{ki} * Y_{ii} * Y_{ik})$

$|S_b(i,j)| = \frac{n(n+1)}{2} - 2$

$= Y_{ii} \,\&\, (Y_{ik} * Y_{kk} * Y_{ki})$

PC-2 (G)
  $n \leftarrow |nodes(G)|$
  $Q \leftarrow \{ (i,k,j) \mid 1 \leq i \leq j \leq n \;\&\; k \neq i \;\&\; k \neq j \}$
  while $Q \neq \emptyset$ do
    SELECT AND DELETE (i,k,j) FROM Q
    if REVISE-PATH (i,k,j) then
      $Q \leftarrow Q \cup$ RELATED-PATHS (i,k,j)

RELATED-PATHS (i,k,j)
  if $i < j$ then return $S_a(i,j)$ else return $S_b(i,j)$

04.02.2009 We
PETER ČERNO

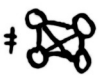
NOPT046 PROG. S OBMEZ. PODMIENKAMI

SMEROVÁ KONZISTENCIA PO CESTE

CSP JE SMEROVO KONZISTENTNÉ PO CESTE (DIRECTIONAL PATH CONSISTENT) PRI DANOM USP. PREMENNÝCH $\Leftrightarrow$
$\forall (i,k,j)$ $i,j \leq k$ SÚ KONZISTENTNÉ
MÔŽME PRIDAŤ PODMIENKU $i \leq j$

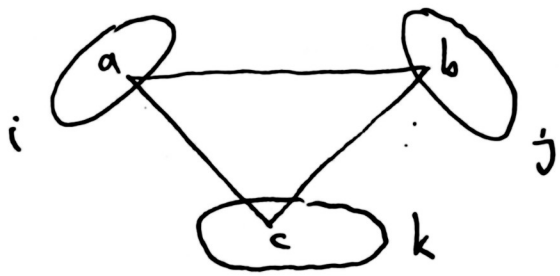
```
DPC-1 (Vars, Constraints)
n ← |Vars| , E ← {(i,j) | i<j & C_ij ∈ Constraints}
for k = n downto 1  for i=1 to k  for j= i to k
  if (i,k)∈E & (j,k)∈E then
      C_ij ← C_ij & (C_ik * C_kk * C_kj) ,  E ← E ∪ {(i,j)}
```

PC $\Rightarrow$ DPC
DPC $\not\Rightarrow$ AC

PC: VYSOKÉ PAMÄŤOVÉ NÁROKY: EXTENZION. REPR. PODMIENOK
VÝKON/CENA JE HORŠÍ NEŽ U AC
PRIDÁVA HRANY (PODMIENKY)
NIE JE DOSTATOČNÉ  D={1,2,3}

OHRANIČENÁ KONZISTENCIA PO CESTE

PC HRANY SA TESTUJÚ IBA AK VYRADENIE DVOJICE MÔŽE VIESŤ K VYRADENIU NIEKTORÉHO Z PRVKOV Z DOMÉNY PR.



VRCHOL i JE <u>OHRANIČENE KONZISTENTNÝ PO CESTE</u>
(RESTRICTED PATH CONSISTENT) $\Leftrightarrow$

(1) KAŽDÁ HRANA VEDÚCA Z i JE AC

(2) PRE KAŽDÉ $a \in D_i$ PLATÍ:

KEĎ JE b JEDINÁ PODPORA VO VRCHOLE j, POTOM V KAŽDOM VRCHOLE k SPOJENOM S i A j EXISTUJE HODNOTA c TAK, ŽE (a,c), (b,c) SÚ KOMPATIBILNÉ S PRÍSL. PODMIENKAMI

INITIALIZE (G) — AKO U AC-4

$Q_{AC} \leftarrow \emptyset$, $Q_{PC} \leftarrow \emptyset$, $S \leftarrow \emptyset$

•

•

•

```
if counter[(i,j), a] = 0 then
    DELETE a FROM D_i,  Q_AC ← Q_AC ∪ {<i,a>}
else if counter[(i,j), a] = 1 then
    foreach k, (i,k), (k,j) ∈ arcs(G) do
        Q_PC ← Q_PC ∪ {(<i,a>, j, k)}
return (Q_AC, Q_PC)
```

PRUNE (Q_{AC}, Q_{PC}) — AC KONZISTENCIA AC-4

```
while Q_AC ≠ ∅ do
    SELECT AND DELETE ANY <j,b> ∈ Q_AC
    foreach <i,a> ∈ S_{j,b} do
        -- counter[(i,j), a]
        if counter[(i,j), a] = 0 then
            DELETE a FROM D_i
            Q_AC ← Q_AC ∪ {<i,a>}
```

*

04.02.2009 We
PETER ČERNO

NOPT046 PROG. S OBMEZ. PODMIENKAMI

```
* else if counter[(i,j), a] = 1 then
    foreach k, (i,k), (k,j) ∈ arcs(G) do
        Q_PC ← Q_PC ∪ {(<i,a>, j, k)}
else
    foreach k, (i,k), (k,j) ∈ arcs(G) do
        if counter[(i,k), a] = 1 then
            Q_PC ← Q_PC ∪ {(<i,a>, k, j)}
return Q_PC    (Q_AC = ∅)
```

NEW

```
RPC(G)
  (Q_AC, Q_PC) ← INITIALIZE(G)
  Q_PC ← PRUNE(Q_AC, Q_PC)
  while Q_PC ≠ ∅ do
      SELECT AND DELETE ANY (<i,a>, j, k) ∈ Q_PC
      if a ∈ D_i then
          {<j,b>} = {<j,x> ∈ S_ia | x ∈ D_j}
          if {<k,c> ∈ S_ia ∩ S_j,b , c ∈ D_k} = ∅ then
              counter[(i,j), a] ← 0
              DELETE a FROM D_i
              Q_PC ← PRUNE({<i,a>}, Q_PC)
```

PRVÝ BEH AC

JEDINÁ PODPORA PRE a

OPAKUJEME AC

ZHRNUTIE:

VRCHOLOVÁ, HRANOVÁ KONZISTENCIA (AC)

REVISE(i,j)

AC-1 V KAŽDEJ ITERÁCII REVIDUJE $\forall$ HRANU, DOKEDY SA MENIA D
NEFEKTÍVNY!

AC-2 for $i \leftarrow 1$ to n

MÁME 2 FRONTY: Q:

Q' PRE REVÍZIU

AC-3 1 FRONTA, PO REVISE(k,m) PRIDÁME $\forall (p,k)$, $p \neq k,m$

AC-4 $S_{j,b} = \{\langle i,a\rangle$ ... PRE KTORÉ JE b PODPORA$\}$

counter$[(i,j),a]$ = POČET PODPOR PRE a

INITIALIZE(G) .. INIC. S, counter, returns $Q = \{\langle i,a\rangle$ BEZ PODPORY$\}$

AC-3.1 EXIST$((i,x),j)$ PAMÄTÁ SI last$[(i,x),j]$ POSL. PODPORU

AC-2001 $Q \leftarrow$ nodes(G) REVISE2001 → POUŽÍVA EXIST$((i,x),j)$

SMEROVÁ HRANOVÁ KONZISTENCIA, STROMOVÉ CSP

DAC-1 for $j \leftarrow n$ downto 1

KONZISTENCIA PO CESTE (PATH CON.), STAČÍ UVAŽOVAŤ CESTY DĽ. 2

REPREZENTÁCIA PODMIENOK

REVISE$(i,j) \equiv R_{ii} \leftarrow R_{ii} \,\&\, (R_{ij} * R_{jj} * R_{ji})$

REVISE-PATH$(i,k,j) \equiv R_{ij} \leftarrow R_{ij} \,\&\, (R_{ik} * R_{kk} * R_{kj})$

PC-1 POČÍTA Y^k_{ij} $\forall k,i,j$ DÓKIM DOSTÁVAME NOVÉ Y^n

PC-2 k REVISE-PATH(i,k,j) PRIDÁ DO Q RELATED-PATHS(i,k,j)

SMEROVÁ KONZISTENCIA PO CESTE

DPC-1 for $k \leftarrow n$ downto 1 $i \leftarrow 1$ to k, $j \leftarrow i$ to k ← KVÔLI SYMETRIÁM

POZOR PO REVISE(i,k,j) TREBA PRIDAŤ HRANU (i,j)!

OHRANIČENÁ KONZISTENCIA PO CESTE

INITIALIZE(G) .. VYCHÁDZA Z AC-4, $Q_{PC} = \{(\langle i,a\rangle, j, k)$
a má 1 PODPORU v j, $(i,k),(j,k) \in$ arcs(G)

PRUNE(Q_{AC}, Q_{PC}) .. AC-4 OBOHATENÁ O ROZŠ. Q_{PC}, VRACIA Q_{PC}

RPC .. POPRI PRUNE (AC) OŠETRUJE Q_{PC}

05. 02. 2009 Th
PETER ČERNO

NOPT046 PROG. S OMEZ. PODMIENKAMI

k-KONZISTENCIA

CSP JE <u>k-KONZISTENTNÉ</u> $\Leftrightarrow$ ĽUBOVOĽNÉ KONZISTENTNE OHODNOTENIE k-1 RÔZNYCH PREMENNÝCH MÔŽEME ROZŠÍRIŤ DO ĽUBOVOĽNEJ k-TEJ PREMENNEJ

<u>SILNO k-KONZISTENTNÝ</u> $\Leftrightarrow$ j-KONZIST. PRE $\forall j \leq k$.

3-KONZISTENTNÝ
NIE JE 2-KONZISTENTNÝ

SILNO KONZISTENTNÝ PRE $\forall k < n$

<u>CSP VYRIEŠIME BEZ NAVRACANIA</u>, AK PRI NEJAKOM USP. PREMENNÝCH MÔŽEME PRE KAŽDÚ PREMENNÚ VŽDY NÁJSŤ HODNOTU KOMPATIBILNÚ S UŽ OHODN. PREMENNÝMI

ŠÍRKA GRAFU = MINIMUM ZO ŠÍROK $\forall$ JEHO USP. GRAFOV

Min Width Ordering, $(G=(V,E))$

POSTUPNE ODSTRÁNIME Z G VRCHOLY S NAJM. STUPŇOM
return TÚTO POSTUPNOSŤ ODSTR. VRCHOLOV, ALE OBRÁTENE!

VETA: NA GRAF ŠÍRKY w NÁM STAČÍ SILNÁ (w+1)-KONZIST.

AC (= SILNÁ 2-KONZISTENCIA) STAČÍ NA STROMOVÉ GRAFY (w=1).
→ DOKONCA DAC

POZOR! PC MENÍ ŠTRUKTÚRU GRAFU

CSP JE <u>SMEROVO k-KONZISTENTNÝ</u> PRI NEJAKOM USP. PREMENNÝCH $\Leftrightarrow$ ĽUBOVOĽNÉ KONZISTENTNE OHODN. k-1 RÔZNYCH PREMENNÝCH MÔŽEME ROZŠÍRIŤ DO ĽUBOVOĽNEJ k-TEJ PREM., KTORÁ JE ZA TOUTO (k-1)-TICOU.

ADAPTÍVNA KONZISTENCIA

ZAISTÍ SMEROVÚ i-KONZISTENCIU, KDE i SA MENÍ PODĽA ŠÍRKY SPRAC. VRCHOLU

VRCHOLY SÚ SPRACOVÁVANÉ PROTI SMERU USPORIADANIA (HRANY PRIBÚDAJÚ V DOTERAZ NESPRAC. ČASTI)

CSP JE <u>(i,j)-KONZIST.</u> $\Leftrightarrow$ ĽUBOVOĽNÉ KONZ. OMODN. i RÔZNYCH PREM. MÔŽEME ROZŠÍRIŤ DO ĽUBOVOĽNEJ MNOŽINY j ALEBO MENEJ NEŽ j PREMENNÝCH

<u>SILNO (i,j)-KONZ.</u> $\Leftrightarrow$ (k,j)-KONZ. PRE $\forall k \leq i$

PRI (i,j)-KONZ. PRACUJEME S i-TICAMI $\Rightarrow$ VEĽKÉ PAM. NÁROKY

POLOŽME $i := 1$ A ZVÄČŠUJME j

(1,k)-KONZ. = <u>INVERZNÁ k-KONZ.</u>

INV. HRAN. KONZ. = (1,1)-KONZ. = AC = HRANOVÁ KONZ.

(1,2)-KONZ. = <u>INVERZNÁ KONZIST. PO CESTE</u> (PIC)

CSP JE <u>INVERZNE KONZIST. PRE OKOLIE (NIC)</u> $\Leftrightarrow$ PRE ĽUBOVOĽNÚ PREM. X A HODNOTU h $\exists$ KONZIST. RIEŠENIE PROBLÉMU VZNIKNUTÉHO Z OKOLIA X

```
NIC(G=(V,E))
Q ← V
while Q ≠ ∅ do
    V ← SELECT AND DELETE FROM Q
    foreach h ∈ D_V do
        if NO SOLUTION FOR NEIGH(X) COMPATIBLE WITH H then
            REMOVE h FROM D_V
    if D_V HAS CHANGED then Q ← Q ∪ NEIGH(X)
```

05. 02. 2009 Th
PETER ČERNO

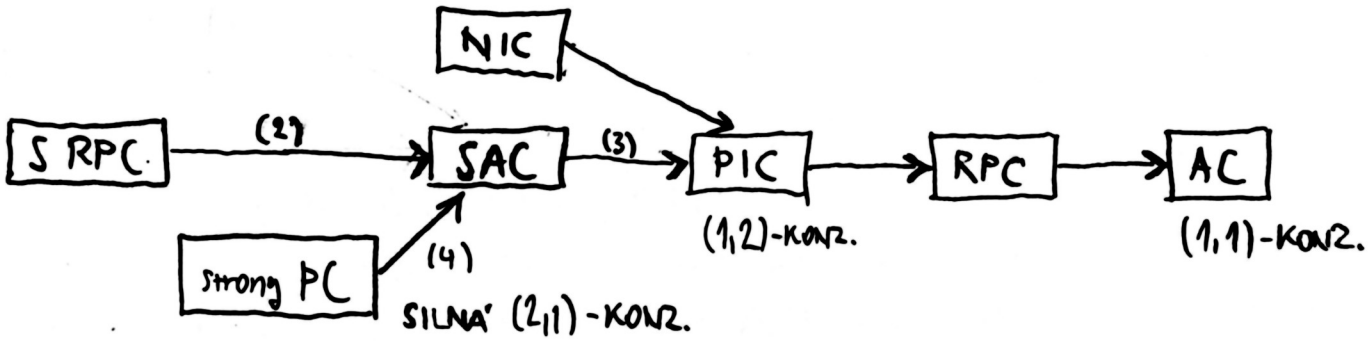
NOPT046 PROG. S OBMEZ. PODMIENKAMI

CSP JE <u>BODOVO A-KONZISTENTNÝ</u> (SINGLETON A-CON.) KDE A JE ĽUBOVOĽNÁ KONZ. TECHNIKA ⇔ PRE ĽUBOVOĽNÚ HODNOTU h PREM. X JE PROBLÉM PO X:=h A-KONZIST.

+ PODOBNE AKO AC, NIC, RPC IBA ODSTRAŇUJE HODNOTY Z DOMÉN
+ ĽAHKÁ IMPLEMENTÁCIA
− ČASOVÁ ZLOŽITOSŤ

(1) BODOVÁ A-KONZ. ≥ A-KONZ.
(2) A-KONZ. ≥ B-KONZ. ⇒ BOD. A-KONZ. ≥ BOD. B-KONZ.
(3) BOD. (i,j)-KONZ. > (i, j+1)-KONZ. (SAC > PIC)
(4) SILNÁ (i+1, j)-KONZ. > BOD. (i,j)-KONZ. (PC > SAC)

NC = 1-KONZ.
AC = (SILNÁ) 2-KONZ. = (SILNÁ) (1,1)-KONZ.
PC = SILNÁ 3-KONZ. = SILNÁ (2,1)-KONZ.
PIC = (1,2)-KONZ.



<u>HODNOTA</u> h PREMENNEJ V JE <u>ZOBECNENE HRAN. KONZ.</u> (GENERALIZED ARC CONSISTENT) VZHĽADOM K PODMIENKE P ⇔ EXISTUJÚ HODNOTY V AKT. DOMÉNACH OSTATNÝCH PREM. V P TAKÉ, ŽE DOMROMADY S h SPĽŇAJÚ P.

<u>PREMENNÁ</u> V JE <u>ZOBECNENE HRAN. KONZ.</u> VZHĽADOM K P ⇔ ∀ h ∈ D_V SÚ GAC VZHĽADOM K P

<u>PODMIENKA</u> P JE <u>Z.H.K.</u> ⇔ ∀ JEJ PREMENNÉ SÚ GAC

AC-3 PRE NEBINÁRNE PODMIENKY
PODMIENKU CHÁPEME AKO SADU PROPAGAČNÝCH METÓD, Z KT. KAŽDÁ UROBÍ JEDNU PREMENNÚ GAC

GAC-3

$Q \leftarrow \{X_s \to Y$ METHOD FOR SOME CONSTRAINT IN $G\}$
while $Q \neq \emptyset$ do
  SELECT AND DELETE $A_s \to B$ FROM Q
  if REVISE$(A_s \to B)$ then
    if $D_B = \emptyset$ then STOP WITH FAIL
    $Q \leftarrow Q \cup \{X_s \to Y \mid B \in X_s\}$

SLABŠIA VERZIA GAC: GAC VLASTNOSŤ JE POŽADOVANÁ IBA PO OKRAJOVÝCH HODNOTÁCH KAŽDEJ DOMÉNY

PREMENNÁ V JE <u>OKRAJOVO KONZ.</u> (BOUNDS CONSISTENT) VZHĽADOM K P $\Leftrightarrow$ OKRAJE D_V SÚ GAC VZHĽADOM K P

ZHRNUTIE

[SILNÁ] k-KONZISTENCIA
RIEŠENIE CSP BEZ NAVRACANIA, ŠÍRKA GRAFU
SMEROVÁ k-KONZISTENCIA PRI NEJAKOM USP.
ADAPTÍVNA KONZISTENCIA - SMEROVÁ i-KON. PODĽA ŠÍRKY, IDEME ODZADU
[SILNÁ] (i,j)-KONZISTENCIA, INV. k-KONZ., PIC PO CESTE
INVERZNÁ KONZ. PRE OKOLIE (NIC)
BODOVÁ A-KONZISTENCIA (SINGLETON ..) - 4 VLASTNOSTI
GRAF POROVNANIA KONZ. TECHNÍK !
NEBINÁRNE PODMIENKY - ZOBECNENÁ HRANOVÁ KONZ. (GAC)
PODMIENKA = SÚBOR PROPAGAČNÝCH METÓD
SLABŠIA VERZIA GAC - OKRAJOVÁ KONZ. VZHĽADOM K P

05.02.2009Th
PETER ČERNO

NOPT046 PROG. S OBMEZ. PODMIENKAMI

PREHĽADÁVANIE PR. RIEŠ. – ÚPLNÉ, POMALÉ (EXPONENT.)
KONZISTENČNÉ TECHNIKY – NEÚPLNÉ, POLYNOMIÁLNE

KOSTRA PREHĽ. ALGORITMU

```
LBL(G, cv)
  if cv > |nodes(G)| then return nodes(G)
  foreach h ∈ D_cv do if CONSISTENT(G, cv) then
    R ← LBL(G, cv+1)
    if R ≠ fail then return R
```

HÁK NA ZAVESENIE KONZ. PROCEDÚRY

LOOK BACK

– KONZISTENCIA MEDZI UŽ OHODN. PREMENNÝMI MÔŽE ZISTIŤ KONFLIKT (A PRÍP. JEHO ZDROJ)
ZÁKLADNÁ METÓDA: BACKTRACKING, AC-BT(G, cv)

FORWARD CHECKING & LOOK AHEAD

```
AC-FC(G, cv)                    FORWARD CHECKING
  Q ← { (V_i, V_cv) ∈ arcs(G), i > cv }
  while Q ≠ ∅
      SELECT AND DELETE (V_k, V_m) FROM Q
*     if REVISE(V_k, V_m) then
        if D_k = ∅ then return false
  return true
```

```
AC3-LA(G, cv)                   FULL LOOK AHEAD
  .
  .
*     if REVISE(V_k, V_m) then
        Q ← Q ∪ { (V_i, V_k) ∈ arcs(G), i ≠ k, m, i > cv }
        if D_k = ∅ then return false
```

MÔŽEME POUŽIŤ AJ INÉ AC ALGORITMY

DAC-LA (G, cv) <u>PARTIAL LOOK AHEAD</u>

```
for i ← cv+1 to n
  foreach (Vi, Vj) ∈ arcs(G), i > j ≥ cv do
    if REVISE(Vi, Vj) then if Di = ∅ then return false
return true
```

BT < FC < PLA < LA

FORWARD CHECKING ZLOŽITOSŤ BACKTRACKINGU PRÍLIŠ NEZVÝŠI
KONZISTENCIU MÔŽEME ZAISTIŤ PRED ŠTARTOM
MAC (MAINTAINING ARC CONSISTENCY)

USPORIADANIE HODNÔT

PRINCÍP PRVÉHO ÚSPECHU (SUCCEED FIRST)
(a) HODNOTA, KT. MÁ NAJVIAC PODPÔR (z AC-4)
(b) NAJMENEJ OBMEDZUJÚCA OSTATNÉ PREM. (ZO SINGLETON KONZ.)
(c) NAJVIAC RIEŠENÍ APROXIMOVANÉHO PROB. (VARIANTA ZISKU PROB.)

USPORIADANIE PREMENNÝCH

PRINCÍP PRVÉHO NEÚSPECHU (FAIL FIRST)
(a) PREMENNÉ S MENŠOU DOMÉNOU (DOM - HEURISTIKA) DYNAMICKÉ USPOR.
(b) PREMENNÉ S VIAC PODMIENKAMI (DEG-HEUR.) AK SÚ DOMÉNY ROVNAKÉ
(c) -||- S MINULÝMI PREMENNÝMI

KEĎ JE CSP ACYKLICKÝ, MEME NÁJSŤ RIEŠ. BEZ NÁVRATOV (AC)
⇒ <u>ELIMINÁCIA CYKLOV</u>

STAČÍ OHODNOTIŤ PREMENNÉ NA CYKLOCH,
PRESNEJŠIE - <u>KRUŽNICOVÝ REZ</u> (CYCLE-CUTSET)
(JEHO ODSTRÁNENIE ZRUŠÍ ∀ CYKLY)

⇝ ALGORITMUS <u>CYCLE CUTSET</u> (CC)

POSTUPNE SKÚŠAME OHODNOCOVAŤ KR. REZ (KONZISTENTNE),
A NÁSLEDNE APLIKUJEME NA STROM DAC A BACKTRACK-FREE SEARCH

05.02.2009 Th
PETER ČERNO

NOPT046 PROG. S OBMEZ. PODMIENKAMI

PRI OHODNOCOVANÍ CYCLE-CUTSET DOCHÁDZA K THRASHINGU

ZAISTÍME KONZISTENCIU PRED ŠTARTOM &
V PRIEBEHU OHODNOCOVANIA

ALGORITMUS MACE MAC EXTENDED

(1) ZAISTI HRANOVÚ KONZISTENCIU (AK NEJDE → FAIL)
(2) ROZDEĽ PREMENNÉ:
  C = CYCLE-CUTSET
  U = PREM., KT. NIE SÚ V ŽIADNOM CYKLE
(3) ODPOJ U Z GRAFU (V U NEBUDEME ZAISŤOVAŤ KONZ.)
(4) while $C \neq \emptyset$ do
  (a) ZVOĽ HODNOTU PRE PREMENNÚ V C
  (b) ZAISTI HRANOVÚ KONZISTENCIU
    (AK NEJDE, SKÚS INÚ HODNOTU, POPR. SPÄŤ K PREDCH. PREM.)
  (c) ODPOJ PREMENNÉ S 1-PRV. DOMÉNOU → DAJ ICH DO U
  (d) —"— NIE SÚ V CYKLE → —"—
(5) ZNOVU PRIPOJ U A ZAISTI SMEROVÚ HRAN. KONZ. DO U
(6) NÁJDI ÚPLNÉ RIEŠ. PREHĽADÁVANÍM BEZ NÁVRATU

NIE JE ZNÁMY POL. ALG. NA HĽADANIE CC

HEURISTIKY:

(1) DO CC DÁVAJ PREMENNÉ OD NAJVYŠŠIEHO STUPŇA
(2) + IBA PREMENNÉ V CYKLE
(3) PODĽA POČTU CYKLOV, V KT. SA PREM. NACHÁDZA

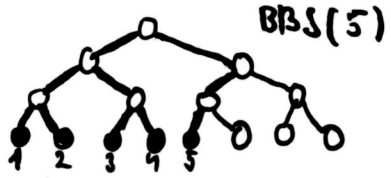
find-cycle-cutset(G)

NEÚPLNÉ STROMOVÉ PREHĽADÁVANIA

NEGARANTUJÚ ÚPLNOSŤ ANI DÔKAZ NEEXIST. RIEŠ.
ČASTO NÁJDU RIEŠ. RÝCHLO

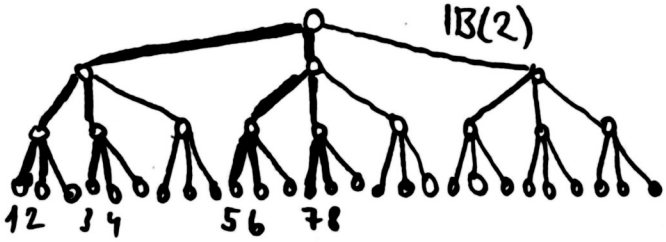
PRERUŠENIE BEHU (CUTOFF) - PO VYČERPANÍ PROSTRIEDKOV
OPAKOVANIE BEHU (RESTART) - POUŽITIE UČIACICH SA TECHNÍK

BOUNDED BACKTRACK SEARCH (BBS)



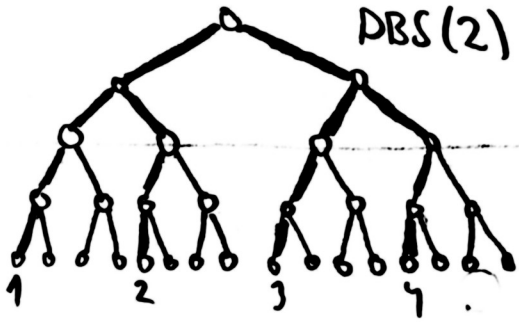
OHRANIČENÝ POČET NÁVRATOV (CUTOFF)
PRI NEÚSPECHU ZVÄČŠÍME P. NÁVRATOV O 1 (RES.)

ITERATIVE BROADENING (IB)



OHRANIČENÝ POČET ALTERNATÍV
PRI NEÚSPECHU ZVÄČŠÍME ŠÍRKU O 1

DEPTH BOUNDED SEARCH (DBS)

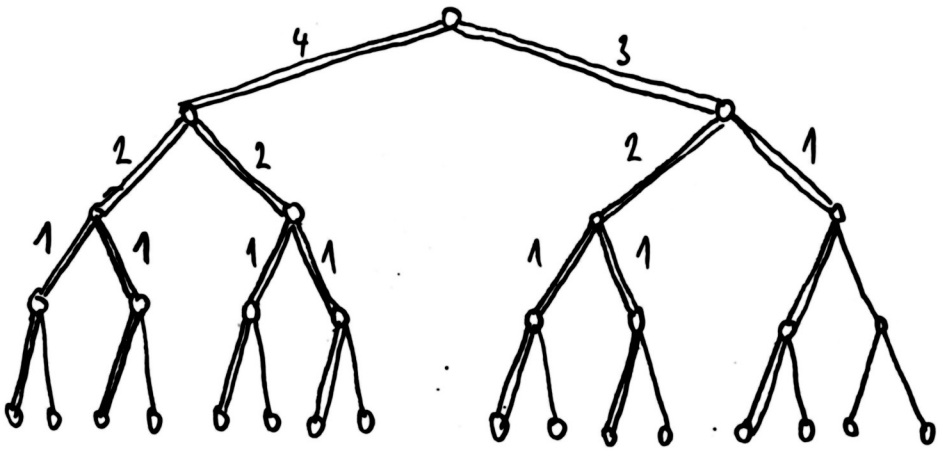


OHRANIČENÁ HĽBKA ÚPL. PREHĽ.
PRI NEÚSP. ZVÝŠ. HĽBKU

CREDIT SEARCH (CS)

OHRANIČENÝ KREDIT (POČET NÁVRATOV)
DELÍ SA MEDZI ALTERNATÍVNE VETVY

PRI NEÚSPECHU
NAVÝŠIME KREDIT O 1



05.02.2009 Th
PETER ČERNO

NOPT046 PROG. S OBMEZ. PODMIENKAMI

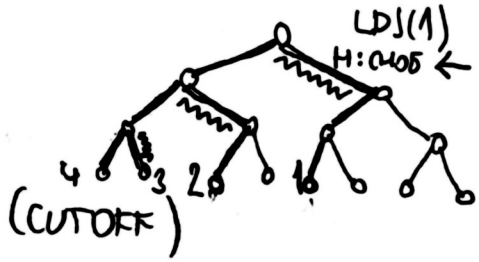
PREHĽADÁVANIE A HEURISTIKY

- POČET PORUŠENÍ HEURISTIKY NA ÚSPEŠNEJ CESTE JE MALÝ
- HEURISTIKY SÚ MENEJ SPOLAHLIVÉ NA ZAČ. PREHĽ.

DISKREPANCIA = PORUŠENIE HEURISTIKY

SKÔR PRESKÚMAME VETVE S MENEJ DISKREPANCIAMI
—||— SO SKORŠÍMI DISKREPANCIAMI

LIMITED DISCREPANCY SEARCH (LDS)



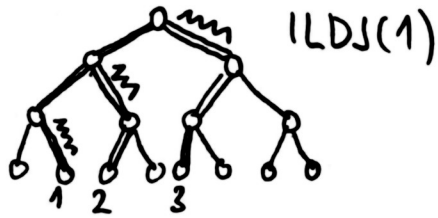
OHRANIČENÝ POČET DISKREPANCIÍ NA CESTE (CUTOFF)

PRE NEBINÁRNE DOMÉNY:
(a) NEDOPORUČENÁ HODNOTA = DISKREPANCIA
(b) ĎALŠIA NEDOP. HODNOTA = NAVÝŠENIE DISKREPANCIE

PRI NEÚSPECHU ZVÝŠIME POČET POVOLENÝCH DISKR.

IMPLEMENTÁCIA: AK JE TO MOŽNÉ SKÚŠAME NAJSKÔR NEDOPOR. HODN.
→ NEVÝHODA: PRECHÁDZAME AJ VETVY Z PREDCH. ITERÁCIE

IMPROVED LDS (ILDS)



CESTY S DISKR. NA KONCI SKÚMAME SKÔR
PEVNÝ POČET DISKREPANCIÍ NA CESTE !!

DEPTH BOUNDED DISCREPANCY SEARCH (DDS)

DISKREPANCIE POVOLENÉ DO DANEJ HĹBKY (CUTOFF)
V TEJTO HĹBKE MUSÍ BYŤ VŽDY DISKREPANCIA !!!
[CESTY S DISKREPANCIAMI NA ZAČIATKU PRESKÚMAME SKÔR]
PRI NEÚSPECHU NAVÝŠIME HĹBKU O 1 (RESTART)

LABELLING = PRIRADOVANIE HODNOT PREMENNÝM
ENUMERÁCIA = POSTUPNÉ SKÚŠANIE HODNÔT

MÔŽEME VYUŽIŤ PROPAGÁCIU AJ KEĎ ZISTÍME, ŽE HODNOTA NIE JE V RIEŠENÍ!

PRÍSLUŠNÚ HODNOTU Z DOMÉNY ZNAŽEME ⇝ KONZIST. TECHNIKA

⇒ TZV. STEP LABELLING

V PODSTATE DISJUNKCIE $X=H \vee X \neq H$

POLENIE DOMÉNY (BISECTION / DOMAIN SPLITTING)

RIEŠIME DISJUNKCIE $X \leq H \vee X > H$, KDE H JE UPROSTRED DOMÉNY

ZHRNUTIE

KONZISTENČNÉ TECHNIKY

VRCH. KONZISTENCIA: NC
HRAN. KONZISTENCIA: AC-1, AC-2, AC-3, AC-4, AC-3.1, AC-2001
SMER. HRAN. KONZ: DAC-1
KONZ. PO CESTE: PC-1, PC-2
SMER. KONZ. PO CESTE: DPC-1
OHRANIČENÁ KONZ. PO CESTE – RPC
[SILNÁ] k-KONZISTENCIA
SMEROVÁ k-KONZ. PRI USPORIADANÍ, ADAPTÍVNA
[SILNÁ] (i,j)-KONZ., INVERZ. k-KONZ., PIC, NIC
BODOVÁ A-KONZISTENCIA, NEBINÁRNE GAC

PREHĽADÁVANIE

LOOK BACK: AC-BT(G,cv)
FORWARD CHECKING: AC-FC
LOOK AHEAD (PART.): DAC-LA
FULL LOOK AHEAD: AC3-LA
ALG. CYCLE-CUTSET (CC)
MAC EXTENDED (MACE)

NEÚPLNÉ STROMOVÉ PREHĽ.

BOUNDED BACKTRACK SEARCH (BBS)
ITERATIVE BROADENING (IB)
DEPTH BOUNDED SEARCH (DBS)
CREDIT SEARCH (CS)

HEURISTIKY

(IMPROVED) LIMITED DISCREPANCY S. (I)LDS
DEPTH BOUNDED DISCR. SEARCH (DDS)

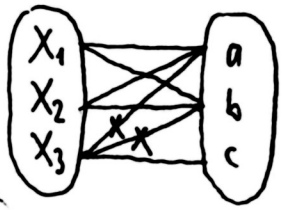
06.02.2009 Fr
PETER ČERNO

NOPTO46 PROG. S OMEZ. PODMIENKAMI

GLOBÁLNE PODMIENKY

all_different $(\{x_1, \dots, x_k\}) = \{(d_1, \dots, d_k) \mid \forall i\ d_i \in D_i\ \&\ \forall i \neq j\ d_i \neq d_j\}$

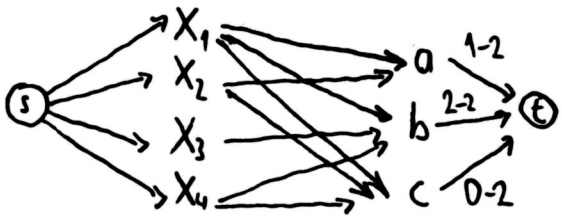
LEPŠIE PROPAGUJE AKO SYSTÉM NEROVNOSTÍ $x_i \neq x_j\ \forall i<j$



INICIALIZÁCIA:
(1) NÁJDI MAX. PÁROVANIE
(2) ODSTRÁŇ $\forall$ HRANY, KT. NELEŽIA V ŽIADNOM MAX. P.

INKREMENT. PROPAGÁCIA:
(1) ODSTRÁŇ NEPLATNÉ HRANY
(2)(3) ANALOGICKÝ POSTUP

gcc (GLOBAL CARDINALITY CONSTRAIN) = ZOVŠEOB. ALL-DIFFERENT
PRE KAŽDÚ HODNOTU ZO ZJEDNOTENIA DOMÉN JE URČENÝ MINIMÁLNY A MAXIMÁLNY POČET VÝSKYTOV



NÁJDEME HRANY, KT. MAJÚ TOK 0 V KAŽDOM MAXIMÁLNOM TOKU
⇝ ODFILTRUJEME PRÍSL. DOMÉNY

lex (LEXIKOGRAFICKÉ USPORIADANIE DVOCH VEKTOROV)

NA ODSTRÁNENIE SYMETRIÍ...

$$lex(\{x_1, \dots, x_n\}, \{y_1, \dots, y_n\}) \equiv (x_1 \leq y_1)\ \&\ (x_1 = y_1 \Rightarrow x_2 \leq y_2) \wedge \dots \wedge (x_1 = y_1 \wedge \dots \wedge x_{n-1} = y_{n-1} \Rightarrow x_n = y_n)$$

GLOBÁLNA FILTRÁCIA POUŽÍVA 2 UKAZATELE:

α: PREMENNÉ PRED α SÚ INŠTANCIOVANÉ A MAJÚ ROVNAKÚ HODNOTU

β: VEKTORY ZAČÍNAJÚCE β SÚ USP. OBRÁTENE
$floor(\{x_\beta, \dots, x_n\}) >_{lex} ceiling(\{y_1, \dots, y_n\})$

DOMÉNY x_α, y_α UPRAVÍME PODĽA $x_\alpha \leq y_\alpha$.
NÁSLEDNE AK DOSTANEME 1-OVÉ DOMÉNY, POSUNIEME α
AK $\alpha = \beta - 1$ MNÚTIME PODMIENKU $x_\alpha < y_\alpha$.

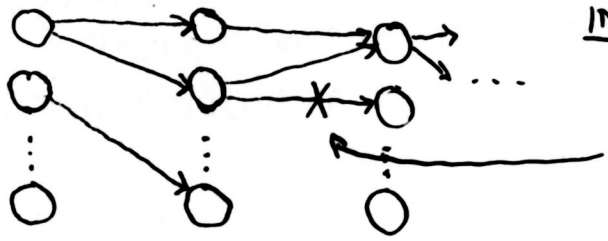
ROSTERING = ROZVRHOVANIE ZMIEN

ŠPECIÁLNE OBMEDZENIA NA POSTUPNOSŤ ZMIEN PRE DANÚ OBORU

$\underline{\text{regular}}(A, \{X_1, \dots, X_k\}) = \{(d_1, \dots, d_k) \mid \forall i\ d_i \in D_i,\ d_1 \dots d_k \in L(A)\}$

FILTRÁCIA ZALOŽENÁ NA ROZKLADE VÝPOČTU KONEČNÉHO AUTOMATU POMOCOU VRSTVENÉHO ORIENT. GRAFU

VRSTVY = STAVY HRANY = PRECHODY



INICIALIZÁCIA:

PRI SPÄTNOM CHODE ODSTRÁNIME HRANY, KT. NEMAJÚ POKRAČOVANIE

$\{a, b, c\}$ $\{c, \cancel{d}\}$ ← ODSTRÁNIME ZNAKY BEZ HRANY

POUŽITÉ HODNOTY

INKREMENTÁLNA PROPAGÁCIA:

ODSTRÁNIME HRANY PRE DANÝ ZNAK

ZMENU PROPAGUJEME OBOMA SMERMI

ODSTRÁNIME ZNAKY BEZ HRANY

$\underline{\text{gramar}}(G, \{X_1, \dots, X_k\}) = \{(d_1, \dots, d_k) \mid \forall i\ d_i \in D_i,\ d_1 \dots d_k \in L(G)\}$

ALGORITMUS CYK PRE CHOMSKÉHO NF $(A \to BC \mid x)$

$X_{i,j} := \{A \mid A \Rightarrow^* a_i \dots a_j\}$

$a_1 \dots a_n \in L(G) \Leftrightarrow S \in X_{1,n}$

$\underline{\text{slide}_j}(C, \{X_1, \dots, X_n\}) \equiv \forall i\ C(X_{i\,j+1}, \dots, X_{i\,j+k})$

SKOK, O KOĽKO SKÁČEME — k-ÁRNA PODMIENKA

PODMIENKA KĽÚČE PO POSTUPNOSTI PREMENNÝCH

06.02.2009 Fr
PETER ČERNO

NOPT046 PROG. S OBMEZ. PODMIENKAMI

$regular(A, \{X_1, \dots, X_n\}) \equiv slide_2(C, \{Q_0, X_1, Q_1, \dots, X_n, Q_n\})$

$Q_0 = \{q_0\}$, $Q_n = F$, $C(P, X, \underset{\sim}{Q}) \sim \delta(P, X) = Q$

$lex(\{X_1, \dots, X_n\}, \{Y_1, \dots, Y_n\}) \equiv$

$slide_3(C, \{B_0, X_1, Y_1, B_1, \dots, X_n, Y_n, B_n\})$

$C(B, X, Y, C) \equiv (B = C = 1) \vee (B = C = 0 \;\&\; X = Y) \vee (B = 0 \wedge C = 1 \wedge X < Y)$

$B_0 = 0$, $B_n = 1$ (STRIKTNÝ LEX), $B_n \in \{0, 1\}$ (NE-STRIKTNÝ LEX)

$stretch(\{X_1, \dots, X_n\}, s, l, t) \equiv slide_2(C, \{X_1, S_1, \dots, X_n, S_n\})$

$C(X_i, S_i, X_{i+1}, S_{i+1}) \equiv X_i = X_{i+1}, S_{i+1} = S_i + 1 \leq l(X_i),$

$\vee\; X_i \neq X_{i+1}, S_i \geq s(X_i), S_{i+1} = 1, (X_i, X_{i+1}) \in t$

ROZVRHOVACÍ PROBLÉM

UNÁRNY ZDROJ – V DANEJ CHVÍLI MÔŽE SPRACOVÁVAŤ JEDINÚ AKTIVITU. KAŽDÁ AKTIVITA MÁ PEVNÚ DĹŽKU TRVANIA A PRIRADENÉ ČASOVÉ OKNO, V KTOROM MUSÍ BYŤ SPRACOVANÁ.

MODEL:

PREMENNÉ – ŠTARTOVACÍ ČAS AKTIVÍT: $start(A)$

PODMIENKY – NEPREKRÝVANIE SA AKTIVÍT

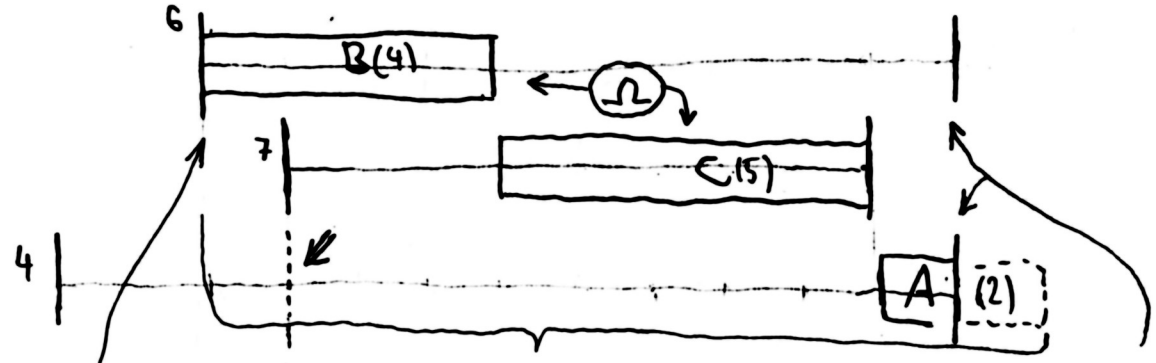
$start(A) + p(A) \leq start(B) \vee$

$start(B) + p(B) \leq start(A)$

POPR. RELÁCIE TYPU PRECEDENCIA $A \ll B$

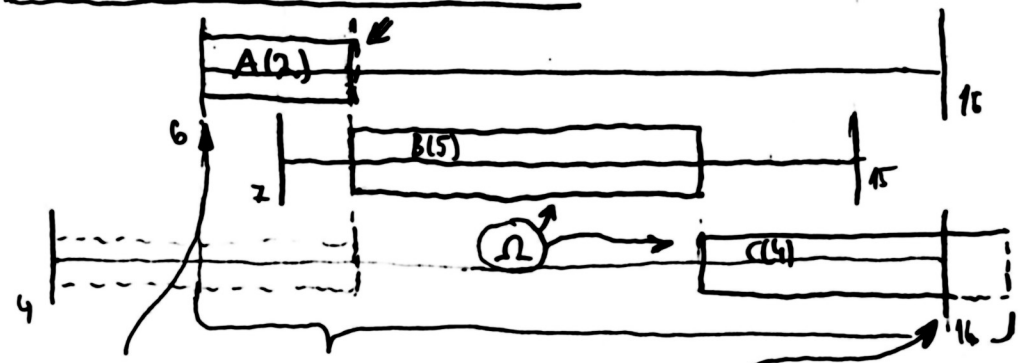
DISJUNKCIE MÁLO PROPAGUJÚ!

PODMIENKA edge_finding



$$\min(\text{start}(\Omega)) + p(\Omega \cup \{A\}) > \max(\text{end}(\Omega \dot{\cup} \{A\})) \Rightarrow \underline{\underline{A \ll \Omega}}$$

PODMIENKA not_first



$$\text{start}(A) + p(\{A\} \cup \Omega) > \max(\text{end}(\Omega)) \Rightarrow \neg\, \underline{\underline{A \ll \Omega}}$$

07.02.2009 Sa
PETER ČERNO

NOPT046 PROG. S OMEZ. PODMIENKAMI

CSOP (CONSTRAINT SATISFACTION OPTIMISATION PROBLEM)

PROBLÉM OPTIMÁLNEHO SPĽŇOVANIA PODMIENOK
RIEŠENIE CSOP - MAX./MIN. OBJEKTÍVNU FCIU f

BRANCH & BOUND (METÓDA VETIEV A MEZÍ)

- HEURISTICKÁ FUNKCIA h ODHADUJÚCA OBJ. FCIU.
  KOREKTNÁ HEUR. PRE MIN. $h(x) \leq f(x)$ (PRE MAX. $\geq$)

PRI PREHĽADÁVANÍ JE PODSTROM ODREZANÝ AK:
(a) V PODSTROME NEEXISTUJE RIEŠ.
(b) V PODSTROME NIE JE OPT. RIEŠENIE: $mez \leq h(x)$

S OBJEKTÍVNOU FCIOU. PRACUJEME AKO S PODMIENKOU $v = f(x)$
→ PRVÉ RIEŠ. ZÍSKAME BEZ OHRANIČENIA v
→ PO ĎALŠÍCH POŽADUJEME, ABY BOLI LEPŠIE

- ALGORITMUS BRANCH & BOUND

```
BB-Min (Variables, V, Constraints)
    Bound ← sup , NewSolution ← fail
    repeat
        solution ← NewSolution
        NewSolution ← Solve (Variables, Constraints ∪ {V<Bound})
        Bound ← value of V in NewSolution (if any)
    until NewSolution = fail
    return Solution
```

HEURISTIKA h JE UKRYTÁ V PROPAGÁCII PODMIENKY $V = h(x)$

BB S POLENÍM (DOLNÁ A HORNÁ MEZ)

```
repeat
  TempBound ← (UBound + LBound) / 2
  NewSolution ← Solve (Var., Cons. ∪ {V ≤ TempBound})
  if NewSolution = fail then LBound ← TempBound + 1
  else UBound ← TempBound
until LBound = UBound
```

PRÍLIŠ OBMEDZENÉ PROBLÉMY (OVER-CONSTRAINED)

(1) ZVÄČŠÍME DOMÉNU PREMENNÝCH
(2) ZVÄČŠÍME DOMÉNU PODMIENKY
(3) VYRADÍME PODMIENKU
(4) VYRADÍME PREMENNÚ

ČIASTOČNÉ SPĽŇOVANIE PODMIENOK

PRIESTOR PROBLÉMOV CSP $(PS, \leq)$

$P_1 \leq P_2 \Leftrightarrow$ MNOŽINA RIEŠ. $P_2 \subseteq$ MNOŽINA RIEŠ. P_1

TENTO PRIESTOR ZÍSKAME OSLABENÍM PÔVODNÉHO PROB.

PROBLÉM ČIAST. SPĽ. PODMIENOK (PARTIAL CSP, PCSP)
JE $\langle P, (PS, \leq), M, (N, S) \rangle$

M JE METRIKA NA $(PS, \leq)$ (POČET ROZN. RIEŠENÍ / n-TÍC V PODM. ...)
N JE MAXIMÁLNA MOŽNÁ HOD., S JE POSTAČUJÚCA HOD.

RIEŠENÍM PCSP JE PROBLÉM P' (A JEHO RIEŠENIE)
T.Ž. $M(P, P') < N$

POSTAČUJÚCE RIEŠENIE PCSP : $M(P, P') \leq S$
OPTIMÁLNE —||— : MINIMALIZUJE HODNOTU OD P.

V PRAXI PRACUJEME S PÔVODNÝM PROBLÉMOM,
POUŽÍVAME EVALUAČNÚ FCIU g

07.02.2009 Sa
PETER ČERNO

NOPT046 PROG. S OBMEZ. PODMIENKAMI

PCSP ZOBECŇUJE CSOP: $g(x) = f(x)$ AK JE x RIEŠENIE CSP
$g(x) = \infty$ INAK

PRI RIEŠENÍ PCSP NUTNÉ UPRAVIŤ PROPAGAČNÉ ALGORITMY
MOŽNO POUŽIŤ LOK. PREHĽ.

VÁŽENE (WEIGHTED) CSP - KAŽDEJ PODMIENKE PRIRADÍME VÁHU, CIEĽOM JE MINIMALIZOVAŤ SÚČET VÁH NESPL. PODM.

OCENENÉ SPĽŇOVANIE PODMIENOK (VALUED CSP)

STRUKTÚRA OCENENIA $(E, \otimes, >, \bot, \top)$

E = MNOŽINA OCENENÍ ÚPLNE USP. POMOCOU $>$
$\bot$ = MIN. PRVOK, $\top$ = MAX. PRVOK

$\otimes$ = KOM. ASOC. OP NA E, $\bot \otimes a = a$, $\top \otimes a = \top$ (ABSORPČNÝ PR.)
ZACHOVÁVA MONOTÓNIU $a \geq b \Rightarrow a \otimes c \geq b \otimes c$

$\varphi : C \to E$ MAPUJE PODMIENKY NA OCENENIA

RIEŠENIE PROBLÉMU: MINIMALIZUJE
AGREG. OCENENIE $v(A) = \bigotimes \varphi(c)$
A NIE JE KONZ. S $c \in C$

FRAMEWORK	E	$\otimes$	$\bot$	$\top$
CLASSICAL CSP	$\{true, false\}$	$\wedge$	true	false
WEIGHTED CSP	$\mathbb{N} \cup \{+\infty\}$	$+$	0	$+\infty$
PROBABILISTIC CSP	$\langle 0,1 \rangle$	$\times$	1	0
POSSIBILISTIC CSP	$\langle 0,1 \rangle$	max	0	1
LEXICOGRAPHIC CSP	$\mathbb{N}^{\langle 0,1 \rangle} \cup \{T\}$	$\cup$	$\emptyset$	T

$<_{lex}$

CSP NAD POLO-OKRUHMI (SEMIRING-BASED CSP)

KAŽDEJ N-TICI HODNÔT PREMENNÝCH V PODM. → PREFERENCIA (AKO DOBRE SPĹŇA PODM.)
MAXIMALIZUJEME AGREGOVANÚ PREFERENCIU

ŠTRUKTÚRA C-POLOOKRUHU $(A, +, \times, 0, 1)$
A JE MNOŽINA PREFERENCIÍ
+ JE KOMUTATÍVNA, ASOCIATÍVNA, IDEMPOTENTNÁ $(a+a=a)$
BIN. OP. NA A S JEDNOTKOVÝM PRVKOM 0 $(0+a=a)$
A ABSORPČNÝM PRVKOM 1 $(1+a=1)$
⇒ INDUKUJE USPORIADANIE: $a \leq b \Leftrightarrow a+b=b$
× JE KOMUTATÍVNA, ASOCIATÍVNA BIN. OP. NA A
S JEDNOTKOVÝM PRVKOM 1 $(1 \times a = a)$ A ABSORPČNÝM
PRVKOM 0 $(0 \times a = 0)$, KT. JE DISTRIBUTÍVNA NAD +

RIEŠENIE JE TAKÉ OHODNOTENIE PREMENNÝCH V, KTORÉ
DÁVA NAJVÄČŠIU AGREGOVANÚ PREFERENCIU $p(V)$:

$$p(V) = \bigtimes_{c \in C} \delta_c(V \downarrow vars(c))$$

N-TICE V PODMIENKE C SÚ MAPOVANÉ NA POLO-OKRUH FC. $\delta_c : V \to A$

FRAMEWORK	A	+	×	1	0
CLASSICAL CSP	{false, true}	∨	∧	true	false
WEIGHTED CSP	$\mathbb{N} \cup \{+\infty\}$	min	+	0	+∞
PROBABILISTIC CSP	$\langle 0,1 \rangle$	max	×	1	0
POSSIBILISTIC CSP	$\langle 0,1 \rangle$	min	max	0	1
FUZZY CSP	$\langle 0,1 \rangle$	max	min	1	0
LEXICOGRAPHIC CSP	$\mathbb{N}^{\langle 0,1 \rangle} \cup \{T\}$	max_{lex}	∪	∅	T

07.02.2009 Sa
PETER ČERNO

NOPT046 PROG. S OMEZ. PODMIENKAMI

HIERARCHIA OBMEDZUJÚCICH PODMIENOK

KAŽDEJ PODMIENKE PRIRADÍME PREFERENCIU (Z LIN. USP. MN.)
VÝZNAČNÁ JE PREFERENCIA REQUIRED (HARD CONSTR.) MUSÍ BYŤ SPLNENÁ
OSTATNÉ NEMUSIA BYŤ (SOFT CONSTRAINTS)

HIERARCHIA PODMIENOK H : H_0 = NUTNÉ PODM., H_1 = NADVIAC PREF., ...

RIEŠENÍM HIERARCHIE JE OHODNOTENIE PREM., KT. SPĽŇA NUTNÉ PODM. A NAJLEPŠIE SPĽŇA PREFER. PODMIENKY

$$S_{H,0} = \{ \sigma \mid \forall c \in H_0 \ c\sigma \text{ platí} \}$$

$$S_H = \{ \sigma \in S_{H,0} \mid \forall \omega \in S_{H,0} \ \neg better(\omega, \sigma, H) \}$$

KOMPARÁTOR (POROVNANIE OHODN. PREM. VZHĽADOM K HIERARCHII)
ANTIREFLEXÍVNA, TRANZITÍVNA RELÁCIA

RESPEKTUJE HIERARCHIU - AK NEJAKÉ OHODNOTENIE SPĽŇA VŠETKY PODMIENKY AŽ DO ÚROVNE k, POTOM TO ISTÉ SPĽŇAJÚ VŠETKY LEPŠIE OHODNOTENIA

CHYBOVÁ FUNKCIA $e(c, \sigma)$: PREDIKÁTOVÁ / METRICKÁ

LOKÁLNE KOMPARÁTORY - POROVNÁVAJÚ CHYBU ZVLÁŠŤ U KAŽDEJ PODM.

$$\begin{aligned} locally_better(\omega, \sigma, H) \equiv \exists k > 0 \\ \forall i < k \ \forall c \in H_i \ e(c, \omega) = e(c, \sigma) \ \& \\ \forall c \in H_k \ e(c, \omega) \leq e(c, \sigma) \ \& \\ \exists c \in H_k \ e(c, \omega) < e(c, \sigma) \end{aligned}$$

GLOBÁLNE KOMPARÁTORY - AGREGUJÚ CHYBU PRE CELÚ ÚROVEŇ

$$\begin{aligned} globally_better(\omega, \sigma, H) \equiv \exists k > 0 \\ \forall i < k \ g(H_i, \omega) = g(H_i, \sigma) \ \& \\ g(H_k, \omega) < g(H_k, \sigma) \end{aligned}$$

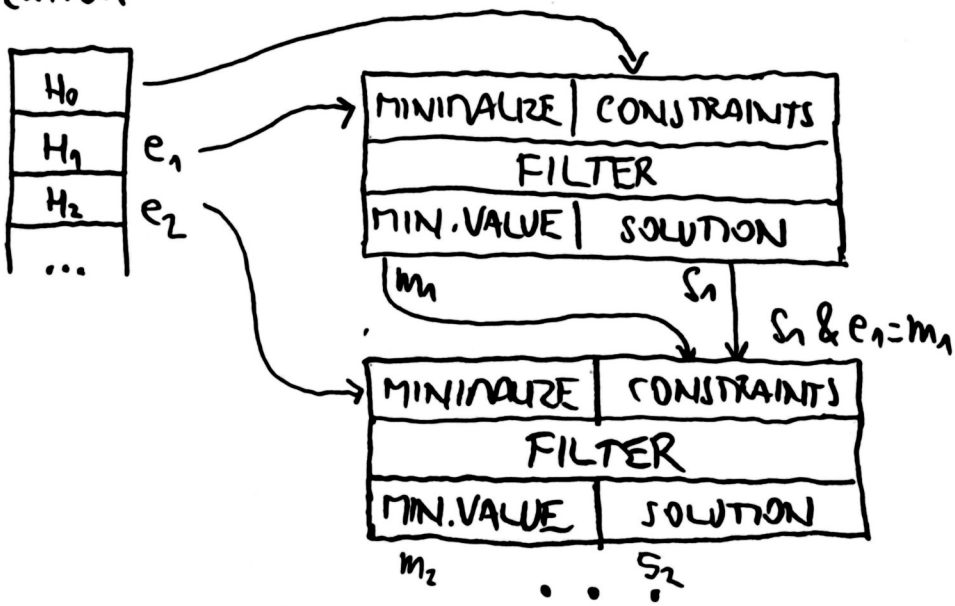
VÁŽENÉ SÚČTY,
SÚČTY ŠTVORCOV,
NAJHORŠÍ PRÍPAD

DELTA STAR

RIEŠENIE PODMIENOK PO ÚROVNIACH POSTUPNÝM ZJEMŇOVANÍM MNOŽINY RIEŠENÍ

```
DeltaStar(H: constraint hierarchy)
  i ← 1
  Solution ← SOLUTION OF REQUIRED CONS. FROM H
  while not unique Solution & i < number of levels do
    Solution ← filter(Solution, H_i),  ++i
  return Solution
```

SIMPLEX:



LOKÁLNA PROPAGÁCIA

HĽADANIE RIEŠENIA POSTUPNÝM SPĽŇOVANÍM PODMIENOK
PODMIENKA JE POPÍSANÁ METÓDAMI PRE VÝPOČET HODNÔT JEJ PREM.

VÝHODY: CEZ SIEŤ PODM. MÔŽEME PROPAGOVAŤ ZMENU HODN.
METÓDY MOŽNO POPREDU SKOMPILOVAŤ

OBMEDZENIA: IBA PRE FUNKCIONÁLNE PODMIENKY (ROVNOSTI)
V GRAFE METÓD NESMIE BYŤ CYKLUS
NÁJDE IBA 1 RIEŠENIE
FUNGUJE IBA S LOK. PRED. KOMPARÁTOROMI

07.02.2009 Sa
PETER ČERNO

NOPT046 PROG. S OBMEZ. PODMIENKAMI

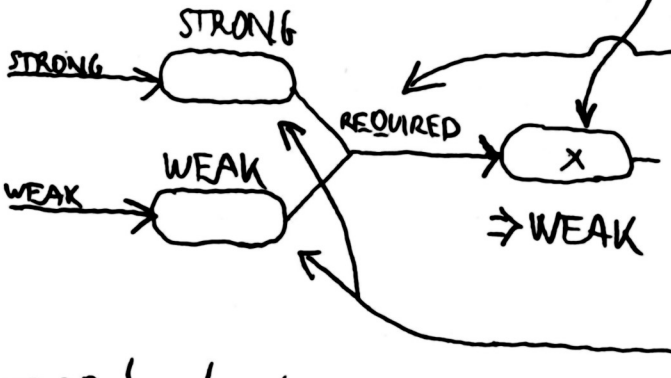
DELTABLUE

PRACUJE S METÓDAMI MAJÚCIMI JEDINÝ VÝSTUP
(1) NAJSKÔR VYBERIE PRE KAŽDÚ PODM. METÓDU - PLÁNOVANIE
(2) CEZ ZVOLENÚ SIEŤ PROPAGUJE HODNOTY - EXEKÚCIA

INKREMENTÁLNE PLÁNOVANIE - PO PRIDANÍ PODM. UPRAVÍ SIEŤ

ALGORITMUS PRIECHODZEJ PREFERENCIE (WALKABOUT STRENGTH)

PRIECHODZA PREFERENCIA PREMENNEJ x JE SLABŠIA Z PREFERENCIÍ PODMIENKY



KTOREJ METÓDA URČUJE HODNOTU PREMENNEJ x

A Z PRIECHODZÍCH PREFERENCIÍ PREMENNÝCH

KTORÉ SÚ VÝSTUPOM ZVYŠNÝCH METÓD TEJTO PODMIENKY

V PODSTATE TO ZNAMENÁ, ŽE KEĎ ZAVEDIEME NOVÚ CONSTRAINT DO x, TAK MÔŽEME PODMIENKY PREORIENTOVAŤ TAK, ŽE TO „SCHYTÁ" WEAK PODMIENKA

AddConstraint (c: constraint)
  SELECT THE POTENTIAL OUTPUT VAR. V OF c WITH WEAKEST WALKABOUT ST.
  if WALKABOUT STR. OF V < STRENGTH OF c then
    c' ← CONSTRAINT DETERMINING V
    MAKE c' UNSATISFIED
    SELECT THE METOD DETERMINING V IN c
    RECOMPUTE WALKABOUT STRENGTHS OF DOWNSTREAM VARS.
    AddConstraint (c') ← VYRADENÚ PODMIENKU SKÚSIME ZNOW PRIDAŤ

DELTABLUE PRACUJE IBA S FUNKC. PODM. S JEDINÝM VÝSTUPOM
SKYBLUE ZOBECŇUJE DELTABLUE O METÓDY S VIAC VÝSTUPMI

INDIGO - NAVRHNUTÝ PRE ACYKLICKÉ HIERARCHIE
S NEFUNKC. PODMIENKAMI A S LOK. METR. KOMPARÁTOROM
<u>PROPAGUJE OKRAJE DOMÉN</u> (ROBÍ PROBLÉM OKRAJ. KONZIST.)
POSTUPNE PRIDÁVA PODMIENKY OD NAJSILNEJŠEJ PO NAJSLABŠIU

<u>PROJEKČNÝ ALGORITMUS</u> RIEŠENIE LIN. ROVNOSTÍ A NEROVN. GAUSSOVOU A FOUR. ELIMINÁCIOU

$C(0,x)$ ≡ PODMIENKY NEOBSAHUJÚCE x
$C(=,x)$ ≡ ROVNOSTI OBSAHUJÚCE x
$C(+,x)$ ≡ NEROVNOSTI, KTORÉ MOŽNO PREVIESŤ NA $x \leq e$
$C(-,x)$ ≡ NEROVNOSTI, KT. MOŽNO PREVIESŤ NA $e \leq x$

```
project (C: set of constraints, x: variable)
  if ∃c∈C(=,x) where c is x=e then
    D ← C-{c} with every ocurence of x repl. by e
  else
    D ← C(0,x)
    foreach c in C(+,x) where c is x≤e+
      foreach c in C(-,x) where c is e-≤x
        D ← D ∪ {e-≤e+}
```

PROJEKCIAMI POSTUPNE ODSTRÁNIME ∀ PREMENNÉ
A POTOM SPÄŤNE DOPOČÍTAME HODNOTY PREM.

HIERARCHICKÉ PODMIENKY:
PRE LOKÁLNY METRICKÝ KOMPARÁTOR: PODMIENKY
e?b@pref PREVEDIEME NA e?v_e@required, v_e=b @pref
PREMENNÉ ELIMINUJEME <u>OD NAJSLABŠEJ</u>
BERIEME HODNOTU <u>NAJBLIŽŠIU KONŠT. b Z NAJSILNEJŠEJ PODM.</u>

08.02.2009 Su
PETER ČERNO

NOPT046 PROG. S OBMEZ. PODMIENKAMI

DUÁLNE KÓDOVANIE : $(k-)c \rightsquigarrow$ PREMENNÁ v_c, MEDZI $v_c, v_{c'}$ BIN.P.
KÓDOVANIE SO SKRYTOU PREM. : $(k-)c \rightsquigarrow v_c$, $\forall x \in c$: BIN.P. x A v_c
HYBRIDNÉ KÓDOVANIE

LOKÁLNE PREHĽADÁVANIE

HC (HILL CLIMBING) - NAJLEPŠÍ STAV V OKOLÍ, REŠTART Z LOK. MIN.
MC (MIN-CONFLICTS) - ZVOĽ KONFLIKTNÚ PREM., ZLEPŠI, MaxMoves x
RW (RANDOM WALK) - STAV Z OKOLIA JE VOLENÝ NÁHODNE
MCRW - ZVOĽ NÁH. KONFL. PR., S PRST. p NASTAV NÁH., INAK MC
SDRW - S PRST. p NÁHODNÝ STAV V OKOLÍ, INAK HC

TABU ZOZNAM, ASPIRAČNÉ KRITÉRIUM, TABU-SEARCH

INTENZIFIKÁCIA (HC/MC), DIVERZIFIKÁCIA (restart/RW/TABU)
GSAT (ϕ IN CNF)
GENET
SA (SIMULATED ANNEALING)

SYSTEMATICKÉ PREHĽADÁVANIE

BACKTRACKING - REKURZÍVNE BT(X: vars, V: assigns, C: cons)

ITERATÍVNE

```
while 1 ≤ i ≤ n do
    INSTANTIATE-AND-CHECK (i, C)
    if x_i = null then --i else ++i, D'_i ← D_i
```

PROBLÉMY :
TRASHING : NEDOKÁŽE VYUŽIŤ INFO O ZDROJI KONFL. → BACKJUMPING
REDUNDAN. PRÁCA : NEPAMÄTÁ SI UŽ ODHALENÉ KONFL. → BACKMARKING
ONESK. ODHALENIE CHYBY : → FORWARD CHECKING

GRAPH-DIRECTED BACKJUMPING - SLEPÝ UZOL, anc(x)
SKÁČEME NA NAJBL. Z $anc(x) \cup anc(y_1) \cup \dots \cup anc(y_k) - \{y_1, \dots, y_k\}$
GraphBJ REKURZÍVNE

```
if R = fail (JumpSet) then
    if x ∉ JumpSet then return R
    conflict ← ans(x) ∪ JumpSet - {x}
else return R
```

GraphBJ ITERATÍVNE:

```
if x_i = null then j ← LATEST INDEX IN I_i
    I_j ← I_j ∪ I_i − {x_j},  i ← j
else ++i,  D'_i ← D_i,  I_i ← anc(x_i)
```

GASCHNIGOV BACKJUMPING

consistent(Labelled, Constraints, Level) x/h/Level ∈

METÓDA ZISTÍ, ČI SÚ Constraints KONZISTENTNÉ s Labelled, A AK NIE, TAK VRÁTI fail(J), KAM TREBA SKOČIŤ

```
foreach C NONSATISFIED do
  J ← min(J, max{L | x/v/L ∈ Labelled, x ∈ vars(C), L < Level})
```

KONFLIKTOM RIADENÉ BACKJUMPING (CBJ)
SPOJENIE VÝHOD OBOCH METÓD

DYNAMICKÝ BACKTRACKING

BACKJUMPING + PAMÄTANIE SI DÔVODU KONFLIKTU + PRENOS DÔVODU KONFLIKTU + ZMENA POREDIA PREM.

```
if Values_x = ∅ then
  NECH E JE VYSVETLENIE KONFLIKTU
  if E = ∅ then failure
  else NECH Y JE NAJVIAC NEDÁVNA PREM. v E
    ODSTRÁNIME Y z Labelled s VYSVETLENÍM E − {Y}
    ODSTRÁNIME ∀ VYSV. KT. OBSAHUJÚ Y
```

BACKMARKING

MARK(X, V) = NAJVZDIALENEJŠÍ KONFLIKT s X := V
BACKTO(X) = MIESTO NÁVRATU OD POSLEDNÉHO OHODN. X
AK MARK(X, V) < BACKTO(X), POTOM X := h JE STÁLE NEKONZ.

TEST KONZISTENCIE:
consistent(X/V, Labelled, Constraints, Level) ⇝ NASTAVUJE MARK(X, V)

BM

```
foreach h ∈ D_x do
  if MARK(X, h) ≥ BACKTO(x) then
    ...
BackTo(x) ← Level − 1
foreach Y in Unlabelled do
  BackTo(Y) ← min(Level − 1, BackTo(Y))
```

08.02.2009 Sa
PETER ČERNO

NOPT046 PROG. S OBMEZ. PODMIENKAMI

KONZISTENČNÉ TECHNIKY

VRCHOLOVÁ, HRANOVÁ KONZISTENCIA, REVISE(i,j)
AC-1, AC-2, AC-3

PODPORA PRE $a \in D_i = \{ \langle j,b \rangle \mid b \in D_j, (a,b) \in C_{i,j} \}$

$S_{j,b}$ = MNOŽINA DVOJÍC $\langle i,a \rangle$, PRE KTORÉ JE $b \in D_j$ PODPORA
counter$[(i,j),a]$ = POČET PODPÔR, KT. MÁ $a \in D_i$ V PREN. V_j

INITALIZE(G) INICIALIZUJE S A counter A VRÁTI Q FRONTU HODNÔT $\langle i,a \rangle$, KTORÉ BOLI ODSTRÁNENÉ Z DOMÉNY D_i, PRETOŽE NEMAJÚ PODPORU V NEJAKEJ PODM.

AC-4

```
• SELECT AND DELETE ANY <j,b> ∈ Q
  foreach <i,a> ∈ S_{j,b} do   -- counter[(i,j),a]
    if counter[(i,j),a] = 0 & a ∈ D_i then ...
```

AC-3.1 PRE KAŽDÚ HODNOTU SI PAMÄTÁM POSLEDNÚ NÁJDENÚ PODPORU
EXIST$((i,x),j)$ .. last$[(i,x),j]$

AC-2001 - FRONTA VRCHOLOV

SMEROVÁ HRANOVÁ KONZISTENCIA, DAC-1

KONZISTENCIA PO CESTE, REPREZENTÁCIA PODMIENOK: R_{ij}
REVISE$(i,j) \equiv R_{ii} \leftarrow R_{ii} \,\&\, (R_{ij} * R_{jj} * R_{ji})$
REVISE-PATH$(i,k,j) \equiv R_{ij} \leftarrow R_{ij} \,\&\, (R_{ik} * R_{kk} * R_{kj})$

PC-1, PC-2: $S_a(i,j)$, $S_b(i,j)$

SMEROVÁ KONZISTENCIA PO CESTE $\forall (i,j,k)$ $i,j \leq k$ SÚ KONZ.

DPC-1
```
for k ← n downto 1, i ← 1 to k, j ← i to k
  if (i,k) ∈ E & (j,k) ∈ E then C_ij ← ...,  E ← E ∪ {(i,j)}
```

OHRANIČENÁ KONZISTENCIA PO CESTE
VRCHOL i JE OHRANIČENE KONZISTENTNÝ PO CESTE $\Leftrightarrow$ ...
Q_{PC} $(\langle i,a \rangle, j, k)$ counter$[(i,j),a] = 1$, $(i,k),(k,j) \in$ arcs(G) RPC

k-KONZISTENTNÉ CSP, [SILNO], CSP MRIEŽ. BEZ NAVRACANIA
SMEROVO k-KONZ., ADAPTÍVNA KONZ. - ZAISTUJE SMEROVÚ i-KONZ.

(i,j)-KONZ. [SILNO], (1,k)-INVERZNÁ k-KONZ.

(1,2)-KONZ. = INV. KONZ. PO CESTE PIC, NIC

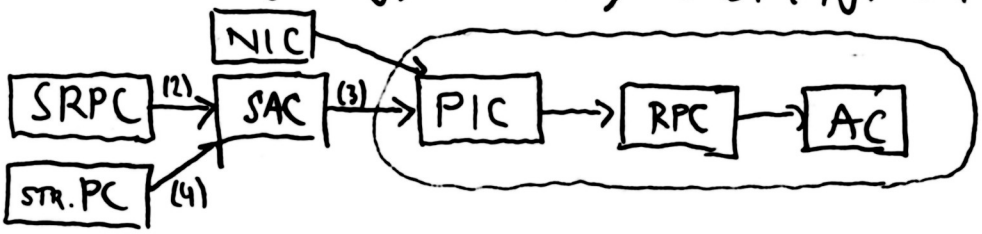
BODOVÁ A-KONZ. (SINGLETON A-CON.)

(1) BODOVÁ A-KONZ. ≥ A-KONZ.

(2) A-KONZ. ≥ B-KONZ ⇒ BODOVÁ A-KONZ. ≥ BODOVÁ B-k.

(3) BOD. (i,j)-KONZ > (i, j+1)-KONZ. [SAC>PIC] ▽

(4) SILNÁ (i+1, j)-KONZ. > BOD. (i,j) KONZ. [PC>SAC] ∘



ZOBECNENÁ HRANOVÁ KONZ. (GAC).

GAC-3 Q ← { $X_s \to Y$ METHOD FOR CONSTR. IN G }

SLABŠIA GAC - OKRAJOVÁ KONZISTENCIA

PREHĽADÁVANIE + KONZ. TECHNIKY

LOOK BACK, FORWARD CHECKING,
FULL/PARTIAL LOOK AHEAD

USPORIADANIE HODNOT, PREM.

ELIMINÁCIA OKLOV, CYCLE-CUTSET, MACE: C, U

08.02.2009 Sa
PETER ČERNO

NOPT046 PROG. S OBMEZ. PODMIENKAMI

NEÚPLNÉ STROMOVÉ PREHL. CUTOFF, RESTART

BOUNDED BACKTRACK SEARCH (BBS)
ITERATIVE BROADENING (IB)
DEPTH BOUNDED SEARCH (DBS)
CREDIT SEARCH (CS)

+ HEURISTIKY DISKREPANCIA

LIMITED DISCREPANCY SEARCH (LDS)
IMPROVED LDS (ILDS)
DEPTH BOUNDED DISCR. SEARCH (DDS)

LABELLING, ENUMERÁCIA

GLOBÁLNE PODMIENKY

all-different ($\{X_1, \dots, X_k\}$)
gcc (GLOBAL CARDINALITY CONSTRAINT)
lex (LEXIKOGRAF. USP.) - 2 UKAZATELE: 2, B
regular (A, $\{X_1, \dots, X_k\}$)
grammar (G, $\{X_1, \dots, X_k\}$)
slide$_j$ (C, $\{X_1, \dots, X_k\}$)

ROZVRHOVACÍ PROBLÉM

PODMIENKA edge-finding

$$\min(\text{start}(\Omega)) + p(\Omega \cup \{A\}) > \max(\text{end}(\Omega \cup \{A\}))$$
$$\Rightarrow A \ll \Omega$$

PODMIENKA not_first

$$\text{start}(A) + p(\{A\} \cup \Omega) > \max(\text{end}(\Omega)) \Rightarrow \neg A \ll \Omega$$

CSOP (CONSTR. SATISF. OPTIM. PROBLEM)

BRANCH & BOUND

HEURISTICKÁ FCIA. h ODHAD. OBJ. FCIU f

PRÍLIŠ OBMEDZENÉ PROBLÉMY (OVER-CONSTRAINED)

ČIASTOČNÉ SPĺŇANIE PODMIENOK

PRIESTOR PROBLÉMOV $(PS, \leq)$

$\langle P, (PS, \leq), M, (N, S) \rangle$, POUŽITE EVAL. FCIE. g

WEIGHTED CSP - MIN. SÚČET VÁH NESPL. PODMIENOK

VALUED CSP, ŠTRUKTÚRA OCENENIA $(E, \otimes, >, \perp, \top)$

$\varphi : C \to E$, MINIMALIZUJEME $v(A) = \bigotimes_{A \text{ NEKONZ. S } c \in C} \varphi(c)$

CSP NAD POLOKRUHMI (SEMIRING-BASED CSP)

ŠTRUKTÚRA C-POLOKRUHU $(A, +, \times, 0, 1)$

MAX. AGREG. PREF. $\bigtimes_{c \in C} \delta_c(V \downarrow vars(c))$

HIERARCHIA CSP - PREFERENCIE PODMIENOK

KOMPARÁTOR, REŠPEKTUJE HIER.

CHYBOVÁ FCIA., DELTA STAR

LOKÁLNA PROPAGÁCIA

DELTA BLUE