Klasické plánování II a rozvrhování

Radek Mařík

CVUT FEL, K13133

16. dubna 2013

Obsah

Prohledávání prostoru stavů vs. plánů [Wic11]

prohledávání stavového prostoru

- prohledávání grafu, jehož uzly reprezentují stavy světa

prohledávání prostoru plánů

- prohledávání grafu, jehož uzly reprezentují částečné plány
- uzly: částečně určené plány
- hrany: operace zjemnění plánů
- řešení: částečně uspořádané plány
- částečný plán:
  - podmožina akcí
  - podmnožina organizační struktury
    - časové uspořádání akcí
    - zdůvodnění: co akce znamená pro plán
  - podmnožina vazeb proměnných

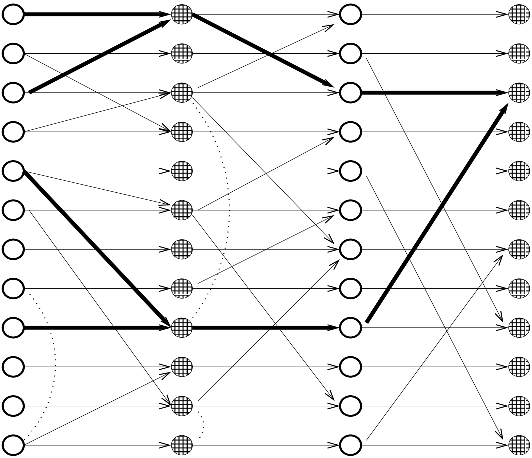
Plánování v prostoru plánů - omezující podmínky [Nau09]

- podmínka předcházení
  - a musí předcházet b
- vazební podmínka
  - podmínky nerovnosti, např. $v_1 \neq v_2$ nebo $v_1 \neq c$
  - podmínky rovnosti a substituce, např. $v_1 = v_2$ nebo $v_1 = c$
- kauzální vazby
  - použij akci a k vytvoření podmínky p potřebné pro akci b

GRAPHPLAN plánovač

- 1997
- plány jsou reprezentovány pomocí *plánovacího grafu*,
  - myšlenka je velmi podobná dynamickému programování či řešení toku sití,
- Všechny plány jsou konstruovány souběžně.
  - rozšiřování grafu (dopředný běh)
  - vyhledání plánu (zpětný běh)
- Plánovač udržuje relaci binární vzájemné výlučnosti (*mutex*) mezi uzly representující aplikované akce a výroky popisující stav.
- Problém s cyklením odstraněn.
- Nelze používat parametrizované akční schémata (instance).
  - Vytváří obrovský prostor výroků.
- Existuje řada podpůrných strategií, které podstatně urychlují plánovaní.
- Implementace jsou schopny zvládnout plány s 50-100 voláními metod do minut.

GraphPlan - plánovací graf



Implementace plánovačů

Počáteční pokusy

- STRIPS [1971] . . . , první plánovač, zpětné plánovaní dle podmínek akcí

Prostor stavů/plánů

- WARPLAN [1973] . . . lineární plánovač, Sussmanova anomálie řešení posouvání akcí
- PWEAK, TWEAK [1987], UCPOP [1992] . . . plánovače s častečným uspořádáním

Plánovací grafy

- GRAPHPLAN[1997] . . . první plánovač s plánovacím grafem
- Blackbox [1998] . . . kombinuje GRAPHPLAN a SATPLAN
- FF [2000] . . . plánovací graf jako heuristika s velmi rychlým dopředným a lokálním prohledáváním

Čas, rozvrhy a zdroje [RN10]

- Reprezentace klasického plánování
  - Co se má dělat,
  - V jakém pořádku.
- Rozšíření
  - Jak dlouho se vykonává daná akce,
  - Kdy se zahájí.
- Rozvrhování
  - časová omezení,
  - omezení na zdroje.
- Příklady
  - Rozvrh aerolinie,
  - Která letadla jsou přiřazena ke kterým letům,
  - Časy odletů a příletů,
  - Aerolinie má omezený počet zaměstnanců.
  - Obsluha, která je na jednom letu, nemůže být v tom samém času na letu jiném.

Obecné metodiky řešení [Rud13]

Úvod

- Grahamova klasifikace rozvrhovacích problémů

Obecné řešící metody

- Přesné řešící metody
  - metoda větví a mezí
- Heuristiky
  - řídící pravidla (*dispatching rules*)
  - paprskové prohledávání (*beam search*)
  - lokální prohledávání:
    simulované žíhání, tabu prohledávání, genetické algoritmy
- Matematické programování: formulace
  - lineární
  - celočíselné
- Programování s omezujícími podmínkami

Specifické metodiky [Rud13]

- **Plánování projektu:** reprezentace projektu, kritická cesta, kompromis mezi časem a cenou, pracovní síla.
- **Plánování úloh:** řídící pravidla, metoda větví a mezí, paprskové prohledávání, matematické prohledávání, posunování kritického místa.
- **Rozvrhování montážních systémů:** montážní linka s flexibilním časem, s fixním časem, s paralelními pracovními stanicemi.
- **Rezervace:** intervalové rozvrhování, rezervační systémy s rezervou.
- **Timetabling:** rozvrhování s operátory, rozvrhování s pracovní silou.
- **Rozvrhování zaměstnanců:** rozvrhování volných dnů, rozvrhování směn, cyklické rozvrhování směn.
- **Univerzitní rozvrhování:** teorie a praxe

Rozvrh [Rud13]

Rozvrh:

- dán umístěním úloh do konkrétního času a na konkrétní zdroje, kde mají být úlohy prováděny

Úplný rozvrh:

- v rozvrhu jsou umístěny všechny úlohy ze zadání problému

Částečný rozvrh:

- některé úlohy ze zadání problému nejsou umístěny/přiřazeny

Konzistentní rozvrh:

- rozvrh, ve kterém jsou splněna všechna omezení kladená na zdroje a umístěné/přiřazené úlohy, např.
  - úloha je naplánována v čase, kdy je dostupná
  - na jednom stroji (s jednotkovou kapacitou) běží nejvýše jedna úloha

Konzistentní úplný rozvrh vs. konzistentní částečný rozvrh

Optimální rozvrh:

- umístění úloh na stroje je optimální vzhledem k zadanému optimalizačnímu kritériu, např.
  - min C_{max}: makespan (čas dokončení poslední úlohy) je minimální

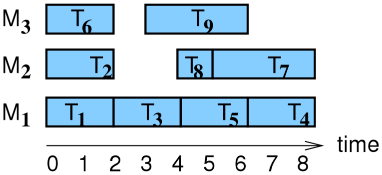
Definice pojmu rozvrhování [Rud13]

Rozvrhování

optimální alokace/přiřazení zdrojů množině úloh v čase

- omezené množství zdrojů,
- maximalizace zisku za daných omezení

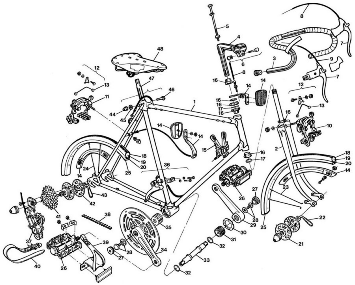
- Stroj $M_i, i = 1, \ldots, m$
- Úloha $J_j, j = 1, \ldots, n$
- (i, j) **operace** nebo provádění úlohy j na stroji i
  - úloha se může skládat z několika operací
  - příklad: úloha 4 má tři operace s nenulovou dobou trvání (2,4),(3,4),(6,4), tj. je prováděna na strojích 2,3,6

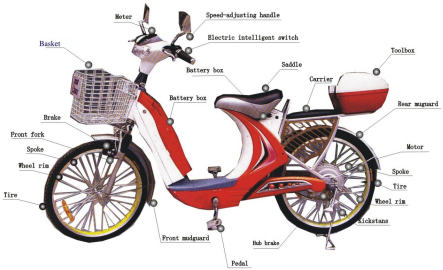


Statické a dynamické parametry úlohy [Rud13]

- Statické parametry úlohy
  - **doba trvání** p_{ij}, p_j:
    doba provádění úlohy j na stroji i
  - **termín dostupnosti** j **(release date)** r_j:
    nejdřívější čas, ve kterém může být úloha j prováděna
  - **termín dokončení (due date)** d_j:
    čas, do kdy by měla být úloha j nejpozději dokončena (preference)
  - vs. **deadline**:
    čas, do kdy musí být úloha j nejpozději dokončena (požadavek)
  - **váha** w_j:
    důležitost úlohy j relativně vzhledem k ostatním úloham v systému
- Dynamické parametry úlohy
  - **čas startu úlohy (start time)** S_{ij}, S_j:
    čas zahájení provádění úlohy j na stroji i
  - **čas konce úlohy (completion time)** C_{ij}, C_j:
    čas, kdy je dokončeno provádění úlohy j na stroji i

Příklad: montáž kola [Rud13]

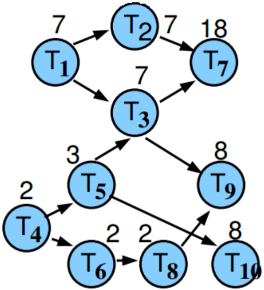


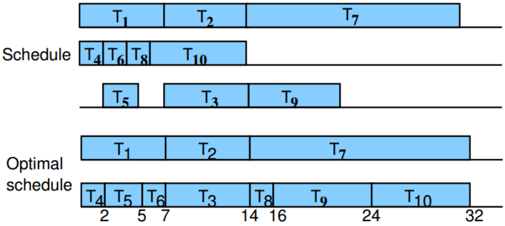


- 10 úloh s danou dobou trvání
- Precedenční podmínky
  - úlohu lze provést až po provedení zadané množiny úloh
- Nepreemtivní úlohy
  - úlohy nelze přerušit
- Optimalizační kritéria
  - minimalizace makespan
  - minimální počet pracovníků

Příklad: řešení montáž kola [Rud13]

- 10 úloh s danou dobou trvání
- Precedenční podmínky
  - úlohu lze provést až po provedení zadané množiny úloh
- Nepreemtivní úlohy
  - úlohy nelze přerušit
- Optimalizační kritéria
  - minimalizace makespan
  - minimální počet pracovníků





Příklady rozvrhování [Rud13]

- Plánování výroby polovodičů
  - velké množství různorodých výrobků,
  - odlišné množství vyráběných položek
  - cena za nastavení strojů, dodržení požadované doby výroby
- Plánování zásobovacích řetězců
  - např. lesnatá oblast – výroba papíru – výrobky z papíru – distribuční centra – koncový zákazník
  - minimalizace ceny výroby, dopravy, skladování
- Plánování výroby papíru
  - vstup dřevo, výstup role papíru, drahé stroje, různé typy papíru,
  - minimalizace výroby na sklad
- Automobilová montážní linka
  - výroba různých typů aut s různým vybavením
  - optimalizace výkonu, rovnoměrná zátěž
- Plnění limonád do lahví
  - 4 příchutě, každá příchuť má vlastní dobu plnění
  - minimalizovat dobu cyklu, jeden stroj

Příklady rozvrhování II [Rud13]

- Rozvrhování sester v nemocnici
  - odlišný počet sester v pracovní dny a o víkendu,
  - menší nároky při obsazování nočních směn,
  - určit přiřazení sester na směny, splnění požadavků, minimalizace ceny
- Plánování v prostředí Gridů
  - clustery, superpočítače, desktopy, speciální zařízení
  - plánování výpočetních úloh na zdroje
  - plánování datových přenosů a datové zpracování
- Univerzitní rozvrhování předmětů
  - Nalezení času a místnosti pro výuku předmětů na univerzitě
  - omezení kladena na umístění předmětů
  - optimalizace preferenčních požadavků na čas a místnosti
  - minimalizace počtu překrývajících se předmětů pro všechny studenty

Scheduling vs. timetabling [Rud13]

Scheduling . . . rozvrhování/plánování

- alokace zdrojů za daných podmínek na objekty umístěných v časoprostoru tak, že je minimalizována celková cena daných zdrojů
- důraz je kladen na uspořádání objektů, precedenční podmínky
  - př. plánování výroby: stanovení pořadí operací, důležitost časových návazností operací
- **schedule . . . rozvrh**: zahrnuje prostorové a časové informace

Timetabling . . . rozvrhování

- alokace zdrojů za daných podmínek na objekty umístěných v časoprostoru tak, že jsou co nejlépe splněna zadaná kritéria
- důraz kladen na konkrétní časové umístění objektů
- často vymezen předem časový horizont (počet rozvrhovaných slotů)
  - př. školní rozvrhování: předmětům přiřazen čas a místo vyuky
- **timetable . . . rozvrh**: ukazuje, kdy a kde se budou události konat.

Sequencing a Rostering [Rud13]

Sequencing . . . seřazení

- za daných podmínek:
  - konstrukce pořadí úloh, ve kterém budou prováděny
- **sequence . . . posloupnost**
  - pořadí, ve kterém jsou úlohy prováděny
- př. plnění limonád do lahví

Rostering . . . rozpis služeb

- umístění zdrojů za daných podmínek do slotů s pomocí vzorů (pattern)
- **roster . . . rozpis**
  - seznam jmen lidí, který určuje, které úlohy budou provádět a kdy
- př. rozpis sester v nemocnici, rozpis řidičů autobusů

Grahamova klasifikace [Rud13]

Grahamova klasifikace $\alpha|\beta|\gamma$

používá se pro popis rozvrhovacích problémů

- α: charakteristiky stroje
  - popisuje způsob alokace úloh na stroje
- β: charakteristiky úloh
  - popisuje omezení aplikovaná na úlohy
- γ: optimalizační kritéria
- složitost pro jednotlivé rozvrhovací problémy

Příklady

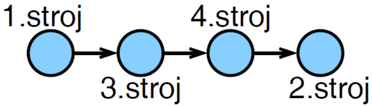
- $P3|prec|C_{max}$: montáž kola
- $Pm|r_j|\sum w_j C_j$: paralelní stroje

Vlastnosti stroje α [Rud13]

- **Jeden stroj 1**: $1|\ldots|\ldots$
- **Identické paralelní stroje** *Pm*
  - *m* identických strojů zapojených paralelně (se stejnou rychlostí)
  - úloha je dána jedinou operací
  - úloha může být prováděna na libovolném z *m* strojů
- **Paralelní stroje s různou rychlostí** *Qm*
  - doba trvání úlohy j na stroji i přímo závislá na jeho rychlosti v_i
  - $p_{ij} = p_j / v_i$
  - př. několik počítačů s různou rychlostí procesoru
- **Nezávislé paralelní stroje s různou rychlostí** *Rm*
  - stroje mají různou rychlost pro různé úlohy
  - stroj i zpracovává úlohu j rychlostí v_{ij}
  - $p_{ij} = p_j / v_{ij}$
  - př. vektorový počítač počítá vektorové úlohy rychleji než klasické PC

Multi-operační (shop) problémy [Rud13]

- **Multi-operační (shop) problémy**
  - jedna úloha je prováděna postupně na několika strojích
    - úloha j se skládá z několika operací (i, j)
    - operace (i, j) úlohy j je prováděna na stroji i po dobu p_{ij}
    - příklad: úloha j se 4 operacemi $(1, j), (2, j), (3, j), (4, j)$



- Multi-operační problémy jsou klasické detailně studované problémy **operačního výzkumu**
- Reálné problémy ale často mnohem komplikovanější
  - využití znalostí o podproblémech nebo zjednodušených problémech a jejich řešicích metodách

Flow shop α [Rud13]

- **Flow shop** *Fm*
  - multi-operační problém s *m* stroji v sérii
  - každá úloha musí být prováděna na všech strojích
  - úloha musí být prováděna na všech strojích ve stejném pořadí
    - nejdříve se úloha provádí na 1. stroji, pak na 2., ...
- **Flexible flow shop** *FFs*
  - zobecnění flow shop problému
  - *s* fází, každé fázi přísluší paralelní stroj
  - tj. multi-operační problém s *s* paralelními stroji
  - úloha musí projít všemi fázemi ve stejném pořadí
    - nejprve se úloha provádí na paralelním stroji 1. fáze, pak na paralelním stroji 2. fáze, ...
  - na paralelním stroji příslušejícím dané fázi může být úloha prováděna na libovolném stroji

Open shop & job shop [Rud13]

- **Job shop** Jm
  - multi-operační problém s m stroji
  - pořadí provádění operací pro každou úlohu je předem určeno
    - doba zpracování úlohy na některých strojích může být nulová
  - $(i,j) \rightarrow (k,j)$ určuje, že úloha j má být prováděna na stroji i dříve než na stroji k
  - příklad: $(2,j) \rightarrow (1,j) \rightarrow (3,j) \rightarrow (4,j)$
- **Open shop** Om
  - multi-operační problém s m stroji
  - doba zpracování úlohy na některých strojích může být nulová
  - rozvrhovač určí, v jakém pořadí je úloha prováděna na strojích

Omezení β [Rud13]

- **Precedenční podmínky** *prec*
  - lineární posloupnost, stromová struktura
  - pro úlohy a, b píšeme $a \rightarrow b$, což znamená $S_a + p_a \leq S_b$
  - příklad: montáž kola
- **Přerušení úlohy (preemptions)** *pmtn*
  - při příchodu úlohy s vyšší prioritou je současná úloha přerušena
- **Vhodnost stroje** M_j
  - podmnožina strojů M_j, na níž lze provádět úlohu j
  - přiřazení místností: postačující velikost učebny
  - hry: počítač s HW grafickou knihovnou
- **Omezení na pracovní sílu** W, W_l
  - do problému zavedeme další typ zdroje
  - stroje mohou potřebovat operátory a úlohy lze provádět jen tehdy, pokud jsou dostupní, W operátorů
  - mohou existovat různé skupiny operátorů se specifickou kvalifikací W_l je počet operátorů ve skupině l

Omezení (pokračování) β [Rud13]

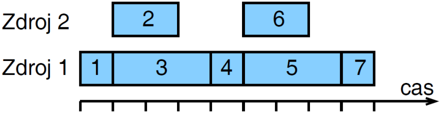
- **Směrovací (*routing*) omezení**
  - udávají, na kterých strojích musí být úloha prováděna
  - pořadí provádění úlohy v multi-operačních problémech
    - job shop problém: pořadí operací předem stanoveno
    - open shop problém: pořadí operací úlohy (route for the job) stanoveno až při rozvrhování
- **Nastavovací (*setup*) doba a cena** $s_{ijk}, c_{ijk}, s_{jk}, c_{jk}$
  - závislé na posloupnosti provádění
  - s_{ijk} čas nutný pro provádění úlohy k po úloze j na stroji i
  - c_{ijk} cena nutná pro provádění úlohy k po úloze j na stroji i
  - s_{jk}, c_{jk} čas/cena nezávislý na stroji
  - příklady
    - plnění limonád do lahví
    - problém obchodního cestujícího $1|s_{jk}|C_{max}$

Optimalizace: výkon a makespan γ [Rud13]

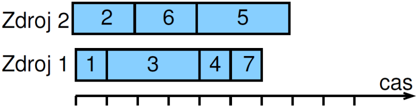
- **Makespan** C_{max}: maximální čas konce úloh

$$C_{max} = max(C_1, \ldots, C_n)$$

- Příklad: $C_{max} = max\{1, 3, 4, 5, 8, 7, 9\} = 9$



- Cíl: **minimalizace makespan** často
  - maximalizuje **výkon (***throughput***)**
  - zajišťuje **rovnoměrné zatížení strojů (***load balancing***)**
  - příklad: $C_{max} = max\{1, 2, 4, 5, 7, 4, 6\} = 7$



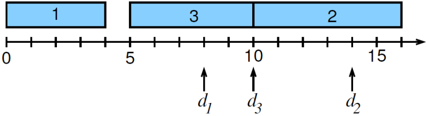
  - Velmi často používané a základní kritérium

Optimalizace: zpoždění γ [Rud13]

- **Zpoždění (*lateness*)** úlohy j: $L_{max} = C_j - d_j$
- **Maximální zpoždění** L_{max}

$$L_{max} = max(L_1, \ldots, L_n)$$

- Cíl: **minimalizace maximálního zpoždění**
- Příklad:

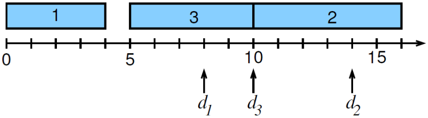


$$\begin{aligned} L_{max} &= max(L_1, L_2, L_3) = \\ &= max(C_1 - d_1, C_2 - d_2, C_3 - d_3) = \\ &= max(4 - 8, 16 - 14, 10 - 10) = \\ &= max(-4, 2, 0) = 2 \end{aligned}$$

Optimalizace: nezáporné zpoždění γ [Rud13]

- **Nezáporné zpoždění (*tardiness*)** úlohy j: $T_j = max(C_j - d_j, 0)$
- **Celkové zpoždění**

$$\sum_{j=1}^{n} T_j$$



- Cíl: **minimalizace celkového zpoždění**
- Příklad: $T_1 + T_2 + T_3 =$

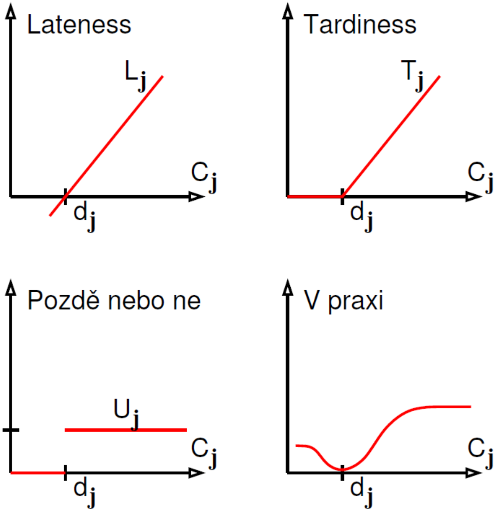
$$\begin{aligned}
&= \max(C_1 - d_1, 0) + \max(C_2 - d_2, 0) + \max(C_3 - d_3, 0) = \\
&= \max(4 - 8, 0) + \max(16 - 14, 0) + \max(10 - 10, 0) = \\
&= 0 + 2 + 0 = 2
\end{aligned}$$

- **Celkové vážené zpoždění**

$$\sum_{j=1}^{n} w_j T_j$$

- Cíl: **minimalizace celkového váženého zpoždění**

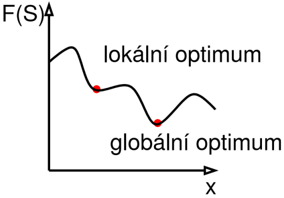
Termín dokončení a grafy γ [Rud13]



Konstruktivní vs. lokální metody [Rud13]

- **Konstruktivní metody**
  - začneme s prázdným rozvrhem
  - do rozvrhu přidáváme postupně jednotlivé úlohy tak, aby byl rozvrh stále konzistentní
- **Lokální prohledávání**
  - začneme s úplným nekonzistentním rozvrhem
    - triviálně: s náhodně vygenerovaným
  - snažíme se najít lepší "podobný" rozvrh lokálními změnami
  - kvalitu rozvrhu posuzujeme optimalizačními kritérii
    - např. makespan
  - optimalizační kritéria vyhodnocují také konzistenci rozvrhu
    - např. počet porušených precedenčních omezení
  - **Hybridní přístupy**
    - kombinace obou metod

Algoritmus lokálního prohledávání [Rud13]



1. **Inicializace**
   - $k = 0$
   - výběr iniciálního rozvrhu S_0
   - zaznamenání dosud nejlepšího rozvrhu: $S_{best} = S_0$ a $cost_{best} = F(S_0)$
2. **Výběr a aktualizace**
   - výběr rozvrhu z okolí: $S_{k+1} \in N(S_k)$
   - pokud kriterium přijetí rozvrhu nesplňuje žádný prvek $N(S_k)$, pak algoritmus končí
   - jestliže $F(S_{k+1}) < cost_{best}$ pak $S_{best} = S_{k+1}$ a $cost_{best} = F(S_{k+1})$
3. **Ukončení**
   - jestliže platí podmínky ukončení, pak algoritmus končí
   - jinak $k = k + 1$ a skok na krok 2.

Jeden stroj + nepreemptivní úlohy [Rud13]

- **Reprezentace rozvrhu**
  - permutace n úloh
  - příklad se šesti úlohami: $1, 4, 2, 6, 3, 5$
- **Definice okolí**
  - párová výměna sousedních úloh
    - $n - 1$ možných okolí
    - příklad: $1, 4, 2, 6, 3, 5$ se změní např. na $1, 4, 2, 6, 5, 3$
  - nebo výběr libovolné úlohy v rozvrhu a umístění na libovolnou pozici
    - $\leq n(n-1)$ možných okolí
    - příklad: z $1, 4, 2, 6, 3, 5$ náhodně vybereme 4 a dáme ji jinam: $1, 2, 6, 3, 4, 5$

Kritérium výběru rozvrhu [Rud13]

- Kritérium výběru rozvrhu
  - **kritérium přijetí/odmítnutí rozvrhu**
- Hlavní rozdíl mezi většinou metod
  - akceptovat vždy lepší rozvrh?
  - někdy akceptovat i horší rozvrh?
- Metoda
  - pravděpodobnostní
    - **náhodná procházka**: s malou pravděpodobností (např. 0.01) akceptujeme i horší rozvrh
    - **simulované žíhání**
  - deterministická
    - **tabu prohledávání**: udržujeme tabu seznam několika posledních stavů/změn, které jsou pro další výběr nepřípustné

Tabu prohledávání [Rud13]

- **Deterministické kritérium přijetí/odmítnutí rozvrhu**
- Udržován **tabu seznam** několika posledních změn v rozvrhu
  - každá nová změna je umístěna na vrchol tabu seznamu
    - př. uchovávané změny: výměna úloh j a k
  - tabu seznam = seznam zakázaných změn
  - okolí omezeno na rozvrhy, které nepožadují změnu z tabu seznamu
    - zabraňuje cyklení
    - příklad triviáního cyklení:
      první krok: prohození úloh 3 a 4, druhý krok: prohození úloh 4 a 3
  - pevná délka seznamu (typicky: 5-9)
    - nejstarší změny z tabu seznamu odstraněny
    - příliš malá délka: nebezpečí cyklení
    - příliš velká délka: může omezit prohledávání příliš
- **Aspirační kritérium**
  - určuje, kdy je možné akceptovat i změny v tabu seznamu
  - př. změna z tabu seznamu povolena, pokud zlepšeno $F(S_{best})$

Algoritmus tabu prohledávání [Rud13]

1. - $k = 1$
   - výběr iniciálního rozvrhu S_1 použitím heuristiky, $S_{best} = S_1$
2. - výběr $S_c \in N(S_k)$
   - jestliže je změna $S_k \rightarrow S_c$ zakázána, protože je v tabu seznamu pak běž na krok 2
3. - jestliže změna $S_k \rightarrow S_c$ není zakázána tabu seznamem pak $S_{k+1} = S_c$,
     ulož reversní změnu na vrchol tabu seznamu
     posuň další pozice v tabu seznamu o pozici níže
     smaž poslední položku z tabu seznamu
   - jestliže $F(S_c) < F(S_{best})$ pak $S_{best} = S_c$
4. - $k = k + 1$
   - jestliže platí podmínka ukončení pak konec
     jinak běž na krok 2.

Příklad: tabu seznam [Rud13]

Uvažujte rozvrhovací problém s $1|d_j|\sum w_j T_j$

- opakování: $T_j = \max(C_j - d_j, 0)$

úlohy	1	2	3	4
p_j	10	10	13	4
d_j	4	2	1	12
w_j	14	12	1	12

- Okolí: všechny rozvrhy získané párovou výměnou sousedních úloh
- Výběr rozvrhu z okolí: vybereme nejlepší rozvrh
- Tabu seznam: páry úloh (j, k), které byly přehozeny při posledních dvou změnách
- Aplikujte tabu prohledávání pro iniciální řešení $(2, 1, 4, 3)$
- Proveďte čtyři iterace

Příklad: tabu seznam - řešení I [Rud13]

úlohy	1	2	3	4
p_j	10	10	13	4
d_j	4	2	1	12
w_j	14	12	1	12

$S_1 = (2, 1, 4, 3)$
$F(S_1) = \sum w_j T_j = 12 \cdot 8 + 14 \cdot 16 + 12 \cdot 12 + 1 \cdot 36 = 500 = F(S_{best})$
$F(1, 2, 4, 3) = 480$
$F(2, \underline{4}, \underline{1}, 3) = 436 = F(S_{best})$
$F(2, 1, 3, 4) = 652$
Tabu seznam: $\{(1, 4)\}$

$S_2 = (2, 4, 1, 3), F(S_2) = 436$
$F(\underline{4}, \underline{2}, 1, 3) = 460$
$F(2, 1, 4, 3)(= 500)$ tabu!
$F(2, 4, 3, 1) = 608$
Tabu seznam: $\{(2, 4), (1, 4)\}$

$S_3 = (4, 2, 1, 3), F(S_3) = 460$
$F(2, 4, 1, 3)(= 436)$ tabu!
$F(4, \underline{1}, \underline{2}, 3) = 440$
$F(4, 2, 3, 1) = 632$
Tabu seznam: $\{(2, 1), (2, 4)\}$

Příklad: tabu seznam - řešení II [Rud13]

úlohy	1	2	3	4
p_j	10	10	13	4
d_j	4	2	1	12
w_j	14	12	1	12

$S_3 = (4, 2, 1, 3), F(S_3) = 460$
$F(2, 4, 1, 3)(= 436)$ tabu!
$F(4, \underline{1}, \underline{2}, 3) = 440$
$F(4, 2, 3, 1) = 632$
Tabu seznam: $\{(2, 1), (2, 4)\}$

$S_4 = (4, 1, 2, 3), F(S_4) = 440$
$F(\underline{1}, \underline{4}, 2, 3) = 408 = F(S_{best})$
$F(4, 2, 1, 3)(= 460)$ tabu!
$F(4, 1, 3, 2) = 586$
Tabu seznam: $\{(4, 1), (2, 1)\}$

$F(S_{best}) = 408$

Poděkování

- První verze této prezentace byla připravena jako zkrácená verze prezentací Hany Rudové [Rud13]

Literatura I

Dana Nau.
CMSC 722, ai planning (fall 2009), lecture notes.
http://www.cs.umd.edu/class/fall2009/cmsc722/, 2009.

Stuart J. Russell and Peter Norvig.
Artificial Intelligence, A Modern Approach.
Pre, third edition, 2010.

Hana Rudová.
PA167 Rozvrhování, lecture notes, in Czech.
http://www.fi.muni.cz/ hanka/rozvrhovani/, March 2013.

Gerhard Wickler.
A4m33pah, lecture notes.
http://cw.felk.cvut.cz/doku.php/courses/a4m33pah/prednasky, February 2011.