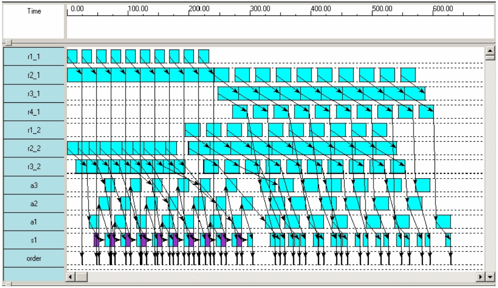
PB165 Grafy a sítě: Úvod k plánování a rozvrhování

Grafy a sítě v plánování a rozvrhování: osnova

1. Problém rozvrhování: vlastnosti stroje, omezení, optimalizace
2. Precedenční omezení a disjunktivní grafová reprezentace: korespondence mezi rozvrhem a grafem
3. Plánování projektu (precedenční omezení): kritická cesta, kompromisní heuristika
4. Barvení grafu: algoritmus saturace a problémy přiřazení místností, rezervační problém, plánování operátorů
5. Hranově ohodnocené grafy: obchodní cestující, doba na dopravu, plánování na počítačových sítích
6. Plánování s komunikací a s precedencemi: plánování seznamem, heuristiky mapování, shlukovací heuristiky
7. Paralelní úlohy s průběžnou komunikací: vyvažování zátěže a rozdělení grafu

Rozvrhování na FI: PA167 Rozvrhování, PA163 Omezující podmínky

Rozvrhování a plánování (scheduling)

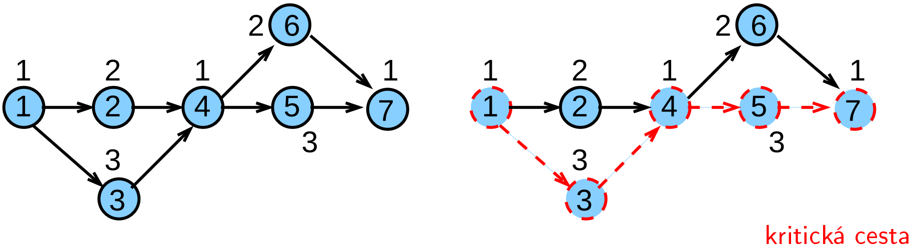


Visopt ShopFloor System

- Zdroj/stroj
  - kapacita
  - dostupnost v čase
  - rychlost
- Úloha/aktivita
  - nejdřívější startovní čas
  - nejpozdější koncový čas
  - doba trvání (na referenčním zdroji)
  - počet zdrojů
  - alternativní zdroje
- Rozvrhování
  - optimální alokace/přiřazení zdrojů v čase množině úloh
    - omezené množství zdrojů
    - maximalizace zisku za daných omezení

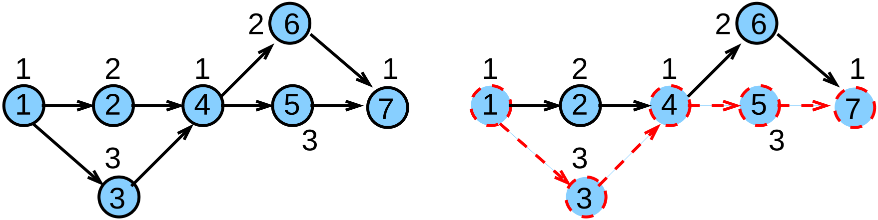
Příklad: rozvrhování s precedencemi

- Rozvrhování 7 úloh na 2 zdrojích
  - doba trvání úlohy + precedenční podmínky
  - nalezení rozvrhu tak, aby se minimalizovala doba nutná na realizaci všech úloh



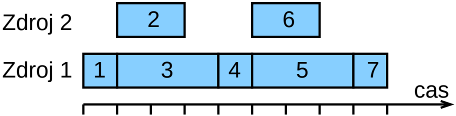
Příklad: rozvrhování s precedencemi

- Rozvrhování 7 úloh na 2 zdrojích
  - doba trvání úlohy + precedenční podmínky
  - nalezení rozvrhu tak, aby se minimalizovala doba nutná na realizaci všech úloh



kritická cesta

- Možný rozvrh
  - na kritické (nejdelší) cestě nesmí vzniknout zdržení



Úlohy, stroje

- Stroje $i = 1, \dots, m$
- Úlohy $j = 1, \dots, n$
- (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 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 musí být úloha j nejpozději dokončena
  - váha w_j:
    důležitost úlohy j relativně vzhledem k ostatním úloham v systému

Úlohy, stroje

- Stroje $i = 1, \ldots, m$
- Úlohy $j = 1, \ldots, n$
- (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 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 musí být úloha j nejpozději dokončena
  - 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, kdy začne 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

Grahamova klasifikace

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

Grahamova klasifikace

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

používá se pro popis rozvrhovacích problémů

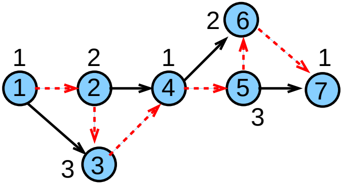
- α: charakteristiky stroje
  - popisuje způsob alokace úloh na stroje
- β: charakteristiky úloh
  - popisuje omezení aplikovaná na úlohy
- γ: optimalizační kritéria

http://www.mathematik.uni-osnabrueck.de/research/OR/class/

- složitost a algoritmy pro jednotlivé rozvrhovací problémy

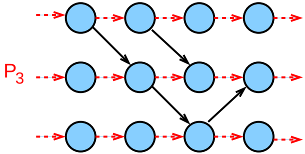
Jeden stroj 1: $1|\ldots|\ldots$

- nejjednodušší varianta
- speciální případ dalších složitějších prostředí stroje



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



Stroje s různou rychlostí α

Paralelní stroje s různou rychlostí Qm

- doba trvání úlohy j na stroji i přímo závislá na jeho rychlosti v_i
- $p_{ij} = p_j / v_i$
- příklad:
  několik počítačů s různou rychlostí procesoru

Stroje s různou rychlostí α

Paralelní stroje s různou rychlostí Qm

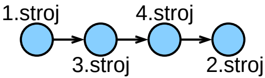
- doba trvání úlohy j na stroji i přímo závislá na jeho rychlosti v_i
- $p_{ij} = p_j / v_i$
- příklad:
  několik počítačů s různou rychlostí procesoru

Nezávislé paralelní stroje s různou rychlostí Rm

- stroje mají různou rychlost pro různé úlohy
- stroj i zpracovává úlohu j rychlostí v_{ij}
- $p_{ij} = p_j / v_{ij}$
- příklad:
  vektorový počítač počítá vektorové úlohy rychleji než klasické PC

Multi-operační *(shop)* problémy

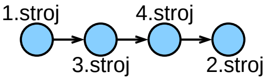
- jedna úloha je prováděna postupně na několika strojích
  - úloha j se skládá z několika operací (i, j)
  - operace (i, j) úlohy j je prováděna na stroji i po dobu p_{ij}
  - příklad: úloha j se 4 operacemi
    $(1, j), (2, j), (3, j), (4, j)$



Multi-operační *(shop)* problémy α

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
    $(1, j), (2, j), (3, j), (4, j)$



Open shop Om

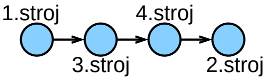
- multi-operační problém s m stroji (žádné nové vlastnosti)

Job shop Jm

- multi-operační problém s m stroji
- pořadí provádění operací je předem určeno
  - příklad: $(2, j) \to (1, j) \to (3, j) \to (4, j)$

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 $(1, j), (2, j), (3, j), (4, j)$



Open shop Om

- multi-operační problém s m stroji (žádné nové vlastnosti)

Job shop Jm

- multi-operační problém s m stroji
- pořadí provádění operací je předem určeno
  - příklad: $(2, j) \rightarrow (1, j) \rightarrow (3, j) \rightarrow (4, j)$

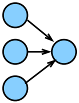
Multi-operační problémy = klasické detailně studované problémy operačního výzkumu

- reálné problémy mnohem komplikovanější
- metody řešení lze použít jako základ pro řešení složitějších problémů
- př. automobilová výrobní linka

Omezení β

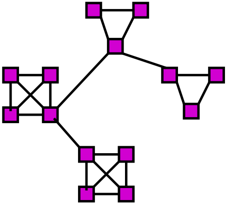
Precedenční podmínky *prec*

- úloha může být prováděna až po skončení další(ch) úloh
- pro úlohy a, b píšeme $a \rightarrow b$, což znamená $S_a + p_a \leq S_b$



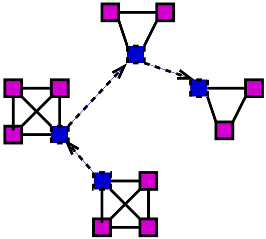
Vhodnost stroje M_j

- podmnožina strojů M_j, na níž lze provádět úlohu j
- př. úloha může být prováděna pouze na těch strojích v počítačové síti, kde jsou dostupná data



Směrovací *(routing)* omezení

- udávají, na kterých strojích musí být úloha prováděna
- vazba na směrování v počítačových sítích
- 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

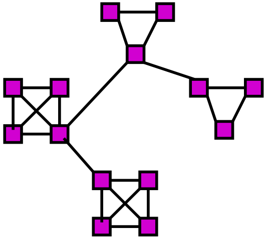


Doprava a komunikace

t_{jkl}, t_{kl}, t_j

- doba nutná na přepravu úlohy j mezi dvěma zařízeními k a l
  - t_{jkl} doba na přepravu ze stroje k na stroj l pro úlohu j
  - t_{kl} doba nezávislá na úloze
  - t_j doba nezávislá na strojích
- omezení na přepravované/přenášené množství a možnou dobu přepravy
- omezení na propustnost (kapacitu) hrany/linky
- omezení na vzdálenost uzlů pro přepravu/přenos

t_{kl} dáno vzdáleností uzlů v síti/grafu:

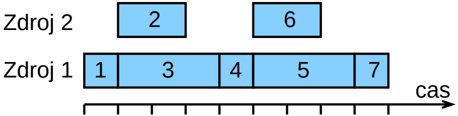


Optimalizace: *makespan* γ

- Maximální čas konce úloh *(makespan)*

$$\mathbf{C}_{\max} = \max(C_1, \ldots, C_n)$$

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

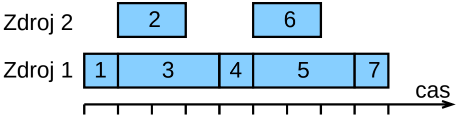


Optimalizace: *makespan* γ

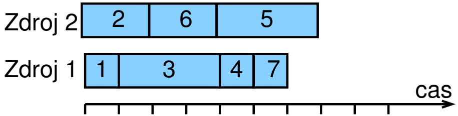
- Maximální čas konce úloh *(makespan)*

$$\mathbf{C}_{\max} = \max(C_1, \ldots, C_n)$$

- 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: $C_{\max} = \max\{1, 2, 4, 5, 7, 4, 6\} = 7$



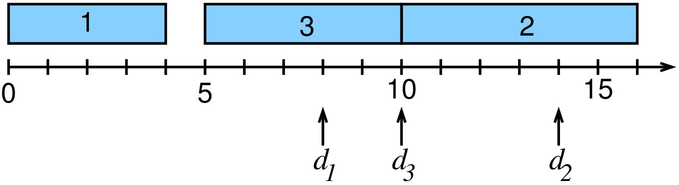
- Velmi často používané kritérium

Optimalizace: zpoždění γ

- Zpoždění *(lateness)* úlohy j: $L_j = C_j - d_j$
- Cíl: minimalizace zpoždění

$$\mathbf{L}_{\max} = \max(L_1, \ldots, L_n)$$

- Příklad:



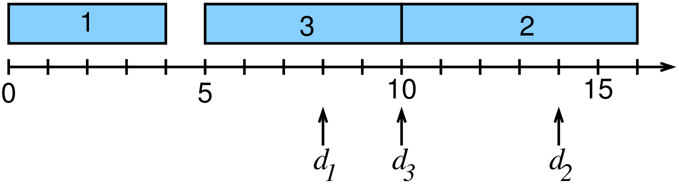
$$\begin{aligned} L_{max} &= \max(L_1, L_2, L_3) = \\ &= \max(C_1 - d_1, C_2 - d_2, C_3 - d_3) = \\ &= \max(4 - 8, 16 - 14, 10 - 10) = \\ &= \max(-4, 2, 0) = 2 \end{aligned}$$

Optimalizace: nezáporné zpoždění γ

- Nezáporné zpoždění *(tardiness)* úlohy j: $T_j = \max(C_j - d_j, 0)$
- Cíl: minimalizace celkového zpoždění

$$\sum_{j=1}^{n} T_j \qquad \text{celkové zpoždění úloh}$$

- Příklad:



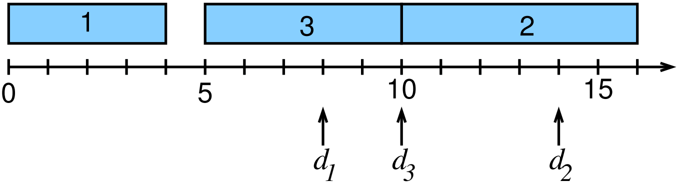
$$T_1 + T_2 + T_3 = \max(C_1 - d_1, 0) + \max(C_2 - d_2, 0) + \max(C_3 - d_3, 0) =$$
$$\max(4-8, 0) + \max(16-14, 0) + \max(10-10, 0) = 0+2+0 = 2$$

Optimalizace: nezáporné zpoždění γ

- Nezáporné zpoždění *(tardiness)* úlohy j: $T_j = \max(C_j - d_j, 0)$
- Cíl: minimalizace celkového zpoždění

$$\sum_{j=1}^{n} T_j \qquad \text{celkové zpoždění úloh}$$

- Příklad:

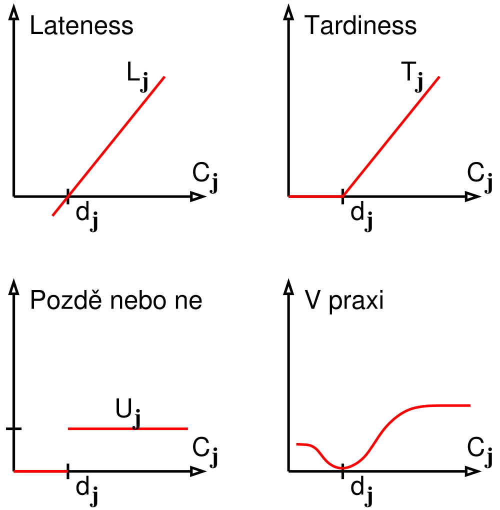


$$T_1 + T_2 + T_3 = \max(C_1 - d_1, 0) + \max(C_2 - d_2, 0) + \max(C_3 - d_3, 0) =$$
$$\max(4-8, 0) + \max(16-14, 0) + \max(10-10, 0) = 0+2+0 = 2$$

- Cíl: minimalizace celkového váženého zpoždění

$$\sum_{j=1}^{n} w_j T_j \qquad \text{celkové vážené zpoždění úloh}$$

Termín dokončení a grafy γ



Příklady rozvrhovacích problému s Grahamovou klasifikací

- $1|prec|C_{\max}$
  - plánování úloh provázaných precedencemi na jednom stroji s cílem minimalizovat makespan

Příklady rozvrhovacích problému s Grahamovou klasifikací

- $1|prec|C_{\max}$
  - plánování úloh provázaných precedencemi na jednom stroji s cílem minimalizovat makespan
- $Pm|r_j, M_j|\sum w_j T_j$
  - systém s m stroji zapojenými paralelně, kde
    - úloha j přijde v čase r_j a má být naplánována do času d_j
    - úloha j může být naplánována pouze na podmnožině strojů dané M_j

    a pokud není úloha j zpracována včas, tak je penalizována $w_j T_j$

Příklady rozvrhovacích problému s Grahamovou klasifikací

- $1|prec|C_{\max}$
  - plánování úloh provázaných precedencemi na jednom stroji s cílem minimalizovat makespan
- $Pm|r_j, M_j|\sum w_j T_j$
  - systém s m stroji zapojenými paralelně, kde
    - úloha j přijde v čase r_j a má být naplánována do času d_j
    - úloha j může být naplánována pouze na podmnožině strojů dané M_j

    a pokud není úloha j zpracována včas, tak je penalizována $w_j T_j$
- $Jm||C_{\max}$
  - job shop problém, kde je cílem minimalizovat makespan
  - velmi často studovaný a velmi známý typ job-shop problému

Příklady rozvrhovacích problému s Grahamovou klasifikací

- $1|prec|C_{\max}$
  - plánování úloh provázaných precedencemi na jednom stroji s cílem minimalizovat makespan
- $Pm|r_j, M_j|\sum w_j T_j$
  - systém s m stroji zapojenými paralelně, kde
    - úloha j přijde v čase r_j a má být naplánována do času d_j
    - úloha j může být naplánována pouze na podmnožině strojů dané M_j

    a pokud není úloha j zpracována včas, tak je penalizována $w_j T_j$
- $Jm||C_{\max}$
  - job shop problém, kde je cílem minimalizovat makespan
  - velmi často studovaný a velmi známý typ job-shop problému
- $P\infty|prec|C_{\max}$
  - problém s neomezeným počtem strojů zapojených paralelně, kde jsou úlohy provázány precedenčními podmínkami a kde je cílem minimalizovat makespan
  - klasický problém plánování projektu

Úvod k plánování a rozvrhování

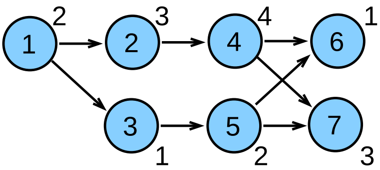
1. Terminologie a klasifikace
   - Úvod
   - Vlastnosti stroje
   - Omezení
   - Optimalizace
   - Příklady

2. Grafová reprezentace pro:
   - Precedenční omezení
   - Disjunktivní grafová reprezentace

Precedenční omezení

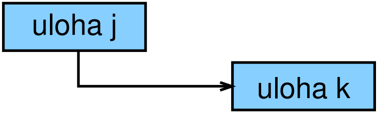
- Úloha může být prováděna až po skončení další(ch) úloh
  - úloha a před úlohou b: $a \rightarrow b : S_a + p_a \leq S_b$
- Orientovaný acyklický vrcholově ohodnocený graf
  - uzly reprezentují úlohy
  - hrany reprezentují precedenční podmínky
  - ohodnocení vrcholu reprezentuje dobu trvání
  - graf bez cyklů (pro cyklický graf neexistuje žádné řešení)

Úloha	Doba trvání	Předchůdci
1	2	–
2	3	1
3	1	1
4	4	2
5	2	3
6	1	4,5
7	3	4,5

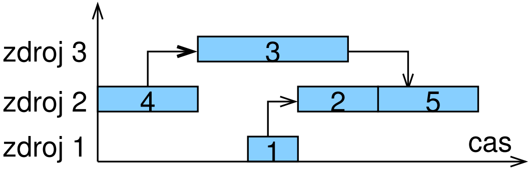


Úloha jako obdélník

- Úloha jako uzel lze převést na úloha jako obdelník



- Horizontální strany obdelníku použity jako časové osy odpovídající době provádění úlohy



Precedenční omezení: aplikace

- Příklad: zprostředkování, instalace
  a testování rozsáhlého počítačového systému
  - projekt zahrnuje
    - evaluace a výběr hardware, vývoj software, nábor a školení lidí, testování a ladění systému, ...
  - precedenční vztahy
    - některé úlohy mohou být prováděny paralelně
    - úloha musí být realizována až po dokončení jiných úloh
  - cíl: minimalizovat čas na realizaci celého projektu, tj. makespan

Precedenční omezení: aplikace

- Příklad: zprostředkování, instalace
  a testování rozsáhlého počítačového systému
  - projekt zahrnuje
    - evaluace a výběr hardware, vývoj software, nábor a školení lidí, testování a ladění systému, ...
  - precedenční vztahy
    - některé úlohy mohou být prováděny paralelně
    - úloha musí být realizována až po dokončení jiných úloh
  - cíl: minimalizovat čas na realizaci celého projektu, tj. makespan
- Obecně: problémy plánování projektu
  - příští přednáška
- Rozšíření: plánování workflows
  1. orientovaný acyklický graf pro provádění úloh na počítačové síti
  2. obecné rozšíření: cyklické grafy + podmínky vyhodnocení cyklů

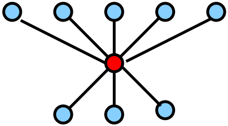
Disjunktivní grafová reprezentace a multi-operační rozvrhování

- n úloh
- m strojů
- Jedna úloha je prováděna postupně na několika strojích
- Operace (i, j): provádění úlohy j na stroji i
- p_{ij}: trvání operace (i, j)
- Pořadí operací úlohy je předem stanoveno:
  - $(i, j) \rightarrow (k, j)$ specifikuje, že úloha j má být prováděna na stoji i dříve než na stroji k
- Cíl: rozvrhovat úlohy na strojích
  - bez překrytí na strojích
  - bez překrytí v rámci úlohy
  - minimalizace *makespan* C_{max}

  tedy jedná se o job shop problém s minimalizací makespan $Jm||C_{\max}$

Aplikace: automobilová montážní linka

- Rozdílné typy aut na montážní lince
  - dvou-dveřové kupé, čtyř-dveřový sedan, ...
  - rozdílné barvy
  - rozdílné vybavení: automatická vs. manuální převodovka, posuvná střech, ...
- Kritická místa (*bottlenecks*)

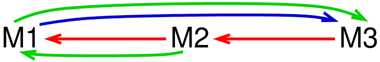


  - výkon stroje ovlivňuje tempo výroby
  - např. lakování (změna barvy vyžaduje časově náročné čištění)
- Cíl
  - maximalizace výkonnosti vhodným seřazením automobilů,
  - rovnoměrná pracovní zátěž na jednotlivých výrobních místech

Příklad: job shop problém

Data:

- stroje: $M1, M2, M3$
- úlohy: $J1 : (3,1) \rightarrow (2,1) \rightarrow (1,1)$
  $J2 : (1,2) \rightarrow (3,2)$
  $J3 : (2,3) \rightarrow (1,3) \rightarrow (3,3)$

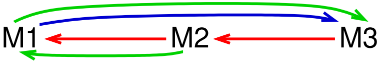


- doby trvání: $p31 = 4, p21 = 2, p11 = 1$
  $p12 = 3, p32 = 3$
  $p23 = 2, p13 = 4, p33 = 1$

Příklad: job shop problém

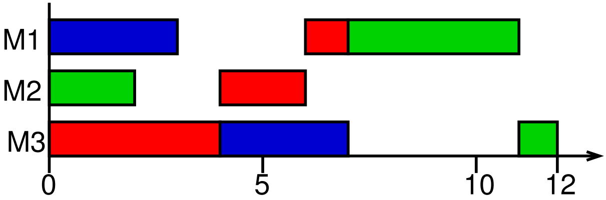
Data:

- stroje: $M1, M2, M3$
- úlohy: $J1 : (3,1) \rightarrow (2,1) \rightarrow (1,1)$
  $J2 : (1,2) \rightarrow (3,2)$
  $J3 : (2,3) \rightarrow (1,3) \rightarrow (3,3)$



- doby trvání: $p31 = 4, p21 = 2, p11 = 1$
  $p12 = 3, p32 = 3$
  $p23 = 2, p13 = 4, p33 = 1$

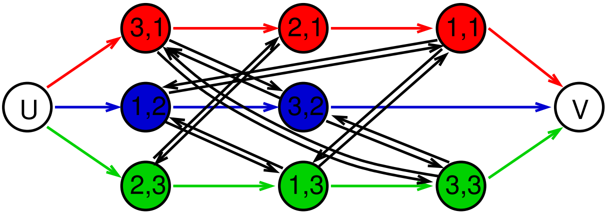
Řešení:



Disjunktivní grafová reprezentace

Graf $G = (N, A \cup B)$

- uzly odpovídají operacím $N = \{(i,j) | (i,j) \text{ je operace}\}$
- konjunktivní hrany A reprezentují pořadí operací úlohy
  - $(i,j) \to (k,j) \in A \iff$ operace (i,j) předchází (k,j)
- disjunktivní hrany B reprezentují konflikty na strojích
  - dvě operace (i,j) a (i,l) jsou spojeny dvěma opačně orientovanými hranami
- dva pomocné uzly U a V reprezentující zdroj a stok
- hrany z U ke všem prvním operacím úlohy
- hrany ze všech posledních operací úlohy do V



Výběr hran

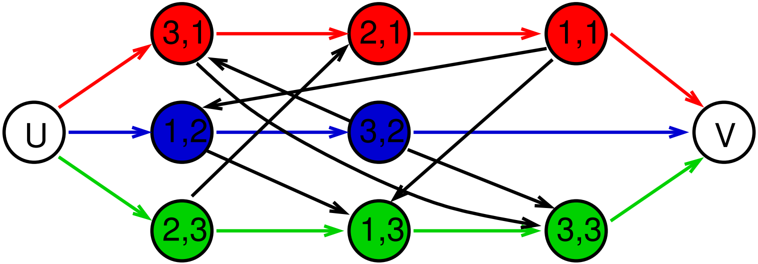
Pojmy:

- Podmnožina $D \subset B$ je nazývána výběr, jestliže obsahuje z každého páru disjunktivních hran právě jednu
- Výběr D je splnitelný, jestliže výsledný orientovaný graf $G(D) = (N, A \cup D)$ je acyklický
  - jedná se o graf s konjunktivními hranami a vybranými diskjunktními hranami

Poznámky:

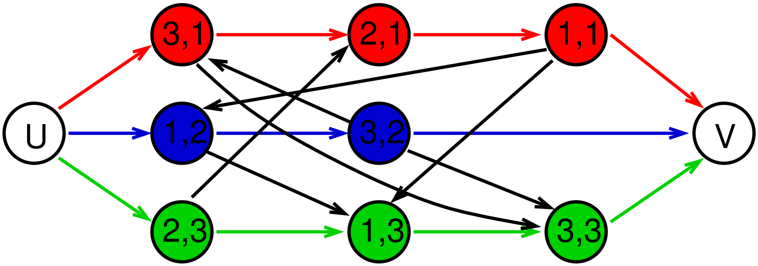
- splnitelný výběr určuje posloupnost, ve které jsou operace prováděny na strojích
- každý (konzistentní) rozvrh jednoznačně určuje splnitelný výběr
- každý splnitelný výběr jednoznačně určuje (konzistentní) rozvrh

Příklad: nesplnitelný výběr



V grafu existuje v důsledku nevhodného výběru hran cyklus:

Příklad: nesplnitelný výběr



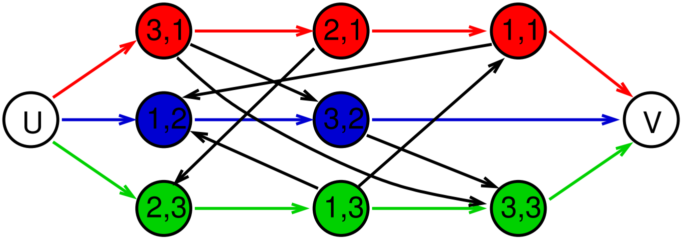
V grafu existuje v důsledku nevhodného výběru hran cyklus:

- $(1,2) \to (3,2)$
- $(3,2) \to (3,1) \to (2,1) \to (1,1) \to (1,2)$

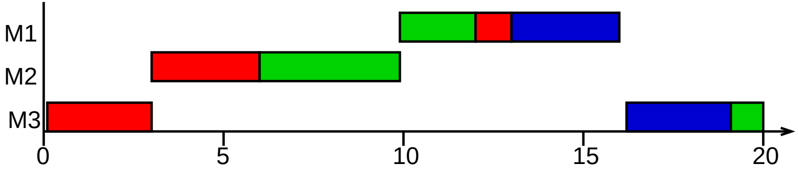
$\Rightarrow$ nelze splnit (k tomuto výběru neexistuje rozvrh)

Příklad: splnitelný výběr

Jakým způsobem nalézt rozvrh pro daný splnitelný výběr?



Tedy: jakým způsobem lze nalézt tento odpovídající rozvrh:



Výpočet rozvrhu pro výběr

Metoda: výpočet nejdelších cest z U do dalších uzlů v $G(D)$

Technický popis:

- uzly (i, j) mají ohodnocení p_{ij}, uzel U má ohodnocení 0
- délka cesty $i_1, i_2, \ldots, i_r$: součet ohodnocení uzlů $i_1, i_2, \ldots, i_{r-1}$
- spočítej délku l_{ij} nejdelší cesty z U do (i, j) a V, např. použitím Dijkstrova algoritmu

Výpočet rozvrhu pro výběr

Metoda: výpočet nejdelších cest z U do dalších uzlů v $G(D)$

Technický popis:

- uzly (i, j) mají ohodnocení p_{ij}, uzel U má ohodnocení 0
- délka cesty $i_1, i_2, \ldots, i_r$: součet ohodnocení uzlů $i_1, i_2, \ldots, i_{r-1}$
- spočítej délku l_{ij} nejdelší cesty z U do (i, j) a V,
  např. použitím Dijkstrova algoritmu

  1. pro všechny uzly (i, j) bez předchůdce: $l_{ij} = 0$
  2. vypočítej postupně pro všechny zbývající uzly (i, j) (a pro uzel V):

$$l_{ij} = \max_{\forall (k,l):(k,l)\rightarrow(i,j)} l_{ij} = l_{kl} + p_{ij}$$

Výpočet rozvrhu pro výběr

Metoda: výpočet nejdelších cest z U do dalších uzlů v $G(D)$

Technický popis:

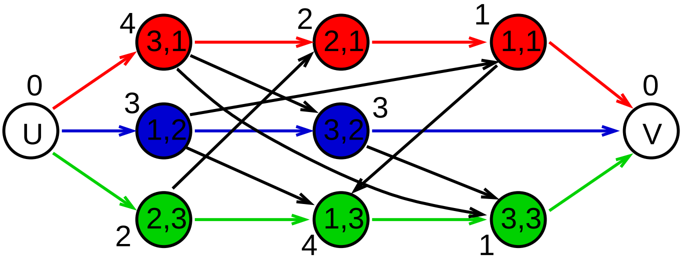
- uzly (i, j) mají ohodnocení p_{ij}, uzel U má ohodnocení 0
- délka cesty $i_1, i_2, \ldots, i_r$: součet ohodnocení uzlů $i_1, i_2, \ldots, i_{r-1}$
- spočítej délku l_{ij} nejdelší cesty z U do (i, j) a V, např. použitím Dijkstrova algoritmu

1. pro všechny uzly (i, j) bez předchůdce: $l_{ij} = 0$
2. vypočítej postupně pro všechny zbývající uzly (i, j) (a pro uzel V):

$$l_{ij} = \max_{\forall (k,l):(k,l)\to(i,j)} l_{ij} = l_{kl} + p_{ij}$$

- zahaj operaci (i, j) v čase l_{ij}
- délka nejdelší cesty z U do V je rovna *makespan*
    - tato cesta je kritická cesta

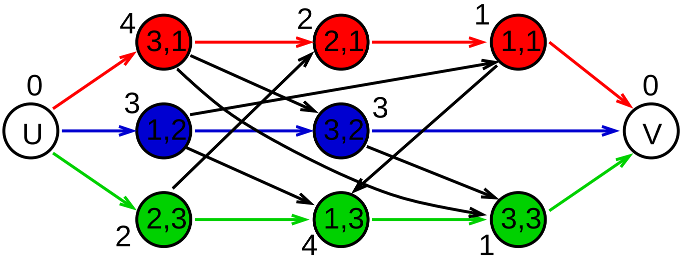
Příklad: výpočet rozvrhu pro výběr



Výpočet l_{ij}:

uzel	(3,1)	(1,2)	(2,3)	(2,1)	(3,2)	(1,1)	(1,3)	(3,3)	V
délka									

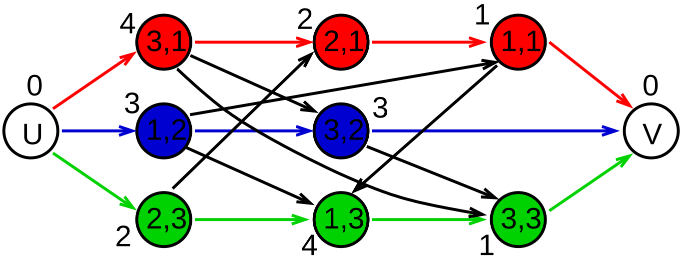
Příklad: výpočet rozvrhu pro výběr



Výpočet l_{ij}:

uzel	(3,1)	(1,2)	(2,3)	(2,1)	(3,2)	(1,1)	(1,3)	(3,3)	V
délka	0	0	0						

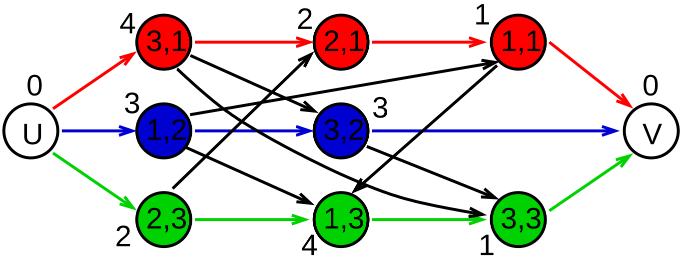
Příklad: výpočet rozvrhu pro výběr



Výpočet l_{ij}:

uzel	(3,1)	(1,2)	(2,3)	(2,1)	(3,2)	(1,1)	(1,3)	(3,3)	V
délka	0	0	0	4	4				

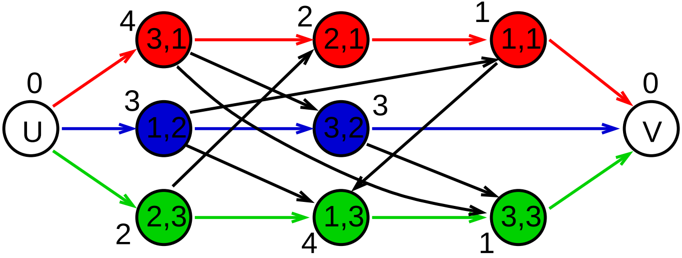
Příklad: výpočet rozvrhu pro výběr



Výpočet l_{ij}:

uzel	(3,1)	(1,2)	(2,3)	(2,1)	(3,2)	(1,1)	(1,3)	(3,3)	V
délka	0	0	0	4	4	6			

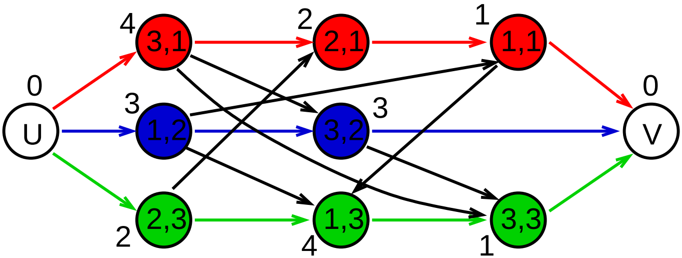
Příklad: výpočet rozvrhu pro výběr



Výpočet l_{ij}:

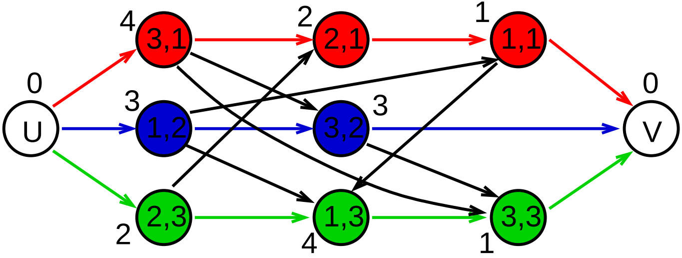
uzel	(3,1)	(1,2)	(2,3)	(2,1)	(3,2)	(1,1)	(1,3)	(3,3)	V
délka	0	0	0	4	4	6	7		

Příklad: výpočet rozvrhu pro výběr



Výpočet l_{ij}: uzel	(3,1)	(1,2)	(2,3)	(2,1)	(3,2)	(1,1)	(1,3)	(3,3)	V
délka	0	0	0	4	4	6	7	11	

Příklad: výpočet rozvrhu pro výběr

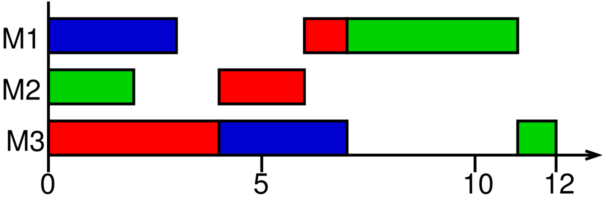


Výpočet l_{ij}:

uzel	(3,1)	(1,2)	(2,3)	(2,1)	(3,2)	(1,1)	(1,3)	(3,3)	V
délka	0	0	0	4	4	6	7	11	12

Konstrukce výběru pro daný rozvrh

Nalezněte výběr hran pro daný rozvrh:



Konstrukce odpovídajícího výběru: vybereme disjunktivní hrany, které odpovídají uspořádání operací úlohy v rozvrhu