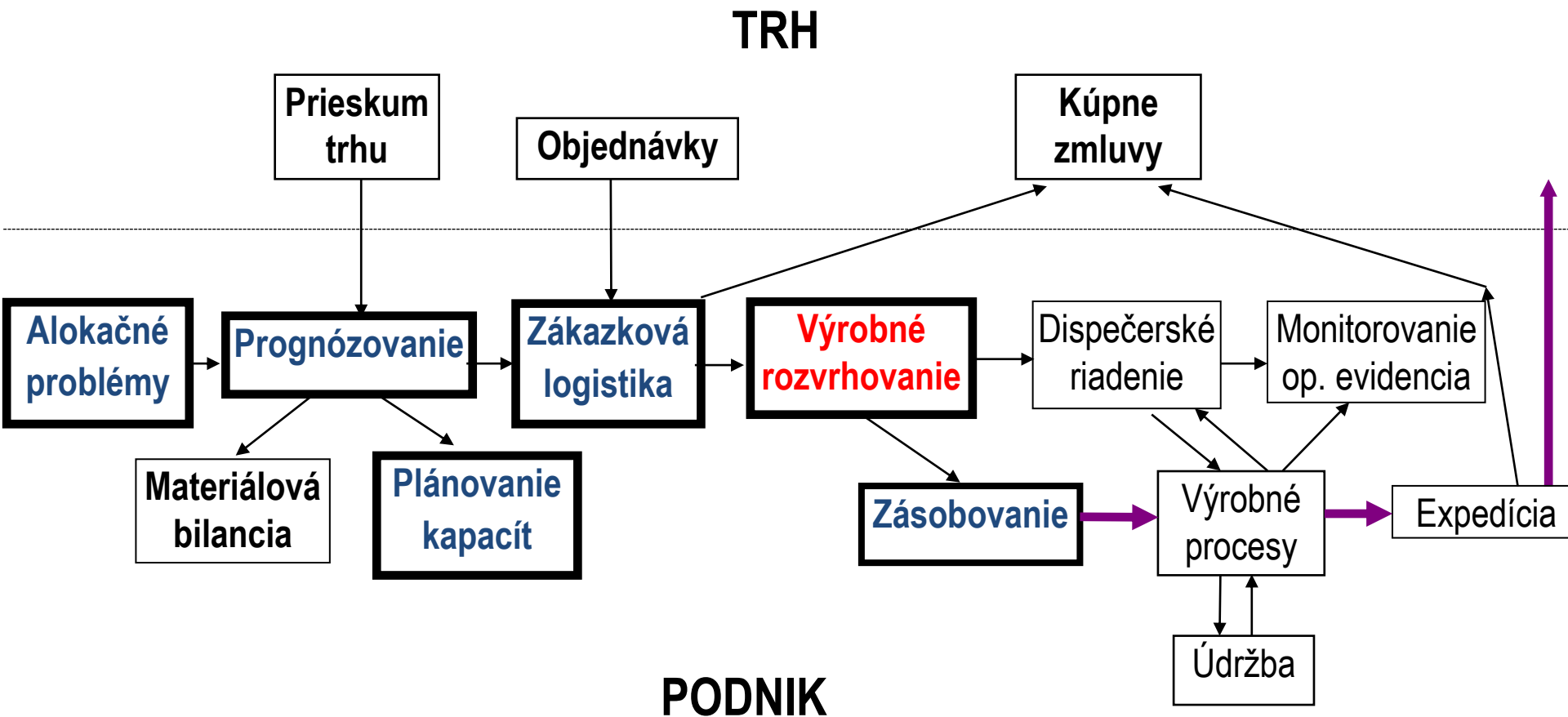


Rozvrhovanie – 3.časť

- Operatívny manažment výrobného procesu
 - Plánovanie vs. rozvrhovanie
- Rozvrhovanie – hlavné a doplnkové charakteristiky, typy úloh
 - Rozvrh, optimálny rozvrh, používané kritériálne funkcie
 - Príklad úlohy rozvrhovania a tvorby rozvrhu (Ganttov diagram)
- **Typy rozvrhovacích úloh a ich riešenie**
 - 1. Rozvrhovanie na paralelných strojoch/procesoroch**
 - A. Rozvrhovanie na jednom stroji/procesore
 - S povoleným prerušením úloh
 - Bez prerušenia úloh
 - B. Rozvrhovanie na viacerých strojoch/procesoroch
 - 2. Rozvrhovanie na dedikovaných (špecializovaných) strojoch**
 - A. Open shop (riešiť ako flow shop)
 - B. Flow shop
 - C. Job shop

Štruktúra činností výrobnjej logistiky



Typy rozvrhovacích úloh

1. Paralelné procesory (stroje)

- A. Rozvrhovanie **na jednom procesore** (jednostupňová výroba) -> úlohy s prerušením -> Jacksonov algoritmus
- B. Rozvrhovanie **na viacerých procesoroch** (viacstupňová výroba) -> heuristiky LPT, SPT, EST, LFT, MWR atď.

2. Dedikované (špecializované) procesory (stroje)

- úlohy sa rozdeľujú do skupín, tzv. zákaziek:

$$J_k = [T_{1,k} \dots, T_{nk,k}]$$

- každá úloha v rámci zákazky J_k beží na inom stroji
- Rozlišujeme **3 základné typy týchto úloh** –

A) open shop, B) flow shop a C) job shop

(podrobnosti sú uvedené na ďalšej strane)

2. Rozvrhovanie na dedikovaných (špecializovaných) procesoroch (strojoch)

2. Rozvrhovanie na dedikovaných procesoroch (strojoch)

Typ rozvrhovacej úlohy	Počet úloh v rámci zákaziek n_k	Poradie úloh v rámci zákaziek J_k
A) Open shop	Rovnaké pre všetky zákazky J_k	Ľubovoľné (napr. rozvrh hodín v škole)
B) Flow shop	Rovnaké pre všetky zákazky J_k	Pevne dané, rovnaké pre všetky zákazky J_k (napr. pásová výroba)
C) Job shop	Rôzne pre jednotlivé zákazky J_k	Pevne dané, rôzne pre jednotlivé zákazky J_k (napr. zákazková výroba)

A. Rozvrhovanie na viacerých dedikovaných procesoroch: **open shop**

A. Open shop

- Tieto úlohy môžeme považovať za istú analógiu úloh flow shop s tým ale, že **nezáleží na poradí úloh**.
- Možno riešiť tak, ako keby to bola úloha flow shop, určite je to platné riešenie open shop verzie úlohy.
- Otázkou ostáva, či neexistuje lepšie riešenie.
- Podobne ako u flow shop riešenie v polynomiálnom čase je známe len pre prípad dvoch procesorov.

B. Rozvrhovanie na viacerých dedikovaných procesoroch: **flow shop**

B. Flow shop

- Vo všeobecnosti ide o kombinatorickú optimalizáciu, existuje len niekoľko málo prípadov riešiteľných v polynomiálnom čase.
- Jedným z takýchto špeciálnych prípadov je **flow shop na dvoch procesoroch** (s ľubovoľným počtom zákaziek J_j , všetky úlohy sú k dispozícii v čase $r_{ij} = 0$)
- Riešenie: **Johnsonov algoritmus**.

B. Rozvrhovanie na viacerých dedikovaných procesoroch: **flow shop**

Johnsonov algoritmus:

1. Z množiny všetkých zákaziek J vytvoríme dva zoznamy:

$$L_1 = \{J_i \mid t_{1i} \leq t_{2i}\} \quad \text{a} \quad L_2 = J - L_1$$

2. Zoznam L_1 usporiadame podľa neklesajúcich procesných časov t_{1i} (t.j. od najkratšej operácie t_{1i} po najdlhšiu) a zoznam L_2 podľa nerastúcich časov t_{2i} (t.j. od najdlhšej operácie t_{2i} po najkratšiu)
3. Optimálny rozvrh v zmysle kritéria C_{max} je tvorený zreťazením usporiadaných zoznamov L_1 a L_2

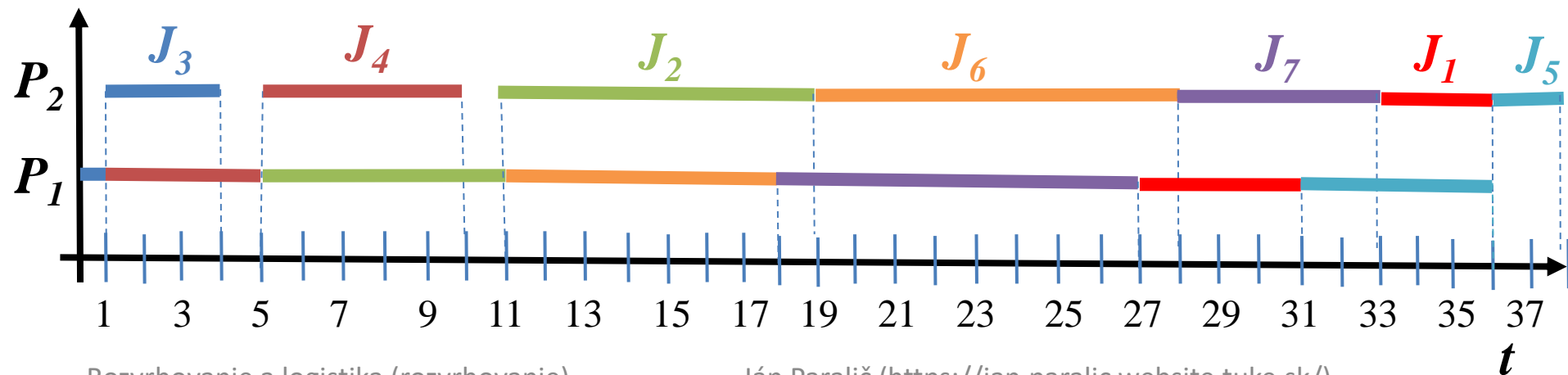
Príklad – flow shop na 2 procesoroch

- Majme 2 procesory a 7 zákaziek J_i ($i = 1 \dots 7$) s danými procesnými časmi:

J_i	1	2	3	4	5	6	7
P_{1i}	4	6	1	4	5	7	9
P_{2i}	3	8	3	5	2	9	5

- Johnsonov algoritmus:

- $L_1 = \{J_2, J_3, J_4, J_6\} \Rightarrow L_2 = \{J_1, J_5, J_7\}$
- Usporiadané: $L_1 = \langle J_3, J_4, J_2, J_6 \rangle$, $L_2 = \langle J_7, J_1, J_5 \rangle$
- Takže výsledný optimálny rozvrh podľa C_{max} zodpovedá zreťazeniu L_1 a L_2
 $R = \langle J_3, J_4, J_2, J_6, J_7, J_1, J_5 \rangle$

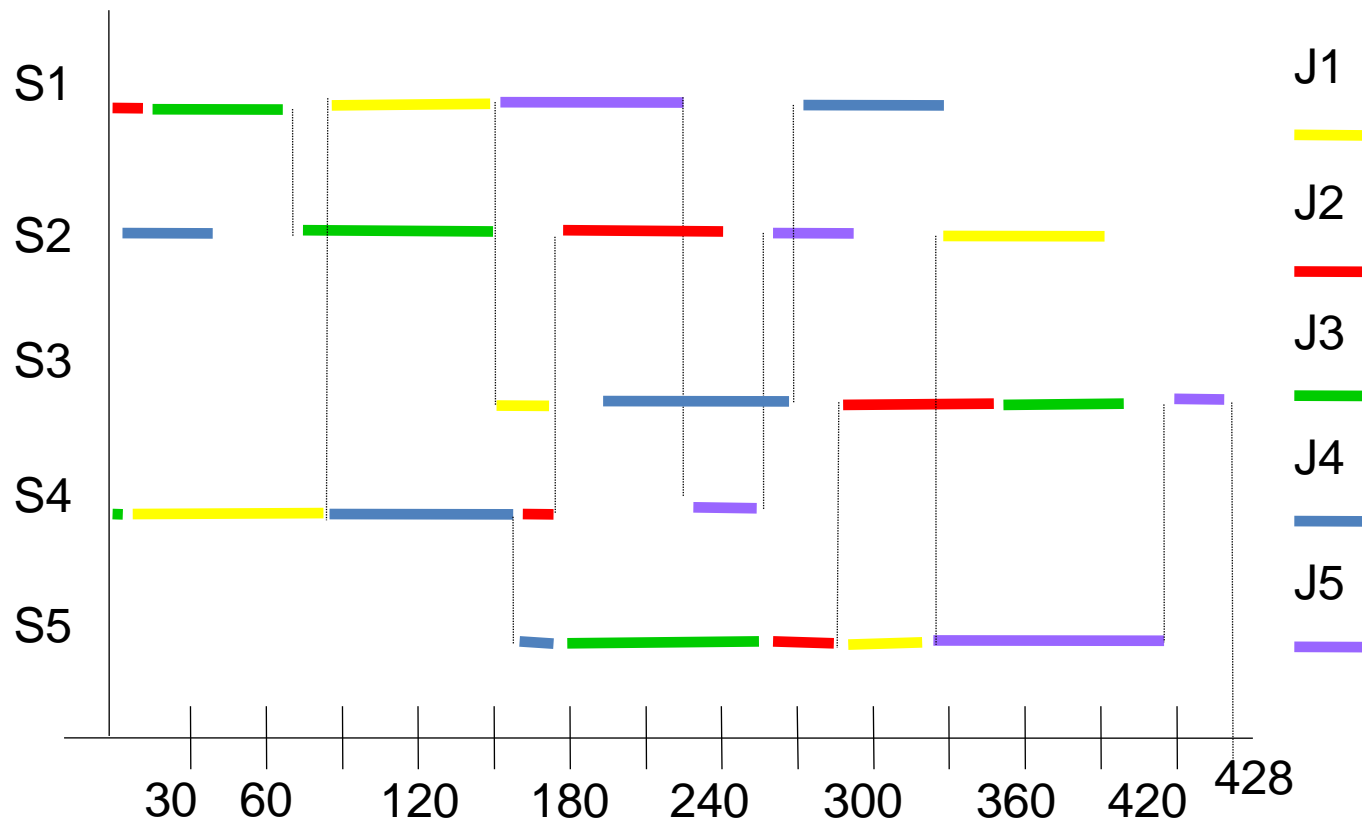


C. Rozvrhovanie na viacerých dedikovaných procesoroch: **job shop**

C. Job shop – definícia úlohy:

- m strojov, n výrobkov,
- výroba každého výrobku je členená na maximálne m operácií, pričom každá je vykonávaná na inom stroji,
- trvanie operácií je presne dané,
- operácie sú bez prerušenia,
- každý stroj môže v jednom okamihu spracovávať nanajvýš jednu operáciu,
- sú presne stanovené poradia operácií pre jednotlivé výrobky (pritom pre každý výrobok to môže byť iné)
- Ide o najzložitejší typ úloh, špeciálnym prípadom je iba prípad 2 zákaziek spracovávaných na ľubovoľnom počte procesorov.

Príklad rozvrhu (Ganttov diagram) pre úlohu job-shop rozmeru 5 x 5



Job shop – špeciálny prípad

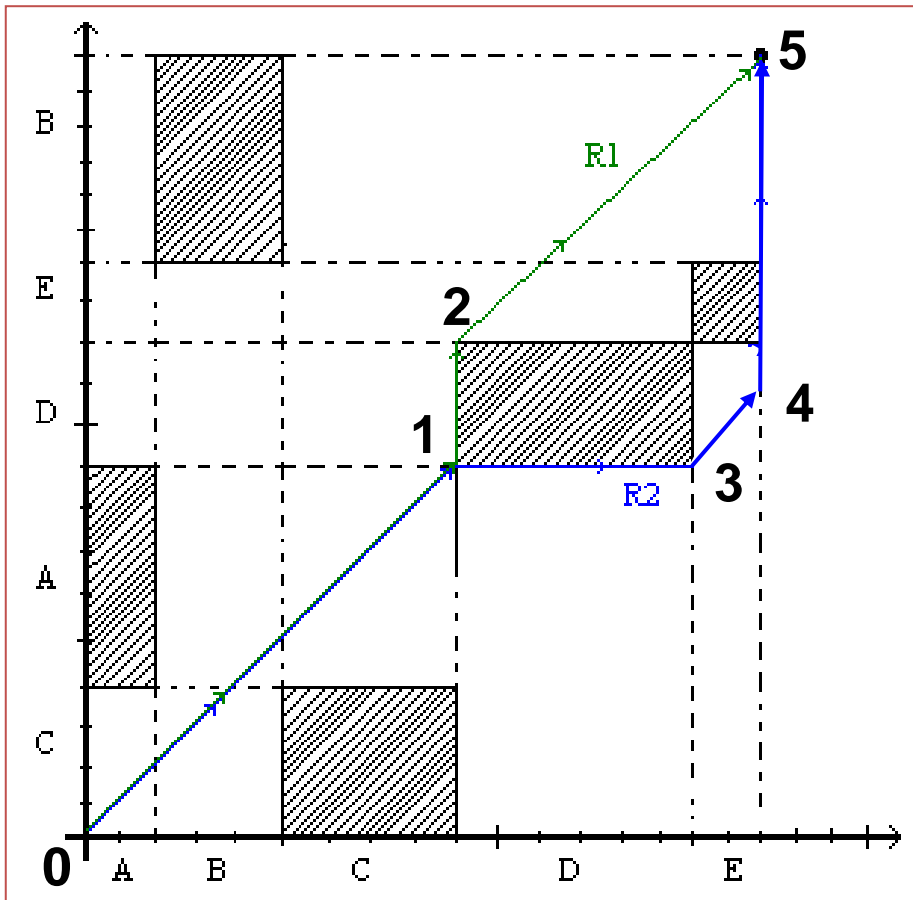
2 zákazky, ľubovoľný počet strojov

POSTUP RIEŠENIA:

1. Na vodorovnú os vynesieme časy spracovania úloh v rámci prvej zákazky v predpísanom poradí podľa jednotlivých procesorov.
2. Na zvislú os vynesieme časy spracovania úloh v rámci druhej zákazky v predpísanom poradí podľa jednotlivých procesorov.
3. Vyšrafujeme tzv. „zakázané oblasti“, t.j. oblasti v ktorých súradnice bodov na vodorovnej aj zvislej osi patria tomu istému procesoru.
4. Rozvrh znázorňujeme tzv. „pracovnou čiarou“ - ide o lomenú čiaru pozostávajúcu z troch typov úsekov:
 - Úseky pod 45° uhlom zodpovedajú paralelnému spracovávaniu oboch úloh.
 - Vodorovný úsek zodpovedá spracovávaniu prvej zákazky, zatiaľ čo druhá čaká na uvoľnenie procesora.
 - Zvislý úsek znamená spracovávanie druhej zákazky, zatiaľ čo prvá čaká na uvoľnenie procesora.
5. Najkratšia pracovná čiara potom zodpovedá optimálnemu rozvrhu.

Príklad:

J_1	Poradie	A	B	C	D	E	Σ
	Trvanie	2	3	4	6	2	17
J_2	Poradie	C	A	D	E	B	Σ
	Trvanie	4	5	3	2	6	20



$R_1: 0 \rightarrow 1 \rightarrow 2 \rightarrow 5$

$R_2: 0 \rightarrow 1 \rightarrow 3 \rightarrow 4 \rightarrow 5$

Dĺžka:

$R_1: 9 + 3 + 8 = 20$

$R_2: 9 + 6 + 2 + 1 + 2 + 6 = 26$

- Optimálny rozvrh je R_1
- Nakreslite Ganttov diagram pre R_1

Ďalšie metódy rozvrhovania (1)

1. **Úplné metódy**, ktoré zaručujú nájdenie (optimálneho) riešenia ak existuje
 - metóda vetvenia a medzí
 - úlohy s ohraničeniami
2. **Neúplné metódy**, ktoré neprehľadávajú celý priestor prehľadávania, iba jeho časť s tým že zaručujú iba nájdenie suboptimálneho riešenia (podrobne o týchto metódach pojednáva predmet Heuristické optimalizačné procesy)
 - genetické algoritmy
 - hill climbing
 - simulované žíhanie
 - tabu search

Rozvrhovanie ako úloha s ohraničeniami

- Ako **premenné** možno zvoliť časy kedy sa začne spracovávať daný výrobok na danom pracovisku (začiatok vykonávania i -tej operácie označme T_i).
- **Ohraničenia** sú potom reprezentované ako nerovnice, napr.
 - $T_1 \geq 7$ (ak je dané r_i – t.j. vo všeobecnosti $T_i \geq r_i$)
 - $T_3 + 4 \leq 10$ (ak je dané d_i – t.j. $T_i + t_i \leq d_i$)
 - $T_1 + 2 \leq T_2$ (precedencia, t.j. $T_i + t_i \leq T_j$)
 - $\forall k \neq l$ zdieľajúce jeden stroj:
 $T_k + t_k \leq T_l$ alebo $T_l + t_l \leq T_k$

Ďalšie metódy rozvrhovania (2)

1. Výpočtový čas versus optimálnosť riešenia

- Pre nájsenie zaručene optimálneho riešenia je nutné použiť niektorú z úplných metód. Avšak tie narážajú na kombinatorickú explóziu. Čas výpočtu totiž exponenciálne narastá s veľkosťou úlohy (hlavným zdrojom zložitosti je počet disjunktných ohraničení).

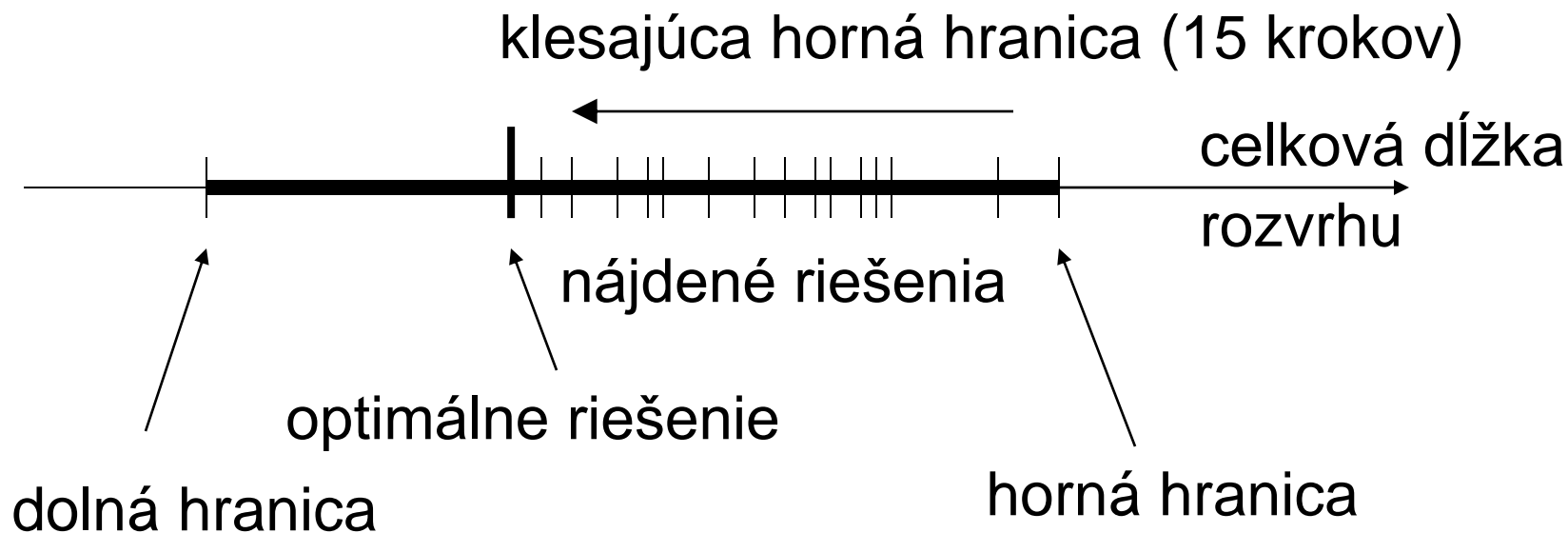
2. Špecifikácia problému - každá z metód si vyžaduje svoju reprezentáciu úlohy.

- Lineárne programovanie narába s množinou lineárnych nerovníc a kriteriálnou funkciou (nie je schopné zachytiť disjunktné ohraničenia, to umožňujú iba CSP – úlohy s ohraničeniami).
- Metódu vetvenia a medzí možno použiť na každý optimalizačný problém, ak je k dispozícii spôsob, ako ohodnotiť kvalitu čiastočného riešenia.
- Hill climbing, simulované žíhanie a tabu search záleží od funkcií susednosti, ktoré systém používa pre nájsenie nového riešenia.
- Efektívnosť genetických algoritmov závisí na tom, ako sú reprezentovaní kandidáti na riešenie a ako je definovaná vyhodnocovacia funkcia.

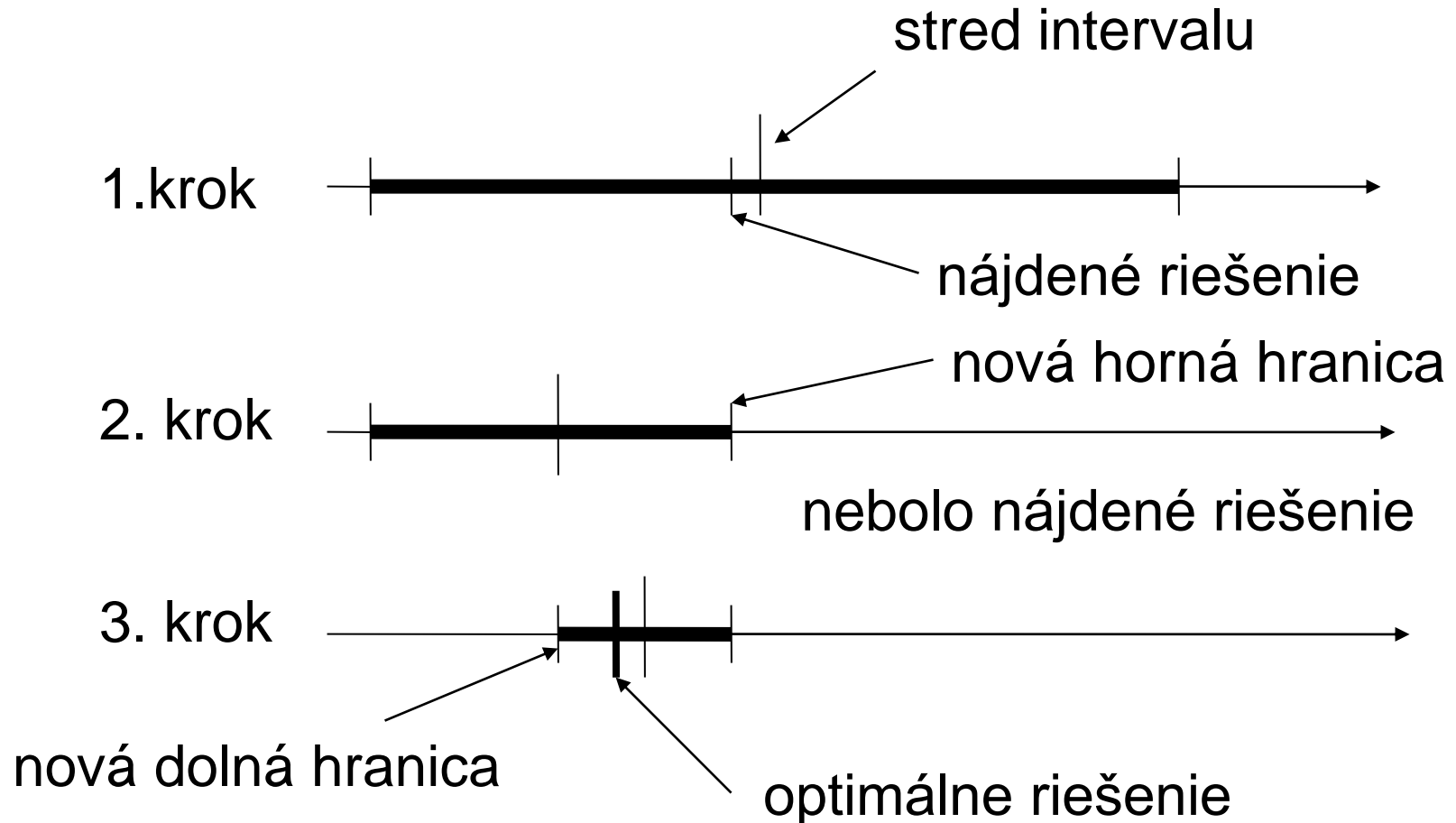
Niektoré naše výsledky pri riešení rozvrhovacích úloh

Optimalizačný algoritmus **min_max**

(zodpovedá klasickej metóde vetvenia a medzí)



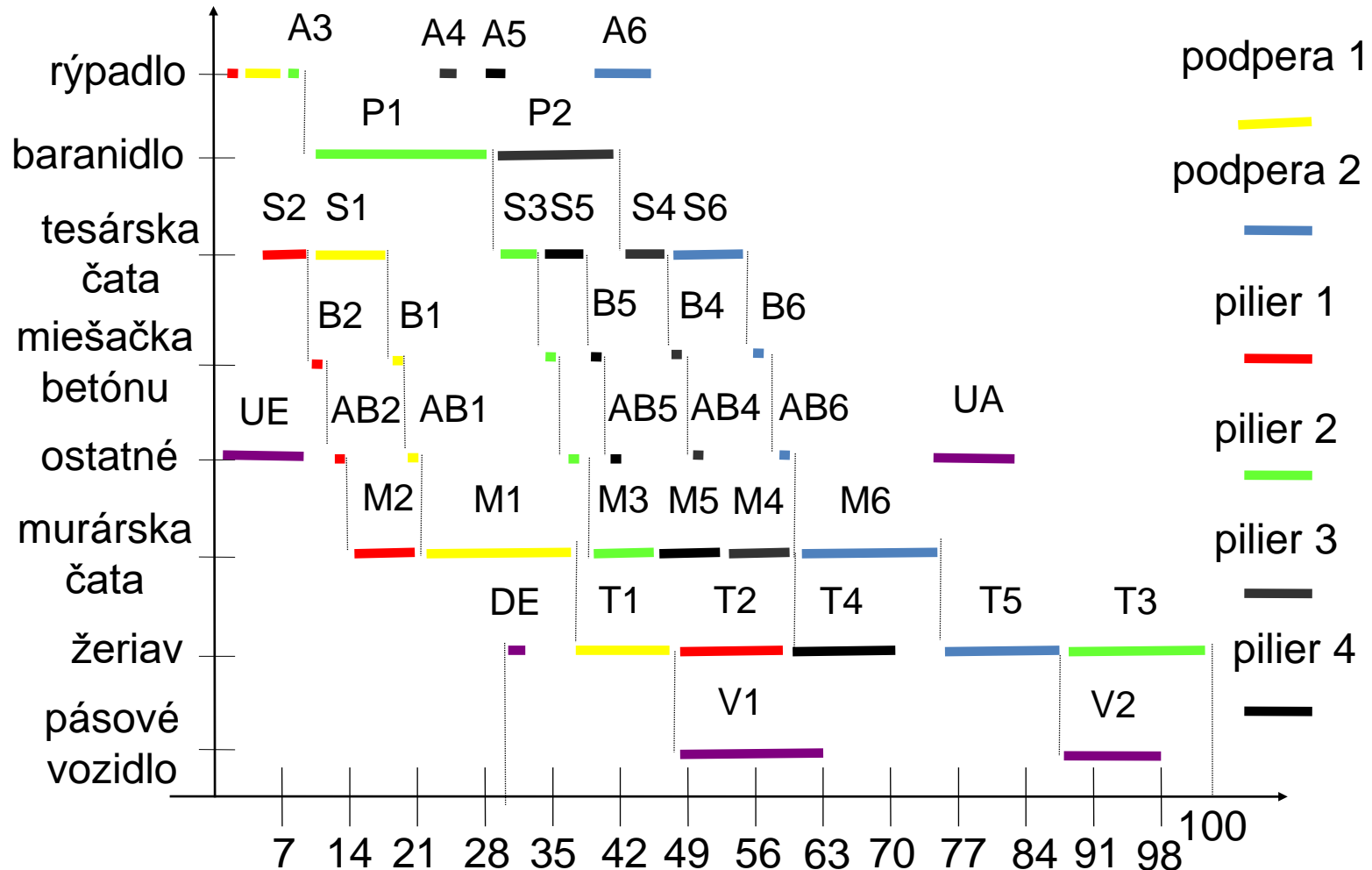
Nový optimalizačný algoritmus **log_min_max**



Výsledky testov

úloha	MIN-MAX		LOG-MIN-MAX		LOG-MINIMIZE	
	čas (s)	návraty	čas (s)	návraty	čas (s)	návraty
8-8-0	30.75	69	12.74	96	10.25	746
8-8-1	40.63	384	5.55	95	5.89	319
8-8-2	98.50	1331	48.71	774	32.65	1238
8-8-3	70.34	2019	7.90	79	9.99	744
8-8-4	68.70	3654	5.95	56	5.47	676
8-8-5	114.01	6534	24.26	467	21.51	873
8-8-6	31.13	50	3.80	6	3.41	452
8-8-7	42.06	356	9.55	104	11.01	755
8-8-8	29.75	73	12.21	173	12.50	604
8-8-9	109.03	1813	34.84	667	30.94	1073
10-10-0	808.42	13517	129.45	1245	108.28	2130
10-10-1	547.48	10287	51.69	288	32.65	883
10-10-2	1064.87	20233	499.71	8446	351.28	8037
10-10-3	1010.60	19271	324.59	3965	289.85	4851
10-10-4	2307.45	111701	128.99	1907	156.26	4506
10-10-5	382.99	8450	73.73	830	56.30	3286
10-10-6	126.64	1425	19.92	148	18.68	1043
10-10-7	10647.00	116817	10239.5	151256	8404.28	15082
10-10-8	590.05	27063	99.97	1846	89.90	2577
10-10-9	6256.24	111350	2565.99	38861	1228.24	29234
Most	28821.2	109805	29103.2	142772	32730.9	14277

Optimálny rozvrh výstavby mostu



Úloha zo 7. prednášky

1. Definujte si vlastnú úlohu rozvrhovania typu job shop pre 2 zákazky J_1 a J_2 na štyroch procesoroch. Zadaťte si pre obe zákazky:
 - časy trvania jednotlivých úloh v zákazke t_{ij} (dĺžka úlohy i v zákazke j)
 - predpísané poradie úloh v rámci jednotlivých zákaziek tak, aby bolo rôzne pre každú zo zákaziek J_1 a J_2
 - úloha musí byť zadaná tak, aby pracovná čiara narazila na aspoň jednu zakázanú oblasť.
2. Postupom vysvetleným na prednáške nájdite optimálny rozvrh pre Vami zadanú úlohu z hľadiska kritéria C_{max} .
3. Zostrojte pre Vami zadanú úlohu rozvrh vo forme Ganttovho diagramu.
4. Pre zostavený rozvrh vypočítajte hodnoty nasledovných kritériálnych funkcií: C_{max} , C , a \bar{C} .