



**KONTAKT 2011**



# ***Algoritmy pro rozvrhování směn***

***Autor: Bc. Roman Václavík (vaclarom@fel.cvut.cz)  
Vedoucí: Ing. Zdeněk Baumelt (baumezde@fel.cvut.cz)***

# Algoritmy pro rozvrhování směn

## Cíle práce

- Studie problematiky rozvrhování směn.
- Výběr a implementace algoritmů pro NRP.
- Implementace vlastního algoritmu.
- Porovnání algoritmů.

## Rozvrhování směn

- NP-těžký problém.
- Přiřazení zdrojů a času k aktivitám.
- Ohodnocovací funkce
  - Tvrdá omezení.
  - Měkká omezení.

1993 Listopad	1							2						
	01	02	03	04	05	06	07	08	09	10	11	12	13	14
	Po	Ut	St	Ct	Pa	So	Ne	Po	Ut	St	Ct	Pa	So	Ne
Josef Novak				TD	TD				TD	TD		TN	TN	TN
Marie Nova	TN	TN			TD	TN	TN			TD	TD			
Oldrich Stary	TD	TD	TN			TD	TD			TN	TN			
Karel Dobrota	TD	TD	TD			TN	TN			TN	TN			

# Algoritmy pro rozvrhování směn

## Implementované algoritmy

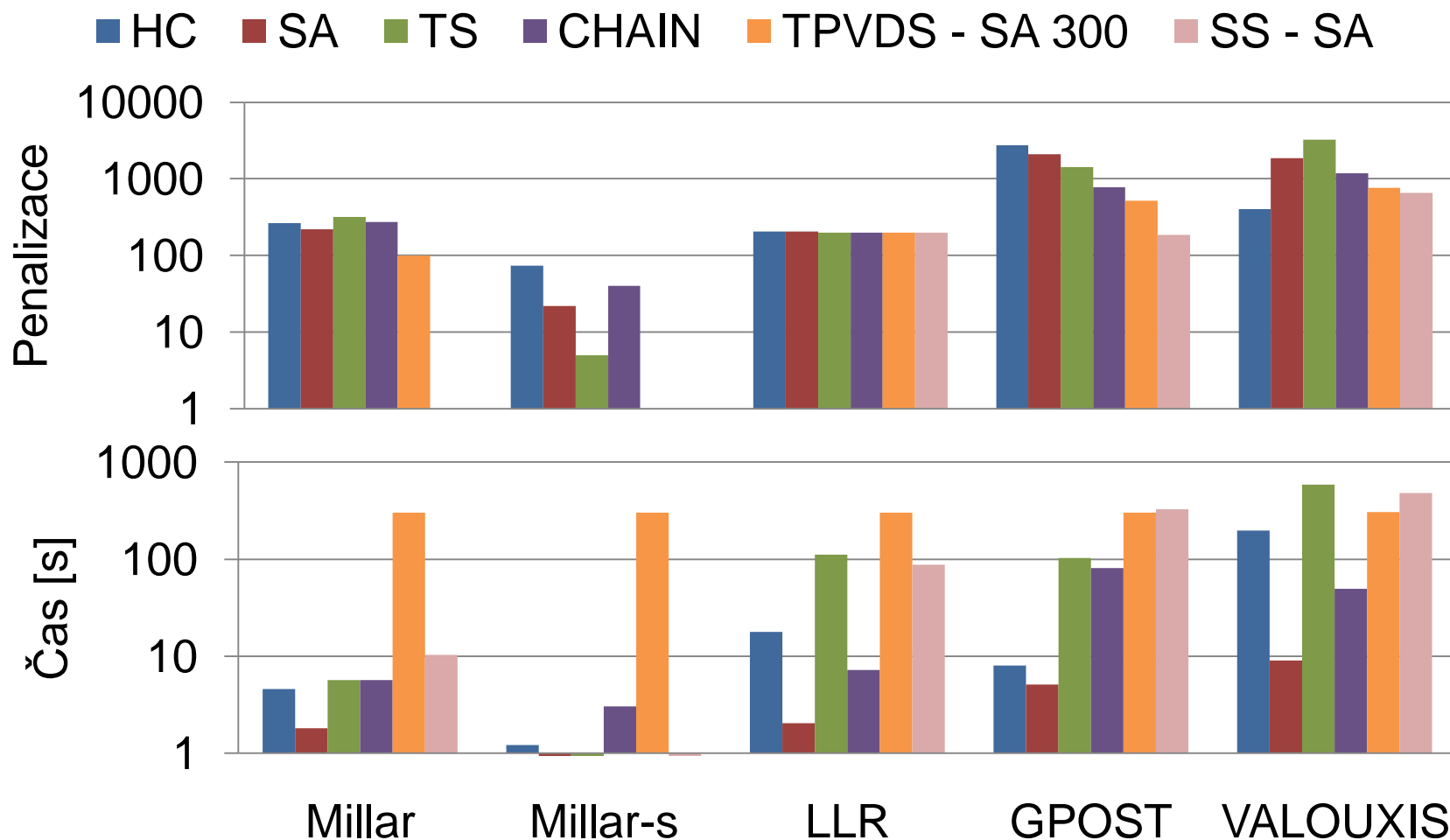
- Hill Climbing
- Simulated Annealing
- Tabu Search
- Time Pre-defined  
Variable Depth Search
- Scatter Search
- **Chain Algorithm**

	1							2							
1993 Listopad	01	02	03	04	05	06	07	08	09	10	11	12	13	14	
	Po	Ut	St	Ct	Pa	So	Ne	Po	Ut	St	Ct	Pa	So	Ne	
Josef Novak				TD	TD				TD	TD			TN	TN	TN
Marie Nova	TN	TN			TD	TN	TN			TD	TD				
Oldrich Stary	TD	TD	TN			TD	TD			TN	TN				
Karel Dobrota	TD	TD				TN	TN			TN	TN				

↓

	1							2						
1993 Listopad	01	02	03	04	05	06	07	08	09	10	11	12	13	14
	Po	Ut	St	Ct	Pa	So	Ne	Po	Ut	St	Ct	Pa	So	Ne
Josef Novak				TD	TD	TD	TD			TD				
Marie Nova	TN	TN	TN		TD	TN	TN		TD	TD	TD			
Oldrich Stary	TD	TD	TD							TN	TN			
Karel Dobrota	TD	TD				TN	TN			TN	TN	TN	TN	TN

# Algoritmy pro rozvrhování směn



Penalizace	0*	0*	204 <sup>+</sup>	25 <sup>+</sup>	20*
------------	----	----	------------------	-----------------	-----

\* Optimální řešení, + Řešení generované programem Roster Booster

# Algoritmy pro rozvrhování směn



Autor: Bc. Roman Václavík (vaclarom@fel.cvut.cz)

Vedoucí: Ing. Zdeněk Baumelt (baumezde@fel.cvut.cz)



## Rozvrhování směn

**Rozvrhování** je disciplína, ve které se snažíme přiřadit zdroje a čas k aktivitám takovým způsobem, abychom dosáhli požadovaných cílů a zároveň došlo ke splnění všech nutných podmínek kladených na výsledný **rozvrh**.

	1							2						
1993 Listopad	01	02	03	04	05	06	07	08	09	10	11	12	13	14
	Po	Ut	St	Ct	Pa	So	Ne	Po	Ut	St	Ct	Pa	So	Ne
Josef Novak														
Marie Nova														
Oldřich Stary														
Karel Dobrota														

## Omezení kladená na rozvrh

- **Tvrdá omezení** – nesmí být nikdy porušena, přičemž nemají žádný vliv na hodnotu kritériální funkce.
- **Měkká omezení** – mohou být porušována, ale snahou ovšem je docílit co nejmenšího počtu porušení s tím, že je nutné brát v úvahu i cenu (váhu) porušení.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Delete th	Pattern 0	0	6 min	3 linear	100	-1						
2		volno	volno										
3		N	N										
4													
5													
6													
7		Add new											

## Kritériální funkce

- Zavedení tzv. ohodnocení rozvrhu.
- Pro výpočet této hodnoty slouží **kritériální funkce P**.
- Rozvrhování potom řeší optimalizační úlohu.

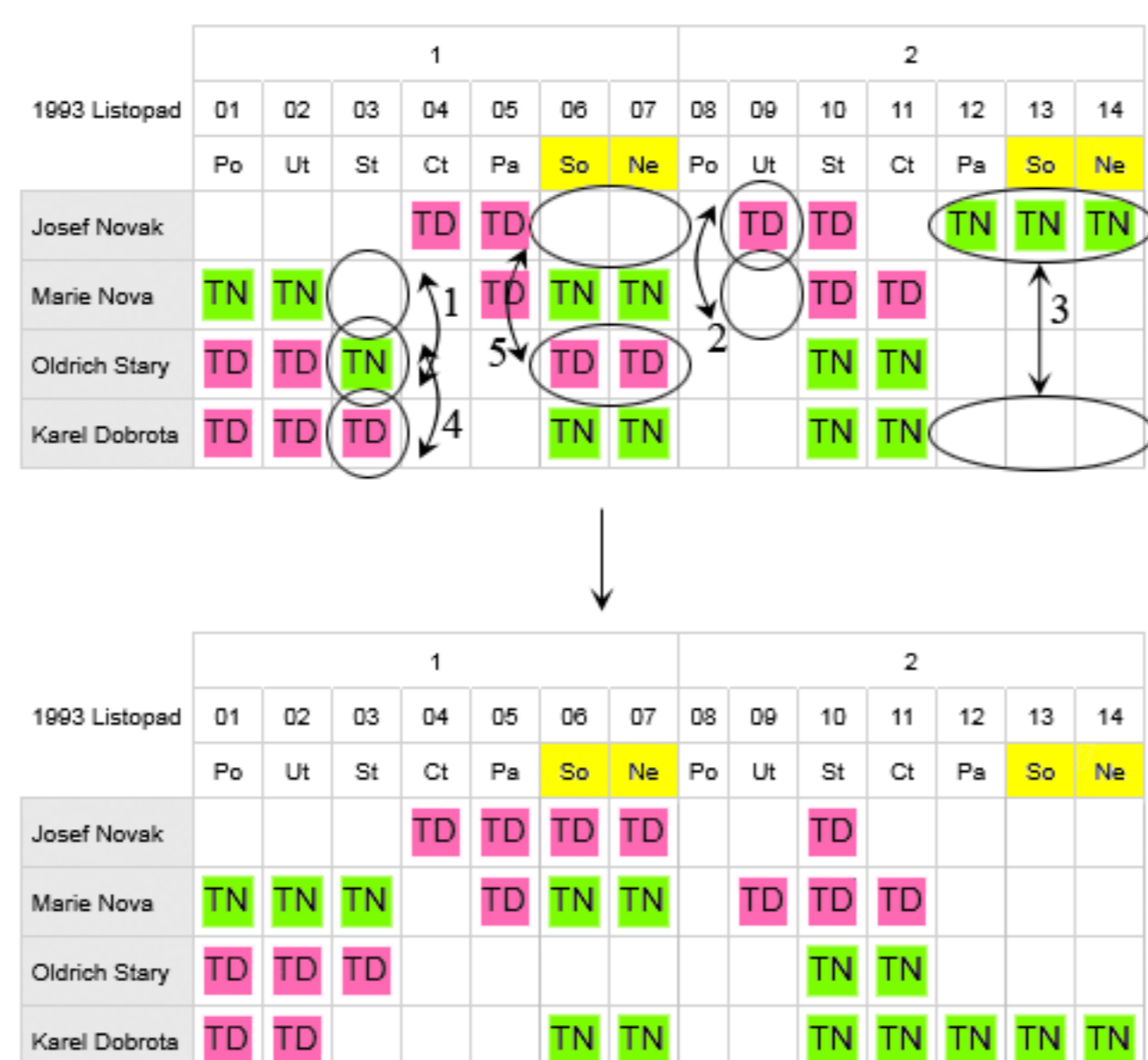
$$\min P$$

$$P = \sum_{c \in SC} w_c \cdot p_{cI} \quad w_c \geq 0$$

SC je množina všech měkkých omezení,  $w_c$  je váha měkkého omezení  $c$  a  $p_{cI}$  je počet porušení měkkého omezení  $c$ .

## Implementované algoritmy

- Všeobecně známé
  - Hill Climbing (HC)
  - Simulated Annealing (SA)
  - Tabu Search (TS)
- Inspirované literaturou
  - A Time Pre-defined Variable Depth Search (TPVDS)
  - Scatter Search (SS)
- Vlastní
  - Chain Algorithm



## Porovnání algoritmů

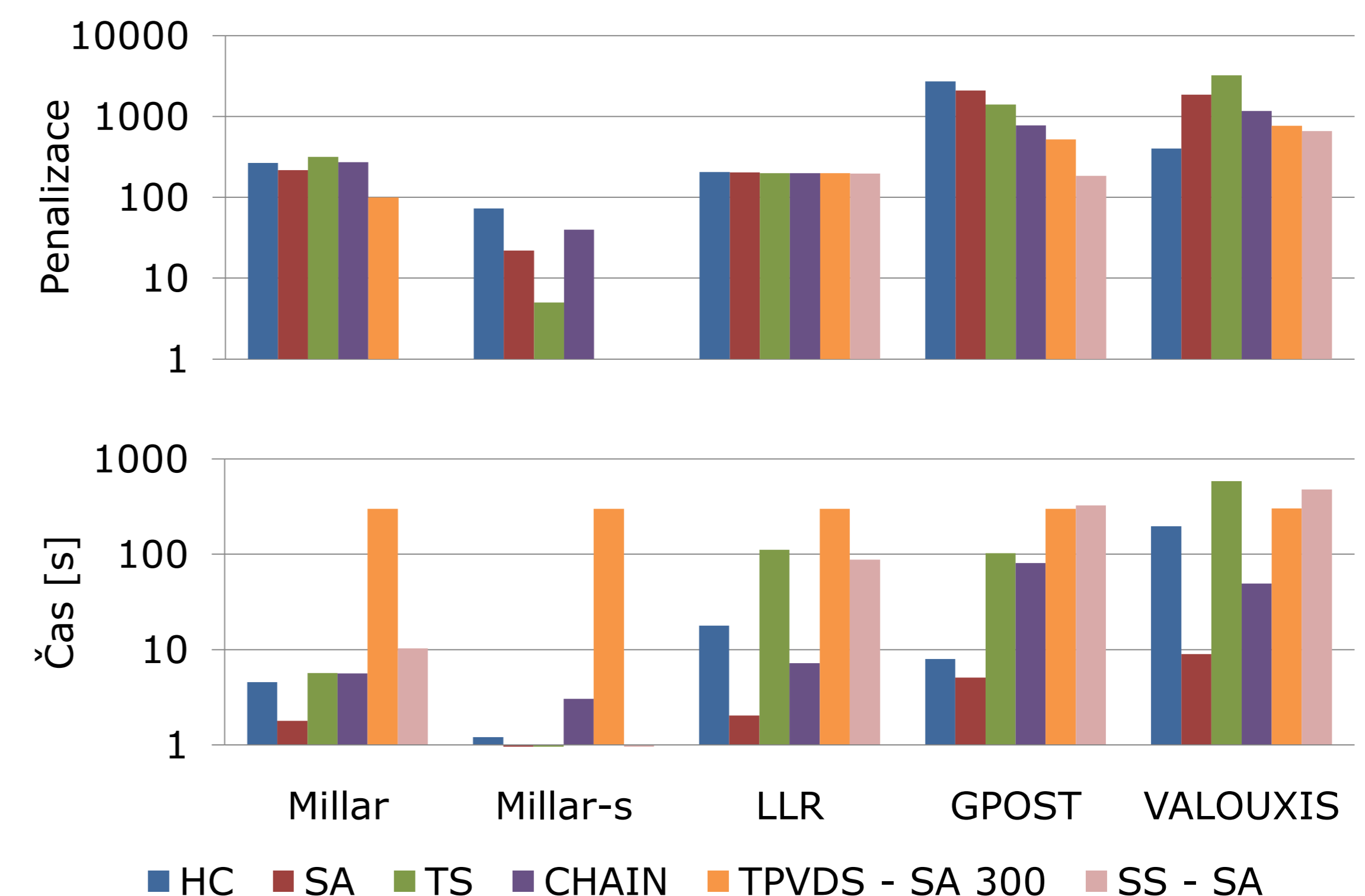
Instance	dnů	zaměstnanců	Počet	
			směn	měkkých/tvrдых omezení
Millar	14	8	2	15/1
Millar-s	14	8	2	10/1
LLR	7	27	3	8/1
GPOST	28	8	2	33/1
VALOUXIS	28	16	3	18/1

Instance	Nejlepší známé řešení	Nejlepší dosažené řešení ^
Millar	0*	0/0/0/0/0/0
Millar-s	0*	0/0/0/0/0/0
LLR	204+	201/199/198/198/198/190
GPOST	25+	230/227/441/231/48/28
VALOUXIS	20*	260/620/640/540/420/160

\* Optimální řešení

+ Řešení generované programem Roster Booster

^ Hodnoty pro algoritmy HC/SA/TS/CHAIN/TPVDS/SS



## Zhodnocení

- **Hill Climbing** spolu se **Simulated Annealing** vykazují nejlepší časy s místy průměrnou až nejhorší dosaženou penalizací.
- **Tabu Search** dosahuje penalizací, které jsou průměrné, ale čas se řadí k těm horším. Velkou výhodou je malý rozptyl mezi hodnotami nejlepší a nejhorší nalezené penalizace.
- **Chain Algorithm** se pyšní nejlepším poměrem výkon/čas.
- **TPVDS** - časový limit je vhodné ponechat na 300 sekundách a jako zlepšovací algoritmus použít SA.
- **Scatter Search** lze bezpochyby považovat za nejlepší testovaný algoritmus. Nevýhodou je vyšší časová náročnost.