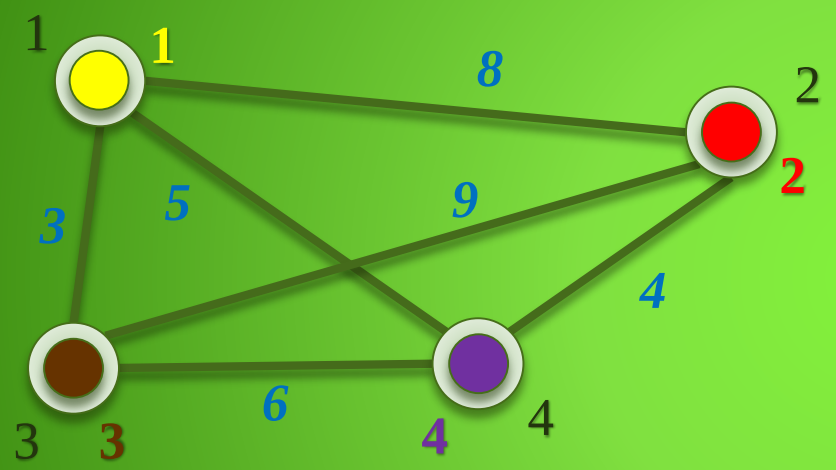
Quadratic Assignment Problem - QAP

1957 - Koopmans, Beckman (lokačný problém)

- priradiť ***n* – zariadení**; na ***n* – lokalít**



Vzdialenosti:

$$B = \begin{pmatrix} 0 & 8 & 3 & 5 \\ 8 & 0 & 9 & 4 \\ 3 & 9 & 0 & 6 \\ 5 & 4 & 6 & 0 \end{pmatrix}$$

Tok:

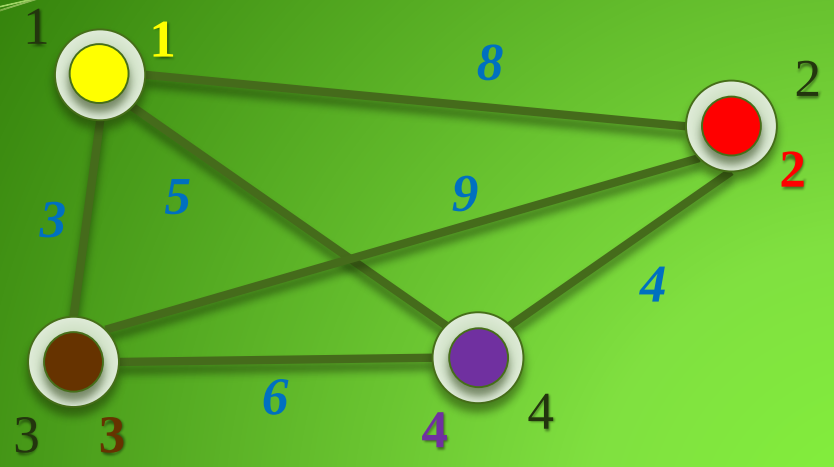
$$A = \begin{pmatrix} 0 & 9 & 3 & 5 \\ 9 & 0 & 2 & 7 \\ 3 & 2 & 0 & 4 \\ 5 & 7 & 4 & 0 \end{pmatrix}$$

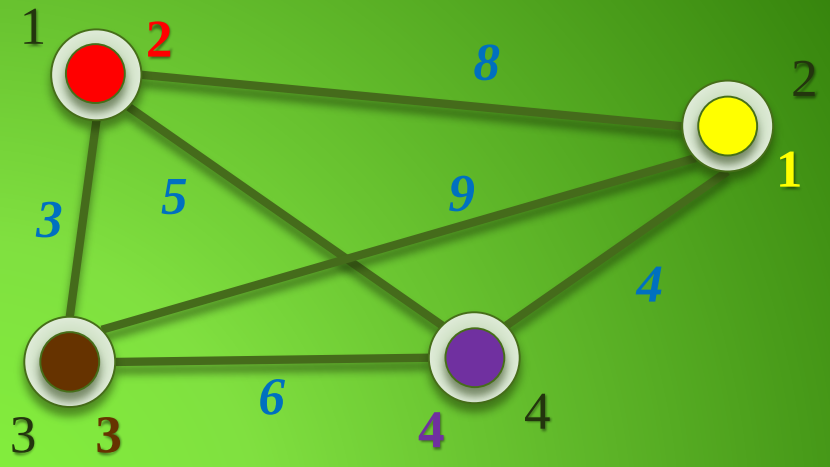
Tok (1, 2) = 9
Tok (1, 3) = 3
Tok (1, 4) = 5
Tok (2, 3) = 2
Tok (2, 4) = 7
Tok (3, 4) = 4

$$\Pi = \{1, 2, 3, 4\}$$

Celkové náklady: $C = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{ij}$

Quadratic Assignment Problem - QAP





$$\Pi = \{1, 2, 3, 4\}$$

$$\Pi = \{2, 1, 3, 4\}$$

Tok:

$$A = \begin{pmatrix} 0 & 9 & 3 & 5 \\ 9 & 0 & 2 & 7 \\ 3 & 2 & 0 & 4 \\ 5 & 7 & 4 & 0 \end{pmatrix}$$

Tok:

$$A = \begin{pmatrix} 0 & 9 & 2 & 7 \\ 9 & 0 & 3 & 5 \\ 2 & 3 & 0 & 4 \\ 7 & 5 & 4 & 0 \end{pmatrix}$$

Quadratic Assignment Problem - QAP

1957 - Koopmans, Beckman (lokačný problém)

- priradiť ***n* – zariadení**
- na ***n* – lokalít**

- $\boldsymbol{a_{ij}}$ - tok medzi zariadením $\boldsymbol{i}$ a $\boldsymbol{j}$ $\Rightarrow$ $(\boldsymbol{a_{ij}}) = \boldsymbol{A}$
- $\boldsymbol{b_{ij}}$ - vzdialenosť medzi zariadeniami $\boldsymbol{i}$ a $\boldsymbol{j}$ $\Rightarrow$ $(\boldsymbol{b_{ij}}) = \boldsymbol{B}$
- náklady - $C = \sum_{i,j} a_{ij} b_{ij}$

$$\min_{\pi \in S_n} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{\pi_i, \pi_j} b_{ij}$$

$\boldsymbol{S_n}$ - množina permutácií $\{1,2, \dots n\}$

$$\boldsymbol{\Pi_{min}} = ?$$

NP – ťažký problém

Rozmiestnenie budov vo vojenskej základni

- *n* budov
- vzdialenosť medzi budovami *i* a *j* - $\boldsymbol{b_{ij}}$
- hustota premávky medzi budovami *i* a *j* - $\boldsymbol{a_{ij}}$
- minimalizovať celkovú prejdenú vzdialenosť za určité obdobie

(podobne napríklad rozmiestnenie ambulancií v nemocnici)

Rozloženie kláves na klávesnici

- *n* kláves (symbolov)
- čas potrebný na stlačenie klávesy na pozícii *i* po stlačení klávesy na pozícii *j* - $\boldsymbol{b_{ij}}$
- frekvencia výskytu párov symbolov *i* a *j* - $\boldsymbol{a_{ij}}$
- minimalizovať čas potrebný na napísanie textu

Rozloženie monitorov (prvkov) na kontrolnom paneli

- *n* monitorov (prvkov)
- minimalizovať únavu očí pri kontrole

...

Konkrétny príklad

C.E. Nugent, T.E. Vollmann and J. Ruml – 1968

Ozn: *[nug12]*

$$A = \begin{pmatrix} 0 & 1 & 2 & 3 & 1 & 2 & 3 & 4 & 2 & 3 & 4 & 5 \\ 1 & 0 & 1 & 2 & 2 & 1 & 2 & 3 & 3 & 2 & 3 & 4 \\ 2 & 1 & 0 & 1 & 3 & 2 & 1 & 2 & 4 & 3 & 2 & 3 \\ 3 & 2 & 1 & 0 & 4 & 3 & 2 & 1 & 5 & 4 & 3 & 2 \\ 1 & 2 & 3 & 4 & 0 & 1 & 2 & 3 & 1 & 2 & 3 & 4 \\ 2 & 1 & 2 & 3 & 1 & 0 & 1 & 2 & 2 & 1 & 2 & 3 \\ 3 & 2 & 1 & 2 & 2 & 1 & 0 & 1 & 3 & 2 & 1 & 2 \\ 4 & 3 & 2 & 1 & 3 & 2 & 1 & 0 & 4 & 3 & 2 & 1 \\ 2 & 3 & 4 & 5 & 1 & 2 & 3 & 4 & 0 & 1 & 2 & 3 \\ 3 & 2 & 3 & 4 & 2 & 1 & 2 & 3 & 1 & 0 & 1 & 2 \\ 4 & 3 & 2 & 3 & 3 & 2 & 1 & 2 & 2 & 1 & 0 & 1 \\ 5 & 4 & 3 & 2 & 4 & 3 & 2 & 1 & 3 & 2 & 1 & 0 \end{pmatrix}; \quad B = \begin{pmatrix} 0 & 5 & 2 & 4 & 1 & 0 & 0 & 6 & 2 & 1 & 1 & 1 \\ 5 & 0 & 3 & 0 & 2 & 2 & 2 & 0 & 4 & 5 & 0 & 0 \\ 2 & 3 & 0 & 0 & 0 & 0 & 0 & 5 & 5 & 2 & 2 & 2 \\ 4 & 0 & 0 & 0 & 5 & 2 & 2 & 10 & 0 & 0 & 5 & 5 \\ 1 & 2 & 0 & 5 & 0 & 10 & 0 & 0 & 0 & 5 & 1 & 1 \\ 0 & 2 & 0 & 2 & 10 & 0 & 5 & 1 & 1 & 5 & 4 & 0 \\ 0 & 2 & 0 & 2 & 0 & 5 & 0 & 10 & 5 & 2 & 3 & 3 \\ 6 & 0 & 5 & 10 & 0 & 1 & 10 & 0 & 0 & 0 & 5 & 0 \\ 2 & 4 & 5 & 0 & 0 & 1 & 5 & 0 & 0 & 0 & 10 & 10 \\ 1 & 5 & 2 & 0 & 5 & 5 & 2 & 0 & 0 & 0 & 5 & 0 \\ 1 & 0 & 2 & 5 & 1 & 4 & 3 & 5 & 10 & 5 & 0 & 2 \\ 1 & 0 & 2 & 5 & 1 & 0 & 3 & 0 & 10 & 0 & 2 & 0 \end{pmatrix}$$

$$\Pi_{min} = \{12,7,9,3,4,8,11,1,5,6,10,2\}$$

QAP – hľadanie rieš.

$\boldsymbol{S_n}$ - množina všetkých permutácií $\{1,2,\dots n\}$

⇒ prehľadanie a porovnanie všetkých riešení: $\boldsymbol{n}! \rightarrow \boldsymbol{O}(\boldsymbol{n}!)$

⇒ použiteľné len pre veľmi nízke hodnoty $\boldsymbol{n}$ $(\boldsymbol{n} \approx \mathbf{20})$

⇒ pri použití $\boldsymbol{n}$ procesorov, zníženie času na $\boldsymbol{O}((\boldsymbol{n}-\mathbf{1})!)$

QAP – hľadanie rieš.

Taillard: Robust taboo search for the QAP **(1990)**

- **TS** – prehľadávanie okolia aktuálneho riešenia; prijatie presunu do novej pozície, pokiaľ tomu nebráni Tabu list
- **Okolie:** každé riešenie získané výmenou práve jedného páru v permutácii aktuálneho riešenia
- **Prehľadávanie okolia:** úplné expandovanie aktuálneho riešenia (vygenerovanie celého okolia) + ohodnotenie celého okolia – výber najlepšieho nového riešenia
- **Tabu list:** po experimentoch Tabu list s variabilnou dĺžkou; pozície vymenených prvkov v permutácii
- **Aspiračné kritérium:** Povolenie výmeny prvkov napriek tomu že sú v Tabu liste, ak zaručia nájdenie riešenia lepšieho ako doteraz najlepšieho riešenia

7	5	1	6	0	4	9	2	3	8

5	**7**	1	6	0	4	9	2	3	8
1	5	**7**	6	0	4	9	2	3	8
6	5	1	**7**	0	4	9	2	3	8

$\{1,3\}, \{5,7\}, \{4,9\}, \ldots$

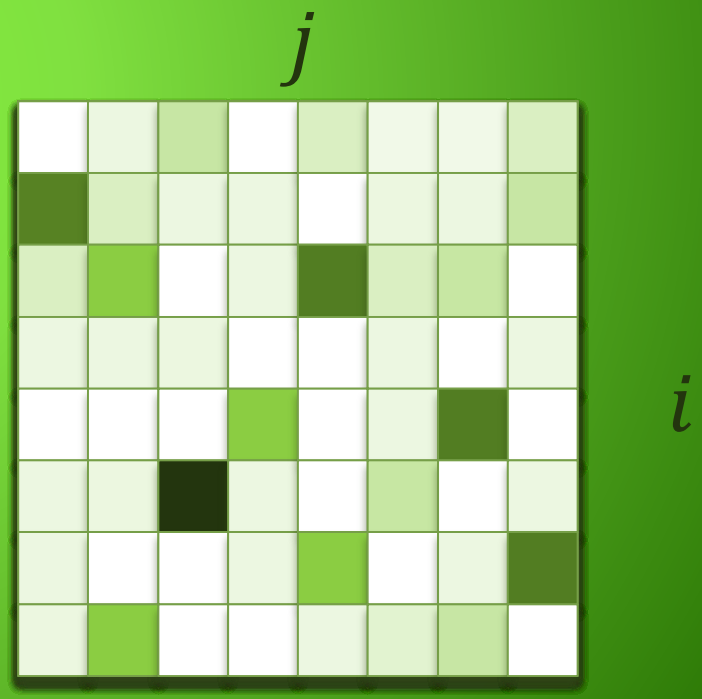
⋮

QAP – hľadanie riešenia

Mideendorf - Reischle - Schmeck: Ant algorithm **for QAP (2002)**

- **Veľkosť kolónie:** 50
- **Priradenie počas prvých *k* iterácií:** náhodné
- **Feromón:** pri priradení i k j - $\tau_{ij} = \tau_{ij} + c.F$
- **Priradenie počas ďalších iterácií:** $\mathrm{P_{ij}} = \frac{1+\tau_{ij}}{n+\sum \tau}$
- **Úbytok feromónu:** $\tau_{ij} = \rho.\tau_{ij}$

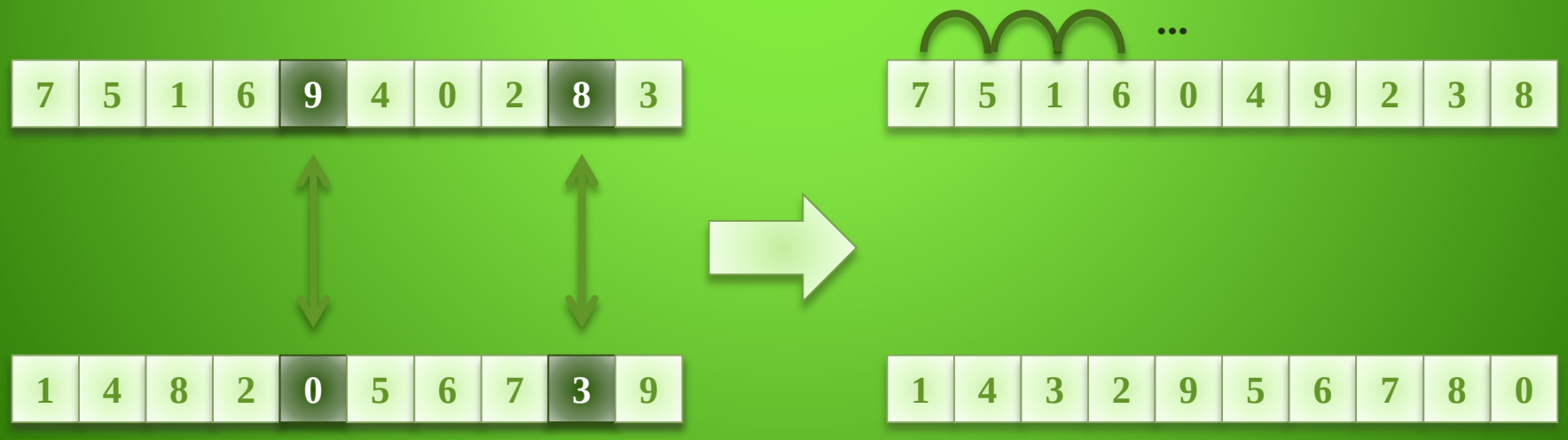
$(\tau_{ij}) =$



QAP – hľadanie riešenia

Ji – Wu – Liu: A new hybrid genetic algorithm **(2006)**

- **Veľkosť populácie:** 100 jedincov
- **Jedince:** vektory – permutácie
- **Selekcia rodičov:** náhodná
- **Kríženie:** uniformné; 20% génov vybraných náhodne
- **Selekcia potomkov:** elitistický výber medzi rodičom a potomkom
- **Lokálna optimalizácia:** výmena párov



QAP – hľadanie rieš.

Hussin - Stutzle: Simulated annealing for solving large QAP instances **(2010)**

- **SA with temperature re-heating** – väčší priestor pre únik z lokálneho minima
- **Začiatočná permutácia π:** vygenerovaná náhodne s uniformným rozdelením pravdepodobnosti
- **Začiatočná teplota:** $T_0 = 0{,}005.f(\pi)$
- **Generovanie okolia π:** exhaustívne (výmena pozícii *i* a *j*) – prijatie výmeny na základe Metropolisovho kritéria
- **Schéma chladenia:**
  - udržiavanie teploty počas $c.n$ výmen ($c = 100$)
  - geometrické chladenie - $T_{i+1} = \alpha.T_0;\ \ \alpha = 0{,}9$
  - pri dosiahnutí teploty $T_i \leq 1$ – obnovenie teploty ($T_i = T_0$) a reštartovanie procesu
- **Ukončenie algoritmu:** po vypršaní časového limitu

$$p(s_{ij}) = \min\left\{1; e^{-\frac{f(\pi')-f(\pi)}{T}}\right\}$$

7	5	1	6	0	4	9	2	3	8

QAP – hľadanie rieš.

Mamaghani - Meybodi: An application of Imperialist Competitive Algorithm to Solve the QAP **(12/2011)**

Aplikácia algoritmu na riešenie QAP:

- **Kolónia:** vektor priradenia
- **Účelová funkcia:** *F*
- **Vládca:** kolónia (vektor) s najmenšou hodnotou účelovej funkcie
- **Asimilačný operátor**
  - náhodný výber 50% buniek vo vektore vládcu
  - skopírovanie vybraných buniek do vektora novej kolónie
  - postupné prechádzanie buniek kolónie a kopírovanie tých, ktoré sa ešte v novej kolónii nenachádzajú

kolónia	2	5	0	7	8	3	1	4	6	9
vládca	1	5	6	2	9	0	3	8	4	7
nová kolónia	1	5	6	2	0	3	4	8	9	7

QAP – hľadanie rieš.

Taillard: Robust taboo search for the quadratic assignment problem

- Oddelené a nezávislé TS s rôznymi začiatočnými riešeniami
- Žiadna výmena informácií medzi vláknami
- Najlepšie riešenie vybrané na záver spomedzi všetkých

Talbi, Hafidi, Geib: Parallel adaptive tabu search for large optimization problems

- Rovnako ako v predchádzajúcom prípade, ale každá inštancia pracuje s inými parametrami
- Použitá spoločná pamäť držiaca priebežne najlepšie globálne riešenie

De Falco, Del Balio: Improving search by incorporating evolution principles in parallel tabu search

- Kooperatívny TS (podstatný rozdiel oproti predošlým)
- Pravidelne sa vyhodnocuje doposiaľ najlepšie nájdené rieš. a časť procesov nahradí svoje nájdené riešenie týmto najlepším

Talbi, Bachelet: A parallel cooperative metaheuristic

- Kombinácia TS a GA – na jednotlivých procesoch samostatný TS – nájdené riešenia sa prenášajú do spoločnej pamäte a tam s nimi pracuje GA
- Zabezpečená dostatočná rôznorodosť populácie GA a zároveň aj vysoká kvalita jedincov

QAP – hľadanie rieš.

Battiti, Tecchiolli: Parallel based search for combinatorial optimization

- Samostatný GA
- Populácia rozdelená na subpopulácie – evolúcia každej na samostatnom CPU
- Občasná výmena informácií – najlepších riešení

Tang, Lim, Ong, Er: Parallel Memetic Algorithm with Selective Local Search for Large Scale Quadratic Assignment Problems

- Pokročilejší hybridný GA – ostrovný model paralelného memetického alg. s LS
- Každý jedinec disponuje vlastnou pamäťou – LS
- Paralelný beh niekoľkých evolúcií, výmena najlepších jedincov

Talbi, Roux, Fonlupt, Robillard: Parallel Ant Colonies for Combinatorial Optimization Problems

- Synchrónny master/workers model
- Master drží centrálnu pamäť, cez ktorú prebieha komunikácia; v pamäti je pheromone matrix a najlepšie nájdené riešenie
- Na staniciach workers prebieha hľadanie na základe ACO a LS s TS – stanica obdrží pher. matrix, skonštruuje riešenie, aplikuje LS s TS a pošle na master

QAP – hľadanie rieš.

- Silná závislosť na riešenom probléme (vstupných údajoch) pri rovnakom n
- Hodnotenie a porovnávanie riešení $\approx \boldsymbol{O(n^2)}$
- **Nevyužívanie znalostí** – neinformované prehľadávanie okolia riešenia
  - Prehľadávanie vedené výlučne metaheuristikami
- Zlyhávanie pri $n > 200$ (veľkosť priestoru >200!)

$$f(\Pi) > f(\Pi') \ ?$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\left(a_{\pi_i,\pi_j}b_{ij} - a_{\pi'_i,\pi'_j}b_{ij}\right) > 0 \ ?$$

7	5	1	6	0	4	9	2	3	8

5	7	1	6	0	4	9	2	3	8
1	5	7	6	0	4	9	2	3	8
6	5	1	7	0	4	9	2	3	8

⋮

QAP – hľadanie rieš.

Znalosti:

$\Pi = \{1,2,3,4\}$:

$$A = \begin{pmatrix} 0 & 1 & 3 & 4 \\ 1 & 0 & 2 & 2 \\ 3 & 2 & 0 & 5 \\ 4 & 2 & 5 & 0 \end{pmatrix} \qquad B = \begin{pmatrix} 0 & 5 & 7 & 10 \\ 5 & 0 & 4 & 3 \\ 7 & 4 & 0 & 12 \\ 10 & 3 & 12 & 0 \end{pmatrix}$$

$$\Pi = \{1,2,3,4\}; f(\Pi) = 140$$

$$\Pi_{1,2} = \{2,1,3,4\}; f\left(\Pi_{1,2}\right) = 123$$

$$\Pi_{2,4} = \{1,4,3,2\}; f\left(\Pi_{2,4}\right) = 101$$

$\Pi = \{1,4,3,2\}$:

$$A = \begin{pmatrix} 0 & 4 & 3 & 1 \\ 4 & 0 & 5 & 2 \\ 3 & 5 & 0 & 2 \\ 1 & 2 & 2 & 0 \end{pmatrix} \qquad B = \begin{pmatrix} 0 & 5 & 7 & 10 \\ 5 & 0 & 4 & 3 \\ 7 & 4 & 0 & 12 \\ 10 & 3 & 12 & 0 \end{pmatrix}$$

QAP – hľadanie rieš.

Znalosti:

$\Pi = \{1,4,3,2\}$:

$$A = \begin{pmatrix} 0 & 4 & 3 & 1 \\ 4 & 0 & 5 & 2 \\ 3 & 5 & 0 & 2 \\ 1 & 2 & 2 & 0 \end{pmatrix} \qquad B = \begin{pmatrix} 0 & 5 & 7 & 10 \\ 5 & 0 & 4 & 3 \\ 7 & 4 & 0 & 12 \\ 10 & 3 & 12 & 0 \end{pmatrix}$$

$$\Pi = \{1,4,3,2\}; f(\Pi) = 101$$

$$\Pi_{1,3} = \{3,4,1,2\}; f\left(\Pi_{1,3}\right) = 100$$

$$A = \begin{pmatrix} 0 & 5 & 3 & 2 \\ 5 & 0 & 4 & 2 \\ 3 & 4 & 0 & 1 \\ 2 & 2 & 1 & 0 \end{pmatrix} \qquad B = \begin{pmatrix} 0 & 5 & 7 & 10 \\ 5 & 0 & 4 & 3 \\ 7 & 4 & 0 & 12 \\ 10 & 3 & 12 & 0 \end{pmatrix}$$

QAP – hľadanie rieš.

Znalosti:

$\Pi = \{3,4,1,2\}$:

$$A = \begin{pmatrix} 0 & 5 & 3 & 2 \\ 5 & 0 & 4 & 2 \\ 3 & 4 & 0 & 1 \\ 2 & 2 & 1 & 0 \end{pmatrix} \qquad B = \begin{pmatrix} 0 & 5 & 7 & 10 \\ 1 & 0 & 4 & 3 \\ 7 & 4 & 0 & 12 \\ 10 & 3 & 12 & 0 \end{pmatrix}$$

$$X = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \rightarrow X = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \rightarrow X = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

$$\rightarrow X = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \rightarrow X = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

$$f(\Pi) = 140 \rightarrow f(\Pi_{\min}) = 100$$

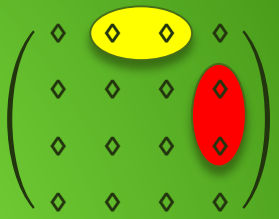
$$\boldsymbol{\Pi = \{3,4,1,2\}; f(\Pi) = 100}$$

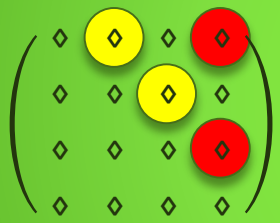
QAP – hľadanie rieš.

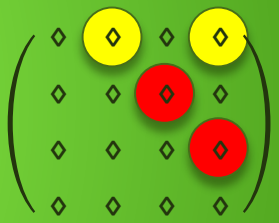
Výmena prvkov na pozíciách v permutácii – zmena matice:

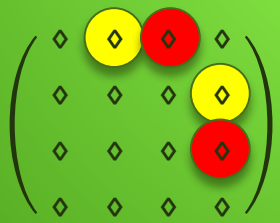
$n = 4$

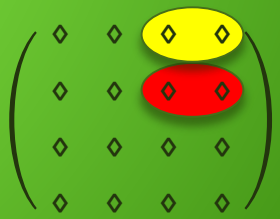
1 - 2 $A =$

2 - 3 $A =$ 

1 - 3 $A =$ 

2 - 4 $A =$ 

1 - 4 $A =$ 

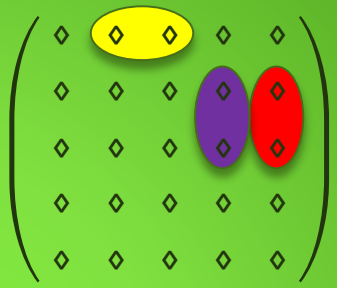
3 - 4 $A =$ 

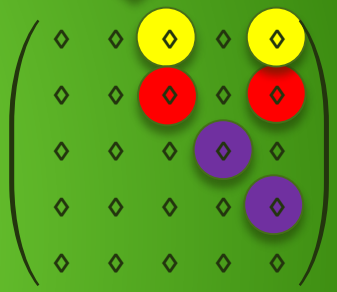
QAP – hľadanie rieš.

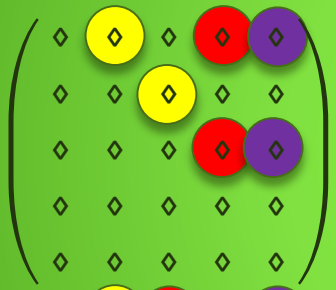
Výmena prvkov na pozíciách v permutácii – zmena matice:

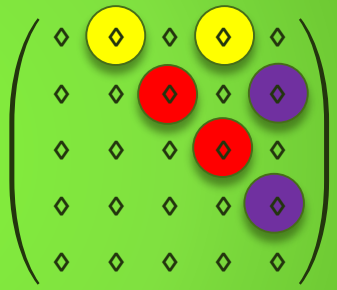
$n = 5$

1 - 2 $A =$

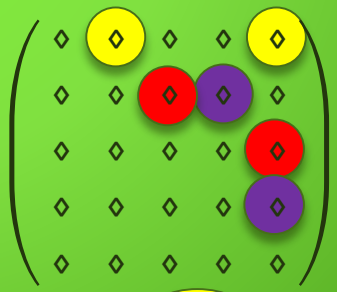
2 - 3 $A =$ 

3 - 5 $A =$ 

1 - 3 $A =$ 

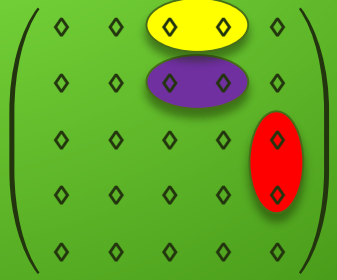
2 - 4 $A =$ 

1 - 4 $A =$

2 - 5 $A =$ 

4 - 5 $A =$

1 - 5 $A =$

3 - 4 $A =$ 

QAP – hľadanie rieš.

Local Search with Simulated Annealing Elements (LSSA)

$\Pi_1 =$

7	5	1	6	0	4	9	2	3	8

L

1	1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0

$n-k$ k

0	0	0	1	0	0	0	0	0	1
0	1	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	1	0	0
0	0	1	0	0	0	1	0	0	0
0	0	1	0	0	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1

QAP – hľadanie rieš.

Local Search with Simulated Annealing Elements (LSSA)

$$\Pi_1 = \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 7 & 5 & 1 & 6 & 0 & 4 & 9 & 2 & 3 & 8 \\ \hline \end{array}$$

	y_1	y_2	y_3	…						y_n
Y_1	0	0	0	1	0	0	0	0	0	1
Y_2	0	1	0	0	0	0	0	1	0	0
	0	0	1	1	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	1	0
⋮	0	0	0	0	0	1	0	1	0	0
	0	0	1	0	0	0	1	0	0	0
	0	0	1	0	0	0	0	0	1	0
Y_L	0	0	0	0	1	0	0	0	0	1

Y^* (min z)

$$z = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{\pi_i,\pi_j} b_{ij} y_i y_j$$

⋮

QAP – hľadanie rieš.

Local Search – *k*-opt

Y^*	0	0	1	0	0	0	1	0	0	0
Π_i	7	5	1	6	0	4	9	2	3	8

Y^*	0	0	1	0	0	0	1	0	0	0
Π_j	6	0	1	5	3	4	9	8	7	2
⋮										
	2	3	1	5	0	4	9	9	8	7
	8	3	1	5	2	0	9	6	4	7
	0	5	1	9	4	8	9	7	2	3
	4	2	1	3	7	0	9	8	6	5
	⋮									

M

$$\Pi^* = \arg\min z(\Pi_j)$$

$$z(\Pi) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{\pi_i,\pi_j} b_{ij}$$

QAP – hľadanie rieš.

Simulated annealing elements

Π_i

7	5	1	6	0	4	9	2	3	8

Π^*

4	2	1	3	7	0	9	8	6	5

ak $z(\Pi^*) < z(\Pi_i) \quad \rightarrow \quad \Pi_{i+1} = \Pi^*$

ak $z(\Pi^*) > z(\Pi_i) \quad \rightarrow \quad \Pi_{i+1} = \Pi^*$ s pravdepodobnosťou P

P:

ak v $(i-1)$ –vej iterácii: $z(\Pi^*) < z(\Pi_{i-1})$:

$$P = 0{,}1 \frac{1}{\lceil z(\pi^*) - z(\pi_i) \rceil} \cdot \frac{k-2}{n},$$

inak:

$$P = \left(1 + 0{,}1 \frac{1}{\lceil z(\pi^*) - z(\pi_i) \rceil}\right) \cdot \frac{k-2}{n} \cdot P$$

QAP – hľadanie rieš.

Simulated annealing elements

- čím dlhšie nevieme nájsť lepšie Π, tým **vyššia** P
- čím horšie Π^* sú skúmané, tým **nižšia** P
- *čím nižšia hodnota* ***k****, tým* ***nižšia*** P – hodnota k sa postupne znižuje

---

P:

ak v $(i-1)$ –vej iterácii: $z(\Pi^*) < z(\Pi_{i-1})$:

$$P = 0{,}1\frac{1}{\lceil z(\pi^*) - z(\pi_i)\rceil} \cdot \frac{k-2}{n},$$

inak:

$$P = \left(1 + 0{,}1\frac{1}{\lceil z(\pi^*) - z(\pi_i)\rceil}\right) \cdot \frac{k-2}{n} \cdot P$$

QAP – hľadanie rieš.

Simulated annealing elements

- čím dlhšie nevieme nájsť lepšie Π, tým **vyššia** *P*
- čím horšie Π* sú skúmané, tým **nižšia** *P*
- *čím nižšia hodnota* ***k****, tým* ***nižšia*** *P* – hodnota *k* sa postupne znižuje

1	1	1	1	0	0	0	0	0	0

$n-k$ k

→

6	0	1	5	3	4	9	8	7	2

⋮

1	1	1	1	1	1	0	0	0	0

$n-k$ k

→

2	5	7	1	3	6	8	4	9	0

⋮

1	1	1	1	1	1	1	1	0	0

$n-k$ k

→

1	5	4	9	7	3	8	9	0	2

Výsledky hľadania

QAP – hľadanie rieš.

Problém	n	BKS	LSSA		[mama] 2010		[ahuja] 2000		[ghand] 2010		[talbi01] 2001		[wu] 2008		[tang] 2006		[james] 2009		[tsu] 2008		[mis] 2003	
			APD	Time	APD	Time	APD	Time	APD	Time	APD	Time	APD	Time	APD	Time	APD	Time	APD	Time	APD	Time
kra30a	30	88900	0	0,17	-	-	0	150,7	-	-	-	-	0	0,29	-	-	-	-	0	2,84	2,45	0,77
kra30b	30	91420	0	0,27	-	-	0	165,3	-	-	-	-	0	0,34	-	-	-	-	0	2,82	1,14	0,77
nug20	20	2570	0	1,45	-	-	0	48,9	0	6,21	-	-	-	-	-	-	-	-	-	-	-	-
nug30	30	6124	0	1,94	-	-	0,07	177,1	-	-	0	83	0	0,36	-	-	0	11	-	-	0,85	0,77
tho40	40	240516	0	1,13	-	-	0,2	479,1	-	-	-	-	0,041	1,98	-	-	-	-	-	-	1,18	1,74
tho50	50	8133398	0,05	2,4	-	-	-	-	-	-	-	-	-	-	0,12	3123,6	-	-	-	-	-	-
sko42	42	15812	0	2,1	-	-	0	503,1	-	-	-	-	0	1,57	-	-	0	52,2	-	-	0,61	1,96
sko49	49	23386	0,006	3,9	-	-	0,21	626,1	-	-	-	-	0,038	3,78	-	-	0	110,9	0,024	12,84	0,57	3,02
sko56	56	34458	0	5,2	-	-	0,02	1488	0,144	522,73	-	-	0	7,36	-	-	0	209,9	-	-	0,57	4,41
sko64	64	48498	0	9,4	-	-	0,17	1894,1	-	-	0,001	974	0	12,11	-	-	0	429,6	0,005	28,13	0,5	6,47
sko72	72	66256	0,002	15,45	-	-	0,27	2539	0,188	1304,26	-	-	0,042	35,39	-	-	0	695	-	-	0,55	9,07
sko81	81	90998	0,009	24,12	-	-	0,2	5482,1	0,07	2004,03	0,048	2041	0,067	57,28	-	-	0	1212,5	0,044	58,79	0,43	12,8
sko90	90	115534	0,013	34,6	-	-	0,27	6348,9	-	-	-	-	0,073	102,5	-	-	0	1935,3	0,05	80,98	0,44	17,3
sko100a	100	152002	0,011	42,4	0,785	-	0,19	8304,1	0,099	4211,41	0	8220	0,051	174,13	0,06	283,2	0	3042,4	-	-	0,36	23,6
sko100b	100	153890	0,005	45,18	0,151	-	0,14	7364,7	-	-	0	8360	0,039	165,5	0,04	823	0	3091,4	-	-	0,35	23,6
sko100c	100	147862	0,006	41,6	1,063	-	0,01	10157,1	-	-	0	8220	0,015	158,54	0,01	531,6	0	3154,4	-	-	0,34	23,4
sko100d	100	149576	0,019	39,45	1,531	-	0,17	10151,1	-	-	0,001	12060	0,022	184,41	0,07	312,6	0	3091,9	-	-	0,43	23,6
sko100e	100	149150	0,007	44,6	0,799	-	0,23	10563,5	-	-	0	-	0,03	167,31	0,01	512,5	0	3085,6	-	-	0,45	23,6
sko100f	100	149036	0,011	43,31	0,896	-	0,19	10739,5	-	-	-	-	0,017	170,88	0,08	872,1	0,003	3096,4	-	-	0,47	23,6

BKS (best known solution) – najlepšie známe riešenie

APD (average percentual deviation) – priemer. perc. odch. od BKS -> $APD = 100 \cdot \frac{C-BKS}{BKS}$

C – hodnota nájdeného riešenia; **Time** – kumulovalný čas CPU

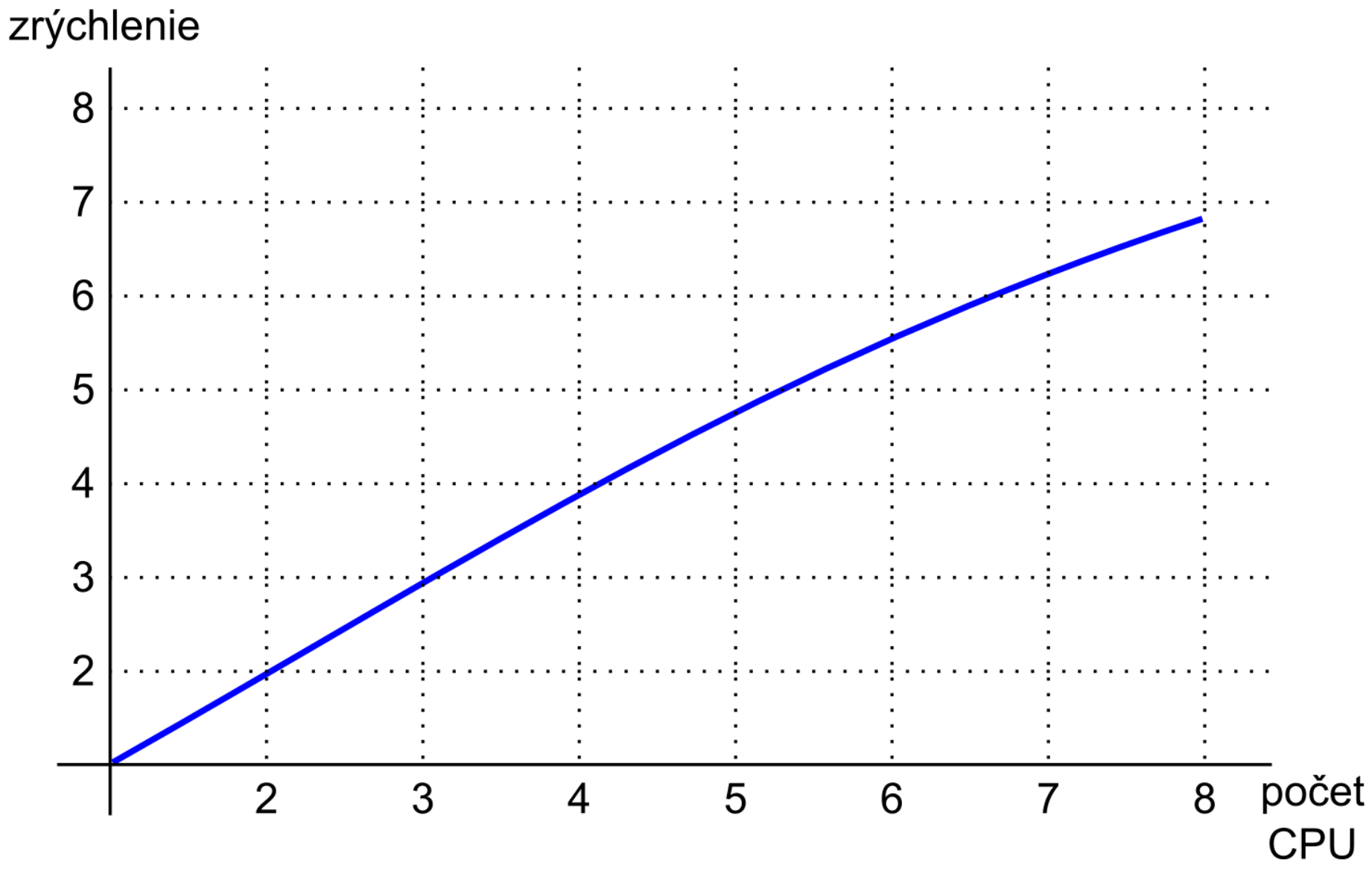
Výsledky hľadania

QAP – hľadanie rieš.

Problém	n	BKS	LSSA		[mama] 2010		[ahuja] 2000		[ghand] 2010		[talbi01] 2001		[wu] 2008		[tang] 2006		[james] 2009		[tsu] 2008		[mis] 2003	
			APD	Time	APD	Time	APD	Time	APD	Time	APD	Time	APD	Time	APD	Time	APD	Time	APD	Time	APD	Time
tai20a	20	703482	0	1,12	0	-	-	-	-	-	0	26	-	-	-	-	0	1,5	-	-	-	-
tai25a	25	1167256	0	4,14	-	-	-	-	-	-	0,736	50	-	-	-	-	0	5,2	-	-	2,32	0,47
tai30a	30	1818146	0	6,94	1,462	-	-	-	0,68	72,64	0,018	87	-	-	-	-	0	10,6	-	-	2,23	0,78
tai35a	35	2422002	0	11,41	-	-	-	-	-	-	0,215	145	-	-	-	-	0	20,2	-	-	2,37	1,21
tai40a	40	3139370	0,094	19,81	2,237	-	-	-	1,349	233,2	0,442	224	-	-	-	-	0,148	37,7	0,495	6,81	2,45	1,76
tai50a	50	4941410	0,331	42,16	2,218	-	-	-	1,803	462,58	0,781	467	-	-	-	-	0,44	104,6	0,738	13,35	2,6	3,31
tai60a	60	7208572	0,349	76,45	2,207	-	-	-	1,93	948,61	0,919	820	-	-	-	-	0,476	265,1	0,881	23,2	2,56	5,6
tai80a	80	13557864	0,418	134,84	2,673	-	-	-	1,487	1242,47	0,663	2045	-	-	-	-	0,57	947,9	0,547	57	1,96	12,9
tai100a	100	21125314	0,218	243,91	1,375	-	-	-	-	-	0,278	8340	-	-	1,13	582,9	0,558	2608,7	-	-	1,82	24,6
tai20b	20	122455319	0	0,6	-	-	-	-	-	-	0	27	-	-	-	-	0	1,7	-	-	-	-
tai25b	25	344355646	0	1,74	-	-	-	-	0,552	19,42	0	50	-	-	-	-	0	6,3	-	-	-	-
tai30b	30	637117113	0	2,96	-	-	-	-	1,374	26,74	0	90	-	-	-	-	0	13,8	-	-	-	-
tai35b	35	283315445	0	5,4	-	-	-	-	1,103	53,34	0,0408	147	-	-	-	-	0	25,9	-	-	-	-
tai40b	40	637250948	0,024	9,48	-	-	-	-	-	-	0,464	240	-	-	-	-	0	48,3	0	6,71	-	-
tai50b	50	458821517	0	17,51	-	-	-	-	-	-	0,2531	480	-	-	-	-	0	138,1	-	-	-	-
tai60b	60	608215054	0,006	35,66	-	-	-	-	-	-	0,2752	855	-	-	-	-	0	305	0,001	23,17	-	-
tai80b	80	818415043	0,051	83,49	-	-	-	-	0,717	1147,28	0,7185	2073	-	-	-	-	0	1106,3	0,208	56,13	-	-
tai100b	100	1185996137	0,004	188,47	-	-	-	-	-	-	-	-	-	-	0	220,7	0,001	2403,8	0,055	114,89	-	-
tai150b	150	498896643	0,052	398,08	0,67	-	-	-	-	-	-	-	-	-	-	-	0,076	73298,8	-	-	-	-
tai256c	256	44759294	0,067	442,13	0,56	-	-	-	-	-	0,085	44460	-	-	-	-	0,136	73298,8	-	-	-	-
wil50	50	48816	0	6,81	-	-	0,07	1057,6	0,068	336,72	0,008	441	0,028	5,06	-	-	-	-	-	-	0,2	3,19
wil100	100	273038	0	167,76	-	-	0,18	10271,9	-	-	0,006	28680	0,041	176,28	0,02	560,1	0	3161,3	-	-	0,22	23,6

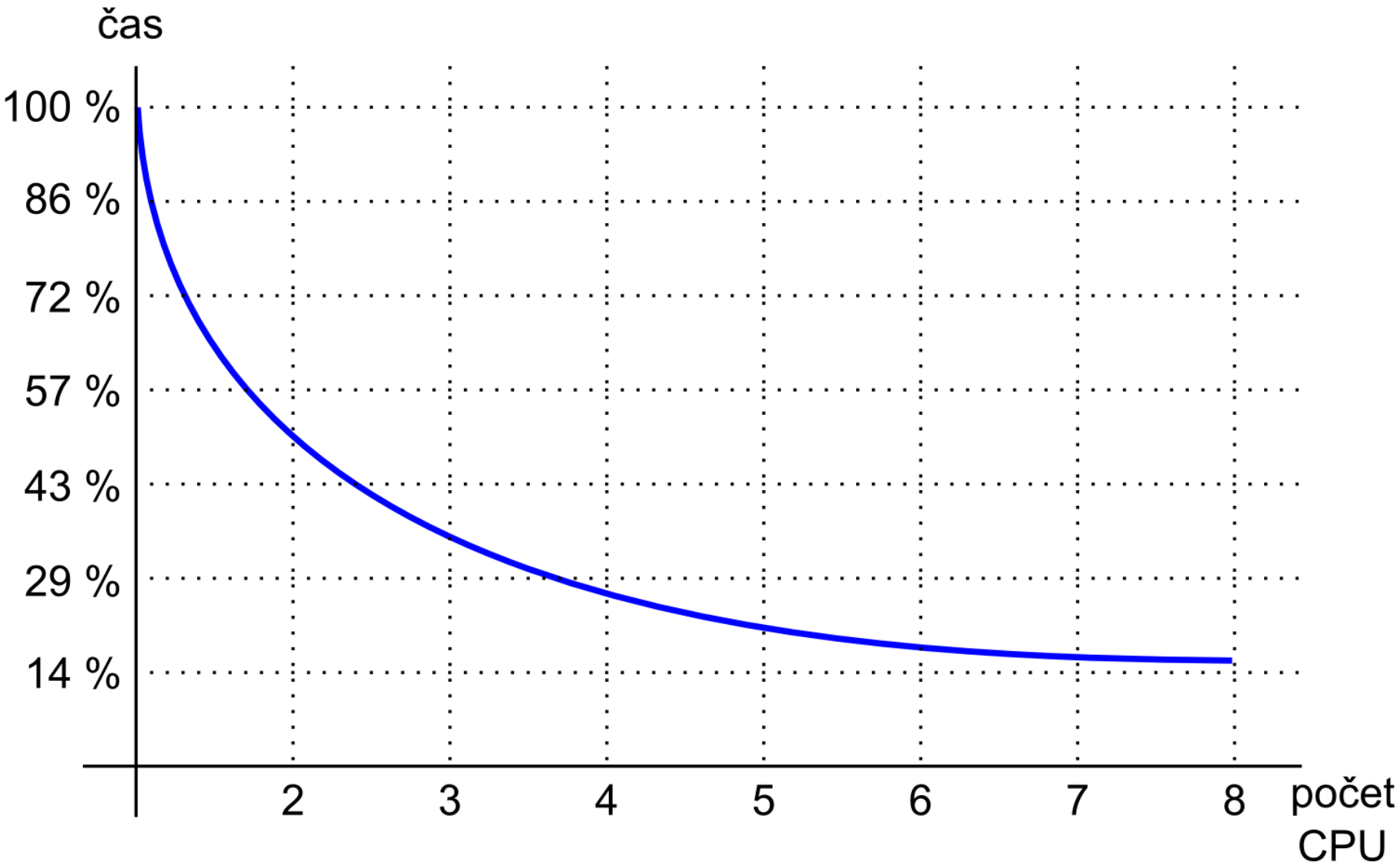
Výsledky hľadania

QAP – hľadanie rieš.



Výsledky hľadania

QAP – hľadanie rieš.



QAP – hľadanie rieš.

Hybrid parallel cooperative island model genetic algorithm (HyPa-CIGA)

island 1

ostrov 1

GA

island 5

ostrov 5

GA

island 3

ostrov 3

GA

island 4

ostrov 4

GA

island 2

ostrov 2

GA

island n

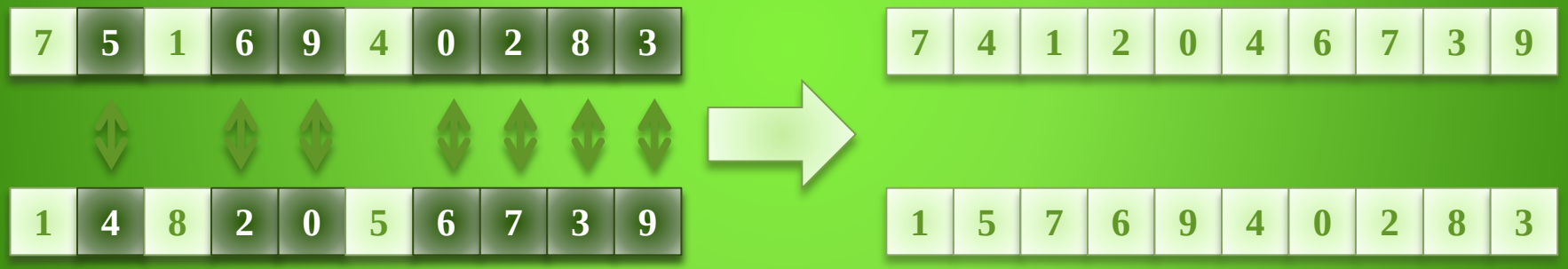
ostrov *n*

GA

QAP – hľadanie rieš.

Genetický algoritmus

1. **Vygenerovanie úvodnej populácie** – náhodné
2. **Ohodnotenie, výber rodičov** – pravdepodobnostný výber – väčšie udržiavanie rôznorodosti populácie
3. **Kríženie** – náhodné párovanie, PMX kríženie – prenesenie, resp. zachovanie min. 70% génov.



4. **Mutácia** – výmenná mutácia (P = 0,01)

5. **Selekcia, vytvorenie novej populácie** – elitistická selekcia (rodičia + potomkovia)

QAP – hľadanie rieš.

Hybrid parallel cooperative island model genetic algorithm (HyPa-CIGA)

island 1

ostrov 1

GA

island 5

ostrov 5

GA

rock island

kamenný ostrov

s-LSSA

island 3

ostrov 3

GA

island 4

ostrov 4

GA

island 2

ostrov 2

GA

island n

ostrov *n*

GA

QAP – hľadanie rieš.

s-LSSA

1. Náhodné vygenerovanie úvodného priradenia
2. Preskúmanie vhodných existujúcich umiestnení vo vektore
3. Local serach (*k*-opt)
4. Princíp simulovaného žíhania

Lokálne hľadanie – lokálna optimalizácia začiatočného riešenia

QAP – hľadanie rieš.

Hybrid parallel cooperative island model genetic algorithm (HyPa-CIGA)

island 1

ostrov 1

GA

island 5

ostrov 5

GA

rock island

kamenný ostrov

s-LSSA

island 3

ostrov 3

GA

island 4

ostrov 4

GA

island 2

ostrov 2

GA

island n

ostrov *n*

GA

QAP – hľadanie rieš.

Ostrovy – úložné miesto

Ostrovy (GA):

- Priebežne aktualizované informácie o generácii, najlepších jedincoch, konvergencii populácie, rôznorodosti jedincov
- Rôznorodosť jedincov (entropia):

$$E_i = -\sum_{j=1}^{Q} p_j \log(p_j)$$

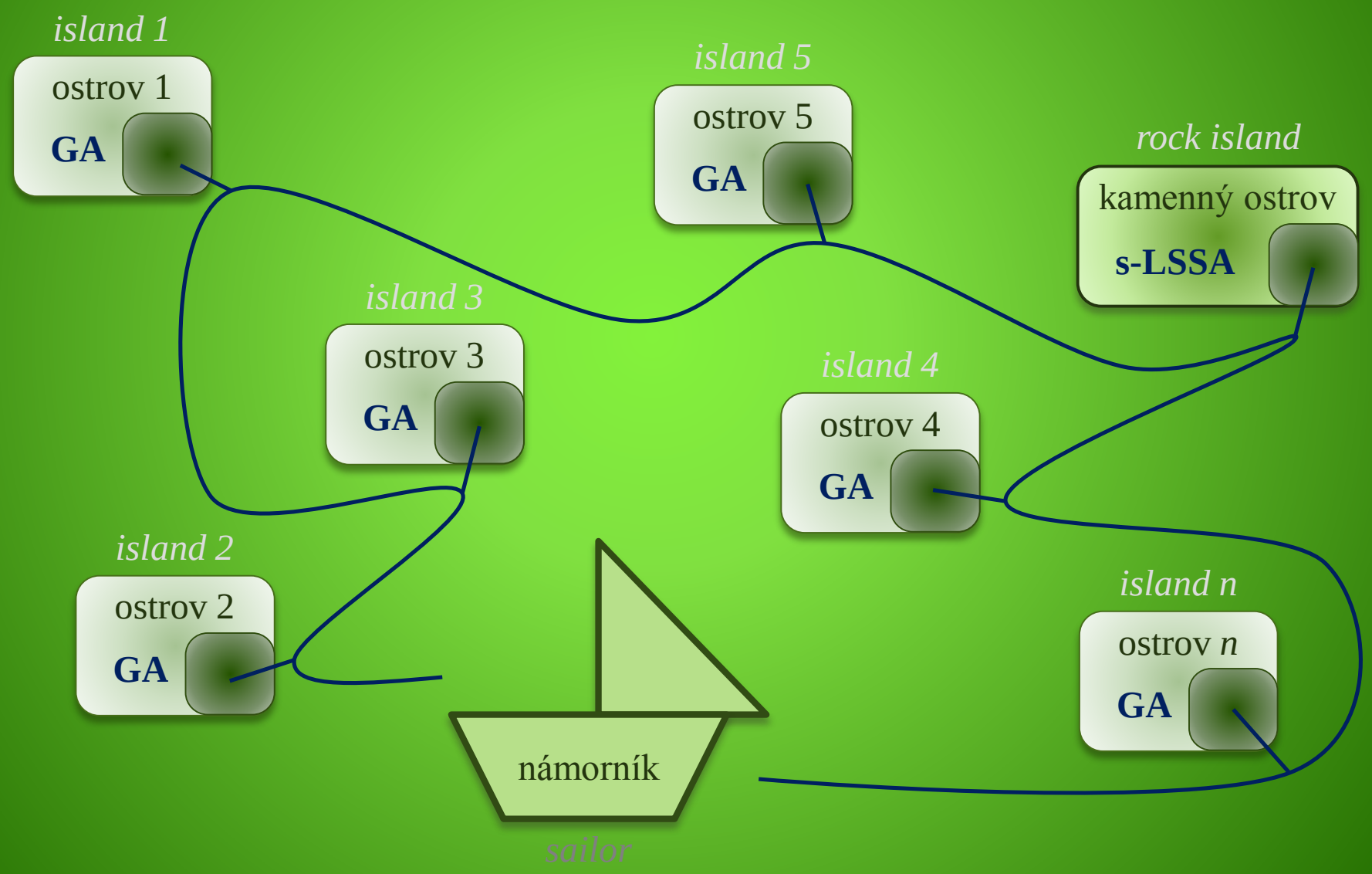
Q – počet navzájom rôznych jedincov v i-tej populácii;
$S_{i1}, S_{i2}, \dots, S_{iQ}$ - disjunktné množiny navzájom rôznych jedincov
$p_j = \frac{|S_{ij}|}{N}$; N – veľkosť populácie

Kamenný ostrov (s-LSSA):

- Priebežne aktualizované informácie o počte krokov LS a hodnotení aktuálneho riešenia

QAP – hľadanie rieš.

Hybrid parallel cooperative island model genetic algorithm (HyPa-CIGA)



QAP – hľadanie rieš.

Námorník

- Pravidelne kontroluje úložné miesta na všetkých ostrovoch vždy v rovnakom poradí
- Vyhodnocuje, uchováva a priebežne aktualizuje globálne najlepšie nájdené riešenia
- Kontroluje rozdiel rôznorodosti populácií na GA ostrovoch
- Zbiera riešenia poskytnuté kamenným ostrovom a dodáva ich na GA ostrovy (zabezpečenie vyššej rôznorodosti populácie a zároveň aplikácia LS do hľadania)
- Na kamenný ostrov dodáva do nového lokálneho hľadania najlepšie aktuálne globálne riešenie nájdené GA
- Pri poklese rôznorodosti populácie na ostrovoch dodáva do danej populácie kvalitné riešenia z iných ostrovov a z kamenného ostrova
- Rozhoduje o ukončení algoritmu

QAP – hľadanie rieš.

Výhody algoritmu

- **Kooperatívny paralelný algoritmus** – potenciál nájsť kvalitné riešenia
- **Hybridný algoritmus** – evolúcia zásobovaná jedincami optimalizovanými pomocou LS s SA – náhodnosť evolúcie ovplyvňovaná usmerneným hľadaním
- **Asynchrónny paralelný algoritmus** – jednotlivé procesy nie sú navzájom synchronizované – odpadá potreba čakať na ostatné proces (úspora času)
- **Distribúcia a analýza dát prostredníctvom externého procesu** – dáta o jednotlivých evolúciách nevyhodnocujú jednotlivé procesy, ale externé vlákno (námorník) – úspora času jednotlivých procesov

Výsledky hľadania

QAP – hľadanie rieš.

Problém	n	BKS	HyPa-CIGA		LSSA		[mama] 2010		[ahuja] 2000		[ghand] 2010		[talbi01] 2001		[wu] 2008		[tang] 2006		[james] 2009		[tsu] 2008		[mis] 2003	
			APD	Time	APD	Time	APD	Time	APD	Time	APD	Time	APD	Time	APD	Time	APD	Time	APD	Time	APD	Time	APD	Time
kra30a	30	88900	0	0,4823	0	0,17	-	-	0	150,7	-	-	-	-	0	0,29	-	-	-	-	0	2,84	2,45	0,77
kra30b	30	91420	0	0,7257	0	0,27	-	-	0	165,3	-	-	-	-	0	0,34	-	-	-	-	0	2,82	1,14	0,77
nug20	20	2570	0	4,1515	0	1,45	-	-	0	48,9	0	6,21	-	-	-	-	-	-	-	-	-	-	-	-
nug30	30	6124	0	5,5871	0	1,94	-	-	0,07	177,1	-	-	0	83	0	0,36	-	-	0	11	-	-	0,85	0,77
tho40	40	240516	0	3,2472	0	1,13	-	-	0,2	479,1	-	-	-	-	0,041	1,98	-	-	-	-	-	-	1,18	1,74
tho50	50	8133398	0,03	6,9856	0,05	2,4	-	-	-	-	-	-	-	-	-	-	0,12	3123,6	-	-	-	-	-	-
sko42	42	15812	0	5,7014	0	2,1	-	-	0	503,1	-	-	-	-	0	1,57	-	-	0	52,2	-	-	0,61	1,96
sko49	49	23386	0	11,125	0,006	3,9	-	-	0,21	626,1	-	-	-	-	0,038	3,78	-	-	0	110,9	0,024	12,84	0,57	3,02
sko56	56	34458	0	15,136	0	5,2	-	-	0,02	1488	0,144	522,73	-	-	0	7,36	-	-	0	209,9	-	-	0,57	4,41
sko64	64	48498	0	26,928	0	9,4	-	-	0,17	1894,1	-	-	0,001	974	0	12,11	-	-	0	429,6	0,005	28,13	0,5	6,47
sko72	72	66256	0	41,233	0,002	15,45	-	-	0,27	2539	0,188	1304,26	-	-	0,042	35,39	-	-	0	695	-	-	0,55	9,07
sko81	81	90998	0	69,347	0,009	24,12	-	-	0,2	5482,1	0,07	2004,03	0,048	2041	0,067	57,28	-	-	0	1212,5	0,044	58,79	0,43	12,8
sko90	90	115534	0,004	100,38	0,013	34,6	-	-	0,27	6348,9	-	-	-	-	0,073	102,5	-	-	0	1935,3	0,05	80,98	0,44	17,3
sko100a	100	152002	0,003	119,03	0,011	42,4	0,785	-	0,19	8304,1	0,099	4211,41	0	8220	0,051	174,13	0,06	283,2	0	3042,4	-	-	0,36	23,6
sko100b	100	153890	0	129,13	0,005	45,18	0,151	-	0,14	7364,7	-	-	0	8360	0,039	165,5	0,04	823	0	3091,4	-	-	0,35	23,6
sko100c	100	147862	0,001	115,46	0,006	41,6	1,063	-	0,01	10157,1	-	-	0	8220	0,015	158,54	0,01	531,6	0	3154,4	-	-	0,34	23,4
sko100d	100	149576	0,003	111,91	0,019	39,45	1,531	-	0,17	10151,1	-	-	0,001	12060	0,022	184,41	0,07	312,6	0	3091,9	-	-	0,43	23,6
sko100e	100	149150	0,005	126,28	0,007	44,6	0,799	-	0,23	10563,5	-	-	0	-	0,03	167,31	0,01	512,5	0	3085,6	-	-	0,45	23,6
sko100f	100	149036	0,007	120,85	0,011	43,31	0,896	-	0,19	10739,5	-	-	-	-	0,017	170,88	0,08	872,1	0,003	3096,4	-	-	0,47	23,6
tai20a	20	703482	0	3,1502	0	1,12	0	-	-	-	-	-	0	26	-	-	-	-	0	1,5	-	-	-	-
tai25a	25	1167256	0	11,422	0	4,14	-	-	-	-	-	-	0,736	50	-	-	-	-	0	5,2	-	-	2,32	0,47

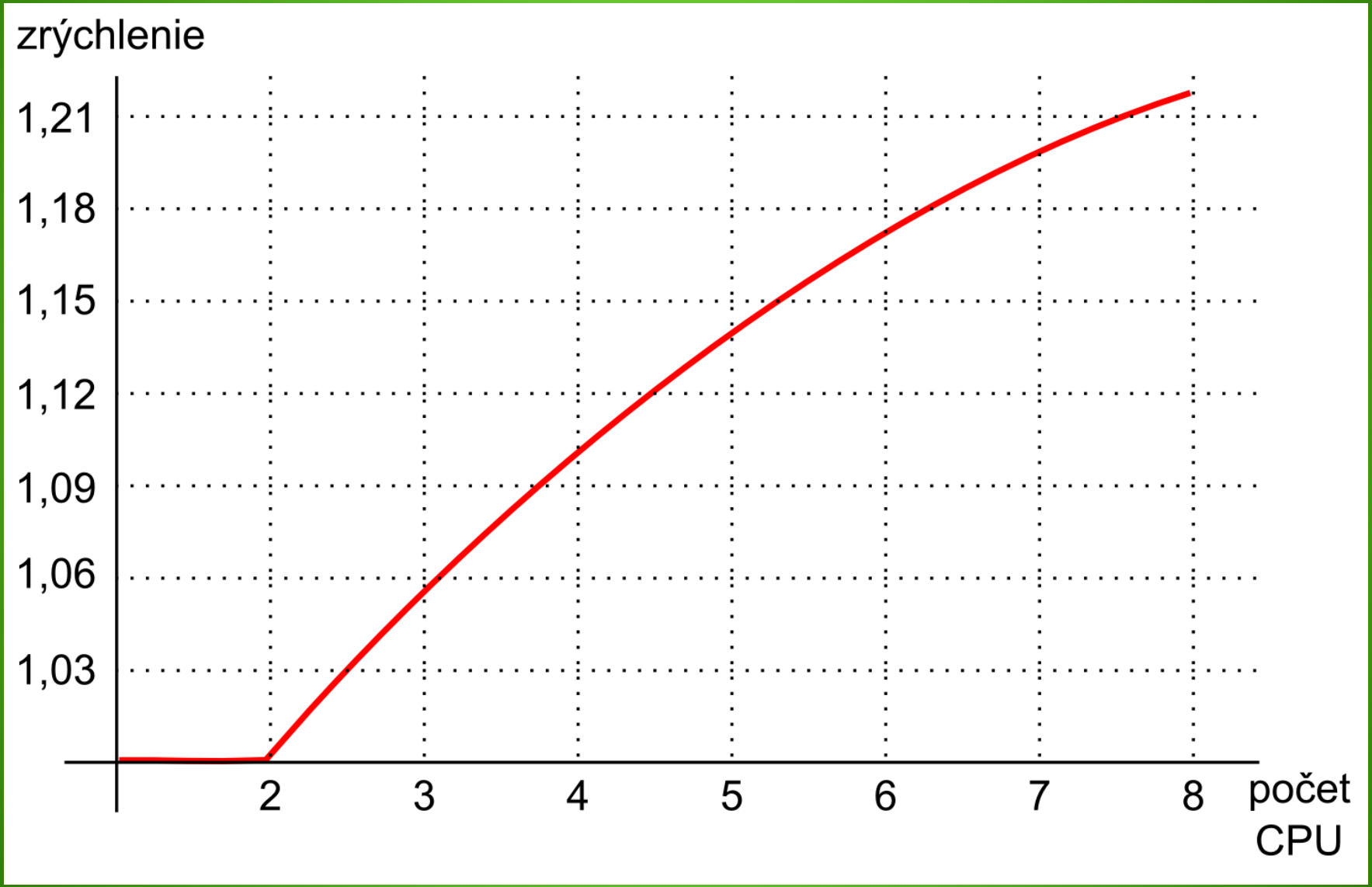
Výsledky hľadania

QAP – hľadanie rieš.

Problém	n	BKS	HyPa-CIGA		LSSA		[mama] 2010		[ghand] 2010		[talbi01] 2001		[wu] 2008		[tang] 2006		[james] 2009		[tsu] 2008		[mis] 2003	
			APD	Time	APD	Time	APD	Time	APD	Time	APD	Time	APD	Time	APD	Time	APD	Time	APD	Time	APD	Time
tai30a	30	1818146	0	20,266	0	6,94	1,462	-	0,68	72,64	0,018	87	-	-	-	-	0	10,6	-	-	2,23	0,78
tai35a	35	2422002	0	33,16	0	11,41	-	-	-	-	0,215	145	-	-	-	-	0	20,2	-	-	2,37	1,21
tai40a	40	3139370	0,018	57,342	0,094	19,81	2,237	-	1,349	233,2	0,442	224	-	-	-	-	0,148	37,7	0,495	6,81	2,45	1,76
tai50a	50	4941410	0,107	123,55	0,331	42,16	2,218	-	1,803	462,58	0,781	467	-	-	-	-	0,44	104,6	0,738	13,35	2,6	3,31
tai60a	60	7208572	0,084	215,74	0,349	76,45	2,207	-	1,93	948,61	0,919	820	-	-	-	-	0,476	265,1	0,881	23,2	2,56	5,6
tai80a	80	13557864	0,097	379,35	0,418	134,84	2,673	-	1,487	1242,47	0,663	2045	-	-	-	-	0,57	947,9	0,547	57	1,96	12,9
tai100a	100	21125314	0,09	694,47	0,218	243,91	1,375	-	-	-	0,278	8340	-	-	1,13	582,9	0,558	2608,7	-	-	1,82	24,6
tai20b	20	122455319	0	1,6214	0	0,6	-	-	-	-	0	27	-	-	-	-	0	1,7	-	-	-	-
tai25b	25	344355646	0	4,746	0	1,74	-	-	0,552	19,42	0	50	-	-	-	-	0	6,3	-	-	-	-
tai30b	30	637117113	0	8,2643	0	2,96	-	-	1,374	26,74	0	90	-	-	-	-	0	13,8	-	-	-	-
tai35b	35	283315445	0	14,94	0	5,4	-	-	1,103	53,34	0,0408	147	-	-	-	-	0	25,9	-	-	-	-
tai40b	40	637250948	0	26,433	0,024	9,48	-	-	-	-	0,464	240	-	-	-	-	0	48,3	0	6,71	-	-
tai50b	50	458821517	0	49,299	0	17,51	-	-	-	-	0,2531	480	-	-	-	-	0	138,1	-	-	-	-
tai60b	60	608215054	0	101,77	0,006	35,66	-	-	-	-	0,2752	855	-	-	-	-	0	305	0,001	23,17	-	-
tai80b	80	818415043	0,011	237,75	0,051	83,49	-	-	0,717	1147,28	0,7185	2073	-	-	-	-	0	1106,3	0,208	56,13	-	-
tai100b	100	1185996137	0	519,82	0,004	188,47	-	-	-	-	-	-	-	-	0	220,7	0,001	2403,8	0,055	114,89	-	-
tai150b	150	498896643	0,002	1164,9	0,052	398,08	0,67	-	-	-	-	-	-	-	-	-	0,076	73298,8	-	-	-	-
tai256c	256	44759294	0,031	1233,9	0,067	442,13	0,56	-	-	-	0,085	44460	-	-	-	-	0,136	73298,8	-	-	-	-
wil50	50	48816	0	18,835	0	6,81	-	-	0,068	336,72	0,008	441	0,028	5,06	-	-	-	-	-	-	0,2	3,19
wil100	100	273038	0	473,99	0	167,76	-	-	-	-	0,006	28680	0,041	176,28	0,02	560,1	0	3161,3	-	-	0,22	23,6

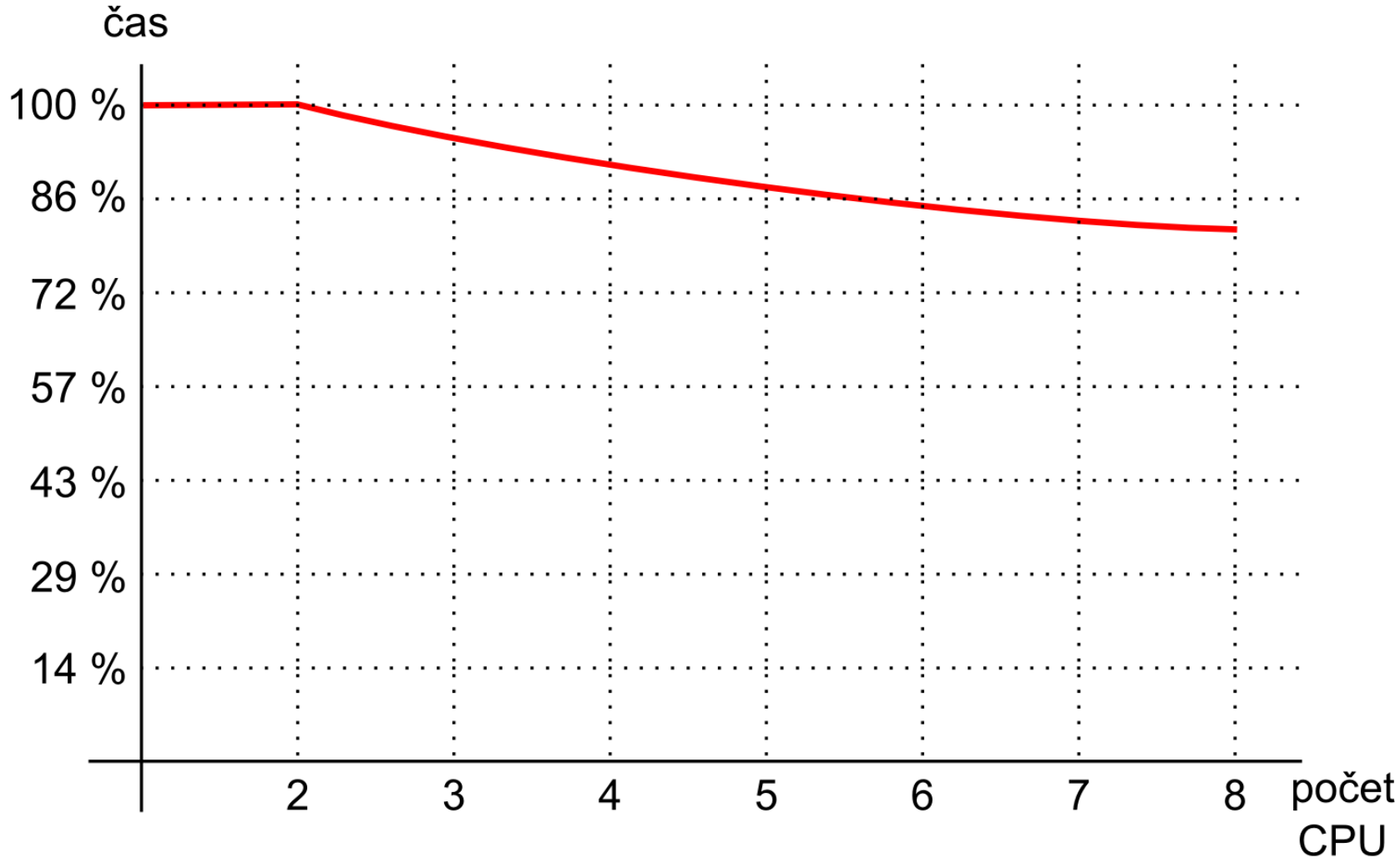
Výsledky hľadania

QAP – hľadanie rieš.



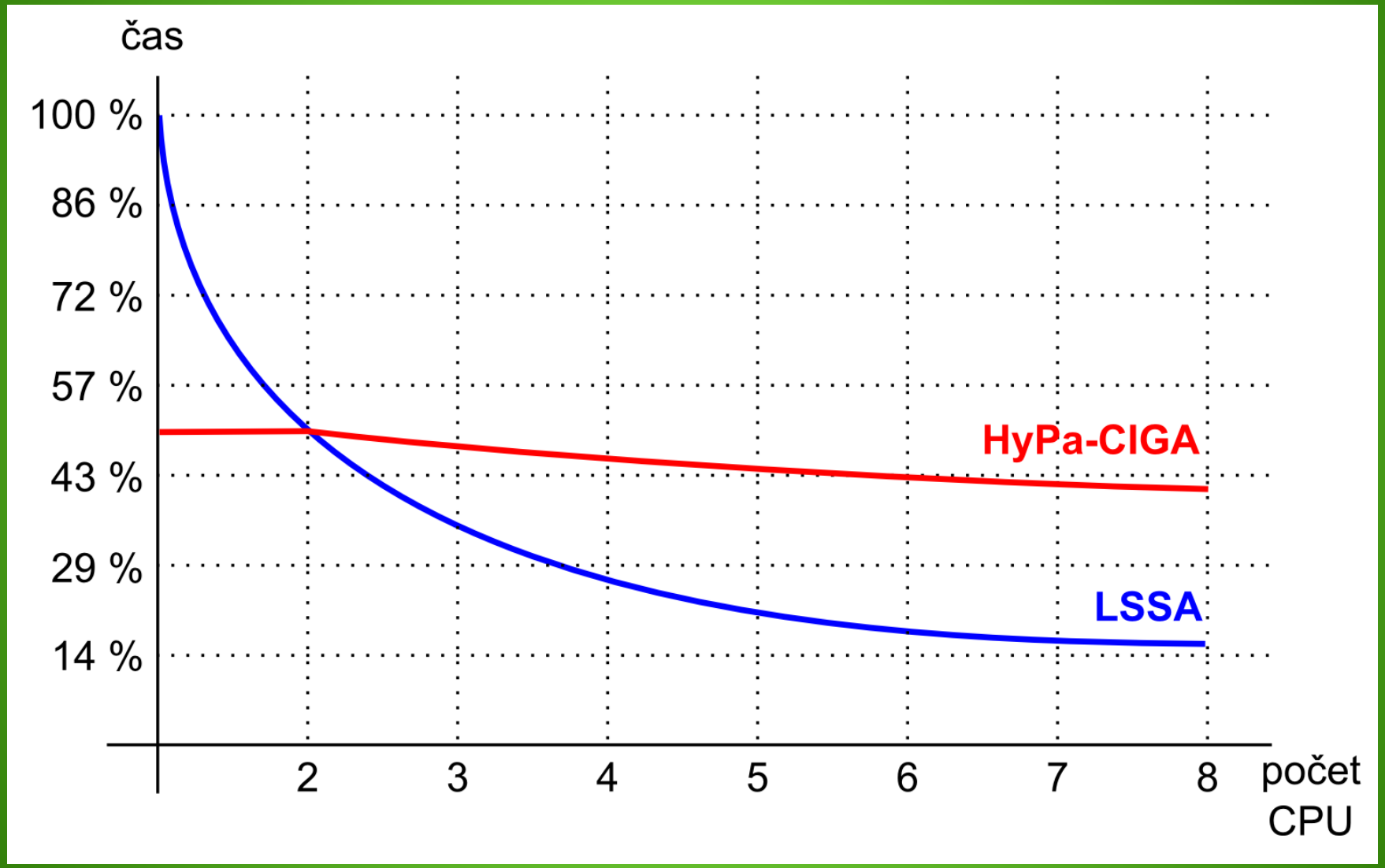
Výsledky hľadania

QAP – hľadanie rieš.



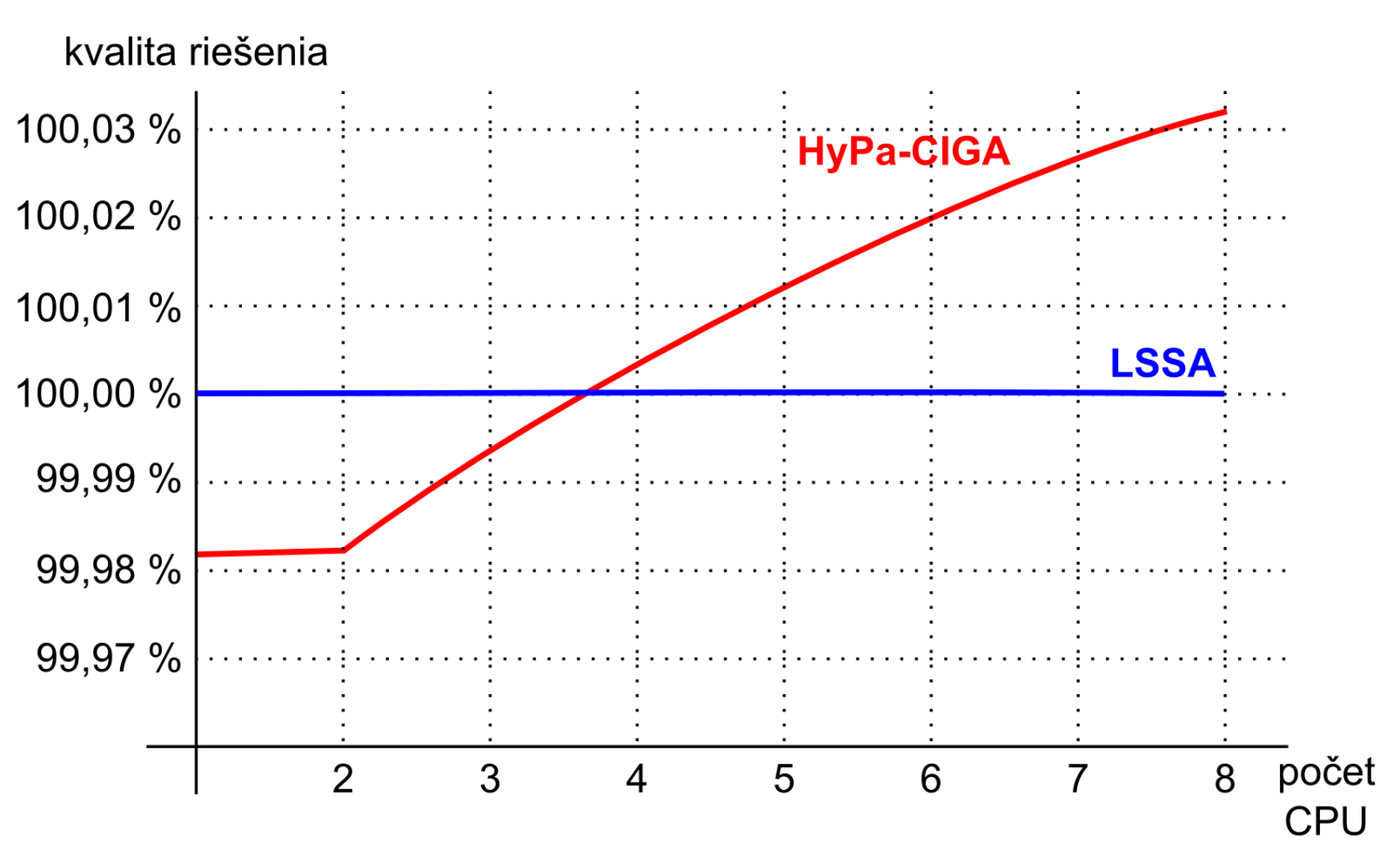
Výsledky hľadania

QAP – hľadanie rieš.



Výsledky hľadania

QAP – hľadanie rieš.



Ďalší prístup

QAP – hľadanie rieš.

$$\pi = \{1,2,\dots n\} \Rightarrow X = (x_{ij}),\ \text{kde } x_{ij} = \begin{cases} 1 & , \quad \pi_i = j \\ 0 & , \quad \pi_i \neq j \end{cases}$$

$$\min_{\pi \in S_n} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{\pi_i,\pi_j} b_{ij} \Rightarrow \min \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} a_{ij}\, b_{kl} x_{ik} x_{jl}$$

Za podmienok:

$$\sum_{i=1}^{n} x_{ij} = 1, \qquad 1 \leq \mathrm{j} \leq \mathrm{n}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \qquad 1 \leq \mathrm{i} \leq \mathrm{n}$$

$$x_{ij} \in \{0,1\}, \qquad 1 \leq \mathrm{i}, \mathrm{j} \leq \mathrm{n}$$

Úloha matematického nelineárneho binárneho programovania

Ďalší prístup

QAP – hľadanie rieš.

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} a_{ij}\, b_{kl} x_{ik} x_{jl}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\sum_{k=1}^{n}\sum_{l=1}^{n} a_{ij}\, b_{kl} x_{ik} x_{jl}\right) = \sum_{i=1}^{n}\sum_{j=1}^{n} \boldsymbol{h_{ij}(x)}$$

$\boldsymbol{h_{ij}(x)}$ – kvadratická forma, kladne definitná

⇒ Kuhn-Tuckerove podmienky sú postačujúce pre existenciu minima v $\boldsymbol{x}$

⇒ Druhá fáza Wolfeovej metódy dospeje ***po konečnom počte krokov k optimálnemu riešeniu***

Výsledky na porovnanie

QAP – hľadanie rieš.

[ghand] GHANDESHTANI, K. S. - MOLLAI, N. - SEYEDKASHI, S. M. H. - NESHATI, M. M.: New Simulated Annealing Algorithm for Quadratic Assignment Problem. In ADVCOMP 2010 : The Fourth International Conference on Advanced Engineering Computing and Applications in Sciences, Florence, Italy: IARIA, 25.-30.10.2010. ISBN: 978-1-61208-101-4, s. 87-92

[tang] TANG, J. - LIM, M. H. - ONG, Y. S. - ER, M. J.: Parallel Memetic Algorithm with Selective Local Search for Large Scale Quadratic Assignment Problems. In International Journal of Innovative Computing, Information and Control, vol. 2, no. 6., 2006, s. 1399-1416.

[talbi01] TALBI, E-G. - ROUX, O. - FONLUPT, C. - ROBILLARD, D.: Parallel Ant Colonies for the quadratic assignment problem. In: Future Generation Computer Systems, Vol. 17, Issue 8, s. 441-449. Elsevier, 2001. ISSN: 0167-739X.

[james2] JAMES, T. - REGO, C. - GLOVER, F.: A cooperative parallel tabu search algorithm for the quadratic assignment problem. In European Journal of Operational Research, vol. 195, 2009, s. 810-826.

[tsu] TSUTSUI, S.: Parallel Ant Colony Optimization for the Quadratic Assignment Problems with Symmetric Multi Processing. In ANTS ’08 Proceedings of the 6th international conference on Ant Colony Optimization and Swarm Intelligence, Springer-Verlag Berlin, Heidelberg: 2008, s. 363-370. ISBN: 978-3-540-87526-0.

[ahuja] AHUJA, R. K. - ORLIN, J. B. - TIWARI, A.: A greedy genetic algorithm for the quadratic assignment problem. In Computers & Operations Research, vol. 27, issue 10., Elsevier Science Ltd. Oxford, UK: september 2000, s. 917-934.

[mama] MAMAGHANI, A. S. - MEYBODI, M. R.: An Application of Imperialist Competitive Algorithm to Solve the Quadratic Assignment Problem. In 6th International Conference on Internet Technology and Secured Transactions, 11-14 December 2011, Abu Dhabi, United Arab Emirates, s. 562-565. ISBN 978-1-4577-0884-8.

[wu] WU, Y. - JI, P.: Solving the Quadratic Assignment Problems by a Genetic Algorithm with a New Replacement Strategy. In International Journal of Computational Intelligence, vol. 4, no. 3., s. 225-229, 2008

[mis] MISEVIČIUS, A.: A Modified Simulated Annealing Algorithm for the Quadratic Assignment Problem. In INFORMATICA, vol. 14, No. 4. Institute of Mathematics and Informatics, Vilnius: 2003. s. 497-514.

Ďakujem za pozornosť