

# Optimalizace rozmístění obdelníkových útvarů pomocí evolučních algoritmů

Vojtěch Hordějčuk

FEL ČVUT

6. června 2011



## Řešený problém

- ▶ optimální rozmístění obdélníků na plochu
- ▶ NP-těžký optimalizační problém
- ▶ hlavní motivace = **ušetřit**

## Využití v praxi

- ▶ VLSI floorplanning
- ▶ urbanistika
- ▶ plánování

1. zvolit vhodnou a efektivní **reprezentaci**
2. navrhnout evoluční **algoritmus**
3. provést **benchmarky** (GSRC, MCNC)
4. **výsledky** vyhodnotit a porovnat

## Možné reprezentace

- ▶ **Slicing Tree** – binární strom, slicing
- ▶ **sekvenční páry** – uspořádaná dvojice permutací, non-slicing
- ▶ **Corner Block List** – uspořádaná trojice, non-slicing
- ▶ **O-Tree** – obecný strom, non-slicing
- ▶ **B\*-Tree** – binární strom, non-slicing

## Používané optimalizační algoritmy

- ▶ deterministické (DFS, BFS)
- ▶ simulované ochlazování
- ▶ evoluční algoritmy

## B\*-Tree reprezentace + Algoritmus POEMS <sup>1</sup>

### B\*-Tree reprezentace

- ▶ binární strom
- ▶ levý potomek - umístit **vedle**
- ▶ pravý potomek - umístit **nad**

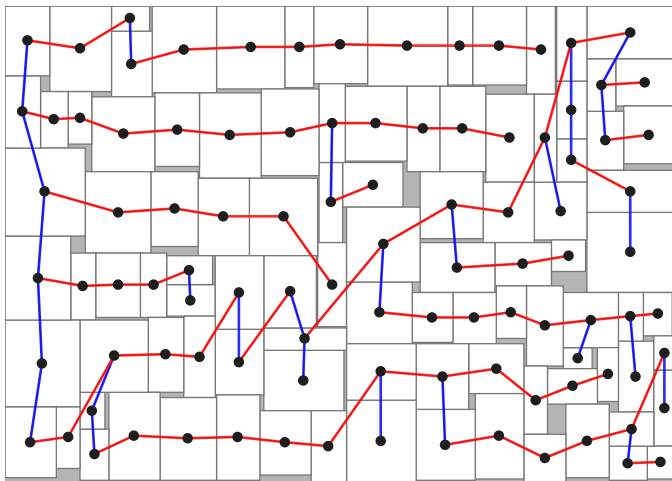
### Algoritmus POEMS

- ▶ počáteční řešení (prototyp)  $\leftarrow$  best-fit heuristika
- ▶ iterativní hledání nejlepší modifikace prototypu
- ▶ modifikace prototypu  $\leftarrow$  sekvence definovaných akcí
- ▶ optimalizace sekvencí  $\leftarrow$  genetický algoritmus

---

<sup>1</sup><http://code.google.com/p/vh-master-thesis/>

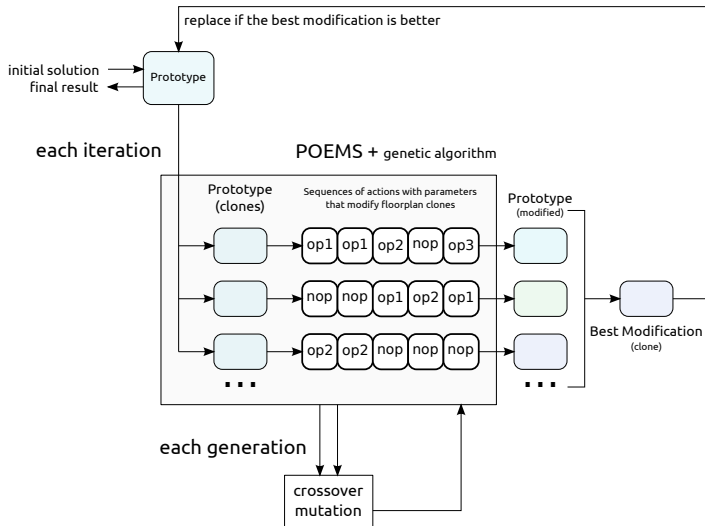
## B\*-Tree reprezentace (příklad) <sup>2</sup>



<sup>2</sup>kořen = vlevo nahoře, červená = levá, modrá = pravá

- ▶ iterativní algoritmus
- ▶ lokální prohledávání pomocí evolučního algoritmu
- ▶ počáteční řešení (prototyp) náhodně či heuristikou
- ▶ modifikace prototypu sekvencemi elementárních akcí
  1. **ROTATE**
  2. **MOVE NODE, MOVE VALUE, FLIP, MIRROR**
  3. **HANG NODE**
- ▶ evoluční algoritmus mění akce a parametry

# Algorithmus POEMS





## Benchmarky

- ▶ MCNC (9 – 49 bloků)
- ▶ GSRC (10 – stovky bloků)

## Algoritmy

- ▶ **B\*-Tree/POEMS** - navržený algoritmus, 50,000 iterací
- ▶ **CompaSS** - O-Tree, Slicing Tree, větve a hranice
- ▶ **B\*-Tree/I** - iterativní algoritmus
- ▶ **B\*-Tree/SA** - simulované ochlazování

# Experimenty: Výsledky

	B*-Tree/POEMS	CompaSS	B*-Tree/I	B*-Tree/SA
apte	0.78% (969 s)	0.78% (0 s)	<b>0.77%</b> (7 s)	1.59% (2 s)
xerox	<b>2.30%</b> (902 s)	3.45% (16 s)	2.48% (25 s)	3.85% (5 s)
hp	3.88% (1,927 s)	2.28% (3 s)	<b>1.35%</b> (55 s)	4.47% (20 s)
n100	<b>1.98%</b> (4,826 s)	7.32% (4 s)	N/A	N/A
n200	<b>3.68%</b> (8,408 s)	6.48% (10 s)	N/A	N/A

- ▶ průměrné hodnoty (10 a více běhů)
- ▶ **většinou lepší výsledky než konkurence**
- ▶ cenou za kvalitu je delší výpočet

- ▶ navzdory nedostatečnému popisu v literatuře navržena **efektivní a úsporná** reprezentace
- ▶ dosažené výsledky ve kvalitě zhruba **dvojnásobně překonávají** slicing reprezentaci
- ▶ naměřené výsledky jsou minimálně srovnatelné a **často lepší** než ostatní porovnávané algoritmy
- ▶ připravujeme **článek** do časopisu

