



Aplikovaná informatika

Podklady předmětu
Aplikovaná informatika
pro akademický rok 2013/2014
Radim Farana

10



Obsah

- Kombinatorické úlohy.
- Neheuristické algoritmy
 - prohledávání do šířky,
 - prohledávání do hloubky.
- Heuristické algoritmy.
- Pravděpodobnostní algoritmy.
- Genetické algoritmy.



Kombinatorické úlohy

- Úloha optimalizace – výběr nejlepší varianty závislé na hodnotách parametrů.
- Kvalitu řešení určuje hodnota ohodnocující (účelové) funkce.
 $f(p_1, p_2, \dots, p_n) \rightarrow \max (\min)$,
- Úlohy minimalizace a maximalizace jsou vzájemně převoditelné.



Kombinatorické úlohy

- Omezení třídy řešených problémů:
 - Omezení počtu parametrů.
 - Vzájemně nezávislé parametry.
 - Omezení hodnot parametrů na množinu hodnot (diskrétní parametry).
- Jednotlivá řešení jsou dána kombinací hodnot jednotlivých parametrů.
- Při nezávislosti parametrů je počet řešení:

$$r = \prod_{i=1}^n m_i \quad \text{kde je } r \text{ - počet možných řešení,}$$

$$m_i \text{ - počet možných hodnot parametru } p_i,$$

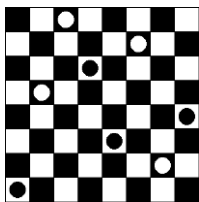
$$n \text{ - počet parametrů.}$$



Řešená úloha

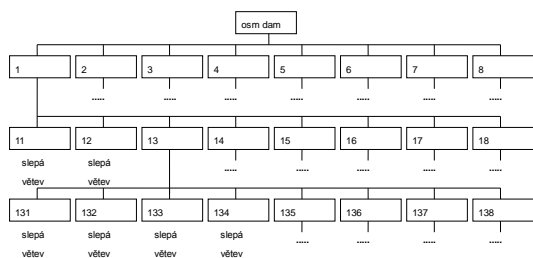
- Rozmístit osm dam na šachovnici aby se vzájemně neohrožovaly.
- Osm parametrů – pozice dámy v jednotlivých sloupcích
- Počet variant

$$r = \prod_{i=1}^8 8 = 8^8 = 16777216 \div 17 \cdot 10^6$$





Strom řešení





Neheuristické algoritmy

- Prohledávání do šířky
 - Postupně rozvíjíme jednotlivé úrovně stromu.
 - Vhodné pokud řešení leží blízko kořene.
 - Nebezpečí vyčerpání paměti.
- Prohledávání do hloubky
 - Postupně rozvíjíme zvolenou větev.
 - Pokud není další úroveň, vrací se zpět.
 - Nebezpečí uvíznutí ve slepé větvi.
- Prohledávání s omezenou hloubkou vnoření



Realizace prohledávání do hloubky

- Datová struktura



- Algoritmus řešení

```

počáteční nastavení
while není nalezeno řešení
  rozviň všechny nerozvinuté prvky a označ je jako rozvinuté
  spočti ohodnocující funkci všech nových prvků
end while
vyhlás nalezené řešení
  
```



Vlastnosti prohledávání do šířky

- Výhody:
 - dobrá rychlost,
 - nehrozí nebezpečí nadměrného vnoření do stromu.
- Nevýhody:
 - velké nároky na paměť,
 - složitější manipulace s datovou strukturou



Vlastnosti prohledávání do hloubky

- Výhody:
 - postupujeme rychle do hloubky,
 - menší nároky na paměť.
- Nevýhody:
 - pokud se řešení nachází na opačné straně stromu budeme ho hledat dlouho ⇒ **náhodná volba následníka**,
 - Pro neznámou úroveň vnošení řešení můžeme postupovat do velkých hloubek bez úspěchu ⇒ **prohledávání s omezenou hloubkou vnošení**.



Vyzkoušení všech kombinací

- Postupně zkusíme všechny kombinace parametrů.
- Postupujeme ve stromu systematicky zleva doprava při určené hloubce vnošení.

```
nastavení počátečního rozložení dam [1, 1, 1, 1, 1, 1, 1, 1]
while nepřeteklo první počítadlo
  if ohodnocující funkce aktuálního řešení = 8 then
    vyhlas řešení
  end if
  generuj další rozložení dam
end while
```



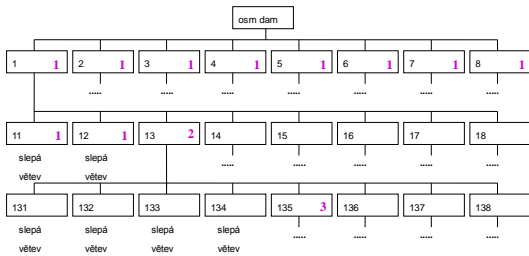
Heuristické algoritmy

- Prohledávání do hloubky s uvážením hodnoty heuristiky.
- Heuristika – počet správně umístěných dam

```
počáteční nastavení
while není nalezeno řešení
  najdi první dosud nerozvinutý prvek
  s maximální hodnotou ohodnocující funkce
  rozviň nalezený prvek a označ ho jako rozvinutý
  spočti ohodnocující funkci všech nových prvků
end while
vyhlas nalezené řešení
```



Heuristické prohledávání





Vyzkoušení všech slibných kombinací

- Využití definované ohodnocující funkce a pole počítadel (hodnot parametrů)

```

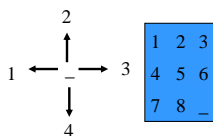
vynulování pole počítadel
i = 1
while nepřetečko první počítadlo
  pam[i] = pam[i] + 1
  if pam[i] > 8 then
    pam[i] = 0
    i = i - 1
  else
    if f(pam[1], ... pam[i]) = i then
      if i = 8 vyhlas řešení
    else
      i = i + 1
    end if
  end if
end if
end while

```



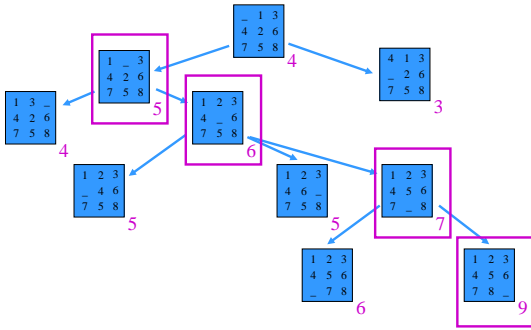
Příklad – řešení hlavolamu „osmička“

- Zjednodušená varianta „patnáctky“.
- Síť 3 x 3 pole, osm kamenů 1 ÷ 8.
- Cílem je uspořádat kameny v určeném pořadí.
- Pro popis stavu sledujeme pohyb mezery.
- Heuristiky:
 - Počet umístěných kamenů
 - Součet vzdáleností kamenů od jejich správných pozic
- Nebezpečí vzniku cyklů.





Příklad – řešení hlavolamu „osmička“





Pravděpodobnostní algoritmy

- Metoda pokusu a omylu.
 - Náhodné zkoušení variant.
 - Malá efektivita.
 - Bez paměti variant není zaručen úspěch.
- Genetické algoritmy
 - „Řízení náhody“.
 - Podobnost s přírodními procesy náhodného výběru.



Metoda pokusu a omylu

- Pravděpodobnost úspěchu náhodného pokusu:

$$P(1) = \frac{92}{16777216} \doteq 5,48 \cdot 10^{-6}$$

- Pravděpodobnost neúspěchu:

$$\bar{P}(1) = 1 - P(1)$$

- Pravděpodobnost neúspěchu v n pokusech:

$$\bar{P}(n) = \prod_{i=1}^n (1 - P(1)) = [1 - P(1)]^n$$



Metoda pokusu a omylu

- Pravděpodobnost nalezení nějakého řešení:

$$P(n) = 1 - [1 - P(1)]^n$$

- Počet pokusů pro nalezení nějakého řešení s pravděpodobností $P(n)$:

$$n = \frac{\log[1 - P(n)]}{\log[1 - P(1)]}$$

- Např. pro $P(n) = 0,1$ (10 %) je $n = 19\ 214$.



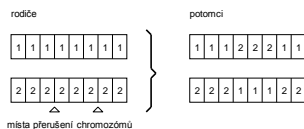
Genetický algoritmus

- Vlastnosti řešení (parametry) jsou jednotlivé **geny**, které tvoří **chromozóm**.
- Chromozóm popisuje **jedince** a současně řešení.
- Tito jedinci tvoří **populaci**.
- Jedinci populace vytvářejí **potomky**, jedince následující **generace**.
- Potomci přebírají vlastnosti svých **rodičů**.
- Nejlepší jedinci přežívají (nejvíce se množí), nejhorší hynou.



Základní genetické operace

- Reprodukce:** potomek přebírá části chromozómů svých rodičů.



- Volba počtu míst přerušení.
- Obvykle vytváříme oba možné potomky.



Základní genetické operace

- **Mutace:** náhodné změny hodnot genů.
- Všichni jedinci tvořící populaci jsou ohodnoceni pomocí **ohodnocující funkce**.
- Nejlepší jedinci se reprodukuje s pravděpodobností odpovídající jejich kvalitě.
- Někdy nejlepší jedinci přecházejí do nové populace.



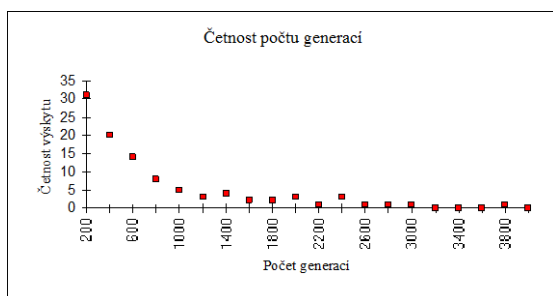
Příklad

- Chromozóm s osmi geny s hodnotami $1 \div 8$.
- Velikost populace 100 jedinců.
- Dvě místa přerušení.
- Nejvýše tři mutace v rozsahu ± 3 .

- Výsledky testování ze 100 průběhů
- Střední počet generací: 685
- Rozptyl: 742

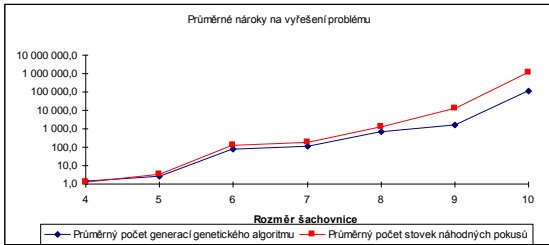


Příklad





Příklad





Příklad – úloha optimalizace

- Hledáme minimum funkce $f(x, y)$
 $(x - 50)^2 + (y - 35)^2$
- pro $x, y \in (0, 100)$; $x, y \in \mathbb{Z}$.
- Chromozom definujeme se čtyřmi geny

desítky	jednotky	desítky	jednotky
x	x	y	y



Příklad – úloha optimalizace

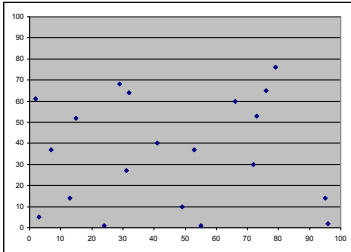
- První generace – náhodně vygenerované chromozómy.

	x-jedn.	x-desit.	y-jedn.	y-desit.	x	y	f(x,y)
1	3	7	3	1	73	13	1013
2	2	5	6	0	52	6	845
3	1	3	6	3	31	36	302
4	6	1	3	2	16	23	1300
5	9	5	7	0	59	7	865
6	5	0	2	0	5	2	3114
7	2	6	9	0	62	9	820
8	6	4	9	5	46	59	592
9	5	2	2	5	25	52	914
10	7	9	1	6	97	61	2885
11	8	9	7	4	98	47	2448
12	4	1	7	1	14	17	1620
13	1	2	5	9	21	95	4441
14	8	6	0	2	68	20	549
15	9	8	8	6	89	68	2510
16	3	6	5	6	63	65	1069
17	5	6	7	3	65	37	229
18	1	3	7	2	31	27	425
19	4	9	3	6	94	63	2720
20	7	2	5	6	27	65	1429



Příklad – úloha optimalizace

- Rozmístění jedinců první generace.





Příklad – úloha optimalizace

- Výběr nejlepších jedinců pro reprodukci.

	x-jedn.	x-desit.	y-jedn.	y-desit.	x	y	f(x,y)
1	3	5	7	3	53	37	13
2	1	4	0	4	41	40	106
3	1	3	7	2	31	27	425
4	2	7	0	3	72	30	509
5	9	4	0	1	49	10	525
6	3	7	3	5	73	53	853
7	6	6	0	6	66	60	881
1	2	3	4	6	32	64	1165
5	6	6	1	0	55	1	1181
19	5	1	2	5	15	52	1514
18	9	2	8	6	29	68	1530
16	6	7	5	6	75	65	1576
17	3	1	4	1	13	14	1810
15	4	2	1	0	24	1	1832
12	7	0	7	3	7	37	1853
10	5	9	4	1	95	14	2465
20	9	7	6	7	79	76	2522
14	2	0	1	6	2	61	2980
4	3	0	5	0	3	5	3109
3	6	9	2	0	96	2	3205



Příklad – úloha optimalizace

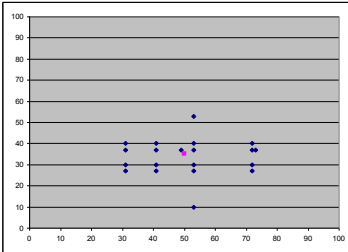
- Druhá generace.

	x-jedn.	x-desit.	y-jedn.	y-desit.	x	y	f(x,y)
1+2	3	5	0	4	53	40	34
1+2	1	4	7	3	41	37	85
1+3	3	5	7	2	53	27	73
1+3	1	3	7	3	31	37	365
1+4	3	5	0	3	53	30	34
1+4	2	7	7	3	72	37	488
2+3	1	4	7	2	41	27	145
2+3	1	3	0	4	31	40	386
2+4	1	4	0	3	41	30	106
2+4	2	7	0	4	72	40	509
3+4	1	3	0	3	31	30	386
3+4	2	7	7	2	72	27	548
1+5	3	5	0	1	53	10	634
1+5	9	4	7	3	49	37	5
1+6	3	5	3	5	53	53	333
1+6	3	7	7	3	73	37	533
1	3	5	7	3	53	37	13
2	1	4	0	4	41	40	106
3	1	3	7	2	31	27	425
4	2	7	0	3	72	30	509



Příklad – úloha optimalizace

- Rozmístění jedinců druhé generace.





Příklad – úloha optimalizace

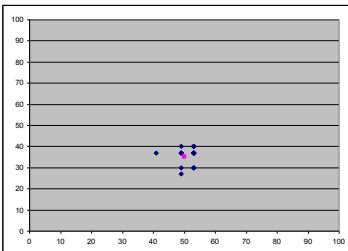
- Třetí generace.

	x-jedn.	x-desit.	y-jedn.	y-desit.	x	y	f(x,y)	
1+2	9	4	7	3	49	37	5	
1+2	3	5	7	3	53	37	13	
1+3	9	4	0	4	49	40	26	
1+3	3	5	7	3	53	37	13	
1+4	9	4	0	3	49	30	26	
1+4	3	5	7	3	53	37	13	
2+3	3	5	0	4	53	40	34	
2+3	3	5	7	3	53	37	13	
2+4	3	5	0	3	53	30	34	
2+4	3	5	7	3	53	37	13	
3+4	3	5	0	3	53	30	34	
3+4	3	5	0	4	53	40	34	
1+5	9	4	7	2	49	27	65	
1+5	3	5	7	3	53	37	13	
1+6	9	4	7	3	49	37	5	
1+6	1	4	7	3	41	37	85	
2	1	9	4	7	3	49	37	5
2	3	3	5	7	3	53	37	13
3	3	3	5	0	4	53	40	34
4	3	3	5	0	3	53	30	34



Příklad – úloha optimalizace

- Rozmístění jedinců třetí generace.





Příklad – úloha optimalizace

- Výsledky:
 - jedinci se postupně blíží k optimální hodnotě,
 - optimální hodnoty nemusí být nikdy dosaženo,
 - po **rozumném** počtu generací je dosaženo **dobrého** řešení (definice obou pojmů je problematická).
- Nebezpečí:
 - pokud má úloha více lokálních minim, hrozí nebezpečí uvíznutí v lokálním minimu.
