# Aplikované evoluční algoritmy 19 - Evoluční řešení sudoku

xheczk<br/>04, Heczková Petra 29.4.2016

## 1 Úvod

V jazyce C++ jsem navrhla evoluční algoritmus řešící hru Sudoku pro zadanou velikost pole. V projektu jsem využila knihovnu GALib, která usnadňuje implementaci genetických algoritmů.

#### 1.1 Popis úlohy

Sudoku je hra zaměřená na logické myšlení. Tvoří ji čtverec o velikosti N sloupců a N řádek (číslo N musí být beze zbytku odmocnitelné 2, nejčastěji N=9). Je rozdělen na N menších čtverců. Součástí zadání je i několik předvyplněných číslic. Pro sudoku o velikosti 9x9 je minimální počet zadaných číslic 17. V takovém případě je řešení jednoznačné, ale nemusí být luštitelem nalezitelné.

Úkolem luštitele je doplnit číslice 1 až N do ostaních políček tak, aby se neopakovaly ve sloupci, řádku ani menším čtverci. Obtížnost sudoku je dána počtem předvyplněných číslic a vazbami mezi nimi.

Počet možných sudoku o velikosti 9x9 vzhledem k pravidlům a omezením je přibližně  $6,671\cdot 10^{21}.$ 

## 2 Popis evolučního algoritmu

V programu je použit evoluční algortimus s operátory křížení a mutace. Algoritmus skončí po nalezení řešení s fitness funkcí 0 nebo po určitém počtu generací.

Program se skládá ze 4 funkcí. Ve funkci main se nachází zpracování vstupních parametrů programu, nastavení evolučního algoritmu a nalezení minima fitness funkce. Vstupem všech ostatních funkcí je instance třídy GAGenome.

Funkce InitializerFce vytvoří 1D pole genomů o zadaném počtu a provede počáteční mutace. Tak se vytvoří první populace pro další zpraco-

vání.

Funkce FitnessFce, která je popsána níže, spočítá fitness pro jednotlivé chody evolučního algoritmu a funkce PrintPhenotype vytiskne na standartní výstup výsledné sudoku s doplněnými číslicemi.

#### 2.1 Vstupní parametry programu

Program musí mít minimálně jeden parametr, který určuje velikost sloupců a řádek čtverce. Je ve formátu -d=number. Dalšímí parametry jsou předvyplněná čísla ve formátu -number=x, y, kde x je souřadnice sloupce a y řádku.

#### 2.2 Zakódování problému

Jednotlivé genomy mají vlastně 3 veličiny - pozici ve sloupci, pozici v řádce a číslici. Genomy třídy GA1DArrayGenome mají pouze 2 parametry. První parametr S udává souřadnice a druhý C číslici. Hodnota sloupce se spočítá pomocí modula a hodnota řádky pomocí podílu.

Vzorce pro výpočet:

$$column = S \bmod N \tag{1}$$

$$row = \frac{N \cdot N}{S} \tag{2}$$

Příklad pro matici 3x3: vlevo je rozmístění genomů reprezentováno pomocí 2D pole a vpravo je výsledné sudoku, které se přepočítá a zobrazí funkcí PrintPhenotype.

	Χ								Χ			I	I					Χ			1		3		2	1
					Х		Х			1						Х			1	1	2		1		3	1
			X								X	   	   	X	 						3		2		1	

### 2.3 Fitness funkce

Fitness funkce projde všechny genomy z jedné populace a spočítá celkovou chybu. Pokud je nalezen další genom se stejnou hodnotou C v určitých polích reprezentující stejný sloupec nebo stejnou řádku. Pokud jsou nalezeny připočítává

se chyba č. 1.

Dále se kontroluje, zda číslice zadané uživatelem, jsou na správných místech. Pokud ne, připočítá se chyba č. 2, která je několikanásobně větší než chyba č. 1.

Příklad pro matici 3x3: Pro políčko obsahující číslici 2 se kontrolují políčka označená jako "X".

									 	1   3   2
X			2	X	X	X			 	2   1   3
									   	3   2   1

### 3 Experimentální výsledky a jejich vyhodnocení

#### 3.1 Parametry evolučního algoritmu

První experiment byl proveden pro nalezení vhodných parametrů evolučního algoritmu. Využívá se sudoku o velikosti sloupce a řádku N=9 bez předvyplněných čísel. Pro každé parametry bylo provedeno 50 pokusů. Největší možná hodnota fitness v tomto případě bude  $9\cdot 9\cdot 8\cdot 8=5184$ , protože u každého z  $N\cdot N$  políček musíme kontrolovat (N-1) políček ve sloupci a (N-1) políček v řádce.

Sloupce v tabulce označené jako P.F. značí průměrnou hodnotu fitness funkce za 50 běhů pokusu. Sloupce Max. F., případně Min. F. obsahují maximální, případně minimální naměřenou hodnotu fitness funkce. Následující 3 sloupce jsou průměrný, maximální a minimální čas trvání 1 běhu v sekundách. v prvním sloupci jsou pak parametry velikost populace, počet generací, pravdě-podobnost mutace a pravděpodobnost křížení. Zvolené hodnoty parametrů pro experiment 2 jsou označeny šedou barvou.

Vel. pop.	P.F.	Max. F.	Min. F	P. Č.	Max. Č.	Min. Č.
100	74,67	80	68	15,83	16	15
200	72,27	76	68	31,87	33	31
300	71,73	76	64	47,67	49	47
400	71,73	76	66	64,53	66	63
500	71,73	74	68	81,83	84	80

Poč. gen.	P.F.	Max. F.	Min. F	P. Č.	Max. Č.	Min. Č.
2000	73,87	78	68	13	14	12
3000	73,53	76	66	19,33	21	18
4000	73,4	78	68	25,93	27	25
5000	72,93	76	68	32,63	34	32
6000	72,07	76	64	38,9	40	38
Prav. mut.	P.F.	Max. F.	Min. F	P. Č.	Max. Č.	Min. Č.
0.3	72,4	76	66	32	33	31
0.4	73,4	76	70	31,57	33	31
0.5	73,33	76	68	31,93	33	31
0.6	72,13	77	64	33,47	35	32
0.7	72,27	76	68	34,97	36	33
Prav. kříž.	P.F.	Max. F.	Min. F	P. Č.	Max. Č.	Min. Č.
0.5	72,93	76	66	33,37	34	32
0.6	73,07	78	66	32,53	34	31
0.7	73,47	78	68	33,13	34	32
0.8	72,8	76	64	34	35	33
0.9	72,53	76	66	34,67	36	33

#### 3.2 Experiment 2

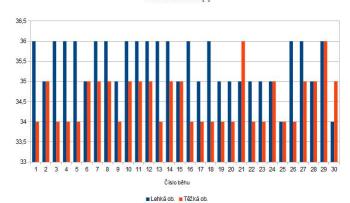
V druhém experimentu jsem se zaměřila na řešení sudoku s předvolenými čísly. Zkoumala jsem jaký počet předvolených čísel má nejlepší výsledky. Lehká obtížnost má 37 číslic z 81 a těžká 26 číslic. Každé sudoku bylo zpracováno 30krát. Velikost populace u jednotlivých běhů byla 300 genů, maximální počet generací byl 5000.

	Lehká obtížnost										
Č.b.	Fit.	Čas [s]	Č.b.	Fit.	Čas [s]	Č.b.	Fit.	Čas [s]			
1	900	36	11	856	36	21	864	35			
2	850	35	12	872	36	22	896	35			
3	860	36	13	908	36	23	864	35			
4	900	36	14	882	36	24	954	35			
5	914	36	15	924	35	25	932	34			
6	866	35	16	816	36	26	902	36			
7	774	36	17	880	35	27	888	36			
8	868	36	18	890	36	28	866	35			
9	908	35	19	874	35	29	932	36			
10	954	36	20	918	35	30	808	34			

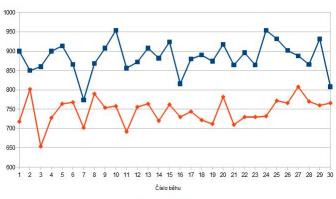
V tabulkách je Č.b.číslo běhu, Fit je hodnota fitness funkce, která není standartizovaná. Je obecně vyšší než v prvním experimentu, protože pokuta za změnu číslice v políčku zadaném uživatel je několikanásobně větší než pokuta za porušení jedinečnosti číslice v sloupci nebo řádce.

	Těžká obtížnost									
Č.b.	Fit.	Čas [s]	Č.b.	Fit.	Čas [s]	Č.b.	Fit.	Čas [s]		
1	718	34	11	692	35	21	710	36		
2	802	35	12	756	35	22	730	34		
3	654	34	13	764	34	23	730	34		
4	728	34	14	720	35	24	732	35		
5	764	34	15	762	35	25	772	34		
6	768	35	16	730	34	26	766	34		
7	702	35	17	744	34	27	808	35		
8	790	35	18	722	34	28	770	35		
9	754	34	19	712	34	29	760	36		
10	758	35	20	782	34	30	766	35		

#### Časová náročnost [s]



# Hodnota fitness funkce

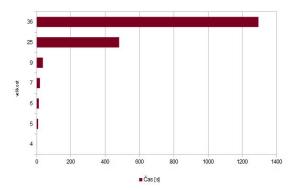


◆Lehká ob. ◆Těžká ob.

#### 3.3 Experiment 3

V experimentu 3 jsem se zaměřila na trvání programu při různých velikostech sudoku. Při velikostech N=25 je čas zpracování kolem 8 minut a N=36 je to téměř 22 minut.

37 1.1 / NT	Č []
Velikost N	Cas [s]
4	0,53
5	8,6
6	12,77
7	19,03
9	$36,\!25$
25	481,2
36	1293,67



## 4 Vyhodnocení

Při provádění experimentů jsem zjistila, že maximální počet generací a velikost populace ovlivňuje nejvýrazněji čas. Lepších výsledků by se dalo dosáhnout zlepšením kódování experimentu nebo jiným zpracováním předvolených čísel.

Z druhého experimentu vyplývá, že sudoku označené jako těžké má lepší fitness, protože obecně obsahuje méně předvolených číslic. Druhý experiment byl také časově náročnější než první, protože se musela navíc provádět kontrola předvolených číslic.

### 5 Závěr

Projekt byl vypracován a testován na serveru *merlin*. Odevzdaný archiv obsahuje i soubor Makefile, který příkazem make převede program do spustitelného souboru a obsahuje i 2 příklady na použití programu, které se spustí příkazy make easy nebo make diff.

### Literatura

http://www.sudokuonline.wz.cz/jaklustit.php

http://sudokuonline.cz/