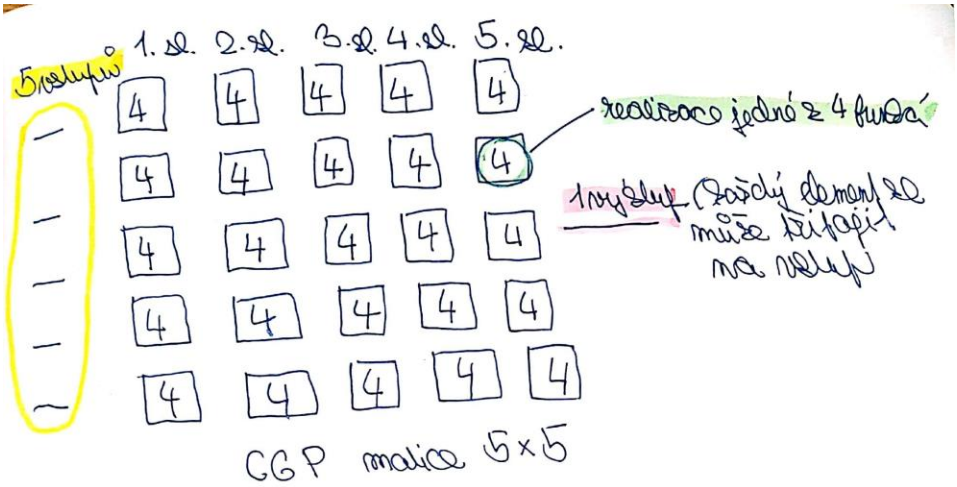
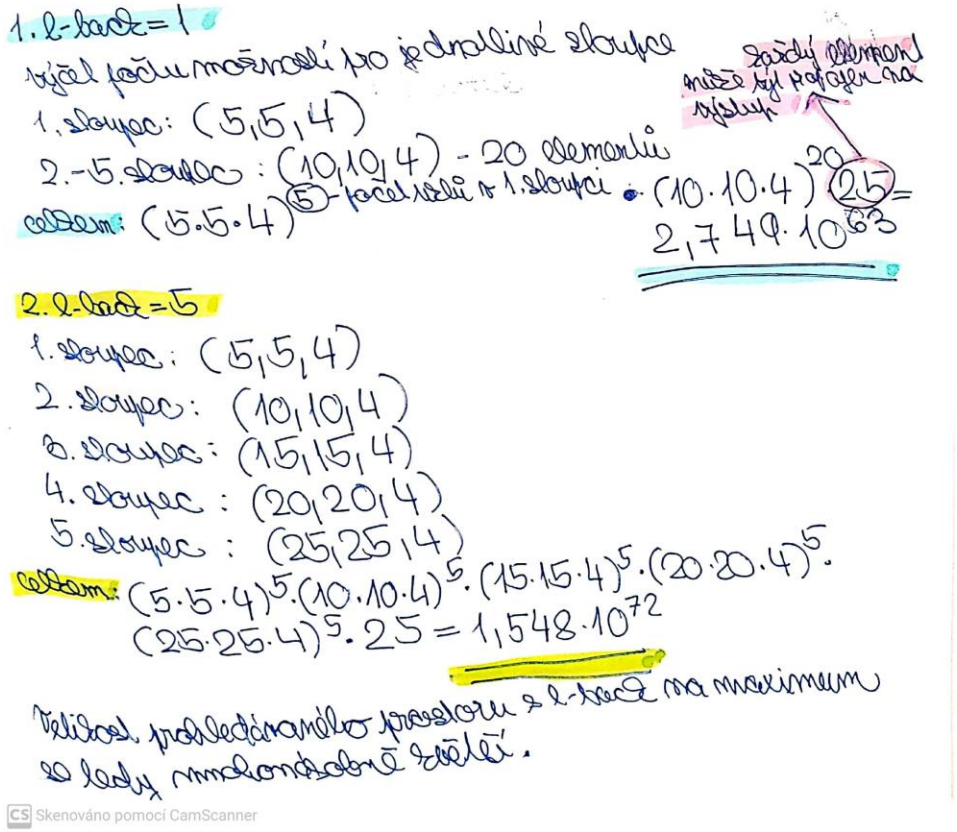
První cvičení z předmětu BIN - Kartézské genetické programování

Jméno: Kateřina Fořtová (xforto00)

První úkol: Určení velikosti prohledávaného prostoru





Skenováno pomocí CamScanner

Druhý úkol: Návrh kombinačního obvodu

Vybrala jsem si simulaci 3-bitové sčítačky (https://www.geeksforgeeks.org/full-adder-in-digital-logic/)

Experimentování s nastavením parametru l-back - pro 10 běhů:

Fixní nastavení:

```
#define POPULACE_MAX 5      //maximalni pocet jedincu populace
#define MUTACE_MAX 3        //max pocet genu, ktery se muze zmutovat behem jedne mutace
#define PARAM_M 5           //pocet sloupcu
#define PARAM_N 5           //pocet radku
#define PARAM_GENERATIONS 100000   //max. pocet generaci evoluce
```

#define PARAM_RUNS 10 //max. pocet behu evoluce

#define FUNCTIONS 4 //max. pocet pouzitych funkci bloku (viz fitness())

L-back	Nejmenší počet užitých elementů	Nejmenší počet generací pro nalezení řešení	Nejhorší počet generací pro nalezení řešení	Průměrný počet generací nalezený pro řešení
1	5	62	2378	967
2	5	98	950	443
3	5	142	1571	815
4	5	413	1969	1140
5	5	65	2734	903

Experimentování s nastavením rozměru matice - pro 10 běhů:

Fixní nastavení:

#define POPULACE_MAX 10 //maximalni pocet jedincu populace

#define MUTACE_MAX 5 //max pocet genu, ktery se muze zmutovat behem jedne mutace

#define L_BACK 5 //1 (pouze predchozi sloupec) .. param_m (maximalni mozny rozsah);

#define PARAM_GENERATIONS 500000 //max. pocet generaci evoluce

#define PARAM_RUNS 10 //max. pocet behu evoluce

#define FUNCTIONS 4 //max. pocet pouzitych funkci bloku (viz fitness())

(Počet sloupců, počet řádků)	Nejmenší počet užitých elementů	Nejmenší počet generací pro nalezení řešení	Nejhorší počet generací pro nalezení řešení	Průměrný počet generací nalezený pro řešení
(5, 5)	5	69	4383	621
(25, 1)	5	64	2290	446
(7, 7)	5	18	1015	379
(10, 10)	5	80	1007	388
(36, 1)	5	97	1714	555

Experimentování s nastavením použitých funkcí - pro 10 běhů:

Fixní nastavení:

#define POPULACE_MAX 5 //maximalni pocet jedincu populace

#define MUTACE_MAX 3 //max pocet genu, ktery se muze zmutovat behem jedne mutace

#define PARAM_M 5 //pocet sloupcu

#define PARAM_N 5 //pocet radku

#define L_BACK 1 //1 (pouze predchozi sloupec) .. param_m (maximalni mozny rozsah);

#define PARAM_GENERATIONS 100000 //max. pocet generaci evoluce

#define PARAM_RUNS 10 //max. pocet behu evoluce

Počet užitých funkcí	Nejmenší počet užitých elementů	Nejmenší počet generací pro nalezení řešení	Nejhorší počet generací pro nalezení řešení	Průměrný počet generací nalezený pro řešení
4	5	57	2610	1085
5	5	1136	104944	12756
6	5	295	17817	4988

Zhodnocení experimentů: U každé vlastnosti bylo provedeno 10 po sobě jdoucích měření. S využitím parametru l-back se nám zvětšuje prohledávaný prostor dané problematiky. Průměrně však nejnižšího počtu generací dosahovalo řešení s nastavením l-back parametru na číslo 2. Při experimentování s nastavením výšky a šířky matice bylo potřeba upravit několik parametrů, zejména zvětšit počet generací. Využití mřížky o rozměrech (25, 1) průměrně zmenšilo počet potřebných generací k vygenerování prvního správného řešení o 175 generací na rozdíl od původní velikosti (5, 5). Při analýze využitých funkcí bylo zjištěno, že přidání dalších funkčních bloků výpočet spíše zatěžuje, než pokud pracujeme s prvními 4 základními funkcemi (propojka, AND, OR, XOR). Nutno podotknout, že metriky nejmenšího/nejhoršího/průměrného počtu generací se odvíjí nikoliv od nejlepšího nalezeného řešení, ale od prvního správného řešení (které ještě nemusí mít minimální počet bloků).