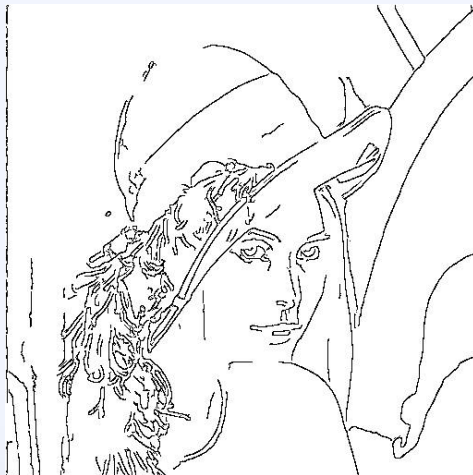


Autor: Petr Dvořáček

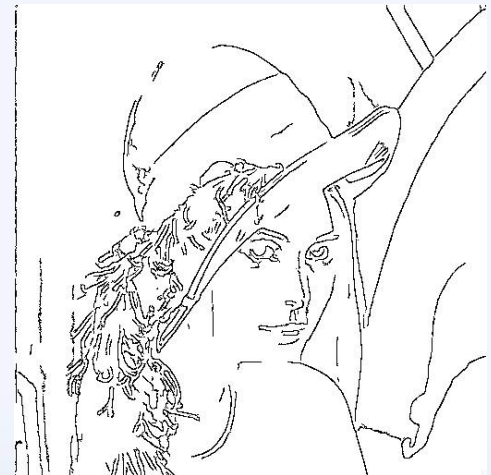
Vedoucí práce: Lukáš Sekanina

Evoluční návrh pro aproximaci obvodů

Když $3*3 = 9$



Když $3*3 = 8$



Návrh obvodů

Snaha minimalizace

- plochy
- zpoždění
- příkonu

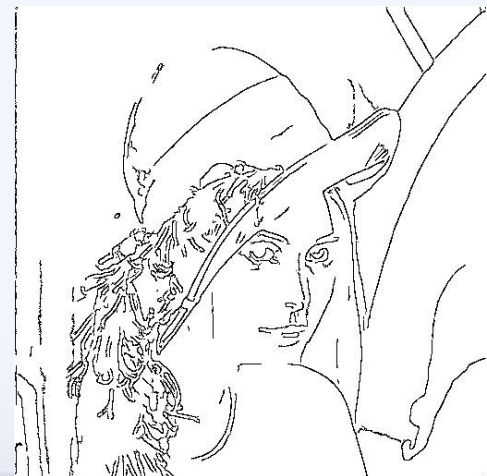
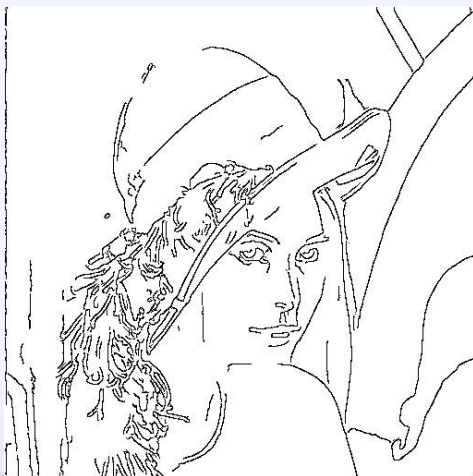
⇒ **Možnost aproximace obvodu.** + minimalizace chyby.

Proč aproximovat obvody?

- aproximace je přirozená

$$17 \cdot 49 > 1$$

$$17 \cdot 49 > 660$$



Metody aproximace obvodů

Modifikace elektrických parametrů obvodu

- Změna pracovní frekvence, napájecího napětí.

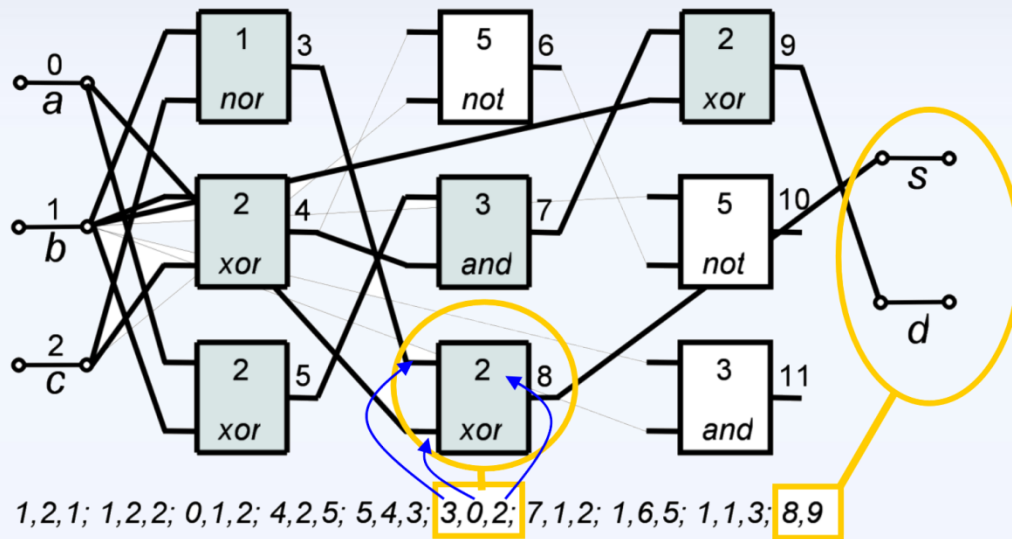
Funkční aproximace

- Odebrání hradla z obvodu.

Metody automatické funkční aproximace

- SALSA (2012)
- SASIMI (2013)
- ABACUS (2014)
- ...
- **Evoluční algoritmy**
 - Kartézské genetické programování

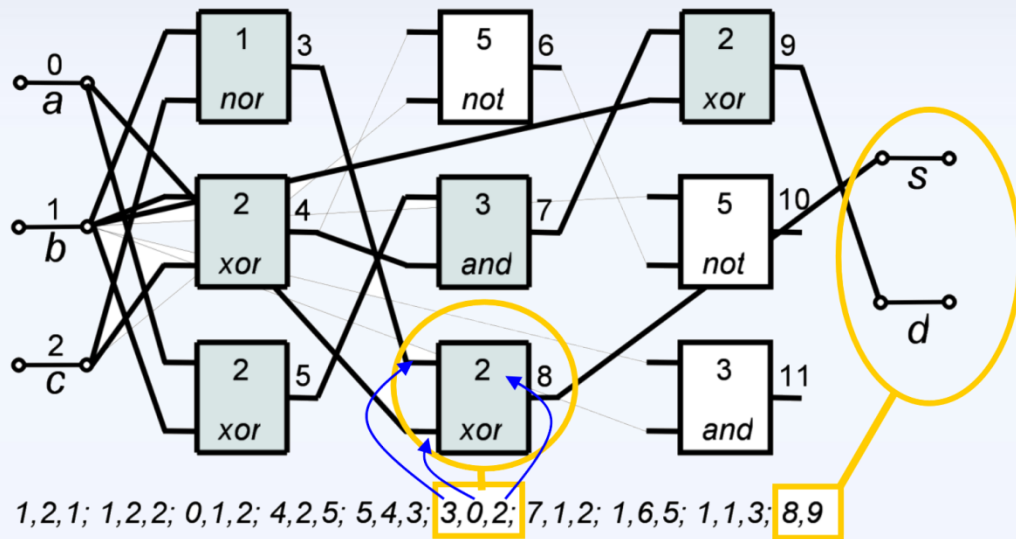
Kartézské genetické programování



Přirozená metoda

- Vychází z Darwinovy teorie evoluce.
- Slabší jedinci umírají a ti silnější přežívají a rozmnožují se.
- Snaha o zmenšení chybovosti obvodu \Rightarrow potřeba simulace.

Kartézské genetické programování



Přirozená metoda

- Vychází z Darwinovy teorie evoluce.
- Slabší jedinci umírají a ti silnější přežívají a rozmnožují se.
- Snaha o zmenšení chybovosti obvodu \Rightarrow potřeba simulace.

S rostoucím počtem vstupů roste doba simulace obvodu exponenciálně \rightarrow potřeba akceleračních technik

- Paralelní simulace (využití 64 bitových registrů)
- Předkompilace simulace
- Heuristická inicializace CGP obvodem

Ohodnocení přibližného obvodu v CGP

Fitness funkce

$$fit = \sum_{i=1}^v \overset{\substack{\text{Výstup CGP} \\ \downarrow}}{|y(i) - \underset{\substack{\uparrow \\ \text{Referenční výstup}}}{t(i)}}|$$

Princip výpočtu

- Bitová transpozice celočíselných hodnot (A0)

Referenční
výstup

0	000
1	001
2	010
3	011
4	100

Výstup
CGP

001	1
001	1
001	1
001	1
110	6

$y(i) - t(i)$

1
0
-1
-2
2

$\text{abs}(x)$

1
0
1
2
2

Ohodnocení přibližného obvodu v CGP

Fitness funkce

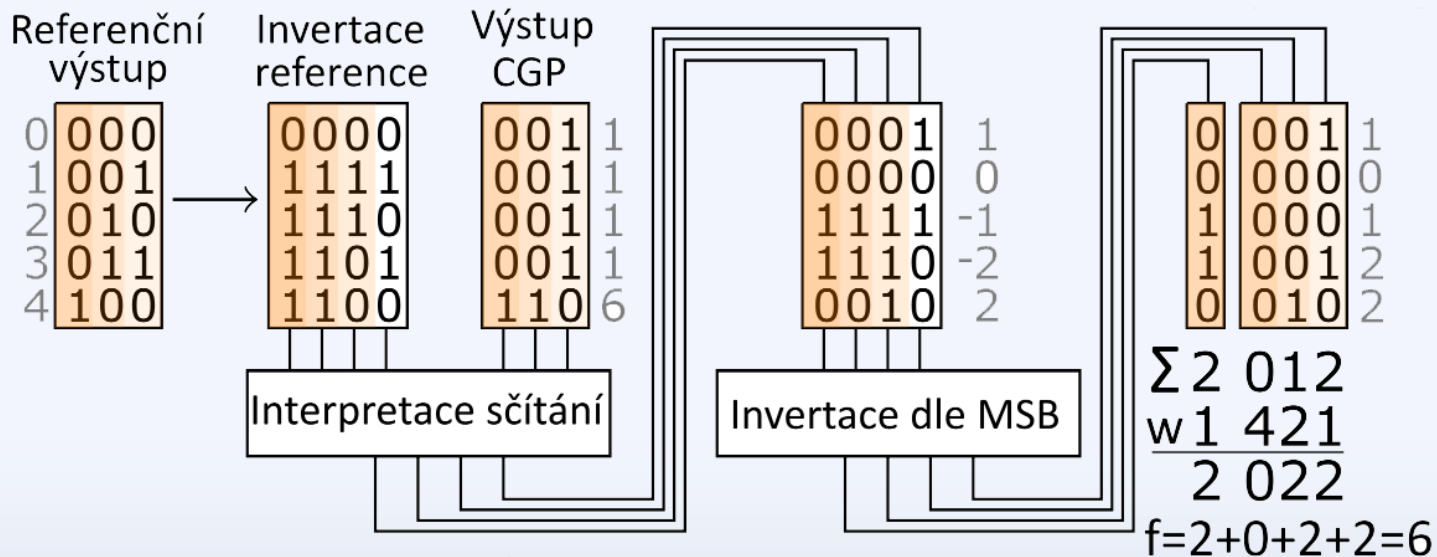
$$fit = \sum_{i=1}^v |y(i) - t(i)|$$

Princip výpočtu

- Bitová transpozice celočíselných hodnot (A0)

Navržená metoda

- výpočet na vektorové úrovni (A1)
- předkompilace metody A1 do strojového kódu (A2)



Ohodnocení přibližného obvodu v CGP

Fitness funkce

$$fit = \sum_{i=1}^v |y(i) - t(i)|$$

Princip výpočtu

↓ Výstup CGP
↑ Referenční výstup

- Bitová transpozice celočíselných hodnot (A0)

Navržená metoda

- výpočet na vektorové úrovni (A1)
- předkompilace metody A1 do strojového kódu (A2)

Násobička	$n_g * 10^6$	A0[eps]	A1[eps]	A2[eps]
4x4	10	54 912	277 923	226 222
5x5	5	12 052	168 688	186 781
6x6	1	2 683	61 673	129 541
7x7	0.1	602	17 644	53 930
8x8	0.01	135	4 782	23 190

170x urychlení

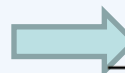
Aproximace v Cannyho detekci hran

Princip detekce

1. Rozmazání
2. Výpočet gradientu
- 3. Výpočet normál a ztenčení hran**
4. Dvojité prahování s hysterezí

Vlastnosti aproximovaných násobiček

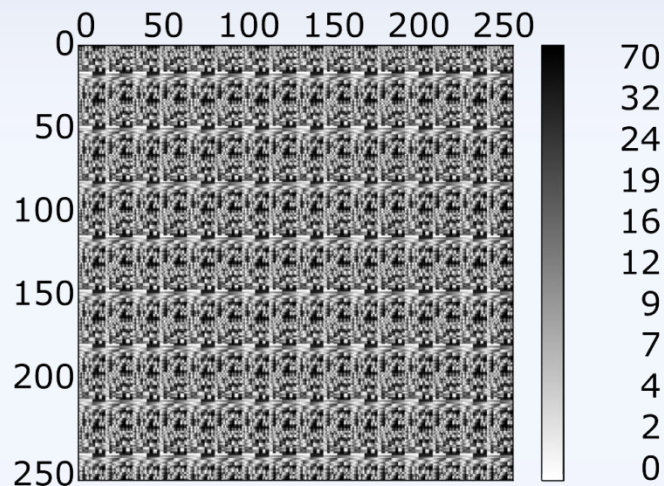
Počet hradel	Relativní plocha	Relativní zpoždění
1	40	1.76
66	3 472	36.11
132	6 760	39.39
198	10 096	48.64
264	13 824	52.56
330	16 752	56.66



(Plně funkční ×)

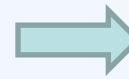


Přibližné ztenčení hran

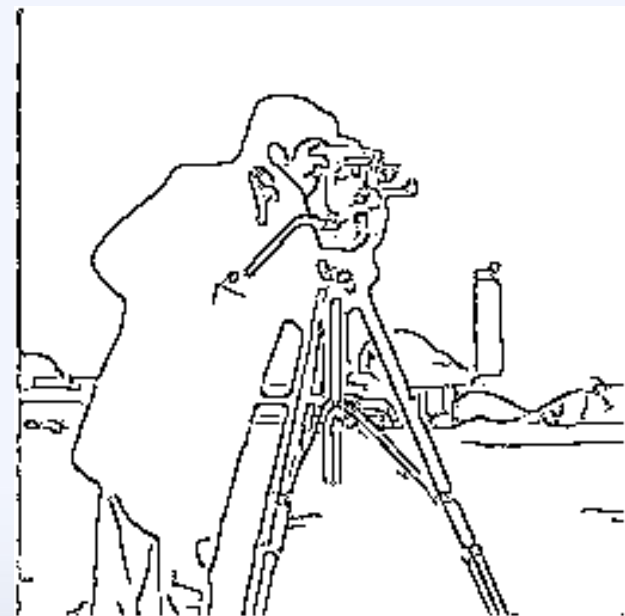


Vlastnosti aproximovaných násobiček

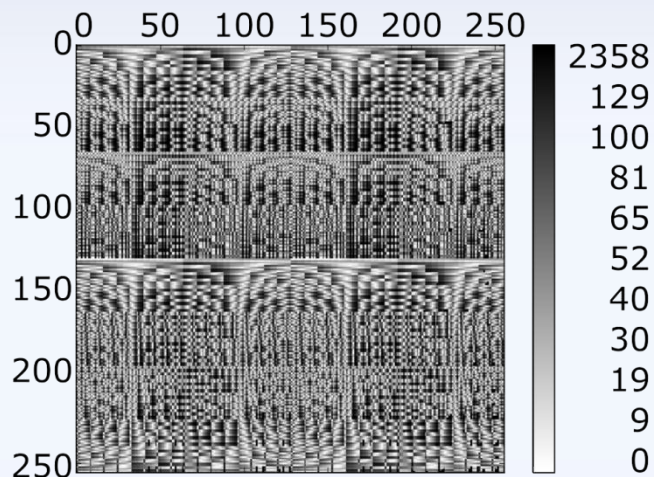
Počet hradel	Relativní plocha	Relativní zpoždění
1	40	1.76
66	3 472	36.11
132	6 760	39.39
198	10 096	48.64
264	13 824	52.56
330	16 752	56.66



(Plně funkční ×)

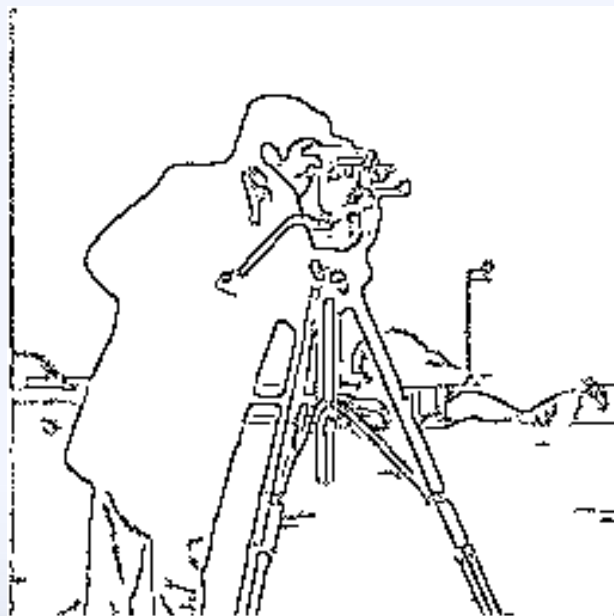
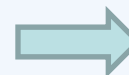


Přibližné ztenčení hran

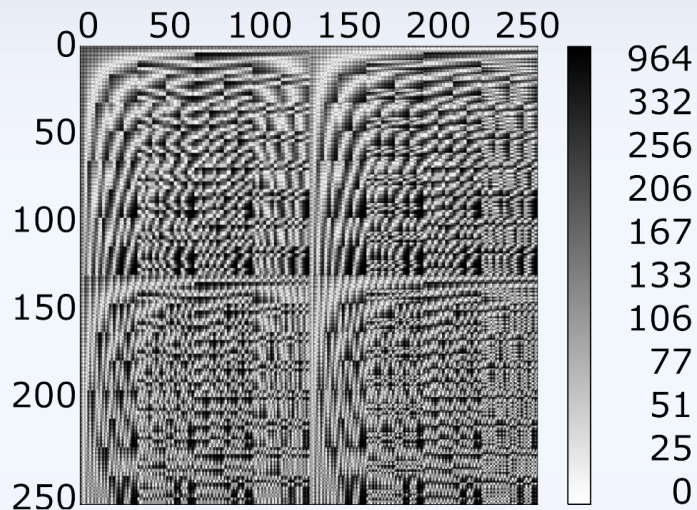


Vlastnosti aproximovaných násobiček

Počet hradel	Relativní plocha	Relativní zpoždění
1	40	1.76
66	3 472	36.11
132	6 760	39.39
198	10 096	48.64
264	13 824	52.56
330	16 752	56.66

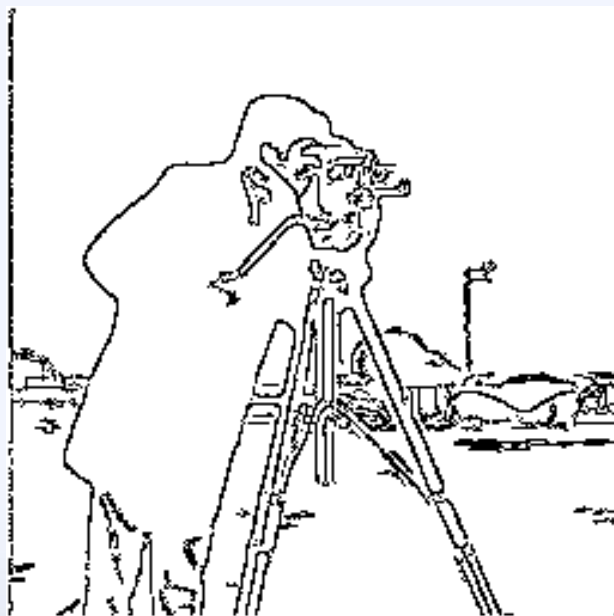
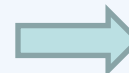


Přibližné ztenčení hran

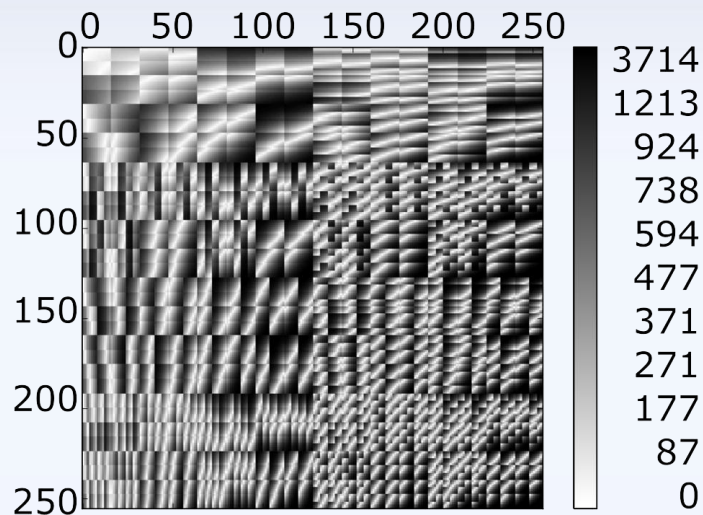


Vlastnosti aproximovaných násobiček

Počet hradel	Relativní plocha	Relativní zpoždění
1	40	1.76
66	3 472	36.11
132	6 760	39.39
198	10 096	48.64
264	13 824	52.56
330	16 752	56.66



Přibližné ztenčení hran

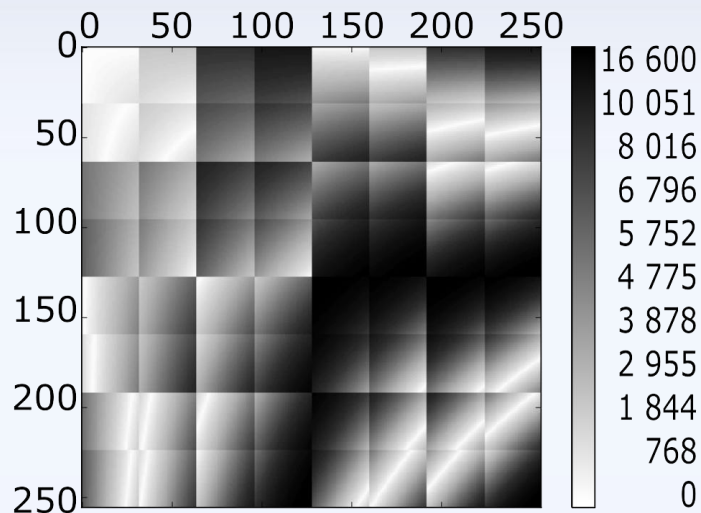


Vlastnosti aproximovaných násobiček

Počet hradel	Relativní plocha	Relativní zpoždění
1	40	1.76
66	3 472	36.11
132	6 760	39.39
198	10 096	48.64
264	13 824	52.56
330	16 752	56.66 (Plně funkční ×)



Přibližné ztenčení hran



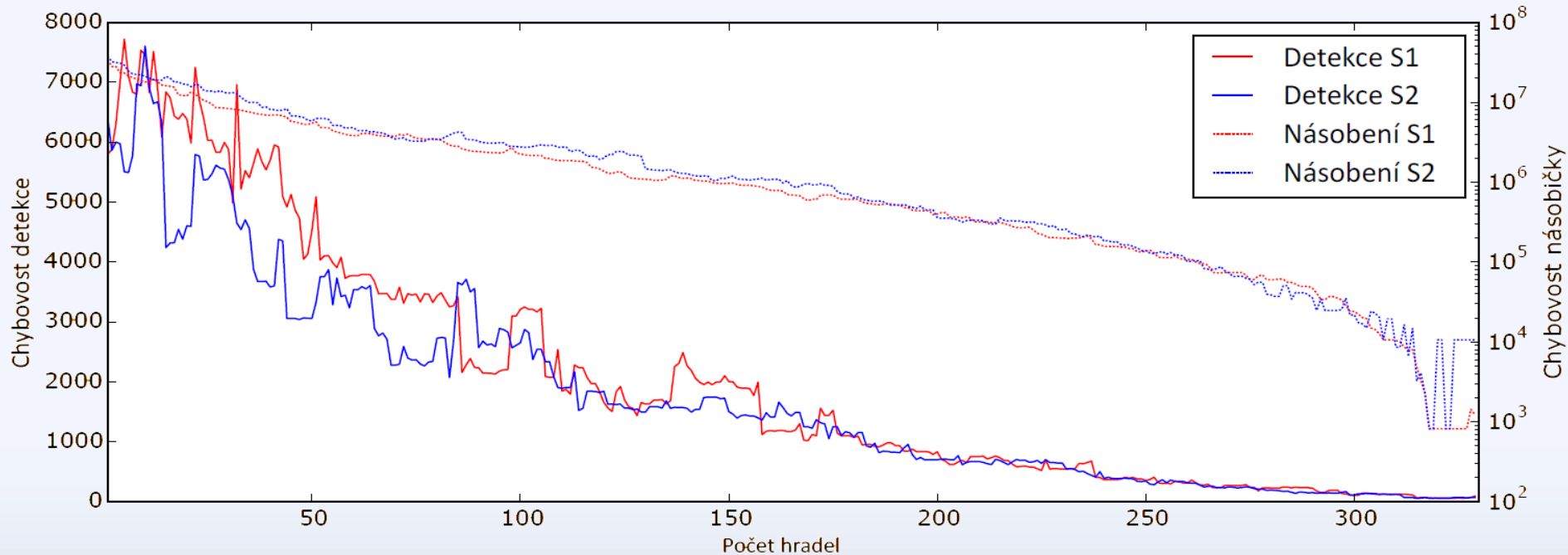
Vlastnosti aproximovaných násobiček

Počet hradel	Relativní plocha	Relativní zpoždění
1	40	1.76
66	3 472	36.11
132	6 760	39.39
198	10 096	48.64
264	13 824	52.56
330	16 752	56.66 (Plně funkční ×)



Přibližné ztenčení hran

*Hůře aproximované násobičky
mohou lépe detekovat hrany!*



Přibližná Sobelova filtrace

Sobelovy filtry

Konvoluce kernely ve tvaru:

Výpočet gradientu

1 2 1

0 0 0

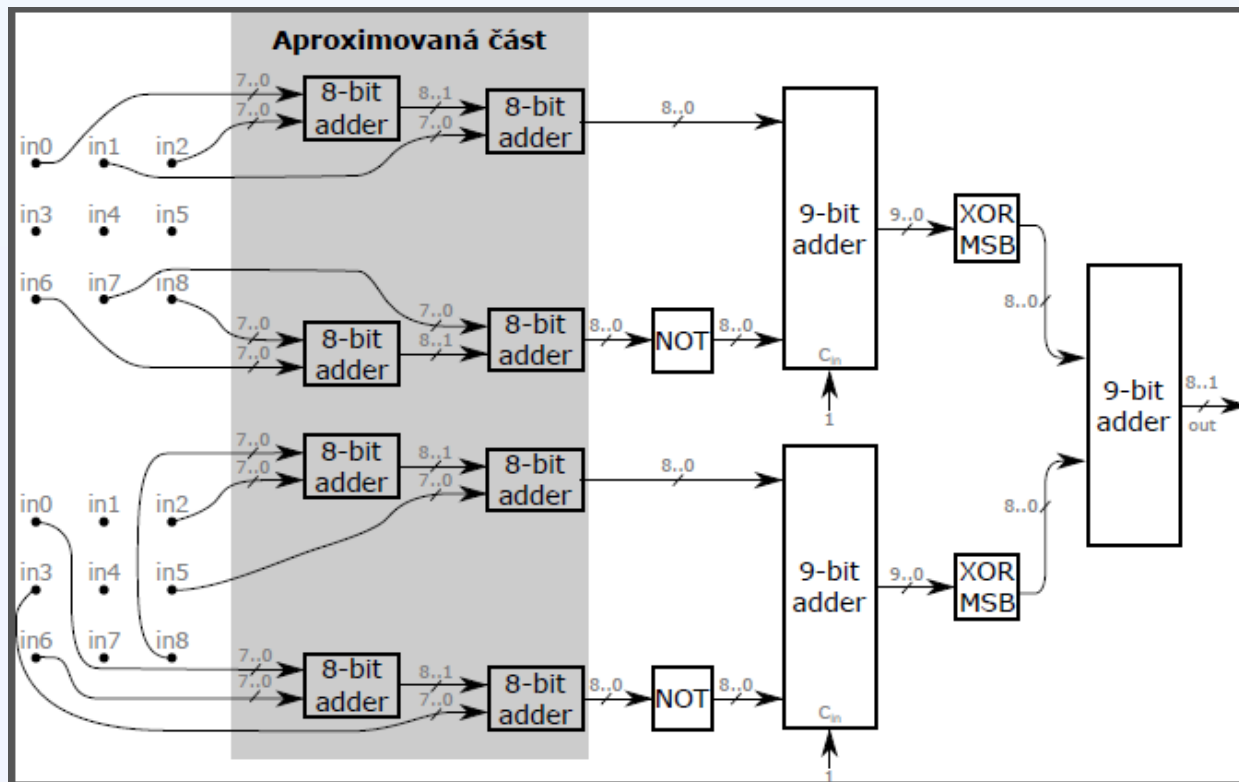
-1 -2 -1

a

1 0 -1

2 0 -2

1 0 -1



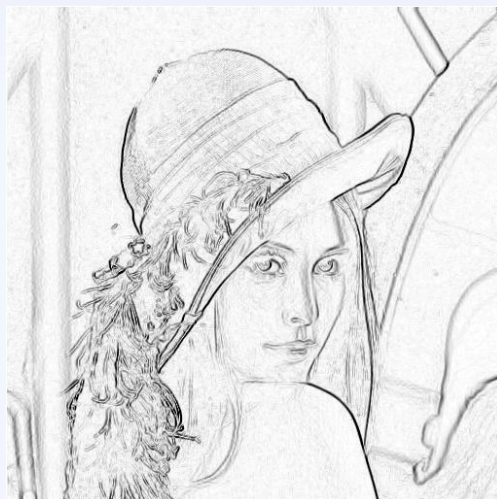
Aproximace sčítání implementované podle rychlých Kogge-Stone sčítaček => pro osmibitové sčítání je jen 7 hradel v kritické cestě!

Přibližná Sobelova filtrace

plně funkční +, 62 hradel



43 hradel



42 hradel (nevhodná)



Detektor hran	Plně funkční	Aproximovaný	Zlepšení (ratio)
Počet hradel	897	745	17%
Relativní plocha	41 716	34 868	16.4%
Relativní zpoždění	65.16	61.13	9%

Závěr

Shrnutí práce

- Výpočet fitness významně urychlen **170x** oproti naivnímu přístupu.
- Ukázáno, že i nepřesné obvody **umí detekovat hrany**.
- Aproximační obvody jsou **rychlejší, menší** a mají **nižší příkon**.

Možné pokračování

- CGP v aproximaci
 - Aplikačně specifické přibližné obvody
 - Eliminace problému škálovatelnosti CGP
- Hledání nových aplikací k aproximaci
- Implementace Sobelova filtru

Otázka

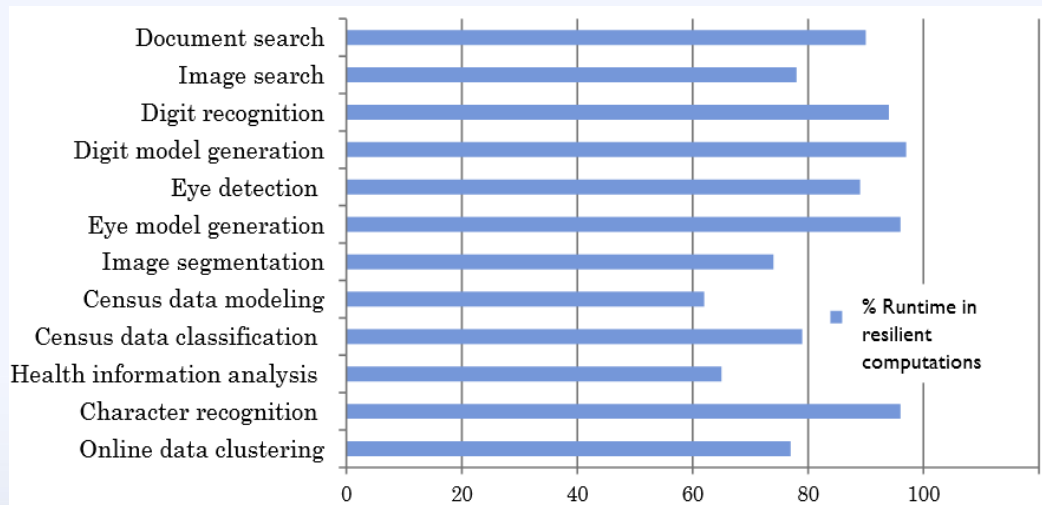
Q: Můžete komentovat dobu trvání experimentu uvedeného v části 7.2, kde se uvádí $N_e=500\ 000$ versus dvoutýdenní běh.

- Jeden běh byl nastaven na 500 000 generací, v každé generaci se ohodnocovali 4 potomci s různě velkým prohledávacím prostorem. Doba jednoho běhu CGP trvala průměrně 12 minut.
- Každá evoluční aproximace čítala 50 běhů, využil jsem školního serveru k paralelnímu spuštění 12 běhů evoluční aproximace.
- Bylo vytvořeno 330 evolučních aproximací - tvořily se různě velké přibližné násobičky, které byly sestaveny z 329 hradel, 328 hradel ... 1 hradlo.

$$330 * 12 * 50 / 12 = 15812 \text{ minut} \sim 263 \text{ hodin} \sim 11 \text{ dní}$$

- Heuristická inicializace CGP obnáší jistou režii. (seq.)
- Připočteme-li režii k nalezení všech přibližných obvodů, doba běhu algoritmu činila cca. dva týdny.

Aplikace, které tolerují chyby



➡ Můžeme aproximovat cca **83%** výpočetního času.