

Ťažké inštancie pre SAT solvery

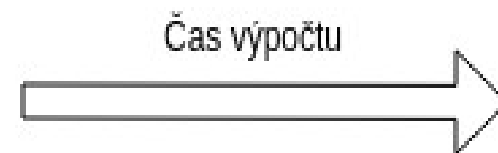
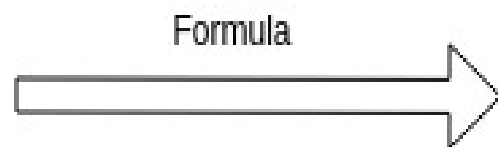
Vedúci:

prof. RNDr. Rastislav Kráľovič, PhD.

Erik Kučák

Motivácia a cieľ

- SAT je NP-úplný problém (Cook-Levinova veta)
- Existuje veľa rôznych SAT solverov s rôznymi heuristikami.
- Je ťažké zistiť, ktorý je rýchlejší.
- Chceme zistiť pre daný SAT solver ťažké inštancie (teda tie formuly, ktoré počíta najdlhšie).



V čom je problém?

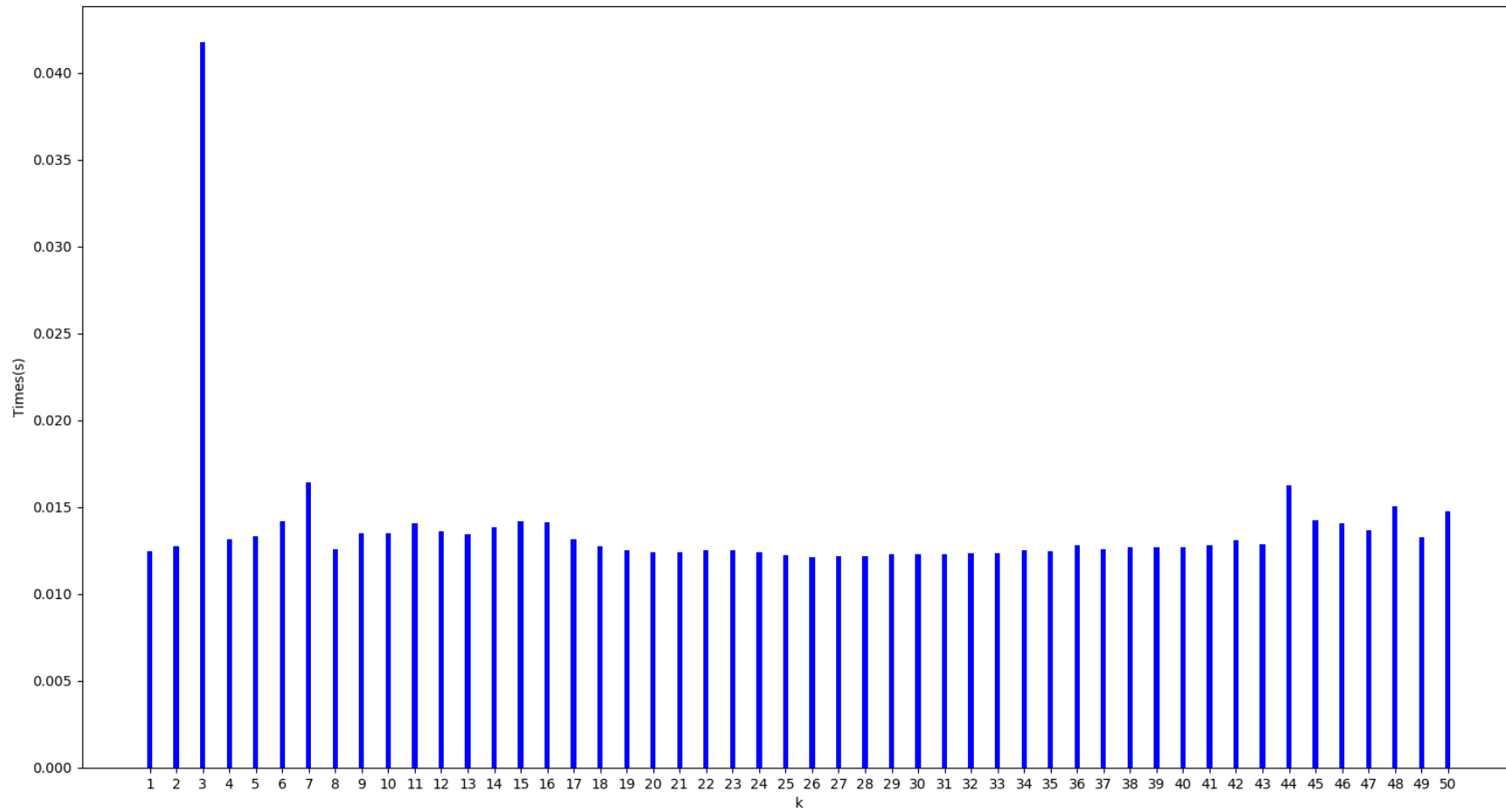
- Ide však o NP-optimalizačný problém, kde žiadny polynomiálny vyhľadávací algoritmus nepoznáme
- Mohli by sme skúsiť vygenerovať všetky rozumné formuly, ktoré by mohli byť ťažké, ale je to časovo nereálne.
- Preto použijeme genetické programovanie
- Budeme sa snažiť vypočítať čo najoptimálnejší výsledok

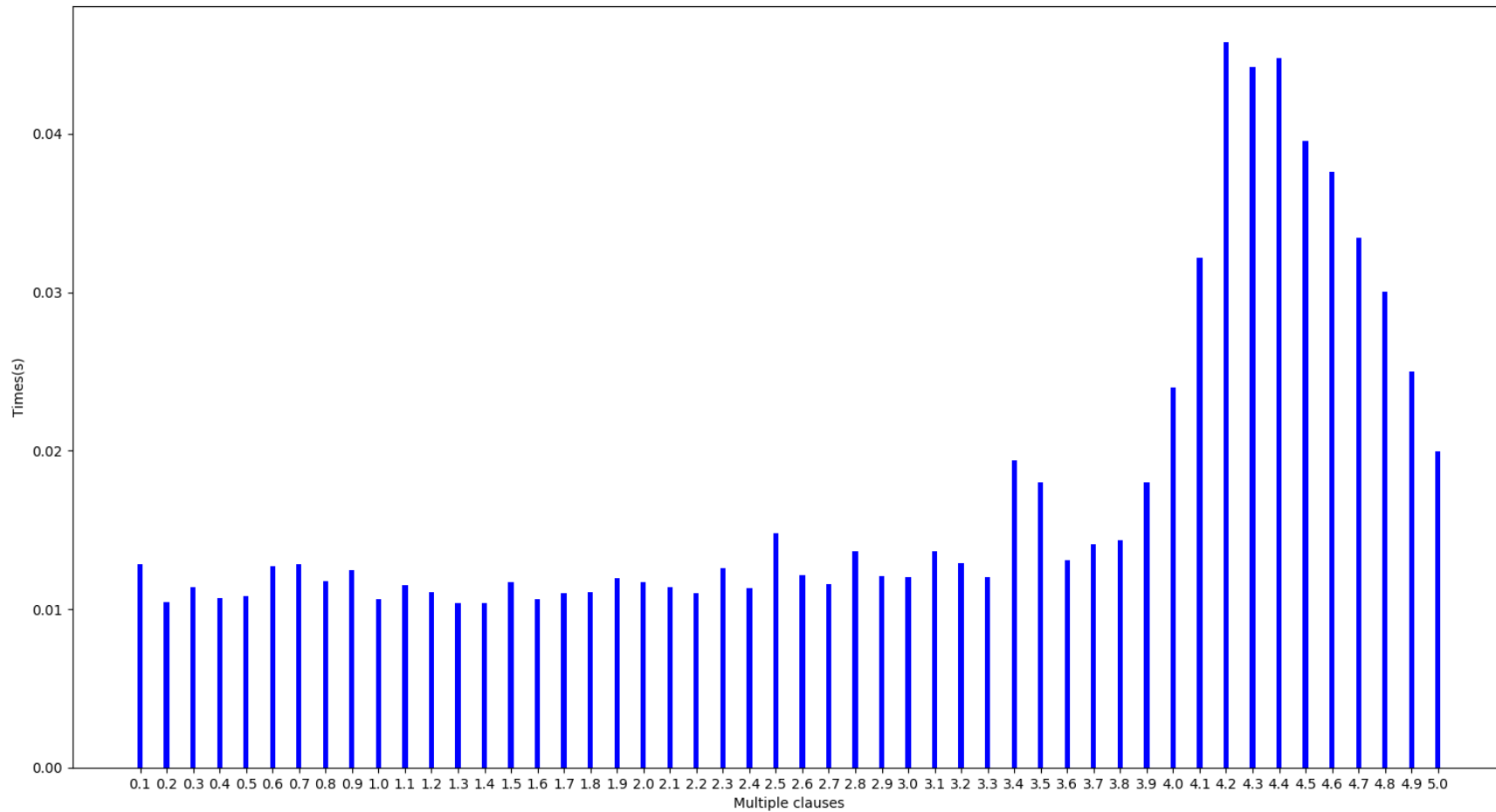
Potrebné pojmy

- (Jedinec) Konkrétny prvok v prostredí, ktorý sa snaží svojimi vlastnosťami prežiť, v našom prípade sa jedinec myslí formula.
- (Zdatnosť) Zdatnosť alebo fitness sa myslí stupeň vlastnosti jedincov, ktoré vyžaduje dané prostredie na to, aby v ňom jedinec dokázal prežiť. Prostredie sa v našom prípade myslí konkrétny SAT solver a vlastnosť prežitia sa myslí rýchlosť výpočtu, ktorou SAT solver vypočíta formulu. Čím pomalší je výpočet, tým je väčšia pravdepodobnosť, že formula prežije.
- (Populácia) Súbor jedincov (formúl), ktorí sa snažia svojimi vlastnosťami prežiť v prostredí.
- (Generácia) Konkrétna populácia.

Analýza SAT solvera

- Všetky dnešné SAT solvery používajú na formuly KNF formu
- KNF formuly v 3-SAT forme sa počítajú ťažšie
- Binárnym vyhľadávaním zistíme približne ideálny počet premenných
- Približne ideálny počet klauzúl je 4.2 krát viac ako premenných



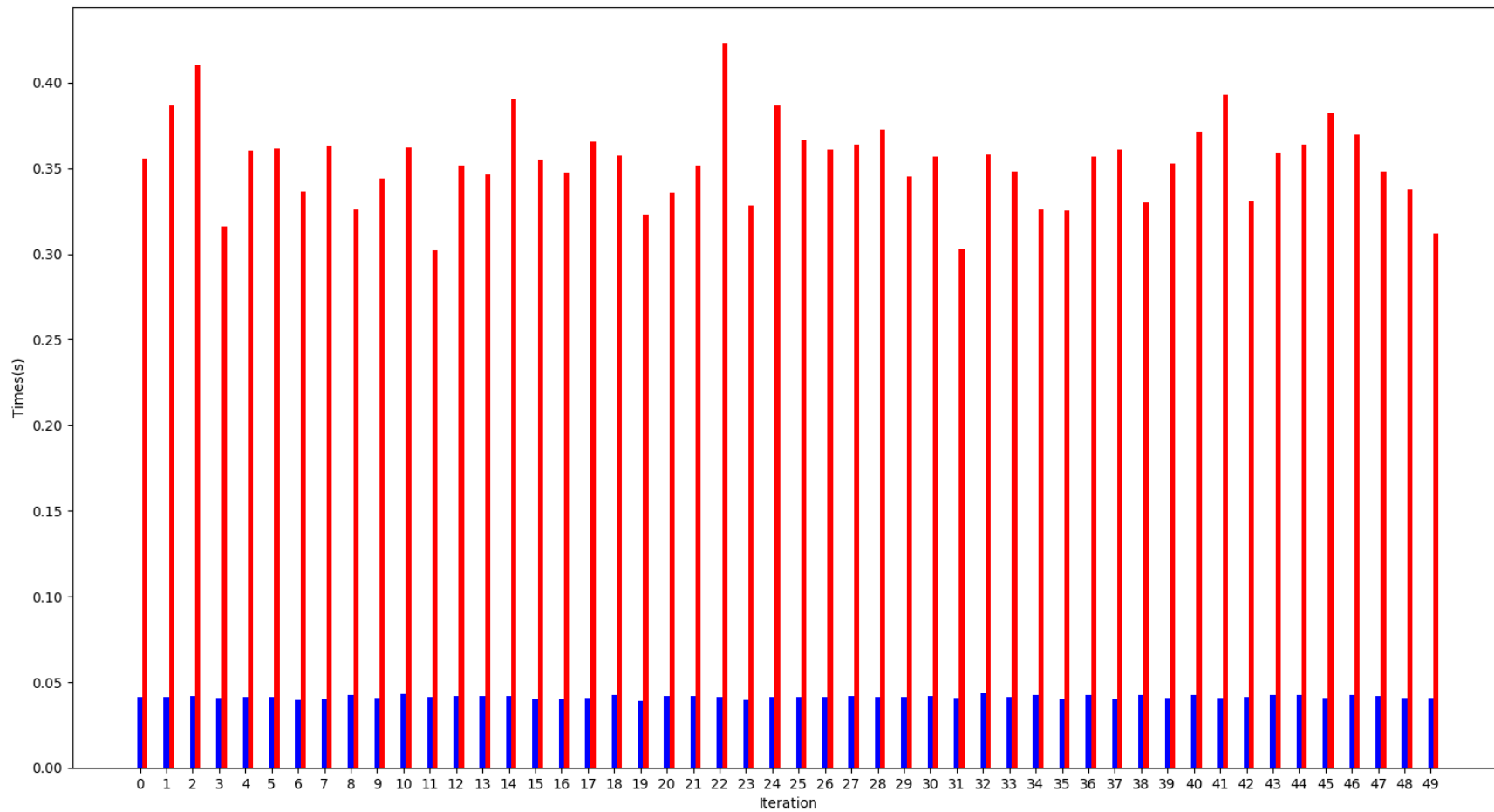


Inicializácia

- Vygenerujeme n formúl s rovnomerne rozmiestnenými premennými, kde sú taktiež rovnomerne rozmiestnené negácie

$$F = (L_1 \vee L_2 \vee L_3) \wedge \dots \wedge (L_{n-1} \vee L_n \vee L_{n+1}) \wedge \dots \wedge (L_{p-2} \vee L_{p-1} \vee L_p)$$

$$\underbrace{L_1, L_2, L_3, \dots, L_{n-1}, L_n}_{\text{permutácia}_1}, \underbrace{L_{n+1}, \dots, L_{2n-1}, L_{2n}}_{\text{permutácia}_2}, \dots, \underbrace{L_{p-2}, L_{p-1}, L_p}_{\text{permutácia}_i}$$



Cyklus genetického algoritmu

- Vyhodnotenie populácie
- Selekcia najpomalších formúl
- Reprodukcia
- Kríženie

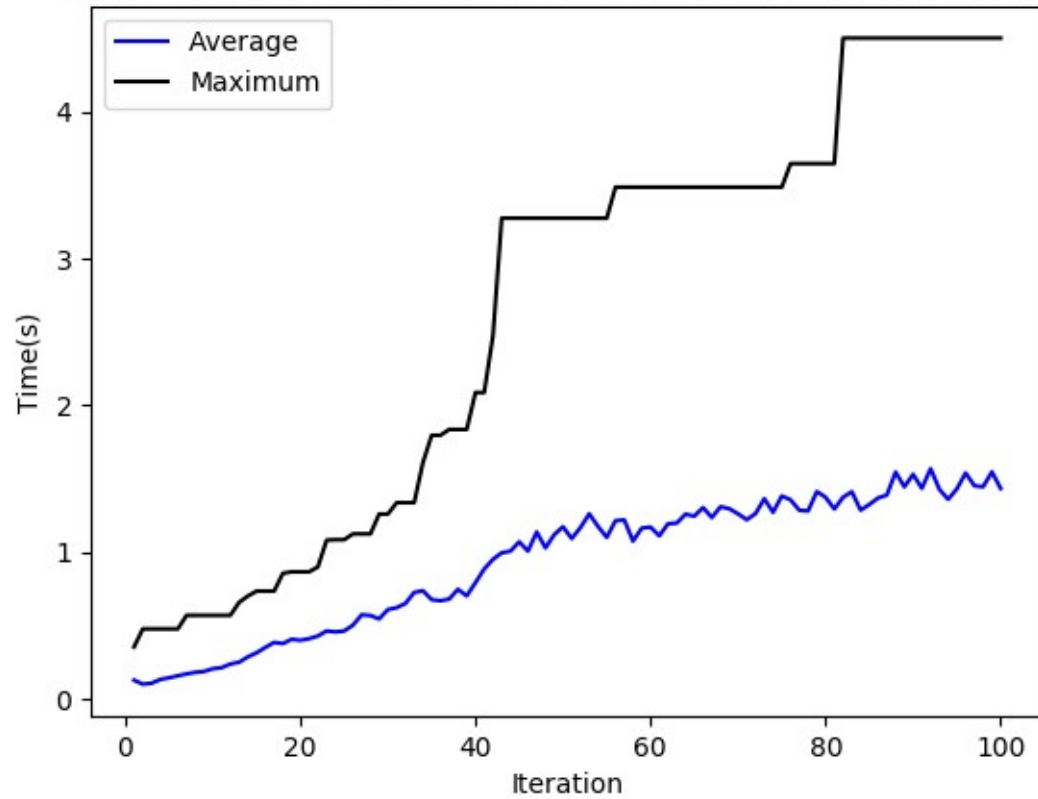
$$\text{Otec} = c_1 \wedge c_2 \wedge \dots \wedge c_t$$

$$\text{Matka} = c'_1 \wedge c'_2 \wedge \dots \wedge c'_t$$

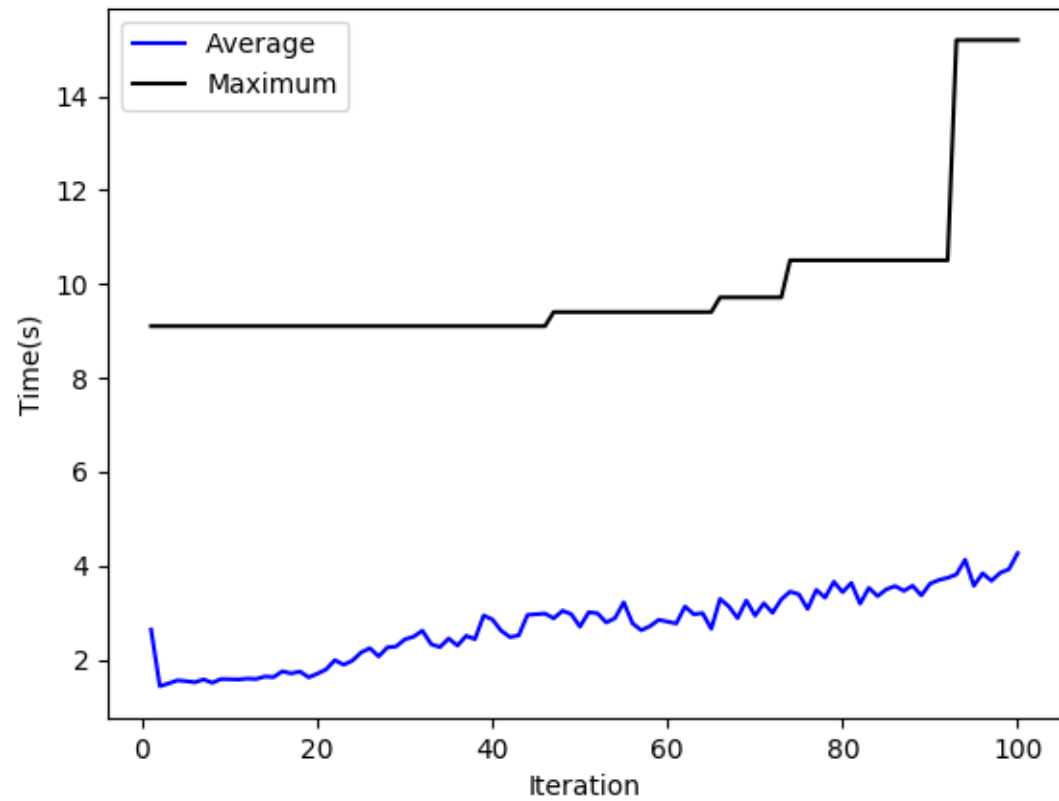
$$\text{Dieťa} = c_1 \wedge c'_2 \wedge \dots \wedge c_t$$

- Mutácia
- Ukončovacia podmienka

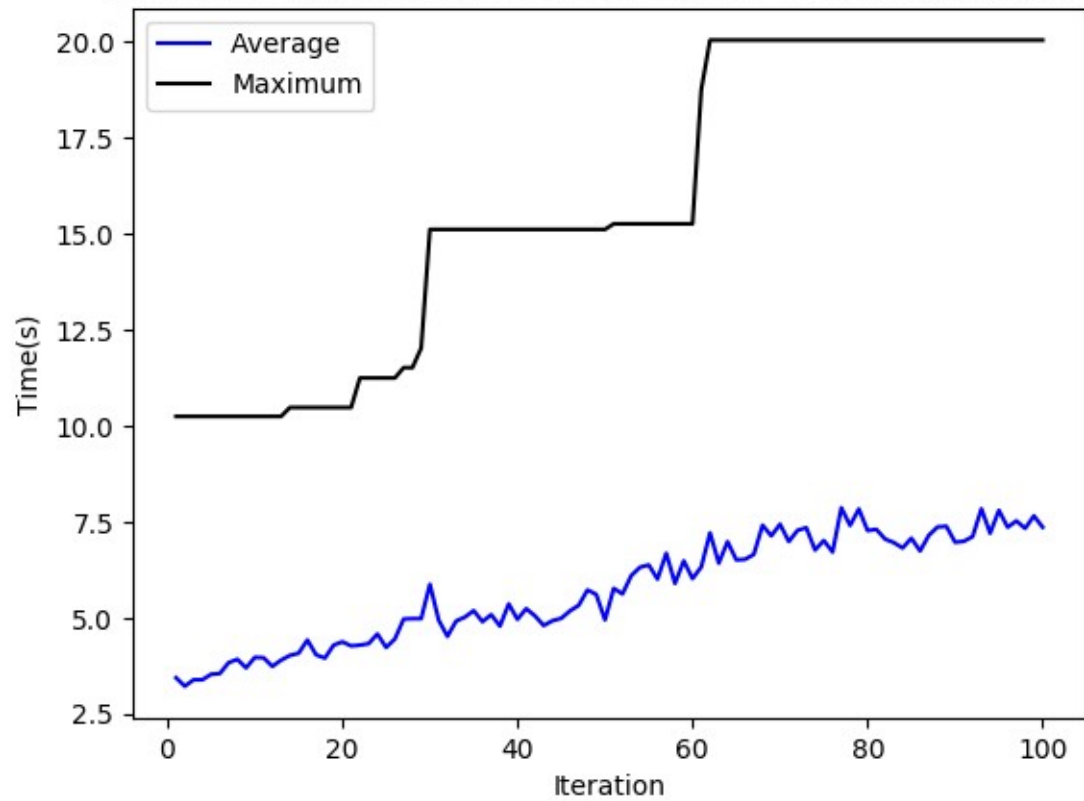
Computing time (Variables: 150, Clauses: 630, Population: 100)



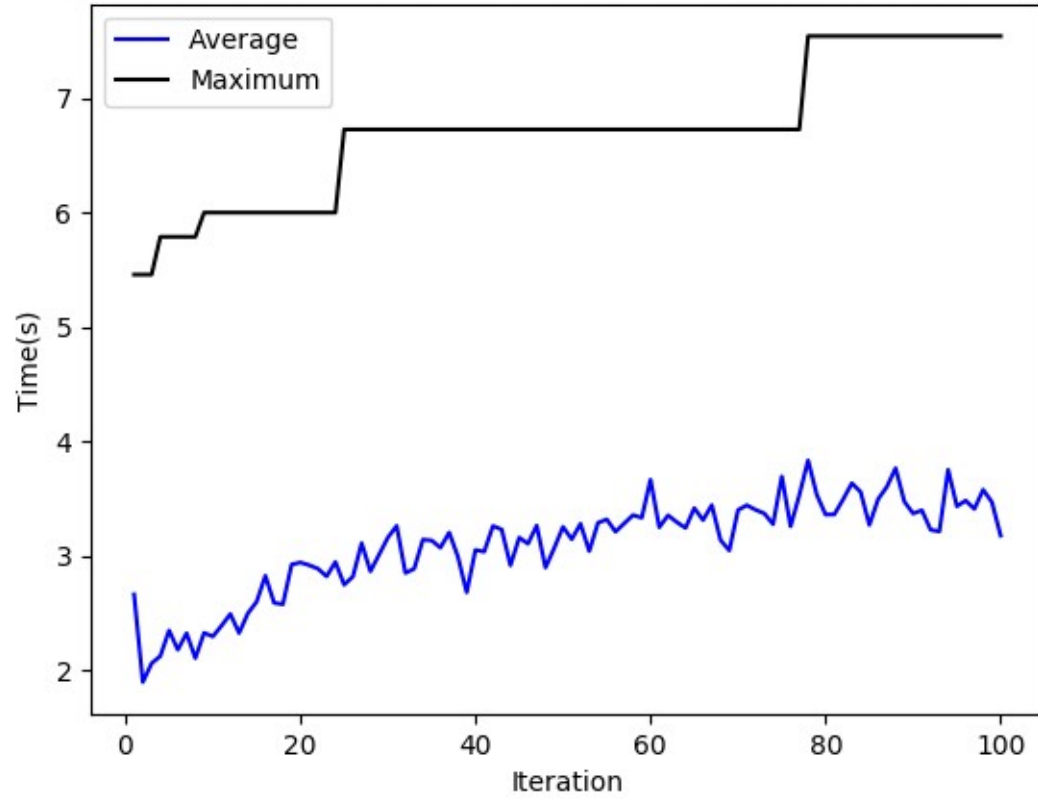
Computing time (Variables: 150, Clauses: 630, Population: 100)



Computing time (Variables: 150, Clauses: 630, Population: 100)



Computing time (Variables: 175, Clauses: 735, Population: 100)



Ďakujem za pozornosť