# EVO project 2023/24

## Simulované žíhání - řešení Sudoku

* Při generování ladit pouze na 9x9 nebo jiné (hlavně generalizace alg, jestli na konstantní velikost sudoku nebo pro různorodé)
* Generování hry pomocí py-sudoku, reprezentace matice pomocí 2d pole
  + Stačí základní python pole nebo pře implementovat do numpy (kvůli rychlosti)
* Vygenerování hodnot na začátku
  + Možno rovnou zaplnit bloky (3x3) korektními hodnotami [1..9] nebo náhodně generovat do celé matice
* Konfigurace běhu - rozměry matice, obtížnost (míra prvotního zaplnění zadáním), parametry simulovaného žíhání (počet iterací, prvotní teplota, update teploty)
* Prohledávání sousedních vztahů jako záměnu hodnoty v jednom bloku
* Jedna iterace – ukončit po jednom prohledaném stavu nebo více?
* Výstup
  + Jeden pokus
  + Mnoho pokusů a statistiky z nich
    - Se stejným nastavením
    - S různým
    - Jako druhý program který bude využívat modul SAsudoku
  + Graf průběhu – jen změna nebo pro každou iteraci
* Teplota
  + Jaký pokles linéární, exponenciální
  + Kdy Znovu nažhavení pro zaseknutí nebo konec neúspěchem
* Prohodit pomocí cyklu (např. tři hodnoty)
* Velikost 16x16
* Nedělat více testů než zvládnu odprezentovat na 6 minut
* S teplotou aspoň dvě možnosti, geometr a linear
* Hlavně zaměřit na velikost a váha konfliktů s předdefinovanými
* Graf – boxplot
* Nebo porovnání pokud se nebude dařit hledat
* Nebo počet konfliktů v generaci (konvergenční křivka)
* Výstupem je hlavně zpráva, kód spuštěn jen jednou
* To co stihnu odprezentovat
  + Říct co je na pevno
  + Mluvit o tom co jsem zkoušel
  + Zmínit typy teploty geo lin
  + Srovnání atd
  + Teoreticky můžu i konstantní teploty
* Nepřegenerovat začátek, lepší bude reheat
* Pro každé nastavení aspoň 30 testů
* Jde o experimentaci, ne o výsledky