

**VBC**

**ZS 2021/22**

Tomáš Benda, 200604

task #4

## Zadání

Nastudujte a naprogramujte jednu z níže uvedených metaheuristik (přiřazení dle cvičícího) a demonstруйте vaši implementaci na řešení níže uvedených úloh (dodržíte specifikaci DOD – definičních intervalů).

### Glowworm Swarm Optimization (GSO)

Jedná se o algoritmus hejnové inteligence inspirovaný chováním svítících červů (Glowworms) vyskytujících se v Austrálii a na Novém Zélandu.

GSO se snaží napodobit chování Glowworms, které jsou díky látce zvané luciferin měnit intenzitu bioluminiscence. V algoritmu se agenti – Glowworms – pohybují směrem k sousedovi, který má větší hodnotu luciferinu. Svícení pomocí luciferinu je reprezentováno ohodnocením fitness funkce.

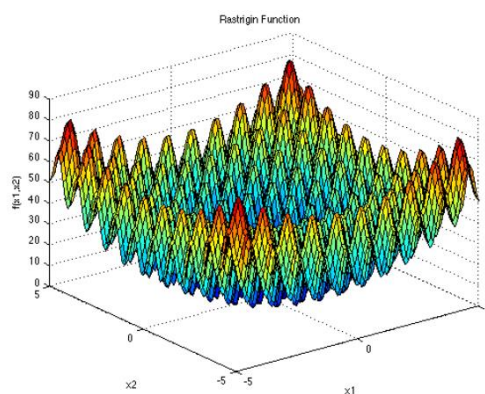
GSO bylo původně navrženo pro řešení numerických optimalizačních problémů, které zahrnovaly výpočet více optim multimodálních funkcí. Tento algoritmus se tedy dá využít, pakliže cílem není nalézt pouze globální minimum.

### Popis chodu algoritmu

1. Na začátku jsou agenti náhodně rozmístěni a je jim přiřazena inicializační hodnota luciferinu.
2. V dalším kroku je hodnota Luciferinu aktualizována na základě ohodnocení fitness funkce.
3. Další fází je pohyb agenta. Nejdříve jsou nalezeni sousedi agenta v jeho rozsahu. Těm je přiřazena pravděpodobnost na základě hodnoty luciferinu, že se k nim daný agent bude pohybovat. Platí, že s větší hodnotou luciferinu je větší pravděpodobnost, že se k danému sousedu bude agent pohybovat. Na základě pravděpodobností je vybrán soused a agent se směrem k němu posune.
4. Posledním krokem je aktualizace dosahu sousedství. Jestliže nebylo dosaženo maximálního počtu restartů, pak se algoritmus vrací do bodu 2.

### Funkce

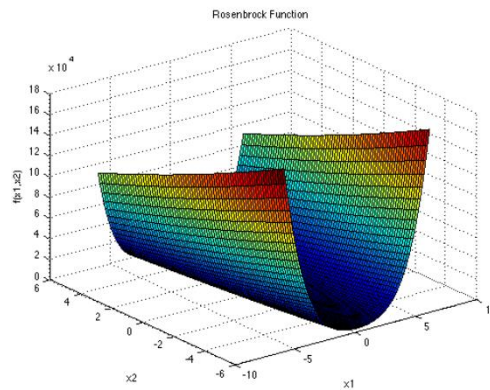
#### Rastriginova funkce



$$f(\mathbf{x}) = 10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)]$$

Rastriginova funkce je definovaná pro  $x_i \in [-5.12, 5.12]$  pro všechna  $i = 1, 2, \dots, d$ . Globální minimum pro 2D je pak v bodě  $x = [0, 0]$  a funkční hodnota je  $y = 0$ .

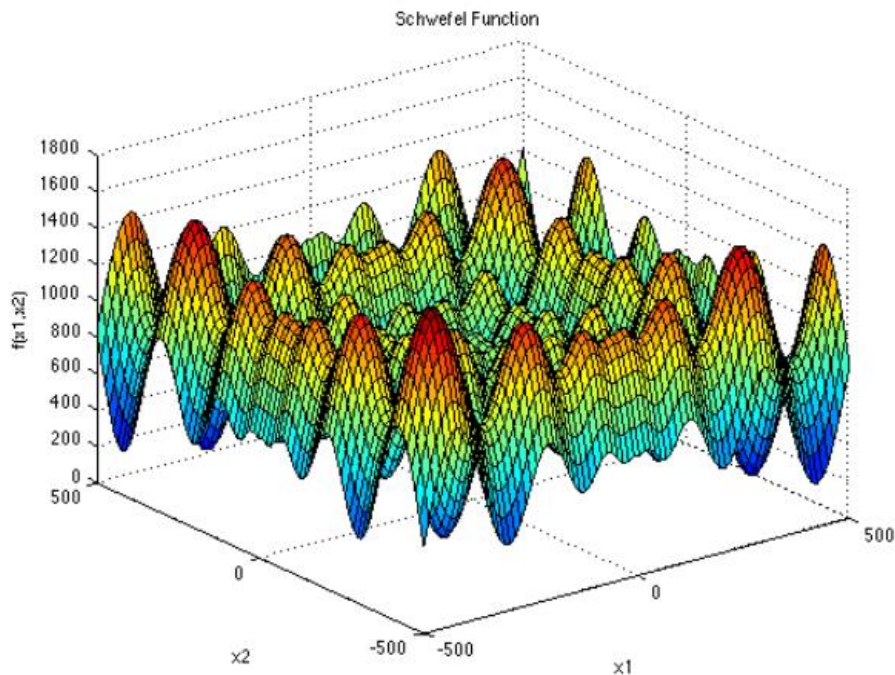
## Rosenbrockova funkce



$$f(\mathbf{x}) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$$

Rosenbrockova funkce je definovaná pro  $x_i \in [-5, 10]$  pro všechna  $i = 1, 2, \dots, d$ . V tomto protokolu je omezena na  $x_i \in [-2.048, 2.048]$  pro všechna  $i = 1, 2, \dots, d$ . Globální minimum pro 2D je pak v bodě  $x = [1, 1]$  a funkční hodnota je  $y = 0$ .

## Funkce Schwefel



$$f(\mathbf{x}) = 418.9829d - \sum_{i=1}^d x_i \sin(\sqrt{|x_i|})$$

Funkce Schwefel je definovaná pro  $x_i \in [-500, 500]$  pro všechna  $i = 1, 2, \dots, d$ . Globální minimum pro 2D je pak v bodě  $x = [420.9687, 420.9687]$  a funkční hodnota je  $y = 0$ .

## Postup řešení

V rámci tohoto úkolu byl vytvořen algoritmus Global Swarm Optimization v programovacím jazyce Python. Program přijímá následující parametry:

- $n_p$  = velikost populace
- $sensor\_range$  = poloměr dosahu sensoru (použité při úpravě Local decision range pro hledání sousedů)
- $range\_min$  = minimální dosah (nastaven ve všech případech na 0)
- $\gamma$  = luciferin enhancement constant = konstanta využitá při aktualizaci hodnoty luciferinu
- $\rho$  = luciferin decay constant = konstanta využitá při aktualizaci hodnoty luciferinu
- $step$  = délka kroku
- $\beta$  = konstanta využitá pro aktualizaci velikostí dosahu sousedství
- $n_a$  = požadovaný počet sousedů – konstanta využitá pro aktualizaci velikostí dosahu sousedství
- $nRuns$  = počet restartů algoritmu

Autoři sami doporučují a ve své publikaci využívají následné hodnoty pro dané parametry. V průběhu testování byly parametry různě měněny.

$\rho$	$\gamma$	$\beta$	$n_t$	$s$	$\ell_0$
0.4	0.6	0.08	5	0.03	5

## Výsledky testování algoritmu

### Rastrigin 2D

Při snaze nastavit algoritmus tak, aby bylo nalezeno globální optimum dle zadání, byly parametry různě měněny. Zpočátku byl nevhodně zvolen menší počet agentů a algoritmus správně našel některá lokální minima, ovšem globální minimum se nalézt nepodařilo. Při zvětšení počtu agentů bylo dosaženo nalezení globálního minima. Cenou za to se ovšem zvýšil se čas výpočtu algoritmu.

Použité parametry:

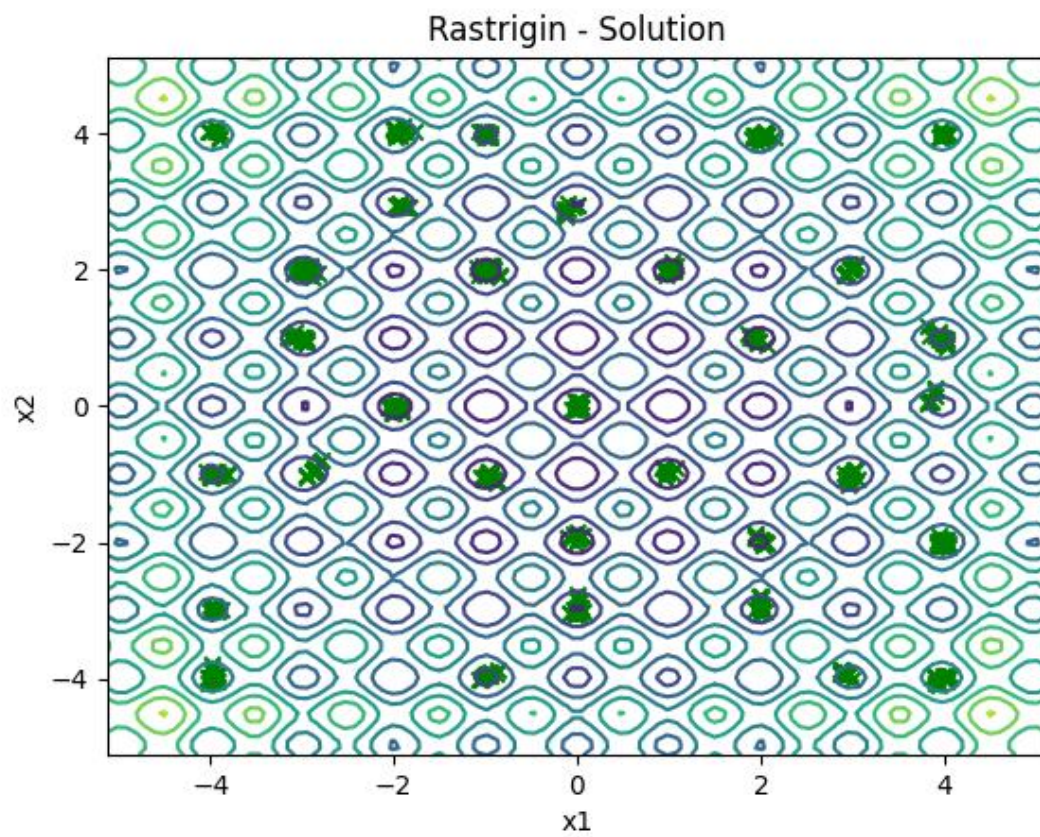
- $n_p = 500$
- $sensor\_range = 4$
- $range\_min = 0$
- $gamma = 0.6$
- $\rho = 0.4$
- $step = 0.03$
- $\beta = 0.08$
- $n_a = 5$
- $nRuns = 1000$

Při vyhodnocení byl sledován čas výpočtu. Zde jsou některé statistické hodnoty pro běh algoritmu při hledání globálního minima:

- Minimální čas výpočtu jednotlivého restartu = 6.7213 s
- Maximální čas výpočtu jednotlivého restartu = 6.9774 s
- Průměrný čas výpočtu jednotlivého restartu = 6.7516 s
- Celkový čas výpočtu algoritmu = 675.1561 s

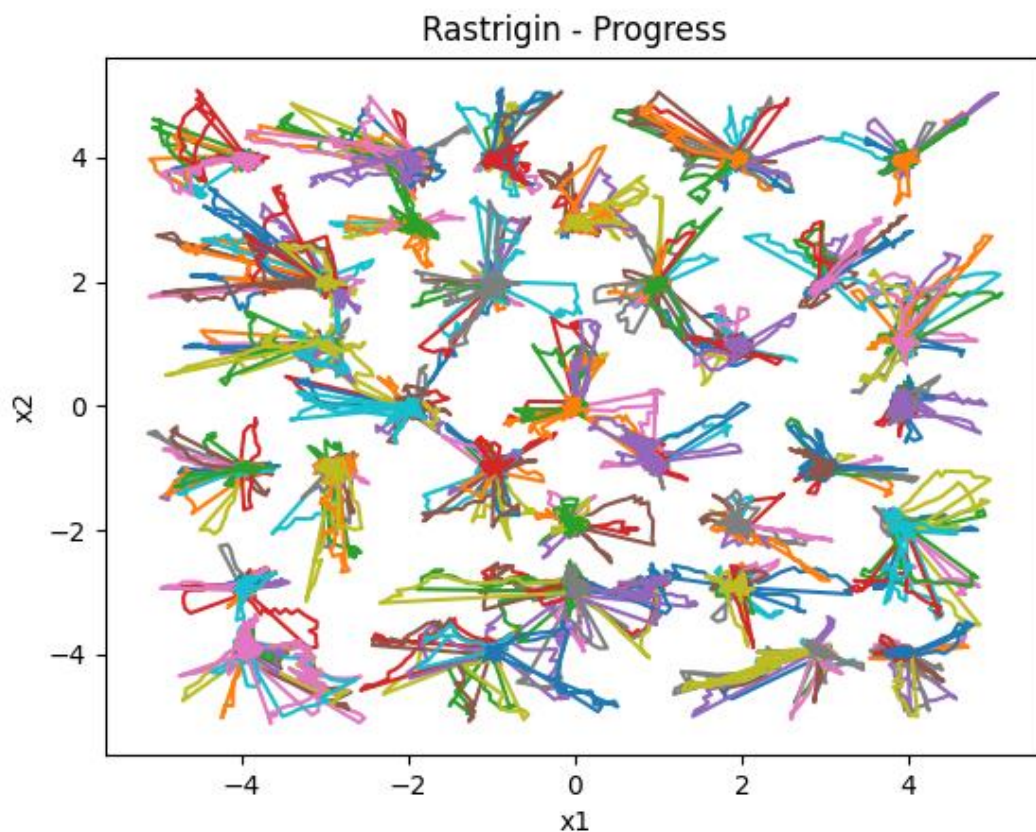
Z výše uvedených statistik je patrné, že čas jednoho restartu algoritmu je značně velký. V další práci by bylo vhodné zaměřit se na optimalizaci kódu a snížit čas výpočtu.

Následující graf ukazuje dosaženého výsledku. Z grafu je patrné, že bylo nalezeno globální minimum stejně jako některá lokální minima. Zvýšením počtu agentů a změnou parametrů by bylo možné nalézt všechna lokální minima v daném parametru.



Další graf zobrazuje posun agentů k jednotlivým minimům. Ačkoliv graf může být nepřehledný kvůli počtu agentů, z grafu je patrné, že se agenti z původní pozice postupně přesunuli do pozice v lokálním respektive globálním minimu.





## Rosenbrock 2D

Parametry pro řešení funkce Rosenbrock byly ponechány téměř nezměněny tak, jak doporučují autoři algoritmu.

Použité parametry:

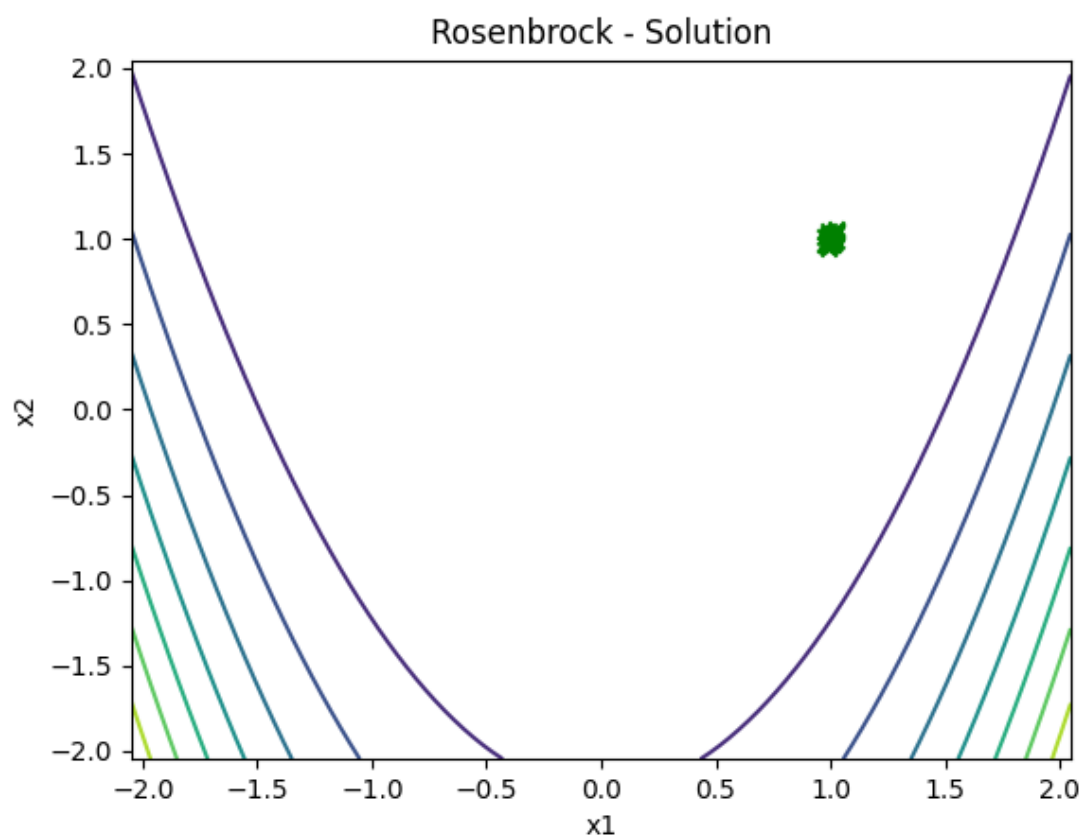
- $n\_p = 100$
- $sensor\_range = 3$
- $range\_min = 0$
- $gamma = 0.6$
- $\rho = 0.4$
- $\beta = 0.03$
- $n\_a = 5$
- $nRuns = 1000$

Při vyhodnocení byl opět sledován čas

- Minimální čas výpočtu jednotlivého restartu = 0.0742 s
- Maximální čas výpočtu jednotlivého restartu = 0.2198 s
- Průměrný čas výpočtu jednotlivého restartu = 80.4588 s
- Celkový čas výpočtu algoritmu = 0.0805 s

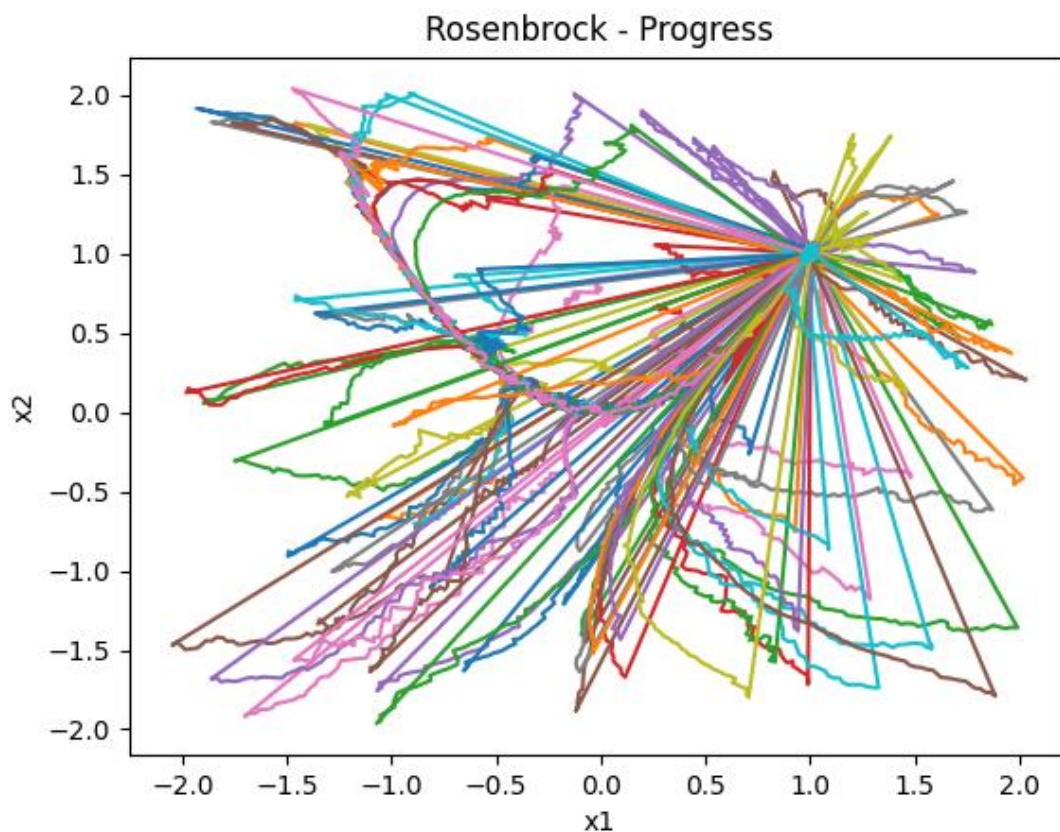
Z výše uvedených statistik je patrné, že čas jednoho restartu algoritmu je značně velký. V další práci by bylo vhodné zaměřit se na optimalizaci kódu a snížit čas výpočtu.

V následujícím grafu je vidět, že algoritmus našel globální minimum. Všichni agenti jsou na pozici blízké globálnímu minimu.



Graf sledující pohyb agentů v jednotlivých generacích jasně ukazuje pohyb směrem ke globálnímu minimu.





## Schwefel 2D

Při testování algoritmu byla většina parametrů zachována stejně jako pro ostatní dvě funkce. Nejvíce se ovšem osvědčilo zvětšení parametru `sensor_range` na 400 a díky tomu byla nalezena největší čtyři lokální minima.

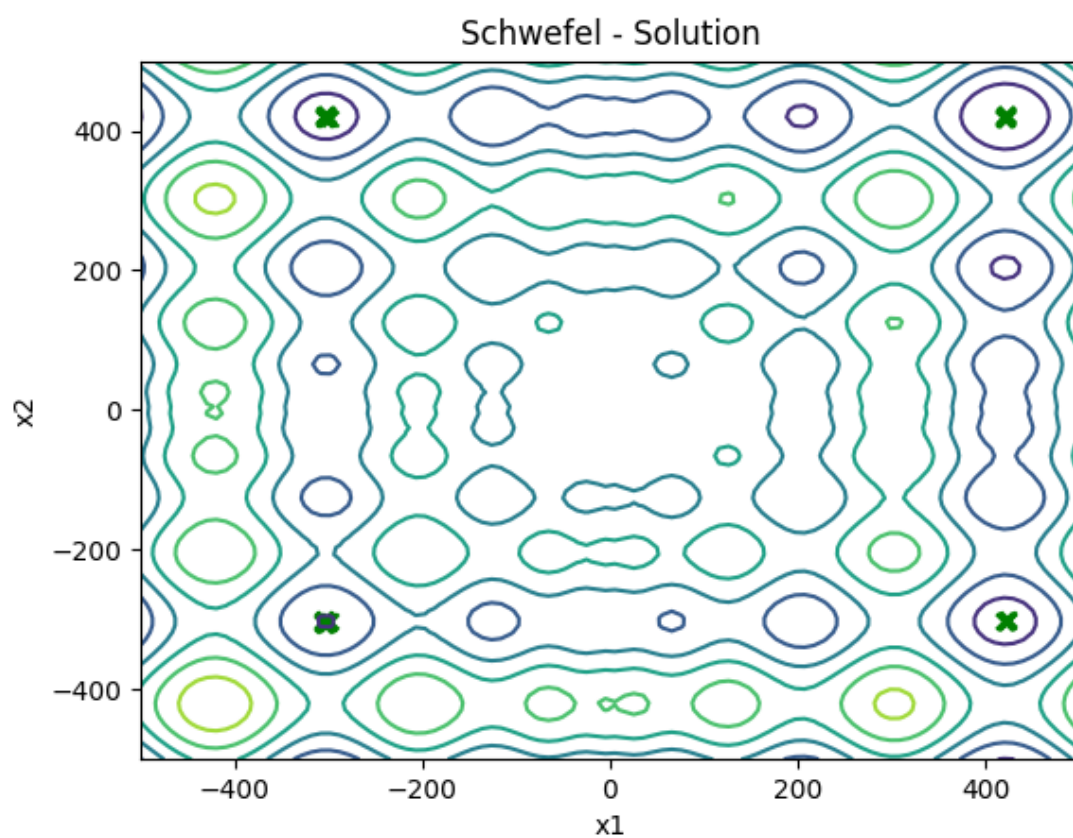
Použité parametry:

- `n_p` = 100
- `sensor_range` = 400
- `range_min` = 0
- `gamma` = 0.6
- `rho` = 0.4
- `beta` = 0.03
- `n_a` = 0.08
- `nRuns` = 1000

Při vyhodnocení byl opět sledován čas výpočtu:

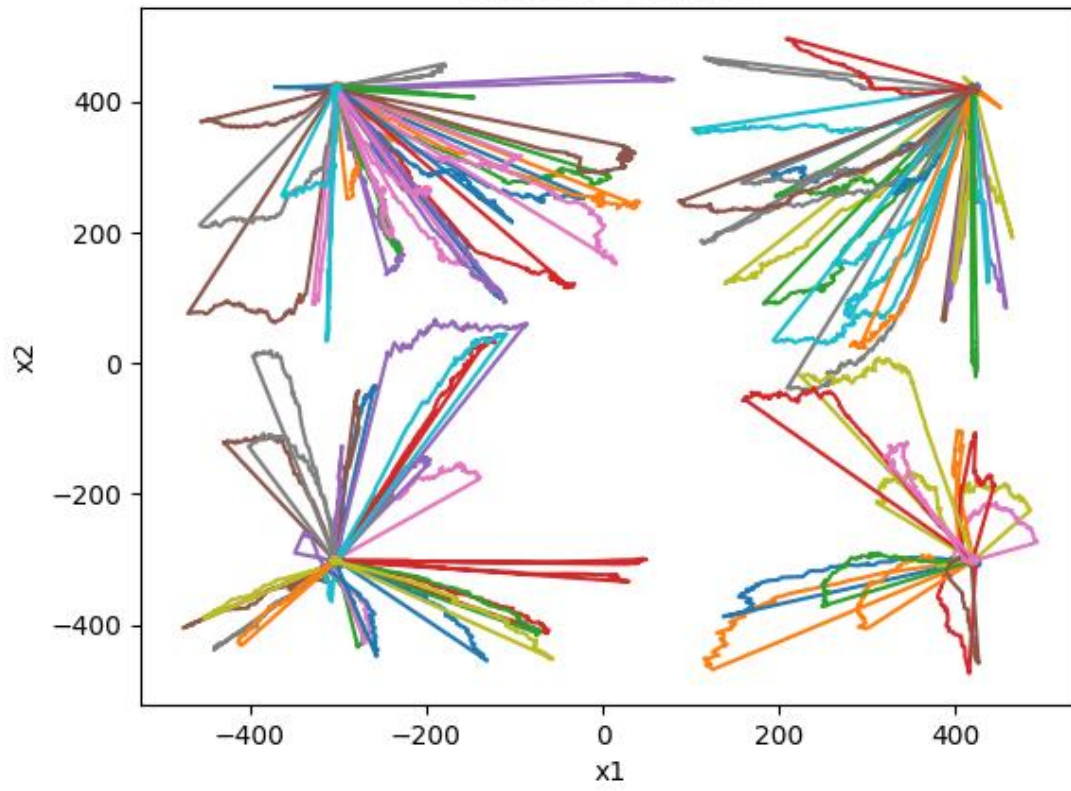
- Minimální čas výpočtu jednotlivého restartu = 0.2897 s
- Maximální čas výpočtu jednotlivého restartu = 0.5577 s
- Průměrný čas výpočtu jednotlivého restartu = 0.2986 s
- Celkový čas výpočtu algoritmu = 298.5617 s

V grafu s výslednými pozicemi agentů je vidět, že agenti našli čtyři největší lokální minima včetně toho globálního.



Z pohybu agentů zobrazeném na dalším grafu je vidět, že všichni agenti nejbližší k jednomu z lokálních minim zamířili právě k němu.

Schwefel - Progress



## Zhodnocení výsledků

V rámci projektu byla vytvořena implementace algoritmu Global Swarm Optimization (GSO). Tento algoritmus byl poté otestován na funkcích Rastrigin, Rosenbrock a Schwefel pro dva parametry.

V průběhu řešení byly testovány různé parametry nastavení algoritmu, aby bylo dosaženo co nejlepších výsledků. U všech funkcí bylo nalezeno globální minimum a další lokální minima, což je hlavním cílem tohoto algoritmu.

V další práci by bylo vhodné zaměřit se na zlepšení rychlosti výpočtu.