

Softcomputing – projektová dokumentace

Demonstrace průběhu řešení TSP pomocí mravenčích algoritmů

Vojtěch Dvořák (xdvora3o)

2. prosince 2024

Úvod

Cílem tohoto projektu na téma *PSO, ACO či jiné přírodou inspirované algoritmy* bylo vytvoření aplikace pro demonstraci průběhu optimalizace při řešení problému obchodního cestujícího pomocí mravenčích algoritmů. Z možných variant zpracování se tedy jedná o variantu první (demonstrace algoritmu). Výsledná aplikace by mohla být např. použita jako pomůcka při výuce či jako nástroj pro hledání zajímavých parametrů mravenčích algoritmů.

Kromě funkcionality demonstrační aplikace, jež je explicitně zmíněna v zadání (načtení souboru s daty, krokování, modifikace parametrů...) aplikace obsahuje také vylepšení, která v něm zmíněna nejsou (validace parametrů, možnost volby konkrétního algoritmu, nástroj pro analýzu výsledků...). Nejdůležitější funkcionalita aplikace je popsána v druhé části této dokumentace v rámci kapitoly 2. Zpracovány byly všechny varianty mravenčích uvedených na přednáškách s výjimkou Rank-based AS. Aplikace může být díky tomu kromě pouhé demonstrace také použita pro jejich porovnání.

1 Popis problému a implementace aplikace

1.1 Řešený problém

Vzhledem k tomu, že cílem bylo zejména demonstrovat fungování mravenčích algoritmů, aplikace řeší standardní problém pro tuto třídu algoritmů, a sice problém obchodního cestujícího (TSP). Jiné problémy lze pomocí těchto algoritmů řešit také např. s využitím redukce. To však není předmětem tohoto projektu.

Aplikace načítá data – souřadnice jednotlivých míst a jejím výstupem je popis nejlepší nalezené cesty, která vede prochází každým místem právě jednou (vyjma počátečního a posledního místa na cestě) a jež začíná a končí ve stejném místě. Vytvořená aplikace dále produkuje také délku této cesty.

1.2 Použité technologie

Projekt je vypracován v jazyce Python ve verzi 3.12. Pro implementaci mravenčích algoritmů byla využita knihovna *numpy* a *scipy*. Aplikace s GUI využívá pro svoje fungování knihovny *tkinter* a *matplotlib*. Důraz byl kladen na využívání co možná nejmenšího počtu knihoven a modulů, které nejsou součástí samotného Pythonu.

Kromě referenčního prostředí byl projekt testován i v prostředí operačního systému Windows 10 s nainstalovaným interpretem jazyka Python v příslušné verzi.

1.3 Mravenčí algoritmy

Jak již bylo uvedeno výše, implementovány byly téměř všechny mravenčí algoritmy uvedené na přednáškách – Ant System, Ant Colony, Elitists Strategy, Ant Density, Ant Quantity a Min-Max Ant System. Cílem bylo umožnit uživatelům aplikace si vybrat, který algoritmus bude řešit daný problém, aby mohli případně sledovat odlišné vlastnosti jednotlivých algoritmů. Neboť některé z těchto algoritmů se liší od jiných uvedených např. pouze způsobu, jakým si mravenci vybírají následující místo, které navštíví, je zde s výhodou využita dědičnost. Třídy obsahující implementaci algoritmů tak mohou sdílet některé metody, které jsou napříč více algoritmy shodné, prostřednictvím společné nadtrždy (např. `AntSystemCommon`).

Základní jednotkou provádění implementovaných algoritmů jsou iterace (cykly). Jedna iterace zahrnuje náhodné umístění mravenců, jejich průchod místy, určení nejlepší nalezené cesty a distribuce feromonu na cesty či jeho vypařování. Běh algoritmu nemůže být přerušen uprostřed iterace a tedy výsledná aplikace umožňuje krokovat s granularitou těchto iterací nebo pozastavit provádění algoritmu až po skončení aktuální iterace.

1.4 Uživatelské rozhraní

1.5 Testování

2 Uživatelský manuál

Závěr