

Druhá semestrální práce – problém obchodního cestujícího

Popis architektury

Aplikace se skládá z tříd pro zajištění načtení jednotlivých hospod ze souboru a ze složky tříd pro samotný genetický algoritmus pro zjištění nejkratší cesty. Třída program (main) pak využívá tyto objekty pro průběh a volání jednotlivých funkcí třídy Generation. Dalšími třídami pro funkčnost genetického algoritmu jsou Pair (dvojice rodičů pro křížení), Individual (jedinec, který obsahuje sekvenci hospod a výslednou vzdálenost danou výpočtem ze zadání), Helper (statická třída pro výpočty) a již zmíněná třída Generation. Generation obsahuje aktuální seznam jednotlivců v generaci a také nejlepšího jedince napříč všemi generacemi. Obsahuje také metody pro jednotlivé části genetického algoritmu.

Fitness funkce

Pro fitness funkci je zde asi pouze jediná možnost v našem případě. Proto je zde za fitness funkci vybrán výpočet celkové vzdálenosti všech hospod.

Funkce pro selekci

Pro selekci jedinců ke křížení je vybrána vlastní funkce. Jednoduchá a přitom efektivní, jedná se o souboj dvou náhodných jedinců, kde pouze jeden z dvojice vyhraje a postoupí tak k možnosti zkřížení. Tato funkce byla původně špatně pochopený inverzní turnaj, nicméně jsem se rozhodl ji v práci zanechat právě z důvodu efektivity. Funkce způsobí přežití nejlepšího jedince. Bylo testováno použití pravého inverzního turnaje i rulety, ale tyto metody nevedly k lepším výsledkům než v případě použití vlastní funkce.

Křížení

Pro křížení je zde dvoubodové křížení s náhodnou délkou zepředu a zezadu od 1 do 10 prvků. Poté je prohozeno prvních x prvků, aplikována korekce a následně je provedeno křížení z druhé strany, po němž následuje opět korekce.

Mutace

Mutace je v tomto případě jednoduché prohození dvou určitých hospod v sekvenci. Je zde 20% pravděpodobnost pro mutaci jednoho nebo druhého potomka. K mutaci dochází pouze u potomků.

Korekce

Ke korekci v této práci dochází po křížení, kdy se rozlišuje, zda se jedná o prohození prvků ze začátku nebo z konce. Pokud se jedná o korekci po křížení ze začátku, pak se prohodí první nalezený prvek, který je duplicitní v daném potomkovi s druhým potomkem. Pokud se jedná o křížení od konce, pak se nahradí druhý nalezený duplicitní prvek.

Nejlepší nalezený výsledek

Nejlepším výsledkem bylo nalezení cesty (čísla označují id hospody):

2 -> 1 -> 3 -> 4 -> 5 -> 35 -> 6 -> 33 -> 34 -> 30 -> 29 -> 28 -> 27 -> 26 -> 31 -> 32 -> 24 -> 25 -> 23 -> 18 -> 19 -> 20 -> 21 -> 22 -> 17 -> 16 -> 15 -> 14 -> 11 -> 10 -> 12 -> 9 -> 13 -> 8 -> 7

S celkovou vzdáleností 16,16km.

Jak aplikaci oživit

Aplikace je napsaná v jazyce C#. Proto je možné aplikaci spustit s jakýmkoliv IDE, které bude schopno pracovat s tímto jazykem. Nicméně v případě absence takového IDE, je zde hotový build aplikace ve složce se zdrojovými kódy.