

ReadMe

Kurze Beschreibung

Eigens geschriebene Methoden des Genetischen Algorithmus befinden sich zur Übersichtlichkeit im package `mulsp.solvers.gaMethods`, da wir mehrere Methoden z.B für die Mutation und Selektion geschrieben haben. Im package `mulsp.solvers.ga` befindet sich weiterhin die Klasse `Individual`, aber auch verschiedene Implementationen der Methode `solve()`. Jede Klasse in diesem package, außer `Individual`, stellt sozusagen eine Idee bzw. Optimierung durch das Aufrufen unterschiedlicher Methoden dar. So wurde z.B in der Klasse `GaSolverSelektion`, die Roulette-Selektion angewandt während bei den anderen Klassen die Wettkampf-Selektion eingesetzt wurde. Insgesamt wurden also verschiedene Möglichkeiten in der Selektion, Mutation und der Entscheidungsvariablen ausprobiert, um ein optimales Ergebnis zu schaffen.

Vom Bestergebnis: gesetzte Entscheidungsvariablen

- Gesetzte Werte:
 - o Populationsgröße: 250
 - o Kreuzungswahrscheinlichkeit: 0,7
 - o Mutationswahrscheinlichkeit: 0,3
 - o Anzahl der Eltern die in die neue Generation übernommen werden: 10
 - o Selektionsdruck/ Anzahl der Eltern die zur Reproduktion gewählt werden dürfen: 10
- Benutzte Methoden
 - o Template-Crossover
 - o Wettkampf-Selektion
 - o Swap-Mutation – mit stetiger Mutationswahrscheinlichkeit
 - o Delete-n-last-Schema
- Wurde mit dem `GaSolverWettkampf` erreicht

Allgemeine Anmerkungen

- Selektion
 - o Wettkampf-Selektion

Für die Reproduktion wird ein Matingpool erstellt, der die n besten Individuen der Elterngeneration besitzt. Selektiert wird zwischen den n besten Individuen aus dem Matingpool per Zufall. Mit Hilfe der Variable `selektionsdruck` lässt sich entscheiden, wie viele Individuen es in den Matingpool schaffen.

- Crossover

- Template-Crossover

Für das Crossover wurde das Template-Crossover gewählt, d.h es wurde ein Template erstellt vom Typ Individuum, das durch die Methode `initRandom()` einen zufällig bestimmten Genotyp besitzt. Außerdem wurden zwei Kinder/Individuen erstellt, die zu Beginn eines der beiden Elternteile reproduzieren.

Schließlich kann es durch die Einführung einer Kreuzungswahrscheinlichkeit sein, dass zu bestimmten Zeiten nicht gekreuzt wird. Durch das Template erhält das erste Kind für jede 1 in der Schablone das entsprechende Allel vom ersten Elternteil sowie für jede 0 das vom zweiten. Für das zweite Kind entsprechend vertauscht.

Da vom Template nicht die Fitness berechnet wird, werden diese Individuen nicht von den 400000 Individuen, die erstellt werden dürfen, abgezogen.

- Mutation

- Swap-Mutation

Bei der Swap-Mutation werden im Genotyp zwei Gene/Bits pro Zeile vertauscht. Dies ist allerdings abhängig von der Kreuzungswahrscheinlichkeit.

- Flip-Mutation

Bei der Flip-Mutation wird jede Zeile und Spalte des Genotyps eines Individuums durchlaufen und ein Bit je nach Kreuzungswahrscheinlichkeit geflippt.

- Ersetzungsschema

- Delete-n-last-Schema

Die n schlechtesten generierten Kinder werden durch Individuen aus der Elterngeneration ersetzt und n zufällig gewählter Eltern werden in die neue Generation übernommen. Dabei wird sichergestellt, dass keine Individuen gewählt werden, die schon zur Reproduktion gewählt wurden. Die Anzahl der übernommenen Eltern wird durch die Variable `anzahlKeepDelete` bestimmt.

Kurze Erklärung der packages & einzelner Klassen:

- `Mulsp.solvers.gaMethods:`

- Klassen:

- ✦ `Crossover.java`

- Beinhaltet die Template-Crossover Methode

- ✦ `Mutation.java`

- Beinhaltet die Swap-Mutation und Flip-Mutation
- Beinhaltet eine Methode mit der die Mutationswahrscheinlichkeit nach einer bestimmten Anzahl an Individuen (bestimmt durch die Variable

prozentVariationPmut) randomly wechselt, aber immer höher ist als die Mutationswahrscheinlichkeit, die zu Beginn gesetzt wurde.

✦ Replace.java

- Beinhaltet die Delete-n-last Methode für das Ersetzungsschema

✦ SearchMin.java

- Beinhaltet die Methoden zur Suche nach dem Minimum und dem Decodieren und Evaluieren der Individuen

✦ SelektionRoulette.java

- Beinhaltet die Methode zur Roulette-Selektion

✦ SelektionWettkampf.java

- Beinhaltet die Methode zur Wettkampf-Selektion ✦ Extras.java
- Wird nicht benutzt, beinhaltet aber eine Methode, mit der die Eingabe der Entscheidungsvariablen durch die Konsole ermöglicht wird.