

iMOPSE ga-greedy runner - User's Manual

Maciej Laszczyk, Paweł B. Myszkowski

maciej.laszczyk@pwr.edu.pl; pawel.myszkowski@pwr.edu.pl

Project page: <http://imopse.ii.pwr.wroc.pl/>

Wrocław University of Technology

April 17, 2017

Introduction

This document describes what the iMOPSE genetic algorithm runner is and how to use it.

iMOPSE genetic algorithm runner is an open source tool released for student and researchers to quickly obtain a solution to the Multi-Skill Resource-Constrained Project Scheduling Problem (MS-RCPSp). Genetic Algorithm does not ensure that the results are optimal but the research has shown that they tend to be better than achieved with a greedy approach and the time required to create them is acceptable for the end user.

Genetic algorithm needs two genetic operators to work: crossover and mutation. Crossover is performed by swapping genetic material between individuals. The simplest way to achieve it is by setting a 'slice point'. For example having two parent individuals: $P1 := (1, 2, 3, 4, 5)$ and $P2 := (6, 7, 8, 9, 10)$ and setting a slice point at the third position, we get two child individuals: $C1 := (1, 2, 3, 9, 10)$ and $C2 := (6, 7, 8, 4, 5)$. In case of mutation, the operator works on a single individual and in the simplest approach randomizes the value of a each gene with given probability. For example, having an individual: $I1 := (1, 2, 3, 9, 10)$ and randomizing its third gene, the individual can mutate into $I1' := (1, 2, 10, 9, 10)$.

Implemented genetic algorithm allows for control of a criteria weight. With it, the user can decide which criteria (duration or cost) is more important. The parameter takes values between 0.0 (for cost optimization) and 1.0 (for duration optimization) and allows for a fluid control of the criteria importance.

The runner will always create a valid schedule, which does not violate any of the constraints.

Technical Requirements

To use the iMOPSE genetic algorithm runner, Java Runtime Environment (version 1.8 or newer) has to be provided .

How to use the Genetic Algorithm Runner - step by step

To use the iMOPSE genetic algorithm runner, a runner .jar file along with the problem definition file is required.

There are ten parameters:

- population_size - number of individuals in the population
- generation_limit - maximum number of generations to create
- evaluation_rate - weights assigned to cost and duration criteria (0.0 - cost only, 1.0 - duration only)
- mutation_probability - probability, with which to mutate each gene
- initial_population_type - currently only RANDOM available
- selection_type - currently only ROULETTE_WHEEL available
- crossover_type - currently only SINGLE_POINT available
- mutation_type - currently only RANDOM available
- schedule_builder_type - FORWARD_SCHEDULE_BUILDER or BACKWARD_SCHEDULE_BUILDER
- definition_file - file path to the .def file

Using the runner works similar to any .jar file:

```
java -jar GARunner.jar population_size generation_limit evaluation_rate  
mutation_probability initial_population_type selection_type crossover_type mutation_type  
schedule_builder_type definition_file
```

Assuming that the definitions file are located inside the definitions directory the following command could be used:

```
java -jar GARunner.jar 100 100 1.0 0.1 RANDOM ROULETTE_WHEEL SINGLE_POINT  
RANDOM FORWARD_SCHEDULE_BUILDER definitions\100_10_26_15.def
```

The command results in the following:

Assignments	Conflicts	Relations	Skills
SUCCESS	SUCCESS	SUCCESS	SUCCESS

Duration: 289
Cost: 128554.39999999997
Time elapsed: 1.551222429

First two lines inform about the validity of the resulting schedule, each criteria constraint is displayed along with the information whether it is violated (FAILURE) or not

(SUCCESS). Next two lines show the duration and the cost of the schedule respectively. Last line show time in seconds that was required to obtain the result.

Reference article

Myszkowski P.B., Laszczyk M., Nikulin I., Skowroński M.E. "iMOPSE: a library for bicriteria optimization in Multi-Skill Resource-Constrained Project Scheduling Problem", in review process, Soft Computing Journal.