Temasek Defence Systems Institute

Task Allocation With Ordering Constraints

Cheng Ze Wei
Supervisors: Prof Antonios Tsourdos & Dr Seo Minguk

Introduction

Objective of the project is to develop a task allocation algorithm that assigns tasks while:

- Maximises rewards
- Respect ordering and operational constraints, and
- Complete in a reasonable amount of time

Greedy Algorithm

Greedy algorithm was chosen as a comparison algorithm. Although it might only provide local solution optima, it is easy to implement and is fast.

Greedy algorithm was designed using the same heuristic function in genetic algorithm, except that:

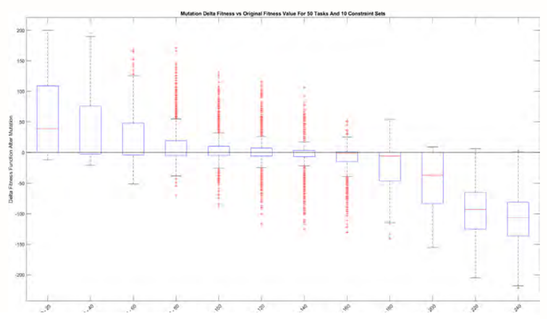
- It is sequential
- Boost selection functions when agents have limited feasible tasks

Genetic Algorithm

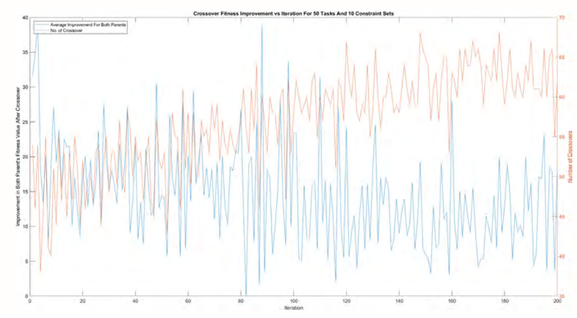
Despite the higher computational time required, genetic algorithm was chosen due to its suitability to accommodate constraints and likelihood of attaining global optimal solutions to complicated problems.

Genetic algorithm was designed to have:

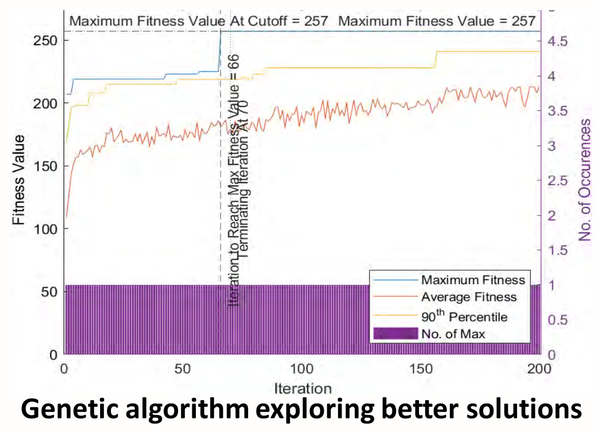
- Random and non-random initial population
  - Some tasks are chosen using a heuristic function
- Dynamic population size based on variance
- Dynamic crossover probability based on variance, skewness, and iteration
- Handle constraints using:
  - Eliminating infeasible solutions
  - Penalising selection function
  - Repairing infeasible solutions
- Time-based single point crossover with agent cross pairing



Mutation performance



Crossover performance



Genetic algorithm exploring better solutions

Results

- Taguchi method improved initialisation by 24.3%
- Non-random initialisation method performed better (+50%) at a cost of more time (+50%)
- Crossover was most effective at the beginning
- Mutation was most effective on weak chromosomes
- Genetic algorithm took significantly longer time
- Initially:
  - Genetic algorithm produced better solution
  - Greedy algorithm's solution deteriorated with complicated test cases

Before	Greedy		Genetic	
Test	Reward	Time (s)	Avg. Sol.	Time (s)
A	218	0.796	218	823
B	119	1.61	212 **(+78.2%)**	1,790
C	106	6.57	224 **(+111%)**	12,800

- After improvement to heuristic function (used in both):
  - Computational time for both algorithm remained similar
  - Solutions for both improved
  - Greedy algorithm produced better solutions
  - Due to more constraint tasks being covered
  - Updated heuristic was able to describe the complex problem sufficiently

After	Greedy		Genetic	
Test	Reward	Time (s)	Avg. Sol.	Time (s)
A	218	0.76	218	836
B	281	1.79	246 **(-12.5%)**	2,220
C	378	8.65	307 **(-18.7%)**	9,442

Conclusion

- Genetic algorithm was able to explore better solutions with time
- However, the impact of constraints on computational time dominated, preventing genetic algorithm from arriving at global optimum solution in a reasonable amount of time
- Improved heuristic allow greedy algorithm to produce good local optimum solutions

Future Works

- Enhancing aggressiveness of mutation operator
- Tuning on probability of non-random initialisation in genetic operators
- Investigate effect of proportion of non-random initialisation and initial population sizing criteria
- Look into and investigate effectiveness of distributed auction method

NUS National University of Singapore