Using a Genetic Algorithm to resolve a Roster Problem

PHD Modelling Week
Verona, September 2016
Alessandro Mella
Ihor Sharmanskyi
Ben Paoletti

Plan

- Problem description
- Genetic algorithm
- Implementation techniques
- Results discussion

The Real Word problem

- Cyclic weekly standard Roster base on the daily shift requirement and the labor contract

  - Cyclic weekly roster:
  - Daily shift requirement
  - Labor contract

Shift requirements

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
W_1	M	M	M	M	M	M	M
W_2	A	A	A	A	A	A	A
W_3	N	N	A	N	N	N	N
W_4			N			N	

Cyclic roster

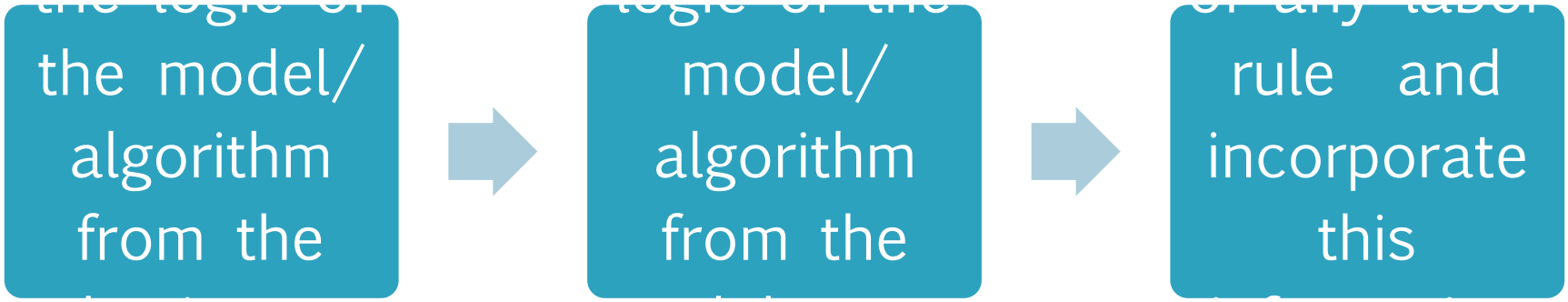
	Mon	Tue	Wed	Thu	Fri	Sat	Sun
W_1	A	M	A	M	N	N	R
W_2	M	A	M	R	A	N	A
W_3	R	N	A	A	R	A	N
W_4	N	R	N	N	M	M	M

Contract

- Rule 1
- Rule 2
- Rule 3
- …

Further Constraint

1.



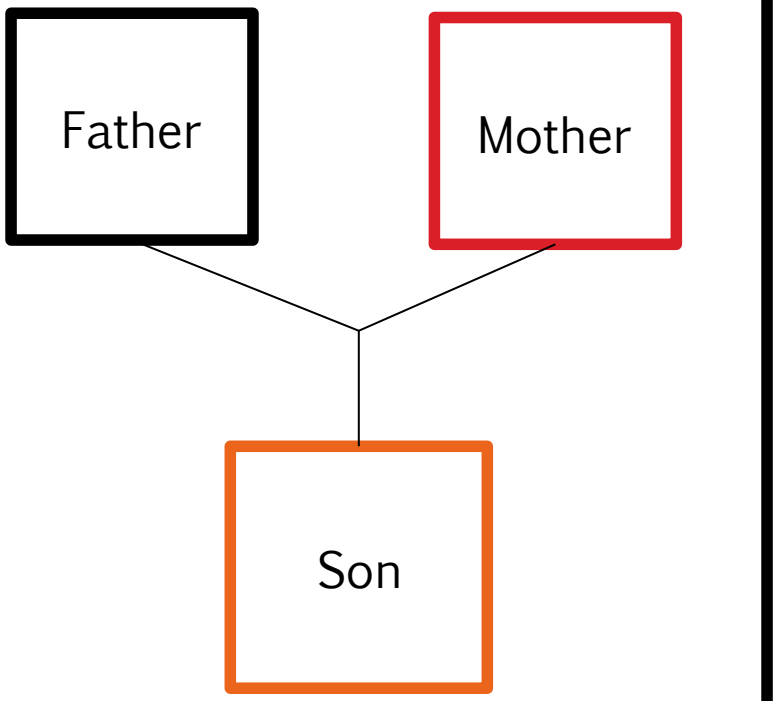
Goal:create the roster which satisfies all the demands.

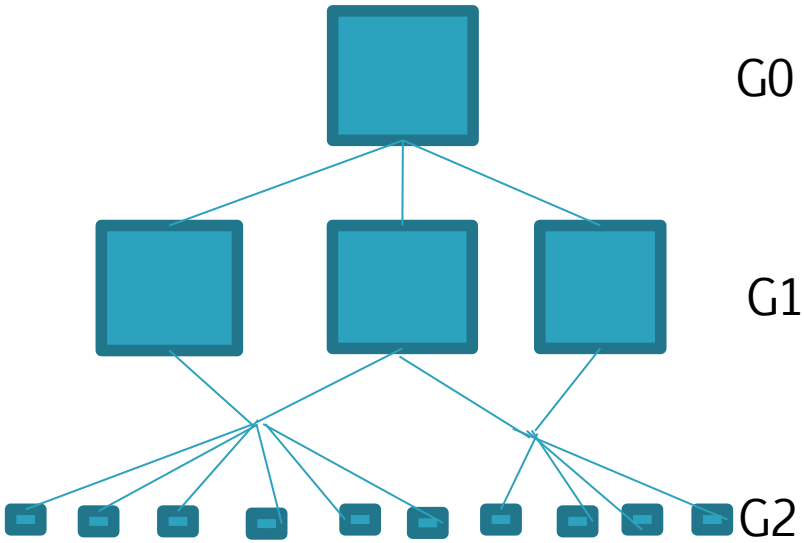
Method: Genetic algorithm

Goal:create the roster which satisfies all the demands.

Method: Genetic algorithm

Genetic algorithm





G0: Initial solution
G1: Mutations of G0
G2: Inheritance and mutations of G1

The Genetic Algorithm Family

Start with an initial population of individual and their survival probability

Do

based on their survival probability, select a couple of individual (father and mother)

From selected mather and father generate a new individual (son) by the crossover function

Simulate a mutation function into the son

Evaluate the son by a fitness function

Insert the son into the new generation of the population based on her/his survival probability and the limit of the population

Loop While The stop critera occurs

Genetic Framework for Roster

- Individual
  - a possible solution
- Crossover
  - Combine two solution taking into account the requirement covering constraint
- Mutation
  - Swap the daily shift between two logical employee
- Fitness
  - Evaluate a possible solution from the labor contract rules viewpoint

Genetic Framework for Roster

Individual

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
W_1	A	M	A	M	N	N	R
W_2	M	A	M	R	A	N	A
W_3	R	N	A	A	R	A	N
W_4	N	R	N	N	M	M	M

Genetic Framework for Roster

Crossover

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
W_1	N	M	A	M	N	N	R
W_2	A	A	M	R	A	N	A
W_3	R	N	A	A	R	A	N
W_4	M	R	N	N	M	M	M

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
W_1	A	M	M	A	N	A	A
W_2	M	N	A	N	A	N	N
W_3	R	A	N	R	M	N	M
W_4	N	R	A	M	R	M	R

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
W_1	N	M	A	M	N	A	A
W_2	A	A	M	R	A	N	N
W_3	R	N	A	A	R	N	M
W_4	M	R	N	N	M	M	R

Genetic Framework for Roster

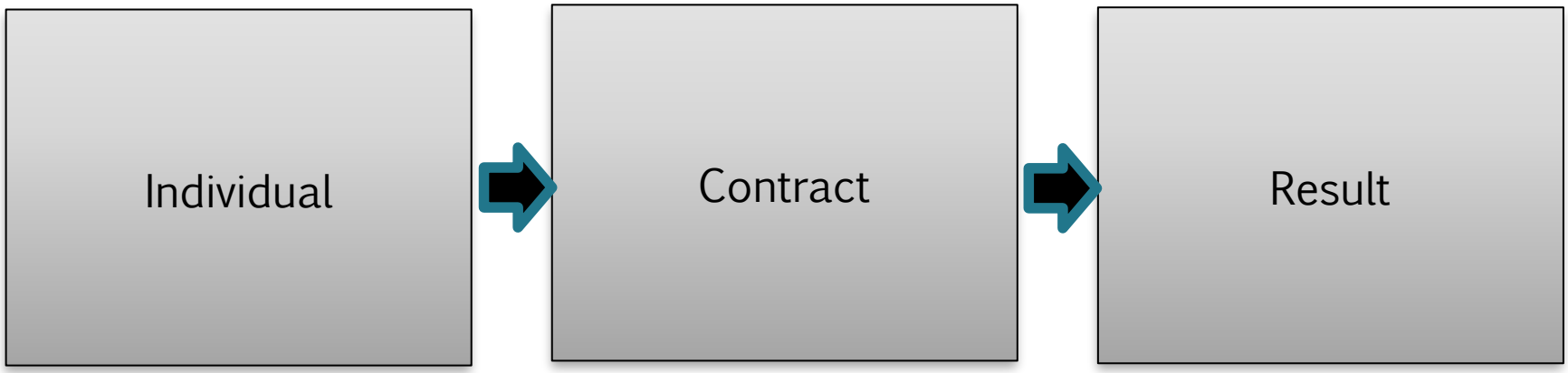
Mutation

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
W_1	A	M	A	M	N	N	R
W_2	M	A	M	R	A	N	A
W_3	R	N	A	A	R	A	N
W_4	N	R	N	N	M	M	M

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
W_1	A	M	A	M	N	N	R
W_2	M	A	A	R	A	N	A
W_3	R	N	M	A	R	A	N
W_4	N	R	N	N	M	M	M

Genetic Framework for Roster

Fitness



$$Result = \prod_{i=1}^{|Rules|} \frac{Num\ of\ correct\ squares}{Max\ square\ number}$$

$$Result \in [0,1]$$

Advance Topics

- Enrich the model/algorithm with a more efficient crossover function
  1. Two sons
  2. Horizontal Crossover
  3. Block Crossover
  4. Random Crossover

  etc

- Enrich the interaction between the model/ algorithm and the Labor Contract

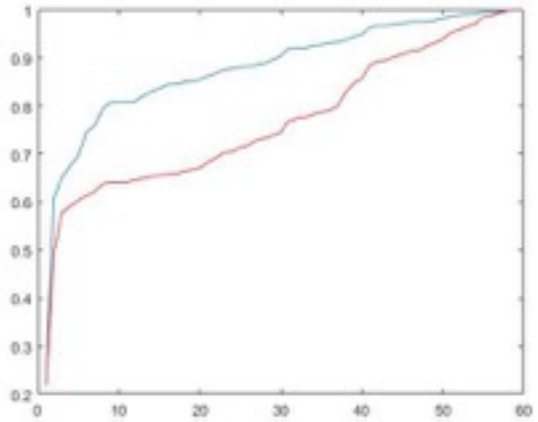
  Error matrix

Results

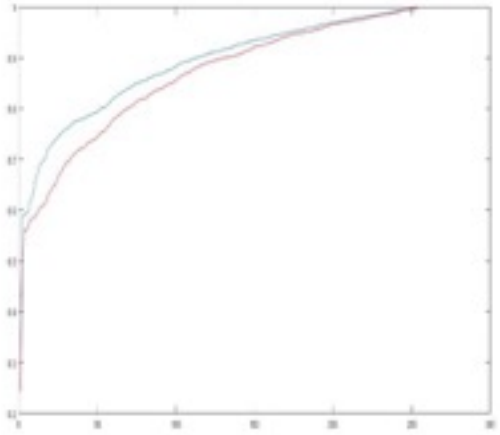
Matrix	Initial Population	Max Population	Generations	Start error	Time (seconds)
100*7	10	200	59	0.2	84
300*7	10	200	149	0.25	625
500*7	10	200	255	0.24	1724
1000*7	10	200	607	0.25	8626

Demo

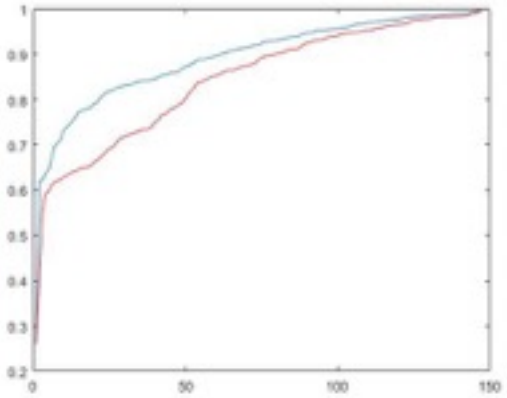
100



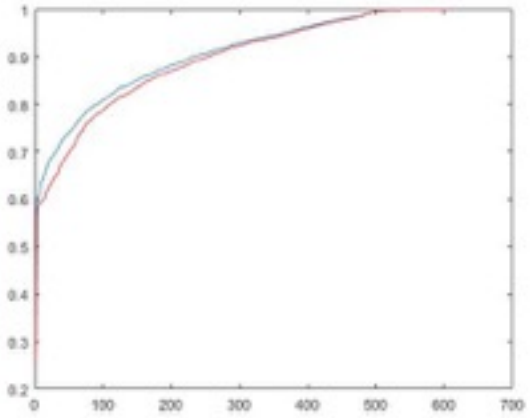
500



300



1000



MAX

AVG

Bibliography

- Ernst, Andreas T., et al. "Staff scheduling and rostering: A review of applications, methods and models." *European journal of operational research*153.1 (2004): 3-27.
- Kim, Sun-Jeong, et al. "A strategy to improve performance of genetic algorithm for nurse scheduling problem." *Int. J. Soft. Eng. Appl* 8.1 (2014): 53-62.
- Kim, Sun-Jeong, et al. "An Efficient Method for Nurse Scheduling Problem using the Genetic Algorithm." *ASTL* 23 (2013): 252-255.
- Morz, M., and N. Musliu. "Genetic algorithm for rotating workforce scheduling problem." *Computational Cybernetics, 2004. ICCC 2004. Second IEEE International Conference on*. IEEE, 2004.
- Puente, Javier, et al. "Medical doctor rostering problem in a hospital emergency department by means of genetic algorithms." *Computers & Industrial Engineering* 56.4 (2009): 1232-1242.