Problem 1

Stojanova A, Kocaleva M, Bikov D, K Lazarova L, Stojkovikj N, Miteva M,
Zlatanovska B, Zdravevska S, Ljubenovska M

Problem description

- Kindergartens and day care centers face with a problem when the employees should be organized throughout the working day, or only partially included in a certain period of the day.
- It is necessary to make a schedule of employees in the kindergartens or in a day care center for children.
- Our aim
  - make an optimization model to deal with the large number of children and employees.
- This problem can be extended to problem for healthcare services for elderly people.

Problem description

Multi Age Preschool Daily Schedule

7:30 -9:00	Drop Off / Early Morning Free Play Activities
9:00 - 9:30	Bathroom Break and Breakfast Time
9:30 - 10:00	Teacher Directed Play
10:00 - 10:30	Group Time - (Children are separated by age during this time)
11:00	Bathroom Break
10:30 - 11:30	Curriculum Activities: Art, Music, Sensory/Science, Fine Motor, Writing Center, Block Area, Dramatic Play, Book Area
11:30 - 12:25	Clean Up and Outside/Large Motor Play (weather permitting)
12:30 - 1:00	Lunch Time / Bathroom Breaks after Lunch
1:00 - 2:55	Nap/Rest Time (quiet activities provided for those awake after a half hour rest)
3:00 - 3:30	Snack Time / Bathroom Break
3:30 - 5:00	Parent Pick-Up / Kids Choice Free Play Activities. (weather permitting we will go outside after snack)

General model proposed by Marta Ferreira (doctoral thesis-2013)

- The general model was developed for the staff scheduling problem of an organization that works continuously, 24 hours a day.
- The day is divided in n_s working shifts. The model considers a set of n_T teams of homogeneous (single skilled and full-time) employees, that must be assigned to either a work or a break shift, in each of the n_D planning period days. Daily shift demand levels must be satisfied, meaning that the model must guarantee a required number of teams working in each shift on each day.
- $d \in \{1, \ldots, n_D\}$, day;
- $t \in \{1, \ldots, n_T\}$, team;
- $s \in \{1, \ldots, n_S\}$, working shift;
- $s' \in \{1, \ldots, 2 \times n_S\}$, extended shift.

This model can be used for healthcare services to elderly people.

Mathematical models for scheduling problem

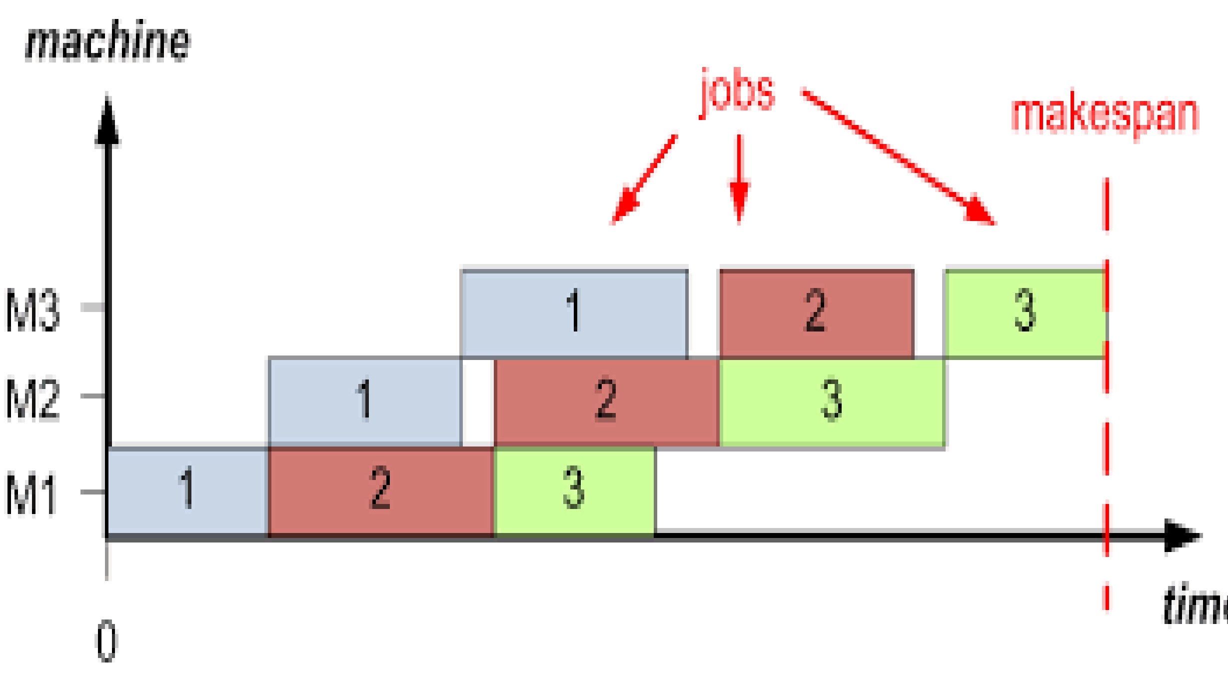
- Base Model (FSMP)
- Single machine model
- Parallel machine model
- Flow shop model
- Job shop model
- Pure job shop model

Proposed models

- Job shop scheduling model
  - Optimization of time needed for completing all tasks
  - Agent based approach with AnyLogic
- Genetic algorithm
  - Used alone or in combination with JSS

Job shop scheduling model

- This model can be described by a set of n jobs J_j, where 1≤j≤n, and each job has to be processed on a set of m machines M_r, where 1≤r≤m.
- Each job has a sequence of machines that must be processed.
- The processing of job J_j on machine M_r is called the operation O_{jr}. Operation O_{jr} requires the exclusive use of M_r for an uninterrupted duration p_{jr}, where p_{jr} is its processing time.
- A schedule is a set of completion times for each operation MS_{jr}, where 1≤j≤n and 1≤r≤m that satisfies those constraints.
- The time required to complete all the jobs is called the makespan MS.
- The scheduling objective is makespan minimization, which means to minimize the completion time of the last operation of any job.



Job shop scheduling model

In our problem we have two sets

- E = { E1, …. E_m} where E are employees (machine of the model)
- K = { K1, …. K_m} where K are kids or elderlies (jobs of the model)

- The aim is to minimize the time to perform all tasks (all needs of the kids or all needs of elderlies in a shortest possible time)
- For the assignment problem of kids or elderlies to employs we use MIP (mixed integer programming) model.
- The MIP formulation is often used to model the classical deterministic JSSP

Job shop scheduling model

- **MIP model**
  - *Parameters:*
- r_{ilk} with value 1, if elderly (kid) *i* requires task *l* from employ *k* , and 0 otherwise.
- p_{ik} is servicing time in which an elderly (kid) *i* has to be serviced from employ *k*.
  - *Decision variables:*
- s_{ik} is start time of servicing an elderly (kid) *i* by employ *k*.
- y_{ijk} has a value 1 when an elderly *j* precedes elderly *i* for caregiver *k*.

Job shop scheduling model

- **MIP model**

$$\sum_{k-1}^{m} r_{imk}(s_{ik} + p_{ik}) \leq MS \qquad i = 1,2,\ldots,n \tag{1}$$

$$\sum_{k=1}^{m} r_{ilk}(s_{ik} + p_{ik}) - \sum_{k=1}^{m} r_{i,l+1,k}\, s_{ik} \leq 0,$$

$$i = 1,2,\ldots,m; \tag{2}$$

$$l = 1,2,\ldots,m-1;$$

$$K(1 - y_{ijk}) + s_{jk} - s_{ik} \geq p_{ik},$$

$$k = 1,2,\ldots m; 1 \leq i < j \leq n \tag{3}$$

$$Ky_{ijk} + s_{ik} - s_{jk} \geq p_{jk}, \quad k = 1,2,\ldots m; 1 \leq i < j \leq n \tag{4}$$

Constraint 1 gives the lower bound for the function *MS*.

Constraint 2 ensures that the starting time of servicing an elderly *i* with task *l* + 1 is not earlier than its finish time in its predecessor, task *l*.

Constraints 3 and 4 ensure that only one elderly is served from employer at any given time. The parameter *K* is a large number, sometimes taken as the sum of all processing times.

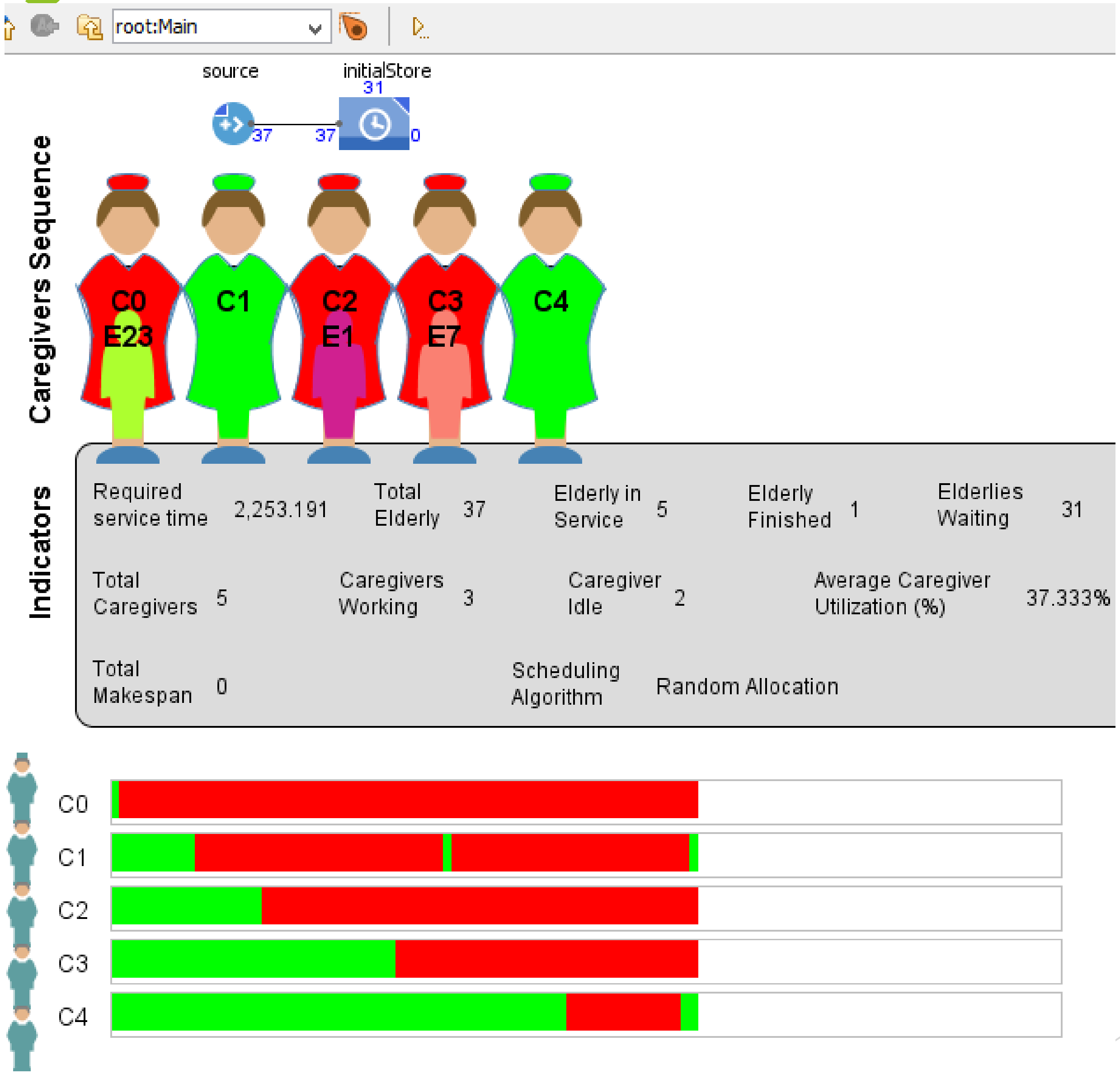
Agent based approach of the problem

- Agent-based models (ABMs) consist of a set of elements (agents) characterized by some attributes, which interact each other through the definition of appropriate rules in a given environment.
- Our agent based approach of the problem is consist of two types of agents.
  - kids (elderly people or patients)
  - employees (caregivers).

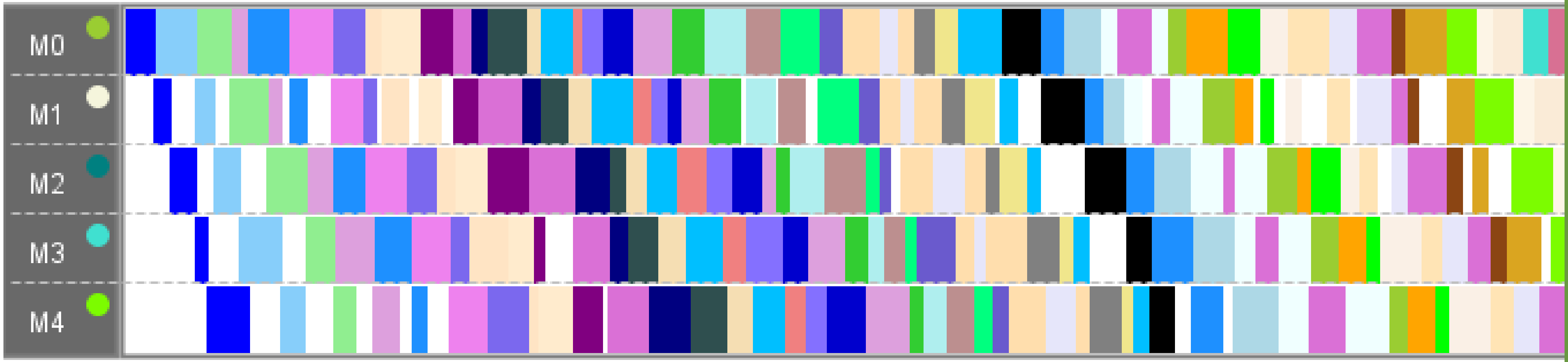
  They communicate among each other in a manner that employees can provide different services to elderlies or kids.

This approach is applied in combination with discrete event simulations

AnyLogic Simulation



AnyLogic Simulation

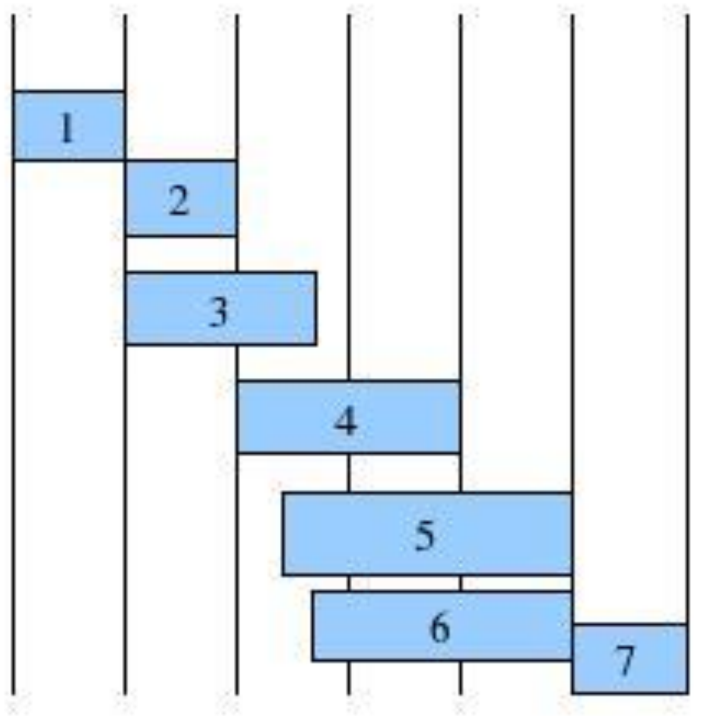


	J0	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15	J16	J17
M0	19.098	16.558	12.847	7.771	8.264	14.787	19.985	19.436	8.07	14.562	19.807	14.257	17.745	9.069	16.457	8.697	10.537	14.66
M1	19.208	14.898	16.16	5.16	11.48	19.765	14.456	8.142	8.824	5.131	7.299	5.422	10.307	15.947	11.416	8.727	11.958	6.657
M2	19.056	7.351	7.13	7.416	8.497	8.101	18.644	7.59	16.669	14.468	15.659	13.872	13.979	18.301	19.988	15.603	7.659	16.619
M3	10.958	10.673	12.226	7.671	18.349	10.62	12.615	13.234	8.375	8.12	17.36	6.095	9.381	5.656	5.929	10.066	15.142	9.597
M4	10.213	7.096	13.168	13.106	5.575	11.95	12.372	13.332	19.747	18.205	6.855	6.111	18.147	11.866	13.795	14.11	15.734	18.209

Approach with Genetic algorithm

- Genetic algorithm can be used in combination with JSS
- To apply a genetic algorithm to a scheduling problem we must first represent it as a chromosome.

  - Implements each schedule as a chromosome/individual in a population of schedules.
  - Each schedule is evaluated with a fitness function
  - Schedules with greater fitness function values are allowed to "mate" with other schedules via crossover.
  - Mutation provides for diversity in the population.
  - The crossover and mutation operations generate new populations of schedules.
  - New generations are created until a schedule is formed that is deemed acceptable.
  - Generally speaking, a schedule is deemed acceptable if its fitness function value is high enough.



Future work

- Try to find application of General model and adapt it for our problem.
- To optimize our proposed approach and to adapt to other constraints needed.
- To extend proposed model with genetic algorithms.