Simulation and Optimization for Decision Support (G54SOD)

Semester 2 of Academic Session 2017-2018

Dr Dario Landa-Silva
dario.landasilva@nottingham.ac.uk
http://www.cs.nott.ac.uk/~pszjds

Workshop 8 – Optimization With ABM Simulation

- Overview of Office Space Allocation Research

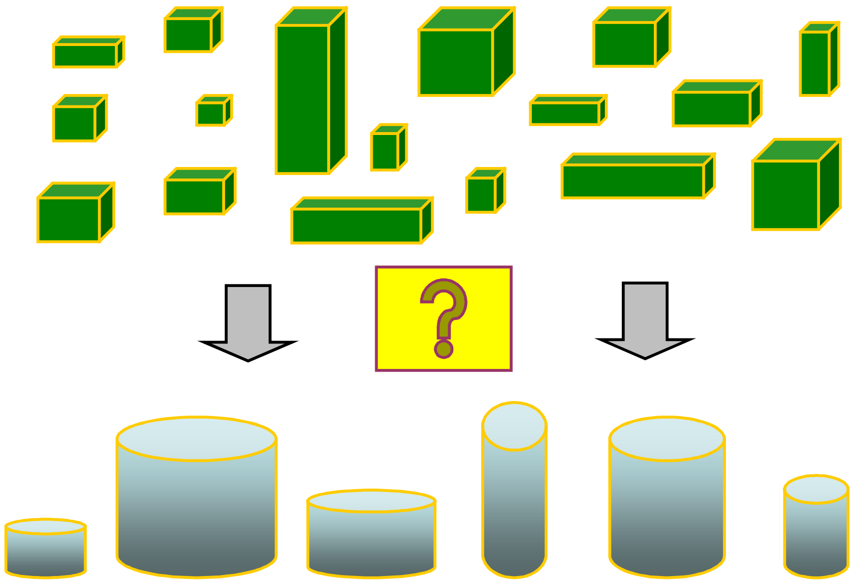
  Describe the OSA optimization problem and outline various optimization approaches that have been applied to solve this combinatorial problem.

- Solving OSA With ABM Simulation

  Outline preliminary results on recent research applying ABM simulation to solve the OSA problem.

Office Space Allocation (OSA) Problem

The problem of allocating office space to entities so as to ensure the optimal utilisation of the space and the satisfaction of additional requirements and constraints. The office space consists of rooms, hallways, etc. and the entities are people, machines or 'functionalities' (e.g. meeting room or stationery room). This problem is related to other combinatorial optimisation problems such as bin packing and multiple knapsack problems.



Office space is a premium commodity in organisations.

Office space allocation is a continuous process in organisations.

Quality of Solutions in OSA

The primary goal is to maximise space utilisation by reducing the misuse of space which refers to underusing space (under utilisation of the room capacity) or overusing space (exceeding the room capacity by overcrowding). Generally, overuse is considered worse than underuse but this depends on the scenario.

Space Underuse

Space Overuse

All entities must be allocated.

Additional goals include the satisfaction of various other requirements and constraints.

- Specific allocations (entity → room)
- Entity sharing or not a rooms
- Entities not to be in the same room
- Entities to be grouped together
- Entities to be adjacent to each other
- Entities to be away from each other

Solution Representation

There are E entities $e_1, e_2, e_3, \ldots, e_{|E|}$ to be allocated

There are R rooms r_1, r2, $r_3, \ldots, r_{|R|}$ available

An allocation is represented as an array of size $|E|$

The index of the array corresponds to the entity

e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	…	$e_{|E|}$
r_5	r_7	r_1	r_9	r_2	r_5	r_3	r_8	…	r_6

The content of the array indicates the room assigned to the entity

A solution is a combination of values 1 to $|R|$

This direct solution encoding ensures all entities are allocated

Handing the other requirements (soft constraints) and (hard) constraints needs additional mechanisms

Measuring Diversity in a Population of Solutions

A population of p = 5 solutions with |E|= 7 and rooms A, B, C, etc.

Diversity measurement based on Hamming distance.

	Five strings representing allocations						
	A	A	A	A	A	A	A
	A	A	B	B	A	B	B
	A	B	B	C	B	C	C
	A	B	B	C	B	D	D
	A	B	B	C	C	D	E
$D(j)$	1	2	2	3	3	4	5
$(D(j) - 1) / (p - 1)$	0	0.25	0.25	0.50	0.50	0.75	1

$V(p) = (3.25 / 7) \times 100 = 46.42$ %

$$V(p) = \frac{\sum_{j=1}^{n} \frac{D(j)-1}{p-1}}{n} \cdot 100$$

- D(j) is the number of unique values in the j^{th} position of the solution
- n = |E| is the length of the string
- V(p) represents the variety or diversity in the population

The Evaluation Function

Minimize the space misuse penalty SMP

- Space is overused is the room r_i capacity is smaller than the total space used by all entities allocated to the room
- Space is underused is the room r_i capacity is greater than the total space used by all entities allocated to the room

$$SMP = \sum_{i=1}^{|R|} max((cap(r_i) - usg(r_i), 2 \times (usg(r_i) - cap(r_i)))$$

Minimize the penalty for not satisfying the soft constraints (requirements)

- Each of the $|C_{soft}|$ soft constraints has a weight or individual penalty $w(c_i)$
- The satisfaction of soft constraint ci is indicated by $v(c_i) = 0$, otherwise $v(c_i) = 1$

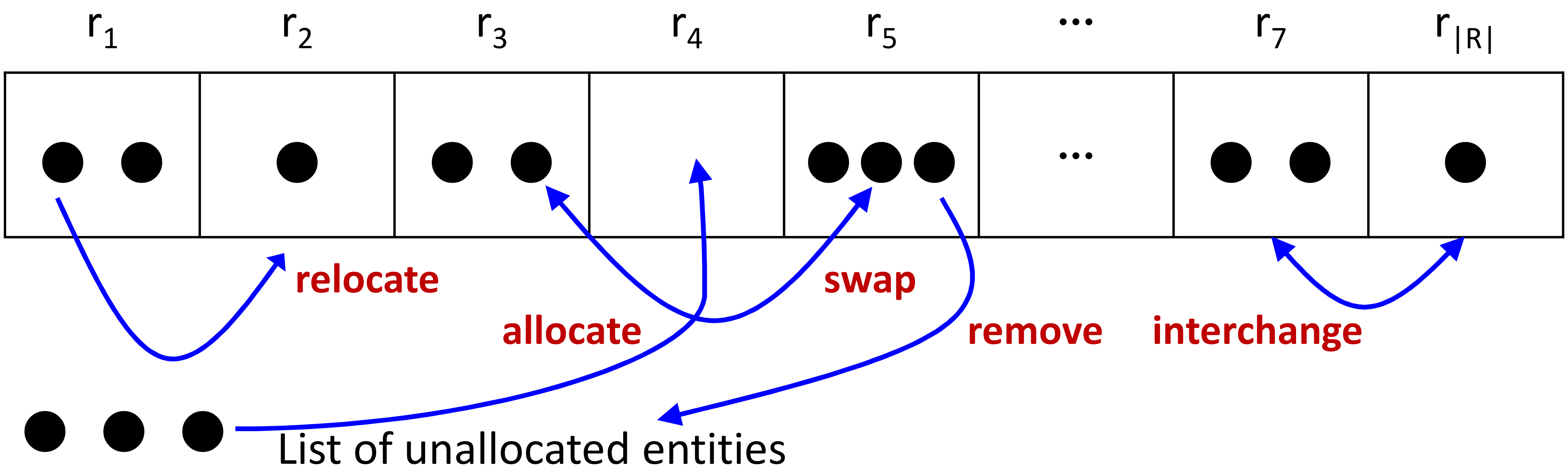
$$SCP = \sum_{i=1}^{|C_{soft}|} (w(c_i) \times v(c_i))$$

Types of Soft Constraints

Allocation To, Non-Allocation To, Same-Room, Not-Same-Room, Not-Sharing, Adjacent To, Nearby To, Away-From, etc.

Neighbourhood Operators or Moves

These operators can be used to generate an initial allocation and also to perform local search to improve the existing solution. The heuristic search can be conducted while maintaining feasibility or not.

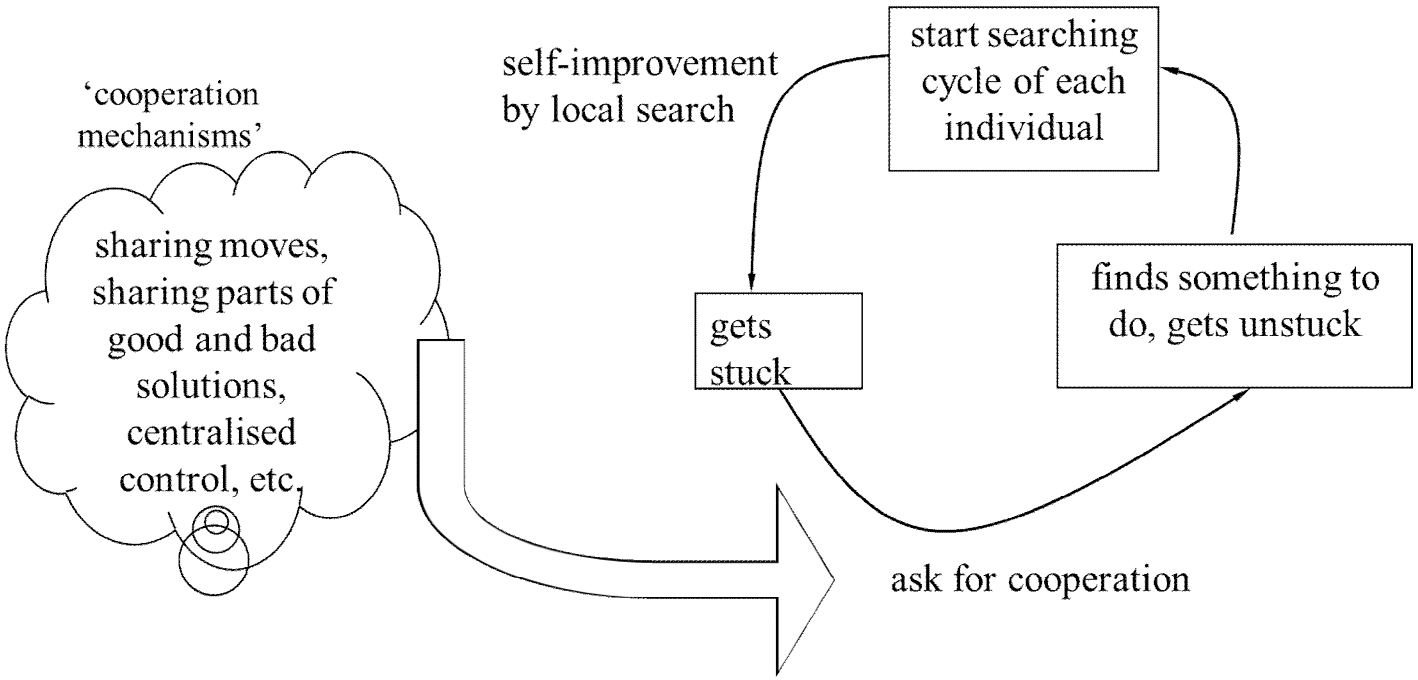


Note that this illustration uses an array on length |R| and not the solution encoding presented before using an array of length |E|.

Various Exact and Heuristic Algorithms Applied to OSA

- Tabu Search
- Hybrid Meatheuristic
- Population-based Local Search
- Mathematical Programming
- Hill Climbing
- Simulated Annealing

The asynchronous cooperative local search scheme



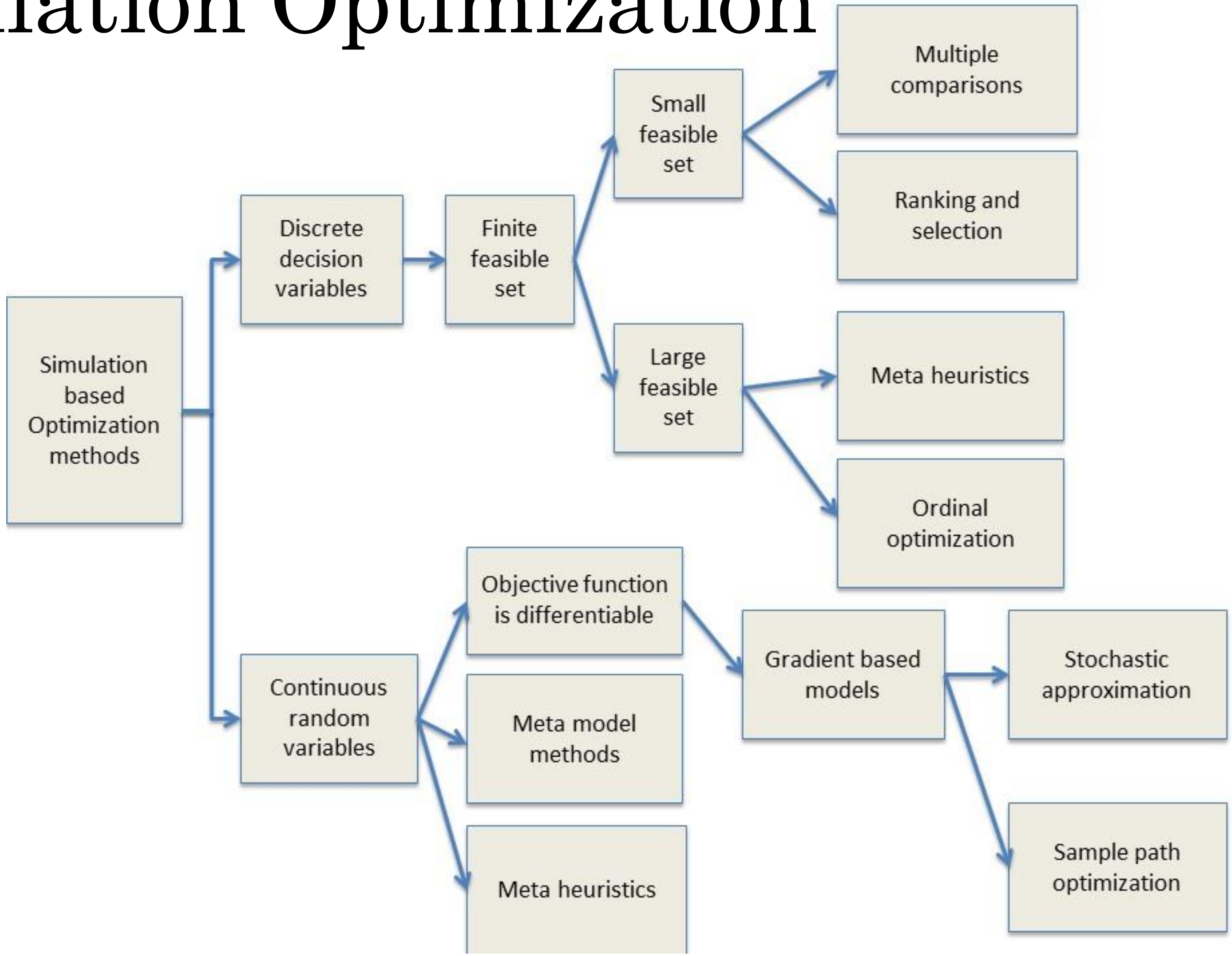
Agent Based Simulation for Office Space Allocation Problem

Alexandra Cristina Dediu

Simulation Optimization

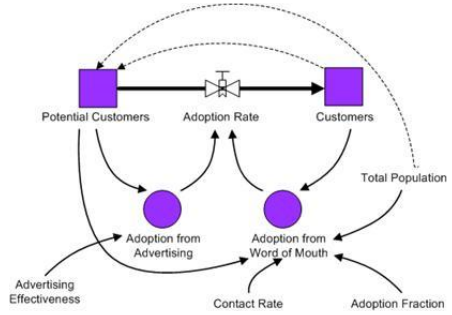
- Simulation community – use optimization to improve the simulation models
- Simulation driven approach
- Evaluate the effect of different values of input parameters on a target objective (what-if analysis)
- Return best values for decision variables according to the combination of the criteria set and the data obtained during the simulation run

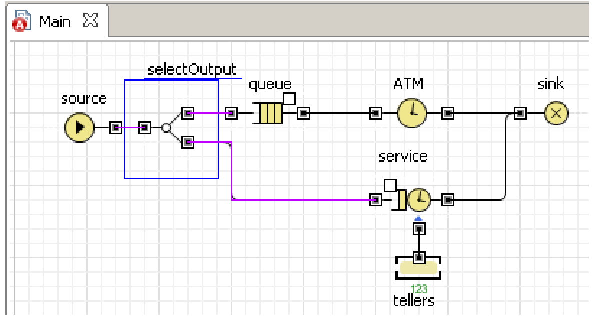
Simulation Optimization



Simulation-Modelling Paradigms

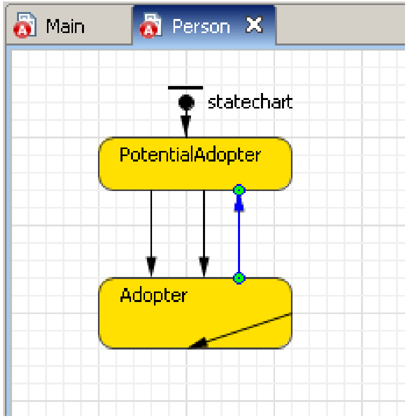
- System Dynamics (SD)
  - Model stock and flow diagrams





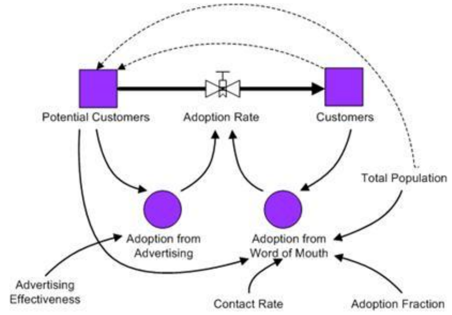
- Discrete Event Modelling (DEM)
  - Model flow charts

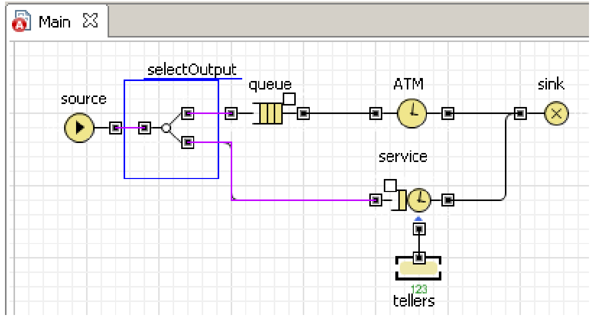
- Agent Based Modelling (ABM)
  - Model equations or state charts



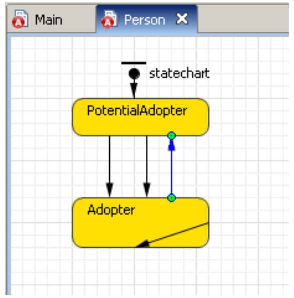
Simulation-Modelling Paradigms

- System Dynamics (SD)
  - Model stock and flow diagrams





- Discrete Event Modelling (DEM)
  - Model flow charts



- Agent Based Modelling (ABM)
  - Model equations or state charts
  - A collection of autonomous decision making entities called agents
  - Typical components: agents, decision making heuristics, learning rules, interaction topology and an environment.

ABM Applications in Optimization

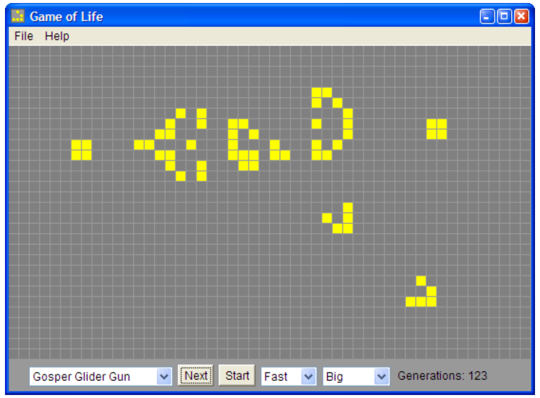
- Agent based approaches for scheduling, transportation, supply chain planning, facility location, bin packing etc.
- Main advantage of ABM – problem decomposition and delegate sub problems to agents

ABM	**Classic optimization**
• large problem	• communication has a high cost
• short time	

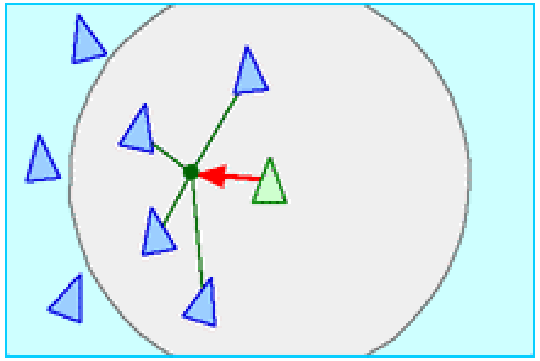
- Better – use a hybrid– **add optimization to agents during their decision making process**

ABM Examples

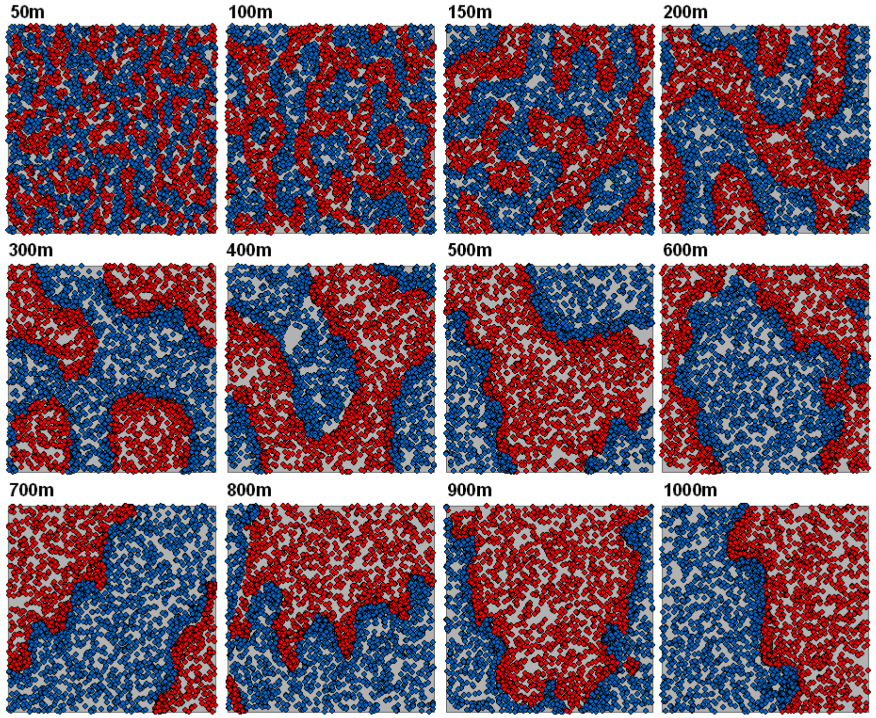
- John Conway – 'Life'



- Craig Reynolds – 'Boids'
  - flocking behaviour of birds



- Thomas Shelling – 'Shelling Segregation Model'



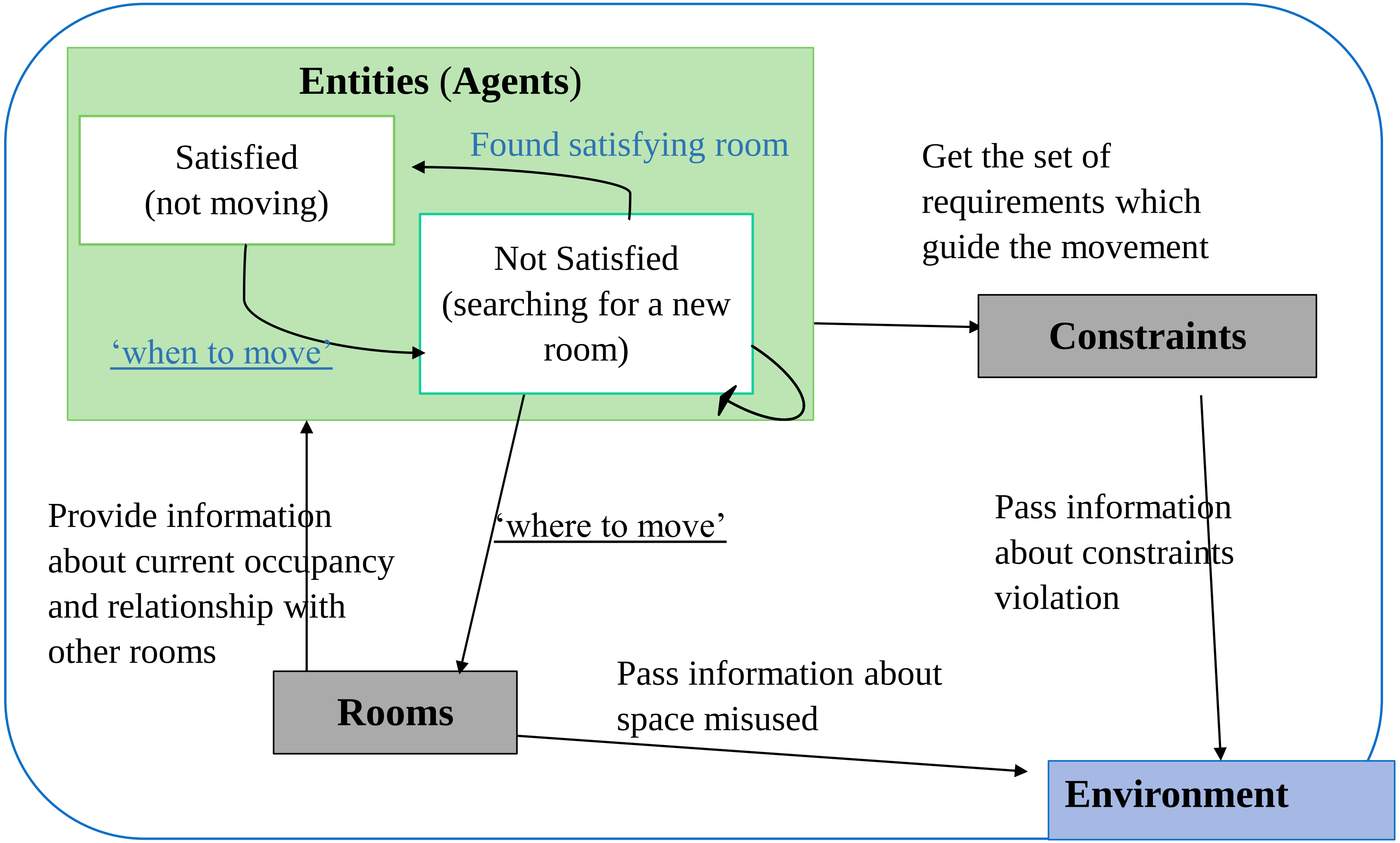
Office Space Allocation

- Set of resources – rooms (with a specified space/capacity and neighbourhood relationships)
- Set of demands – people (with a specified space required and a set of preferences)
- Set of preferences – constraints (which guide the allocation)
- Aim – minimise the space misused and violation of constraints

ABM for Office Space Allocation

- Develop the agents' behaviour so they can self organize and find a solution for OSA
- Components:
  - Rooms – inactive agents (objects)
  - Constraints – inactive agents (objects)
  - Entities (people) - agents
- Agents (people) have an internal goal of finding a better position, which satisfies the constraints associated with itself. To do this, it moves from one room to another, simulating the search process of a traditional heuristic.
- Example – searching for a study room
- Two main decisions to take: 'when to move' and 'where to move'

Agent Based Model



Agent’s Behaviour

- ‘**when to move**’:
  - *Satisfaction-based* (SB) – from the Shelling Segregation Model. If penalty is more than 30% of the total possible penalty, then the entity moves.
  - *Highest-penalized* (HP) – current penalty larger than the average penalty
  - *Always-move* (AM) – all the agents are moving all the time
  - *Random-chance* (RC) – each agent has a chance of 1 in 6 to move to a different location.
- ‘**where to move**’
  - *Random* (Rnd)
  - *Best-potential* (BestPotential) – evaluates the potential of all rooms in the scenario of moving there.
  - *Satisfy-all-constraints* (AllConstr) – find a room which will not violate any associated constraints.
  - *Satisfy-one-by-one* (OneByOne) – picks at random one currently violated constraint and finds a room which will satisfy it in the scenario of moving there.

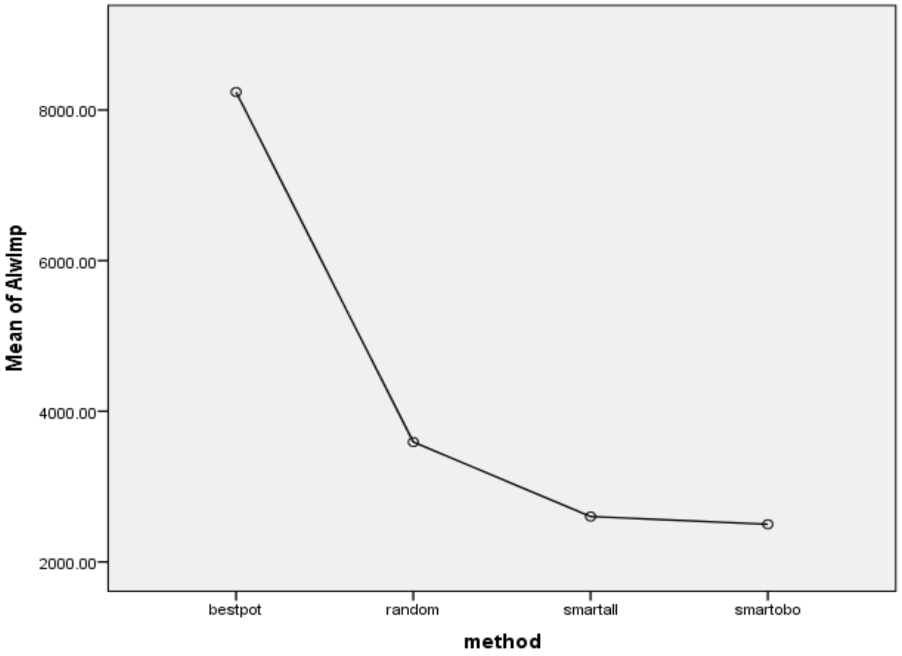
Neighbourhood Heuristics - Relocate

- The neighbourhood heuristics are based on the main idea of the relocation of an entity into a different room. The method of selecting the entity to relocate varies from a completely random one to a greedier one:
  - *RelocateRndRnd* – selects an entity at random and relocates it to another random room
  - *RelocateRndBestRnd* – selects an entity at random and relocates it to the best room, in terms of evaluation function value, out of a randomly selected subset of rooms
  - *RelocateRndBestRoom* – selects an entity at random and relocates it to the best room out of all the available rooms
  - *RelocatePntyBestRnd* – selects the entity with the highest current penalty associated with it and relocates it to the best room out of a randomly selected subset of rooms
  - *RelocatePntyBestRoom* – selects the entity with the highest current penalty and relocates it to the best room out of all the available rooms

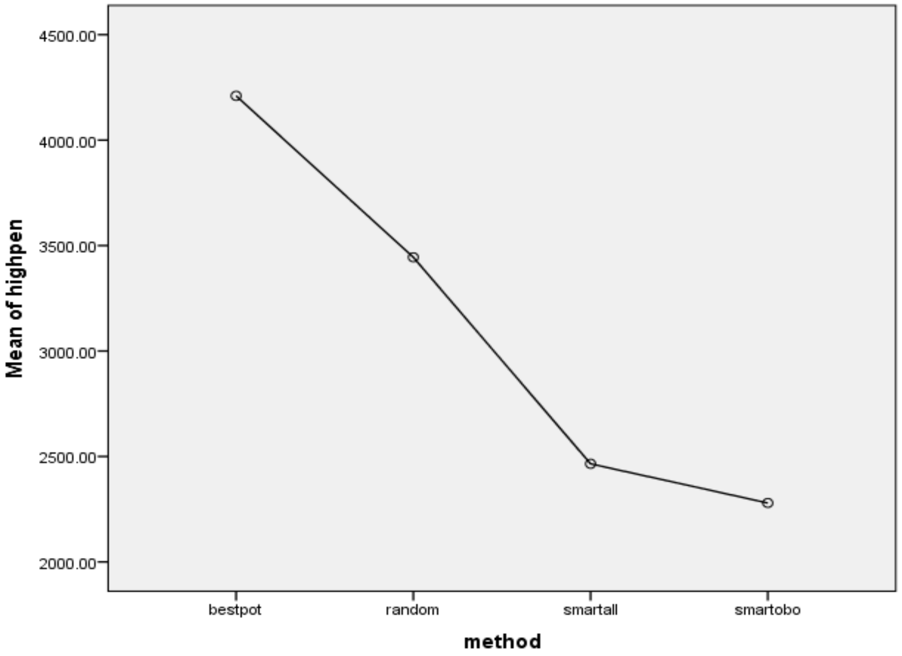
Heuristics vs ABM

Heuristics				
	Diversity	Min	Max	Average
RndRnd	52.34899	1558	2439	1983.27
RndBestRnd	52.67772	1446	2361	1930.56
RndBestAll	50.89714	1364.5	2383	1861.27
PntyBestRnd	51.77373	1785.5	2818	2348.92
PntyBestAll	52.60937	1493	2800.5	2304.8
ABM Simulation				
	Diversity	Min	Max	Average
SB- BestPotential	0	3971.5	3971.5	3971.5
SB-Rnd	50.08903	2878	3553.5	3181.68
SB-AllConstr	30.96836	1969	2808	2432.36
SB-OneByOne	33.36529	1691.5	2710	2247.53
HP- BestPotential	0	4210	4210	4210
HP-Rnd	52.25312	2942.5	3871.5	3444.55
HP-AllConst	27.33872	1889.5	2991	2465.24
HP-OneByOne	31.2423	1751	2761.5	2279.42
AM- BestPotential	0	8238.5	8238.5	8238.5
AM-Rnd	70.78482	2890	4076	3591.48
AM-AllConstr	34.69388	1853.5	3150	2602.69
AM-OneByOne	32.94069	1835	3083.5	2499.79
RC- BestPotential	41.54225	3053	4265.5	3502.37
RC-Rnd	66.31968	2347.5	2952.5	2599.97
RC-AllConstr	21.29845	1536	2318	1899.82
RC-OneByOne	21.40803	1346	2069	1751.5

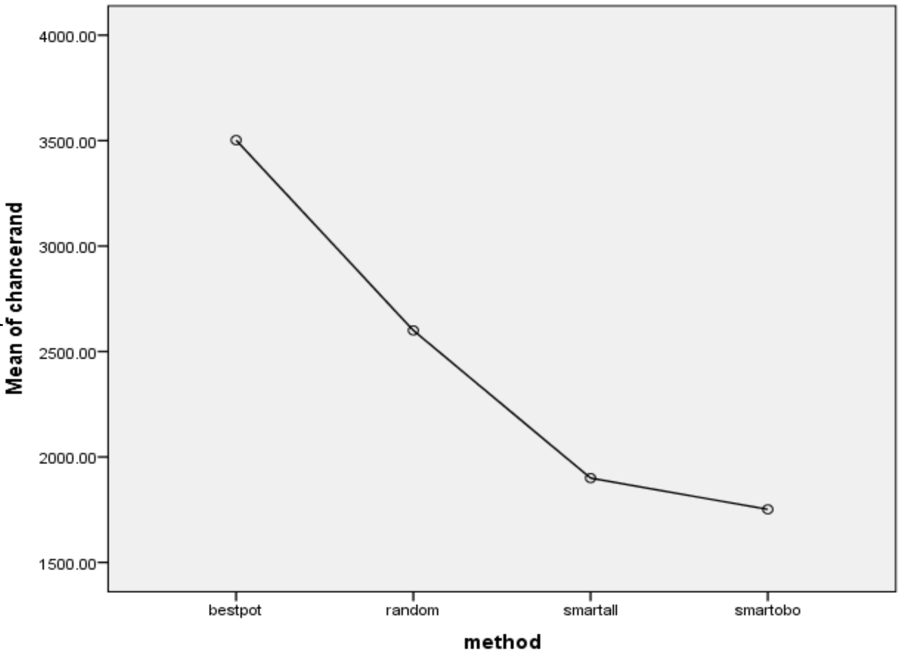
Always Move



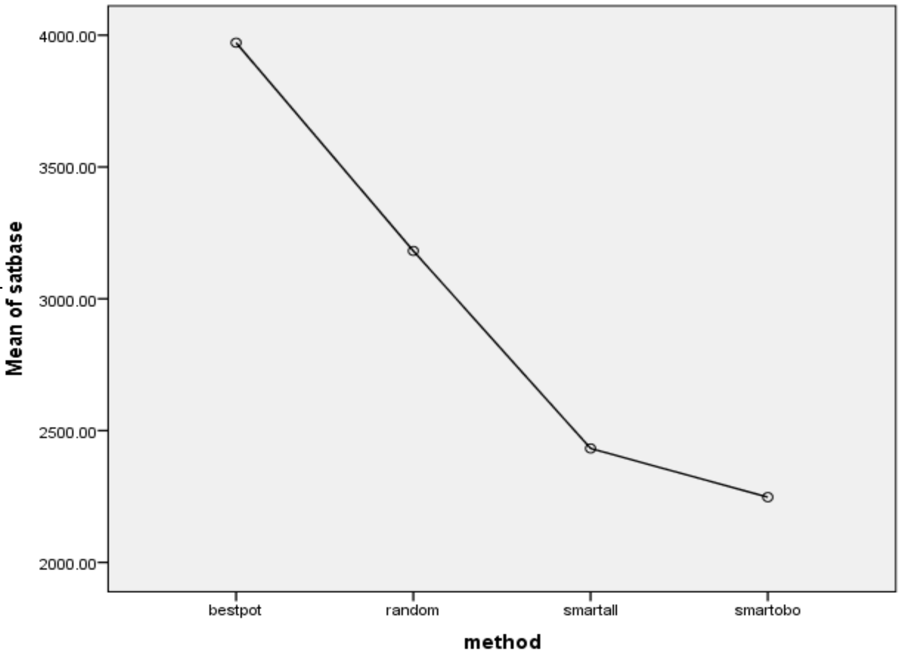
Highest Penalised



Chance Random

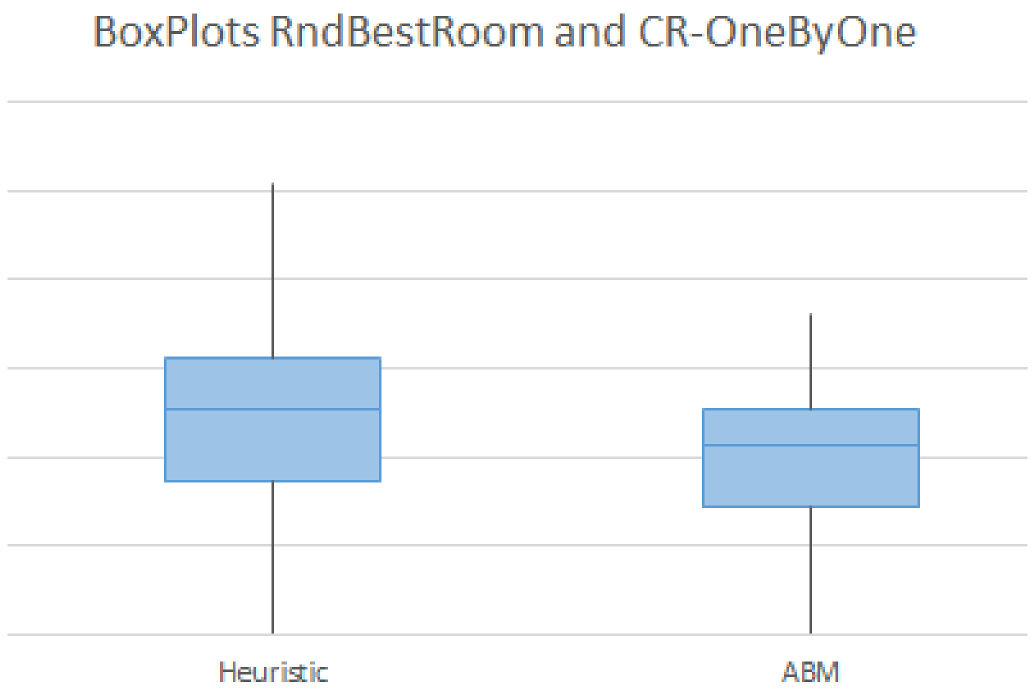


Satisfaction Based



CR-OneByOne vs RndBestAll

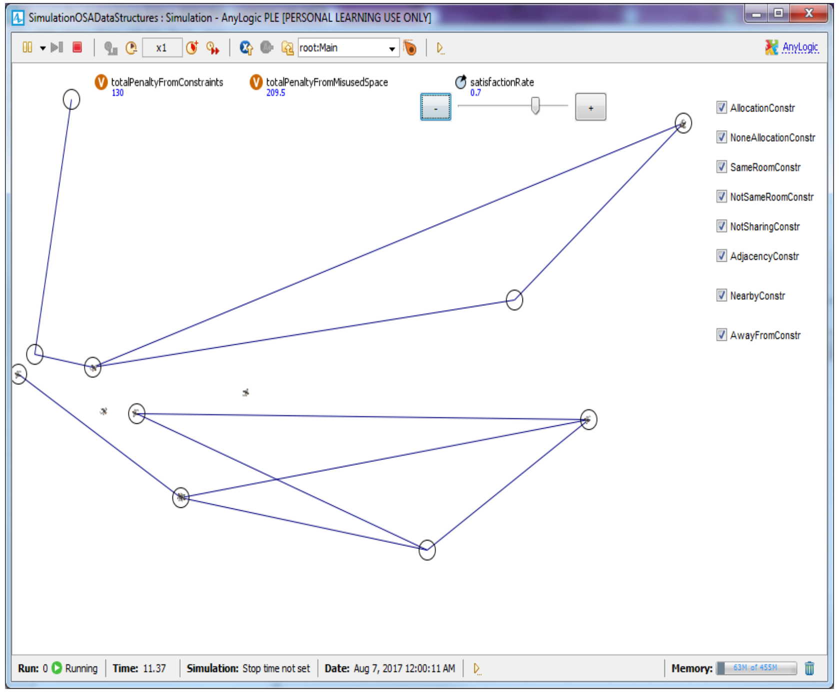
- CR-OneByOne – better in terms of performance
- RndBestAll – better in terms of diversity



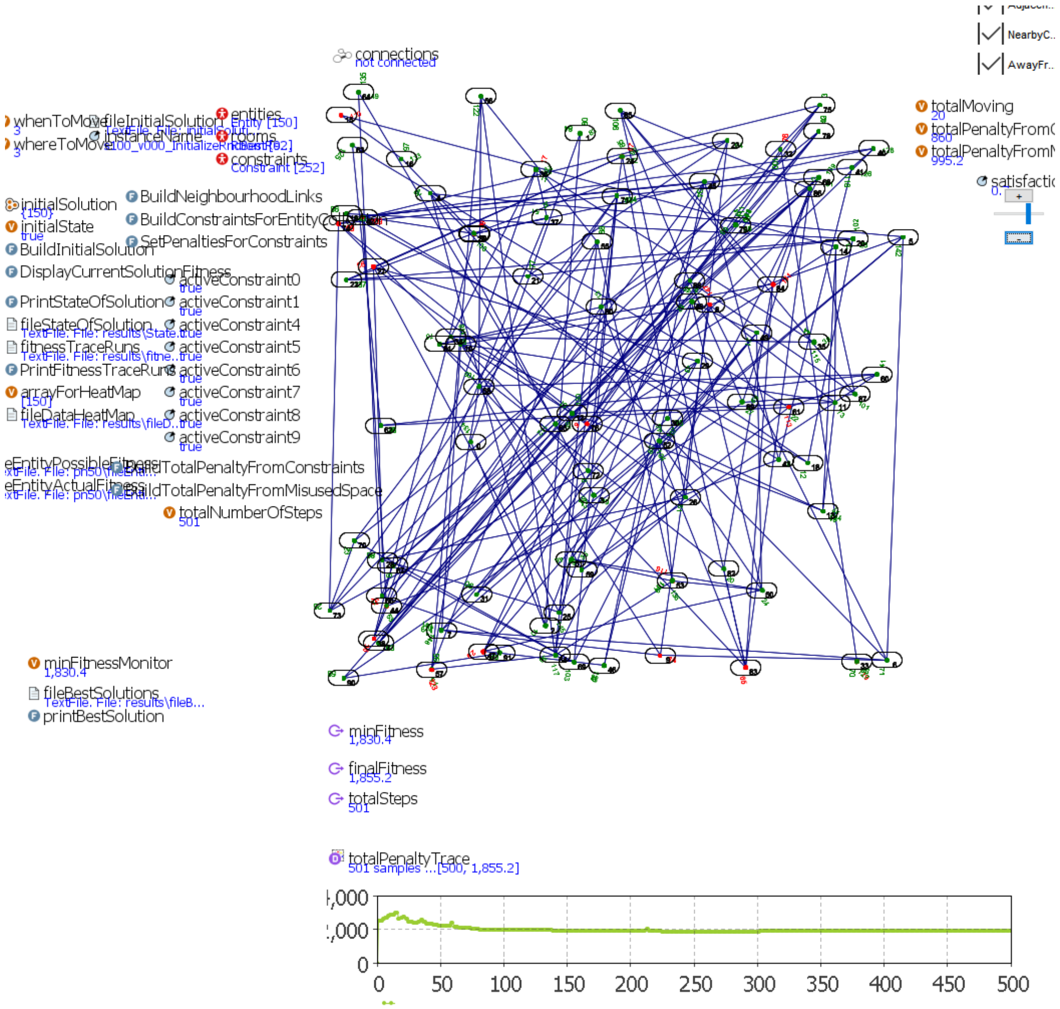
Remarks

- Synchronous nature of the ABM influences drastically the performance of the agents
- Issues which affect the current ABM:
  - Effects of the misleading calculations of possible penalty and actual penalty (before and after the move)
  - Changes of the environment – the position of the other agents
  - Constraints are mostly related to agents and their proximity connections with other agents
- Solution – communication or asynchronous movement
- Heuristics have higher diversity degree because the agents can opt to stay during the ‘where to move’ process

Then…


SimulationOSADataStructures : Simulation - AnyLogic PLE [PERSONAL LEARNING USE ONLY]
root:Main
AnyLogic
totalPenaltyFromConstraints
130
totalPenaltyFromMisusedSpace
209.5
satisfactionRate
0.7
AllocationConstr
NoneAllocationConstr
SameRoomConstr
NotSameRoomConstr
NotSharingConstr
AdjacencyConstr
NearbyConstr
AwayFromConstr
Run: 0 Running
Time: 11.37
Simulation: Stop time not set
Date: Aug 7, 2017 12:00:11 AM
Memory:

Now…


connections
not connected
NearbyC...
AwayFr...
totalMoving
20
entities
rooms
constraints
initialSolution
initialState
true
BuildInitialSolution
BuildNeighbourhoodLinks
SetPenaltiesForConstraints
DisplayCurrentSolutionFitness
PrintStateOfSolution
fileStateOfSolution
fitnessTraceRuns
PrintFitnessTraceRuns
arrayForHeatMap
fileDataHeatMap
activeConstraint0
activeConstraint1
activeConstraint4
activeConstraint5
activeConstraint6
activeConstraint7
activeConstraint8
activeConstraint9
totalNumberOfSteps
501
minFitnessMonitor
1,830.4
fileBestSolutions
printBestSolution
minFitness
1,830.4
finalFitness
1,855.2
totalSteps
501
totalPenaltyTrace
501 samples ...[500, 1,855.2]
0
0
50
100
150
200
250
300
350
400
450
500

Thank you!

Questions?