Signal Processing LABORATORY

http://splab.cz/en

Optimization of Warehouse Processes

Jan Karásek
Brno University of Technology
Faculty of Electrical Engineering and Communication
Department of Telecommunications, Signal Processing Laboratory

esf european social fund in the czech republic
EUROPEAN UNION
MINISTRY OF EDUCATION, YOUTH AND SPORTS
OP Education for Competitiveness
BRNO UNIVERSITY OF TECHNOLOGY
INVESTMENTS IN EDUCATION DEVELOPMENT

Outline

- Introduction
  - Problem Definition, Main Goals, Warehouse Description
- Problem Solutions
  - Generation I, II, III
- Related Problems, New Ideas
- Java Evolutionary Framework
- Grammar Driven Genetic Programming
- GDGP Algorithm Design
- Benchmark Definition
- Experimental Results
- Conclusion + Further Research

Introduction

Problem Definition

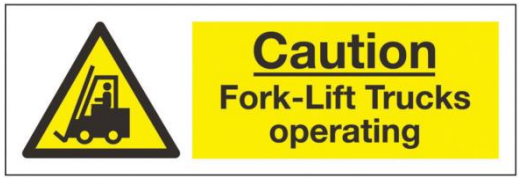
Definition of Research Problem

How to spread the tasks among employees and assign the equipment, so that the system can work as a cooperating whole with minimal time demands and with collision avoidance...

Main goals are

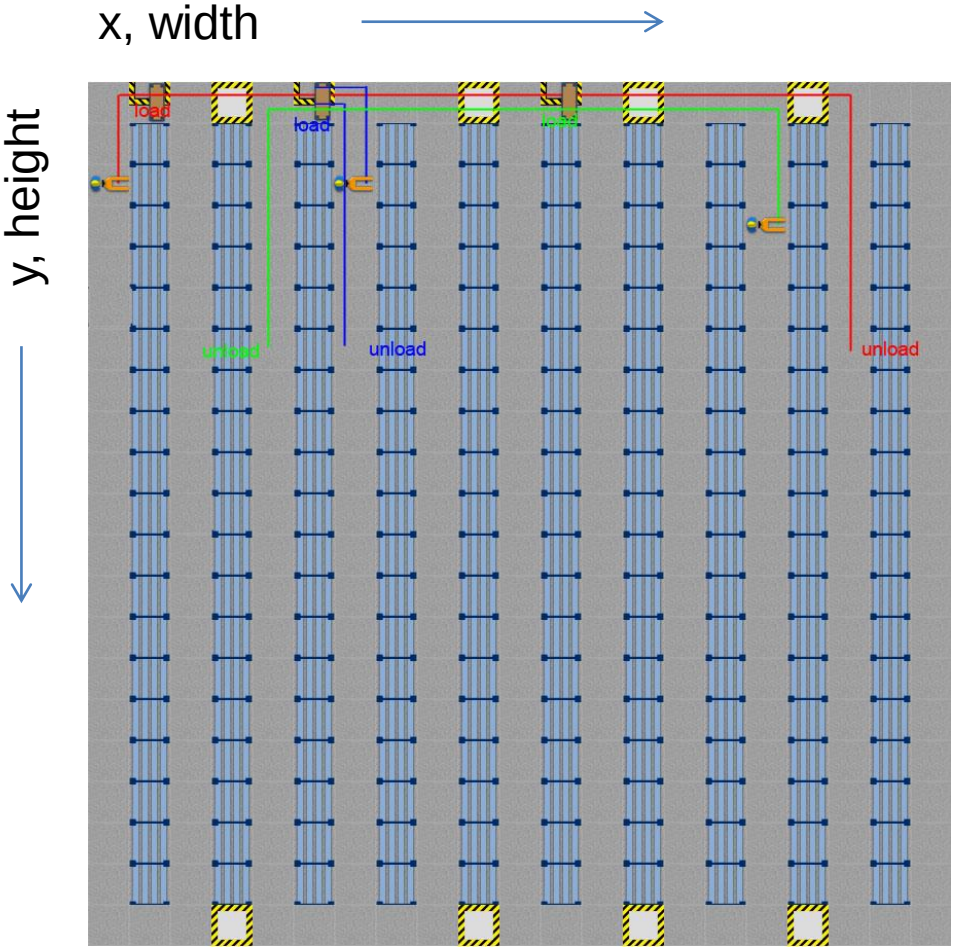
to minimize a time of job processing,

and to avoid the collisions of fork-lift trucks.



Introduction

Warehouse Description



Typical rectangular shape of warehouse.
Coordinates: x, y, z (level of shelf).

Goals, more concretely:

a) to minimize time of:
searching (location of goods)
travelling (how to get there)
picking (both together)

b) to avoid collision by minimization of:
idle times (truck waiting for another)
crash of trucks (were not waiting)
not available path (congestion)

Problem Solutions

The processes in the warehouse are driven by:

- Human Operators/Shift leaders,
- Warehouse Management Systems
- **Methods based on Artificial Intelligence**

Problem Solutions – Generation I

How the problem is being solved usually

?

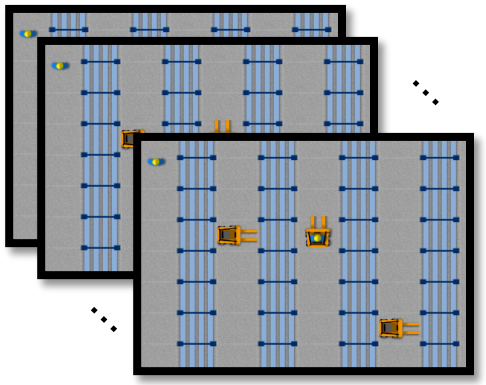
Problem Solutions – Generation II

The TOP solution (Mibcon, Certified by SAP)

?

Problem Solutions – Generation III

Solution based on GDGP we are working on



Related Problems and New Ideas

Warehouse Optimization

- Technical Structure (layout, dimensions, racks, aisles…)
- Operational Structure (random/dedicated storing, grouping, zoning…)
- Management Systems (interconnection between various systems)

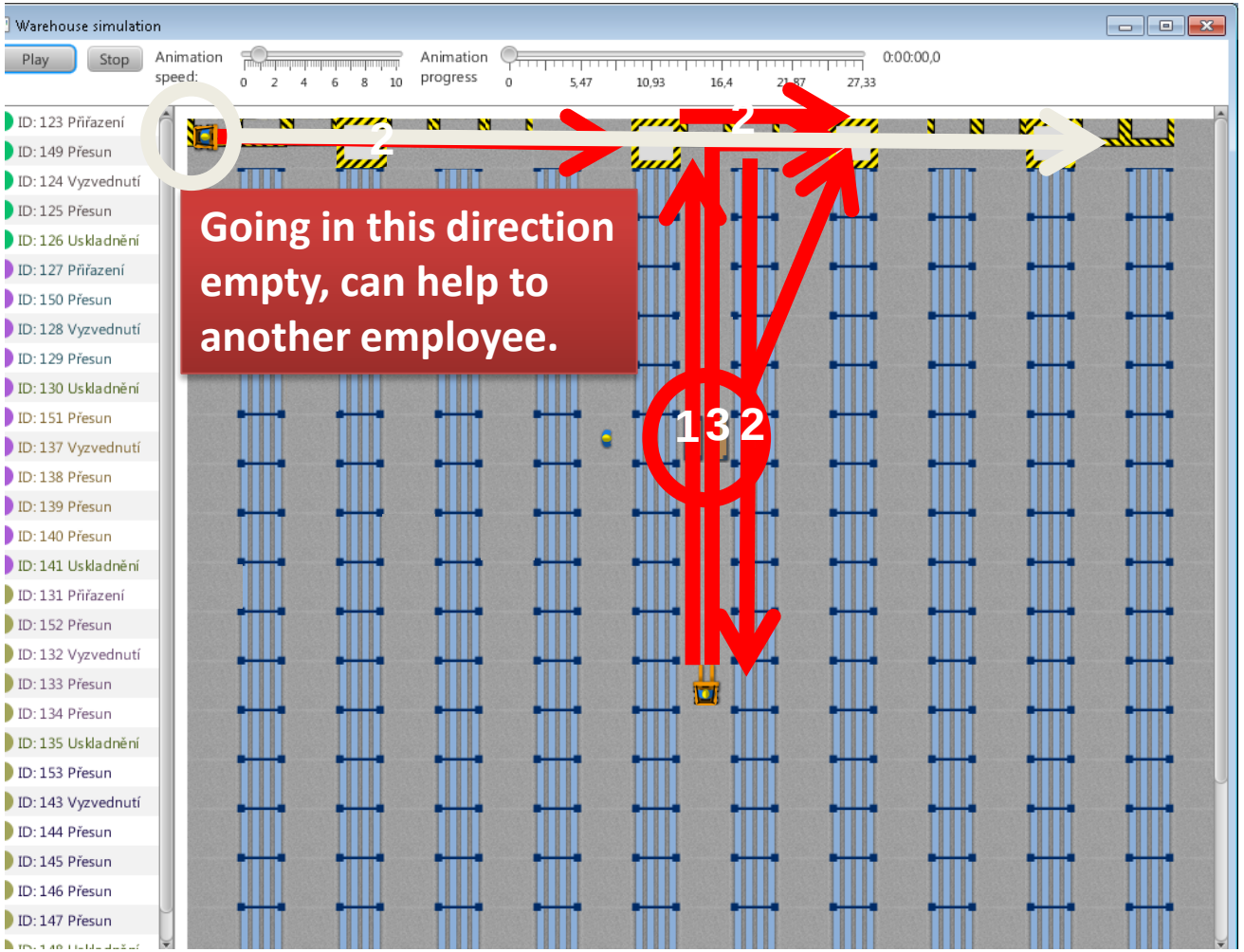
Scheduling Problems (Shop Scheduling)

+

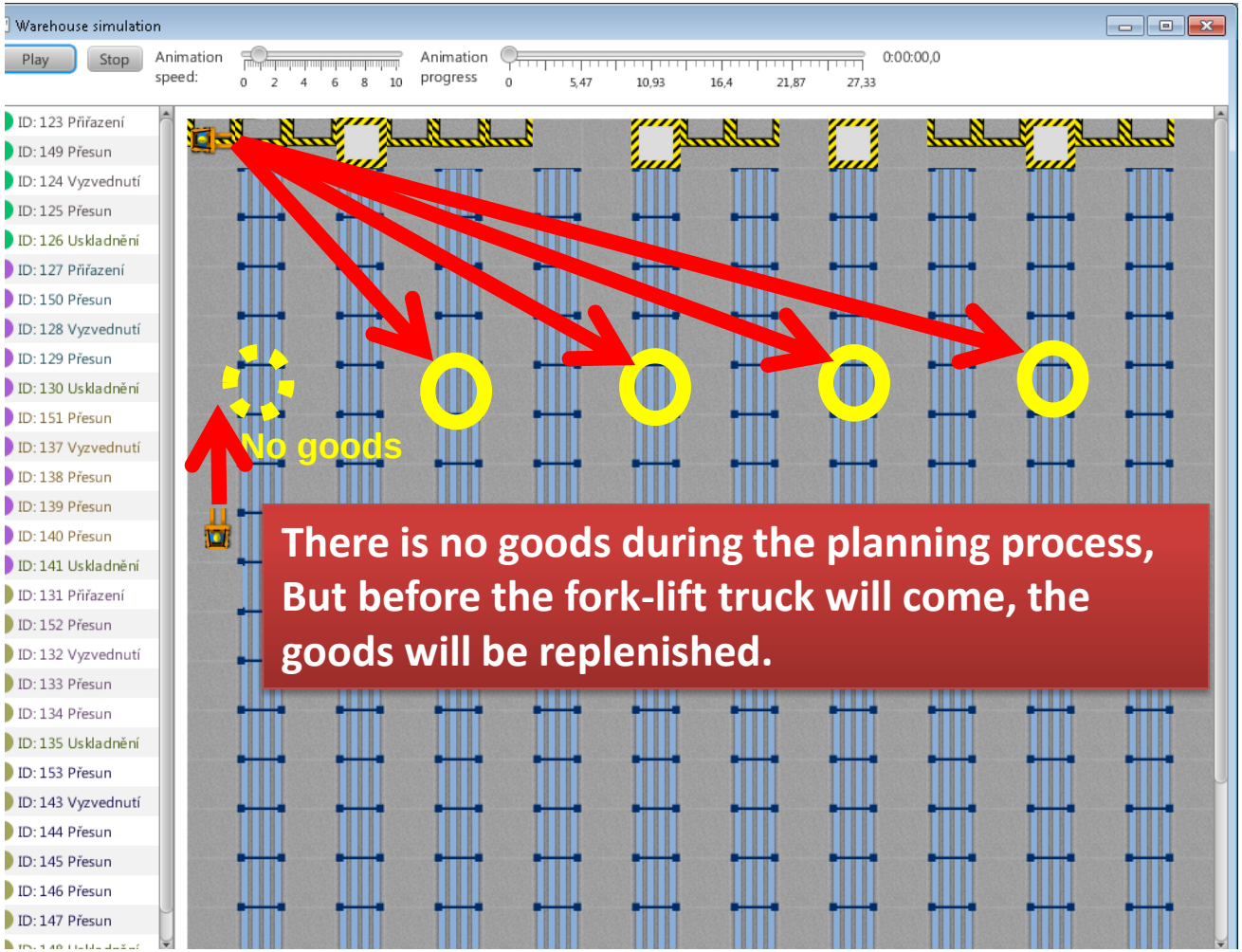
Routing Problems (Vehicle Routing)

} Robotic Cells, Hoists, AGV's

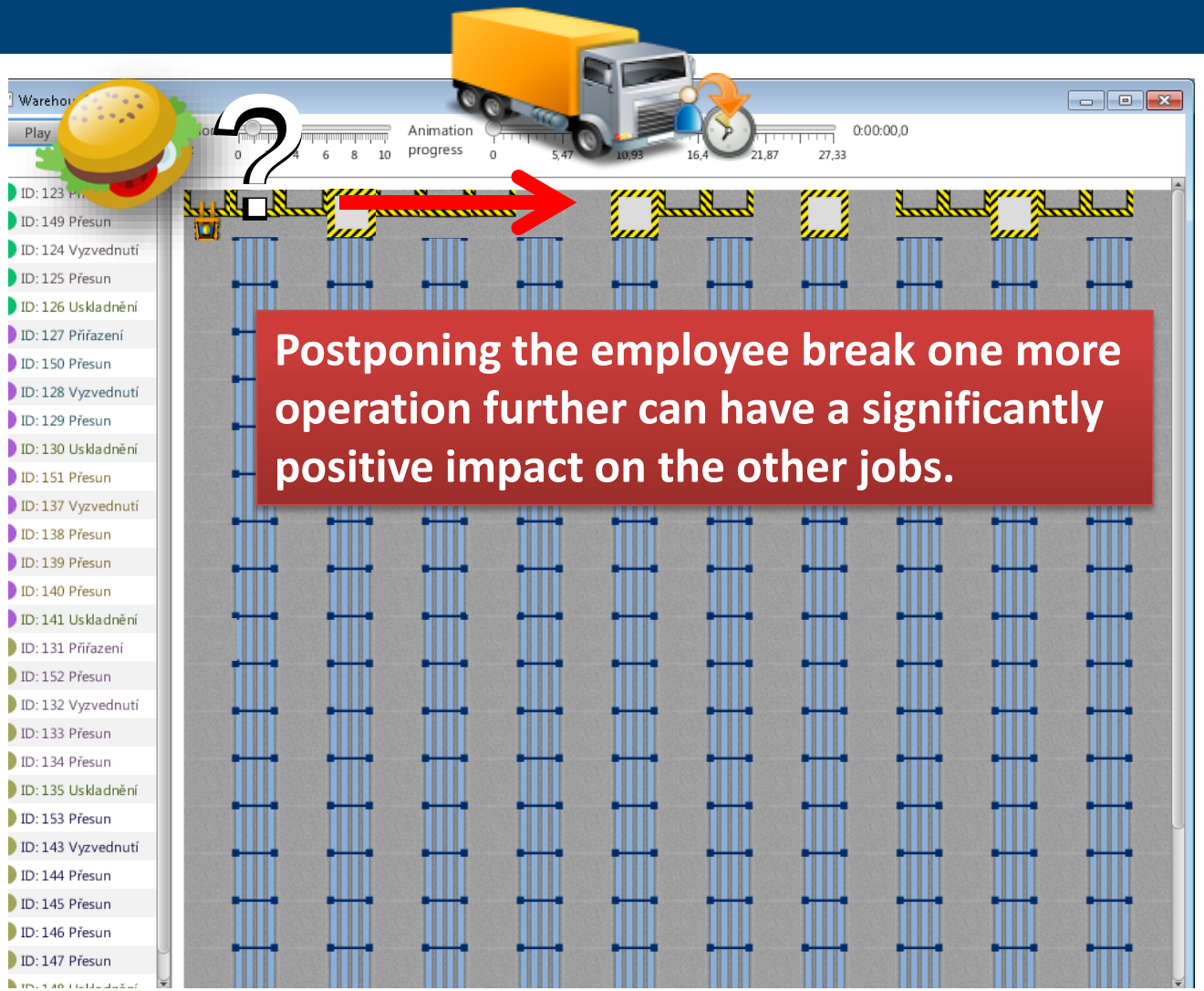
Related Problems and New Ideas



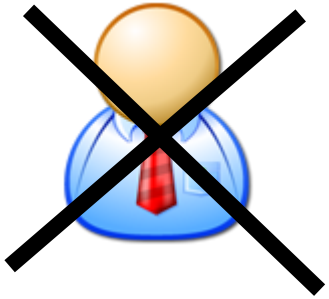
Related Problems and New Ideas



Related Problems and New Ideas

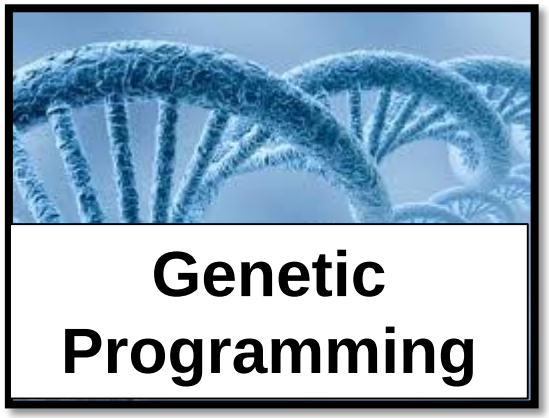


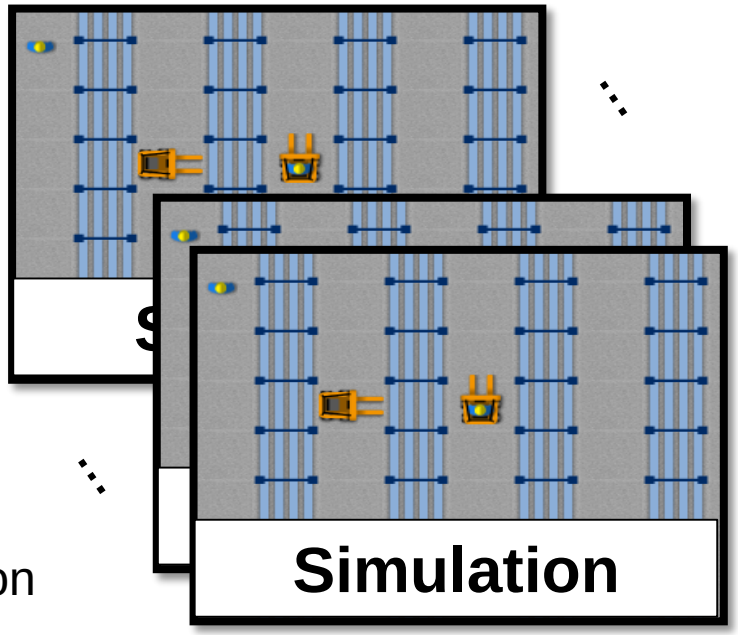
Related Problems and New Ideas



- Solution is less demanding on human resources
- Higher utilization of employees and trucks
- Saves labor energy
- Solution does not require highly skilled operational managers.

Java Evolutionary Framework

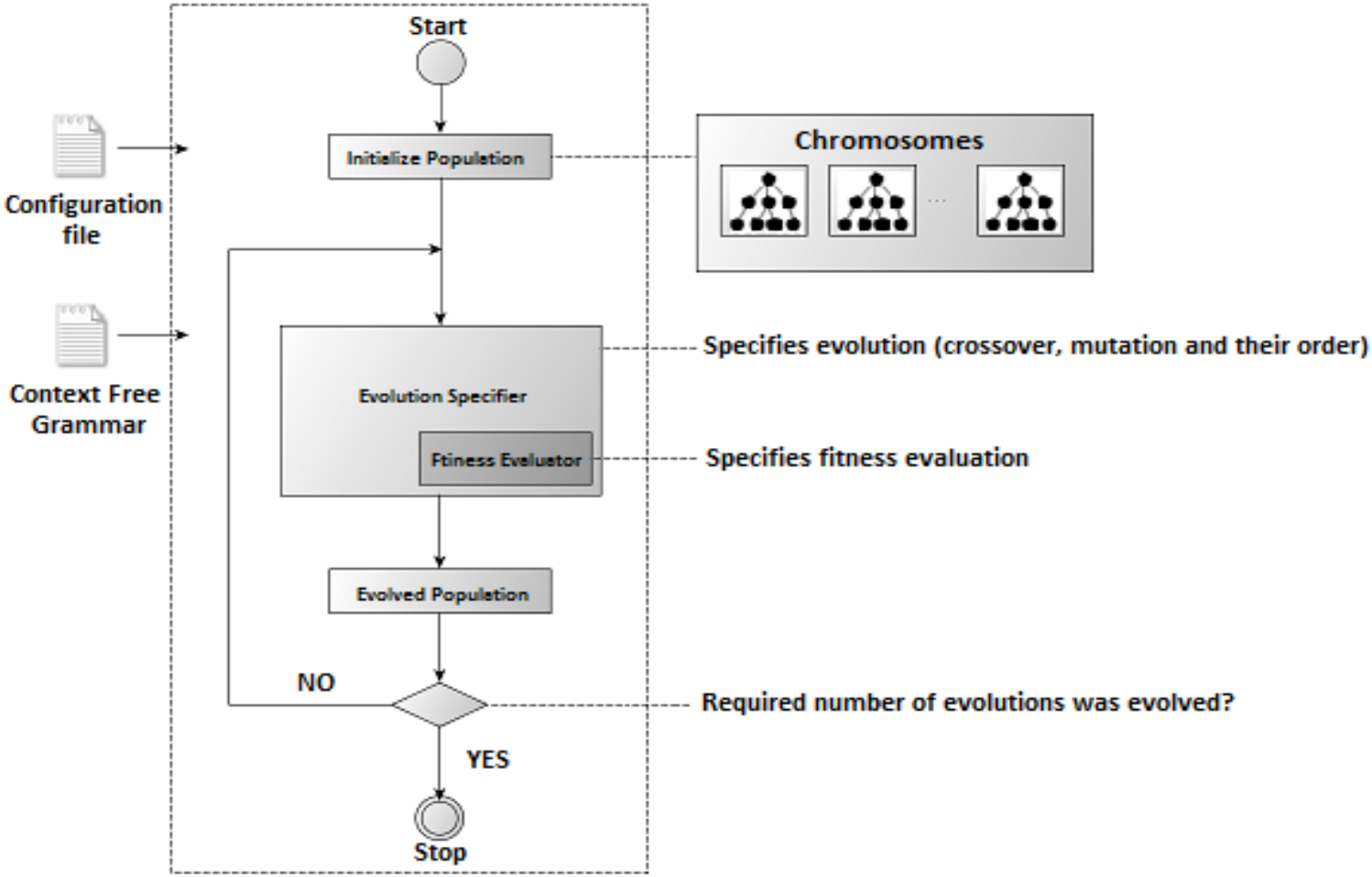




- New Java Evolutionary Framework based on Genetic Programming
- Driven by Context-free Grammar
- Thousands of possible variants of problem
- The more complicated problem, the better and successful solution in comparison with human

GDGP can “see” to the future, and can predict the most probable situations and properly react to them.

Java Evolutionary Framework



Grammar Driven Genetic Programming

- **Set of Terminal Symbols** (inputs)
  - Coordinates[x,y] (start, end), Employee, Equipment, Commodity, Deadline…
- **Set of Non-Terminal Symbols** (functions = tasks)
  - Operation, Job, various Task (TaskInStore, TaskOutStore…)
  - Load, Unload, Move, Relax, Wait…
- **Fitness Function**
  - Minimization of Completion time, Number of Collisions…
- **Run Control**
  - Size of Population, Grammar, Operator Rates…
- **End Control**
  - Number of generations

Grammar Driven Genetic Programming

Definition of Grammar (G = V, ∑, R, S)

- **V (Set of Non-Terminal characters)**
- **∑ (Set of Terminal characters)**
- **R (Set of Rules)**
- **S (Start)**

S ::= Workplan

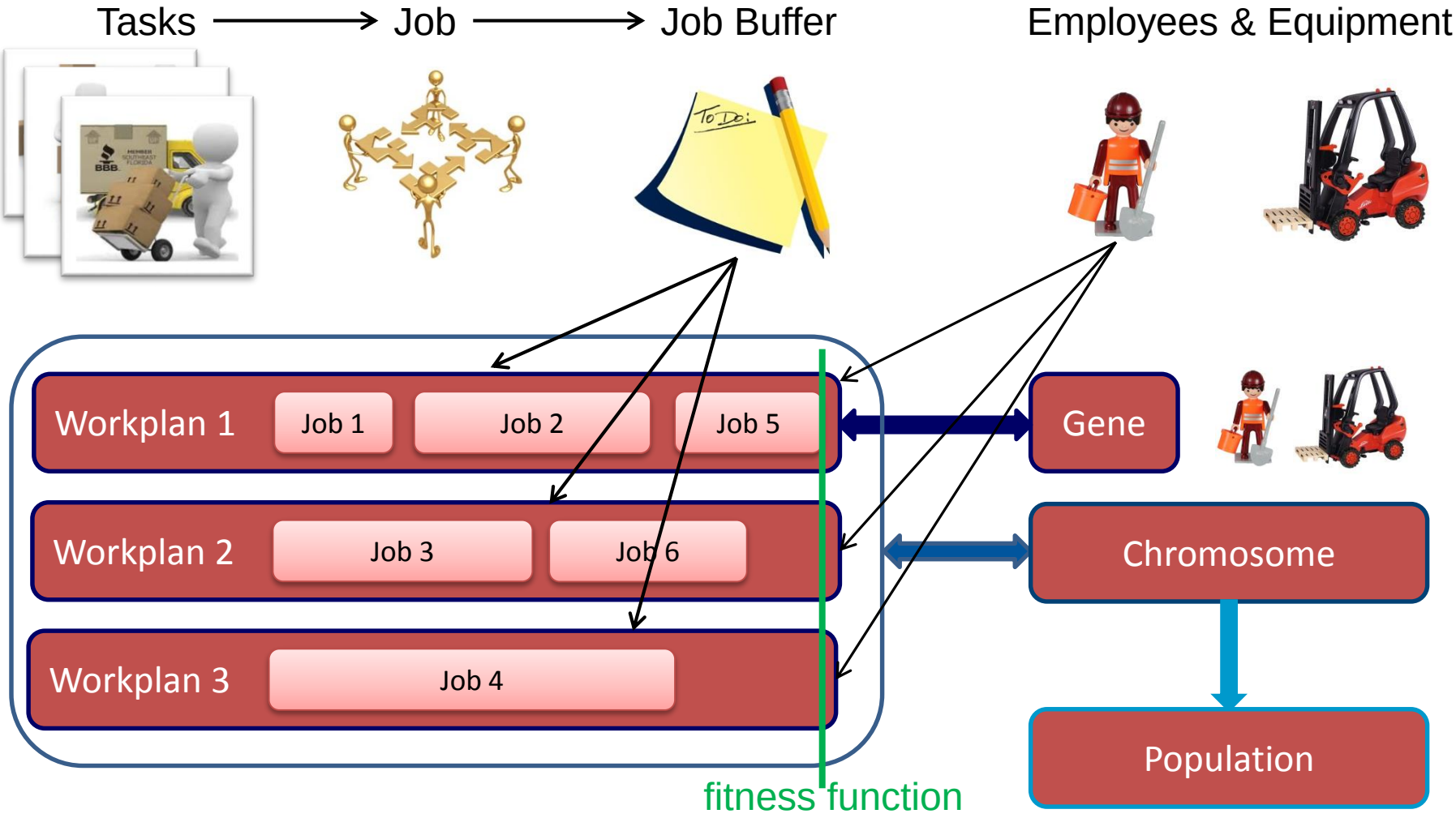
R :: =

```
Workplan      ::= Operation Employee Equipment
Operation     ::= Operation Job | Operation null
Job           ::= TaskInStore | TaskOutStore |
                  TaskRelax | TaskWait….
TaskInStore   ::= Load Move Unload Commodity, Start, End…
TaskOutStore ::= Load Move Unload Commodity, …. Deadline
TaskRelax     ::= Wait Position Duration | Relax …
```

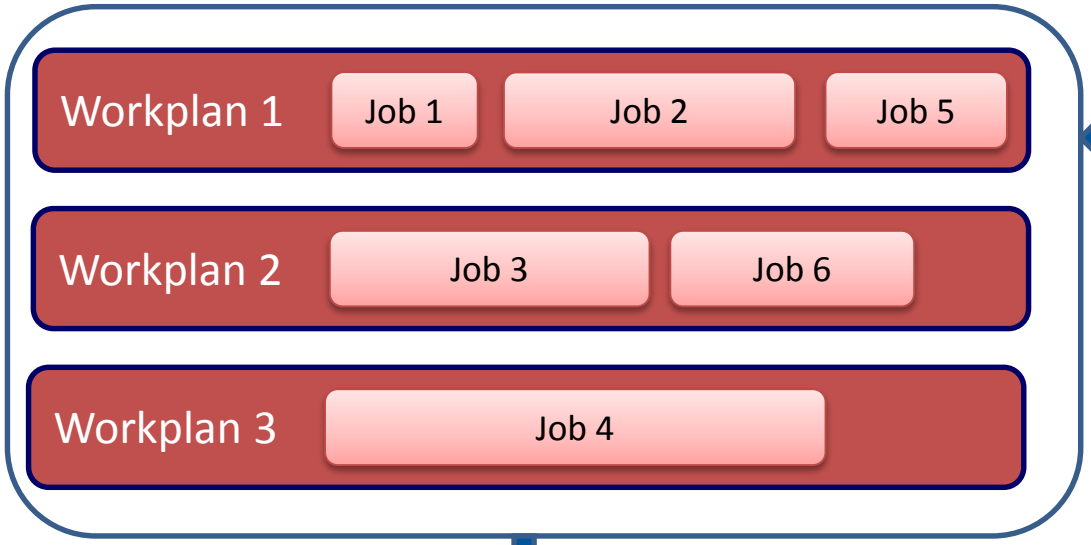
Advantages of Innovative Method

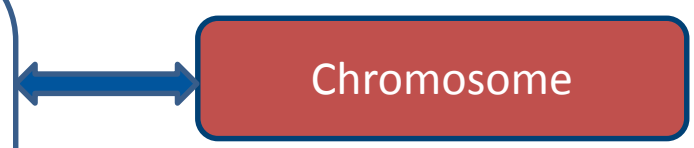
- Simulation of Future and Expected Events/Operations
- It Plans Jobs for Employees, Utilization and Energy
- It looks at the Warehouse as one Cooperating Unit
- Less Requirements on Skilled Human Resources
- An Efficient Break Scheduling
- Increases Resistance:
  - Failure of Operational Manager
  - Effect of Emotions and Stress
  - Panic in Crisis Situations

Algorithm Design



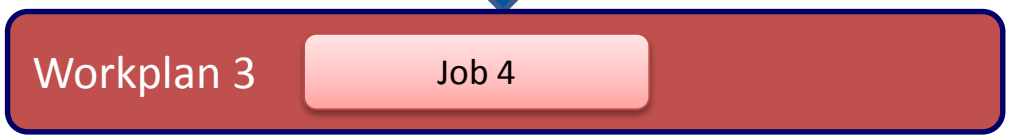
Genetic Operators – Path Mutation



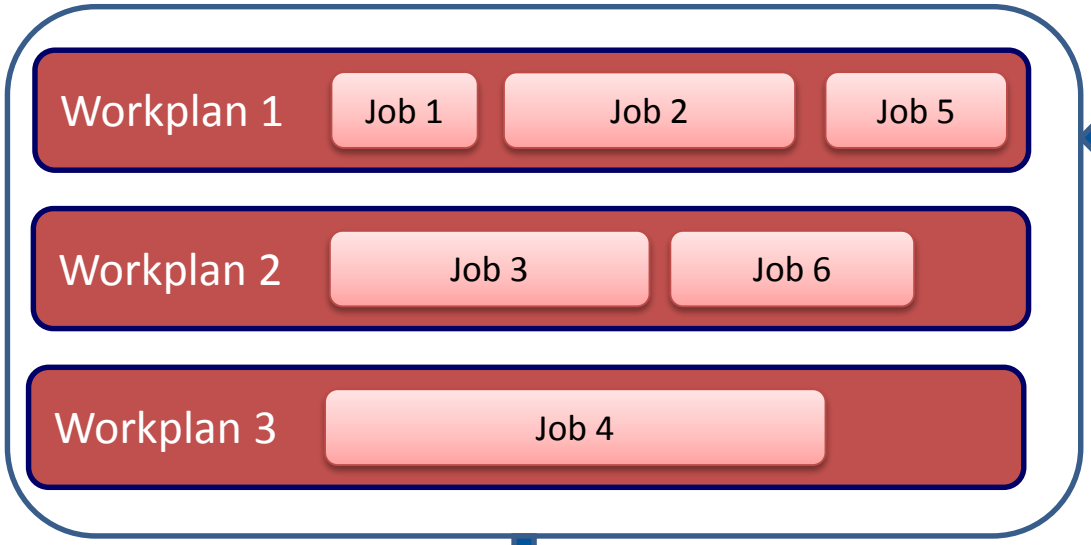


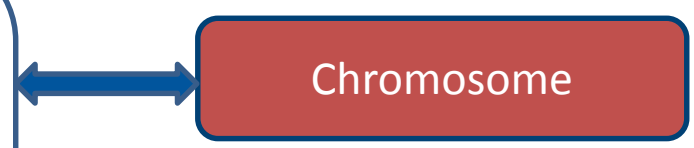
Path mutation with rate R_{PM}

1) Randomly select the Workplan (no. 3)
2) Randomly select the Job (no. 4)
3) Apply Path mutation on Move Task



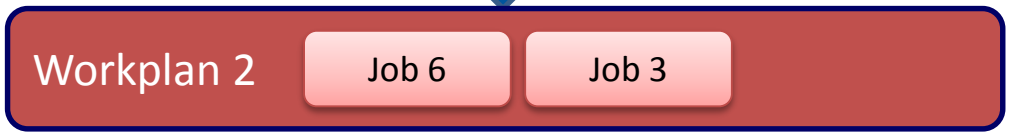
Genetic Operators – Job Order Mutation



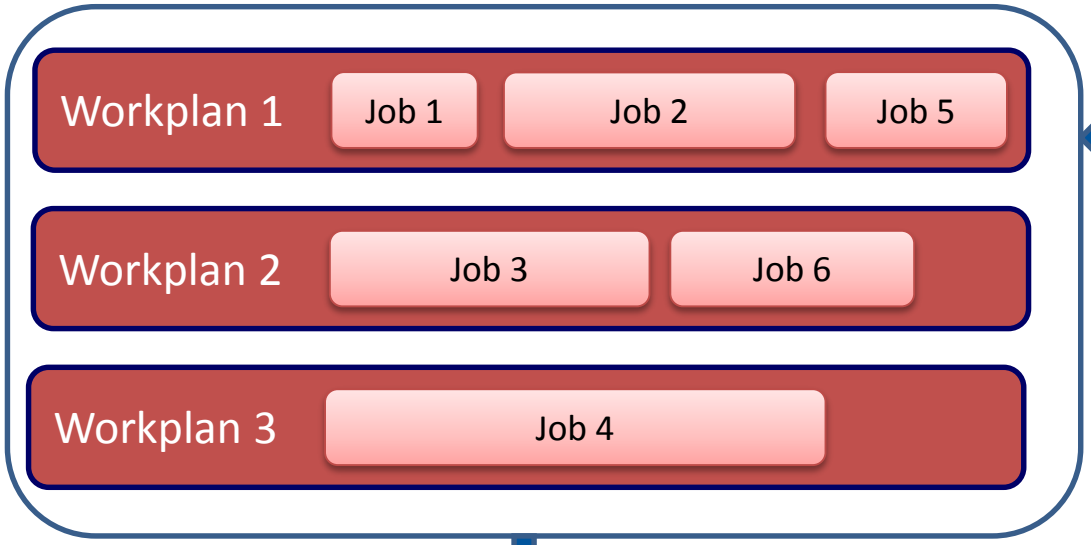


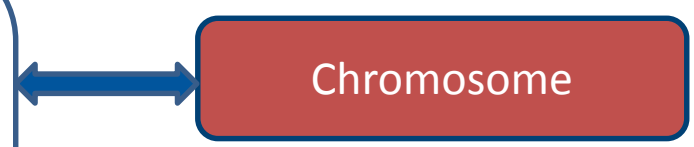
Job Order mutation with rate R_{TO}

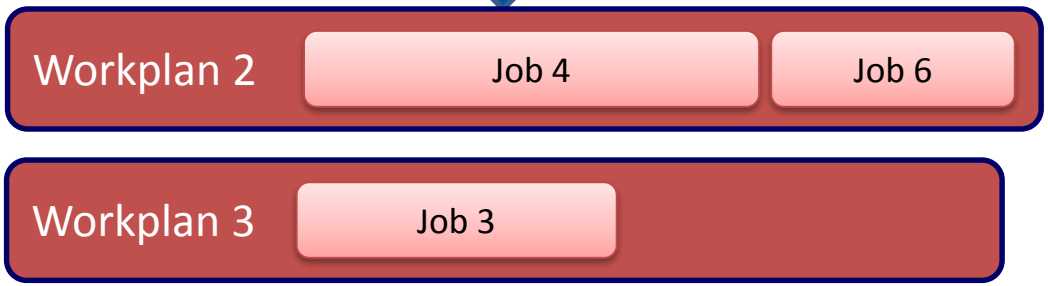
1) Randomly select the Workplan (no. 2)
2) Apply Job Order mutation



Genetic Operators – Swap Job Mutation



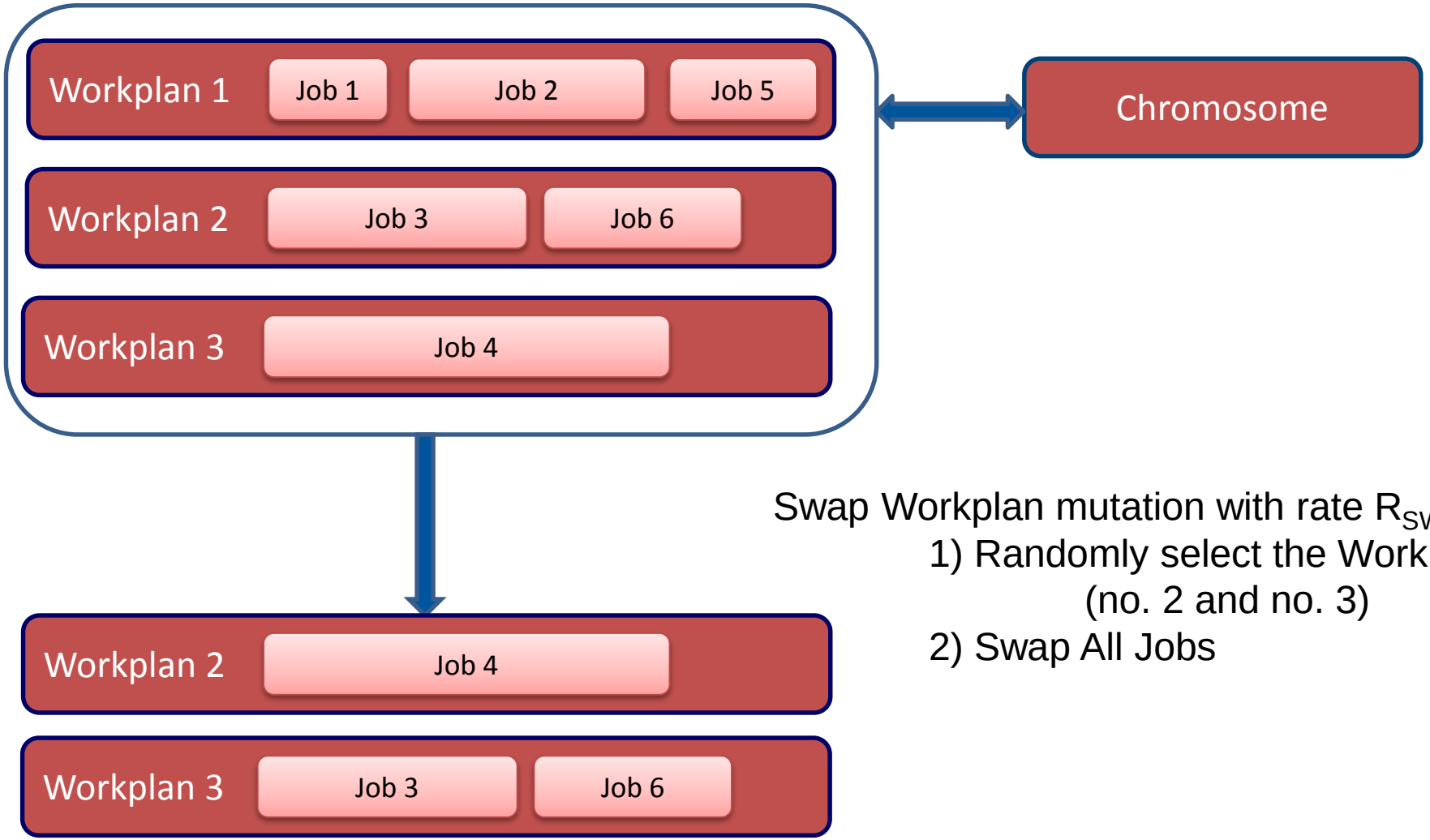




Swap Job mutation with rate R_{ST}

1) Randomly select the Workplans (no. 2 and no. 3)
2) Randomly Select Two Jobs
3) Swap Selected Jobs

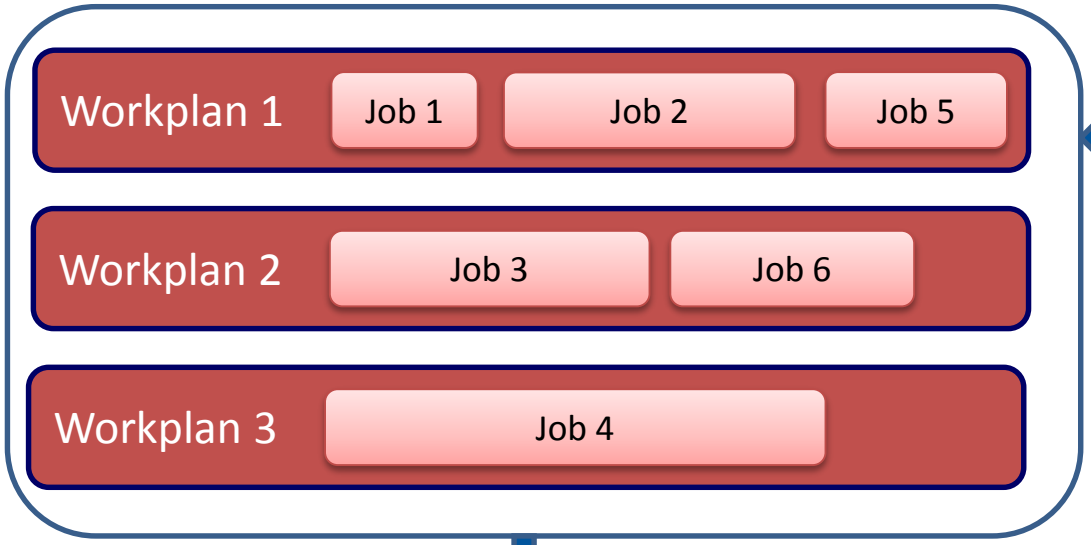
Genetic Operators – Swap Workplan Mutation

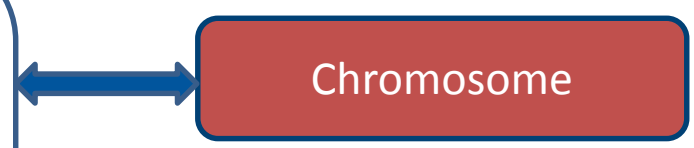


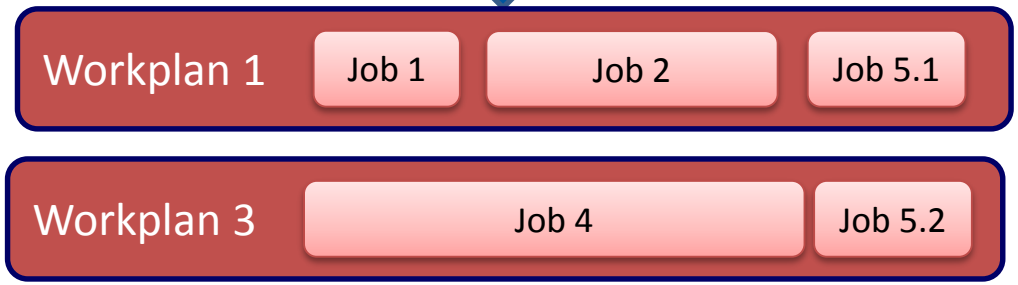
Swap Workplan mutation with rate R_{SW}

1) Randomly select the Workplans (no. 2 and no. 3)
2) Swap All Jobs

Genetic Operators – Split Job Mutation







Split Job mutation with rate R_{SA}

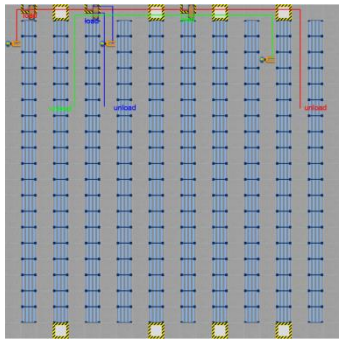
1) Randomly select the Workplan (no. 1)
2) Randomly select Job (Job 5)
3) Apply Task Order mutation

Benchmarks Definition

- **State the benchmark**
  - **Manually measure the time and performance**
  - Based on human decision
  - **Automatically measure the time and performance**
  - Use different genetic operators and combinations
  - Based on evolution

Benchmarks Definition

- **Data from warehousing company** from pre Christmas time, when human operator was stressed, tired . . . **extracted to simple scenarios => benchmark sets**



Visual Validation

- **How the operator spread the work**, e.g.:
  - Emp1, Truck1, WP1
    - T1, T2
  - Emp2, Truck2, WP2
    - T3, T4, T5

empID, truckID, jobEndTime, commID, destX, destY, job, task

- **2 Sets of Simple Benchmarks (2 x 10 Scenarios)**

Experimental Results

Benchmark Set #1

Human Operator	Genetic Programming		
	Path Mutation	Task Order M.	Both Mutation
13,00	15,50	13,00	13,00
16,50	16,50	16,50	16,50
13,00	28,50	28,50	28,50
16,50	16,50	18,50	16,50
12,50	12,50	12,50	12,50
14,50	26,50	26,50	26,50
15,00	15,00	15,00	15,00
9,00	8,00	8,00	8,00
13,00	13,00	12,50	14,00
16,50	16,00	16,00	16,00

values are measured in standardized time units [t.u.]

Benchmark Set #2

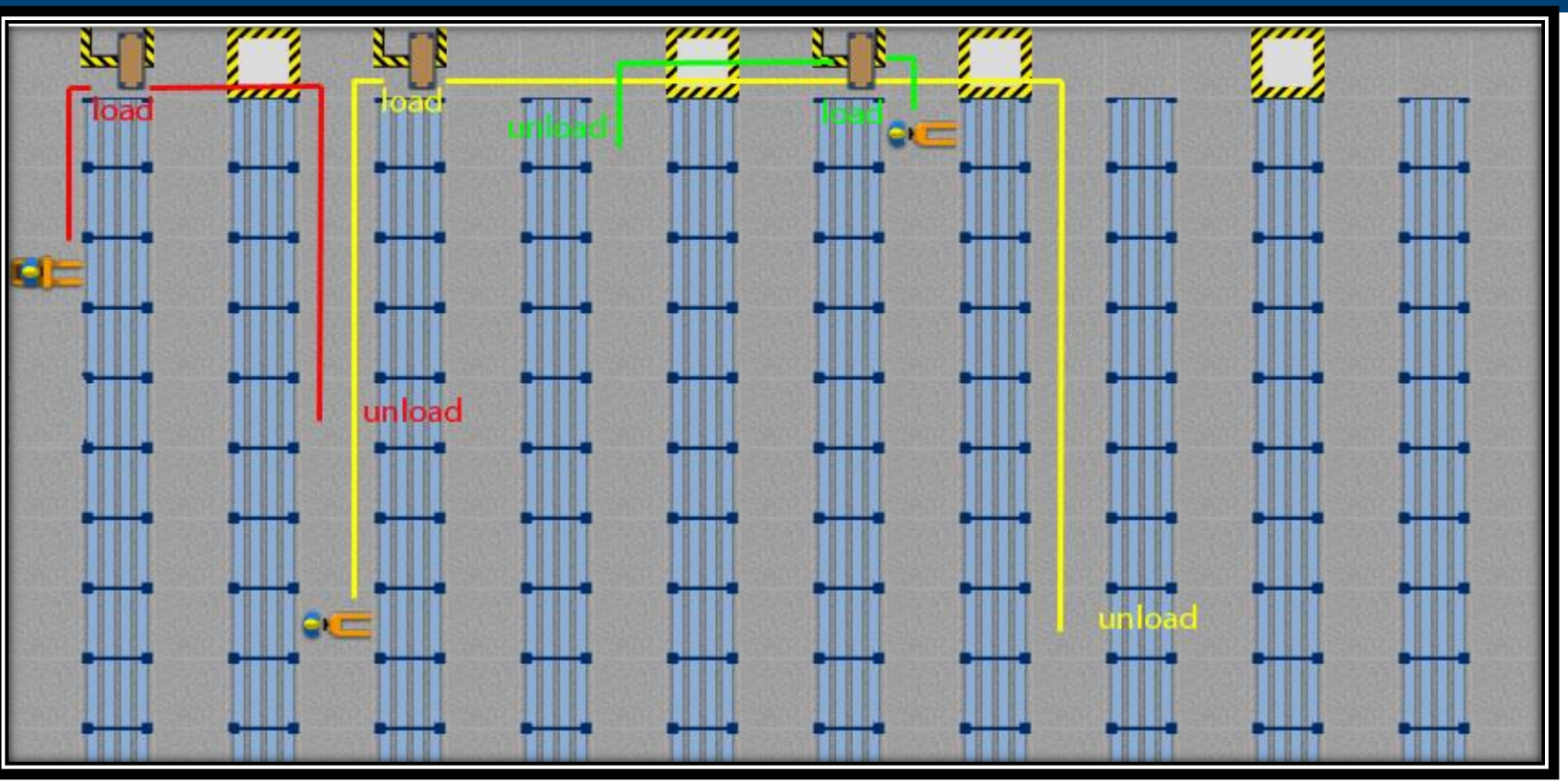
Human Operator	Genetic Programming		
	Path Mutation	Task Order M.	Both Mutation
8,00	11,50	8,00	8,00
14,00	14,50	14,50	14,50
13,00	15,50	13,88	14,63
14,00	12,50	12,25	12,25
11,00	11,00	11,00	11,00
14,50	16,50	15,00	15,00
15,00	11,50	11,50	11,50
8,50	8,00	8,00	8,00
13,00	12,50	12,00	12,00
16,50	12,13	13,00	12,13

2-4 employees with hand pallet truck
Speed of hand pallet truck = 2 t.u.
Collision detection is not implemented
Each employee has own tasks

Size of population: 10
# of generations: 10
Path mutation rate: 30%
Task order mutation rate: 30%
Tournament selection

2-4 emps with hand/low fork-lift
Speed of fork-lift low truck = 8 t.u.
Collision detection is not implemented
Each employee has own tasks

Experimental Results

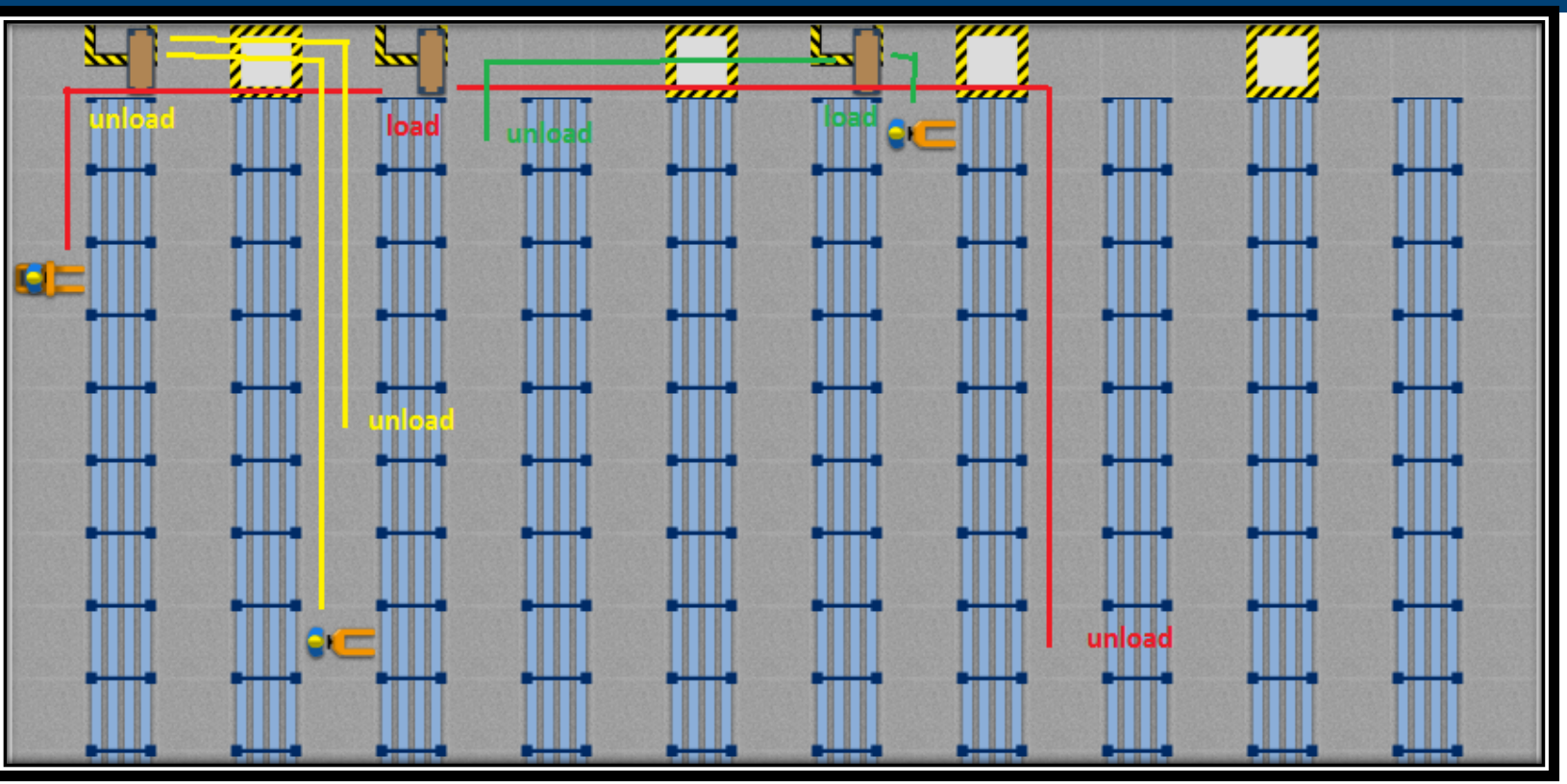


Time:

R: 5.38
Y: 16.50
G: 6.50

Red = [0,3] > [0,0] > [1,0] > [4,0] > [4,5] > [5,5]
Yellow = [4,8] > [4,0] > [5,0] > [14,0] > [14,8] > [15,8]
Green = [12,1] > [12,0] > [11,0] > [8,0] > [8,1] > [7,1]

Experimental Results



Time:

R: 7.00
Y: 13.00
G: 7.50

Red = [0,3] > [0,0] > [4,0] > [5,0] > [14,0] > [14,8] > [15,8]
Yellow = [4,8] > [4,0] > [2,0] > [1,0] > [4,0] > [4,5] > [5,5]
Green = [12,1] > [12,0] > [11,0] > [6,0] -> [6,1] -> [7,1]

Conclusion

- The system can automatically model and evaluate thousands of possible variants of solution
- Competitive results has been reached even if the scenarios were so simple for human to design solution
- The more complex problem the better solution in comparison with human based solutions
- Employee gets into his PDA the information about what to do and how exactly get there
- The system re-computes results each 5 minutes for current tasks, so the employees information are updated

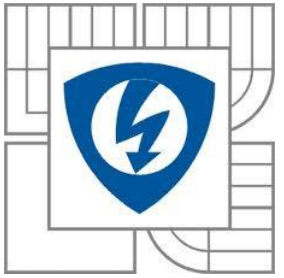
Acknowledgement

Research and development of the system for manufacturing optimization

Project no: FR-TI1/444

Signal Processing LABORATORY

http://splab.cz/en



Thank you for your attention

... time for questions ...

Czech Republic

European Union