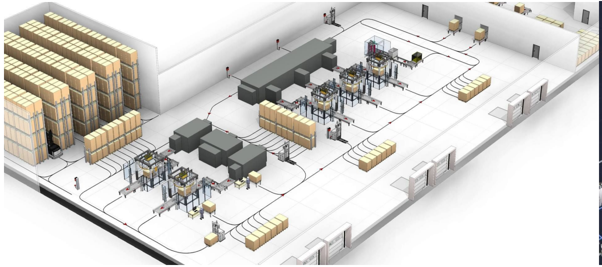
Advanced Multi-objective Facility Layout Planning for Modern Manufacturing Environments

Thomas Seidelmann
Faculty of Computer Science
Otto-von-Guericke-Universität Magdeburg, Germany
thomas.seidelmann@ovgu.de

Table of Contents

Motivation



Today’s topic is all about efficient manufacturing:

- ❖ Minimizing cost
- ❖ Minimizing production time

→ Facility Layout Planning (FLP) and Job-Shop Scheduling (JSP)

Motivation

https://ftt.roto-frank.com/de-de/unternehmen/presse/pressemeldungen/detail/fachpressetag-2021-ftt-fuer-roto-vieles-sehr-gut-gelaufen/

Introduction to Facility Layout Planning

"[FLP] is the problem of determining the most efficient physical arrangement of a number of interacting facilities on the factory floor of a manufacturing system in order to meet one or more objectives"

Ripon et al. (2012)

Introduction to Facility Layout Planning

How can we formalize it? → The Quadratic Assignment Problem

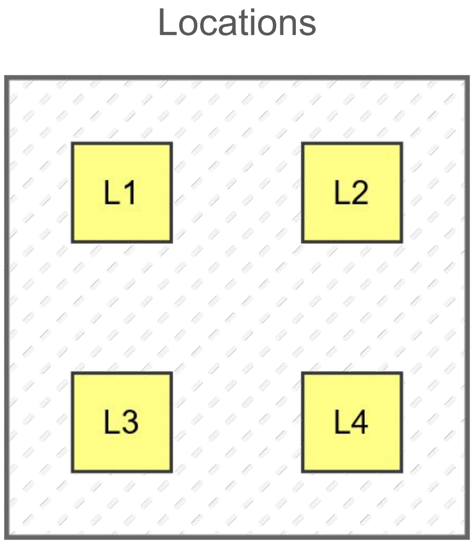
We have given:

- n machines
- n locations
- A distance matrix between locations
- A flow matrix between machines

→ Which assignment of machines to locations minimizes the distance weighted material flow between them?

Introduction to Facility Layout Planning

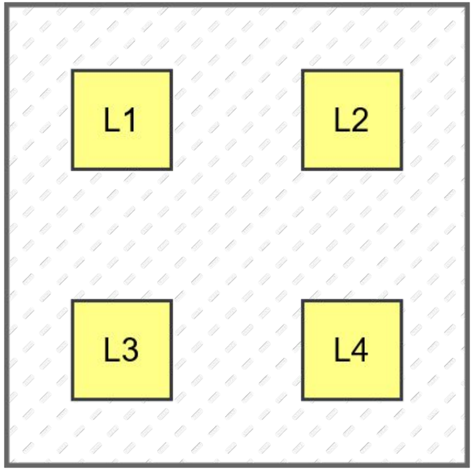
Example: We have 4 machines and 4 locations as follows…



Introduction to Facility Layout Planning

Example: We have 4 machines and 4 locations as follows…

Locations



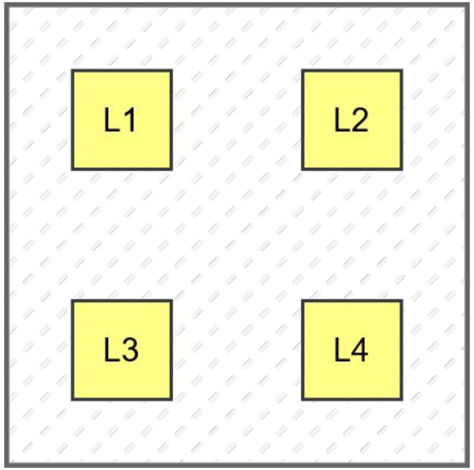
Distance Matrix

	L1	L2	L3	L4
L1		1	1	2
L2	1		2	1
L3	1	2		1
L4	2	1	1	

Introduction to Facility Layout Planning

Example: We have 4 machines and 4 locations as follows…

Locations



Distance Matrix

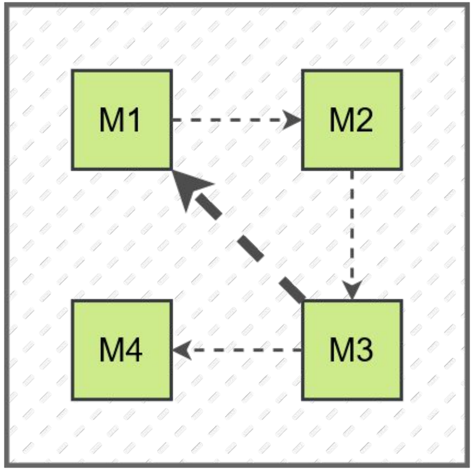
	L1	L2	L3	L4
L1		1	1	2
L2	1		2	1
L3	1	2		1
L4	2	1	1	

Flow Matrix

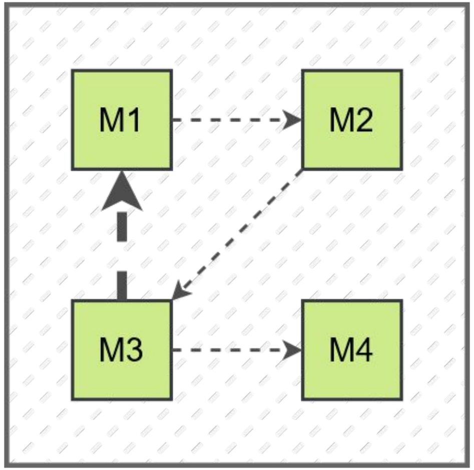
	M1	M2	M3	M4
M1		1		
M2			1	
M3	10			1
M4				

Introduction to Facility Layout Planning

Example: Let's solve it… (sum of material flows times distances)



Layout 1:
1 x 1 + 1 x 1 + 1 x 1 + 2 x **10** = **23**



Layout 1:
1 x 1 + 2 x 1 + 1 x 1 + 1 x **10** = **14**

Introduction to Job-Shop Scheduling

“[JSP] is the problem of allocating machines to competing jobs overtime, subject to the constraint that each machine can handle at most one job at a time.”

Mascis and Pacciarelli (2002)

Introduction to Job-Shop Scheduling

Formalization of the basic Job-Shop Scheduling problem

We have given:

- A set of jobs: J
- A set of operations: O
- A set of machines: M

, where each operation…

- belongs to a specific job
- requires a specific machine
- has a specific processing time
- may have precedence constraints

→ Which schedule of operations on the machines minimizes the makespan?

Introduction to Job-Shop Scheduling

Example: We have 3 machines and 3 jobs as follows…

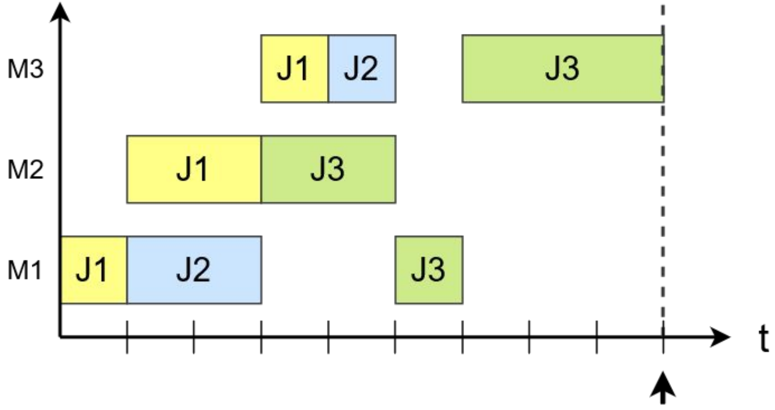
Job / Operations Matrix

	$o_{j,1}$	$o_{j,2}$	$o_{j,3}$
j_1	m_1, 1	m_2, 2	m_3, 1
j_2	m_1, 2	m_3, 1	
j_3	m_2, 2	m_1, 1	m_3, 3

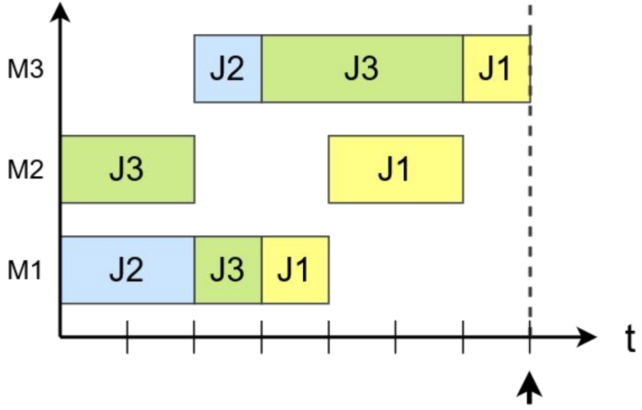
- Job 1 has 3 operations
- Its operations must be processed in order ($o_{1,1} \rightarrow o_{1,2} \rightarrow o_{1,3}$)
- Cells indicate required machine and processing time

Introduction to Job-Shop Scheduling

Example: Let’s solve it… (time of completion)



Schedule 1: Makespan = **9**



Schedule 2: Makespan = **7**

Summary of the Basics

Facility Layout Planning (FLP):

Positioning machines on a shop floor to minimize the total distance of product flows.

→ Material Handling Cost

Job-Shop Scheduling (JSP):

Scheduling operations on machines to minimize the total required processing time.

→ Makespan

I. Introduction

- ➢ Facility Layout Planning (FLP) & Job-Shop Scheduling (JSP)
- ➢ Research Landscape
- ➢ Research Project MOSAIK

II. The integrated FLP and JSP Problem

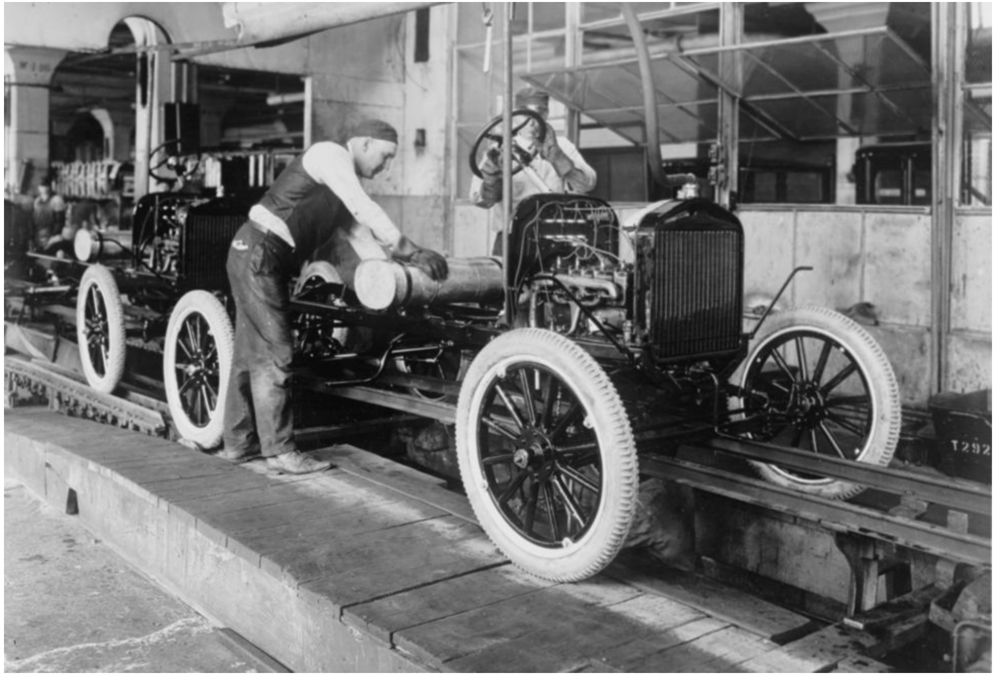
III. Solving the Multi-Objective Optimization Problem

IV. Conclusion

Research History

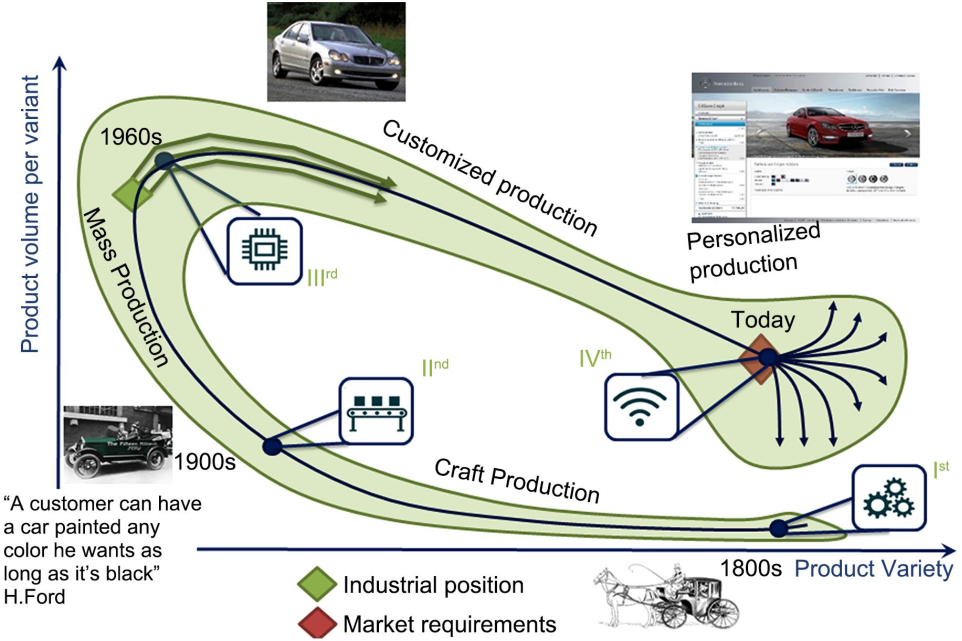
The problem of efficient manufacturing is not new.

However, the manufacturing landscape continues to change over time and new challenges emerge.



https://www.thoughtco.com/henry-ford-and-the-assembly-line-1779201

Paradigm Shifts in the Industry



"A customer can have a car painted any color he wants as long as it's black"
H.Ford

Asadollahi-Yazdi et al. (2020)

Paradigm Shifts in the Industry

Challenges today:

- Increased product variety
- Shorter life cycles
- Smaller lot sizes
- Non-stable demand
- Increased manufacturing complexity

Asadollahi-Yazdi et al. (2020), Maganha et al. (2019)

Influence on FLP and JSP

As a result of these paradigm shifts, many variants of FLP and JSP emerged and continue to emerge:

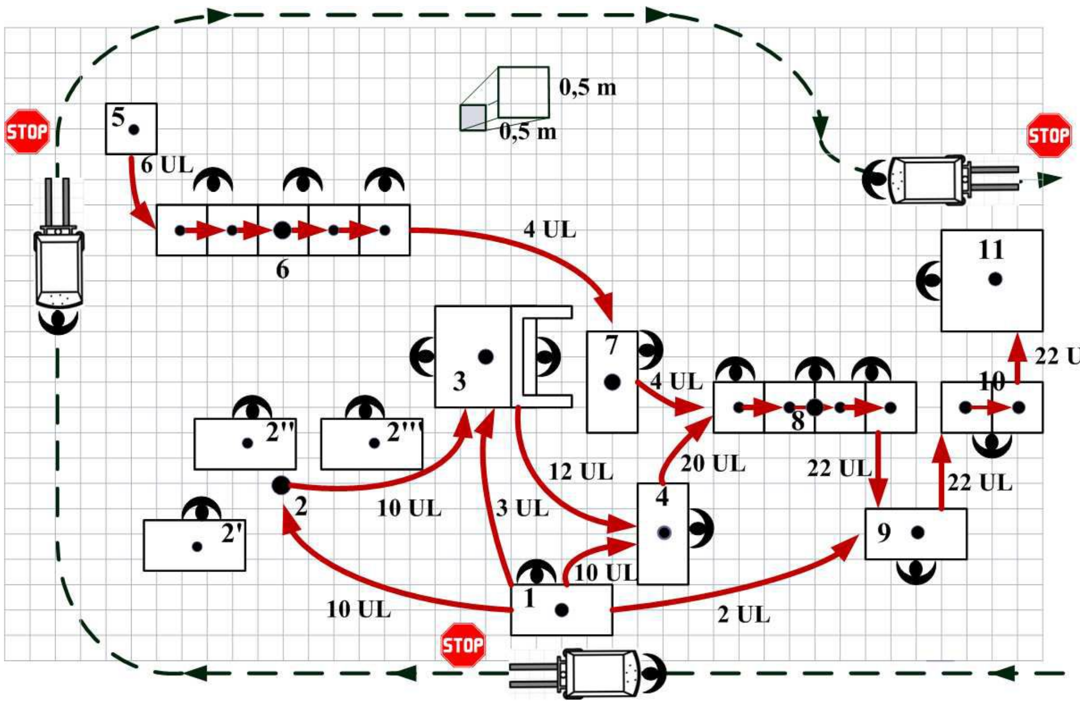
Hits for Google Scholar queries “Facility Layout Planning” & “Job Shop Scheduling” in two year intervals

Examples of FLP variations

- Unequal Area FLP
- Qualitative constraints (adjacency factors)
- Multi-Facility Layout Planning
- Single-Period / Multi-Period Planning
- Workflow Interference
- …

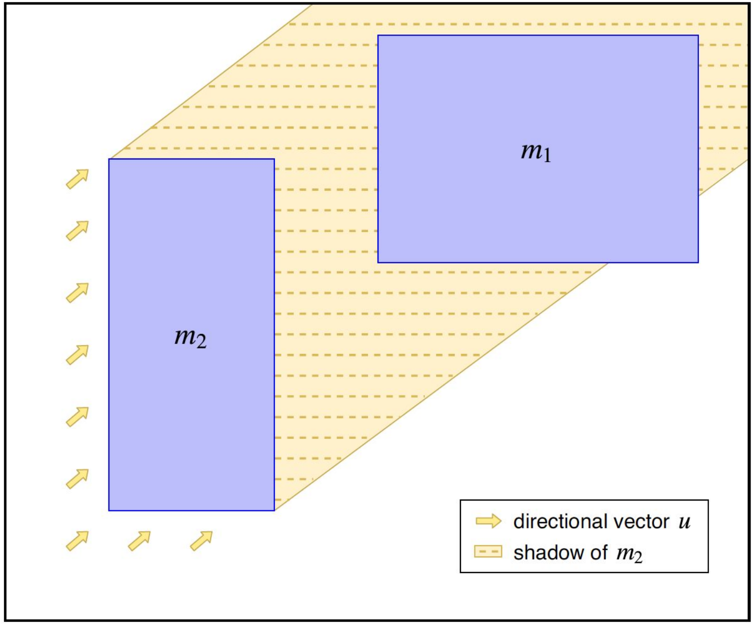
Let’s take a short look.

Unequal Area FLP



Kovács and Kot (2017)

Constraints & Adjacency Factors



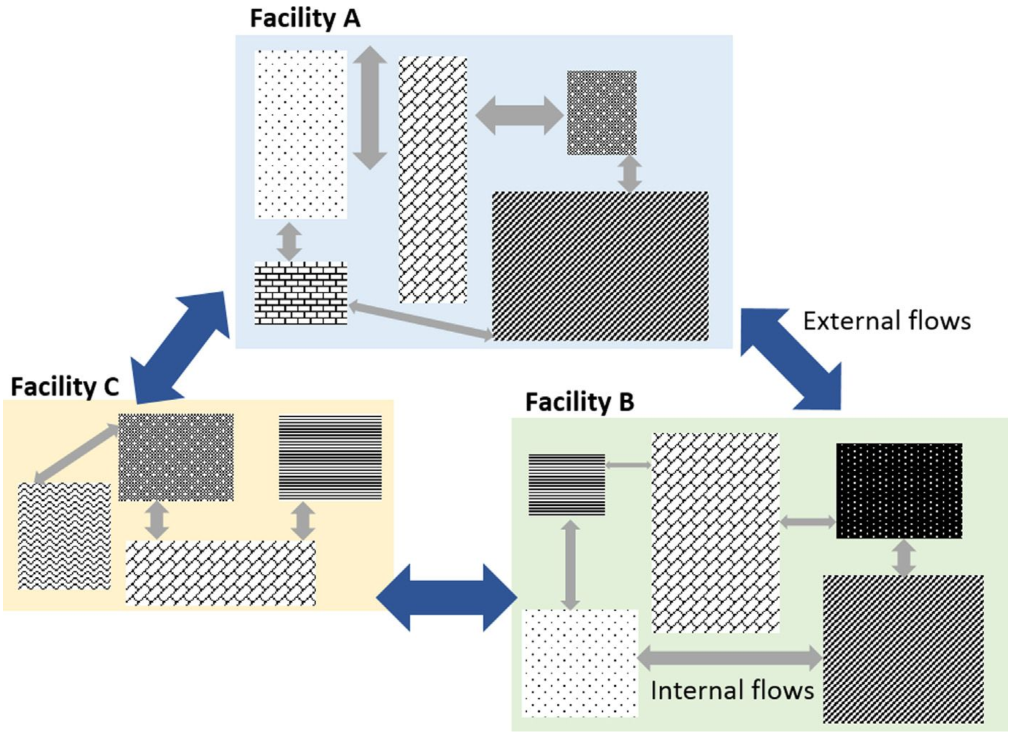
Geometrically Relative Order, Wen and Ting (2018)

i,j	1	2	3	4	5	6
1	-	O	E	A	A	I
2	O	-	X	I	U	O
3	E	X	-	O	A	U
4	A	I	O	-	U	O
5	A	U	A	U	-	X
6	I	O	U	O	X	-

A: Absolutely necessary; E: Especially important; I: Important; O: Ordinary close; U: Unimportant; and X: Undesirable

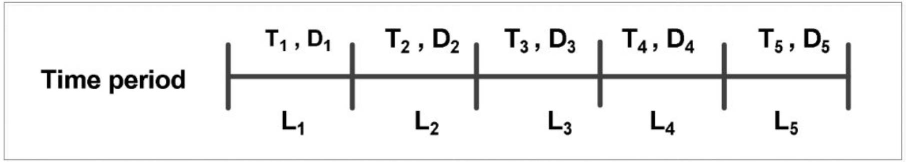
Adjacency Matrix, Tayal and Singh (2019)

Multi-Facility Layout Planning



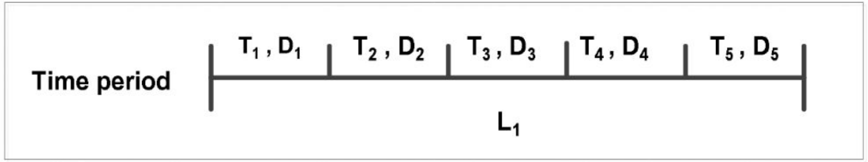
Azevedo et al. (2017)

Multi-Period Layout Planning



a) Re-layout

Designing a new layout for each period



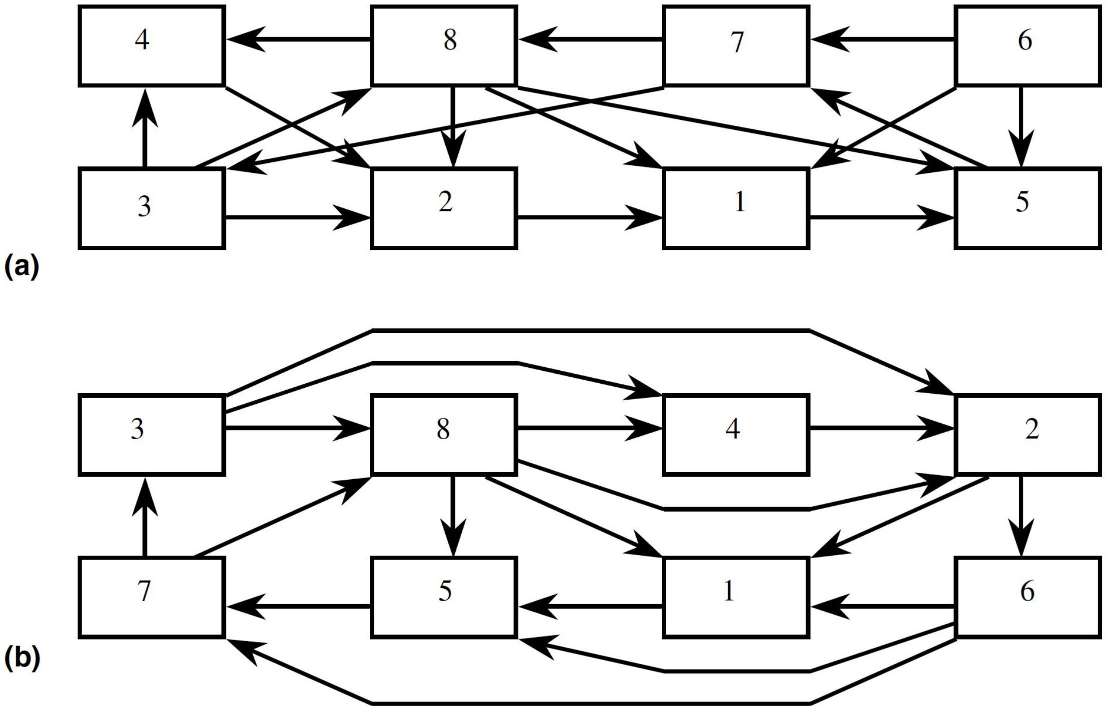
b) Robust layout

Designing a layout that works for all periods

Vitayasak and Pongcharoen (2015)

Minimizing Workflow Interference

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG



Many intersections:
High Conflict Potential

No intersections:
Little Conflict Potential

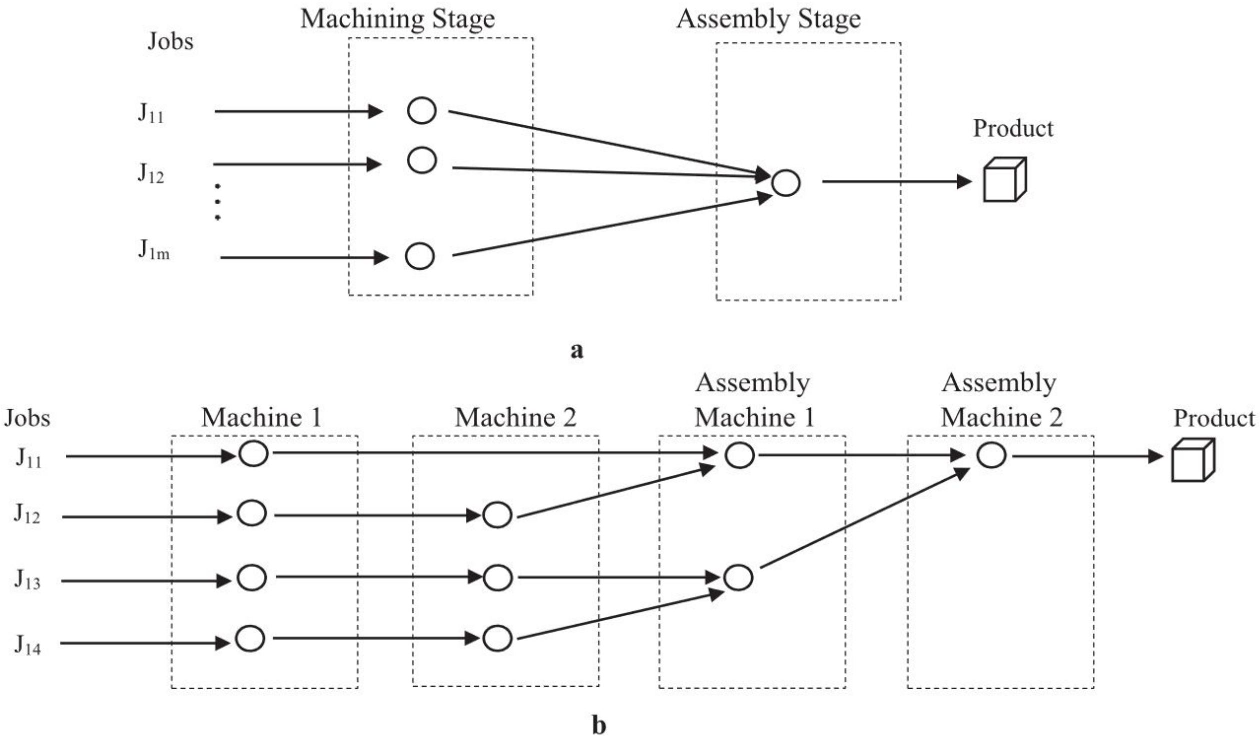
Chiang et al. (2006)

Examples of JSP variations

- Job interdependencies
- Flexible JSP
- Robust JSP
- Dynamic JSP
- Multi-fidelity Models
- …

Let’s take a short look.

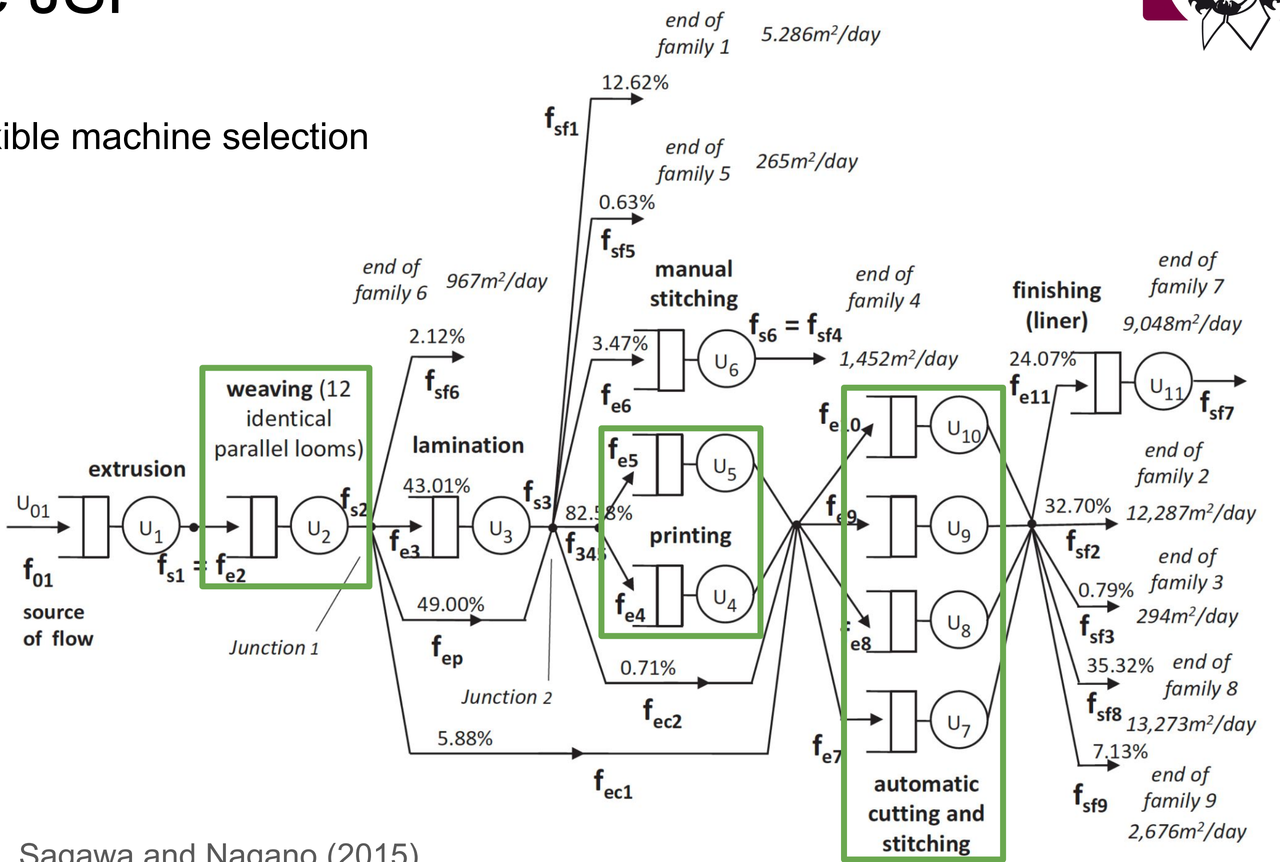
Job Interdependencies



Komaki et al. (2019)

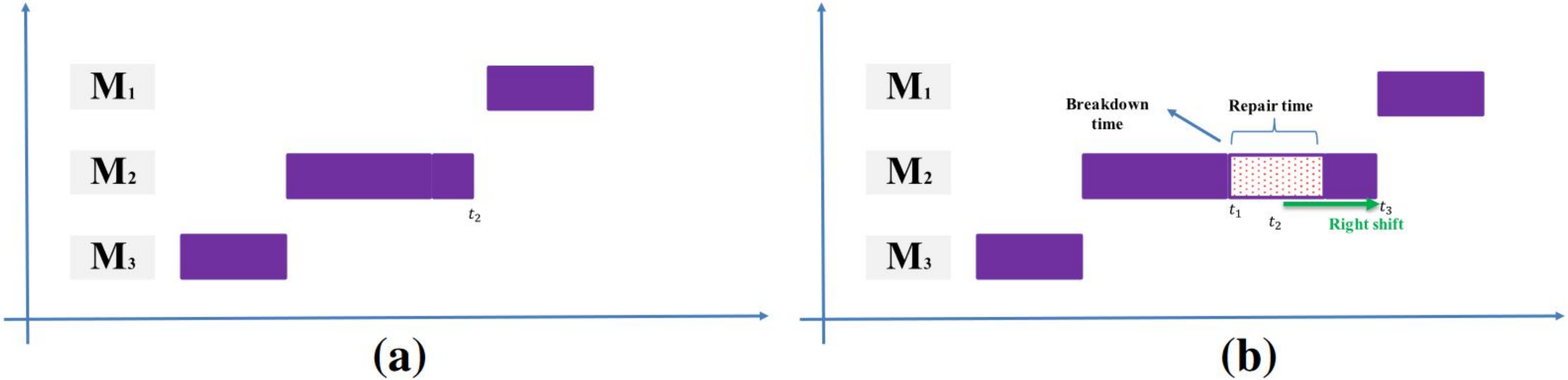
Flexible JSP

= flexible machine selection



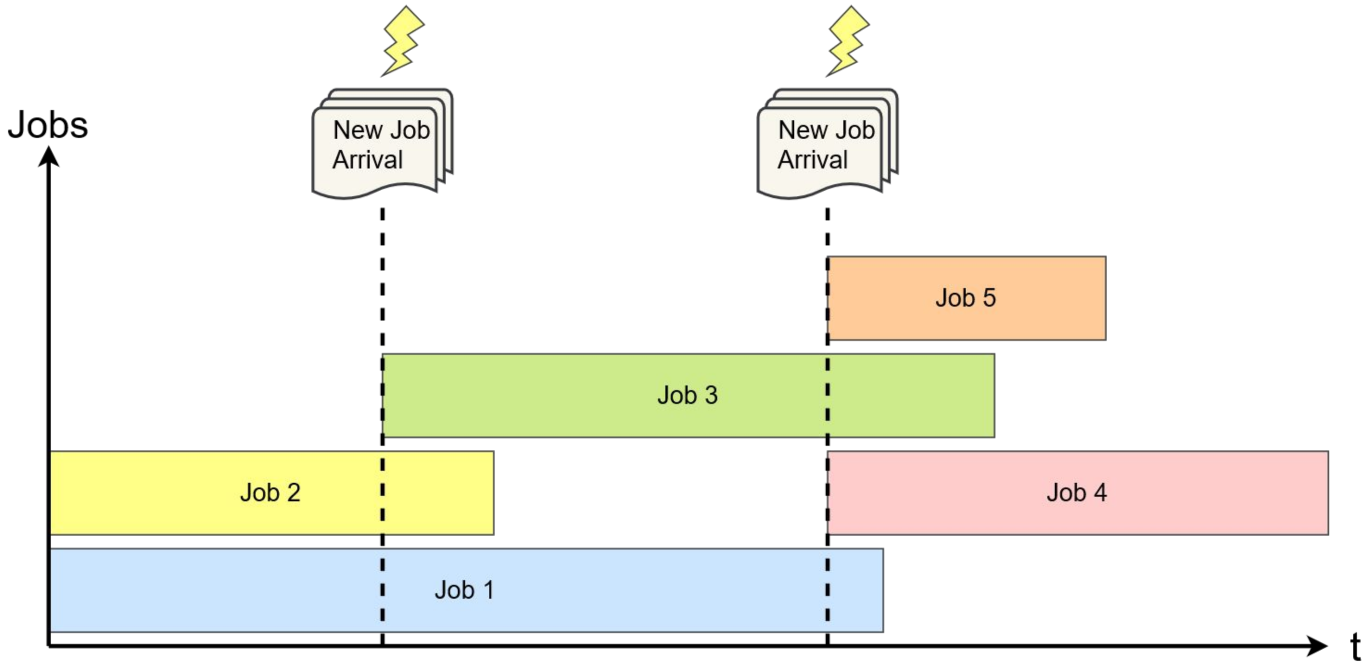
Sagawa and Nagano (2015)

Robust JSP



Teymourifar et al. (2020)

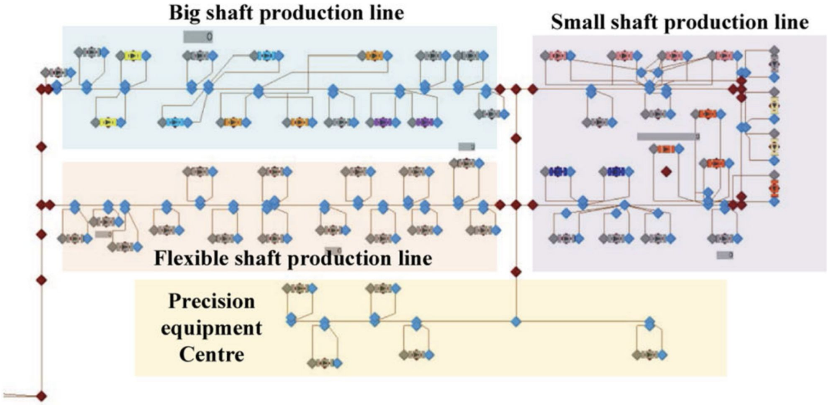
Dynamic JSP



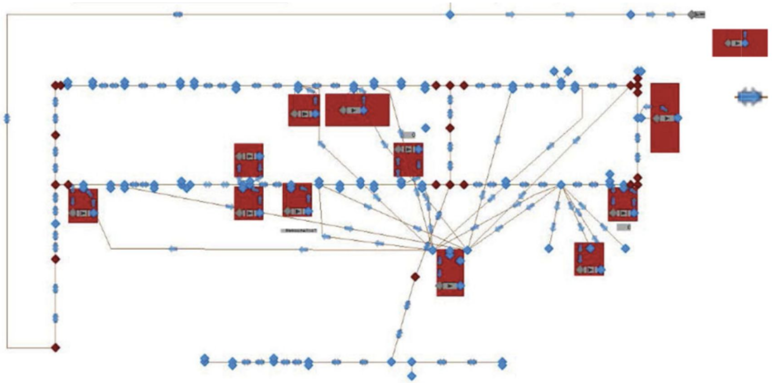
described by Kundakci and Kulak (2016) for example

Multi-Fidelity Models

High-fidelity model



Low-fidelity model



Zhang et al. (2022)

I. Introduction

Project Partners

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

MOSAIK

FAU FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

BOSCH

DFKI Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

NETSYNO

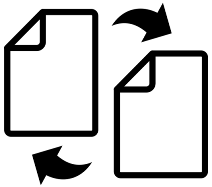
FESTO

Factory of the Future @ Arena2036

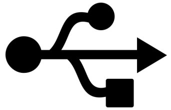
Factory of the Future @ Arena2036

The Vision

A big problem for industry 4.0 is the interoperability of systems:



- Much research aims at creating a universal interface
- Often realized via IoT networks (Internet of Things)
- This research field is very crowded
- Big players from the industry have much influence



The Vision

In MOSAIK, we thought ahead:

- We assumed that a common interface is already established
- Focus on the challenges that lie beyond
- Large degree of automation and connectivity (cyber-physical system)
- Highly dynamic environment

→ How to make the most of it?

The Vision

Due to the complexity, we focused only on the scheduling problem…

- A high fidelity simulation model was developed
- Multiple scheduling approaches were developed
- Several papers published

The facility layout problem remained.

I. Introduction

II. The integrated FLP and JSP Problem

- ➢ Problem Formulation
- ➢ Modeling
- ➢ Solution Approach

III. Solving the Multi-Objective Optimization Problem

IV. Conclusion

Problem Formulation: Challenges

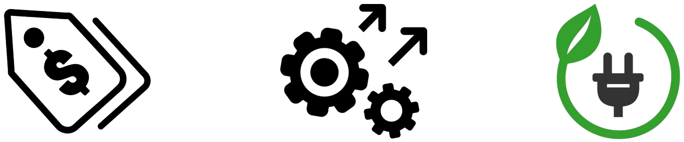
The FLP we now want to solve involves:

- Make to Order / Lot Size One
- Unstable demand
- Flexible manufacturing capabilities
- Job interdependencies
- Transporter management
- Machine selection
- Re-layouting of existing layouts
- Multiple conflicting objectives

Problem Formulation: Objectives

The objectives we want to optimize are as follows:

- Re-layouting cost
- Flow time
- Idle time



Problem Formulation: Integration

How to solve the FLP for this problem, when…

- Material flow is not static?
- Transportation times are not static?
- We don’t know a priori how many machines of which type are ideal?
- Traditional FLP does not predict flow time and idle time?

→ We need to solve the scheduling problem for any given layout to evaluate it!

Problem Formulation: Integration

- Multi-objective,
- robust multi-period,
- machine selection

FLP

- Flexible,
- dynamic,
- interdependent

JSP

Integration

Both are independently NP-hard

Problem Formulation: Parameters

Let's formalize:

- Shopfloor is a grid s_x x s_y
- Workstations can be assigned to a set of cells P
- Workstation types that can be assigned are in a set W
- Workstation allocations are given by the set A
- There is a number of transporters t_n
- An allocation with transporters is a layout L
- An original layout L_o exists

→ We need to find a new layout that improves upon L_o optimally

Problem Formulation: Assumptions

We make the following assumptions:

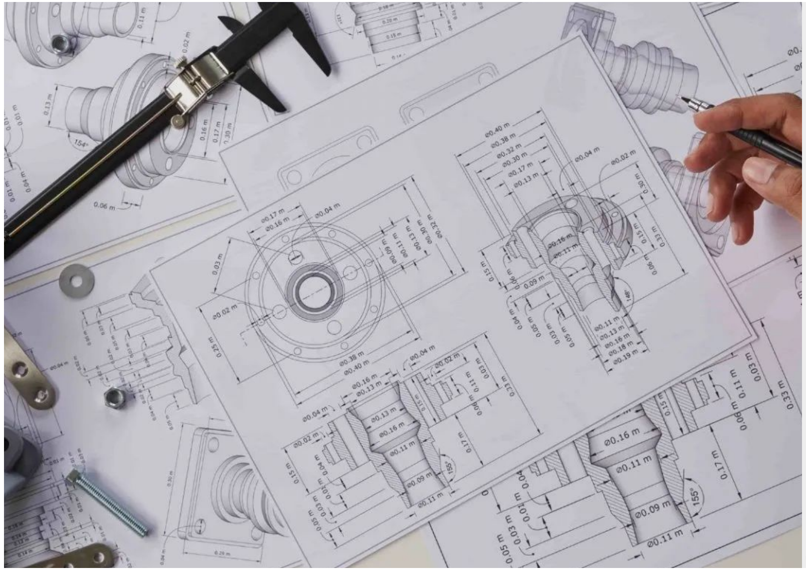
- Number of orders at each time is constant
- All orders are random
- We have given:
  a. Possible products
  b. Precedence relations
  c. Processing times
  d. Transporter speeds
- Unlimited capacities, no collisions, no preemption, no malfunctions, …

Modeling


Output: 0
Idle
Input: 1
Carries: 4
Input: 1
Idle
Input: 0
Idle
Output: 0
Output: 0
Output: 0
Input: 0
Idle
Output: 0
Input: 0
Idle
Input: 0
Carries: 1
Idle
Output: 1
Initialize
Start
Step
Stop
Infobar
Transporter ID: Transporter2
Position: {4; 7}
Size: {1; 1}
Speed: 1.0
Status: busy
Capacity: 1001
Num Owned Products: 4
1. Product42 (Smartphone)
2. Product40 (Ram2Gib)
3. Product47 (Ram4Gib)
4. Product50 (Ram2Gib)
Active Task: TransporterTask456
Task Backlog...
1. TransporterTask456 ({9; 4})

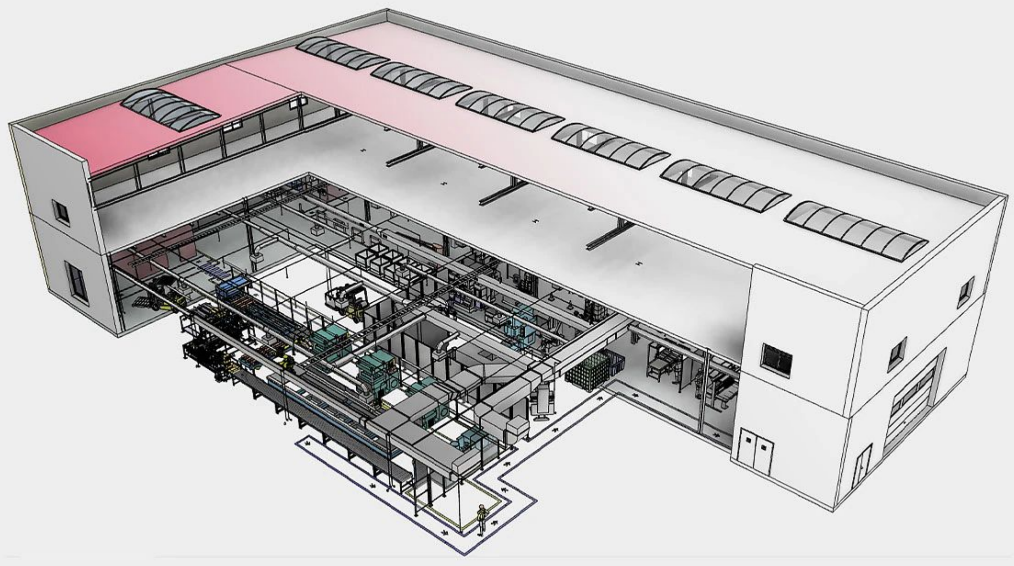
Simulation Architecture

Metamodel



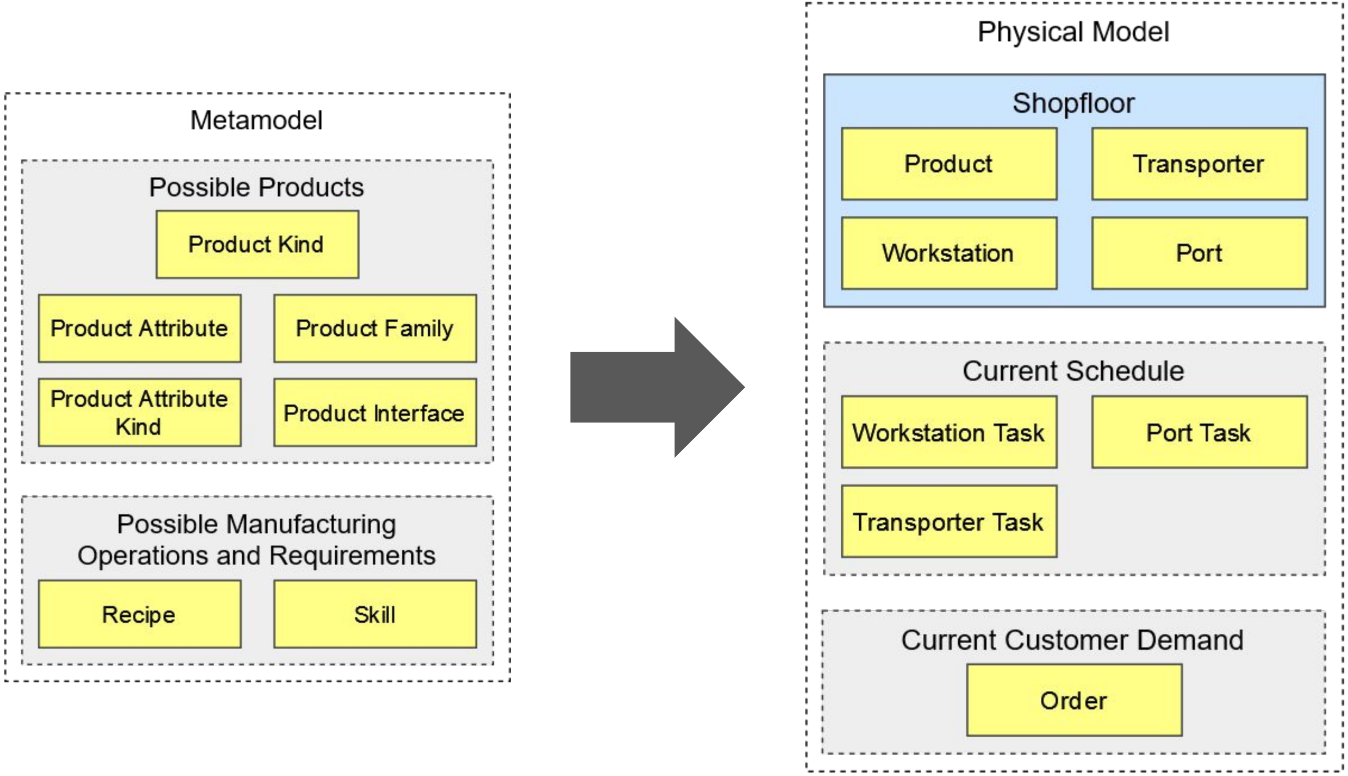
https://katanamrp.com/blog/how-to-read-manufacturing-blueprints/

Physical Model

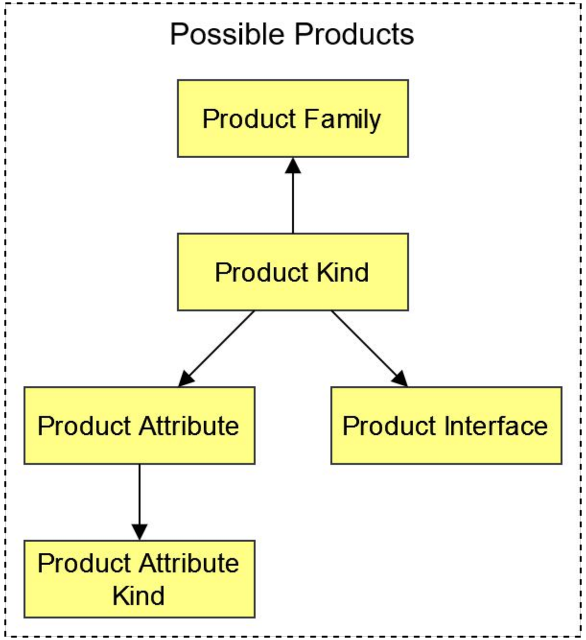


https://www.autodesk.com/products/factory-design-utilities

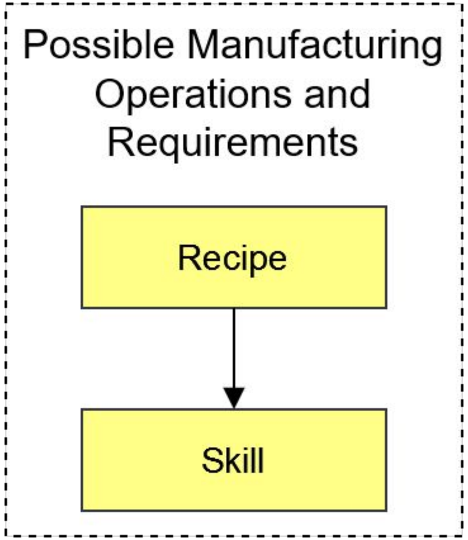
Simulation Architecture

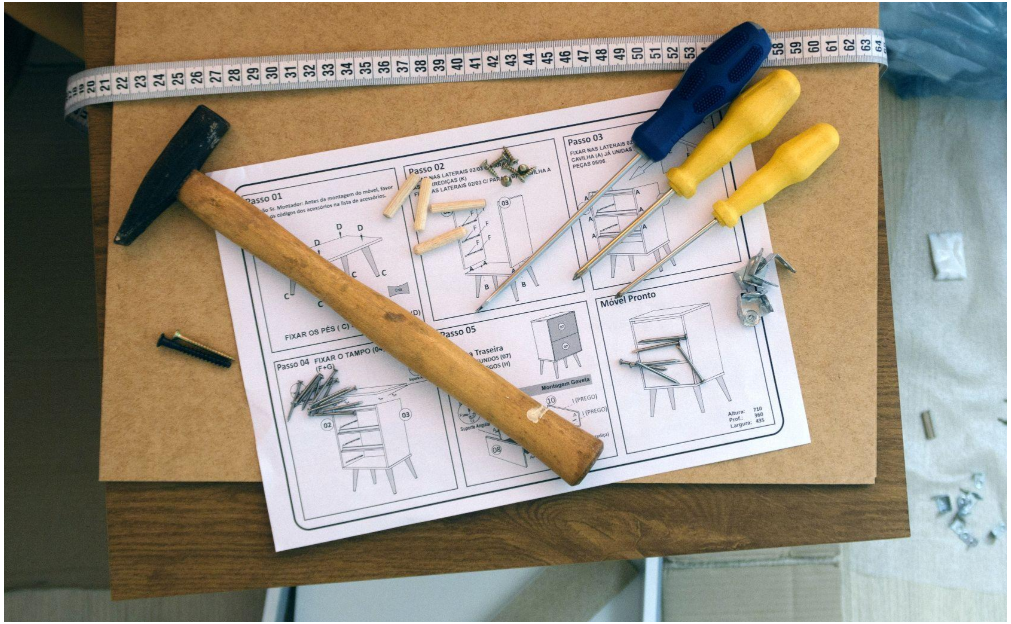


Simulation Architecture: Metamodel

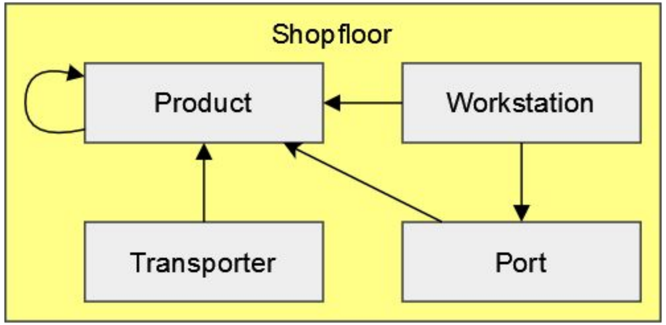


Simulation Architecture: Metamodel

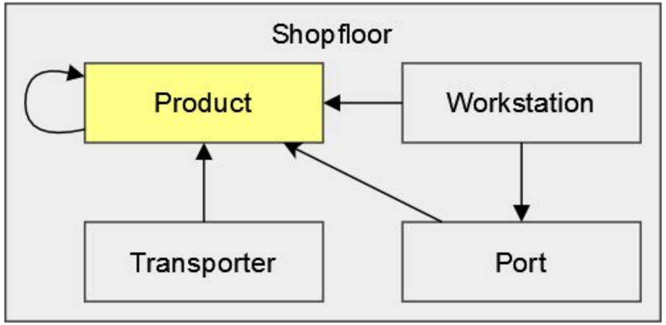




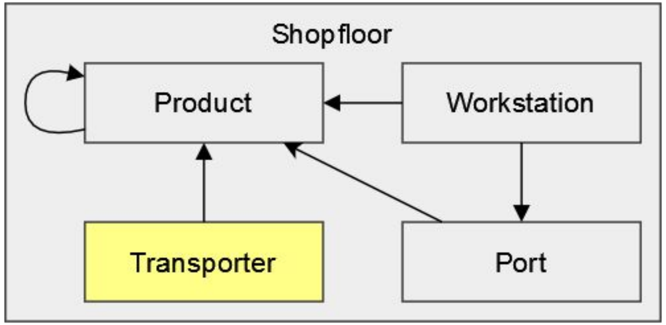
Simulation Architecture: Physical Model



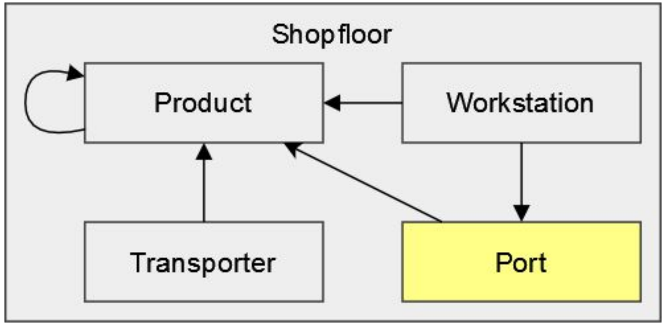
Simulation Architecture: Physical Model



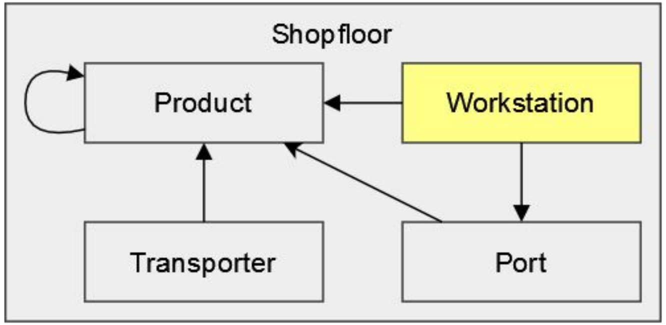
Simulation Architecture: Physical Model

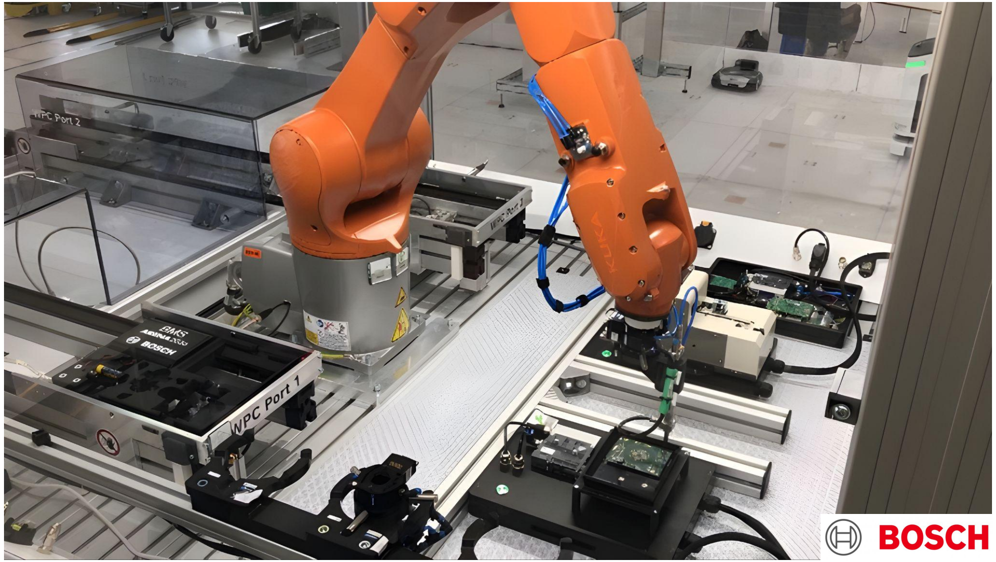


Simulation Architecture: Physical Model



Simulation Architecture: Physical Model





Simulation Architecture: Physical Model

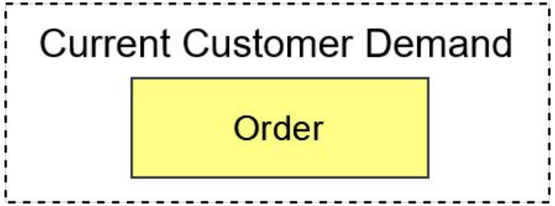
Current Schedule

Workstation Task

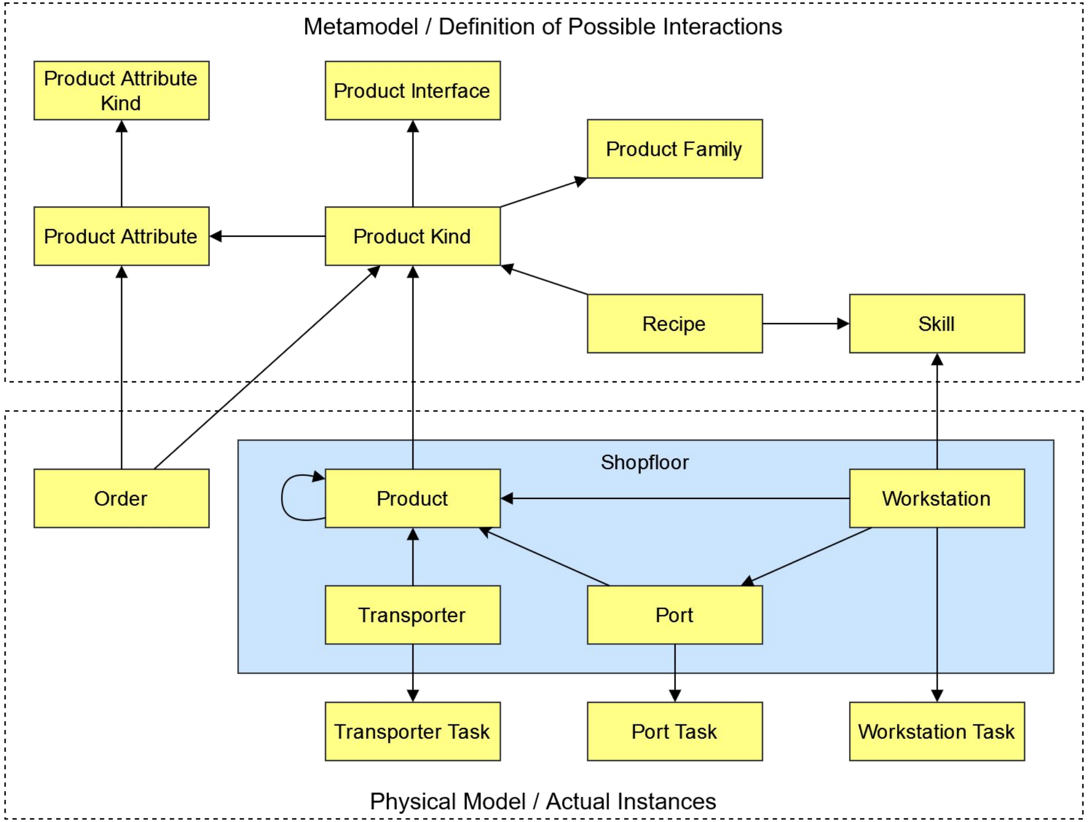
Transporter Task

Port Task

Simulation Architecture: Physical Model



Simulation Architecture



OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

Solution Approach

We want to solve this FLP with multi-objective evolutionary algorithms.
For every layout, we need to solve the JSP:

- Co-evolution of FLP and JSP difficult (strict layout dependency)
- Nested optimization causes prohibitive computing times

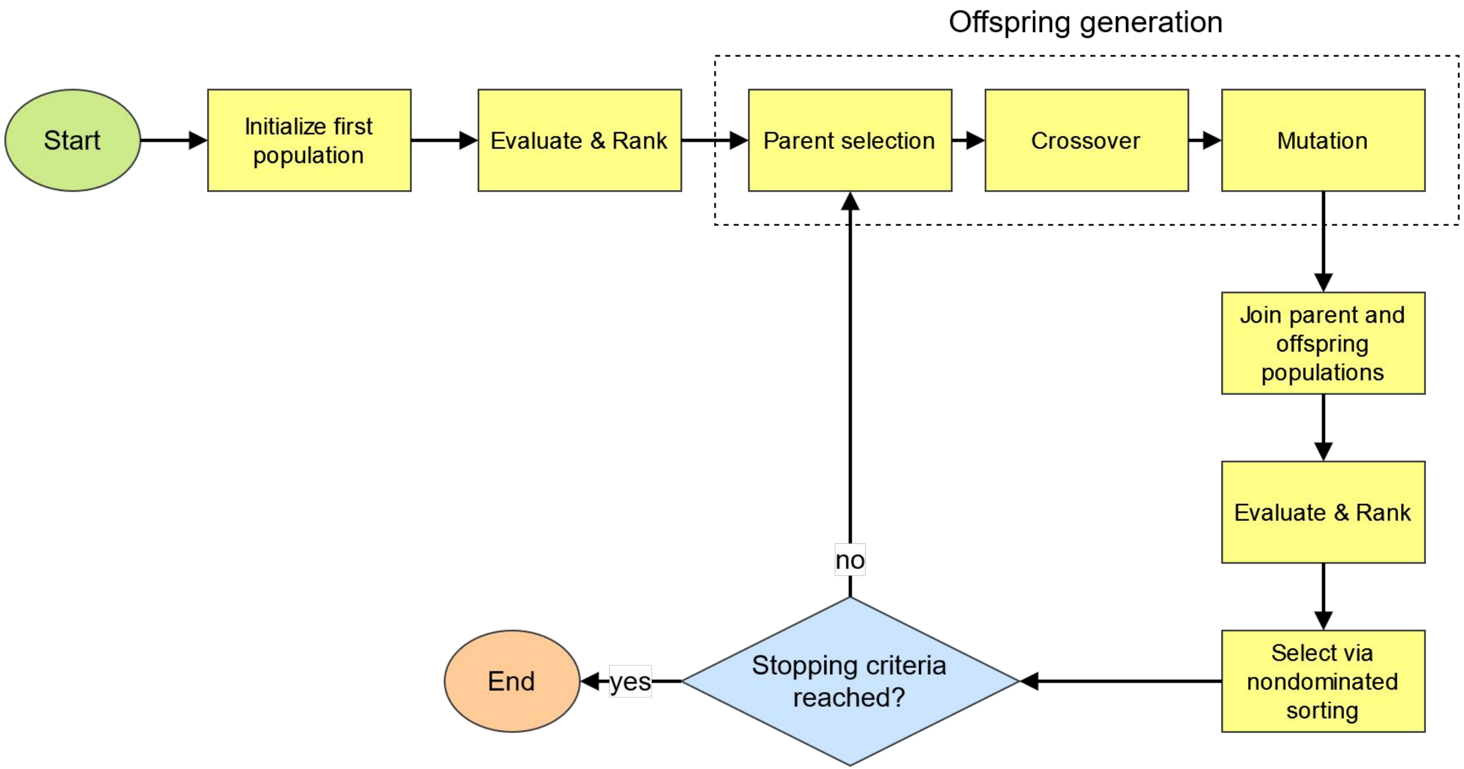
→ We use a simple heuristic scheduling algorithm

Solution Approach

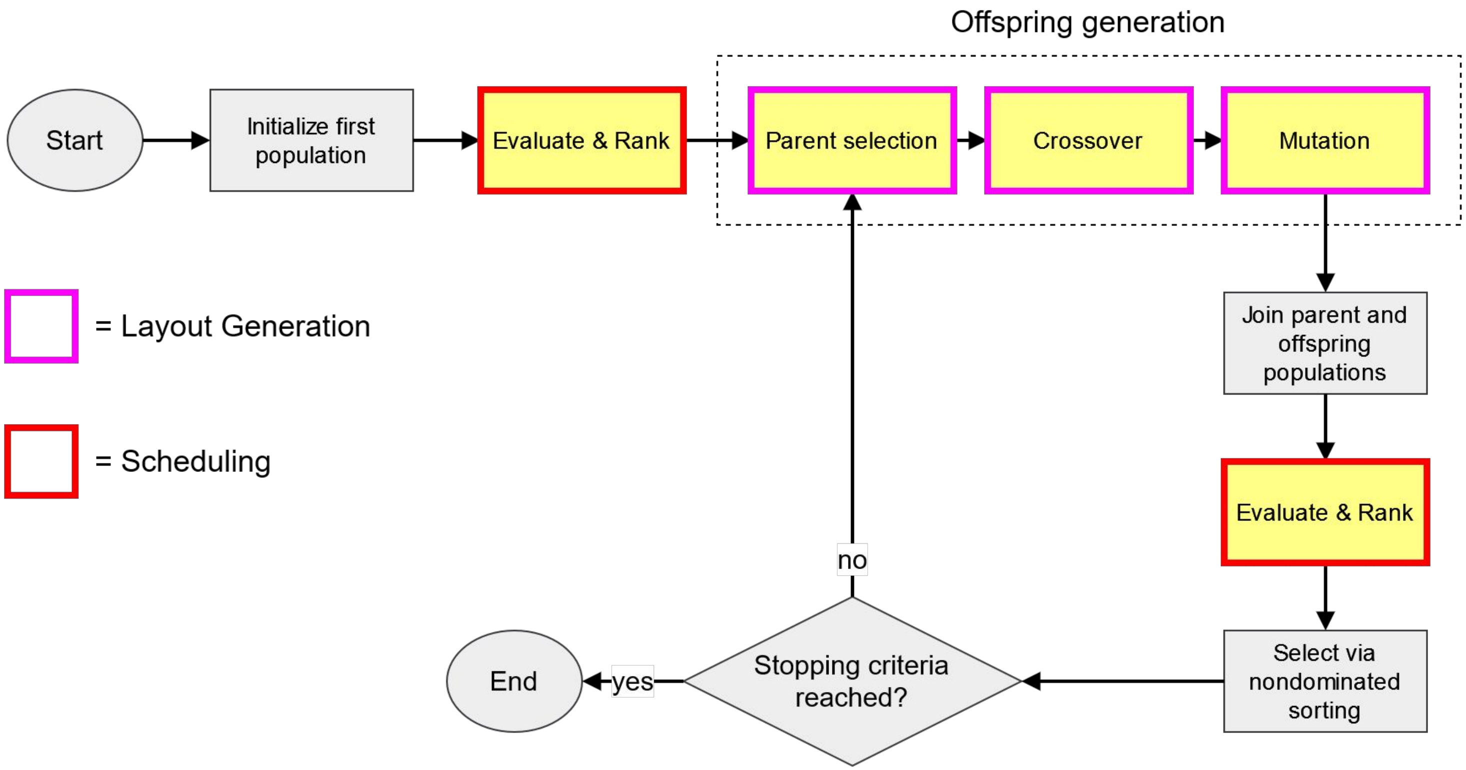
The simple heuristic scheduling algorithm…

- was developed during MOSAIK
- is a system of dispatching rules
- works on a greedy first in, first out basis
- cannot guarantee optimal, or near-optimal, schedules
- is very fast to compute

Solution Approach



Solution Approach



I. Introduction

II. The integrated FLP and JSP Problem

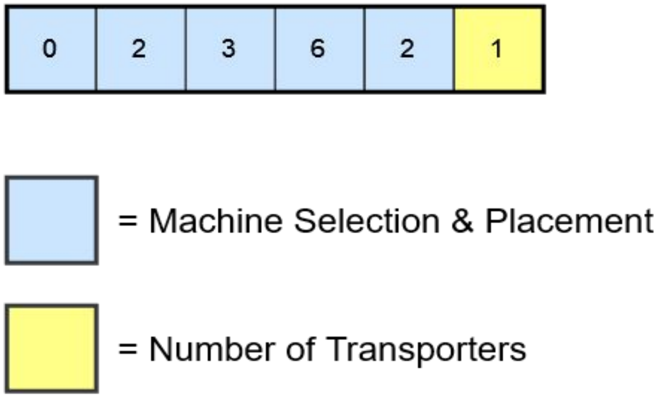
III. Solving the Multi-Objective Optimization Problem

- ➢ Encoding
- ➢ Operators
- ➢ Results & Discussion
- ➢ Future Work

IV. Conclusion

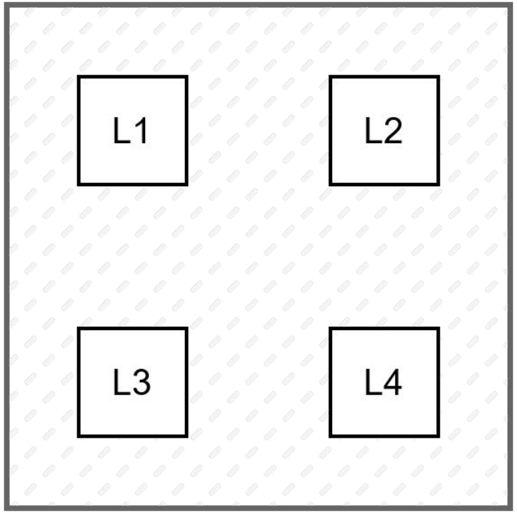
Encoding

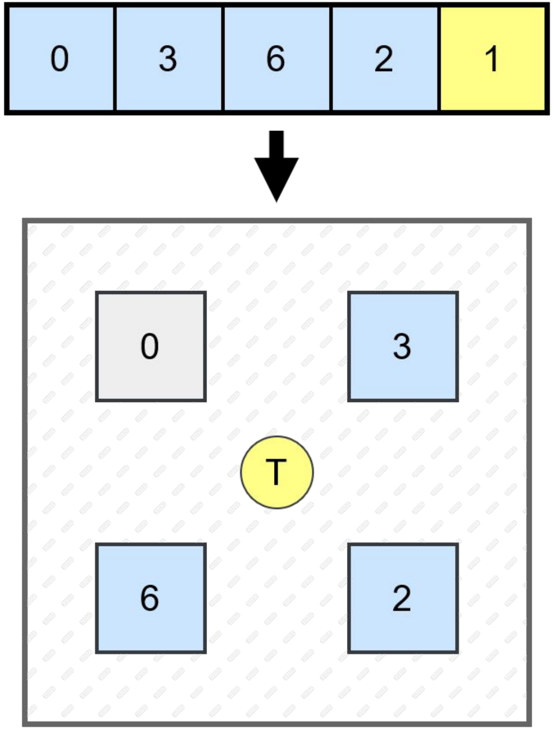
- Sequence of integers
- Length is number of possible allocations +1
- The first n-1 integers map machines to positions on the shop floor
- The final value represents the number of available transporters



Encoding Example

Possible locations:





I. Introduction

II. The integrated FLP and JSP Problem

III. Solving the Multi-Objective Optimization Problem

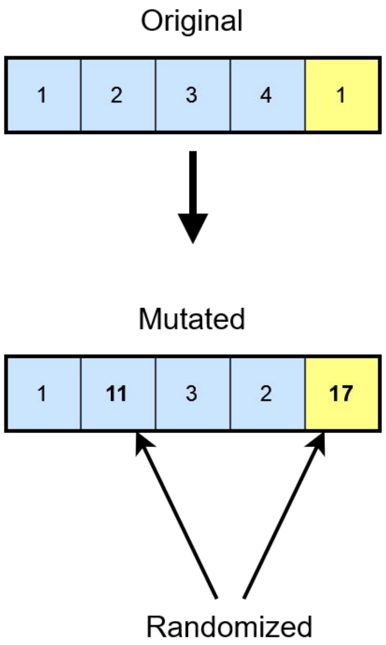
- ➢ Encoding
- ➢ **Operators**
- ➢ Results & Discussion
- ➢ Future Work

IV. Conclusion

Standard Uniform Random Mutation

Parameters:

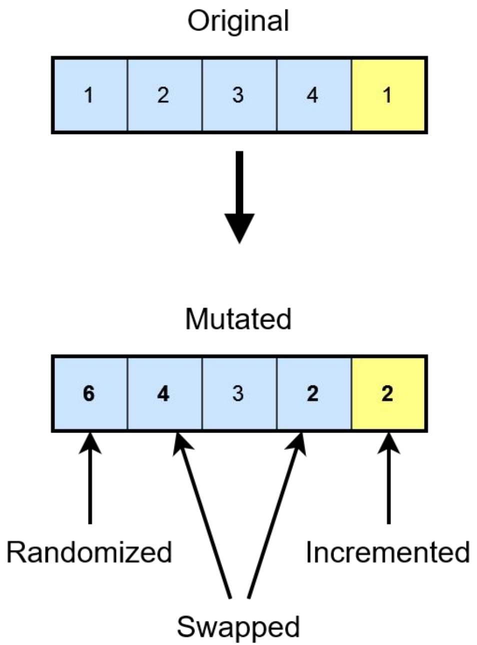
1. Probability to randomize workstation allocation



Randomizing: Assign random integer

Modified Mutation Operator

Parameters:
1. Probability to randomize workstation allocation
2. Probability to swap workstation positions
3. Probability to change the number of transporters



Randomizing: Equal chances to allocate 0 or a random integer

Swapping: Swap current value with random workstation

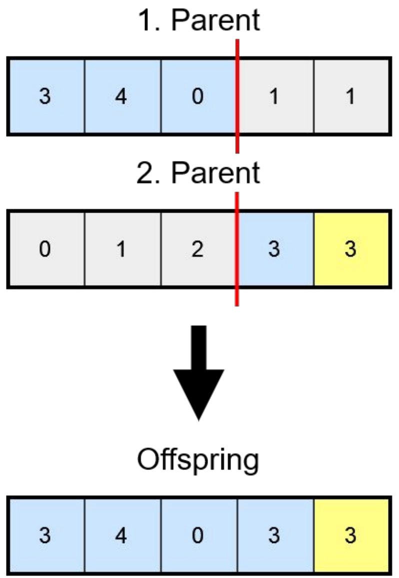
Changing transporters: Equal chances to add or subtract 1

Standard Single-Point Crossover

Parameters:

1. Probability for crossover

- Cut parents at common point
- Use one section from parent 1
- Use the opposite section from parent 2

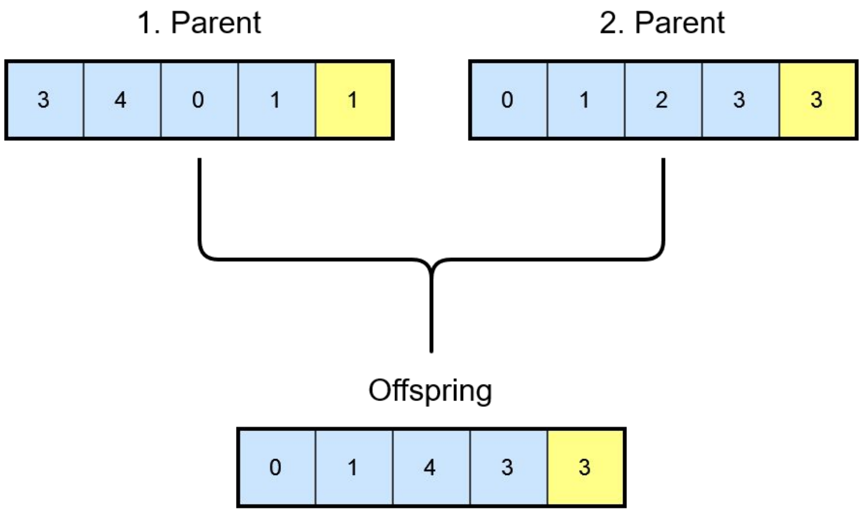


Modified Crossover Operator

Parameters:

1. Probability for crossover

- Take workstations from parent 1
- Allocate them to positions like in parent 2
- Take transporters from parent 2



I. Introduction

II. The integrated FLP and JSP Problem

III. Solving the Multi-Objective Optimization Problem

- ➢ Encoding
- ➢ Operators
- ➢ **Results & Discussion**
- ➢ Future Work

IV. Conclusion

Case Study Scenario

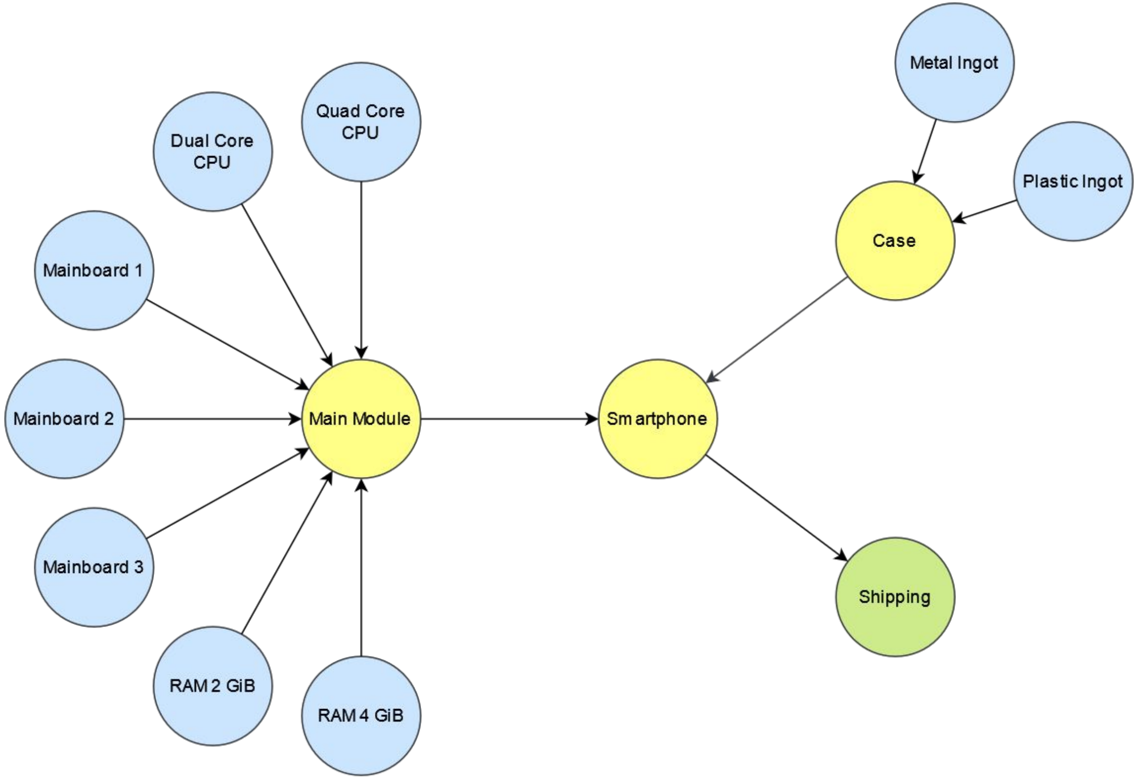
We use a synthetic smartphone production dataset with…

- 34 possible product variants
- 12 workstation types
- 20 locations for allocating workstations

Layouts are evaluated as follows:

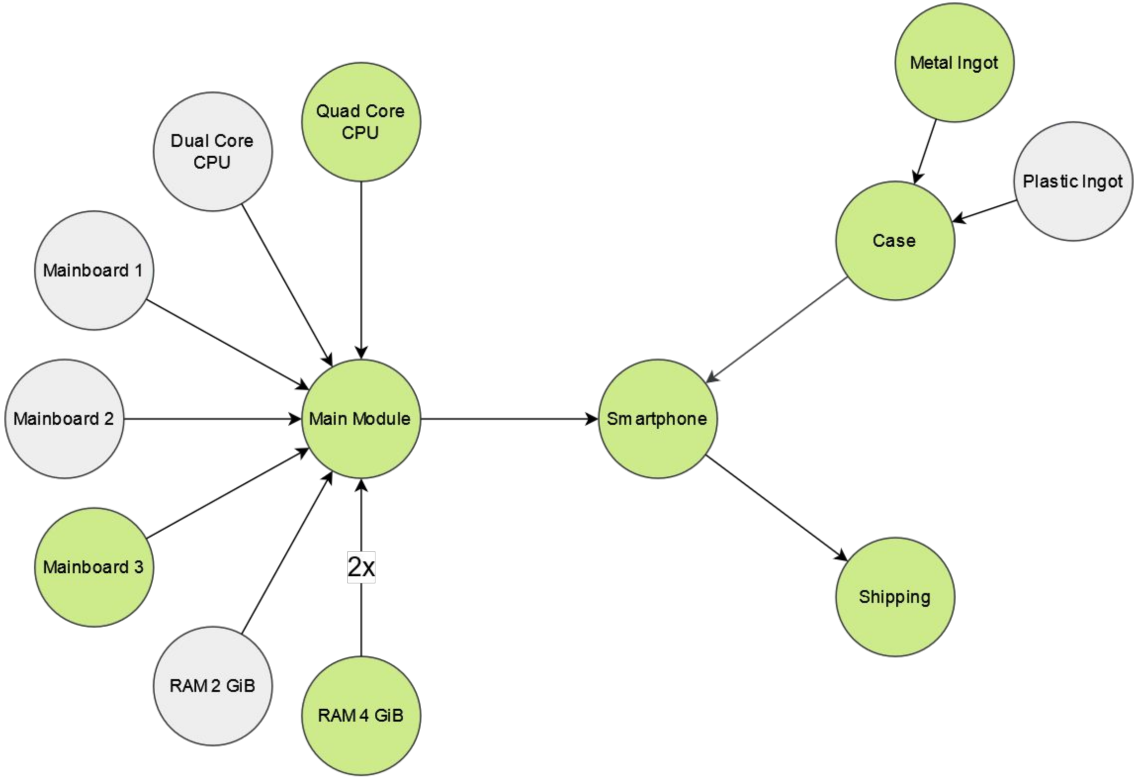
- The JSP is solved for 300 random orders on the layout
- 10 orders are being worked on simultaneously

Case Study Scenario - Precedence Graph



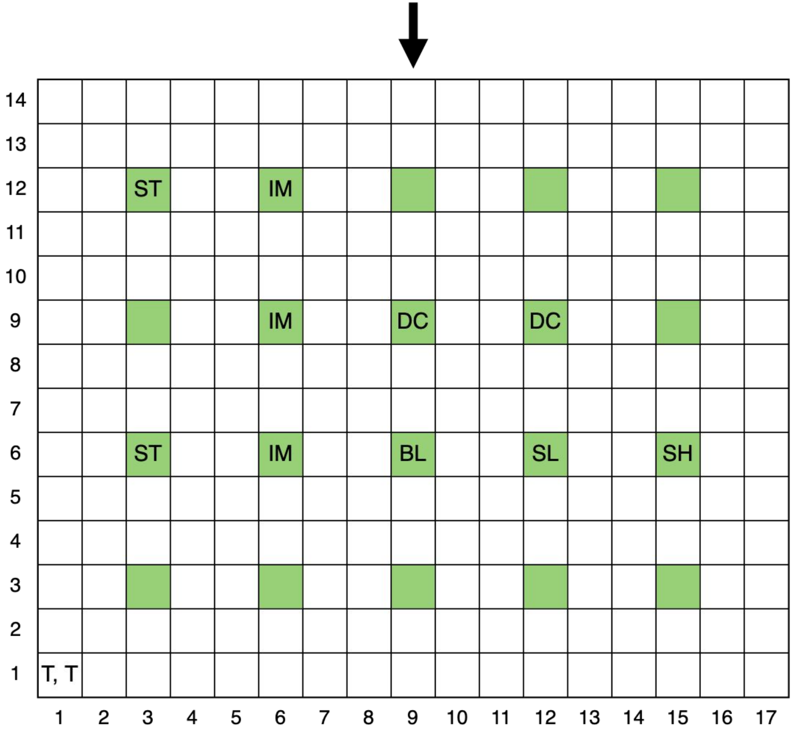
Case Study Scenario - Precedence Graph

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

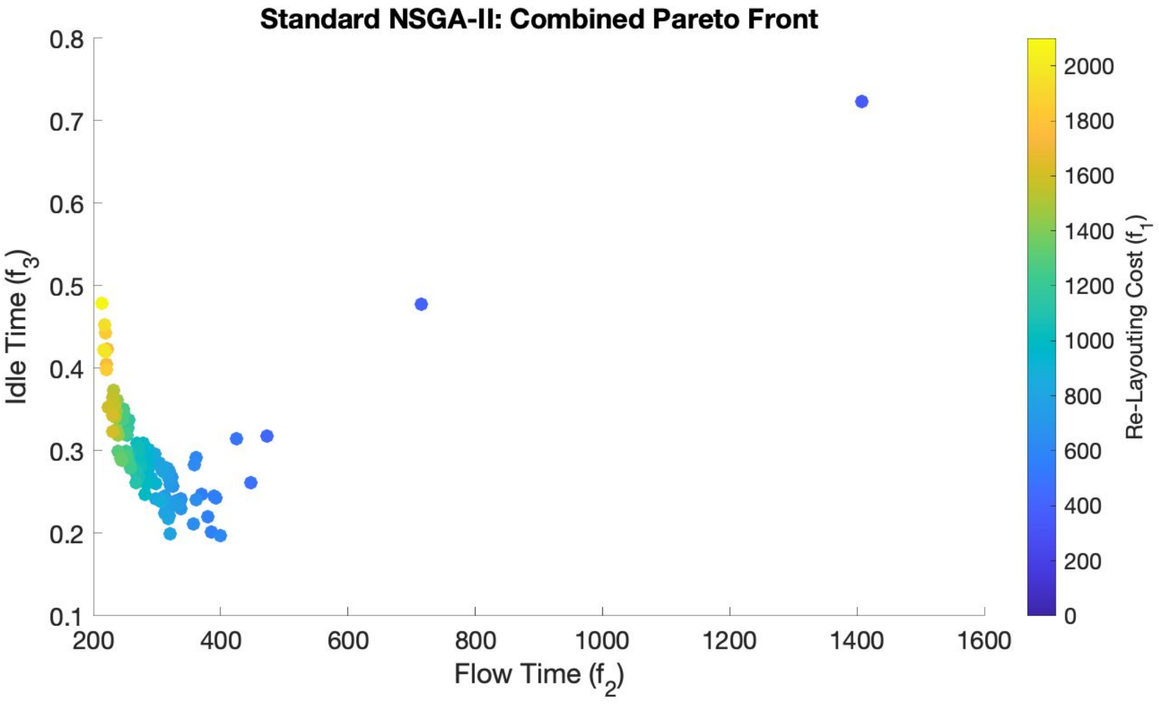


Case Study Scenario - Initial Layout

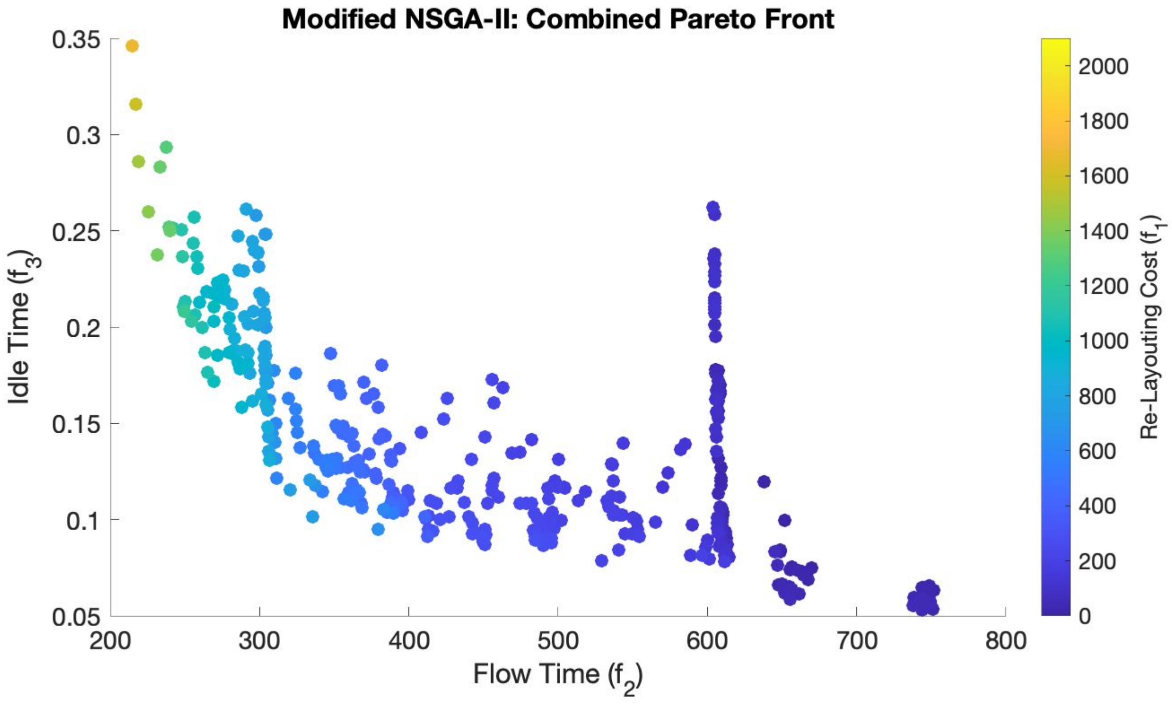
0	0	0	0	0	1	3	5	2	9	0	3	4	4	0	1	3	0	0	0	2



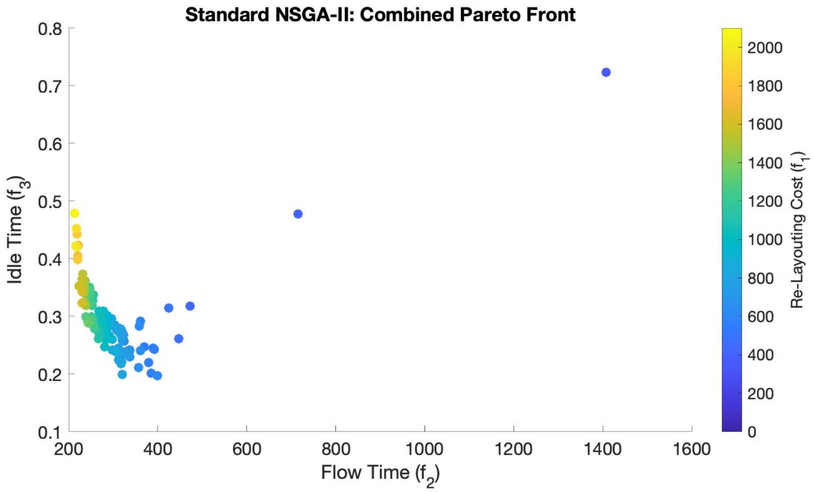
Results: Standard Operators

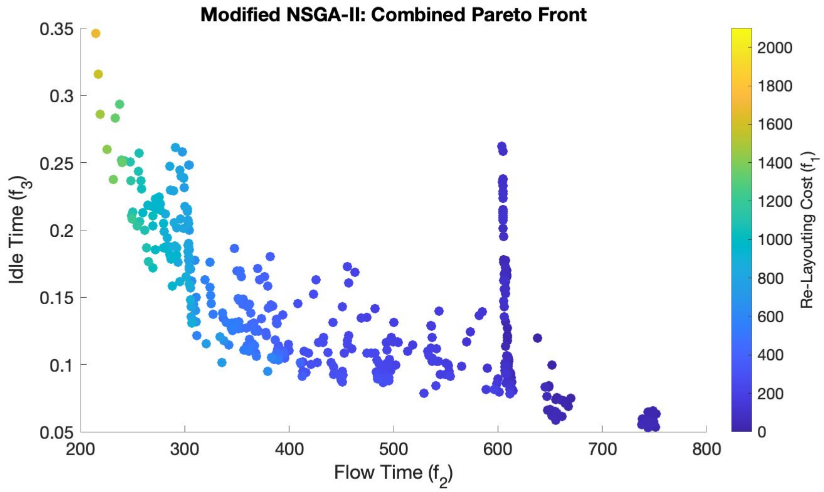


Results: Modified Operators

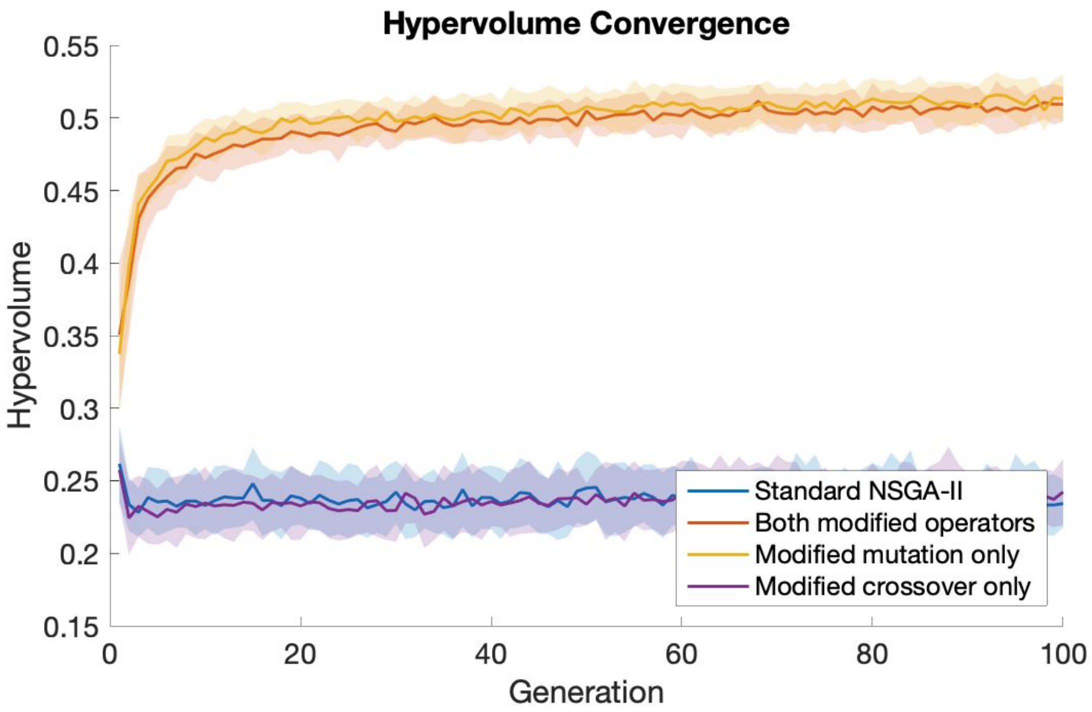


Results: Comparison





Results: Hypervolume Convergence



Discussion

Observations:

- Standard operators produced lacking diversity
- Strong bias towards expensive layouts
- Uniform random mutation is responsible
- Single-point crossover works fine
- A modified mutation operator can restore diversity

Discussion

Why does uniform random mutation introduce bias?

Consider our case study scenario:

- **12** workstations
- On each location, we allocate **0** (nothing), or **1 - 12** (a workstation)
- Uniform random mutation assigns a random integer within bounds
- **1/13** chance to allocate nothing, **12/13** to allocate a workstation

→ Mutation inflates the shopfloor with workstations

Discussion

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

Our modified mutation operator randomizes in two steps:

- First, it decides if a workstation is placed or not
- **1/2** chance to allocate nothing, **1/2** chance to allocate a workstation

- Then, if a workstation should be placed:
- **1/12** chance for any specific workstation

Increased Complexity

We can add several new influences into the simulation:

- Stochastic workstation failures
- Stochastic processing times
- Transporter collisions
- Limited port capacities
- Dynamic order prioritization (rush orders)
- Dynamic product sharing between orders

Surrogate Functions

Function evaluations for flow time and idle time are costly.
Surrogate functions can be used to estimate layout performance.

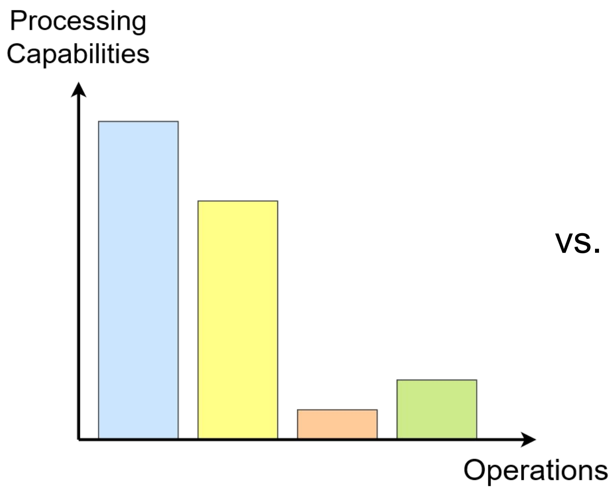
How to construct the surrogate functions?

- Black Box
- Expert Knowledge

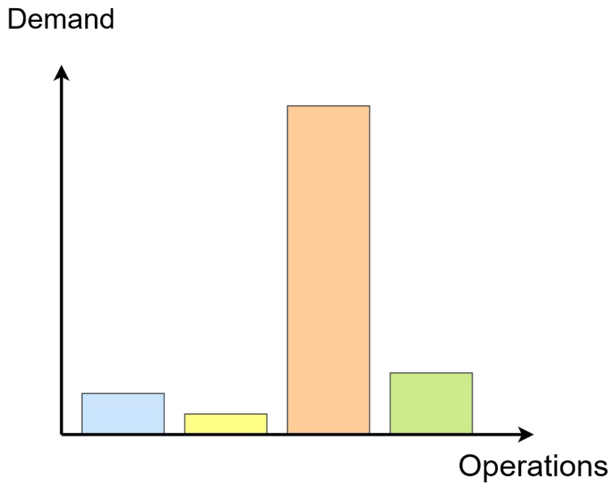
Surrogate Functions

An expert knowledge approach based on 3 measures proved successful:

- Layout imbalance (degree of bottle necking)
- Flow time transporter influence
- Idle time transporter influence



vs.



Surrogate Functions

An expert knowledge approach based on 3 measures proved successful:

- Layout imbalance (degree of bottle necking)
- Flow time transporter influence
- Idle time transporter influence

No collisions → More transporters are always better

Surrogate Functions

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

An expert knowledge approach based on 3 measures proved successful:

- Layout imbalance (degree of bottle necking)
- Flow time transporter influence
- Idle time transporter influence

Assumption: ideal number of transporters depends upon number of machines

Cache previous best combinations for idle time…
→ How far is current layout away from previous best?

Surrogate Functions

- **Flow time** surrogate =
  scale × (layout imbalance + weight × flow time transporter influence)

- **Idle time** surrogate:
  scale × (layout imbalance + weight × idle time transporter influence)

- **Re-layouting** needs no surrogate (cheap to compute)

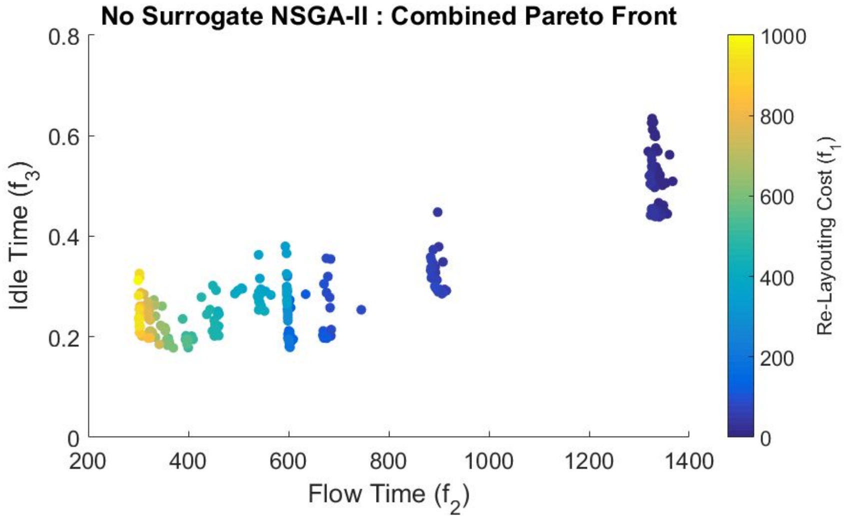
Surrogate Functions

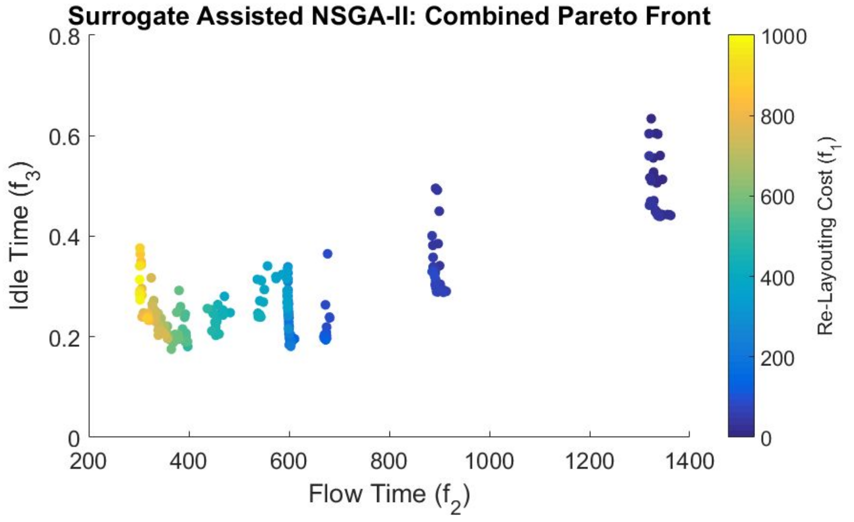
Constructing a surrogate assisted NSGA-II:

- First 500 solutions are fully evaluated → initial training set
- Scales and weights for the surrogate functions are set to fit the training set
- New solutions are only evaluated if their estimates are good
- Every fully evaluated solution adds to the training set

Surrogate Functions

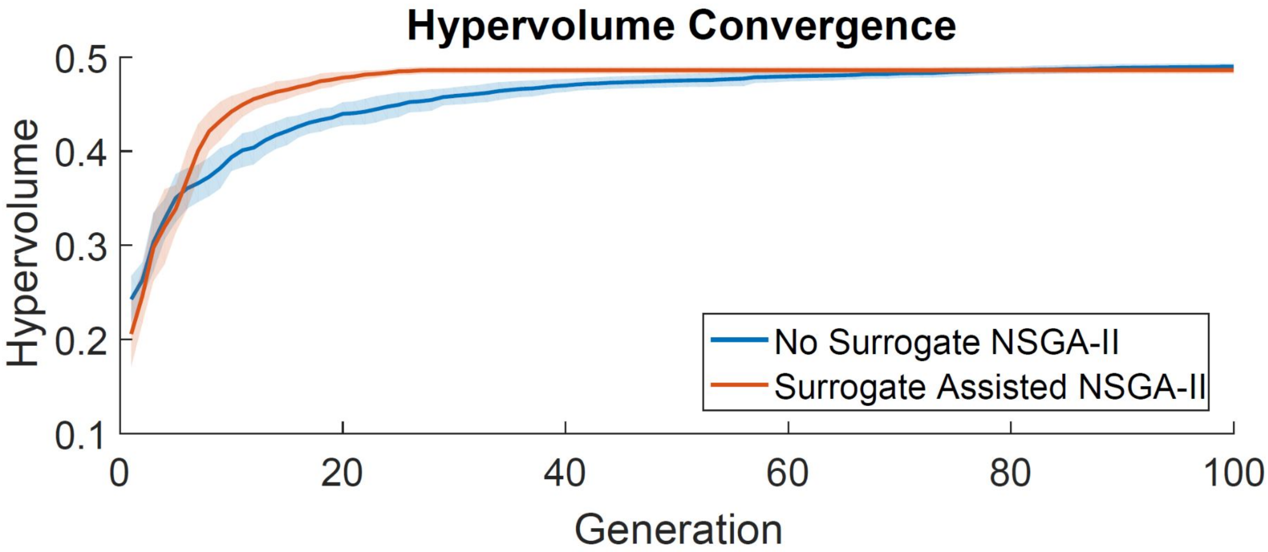
Results:





Surrogate Functions

Results:



I. Introduction

II. The integrated FLP and JSP Problem

III. Solving the Multi-Objective Optimization Problem

IV. Conclusion

Conclusion

We covered:

- ❖ Fundamentals of FLP and JSP
- ❖ History and current challenges of manufacturing
- ❖ Advanced variants of FLP and JSP
- ❖ The formalization of an integrated FLP and JSP
- ❖ Modeling modern manufacturing environments
- ❖ Using MOEA's to solve the problem
- ❖ Designing custom operators
- ❖ Designing expert knowledge surrogate functions

Conclusion

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

We learned:

- FLP's for modern manufacturing systems require new methods
- Uniform random mutation is not suited for this problem / encoding
- Introducing a new mutation without bias solves the problem
- Surrogates can be effective for this problem
- There is still a lot more to be done…

References

Ripon, K. S. N., Glette, K., Hovin, M., & Torresen, J. (2012). **A multi-objective evolutionary algorithm for solving integrated scheduling and layout planning problems in manufacturing systems.** *2012 IEEE Conference on Evolving and Adaptive Intelligent Systems, Madrid, Spain*, 157–163. https://doi.org/10.1109/EAIS.2012.6232822

Mascis, A., & Pacciarelli, D. (2002). **Job-shop scheduling with blocking and no-wait constraints.** *European Journal of Operational Research*, *143*(3), 498–517. https://doi.org/10.1016/S0377-2217(01)00338-1

Azevedo, M. M., Crispim, J. A., & Pinho de Sousa, J. (2017). **A dynamic multi-objective approach for the reconfigurable multi-facility layout problem.** In *Journal of Manufacturing Systems* (Vol. 42, pp. 140–152). Elsevier B.V. https://doi.org/10.1016/j.jmsy.2016.12.008

Asadollahi-Yazdi, E., Couzon, P., Nguyen, N. Q., Ouazene, Y., & Yalaoui, F. (2020). **Industry 4.0: Revolution or Evolution?** *American Journal of Operations Research*, *10*(06), 241–268. https://doi.org/10.4236/ajor.2020.106014

Maganha, I., Silva, C., & Ferreira, L. M. D. F. (2019). **The layout design in reconfigurable manufacturing systems: a literature review.** *International Journal of Advanced Manufacturing Technology*, *105*(1–4), 683–700. https://doi.org/10.1007/s00170-019-04190-3

References

Chiang, W. C., Kouvelis, P., & Urban, T. L. (2006). **Single- and multi-objective facility layout with workflow interference considerations.** *European Journal of Operational Research*, *174*(3), 1414–1426. https://doi.org/10.1016/j.ejor.2005.03.007

Kovács, G., & Kot, S. (2017). **Facility layout redesign for efficiency improvement and cost reduction.** *Journal of Applied Mathematics and Computational Mechanics*, *16*(1), 63–74. https://doi.org/10.17512/jamcm.2017.1.06

Wen, Y.-W., & Ting, C.-K. (2018). **Designing Facility Layouts with Hard and Soft Constraints by Evolutionary Algorithm.** *2018 First International Conference on Artificial Intelligence for Industries (AI4I), Laguna Hills, CA, USA*, 44–47. https://doi.org/10.1109/AI4I.2018.8665682

Tayal, A., & Singh, S. P. (2019). **Formulating multi-objective stochastic dynamic facility layout problem for disaster relief.** *Annals of Operations Research*, *283*(1–2), 837–863. https://doi.org/10.1007/s10479-017-2592-2

Vitayasak, S., & Pongcharoen, P. (2015). **Re-Layout and Robust Machine Layout Design under Stochastic Demand.** *Applied Mechanics and Materials*, *789–790*, 1252–1257. https://doi.org/10.4028/www.scientific.net/amm.789-790.1252

References

Sagawa, J. K., & Nagano, M. S. (2015). **Modeling the dynamics of a multi-product manufacturing system: A real case application.** *European Journal of Operational Research*, *244*(2), 624–636. https://doi.org/10.1016/j.ejor.2015.01.017

Zhang, Z., Guan, Z., Gong, Y., Luo, D., & Yue, L. (2022). **Improved multi-fidelity simulation-based optimisation: application in a digital twin shop floor.** *International Journal of Production Research*, *60*(3), 1016–1035. https://doi.org/10.1080/00207543.2020.1849846

Komaki, G. M., Sheikh, S., & Malakooti, B. (2019). **Flow shop scheduling problems with assembly operations: a review and new trends.** *International Journal of Production Research*, *57*(10), 2926–2955. https://doi.org/10.1080/00207543.2018.1550269

Teymourifar, A., Ozturk, G., Ozturk, Z. K., & Bahadir, O. (2020). **Extracting New Dispatching Rules for Multi-objective Dynamic Flexible Job Shop Scheduling with Limited Buffer Spaces.** *Cognitive Computation*, *12*(1), 195–205. https://doi.org/10.1007/s12559-018-9595-4

Kundakci, N., & Kulak, O. (2016). **Hybrid genetic algorithms for minimizing makespan in dynamic job shop scheduling problem.** *Computers & Industrial Engineering*, *96*, 31–51. https://doi.org/10.1016/j.cie.2016.03.011

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

Questions?