Scheduling Heuristics in Practice -
Flexible Flow Shops and Flow Shops with Reentry

IFAC 2022 – Nantes, France -- June, 2022

Michael Pinedo

In collaboration with:
Juntaek Hong, Kangbok Lee, Kwansoo Lee,
Kyunduk Moon, and Tae-Sun Yu

Variations of Flow Shop Scheduling

Part I

Scheduling in
Continuous Cast Steelmaking

Flexible flow shops

Part II

Scheduling in
Semiconductor Manufacturing

Flow Shops with Reentry

Part I

Scheduling in Continuous Cast Steelmaking

(joint work with **Juntaek Hong, Kangbok Lee, Kwansoo Lee, Kyunduk Moon**)

- Steelmaking Problem description
- Solution method: Iterated Greedy Matheuristic
- Experimental results
- Conclusion

Introduction

Pressure on Steelmaking Industry against Facility Expansions

REUTERS July 13, 2019:

COMMODITIES NEWS

JULY 13, 2019 / 2:49 PM / UPDATED 2 YEARS AGO

China plans to toughen emission checks on steel mills

BEIJING (Reuters) - China will continue to enforce production restrictions in heavy industry in winter this year and will tighten its emission assessment on steel mills when granting exemptions from curbs already in place, an environment ministry official said

Bloomberg Green March 12, 2021:

Green

China Pollution Crackdown Exposes Rule Breakers in Top Steel Hub

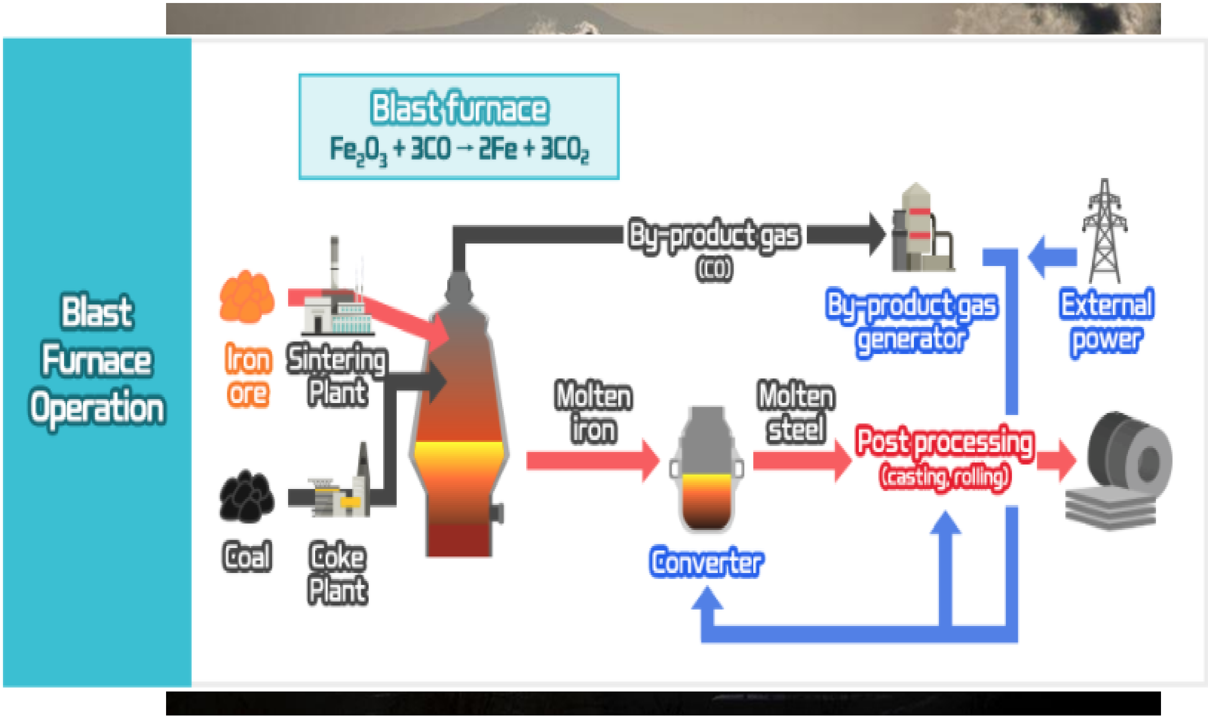
China's top environmental official vowed to reinforce pollution curbs after inspections found some steel mills were violating output restrictions and faking documents.

A team led by Huang Runqiu, the minister of ecology and environment, on Thursday found four mills in the steelmaking hub of Tangshan weren't complying with production cuts put in place to reduce heavy pollution.

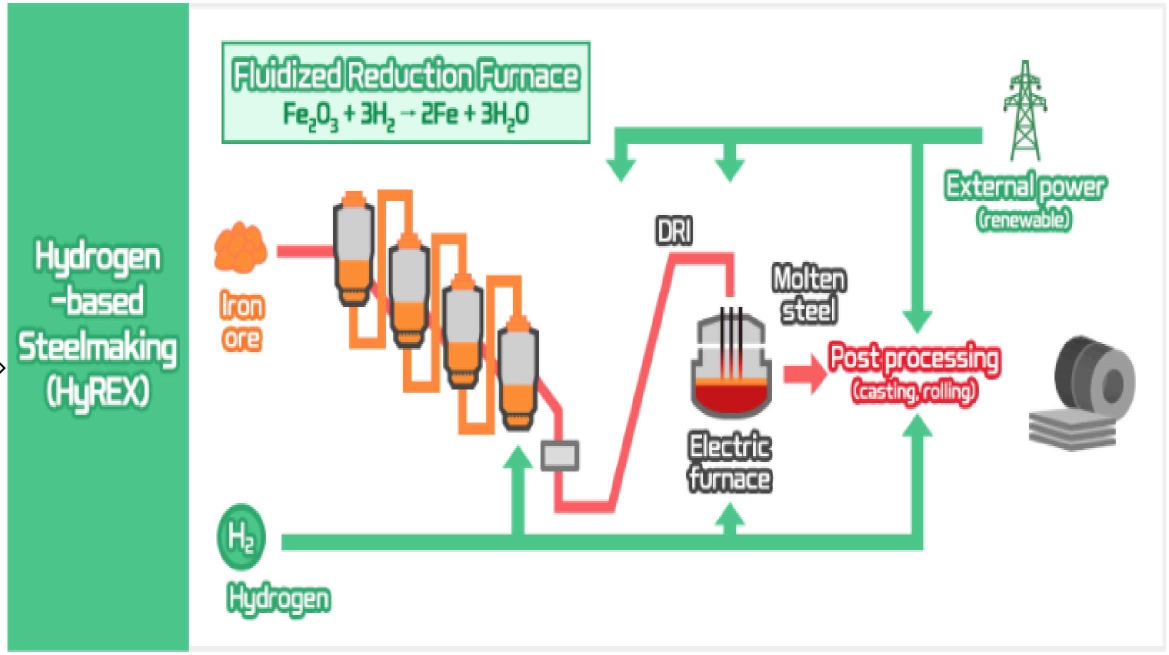
→ **Expansion of conventional facility is impossible**

Introduction

Pressure on Steelmaking Industry against Facility Expansion



Conventional facilities will be used for 30 years

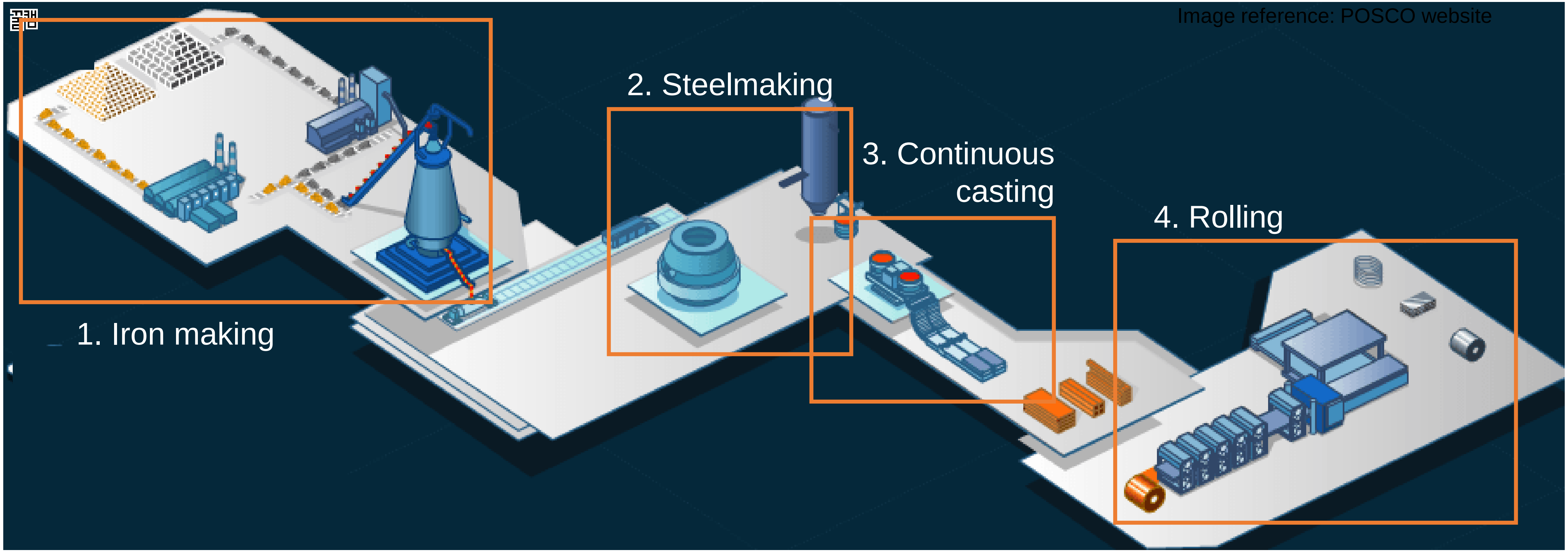


A steelmaker's plan: commercialize by 2050

→ **Efficient operations of existing facilities are still crucial**

Introduction

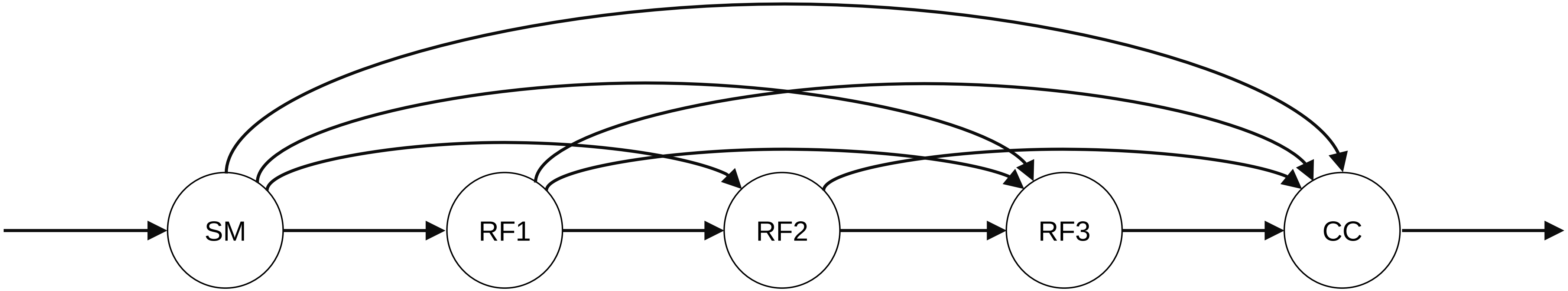
Steel Production



→ **Steelmaking – Continuous Casting (SCC) process is typically the bottleneck**

Problem Description

SCC Process



Charge: a pot of molten steel

Semi-finished products:

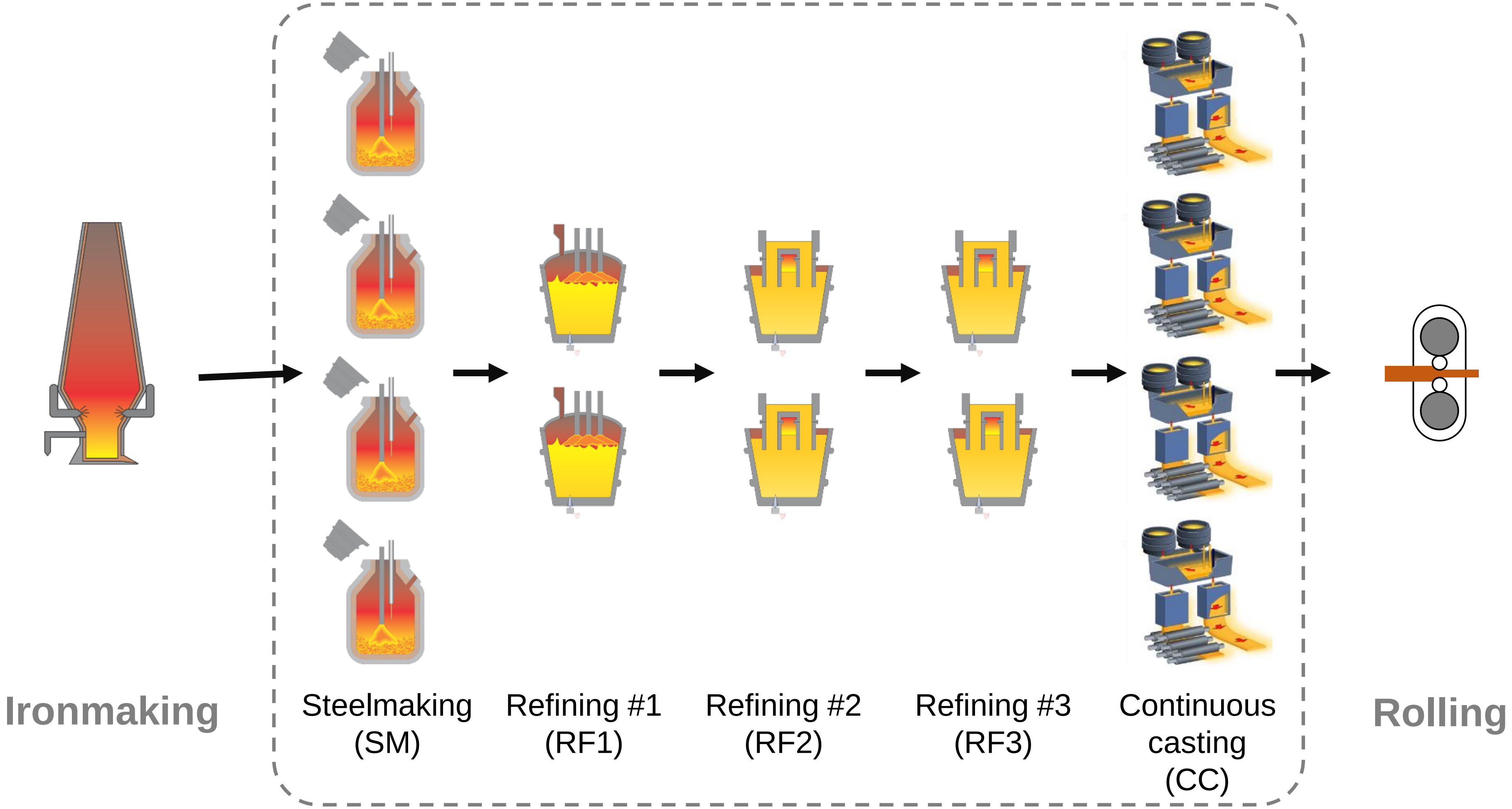
Slab

Bloom

Billet

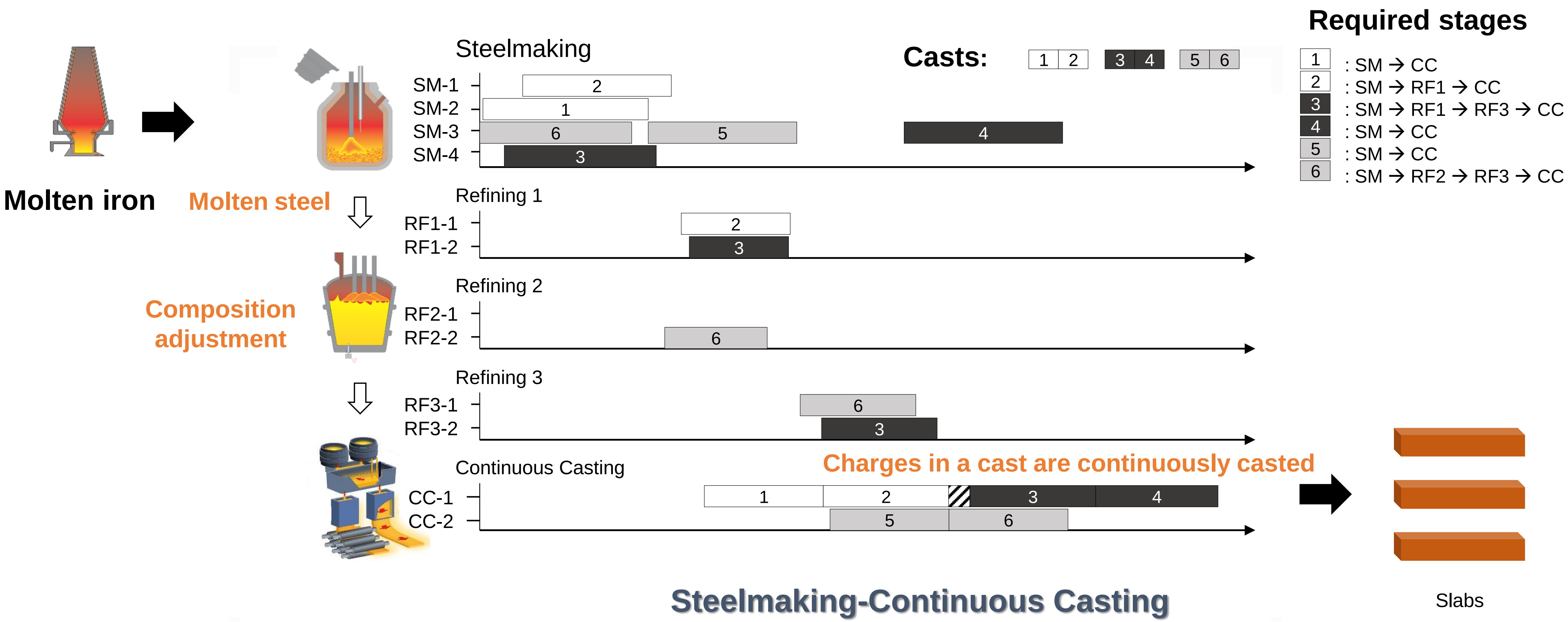
Problem Description

SCC Process



Problem Description

SCC Process



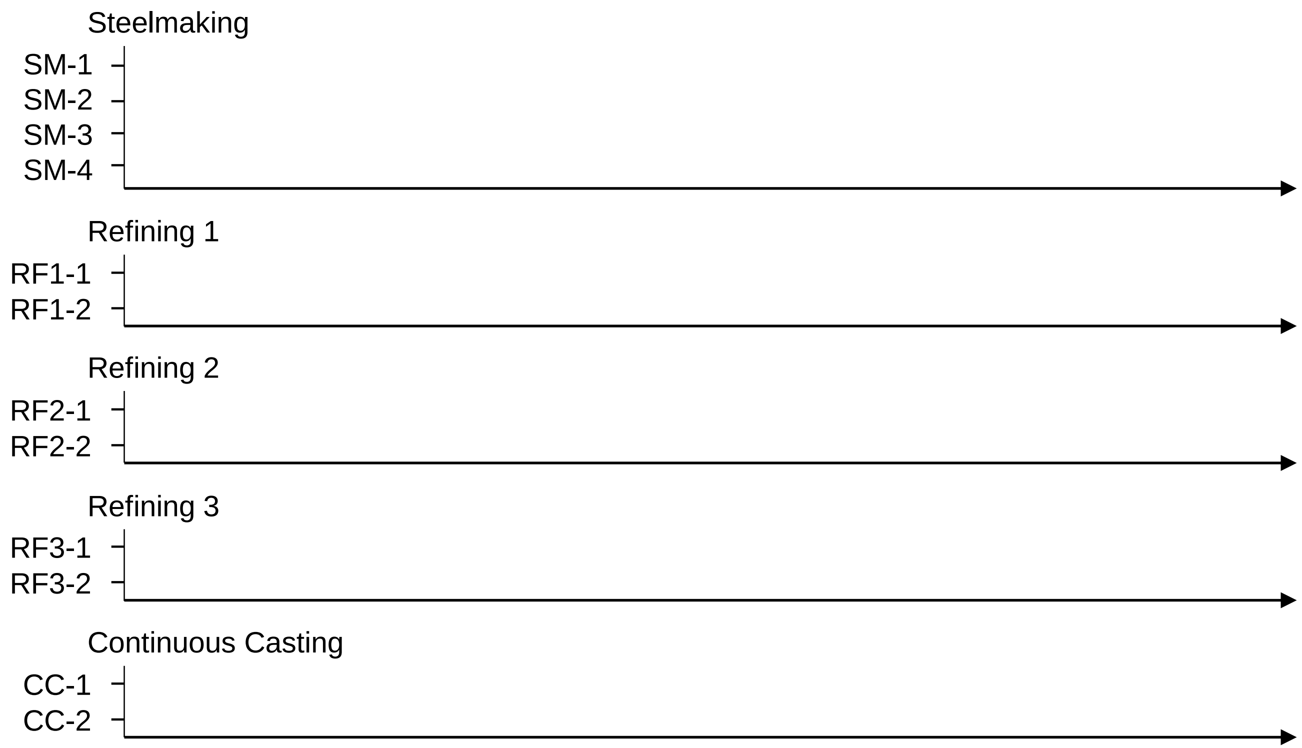
Problem Description

SCC Scheduling Problem

- **Parameters**
  - SCC environment
    - Stages, machines, transportation time
  - Charge
    - Required refining stages
    - Processing time on each machine
    - Due date (at the last stage)
  - Cast: a sequence of charges at the last stage (processed one after another).
    - Setup time at the last stage before processing the first charge
- **Variables**
  - Machine assignment of each charge at each stage
  - Completion time of each charge at each stage

Problem Description

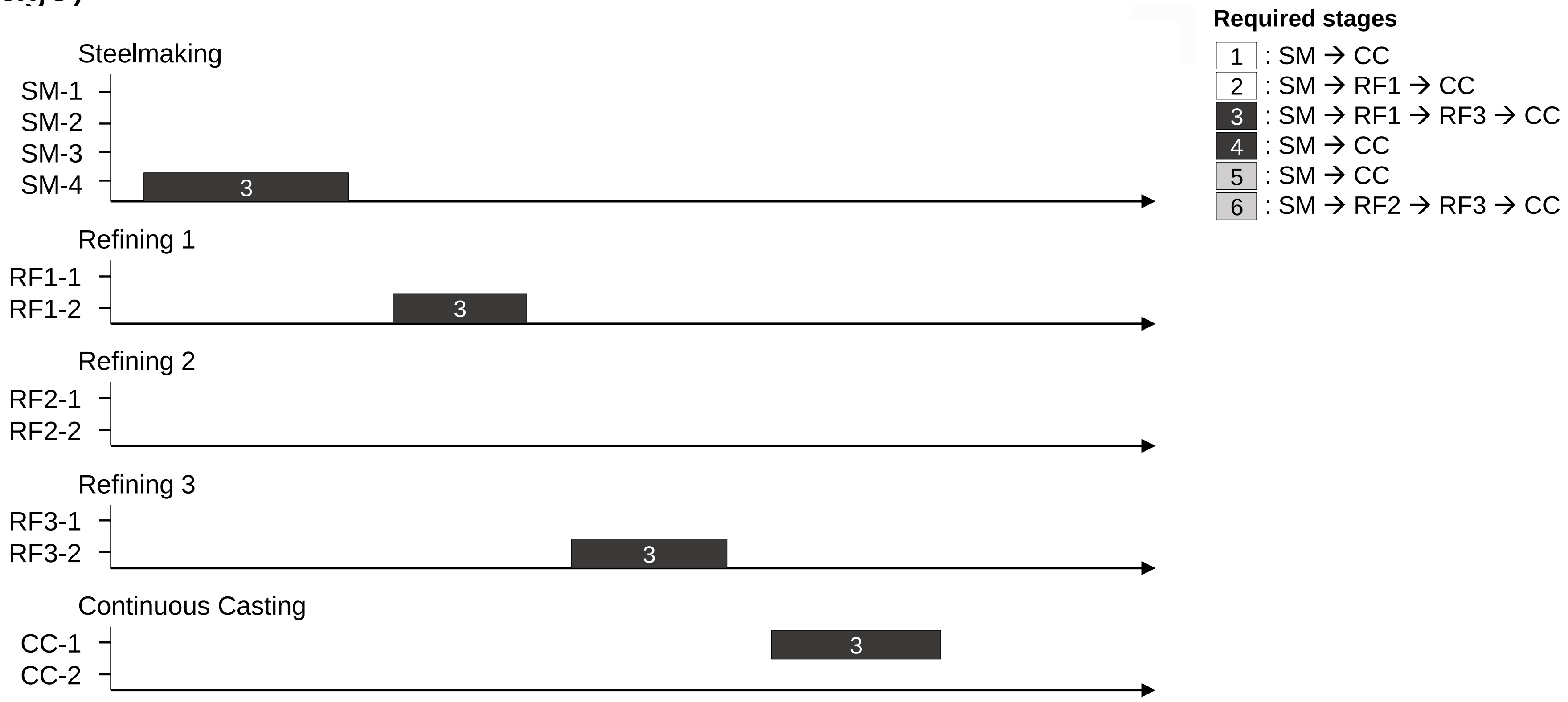
- **SCC environment**
  - Stages & machines
  - Transportation time between each pair of machines



Problem Description

▪ Charge

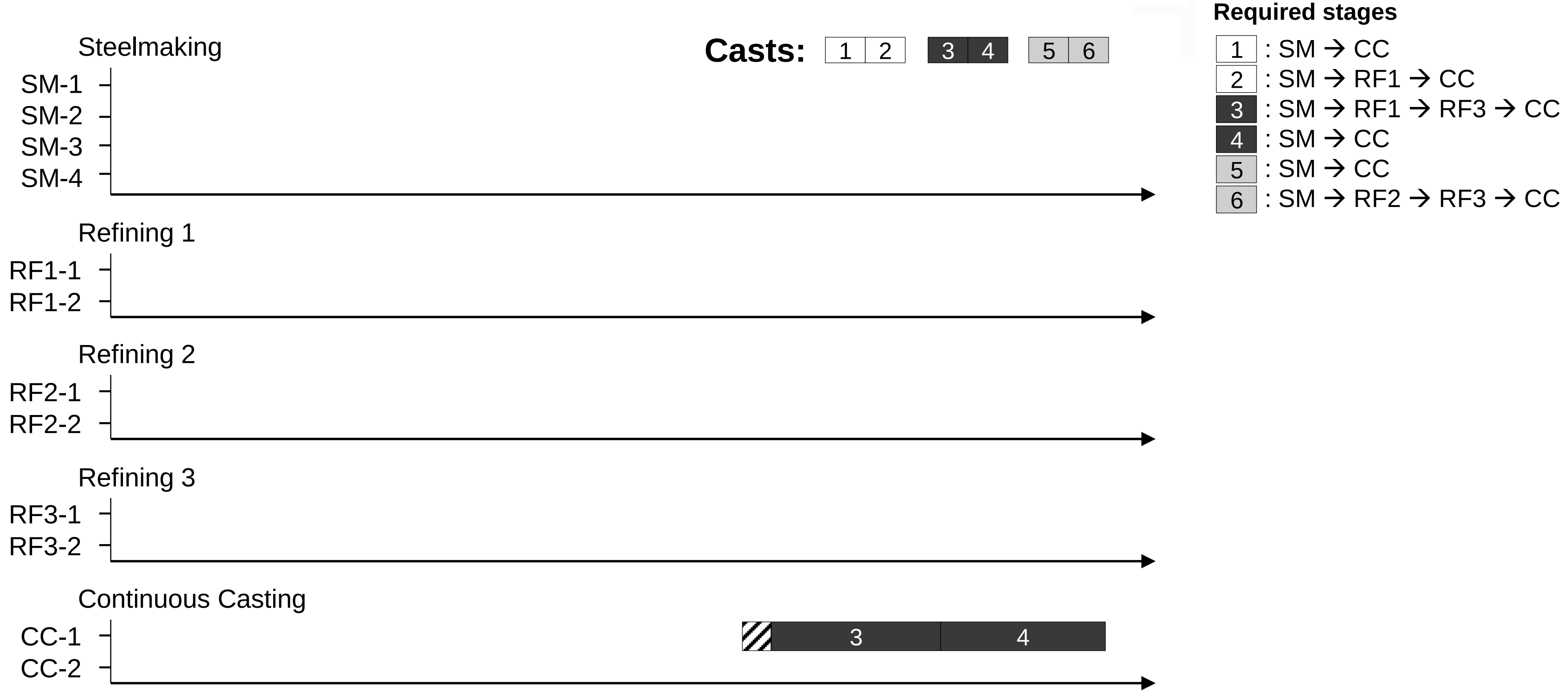
- Required refining stages & Processing time on each machine
- Due date (at the last stage)



Problem Description

SCC Scheduling Problem: Parameters

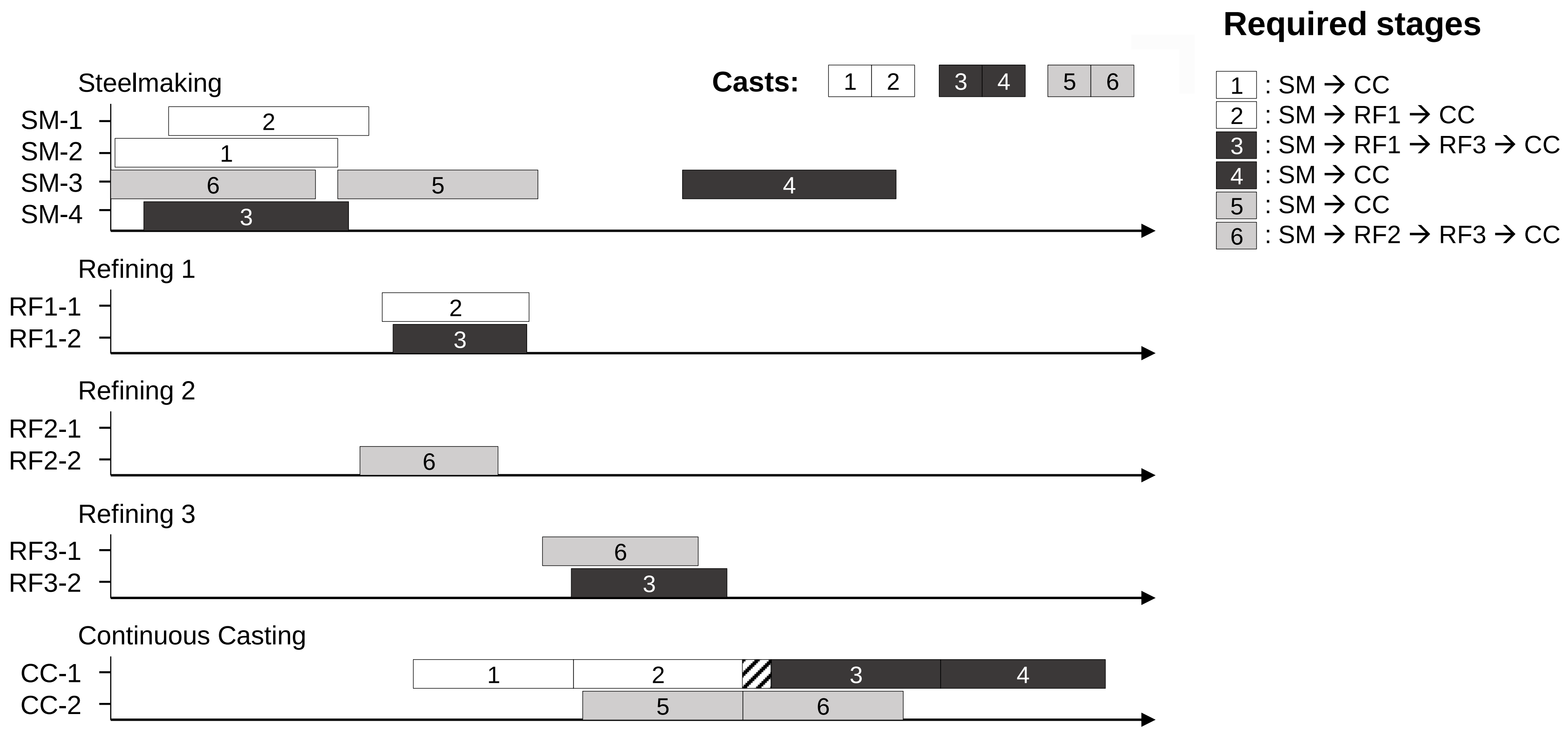
- **Cast**
  - A sequence of charges
  - Setup time at the last stage before processing the first charge



Problem Description

SCC Scheduling Problem: Variables

- Machine assignment of each charge at each stage
- Completion time of each charge at each stage



Problem Description

SCC Scheduling Problem: obj. & Constr.

- **Parameters**
  - SCC environment
    - Stages, machines, transportation time
  - Charge
    - Required refining stages
    - Processing time on each machine
    - Due date (at the last stage)
  - Cast: a sequence of charges & setup time
- **Variables**
  - Machine assignment of each charge at each stage
  - Completion time of each charge at each stage

- **Objective: to minimize**
  - **Cast breaks**
  - **Total waiting time (between stages)**
  - **Total earliness**
  - **Total tardiness**

Imagine how the objective function of a MIP would look like !!

- **Constraints**
  - At most one charge at a time in each machine
  - CC stage
    - One CC machine for all charges in a cast
  - Maximum waiting time (between stages)``

Related Literature (quite extensive)

- **Machine Environments**
  - Flexible (Hybrid) Flow Shops (e.g., Ruiz and Vazquez-Rodriguez (2010))
  - Steel Making Continuous Cast (e.g., Tang, Liu, Rong, and Yang (2002))

- **Heuristic Procedures**
  - Genetic Algorithms (e.g., Deb et al. (2002))
  - Iterated Greedy (IG) Heuristics (e.g., Ruiz and Stuetzle (2007))
  - Constraint Guided Heuristic Search (e.g., Gay, Schaus, De Smedt (2014))
  - Matheuristics (e.g., Boschetti and Maniezzo (2022))
  - Hybrid Heuristics …….

Literature on SCC Scheduling (2002-2021)

Very important problem!!

Author (year)	Problem type*	Ca-CC fix	Objectives: E&T†	Objectives: Completion time‡	Objectives: Waiting time‡	Constraints: Max waiting time	Constraints: Diff. Ch routes	Constraints: MC uniformity††	Constraints: Controllable time‡‡	Data: # RF stages	Data: Max charges	Method: Algorithm	Method: Time limit (sec)
Tang et al. (2002)	I		Ch		W			P		1	12	LR	222
Pacciarelli and Pranzo (2004)	I			M		O		P		3	114	Heu	324
Bellabdaoui and Teghem (2006)	I	O		M		O		P	C	1	8	MIP	6
Xuan and Tang (2007)	I			W	W			P		1	12	LR	623
Atighehchian, Bijari, and Tarkesh (2009)	I			M	S	O		R		1	108	ACO+NLP	300
Pan et al. (2013)	I	O	Ca		S			P		1	120	ABC	30
Sun and Wang (2013)	I		Ca		S	O	O	R		4	7	Heu	-
Tang, Zhao, and Liu (2014)	R			M	S	O	O	P	A	3	100	DE	60
Mao et al. (2014a)	R	O		S	S		O	P	A	2	120	LR	116
Mao et al. (2014b)	I	O	Ca		S			P		3	40	LR	176
Li et al. (2014)	I	O	Ca		S			P		3	120	FOA	20
Sbihi, Bellabdaoui, and Teghem (2014)	I	O		S		O		R	C	3	49	MIP	∞
Mao et al. (2015)	I	O		S	W			P		2	120	LR	54
Hao et al. (2015)	I	O			W		O	P		1	900	PSO	150
Jiang et al. (2015)	I	O	Ca		S		O	P	C	2	100	DE+VNS	400
Li, Pan, and Mao (2016)	R	O	Ca		S			P	A	1	120	FOA+IG	100
Pan (2016)	I			M	S			P		4	180	ABC	54
Long et al. (2016)	I	O	Ch		S		O	P		2	-	GA+LP	400
Jiang et al. (2016)	I	O		S	S		O	P	C	2	150	Heu	30
Yu, Chai, and Tang (2016)	R	O			S		O	P	A	1	30	Heu	-
Cui and Luo (2017)	I	O	Ca		W			P		2	20	LR	60
Jiang, Liu, and Hao (2017)	I	O	Ca		S		O	P		2	120	GA+LS	600
Long, Zheng, and Gao (2017)	R	O	Ch		W		O	P	A	2	66	GA+VNS	250
Sun et al. (2017)	R	O	Ch		W			P	A	2	40	LR	135
Fazel Zarandi and Dorry (2018)	I	O		M	S	O		P		1	61	PSO+LP	300
Jiang, Zheng, and Liu (2018)	I				S	O		P		1	150	CRO	330
Li et al. (2018)	I			M				P		1	120	ABC	100
Long et al. (2018a)	I				S		O	P	A	5	104	GA	-
Long et al. (2018b)	I	O		M	S		O	P	A	5	140	GA	450
Peng et al. (2018)	R	O	Ca		S			P	A	1	240	ABC	10
Sbihi and Chemangui (2018)	I	O		M		O		R	C	1	49	GA+LP	1800
Cui, Luo, and Wang (2020)	I	O	Ca		W			P		1	45	LR	150
Peng et al. (2020)	R	O	Ca		S			P		1	120	ICA+LS	30
Han et al. (2021)	I			W	W		O	P		3	62	LR	1200
This paper (2021)	I		Ch		S	O	O	R		3	36	IG+MIP	600

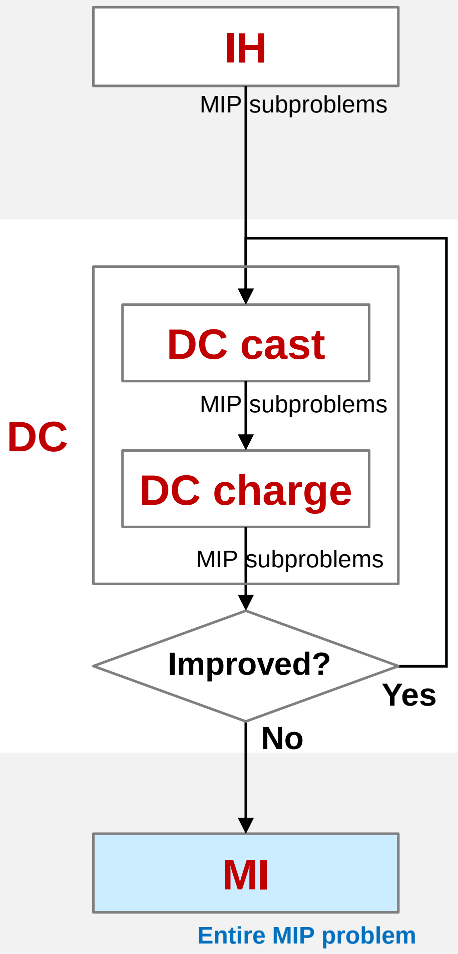
Problem Description

Contribution to the Literature: **An efficient method useful in practice**

- **Parameters**
  - SCC environment
    - Stages, machines, transportation time
  - Charges
    - Required refining stages
    - Processing time on each machine
    - Due date (at the last stage)
  - Cast: a sequence of charges & setup time
- **Variables**
  - Machine assignment of each charge at each stage
  - Completion time of each charge at each stage

- **Objective**: to minimize
  - Cast breaks
  - Total waiting time (between stages)
  - Total earliness
  - Total tardiness
- **Constraints**
  - At most one charge at a time in each machine
  - Continuous Casting (CC) stage
    - One CC machine for all charges in a cast
  - Maximum waiting time (between stages)

Iterated Greedy Matheuristic Overview



Initial Heuristic (IH)
for a good initial schedule

- We put one cast at a time
- while preserving the former schedule
  – machine assignment
  – precedence relationship

Destruction & Construction (DC)
for a better schedule

- We rearrange some casts and some charges by destruction & construction
- In construction procedure, we preserve the current destructed schedule
  – machine assignment
  – precedence relationship
- How to choose charges
  – DC cast : charges in a cast
  – DC charge : charges in a time window

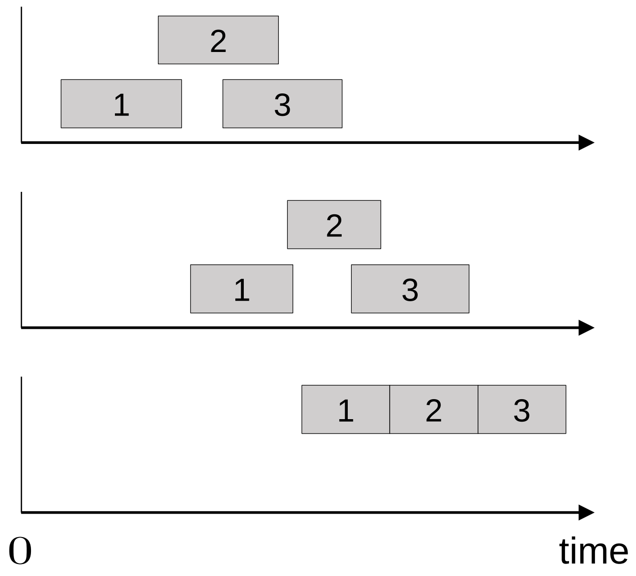
MIP Improvement (MI)
for an improve schedule (possibly optimal)

- We solve the entire MIP problem given an incumbent solution

Iterated Greedy Matheuristic

Initial Heuristic (IH)

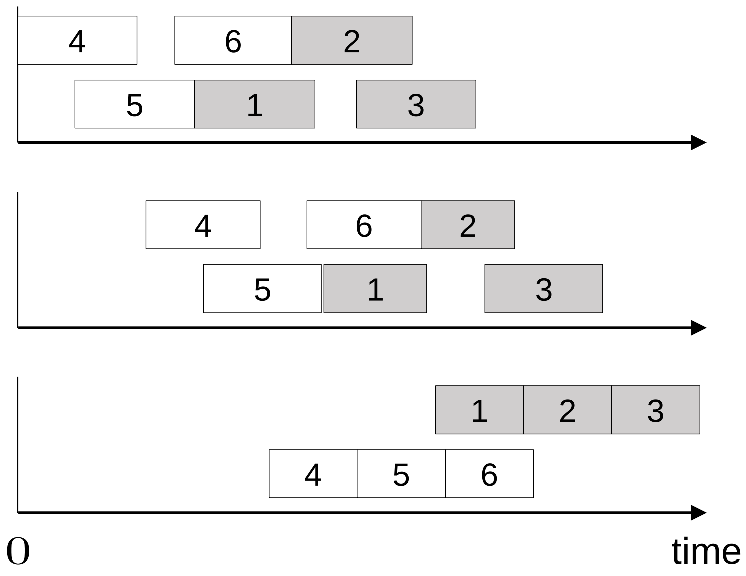
Cast sequence: 1 2 3 4 5 6 7 8 9



Iterated Greedy Matheuristic

Initial Heuristic (IH)

Cast sequence: 1 2 3 4 5 6 7 8 9

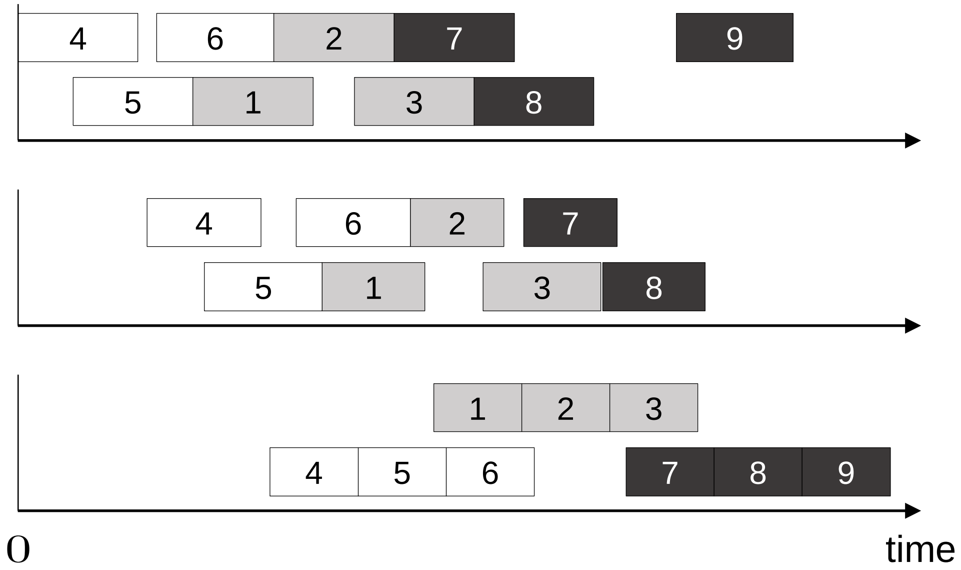


Iterated Greedy Matheuristic

Initial Heuristic (IH)

Cast sequence:

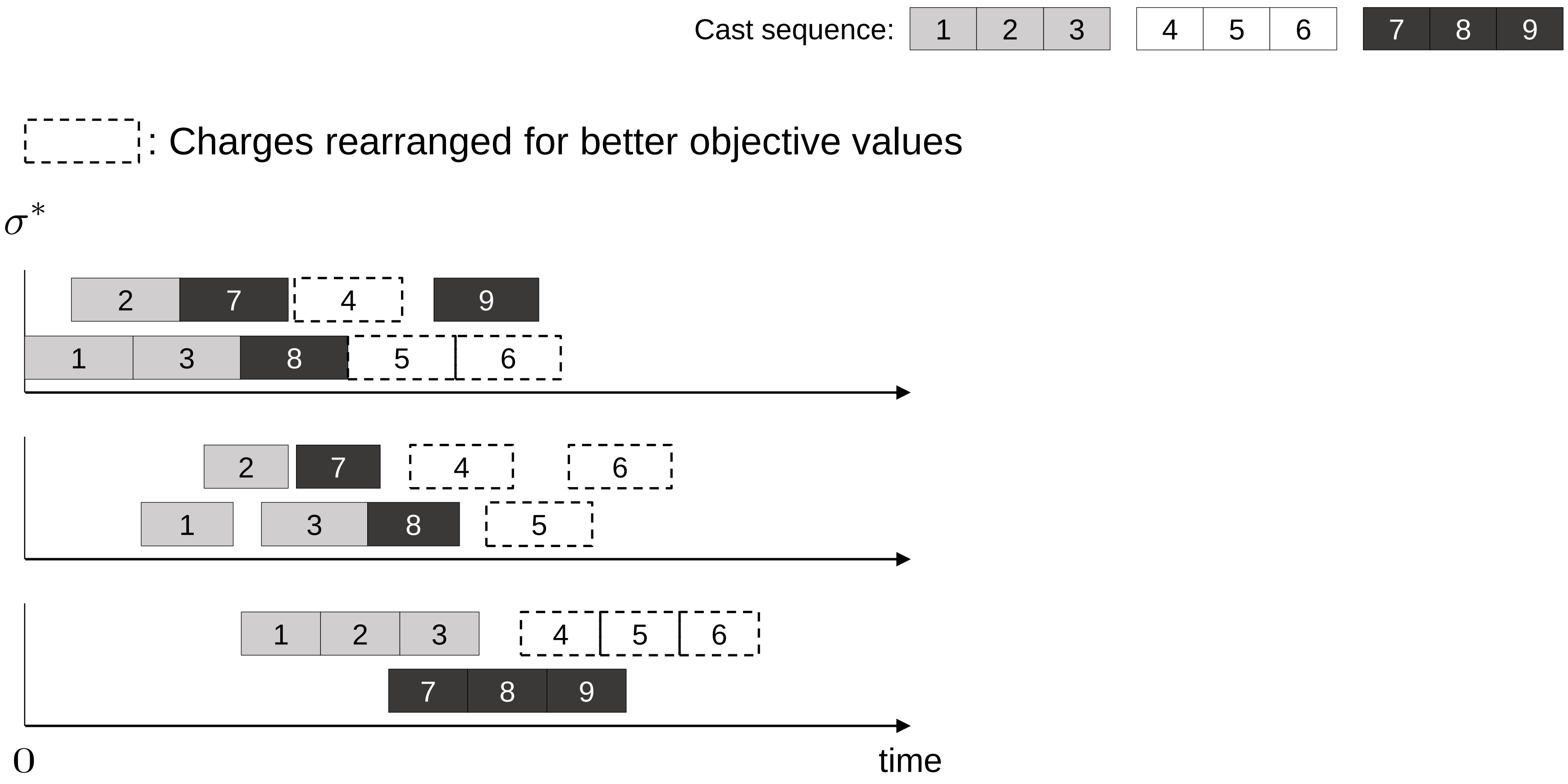
- while preserving the former schedule
  - machine assignment of charge
  - precedence relationship between charges



Iterated Greedy Matheuristic

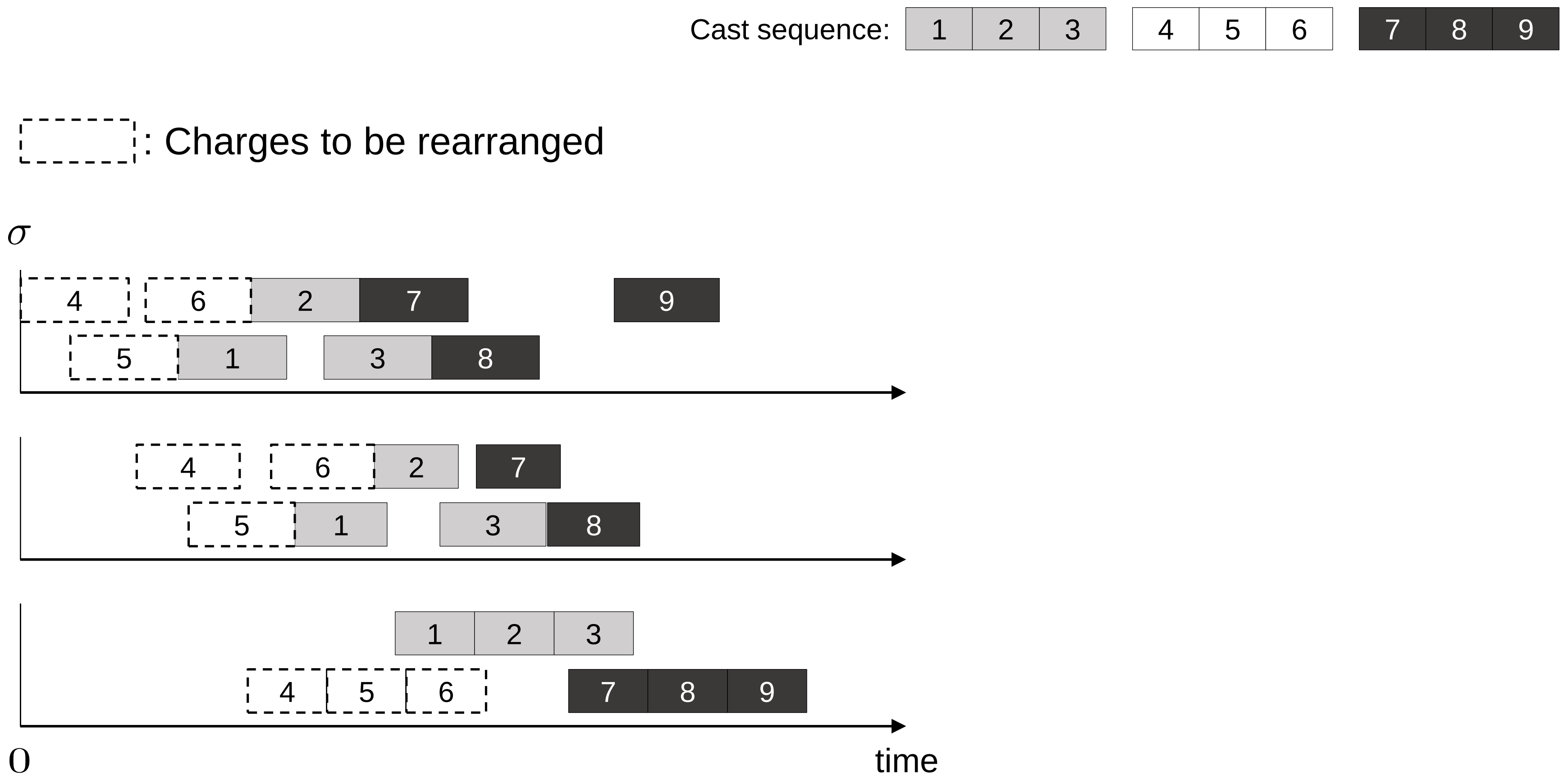
DC Cast (Dcast) Step

- while preserving the other charges' schedule with regard to
  - machine assignment of charge
  - precedence relationship between charges



Iterated Greedy Matheuristic

DC Cast (D_{cast}) Step

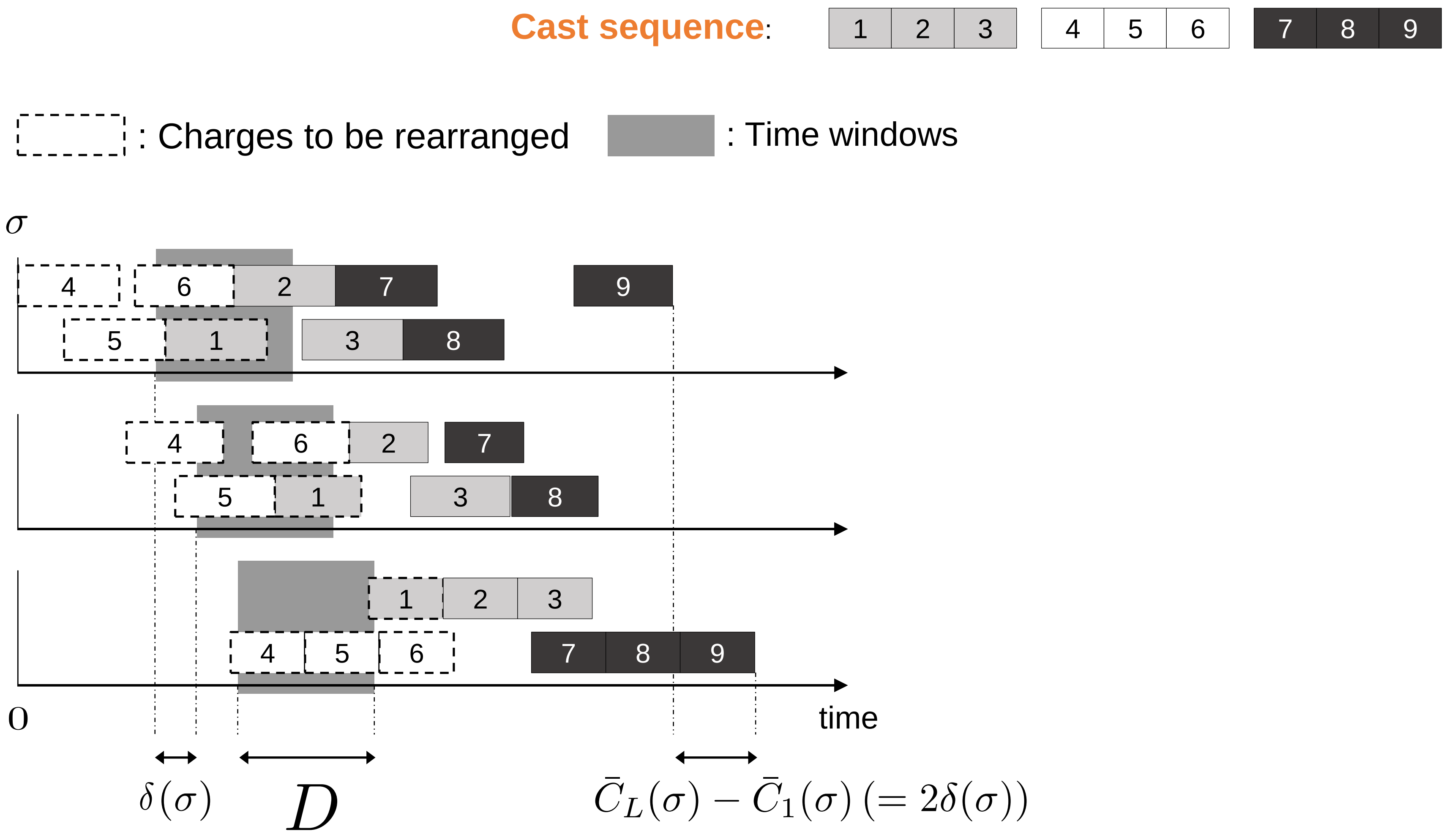


Iterated Greedy Matheuristic

DC Charge (Dcharge) Step

D is time window in which charges are being rearranged and may end up being assigned to different machines. (charges of which the completion time in at least one stage are in the time window).

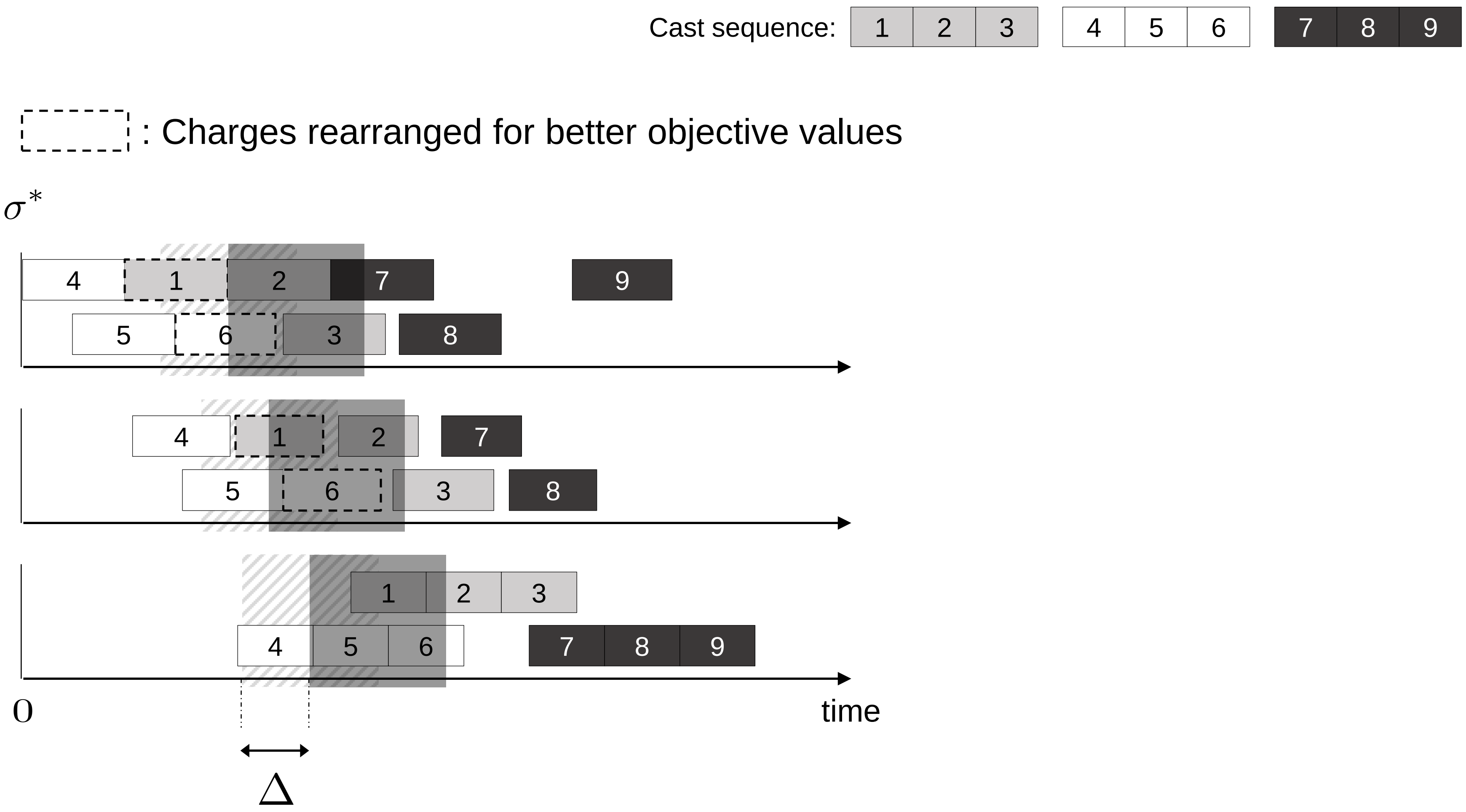
Time windows of successive stages need to be delayed (in order not to have to consider too many jobs)



Iterated Greedy Matheuristic

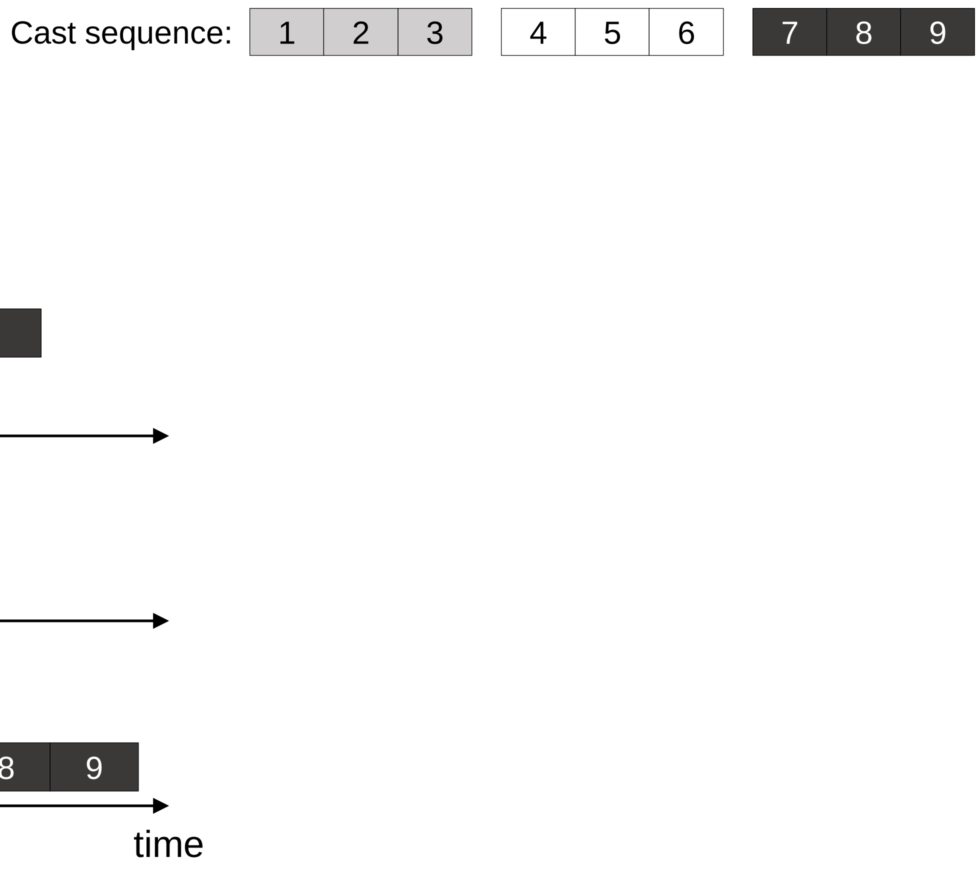
DC Charge (D_{charge}) Step

- while preserving the other charges' schedule
  - machine assignment of charge
  - precedence relationship between charges



Iterated Greedy Matheuristic

Compare IGM Schedule with Schedule Generated by Initial Heuristic (IH)



Experimental Results

Test Data Summary

- Three problem sizes
  - small: 2~3 casts, 6~12 charges
  - medium: 3~4 casts, 15~24 charges
  - practical: 4~7 casts, 30~36 charges
- Random processing times
  - SM: 45~55 min
  - RF: 30~40 min
  - CC: 35~45 min
- Transportation time: 10 minutes between all machines
- Maximum waiting time: 30 minutes

- Total 90 problem instances
  - 30 small-sized problems
  - 30 medium-sized problems
  - 30 practical-sized problems

Experimental Results

Comparison of algorithms

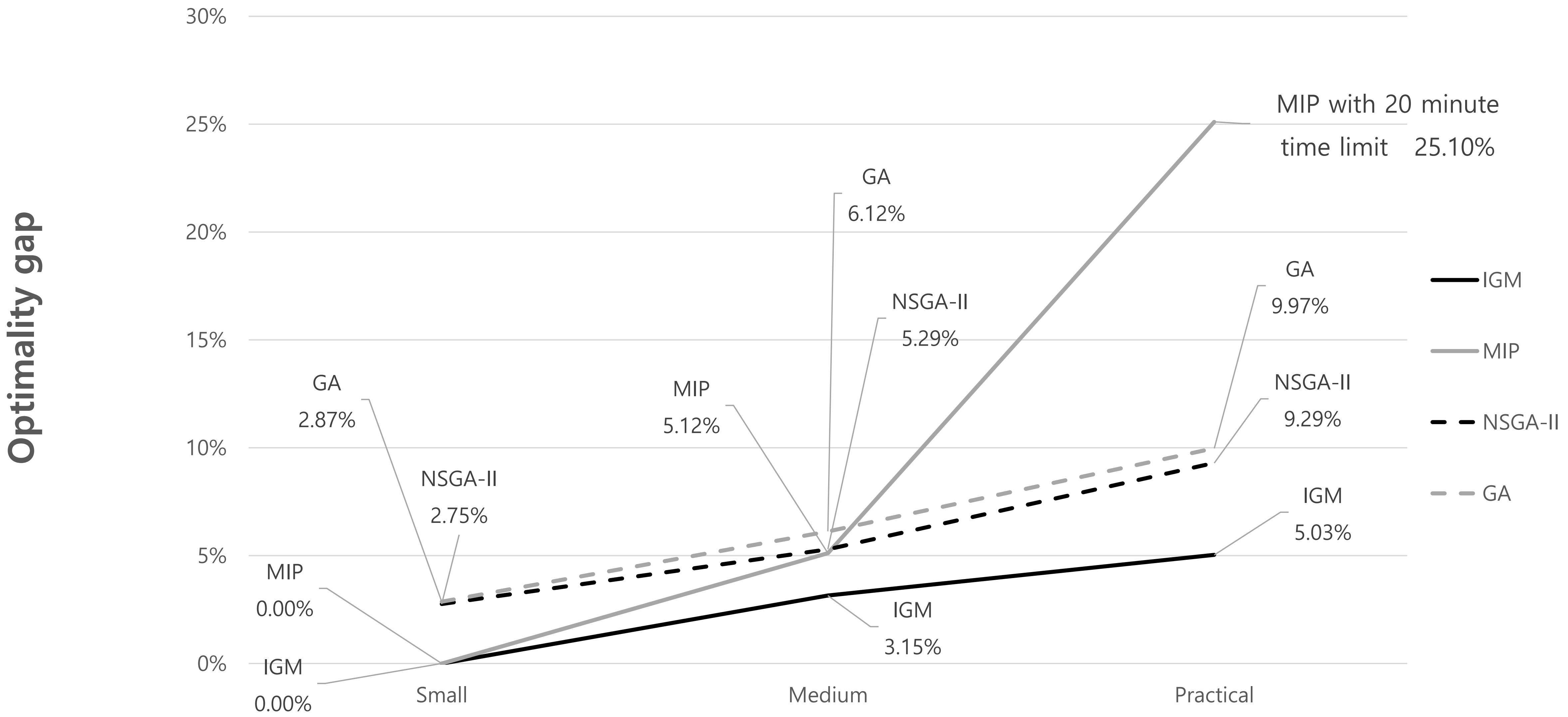
- Iterated Greedy Matheuristic (IGM) → 10 minutes
- Solving the whole MIP model (MIP)
- NSGA-II **
- Simple genetic algorithm (GA)

} → 20 minutes time limit

** Non-Dominated Sorting Genetic Algorithm –II by **Deb, Pratap, Agarwal, Meyarivan (2002)**
IEEE Transactions on Evolutionary Computation
(GA Method that is especially suitable for multi-objective optimization problems).

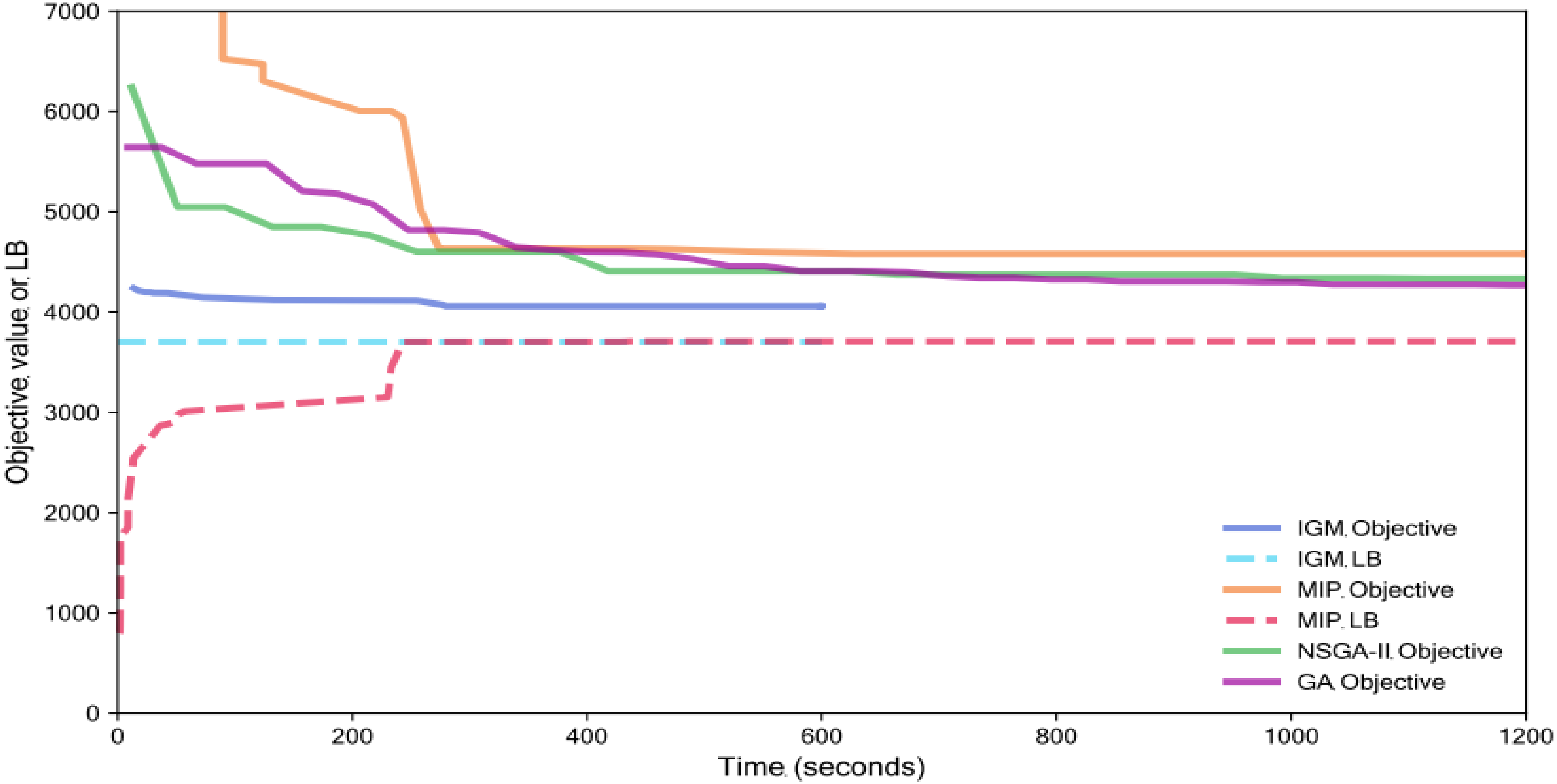
Experimental Results

The average optimality gaps



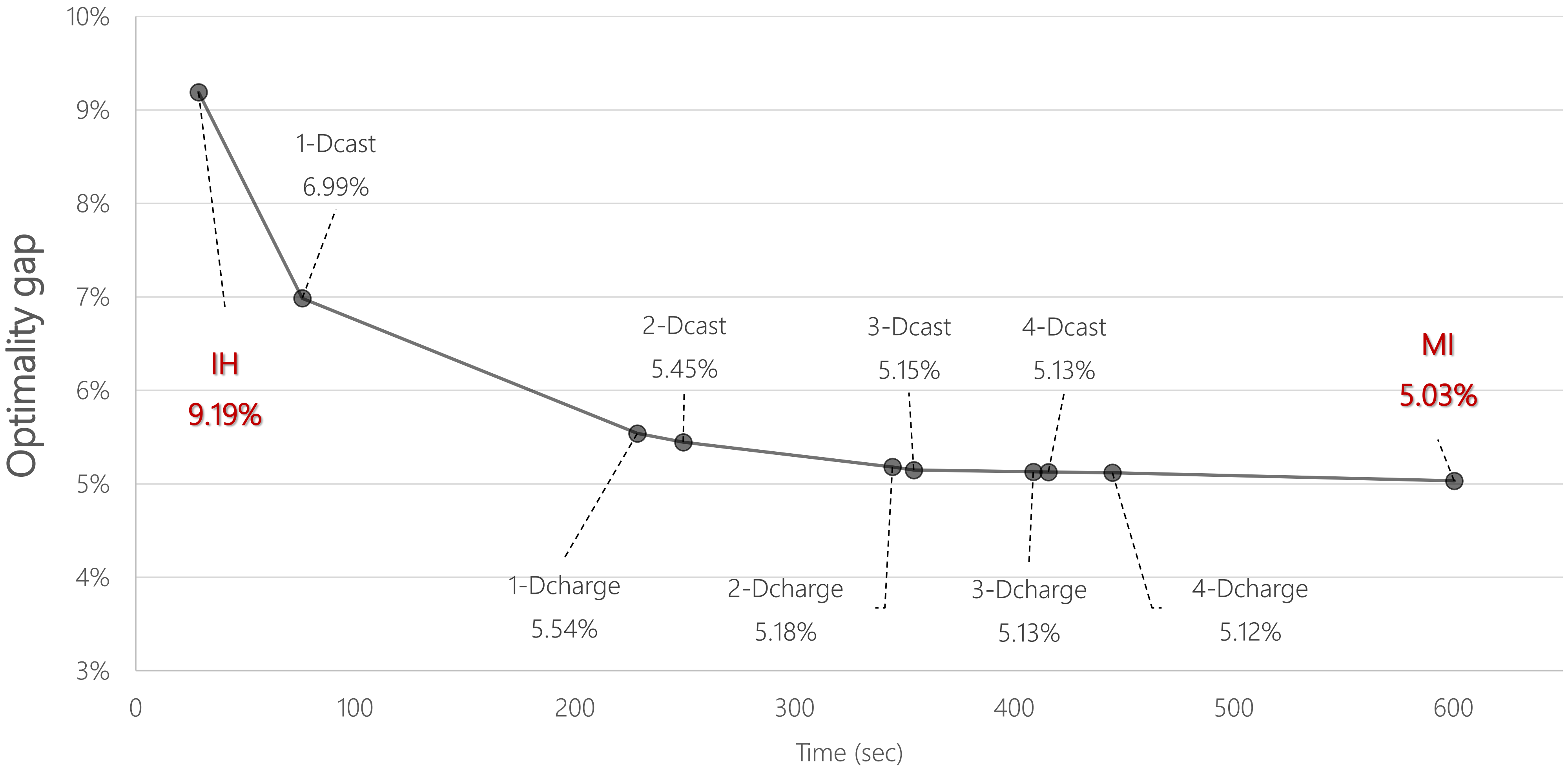
Experimental Results

Objective value over time on practical size problem #3



Experimental Results

Average performance of IGM over time on practical size problems



Conclusions Steelmaking

- IGM is effective and may also be applied to:
  - Practical hybrid flowshop scheduling problem considering:
    - sequence-dependent setup times
    - precedence constraints
    - machine eligibility constraints
  - Scheduling problems in more general machine environments (e.g., flexible job shop)

Literature: Hong, J., Moon, K., Lee, K., Lee, K., & Pinedo, M. L. (2022).
“An iterated greedy matheuristic for scheduling in steelmaking-continuous casting process”
International Journal of Production Research, 60(2), 623-643.

Part II

Scheduling in Semiconductor Manufacturing

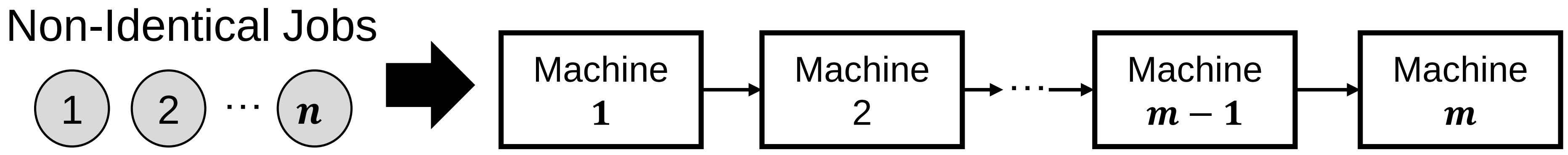
(joint work with **Tae-Sun Yu**)

- Waferfab problem description
- Flow shops with Reentry
- Priority rules
- Conclusions

Classical Shop Scheduling

Conventional 'Flow Shop'

- $\boldsymbol{n}$ **jobs** are processed by $\boldsymbol{m}$ **machines** sequentially.
- Each machine $i = 1, \ldots, m$ is visited only once.

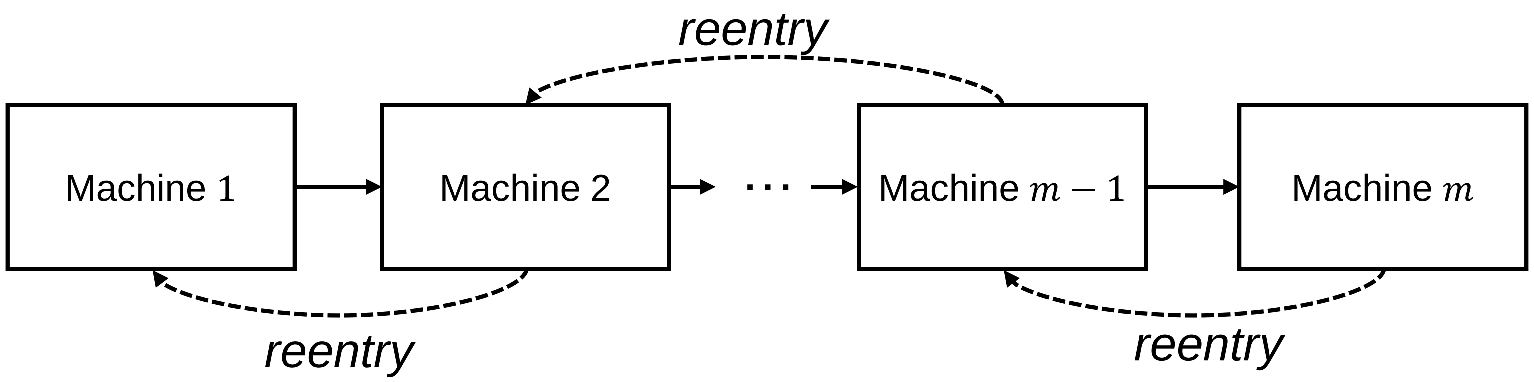


"Find the Optimal Job Sequence"
(the optimal permutation of $1, \ldots, n$)

Flow Shops with '*Reentry*'

Reentrant Flow Shop

- Each job is allowed to ***recirculate*** the system, i.e., some **machines can be visited more than once**.

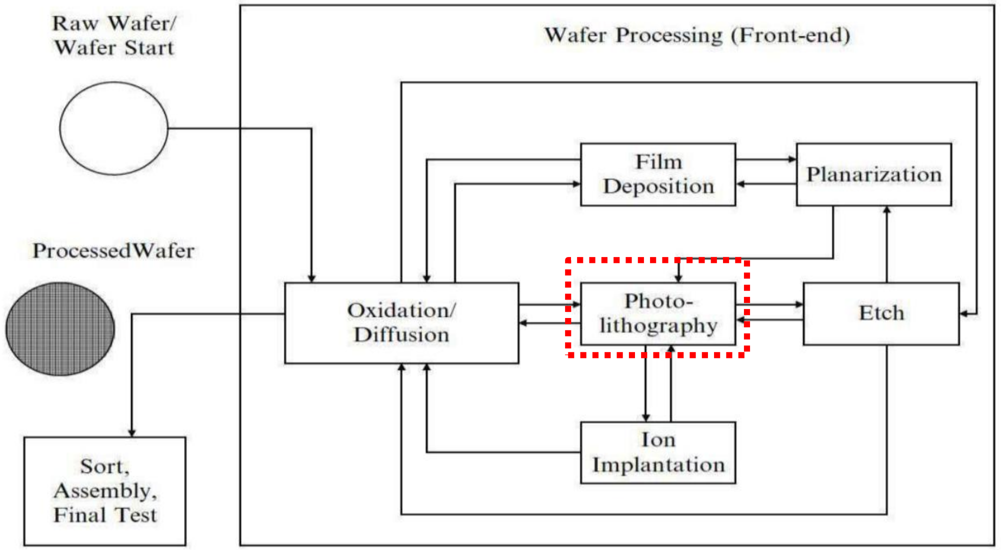


- Motivations: Job **Repair**, Job **Rework**, **Repetitive Processing**, etc.
  (Examples: Semiconductor, LED, Solar Cell, Printed Circuit Board, Textile Fabric, etc.)

Semiconductor Manufacturing

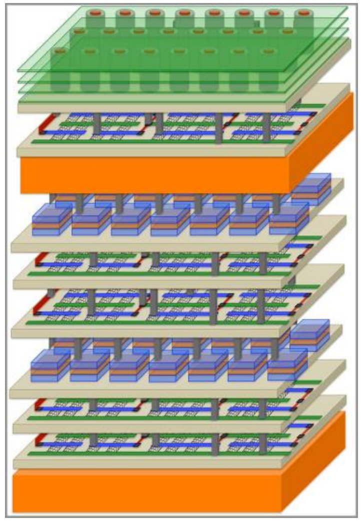
Wafer Fabrication Stage - Reentry is a common occurrence

- Types of Job Reentry
  - *Repetition*
    - When a recipe requires **a process (or equipment) to be repeatedly used**
    - General Wafer Fabrication Procedure: involves more than 24 layers
      - 1 Photolithography for 1 Layer
      - When conventional *Immersion Tools* are used: 8 Immersions for 1 Layer

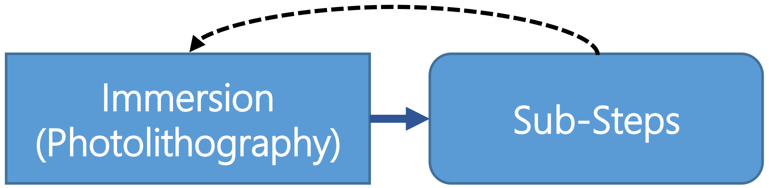


Photolithography
[Monch 2011]

Many Layers of Transistors
& Interconnects



Repeated 8 times for each layer

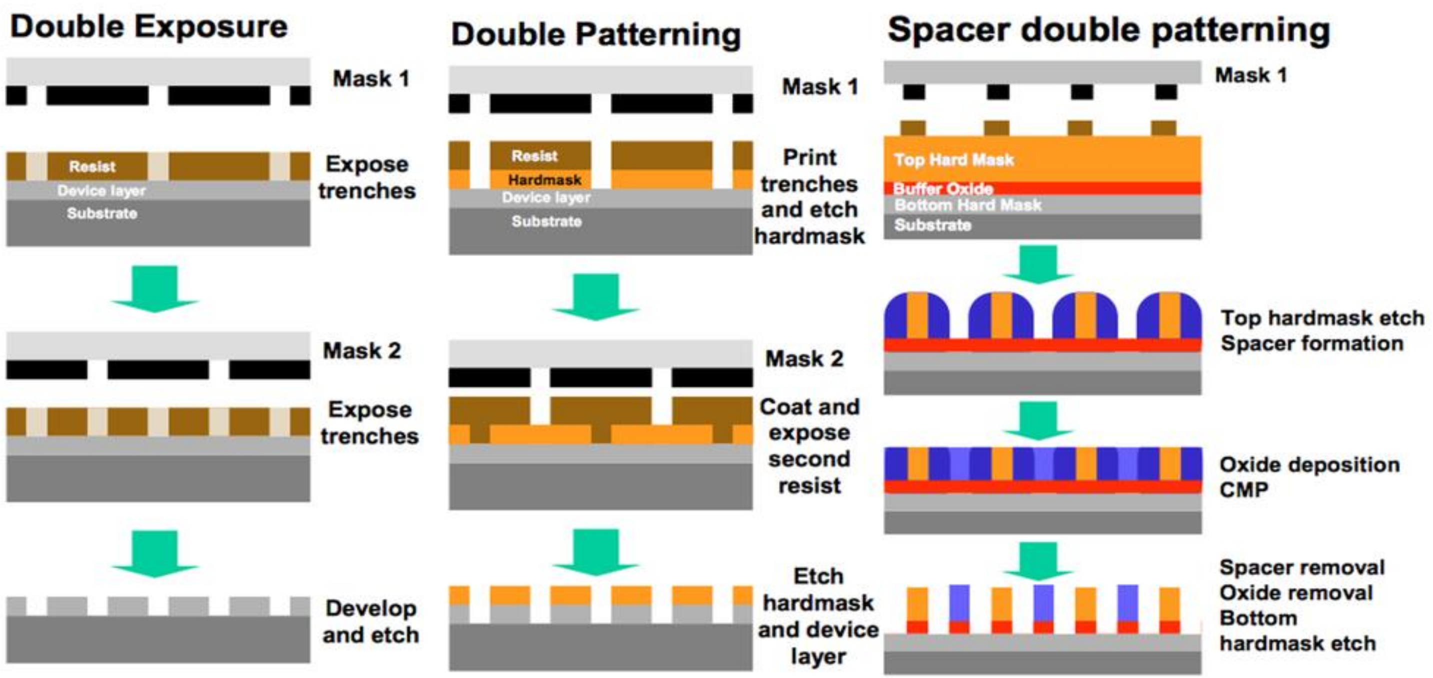


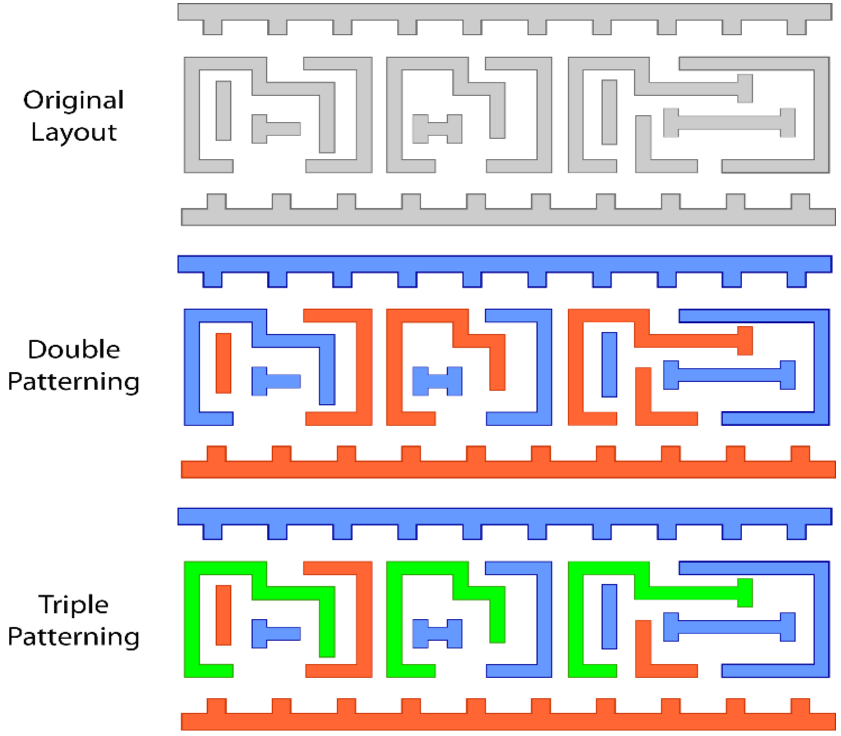
[1] "Chemistry at the bottom: Atomic layer deposition," Materialstoday.

Semiconductor Manufacturing

Wafer Fabrication Stage - Reentry is a common occurrence

- Types of Job Reentry
  - *Repetition*
    - Multi-Patterning Technology:
      - Multiple Photolithography Exposures & Etching to Increase Feature Density





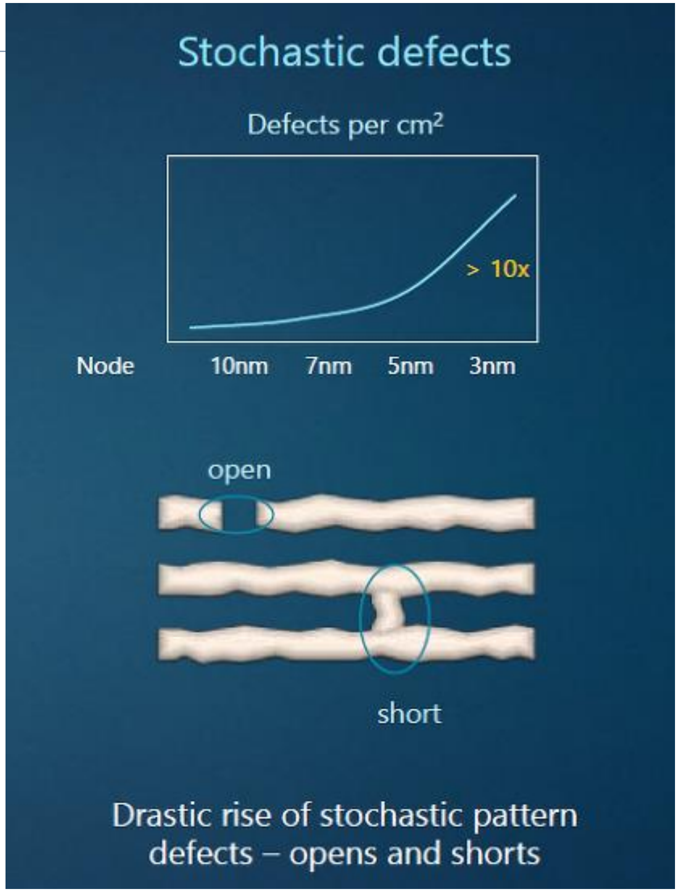
[1] "Chemistry at the bottom: Atomic layer deposition," Materialstoday.

Semiconductor Manufacturing

Wafer Fabrication Stage - Reentry is a common occurrence

- Types of Job Reentry
  - *Rework*
    - After the inspection when a process is found to be abnormally completed
    - Even in modern fabs some process steps' **rework rate are greater than 80%**
    - Ex: EUV Tools for *Photolithography* step ➔ Essential for nodes less than 5nm

Challenges in using EUV



150nm ➔ 110nm ➔ 80nm ➔ 55nm ➔ 40nm ➔ 28nm ➔ 20nm➔14nm➔ 10nm ➔ 7nm ➔ 5nm ➔ 3nm ➔ ???

Yield= 40~70%

[1] "Chemistry at the bottom: Atomic layer deposition," Materialstoday.

https://infohunter.tistory.com/20

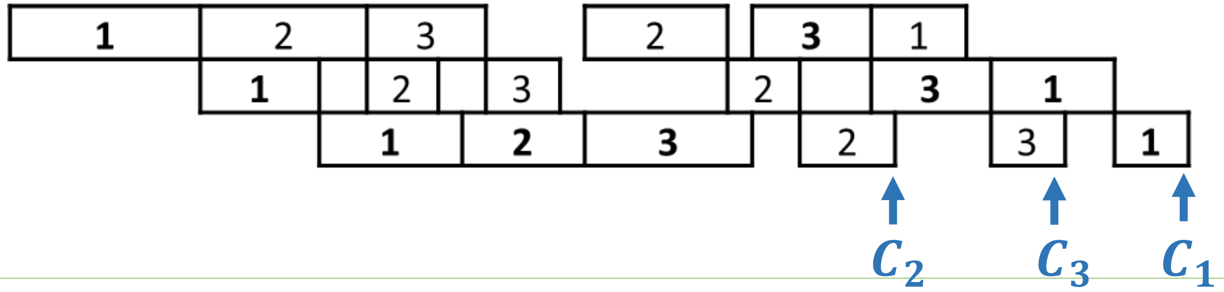
Machine Environment

Notations

- Job $j = 1, \ldots, n$ ***recirculates*** N_j times ➔ Job j proceeds N_j ***loops***.
- Each job is processed by machines $i = 1, \ldots, m$ sequentially.
- Let p_{ijk} be the ***processing time*** of job j on machine i in *loop* k.
- Let C_j be the ***completion time*** of job j.

Scheduling Measure

- C_{max}: The ***maximum completion time*** $\max_{j=1,\ldots,n} C_j$ ➔ ***Makespan***



Then, $C_{max} = C_1$

Preliminaries

"General flow shops are NP-hard for most scheduling measures when $m \geq 3$"

- Therefore, we consider the following subclasses of the ***Ordered Flow Shop***

  - **Proportionate** Flow Shop [Smith et al. 1975]
    ➔ $p_{ij} = p_j$ for all $i = 1, \dots, m$ and $j = 1, \dots, n$.

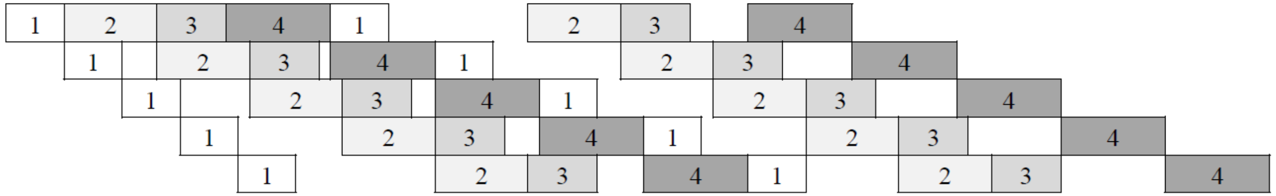
  - **Machine-Ordered** Flow Shop:
    ➔ $p_{ij} = p_i$ for all $i = 1, \dots, m$ and $j = 1, \dots, n$.

Makespan Invariance Property for Conventional Proportionate or Machine Ordered Flow Shops without Reentry

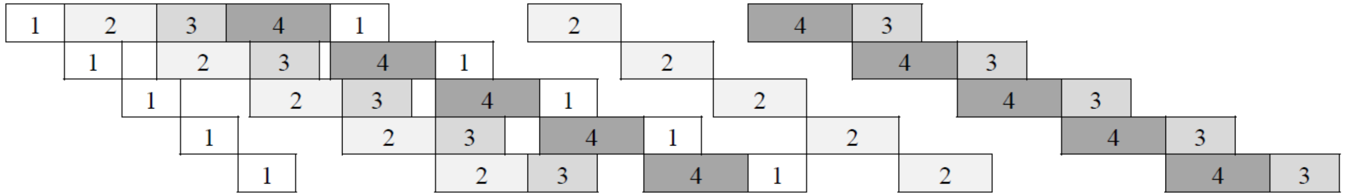
∴ Any permutation sequence yields the same makespan and is *optimal*

Makespan Minimization   Job Reentry with Same Number of Loops

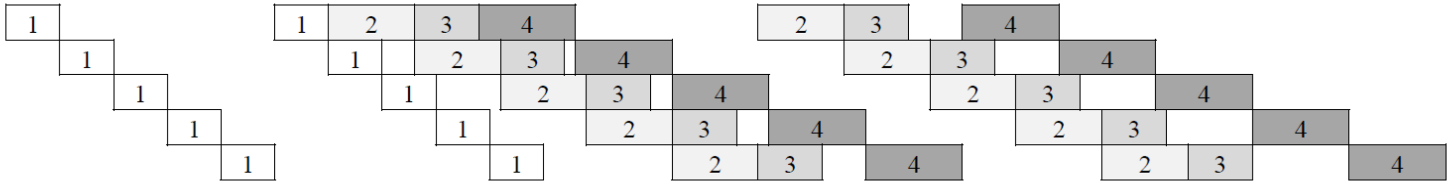
- ***Makespan Invariance Property*** does not hold when ***Job Reentry*** is considered
  - That is, the makespan is not the same for all permutation schedules



(a) Sequence $\ell_{11} \to \ell_{21} \to \ell_{31} \to \ell_{41} \to \ell_{12} \to \ell_{22} \to \ell_{32} \to \ell_{42}$.



(b) Sequence $\ell_{11} \to \ell_{21} \to \ell_{31} \to \ell_{41} \to \ell_{12} \to \ell_{22} \to \ell_{42} \to \ell_{32}$.



(c) Sequence $\ell_{11} \to \ell_{12} \to \ell_{21} \to \ell_{31} \to \ell_{41} \to \ell_{22} \to \ell_{32} \to \ell_{42}$.

We first examine the makespan minimization problem

Loopwise Cyclic (LC) Sequence

- We first define a particular class of permutation sequences: ***LC Sequences***

Loopwise Cyclic (LC) Sequence
Condition $(\boldsymbol{i})$**:** Lower indexed loops are sequenced earlier than higher indexed loops, i.e., loop ℓ_{jk} precedes loop $\ell_{j'k'}$ for all $j' \neq j$ and $k' > k$. ***Condition*** $(\boldsymbol{ii})$**:** Job processing order on all m machines are the same, i.e., each job maintains the same priority within each loop index.

- We confirm that the **Makespan Invariance Property now holds** among all possible LC sequences
- The following theorem is established as well:

Theorem
Any LC sequence minimizes the makespan of a *proportionate (machine-ordered)* reentrant flow shop.

Extension: Loop Effects

- Suppose that the job processing time p_{ijk} depends on the loop index k:

$$p_{ijk} = p_j + q_k$$

(q_k: loop effect of loop k)

"The makespan invariance property does not hold even among LC sequences"

Theorem
If q_k is increasing in k, the makespan is minimized by prioritizing the jobs of an LC sequence according to SPT.

*SPT≡ Shortest Processing Time First

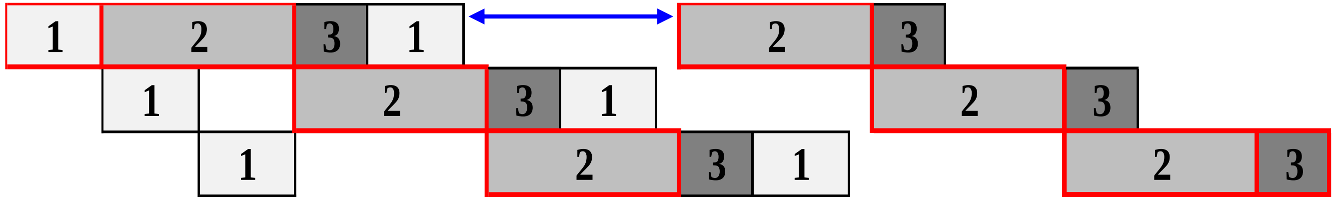
*Further Extension by **Makespan Reversibility***: LPT is optimal when q_k is decreasing in k

Managerial Insights on LC-SPT

Key Insights

- Application of SPT rule to LC sequences:

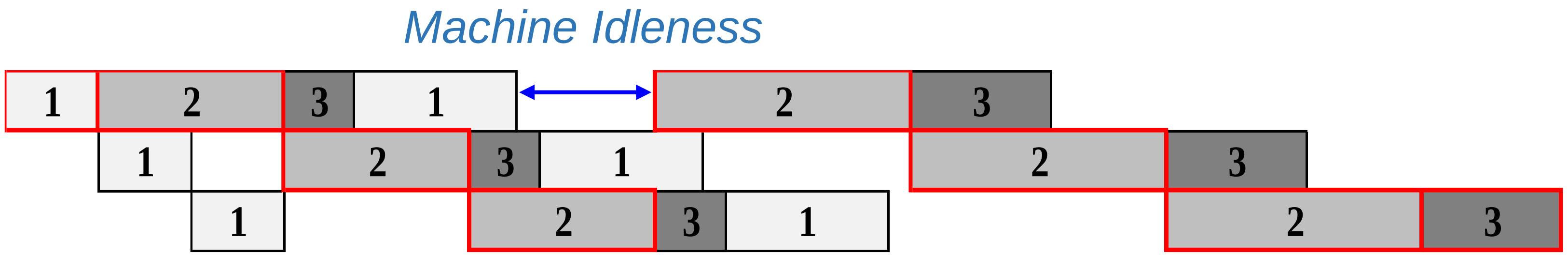
"Minimizes the *machine idleness* between the loops of the *bottleneck job*"



"When job recirculates ***machine idleness*** occurs due to the **bottleneck job**"

"Makespan is **determined by the position of bottleneck job** $j^* = \arg\max_{j=1,..,n} p_j$"

Managerial Insights on LC-SPT



To minimize such *Machine Idleness* caused by the bottleneck job,
the **workload between two consecutive loops** of the bottleneck job has to be **maximized**

SPT rule maximizes the workload between consecutive loops of the bottleneck job in LC sequences

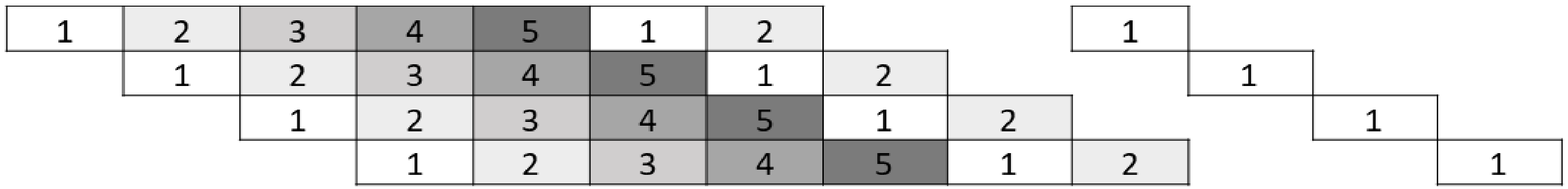
LC-SPT (LC-LPT) is optimal when loop effect q_k is increasing (decreasing) in k.

Reentry with Different Number of Loops

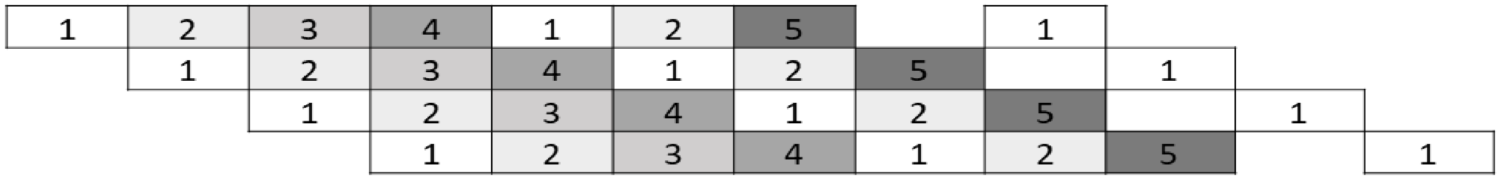
Dynamic Dispatching Method: *MRL-Dispatching*

More Remaining Loops First (MRL)
A job with larger number of remaining loops has priority over a job with a smaller number of remaining loops.

- Static method: LC-MRL minimizes makespan under certain conditions



- Dynamic method: MRL dispatching minimizes makespan under certain conditions



Reentry with Different Number of Loops

Dynamic Dispatching Method: *LRL-Dispatching*

Least Remaining Loops First (LRL)
A job with a smaller number of remaining loops has priority over a job with a larger number of remaining loops.

- Static method: LC-LRL minimizes the Total Completion Time (flow time) under fairly general conditions
- Dynamic method: LRL dispatching minimizes the Total Completion Time under fairly general conditions

Optimality Conditions of *MRL-Dispatching*

Experimental Results on MRL-Dispatching:

- Average Optimality Gap $\leq 1\%$
- Worst Case Performance = $\frac{C_{max}(MRL)}{C_{max}(OPT)} = 1.07.$

- Special Case Analysis (*Agreeability Conditions*):
  - MRL-Dispatching minimizes the makespan in reentrant flow shops if $p_{ijk} = p$ for all i, j, k.
  - MRL-Dispatching minimizes the makespan if

    (i) $np_1 \leq \sum p_i$ in a machine-ordered flow shop; or

    (ii) $np_j \leq \sum p_j$ for all j in a proportionate flow shop.

Future Extensions of Reentry Models

- **Extensions of Machine Environment**
  - $p_{ijk} = \alpha_i + \beta_j$
  - $p_{ijk} = p_j / s_i$ (different machine speeds)

- **Different Objective Functions (Current Research)**
  - Total Completion Time ➔ Least Remaining Loops First (LRL)
  - Due date related objective functions

- **Stochastic Environment (Current Research)**
  - Stochastic Processing Time (Stochastic Convexity)
  - Probabilistic Reentry

Future Research Directions for Flow Shops

- **Flexible Flow Shops with Reentry**
  - Probabilistic Reentry

- **Hybrid Techniques Using also Constraint Programming**
  - Framework Design
  - Hybrid Heuristics

Thank You !

Any Questions ?

Iterated Greedy Matheuristic Overview

- **Initial heuristic (IH)**
  - On the empty schedule,
  - we put one cast at a time
  - while preserving the former schedule
    - machine assignment of charge
    - precedence relationship between charges

➢ **to achieve a good initial schedule**

- **Destruction & Construction (DC)**
  - We select some charges to be rearranged
    - DC cast (**Dcast**): change a cast's position
    - DC charge (**Dcharge**): change machine assignments
  - We rearrange selected charges by solving a MIP subproblem which is much smaller than the MIP model describing the whole problem

➢ **to find a better schedule**

➢ **IGM:** To search for a good schedule by **IH** → **n * [Dcast → Dcharge → MIP$_{sub}$]** →

(MIP improvement -- potentially optimal)