

# Integration of control theory and scheduling methods for supply chain management

Kaushik Subramanian   James B. Rawlings   Christos T. Maravelias  
Jesus Flores-Cerrillo   Lawrence Megan

Department of  
Chemical and Biological Engineering

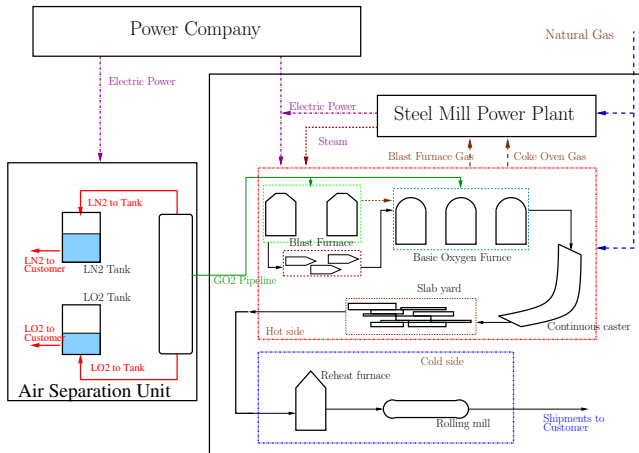


Praxair, Inc.



FOCAPO/CPC 2012  
Savannah, Georgia  
January 8–13, 2012

# Motivation: Praxair-Steel mill supply chain



- Scheduling model for the steel mill
- Inventory management for Praxair
- Steel mill and Praxair choices are coupled
- Cooperative operation can reduce operating costs

## We present results on

- 1 Cooperative model predictive control for inventory management
  - ▶ Tailored to fit the distributed nature of inventory management
- 2 State space approach to scheduling
  - ▶ Using control tools in scheduling

## We present results on

- 1 Cooperative model predictive control for inventory management
  - ▶ Tailored to fit the distributed nature of inventory management
- 2 State space approach to scheduling
  - ▶ Using control tools in scheduling

## as steps towards implementing MPC on a complex supply chain

- Distributed approach: each node of the supply chain makes its own decisions
- Guaranteed closed-loop properties: **feasibility, stability**

- Detailed optimization model + rolling horizon guarantees . . .

- Detailed optimization model + rolling horizon guarantees . . .  
no reasonable closed-loop properties!

- Detailed optimization model + rolling horizon guarantees . . .  
no reasonable closed-loop properties!
- The last 15 years of centralized MPC research provide design guidelines that ensure:

- Detailed optimization model + rolling horizon guarantees . . .  
no reasonable closed-loop properties!
- The last 15 years of centralized MPC research provide design guidelines that ensure: (i) recursive feasibility, (ii) nominal stability, and (iii) robustness to unmodeled disturbances

- Detailed optimization model + rolling horizon guarantees . . .  
no reasonable closed-loop properties!
- The last 15 years of centralized MPC research provide design guidelines that ensure: (i) recursive feasibility, (ii) nominal stability, and (iii) robustness to unmodeled disturbances
- Distributed MPC enables use of many interacting decision makers to improve overall system performance

# Conclusions

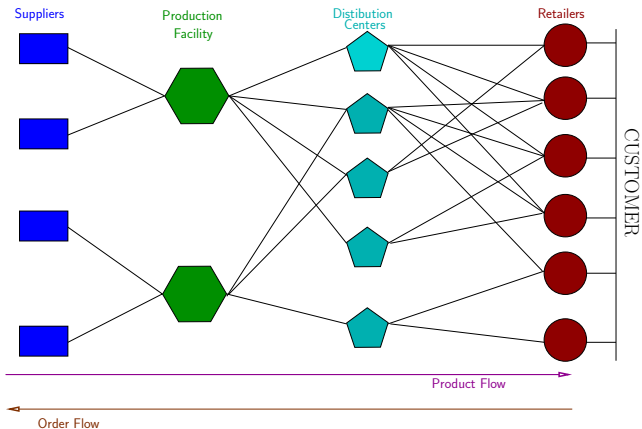
- Detailed optimization model + rolling horizon guarantees . . .  
no reasonable closed-loop properties!
- The last 15 years of centralized MPC research provide design guidelines that ensure: (i) recursive feasibility, (ii) nominal stability, and (iii) robustness to unmodeled disturbances
- Distributed MPC enables use of many interacting decision makers to improve overall system performance
- Current status: suboptimal MPC theory + distributed design so

# Conclusions

- Detailed optimization model + rolling horizon guarantees . . .  
no reasonable closed-loop properties!
- The last 15 years of centralized MPC research provide design guidelines that ensure: (i) recursive feasibility, (ii) nominal stability, and (iii) robustness to unmodeled disturbances
- Distributed MPC enables use of many interacting decision makers to improve overall system performance
- Current status: suboptimal MPC theory + distributed design so  
distributed MPC  $\subset$  suboptimal MPC

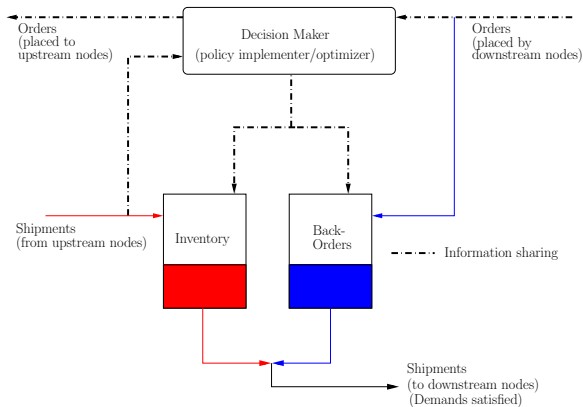
# Supply chain modeling

Beamon (1998); Angerhofer and Angelides (2000); Shah (2005)



- Each unit is a node.
- Every connection indicates two types of flows between the nodes.
- Production facility model (second part of talk)

# Modeling a node



- Each node modeled as two tanks
- States: inventory, back-orders
- Inputs: shipments, orders
- Disturbances: inputs of other nodes, customer demand

## Classical control theory

- Block diagram, Transfer function based approach
- Towill (1982); Dejonckheere et al. (2003); White (1999); Lin et al. (2004); Perea López et al. (2001) among others

## Classical control theory

- Block diagram, Transfer function based approach
- Towill (1982); Dejonckheere et al. (2003); White (1999); Lin et al. (2004); Perea López et al. (2001) among others

## Stochastic dynamic programming

- Classic results like  $(s, S)$  policy, order-up-to policy
- Clark and Scarf (1960); Federgruen and Zipkin (1984); Federgruen (1993); Shang and Song (2003); Dong and Lee (2003) among others

# Literature survey

## Classical control theory

- Block diagram, Transfer function based approach
- Towill (1982); Dejonckheere et al. (2003); White (1999); Lin et al. (2004); Perea López et al. (2001) among others

## Stochastic dynamic programming

- Classic results like  $(s, S)$  policy, order-up-to policy
- Clark and Scarf (1960); Federgruen and Zipkin (1984); Federgruen (1993); Shang and Song (2003); Dong and Lee (2003) among others

## Game theory

- Studies the strategic interaction between agents
- Leng and Parlar (2005); Cachon and Netessine (2006); Cachon and Zipkin (1999); Raghunathan (2003); Nagarajan and Sošić (2008) among others

# Rolling horizon optimization for supply chain operation

Authors	model	centralized	decent	noncoop	coop
Perea López et al. (2003)	M1	✓	✓	×	×
Mestan et al. (2006)	M1	✓	✓	✓	×
Dunbar and Desa (2007)	M2	×	×	✓	×
Seferlis and Giannelos (2004)	M3	✓	×	×	×
Braun et al. (2002)	M3	×	×	✓	×
Maestre et al. (2009)	M2	✓	×	✓	×
Li and Marlin (2009)	M3	robust	×	×	×

**M1** Multiple product supply chain with mixed integer production model

**M2** Single product supply chain with approximate production model

**M3** Multiple product supply chain with approximate production model

## Distributed MPC: consider two subsystems

Objective functions	$V_1(u_1, u_2), V_2(u_1, u_2)$
and	$V(u_1, u_2) = w_1 V_1(u_1, u_2) + w_2 V_2(u_1, u_2)$
decision variables	$u_1 \in \Omega_1, u_2 \in \Omega_2$

## Distributed MPC: consider two subsystems

Objective functions	$V_1(u_1, u_2), V_2(u_1, u_2)$
and	$V(u_1, u_2) = w_1 V_1(u_1, u_2) + w_2 V_2(u_1, u_2)$
decision variables	$u_1 \in \Omega_1, u_2 \in \Omega_2$
Centralized Control	$\min_{u_1, u_2 \in \Omega_1 \times \Omega_2} V(u_1, u_2)$
(Pareto optimal)	

# Distributed MPC: consider two subsystems

Objective functions	$V_1(u_1, u_2), V_2(u_1, u_2)$
and	$V(u_1, u_2) = w_1 V_1(u_1, u_2) + w_2 V_2(u_1, u_2)$
decision variables	$u_1 \in \Omega_1, u_2 \in \Omega_2$
Centralized Control	$\min_{u_1, u_2 \in \Omega_1 \times \Omega_2} V(u_1, u_2)$
(Pareto optimal)	
Cooperative Control	$\min_{u_1 \in \Omega_1} V(u_1, u_2) \quad \min_{u_2 \in \Omega_2} V(u_1, u_2)$
(Pareto optimal)	

# Distributed MPC: consider two subsystems

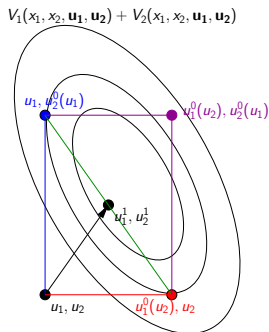
Objective functions	$V_1(u_1, u_2), V_2(u_1, u_2)$
and	$V(u_1, u_2) = w_1 V_1(u_1, u_2) + w_2 V_2(u_1, u_2)$
decision variables	$u_1 \in \Omega_1, u_2 \in \Omega_2$
Centralized Control	$\min_{u_1, u_2 \in \Omega_1 \times \Omega_2} V(u_1, u_2)$
(Pareto optimal)	
Cooperative Control	$\min_{u_1 \in \Omega_1} V(u_1, u_2) \quad \min_{u_2 \in \Omega_2} V(u_1, u_2)$
(Pareto optimal)	
Noncooperative Control	$\min_{u_1 \in \Omega_1} V_1(u_1, u_2) \quad \min_{u_2 \in \Omega_2} V_2(u_1, u_2)$
(Nash equilibrium)	

# Distributed MPC: consider two subsystems

Objective functions	$V_1(u_1, u_2), V_2(u_1, u_2)$
and	$V(u_1, u_2) = w_1 V_1(u_1, u_2) + w_2 V_2(u_1, u_2)$
decision variables	$u_1 \in \Omega_1, u_2 \in \Omega_2$
Centralized Control	$\min_{u_1, u_2 \in \Omega_1 \times \Omega_2} V(u_1, u_2)$
(Pareto optimal)	
Cooperative Control	$\min_{u_1 \in \Omega_1} V(u_1, u_2) \quad \min_{u_2 \in \Omega_2} V(u_1, u_2)$
(Pareto optimal)	
Noncooperative Control	$\min_{u_1 \in \Omega_1} V_1(u_1, u_2) \quad \min_{u_2 \in \Omega_2} V_2(u_1, u_2)$
(Nash equilibrium)	
Decentralized Control	$\min_{u_1 \in \Omega_1} \tilde{V}_1(u_1) \quad \min_{u_2 \in \Omega_2} \tilde{V}_2(u_2)$

## Centralized MPC

$$\begin{aligned} \min_{\mathbf{u}_1, \mathbf{u}_2} & V_1(x_1, x_2, \mathbf{u}_1, \mathbf{u}_2) + V_2(x_1, x_2, \mathbf{u}_1, \mathbf{u}_2) \\ \text{s.t.} & \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^+ = \begin{bmatrix} A_1 & \\ & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\ & u_1 \in \Omega_1 \quad u_2 \in \Omega_2 \end{aligned}$$



## Cooperative MPC

- Each subsystem solves for *its local decision variables*
- Each iterate is **feasible**
- Iterations converge to **centralized optimal** solution
- Equivalent to *Gauss-Jacobi* parallel optimization algorithm to solve the centralized problem

# Closed-loop properties

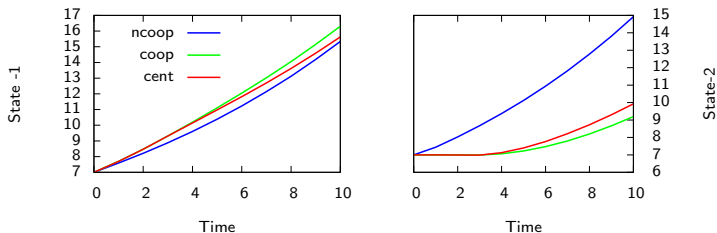
- 1 Input  $u = \kappa(x)$ : Solution of the optimization problem
- 2 Closed loop evolves as  $x^+ = Ax + B\kappa(x)$

What can we conclude about the closed-loop dynamics?

# Closed-loop properties

- 1 Input  $u = \kappa(x)$ : Solution of the optimization problem
- 2 Closed loop evolves as  $x^+ = Ax + B\kappa(x)$

What can we conclude about the closed-loop dynamics?

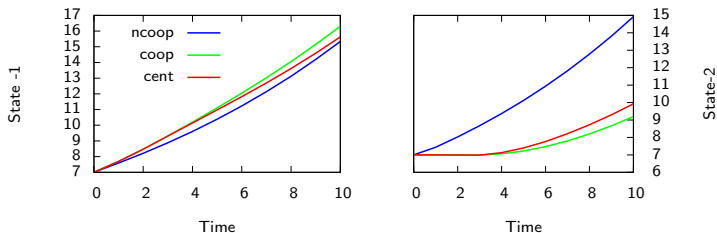


**Nothing!** The optimization has not even stabilized the system

# Closed-loop properties

- 1 Input  $u = \kappa(x)$ : Solution of the optimization problem
- 2 Closed loop evolves as  $x^+ = Ax + B\kappa(x)$

What can we conclude about the closed-loop dynamics?



**Nothing!** The optimization has not even stabilized the system

- 3 Need to *design* closed-loop stable distributed MPC algorithms

- Cost function: 
$$V_N = \sum_{j=0}^{N-1} \ell(x(j), u(j)) + V_f(x(N))$$

- **Cost function:** 
$$V_N = \sum_{j=0}^{N-1} \ell(x(j), u(j)) + V_f(x(N))$$

- **Optimization:** 
$$\min_{\mathbf{u}} V_N(\mathbf{u}, x(0))$$

subject to:

$$\begin{aligned} \mathbf{u} &= \{u(0), u(1), \dots, u(N-1)\} \\ x^+ &= f(x, u) \quad x(N) \in \mathbb{X}_f \quad \mathbf{u} \in \mathbb{U}^N \end{aligned}$$

- **Cost function:** 
$$V_N = \sum_{j=0}^{N-1} \ell(x(j), u(j)) + V_f(x(N))$$
- **Optimization:** 
$$\min_{\mathbf{u}} V_N(\mathbf{u}, x(0))$$

subject to:

$$\mathbf{u} = \{u(0), u(1), \dots, u(N-1)\}$$
$$x^+ = f(x, u) \quad x(N) \in \mathbb{X}_f \quad \mathbf{u} \in \mathbb{U}^N$$
- **Closed-loop dynamics:**  $x^+ = f(x, u^0(x))$

- **Cost function:**  $V_N = \sum_{j=0}^{N-1} \ell(x(j), u(j)) + V_f(x(N))$

- **Optimization:**  $\min_{\mathbf{u}} V_N(\mathbf{u}, x(0))$

subject to:

$$\mathbf{u} = \{u(0), u(1), \dots, u(N-1)\}$$
$$x^+ = f(x, u) \quad x(N) \in \mathbb{X}_f \quad \mathbf{u} \in \mathbb{U}^N$$

- **Closed-loop dynamics:**  $x^+ = f(x, u^0(x))$

- Then, we can show that the **optimal cost**  $V_N^0(x)$  is a **Lyapunov function**

$$V_N^0(x^+) - V_N^0(x) \leq -\alpha(|x|) \quad \text{cost decrease}$$

- **Cost function:** 
$$V_N = \sum_{j=0}^{N-1} \ell(x(j), u(j)) + V_f(x(N))$$

- **Optimization:** 
$$\min_{\mathbf{u}} V_N(\mathbf{u}, x(0))$$

subject to:

$$\begin{aligned} \mathbf{u} &= \{u(0), u(1), \dots, u(N-1)\} \\ x^+ &= f(x, u) \quad x(N) \in \mathbb{X}_f \quad \mathbf{u} \in \mathbb{U}^N \end{aligned}$$

- **Closed-loop dynamics:**  $x^+ = f(x, u^0(x))$

- Then, we can show that the **optimal cost**  $V_N^0(x)$  is a **Lyapunov function**

$$V_N^0(x^+) - V_N^0(x) \leq -\alpha(|x|) \quad \text{cost decrease}$$

- **Lyapunov theory ensures closed-loop stability, asymptotic convergence to origin, and more...**

- Consider **feasible input sequence**  $\mathbf{u} = (u(0), u(1), \dots, u(N-1))$  for state  $x$ .

$$\mathbf{u} \in \mathbb{U}^N \quad x(N) \in \mathbb{X}_f$$

- Consider **feasible input sequence**  $\mathbf{u} = (u(0), u(1), \dots, u(N-1))$  for state  $x$ .

$$\mathbf{u} \in \mathbb{U}^N \quad x(N) \in \mathbb{X}_f$$

- For successor state  $x^+ = f(x, u^0(x))$ , consider **warm start**

$$\tilde{\mathbf{u}} = (u(1), u(2), \dots, u(N-1), \kappa_f(x(N)))$$

- Consider **feasible input sequence**  $\mathbf{u} = (u(0), u(1), \dots, u(N-1))$  for state  $x$ .

$$\mathbf{u} \in \mathbb{U}^N \quad x(N) \in \mathbb{X}_f$$

- For successor state  $x^+ = f(x, u^0(x))$ , consider **warm start**

$$\tilde{\mathbf{u}} = (u(1), u(2), \dots, u(N-1), \kappa_f(x(N)))$$

- Warm start satisfies cost decrease due to properties of  $\mathbb{X}_f$

$$V_N(x^+, \tilde{\mathbf{u}}) - V_N(x, \mathbf{u}) \leq -\ell(x, u^0(x))$$

- Consider **feasible input sequence**  $\mathbf{u} = (u(0), u(1), \dots, u(N-1))$  for state  $x$ .

$$\mathbf{u} \in \mathbb{U}^N \quad x(N) \in \mathbb{X}_f$$

- For successor state  $x^+ = f(x, u^0(x))$ , consider **warm start**

$$\tilde{\mathbf{u}} = (u(1), u(2), \dots, u(N-1), \kappa_f(x(N)))$$

- Warm start satisfies cost decrease due to properties of  $\mathbb{X}_f$

$$V_N(x^+, \tilde{\mathbf{u}}) - V_N(x, \mathbf{u}) \leq -\ell(x, u^0(x))$$

- We can show that  $V_N(x, \mathbf{u})$  is a Lyapunov function.

## Further optimization

- Must ensure that the iterates are feasible  $\mathbf{u} \in \mathbb{U}^N$
- Must ensure cost drop
- Control is *very* nonunique!
  - ▶  $V_N(x, \mathbf{u}) \leq V_N(x, \tilde{\mathbf{u}})$

# Suboptimal MPC

## Further optimization

- Must ensure that the iterates are feasible  $\mathbf{u} \in \mathbb{U}^N$
- Must ensure cost drop
- Control is *very* nonunique!
  - ▶  $V_N(x, \mathbf{u}) \leq V_N(x, \tilde{\mathbf{u}})$

## Cooperative algorithm ensures

- Feasible iterates
- that converge to the optimal solution

for convex optimization problems with **uncoupled constraints**

# Suboptimal MPC

## Further optimization

- Must ensure that the iterates are feasible  $\mathbf{u} \in \mathbb{U}^N$
- Must ensure cost drop
- Control is *very* nonunique!
  - ▶  $V_N(x, \mathbf{u}) \leq V_N(x, \tilde{\mathbf{u}})$

## Cooperative algorithm ensures

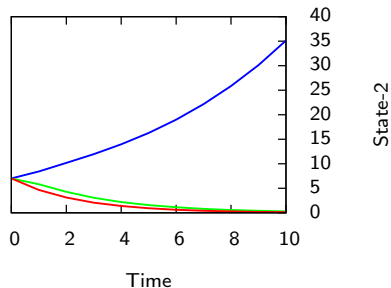
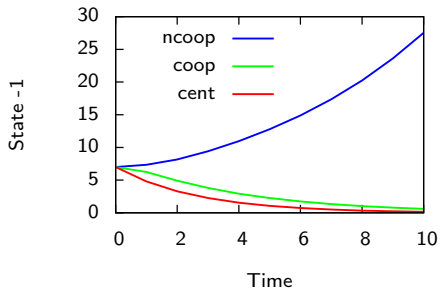
- Feasible iterates
- that converge to the optimal solution

for convex optimization problems with **uncoupled constraints**

## Stability theory for cooperative MPC

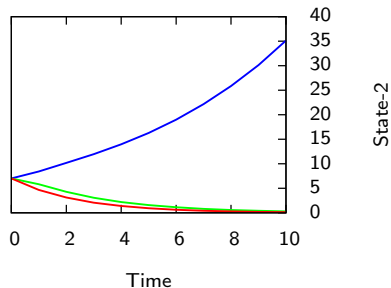
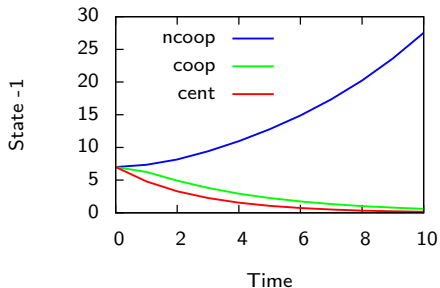
**Remove coupled constraints** in the cooperative MPC optimization problem so that cooperative MPC is an implementation of suboptimal MPC.

## Revisit the example...



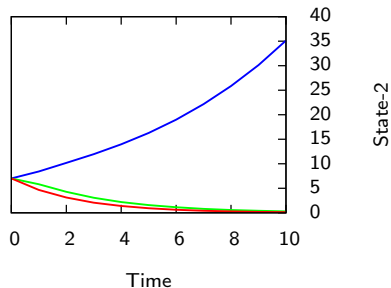
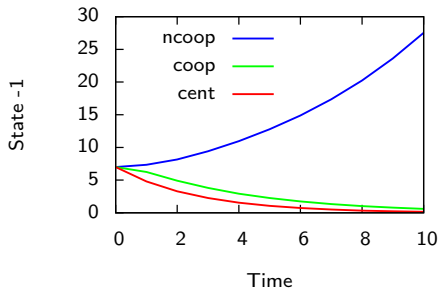
- optimal MPC theory ensures cent is stable ...

## Revisit the example...



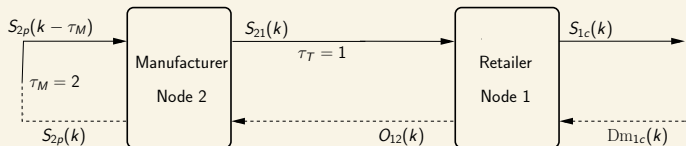
- optimal MPC theory ensures cent is stable ...
- suboptimal MPC theory ensures coop is stable ...

## Revisit the example...



- optimal MPC theory ensures cent is stable ...
- suboptimal MPC theory ensures coop is stable ...
- noncoop and decent are on their own!

# Supply chain



- Transportation delay: 1 time unit. Manufacturing delay: 2 time units.
- **Control objective:** To keep inventories close to target inventory while satisfying demands.
- **Nominal demand:** 8 units every time period.

- Individual node models do not contain enough information to predict how local orders affect future inventory.

- Individual node models do not contain enough information to predict how local orders affect future inventory.
- Order-up-to policy: “proportional controller”

$$O_{12}(k) = \begin{cases} I_{v_t} - I_{v_1}(k) & \text{if } I_{v_1} \leq I_{v_t} \\ 0 & \text{otherwise} \end{cases}$$

- Individual node models do not contain enough information to predict how local orders affect future inventory.
- **Order-up-to policy:** “proportional controller”

$$O_{12}(k) = \begin{cases} I_{v_t} - I_{v_1}(k) & \text{if } I_{v_1} \leq I_{v_t} \\ 0 & \text{otherwise} \end{cases}$$

- **Inventory position control:** Define a new “controlled output”

$$I_p(k) = I_{v_1}(k) - S_{1c}(k) + O_{12}(k)$$

## Decentralized operation

- No sharing of information among nodes.
- Nodes decide ordering based on a supply chain policy.
- Each node minimizes its control objective

# Supply chain example

## Decentralized operation

- No sharing of information among nodes.
- Nodes decide ordering based on a supply chain policy.
- Each node minimizes its control objective

## Noncooperative operation

- Information shared among nodes. Each node knows the other nodes shipments and orders
- Nodes decide ordering based on a supply chain policy
- Each node minimizes its control objective

# Supply chain example

## Decentralized operation

- No sharing of information among nodes.
- Nodes decide ordering based on a supply chain policy.
- Each node minimizes its control objective

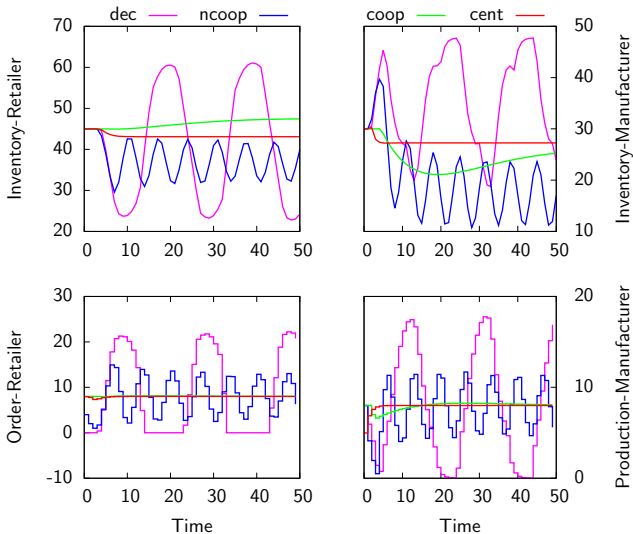
## Noncooperative operation

- Information shared among nodes. Each node knows the other nodes shipments and orders
- Nodes decide ordering based on a supply chain policy
- Each node minimizes its control objective

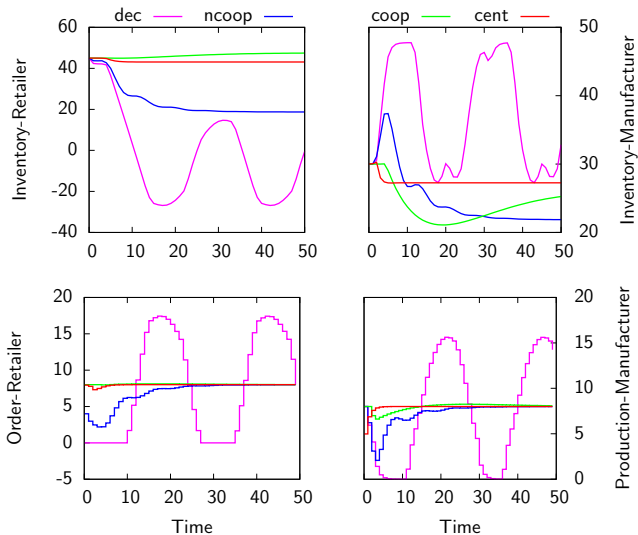
## Cooperative operation

- Information shared among nodes
- Each node minimizes **the supply chain control objective**
- **Nodes do not implement supply chain policies.** Nodes use supply chain model to predict dynamics.

# Results: Order-up-to policy



# Results: Inventory position control policy



## Confirm previous observations

- Centralized optimization reduces bullwhip effect
- Information sharing can improve performance

# Key observations

## Confirm previous observations

- Centralized optimization reduces bullwhip effect
- Information sharing can improve performance

## New results

- Demonstrated cooperative MPC
  - ▶ Compatible with the distributed nature of inventory control
  - ▶ Efficient operation (if the entire supply chain is owned by a single firm)
- Established desirable closed-loop properties of cooperative MPC

## Distributed supply chain optimization (part I)

- Distributed restart for coupled unstable systems. Handle warm start infeasibility caused by unmodeled disturbances.

## Distributed supply chain optimization (part I)

- Distributed restart for coupled unstable systems. Handle warm start infeasibility caused by unmodeled disturbances.
- Cooperative game theory. Explore using game theory methods for encouraging cooperation among competing players.

## Distributed supply chain optimization (part I)

- Distributed restart for coupled unstable systems. Handle warm start infeasibility caused by unmodeled disturbances.
- Cooperative game theory. Explore using game theory methods for encouraging cooperation among competing players.
- Issues of scale. Hierarchical decomposition (Stewart et al., 2010).

## Distributed supply chain optimization (part I)

- Distributed restart for coupled unstable systems. Handle warm start infeasibility caused by unmodeled disturbances.
- Cooperative game theory. Explore using game theory methods for encouraging cooperation among competing players.
- Issues of scale. Hierarchical decomposition (Stewart et al., 2010).
- Optimizing economic performance. Economic MPC (Angeli et al., 2011; Diehl et al., 2011)

# What is Scheduling?

*It is the allocation of **limited** resources to tasks over time*

## **Given are:**

- a) **Production facility data** (e.g., processing unit capacities)
- b) **Production recipes** (e.g., mixing rules, processing times/rates)
- d) **Production costs** (e.g., raw materials, utilities, changeovers)
- e) **Material availability** (e.g., raw material deliveries)
- f) **Resource availability** (e.g., maintenance schedule)
- g) **Production targets or orders with due dates.**

**Our goal is to find a least cost schedule that meets production targets subject to resource constraints.**

# What is Scheduling?

*It is the allocation of **limited** resources to tasks over time*

## Given are:

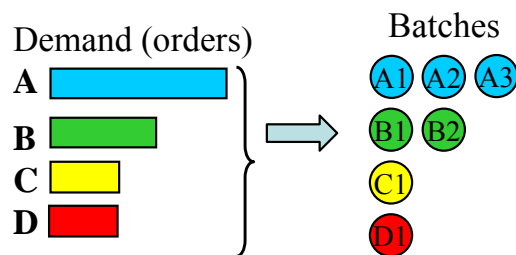
- Production facility data (e.g., processing unit capacities)
- Production recipes (e.g., mixing rules, processing times/rates)
- Production costs (e.g., raw materials, utilities, changeovers)
- Material availability (e.g., raw material deliveries)
- Resource availability (e.g., maintenance schedule)
- Production targets or orders with due dates.

**Our goal is to find a least cost schedule that meets production targets subject to resource constraints.**

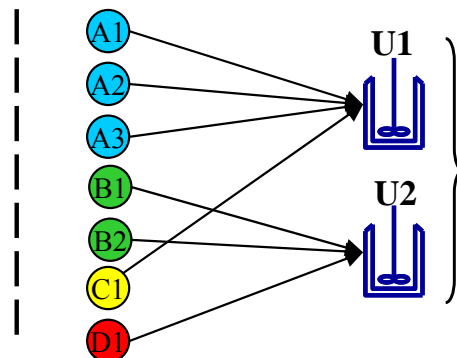
In the general problem, we seek to optimize our objective by making four types of decisions:

- Selection and sizing of batches to be carried out (batching)
- Assignment of batches to processing units or general resources.
- Sequencing of batches on processing units.
- Timing of batches.

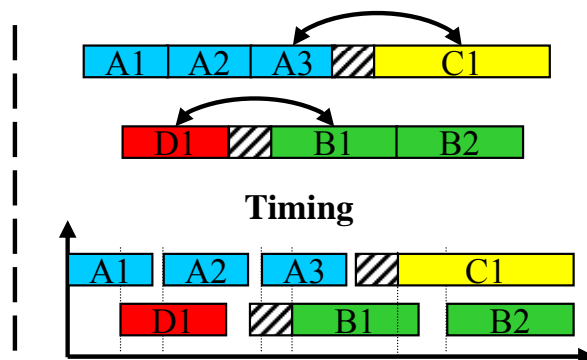
### Task selection (batching)



### Task-resource Assignment



### Sequencing (for unary resources)



# Preliminaries

---

- Systematic scheduling practiced in manufacturing since early 20<sup>th</sup> century
- First scheduling publications in the early 1950s <sup>1,2</sup>
- Extensive research in 1970s
  - Closely related to developments in computing and algorithms
  - Computational Complexity: *Job Sequencing* one of 21 NP-complete problems in (Karp, 1972)

<sup>1</sup> Salveson ME, *Econometrica*, 20(9), **1952**;

<sup>2</sup> Johnson SM, *Naval Research Logistics Quarterly*, 1(1), 61-68, **1954**.

# Preliminaries

- Systematic scheduling practiced in manufacturing since early 20<sup>th</sup> century
- First scheduling publications in the early 1950s <sup>1,2</sup>
- Extensive research in 1970s
  - Closely related to developments in computing and algorithms
  - Computational Complexity: *Job Sequencing* one of 21 NP-complete problems in (Karp, 1972)
- Widespread applications
  - Airlines industry (e.g., crew scheduling), transportation (e.g., vehicle routing), sports, educational institutions (e.g., timetabling), services (e.g., service center scheduling), manufacturing
- Chemical industries
  - Batch process scheduling (e.g., pharma, food industry, fine chemicals)
  - Continuous process scheduling (e.g., polymerization)
  - Transportation and unloading of crude oil

<sup>1</sup> Salveson ME, *Econometrica*, 20(9), **1952**;

<sup>2</sup> Johnson SM, *Naval Research Logistics Quarterly*, 1(1), 61-68, **1954**.

# Preliminaries

- Systematic scheduling practiced in manufacturing since early 20<sup>th</sup> century
- First scheduling publications in the early 1950s <sup>1,2</sup>
- Extensive research in 1970s
  - Closely related to developments in computing and algorithms
  - Computational Complexity: *Job Sequencing* one of 21 NP-complete problems in (Karp, 1972)
- Widespread applications
  - Airlines industry (e.g., crew scheduling), transportation (e.g., vehicle routing), sports, educational institutions (e.g., timetabling), services (e.g., service center scheduling), manufacturing
- Chemical industries
  - Batch process scheduling (e.g., pharma, food industry, fine chemicals)
  - Continuous process scheduling (e.g., polymerization)
  - Transportation and unloading of crude oil
- Challenging problem: small instances can be hard
  - Most *Open* problems in MIPLIB are scheduling related  
e.g., railway scheduling: 1,500 constraints, 1,083 variables, 794 binaries

<sup>1</sup> Salveson ME, *Econometrica*, 20(9), **1952**;

<sup>2</sup> Johnson SM, *Naval Research Logistics Quarterly*, 1(1), 61-68, **1954**.

# Preliminaries

- Systematic scheduling practiced in manufacturing since early 20<sup>th</sup> century
- First scheduling publications in the early 1950s <sup>1,2</sup>
- Extensive research in 1970s
  - Closely related to developments in computing and algorithms
  - Computational Complexity: *Job Sequencing* one of 21 NP-complete problems in (Karp, 1972)
- Widespread applications
  - Airlines industry (e.g., crew scheduling), transportation (e.g., vehicle routing), sports, educational institutions (e.g., timetabling), services (e.g., service center scheduling), manufacturing
- Chemical industries
  - Batch process scheduling (e.g., pharma, food industry, fine chemicals)
  - Continuous process scheduling (e.g., polymerization)
  - Transportation and unloading of crude oil
- Challenging problem: small instances can be hard
  - Most *Open* problems in MIPLIB are scheduling related  
e.g., railway scheduling: 1,500 constraints, 1,083 variables, 794 binaries
- Wide range of mixed-integer programming (MIP) approaches
  - *Batch-based* approaches for *sequential* production environment
  - *Material-based* approaches for *network* production environment

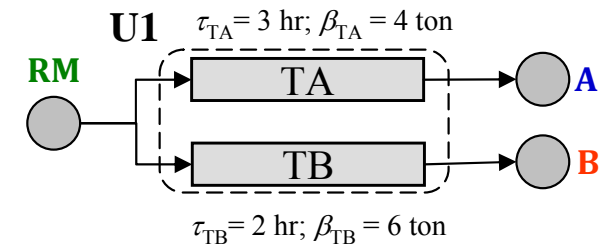
<sup>1</sup> Salvesson ME, *Econometrica*, 20(9), **1952**;

<sup>2</sup> Johnson SM, *Naval Research Logistics Quarterly*, 1(1), 61-68, **1954**.

# Unified Framework

- **Unified framework**
  - Material-based representation of all types of processing
  - *Sequential* processing represented using materials with special properties (constraints)
- **Basic concepts** (from STN; Kondili et al., 1993)
  - Processing units,  $j \in \mathbf{J}$
  - Processing tasks  $i \in \mathbf{I}$  (duration,  $\tau_i$ ; batchsize,  $\beta_i$ )
  - Materials (states),  $m \in \mathbf{M}$  (storage capacity,  $\zeta_m$ )

(Sundaramoorthy & Maravelias, 2011)



# Unified Framework

## Unified framework

- Material-based representation of all types of processing
- *Sequential* processing represented using materials with special properties (constraints)

## Basic concepts (from STN; Kondili et al., 1993)

- Processing units,  $j \in \mathbf{J}$
- Processing tasks  $i \in \mathbf{I}$  (duration,  $\tau_i$ ; batchsize,  $\beta_i$ )
- Materials (states),  $m \in \mathbf{M}$  (storage capacity,  $\zeta_m$ )

## Discrete-time representation

- Horizon,  $\eta$ , partitioned into  $T$  time periods,  $t \in \mathbf{T}$ , of length  $\delta = \eta/T$

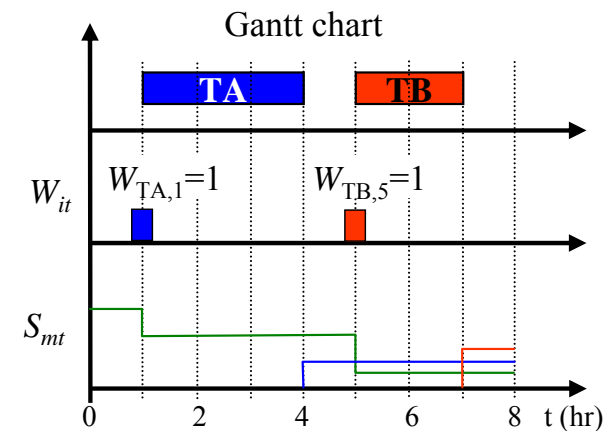
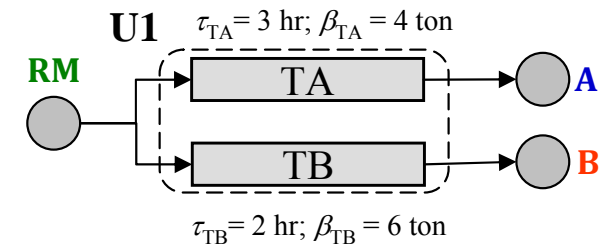
## Decision variables

- $W_{it} \in \{0, 1\}$ : = 1 if task  $i$  starts at time point  $t$
- $S_{mt} \in [0, \zeta_m]$ : inventory level of material  $m$  during period  $t$

## Mixed-integer programming (MIP) model

- Resource constraint: a unit can process at most one task at a time
- Material balance: calculation of inventory over time

(Sundaramoorthy & Maravelias, 2011)



# Unified Framework

## Unified framework

- Material-based representation of all types of processing
- *Sequential* processing represented using materials with special properties (constraints)

## Basic concepts (from STN; Kondili et al., 1993)

- Processing units,  $j \in \mathbf{J}$
- Processing tasks  $i \in \mathbf{I}$  (duration,  $\tau_i$ ; batchsize,  $\beta_i$ )
- Materials (states),  $m \in \mathbf{M}$  (storage capacity,  $\zeta_m$ )

## Discrete-time representation

- Horizon,  $\eta$ , partitioned into  $T$  time periods,  $t \in \mathbf{T}$ , of length  $\delta = \eta/T$

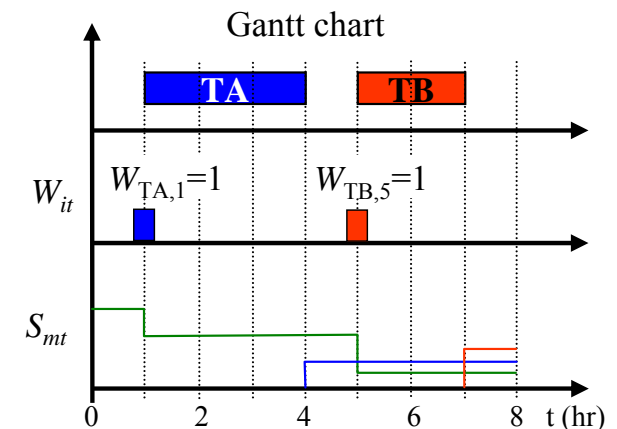
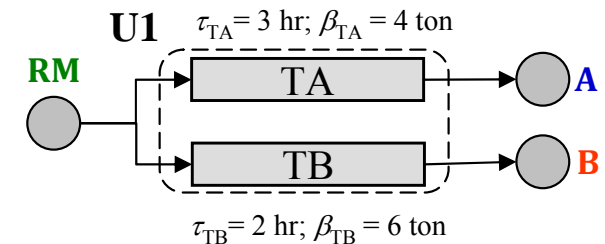
## Decision variables

- $W_{it} \in \{0, 1\}$ : = 1 if task  $i$  starts at time point  $t$
- $S_{mt} \in [0, \zeta_m]$ : inventory level of material  $m$  during period  $t$

## Mixed-integer programming (MIP) model

- Resource constraint: a unit can process at most one task at a time
- Material balance: calculation of inventory over time

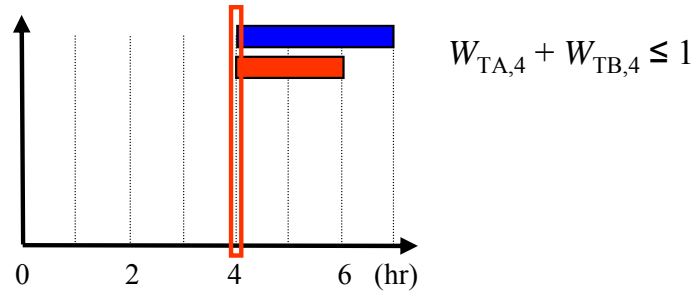
(Sundaramoorthy & Maravelias, 2011)



**Can we express the general scheduling MIP model in state-space form?**

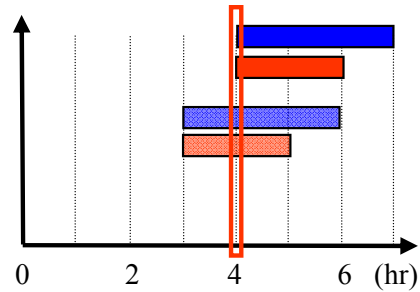
# Mixed-integer Programming Model

- Resource constraints
- Decision variable:  $W_{it} = 1$  when a task starts



# Mixed-integer Programming Model

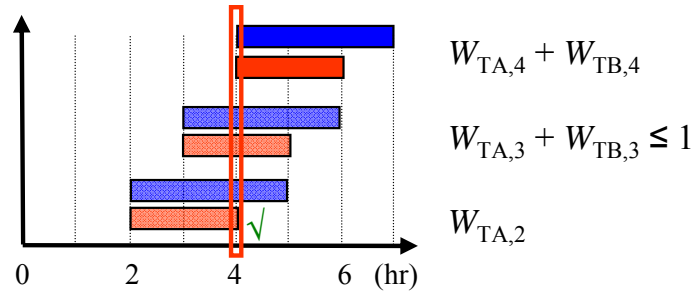
- Resource constraints
- Decision variable:  $W_{it} = 1$  when a task starts



$$W_{TA,4} + W_{TB,4} \leq 1$$
$$W_{TA,3} + W_{TB,3}$$

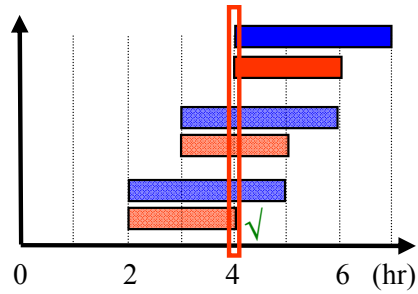
# Mixed-integer Programming Model

- Resource constraints
- Decision variable:  $W_{it} = 1$  when a task starts



# Mixed-integer Programming Model

- Resource constraints
- Decision variable:  $W_{it} = 1$  when a task starts



$$W_{TA,4} + W_{TB,4}$$

$$W_{TA,3} + W_{TB,3} \leq 1$$

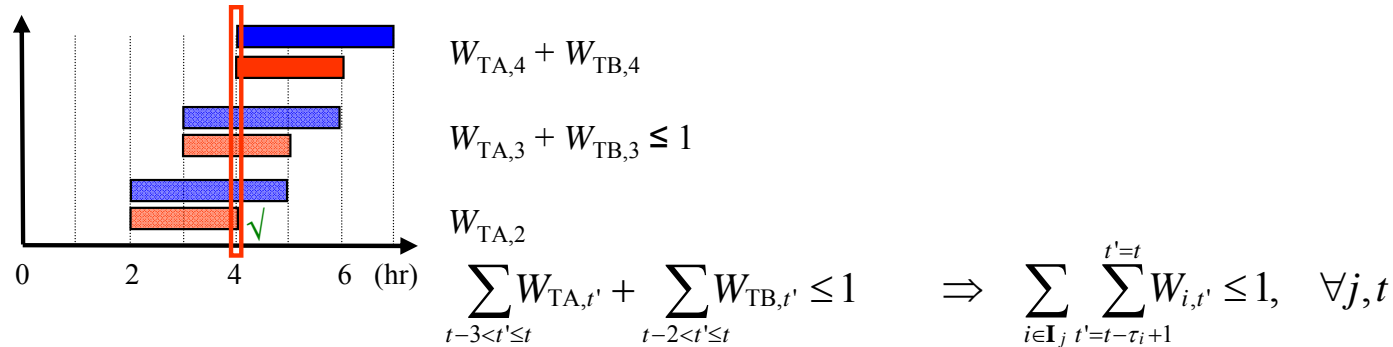
$$W_{TA,2}$$

$$\sum_{t-3 < t' \leq t} W_{TA,t'} + \sum_{t-2 < t' \leq t} W_{TB,t'} \leq 1$$

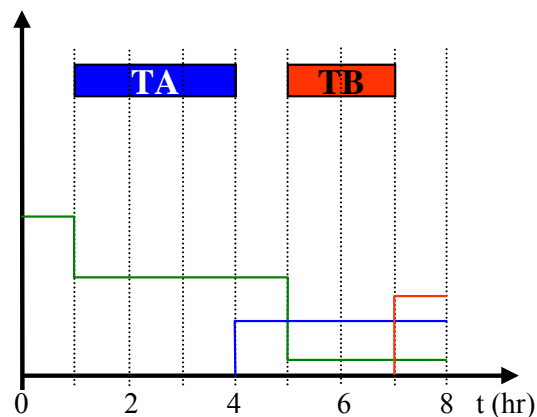
$$\Rightarrow \sum_{i \in \mathbf{I}_j} \sum_{t'=t-\tau_i+1}^{t'=t} W_{i,t'} \leq 1, \quad \forall j, t$$

# Mixed-integer Programming Model

- Resource constraints
- Decision variable:  $W_{it} = 1$  when a task starts

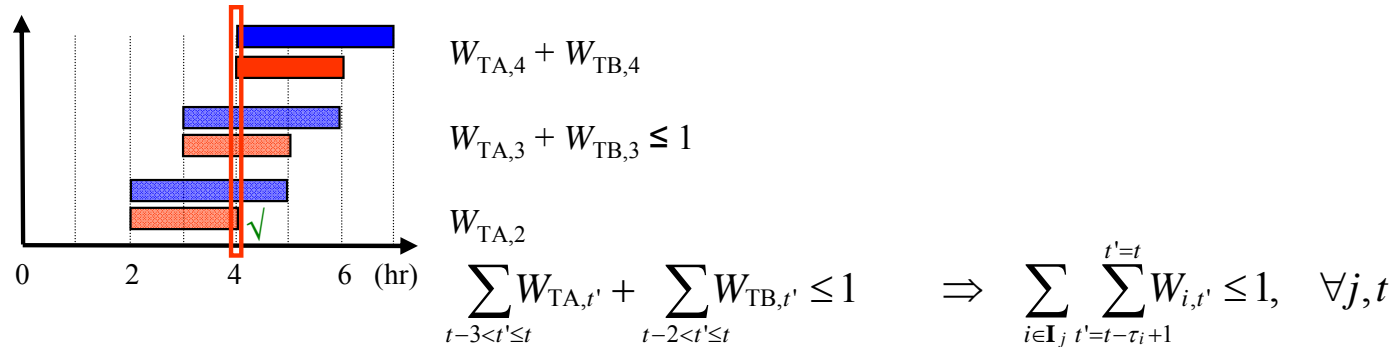


- Material balance constraints
- Inventory levels ( $S_{mt}$ ) are the results of our scheduling decisions  $W_{it}$

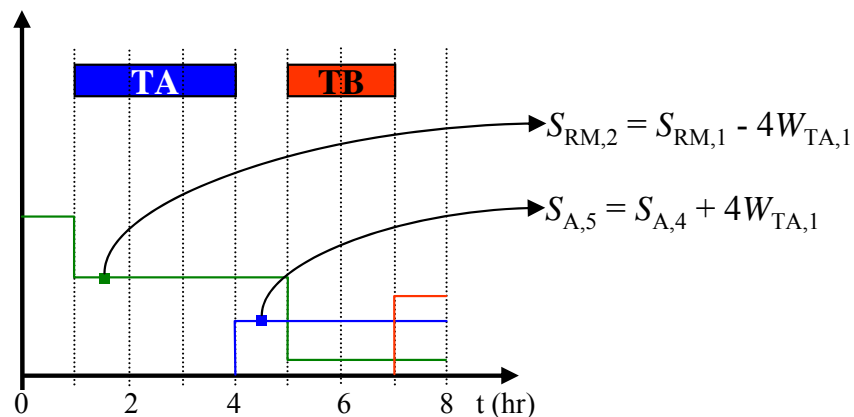


# Mixed-integer Programming Model

- Resource constraints
- Decision variable:  $W_{it} = 1$  when a task starts

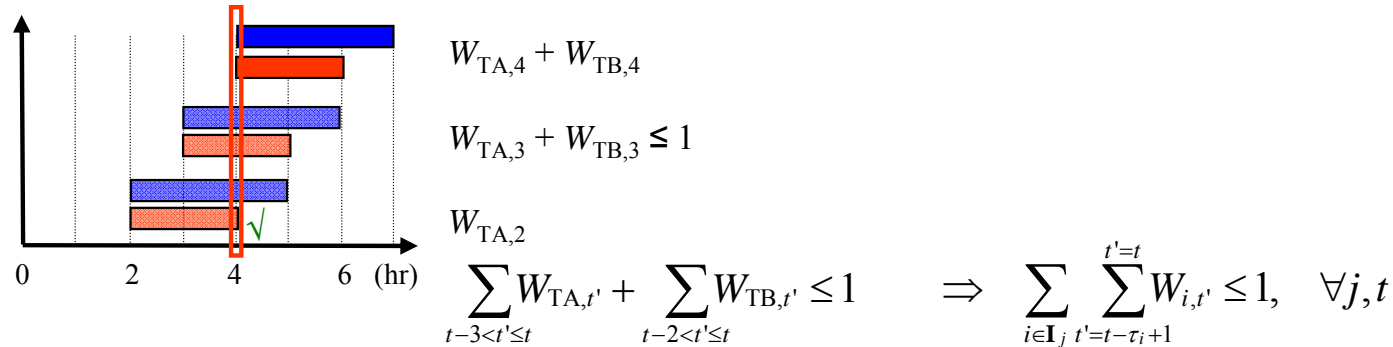


- Material balance constraints
- Inventory levels ( $S_{mt}$ ) are the results of our scheduling decisions  $W_{it}$

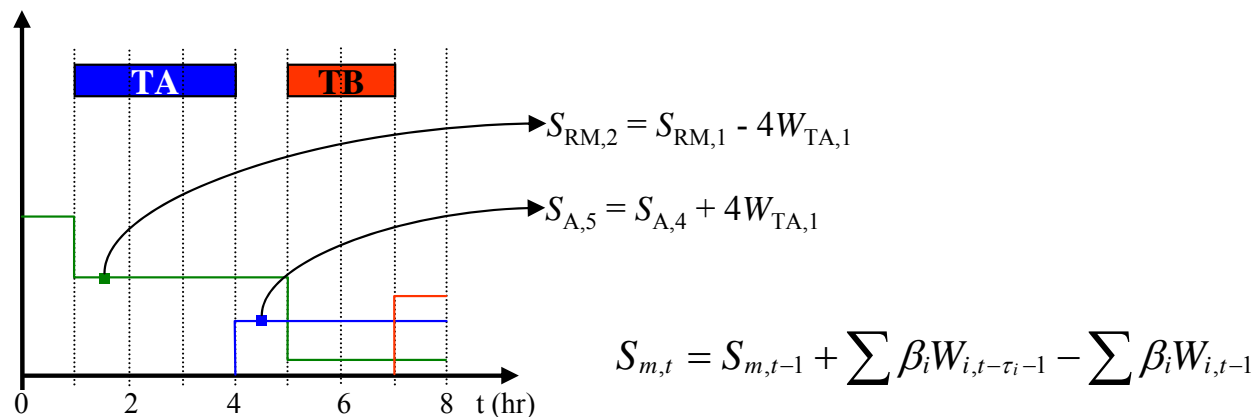


# Mixed-integer Programming Model

- Resource constraints
- Decision variable:  $W_{it} = 1$  when a task starts

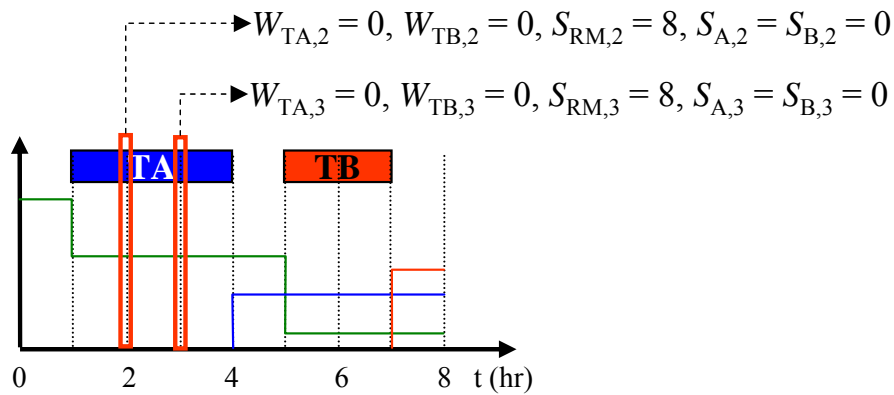


- Material balance constraints
- Inventory levels ( $S_{mt}$ ) are the results of our scheduling decisions  $W_{it}$



# Scheduling State-space Model

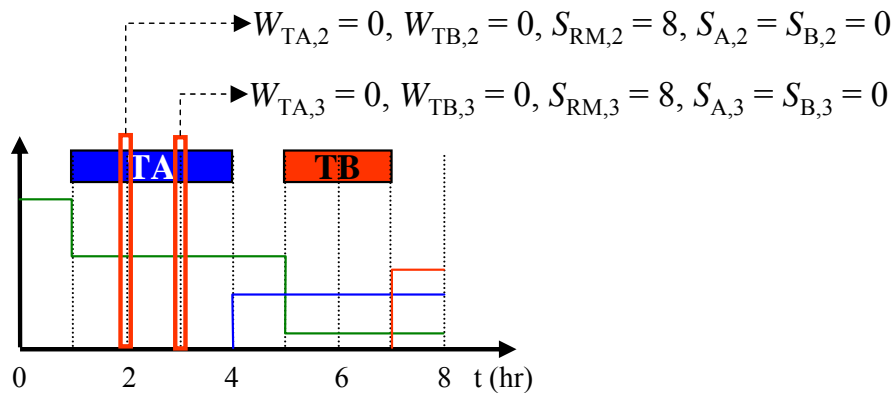
- Inputs: task start,  $W_{it}$
- States: material inventory level,  $S_{mt}$



The state of the system is not fully described

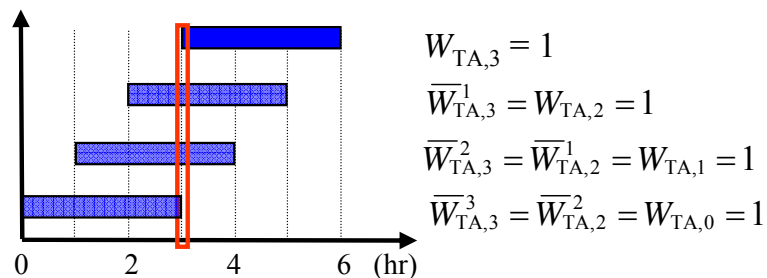
# Scheduling State-space Model

- Inputs: task start,  $W_{it}$
- States: material inventory level,  $S_{mt}$



The state of the system is not fully described

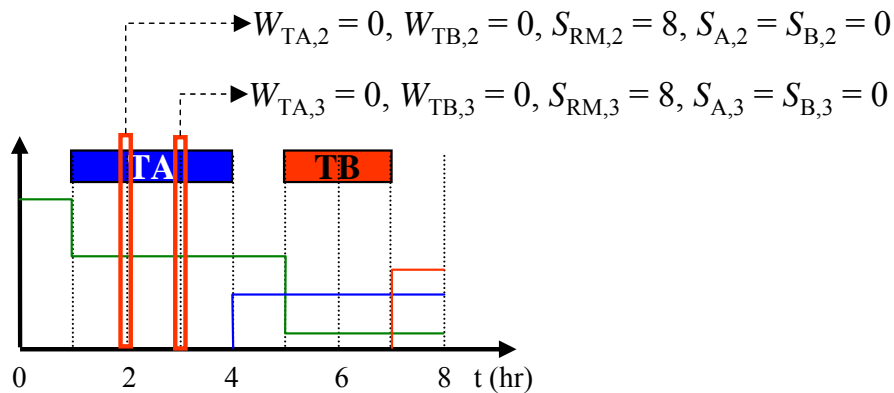
- Lifting of task variables



$$\bar{W}_{i,t+1}^n = \bar{W}_{i,t}^{n-1}, \quad \text{with} \quad \bar{W}_{it}^0 = W_{it}$$

# Scheduling State-space Model

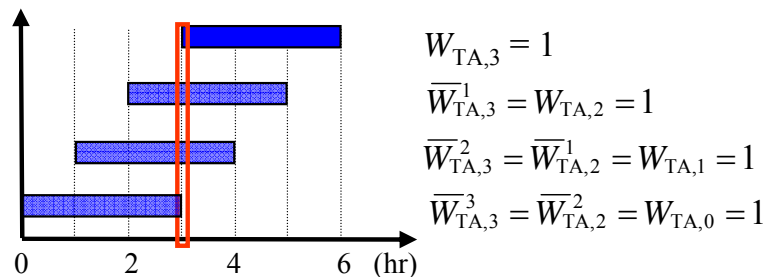
- Inputs: task start,  $W_{it}$
- States: material inventory level,  $S_{mt}$



$$\bar{W}_{TA,2}^1 = 1, \bar{W}_{TA,2}^2 = 0, \bar{W}_{TA,2}^3 = 0$$

$$\bar{W}_{TA,3}^1 = 0, \bar{W}_{TA,3}^2 = 1, \bar{W}_{TA,3}^3 = 0$$

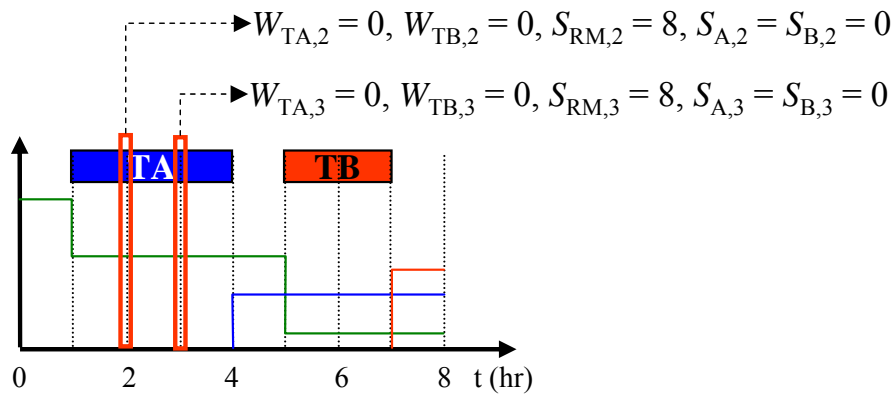
- Lifting of task variables



$$\bar{W}_{i,t+1}^n = \bar{W}_{i,t}^{n-1}, \quad \text{with} \quad \bar{W}_{it}^0 = W_{it}$$

# Scheduling State-space Model

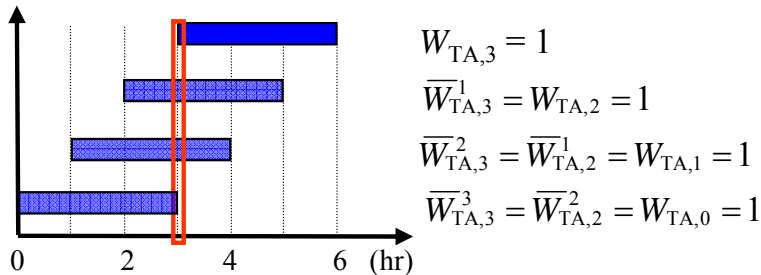
- Inputs: task start,  $W_{it}$
- States: material inventory level,  $S_{mt}$



$$\bar{W}_{TA,2}^1 = 1, \bar{W}_{TA,2}^2 = 0, \bar{W}_{TA,2}^3 = 0$$

$$\bar{W}_{TA,3}^1 = 0, \bar{W}_{TA,3}^2 = 1, \bar{W}_{TA,3}^3 = 0$$

- Lifting of task variables



$$\bar{W}_{i,t+1}^n = \bar{W}_{i,t}^{n-1}, \quad \text{with } \bar{W}_{it}^0 = W_{it}$$

- Material balances:

$$S_{m,t+1} = S_{m,t} + \sum \beta_i W_{i,t-\tau_i} - \sum \beta_i W_{i,t}$$

$$S_{m,t+1} = S_{m,t} + \sum \beta_i \bar{W}_{it}^{\tau_i} - \sum \beta_i W_{i,t}, \quad \forall m, t$$

- Resource constraints:

$$\sum_i \sum_{t'=t-\tau_i+1}^{t'} W_{i,t'} \leq 1$$

$$\sum_i W_{i,t} + \sum_i \sum_{n=t-\tau_i}^{n=1} \bar{W}_{i,t}^n \leq 1, \quad \forall j, t$$

# Scheduling State-space Model

---

- Inputs: task start  $u = [W_i]$
- States: inventory and task-status  $x = [S_m, \bar{W}_i^n]$

# Scheduling State-space Model

- Inputs: task start  $u = [W_i]$
- States: inventory and task-status  $x = [S_m, \bar{W}_i^n]$
- Dynamic model: task-status  $\bar{W}_i^n(t+1) = \bar{W}_i^{n-1}(t)$   
inventory  $S_m(t+1) = S_m(t) + \sum \beta_i \bar{W}_i^{\tau_i}(t) - \sum \beta_i W_i(t)$

# Scheduling State-space Model

- Inputs: task start  $u = [W_i]$
- States: inventory and task-status  $x = [S_m, \bar{W}_i^n]$
- Dynamic model: task-status  $\bar{W}_i^n(t+1) = \bar{W}_i^{n-1}(t)$   
inventory  $S_m(t+1) = S_m(t) + \sum \beta_i \bar{W}_i^{\tau_i}(t) - \sum \beta_i W_i(t)$

$$x(t+1) = Ax(t) + Bu(t)$$

$$\begin{bmatrix} \bar{W}_{TA}^1 \\ \bar{W}_{TA}^2 \\ \bar{W}_{TA}^3 \\ \bar{W}_{TB}^1 \\ \bar{W}_{TB}^2 \\ S_{RM} \\ S_A \\ S_B \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & 1 & & & & & & \\ & & & 1 & & & & & \\ & & & & \beta_{TA} & & & & \\ & & & & & \beta_{TB} & & & \\ & & & & & & 1 & & \\ & & & & & & & 1 & \\ & & & & & & & & 1 \end{bmatrix} \begin{bmatrix} \bar{W}_{TA}^1 \\ \bar{W}_{TA}^2 \\ \bar{W}_{TA}^3 \\ \bar{W}_{TB}^1 \\ \bar{W}_{TB}^2 \\ S_{RM} \\ S_A \\ S_B \end{bmatrix}_k + \begin{bmatrix} 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & 1 & & & & & & \\ & & & 1 & & & & & \\ & & & & -\beta_{TA} & & & & \\ & & & & & -\beta_{TB} & & & \\ & & & & & & 1 & & \\ & & & & & & & 1 & \\ & & & & & & & & 1 \end{bmatrix} \begin{bmatrix} W_{TA} \\ W_{TB} \end{bmatrix}_k$$

# Scheduling State-space Model

- Inputs: task start

$$u = [W_i]$$

- States: inventory and task-status

$$x = [S_m, \bar{W}_i^n]$$

- Dynamic model: task-status  
inventory

$$\bar{W}_i^n(t+1) = \bar{W}_i^{n-1}(t)$$

$$S_m(t+1) = S_m(t) + \sum \beta_i \bar{W}_i^{\tau_i}(t) - \sum \beta_i W_i(t)$$

$$x(t+1) = Ax(t) + Bu(t)$$

$$\begin{bmatrix} \bar{W}_{TA}^1 \\ \bar{W}_{TA}^2 \\ \bar{W}_{TA}^3 \\ \bar{W}_{TB}^1 \\ \bar{W}_{TB}^2 \\ S_{RM} \\ S_A \\ S_B \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & 1 & & & & & & \\ & & & 1 & & & & & \\ & & & & 1 & & & & \\ & & & & & \beta_{TA} & & & \\ & & & & & & \beta_{TB} & & \\ & & & & & & & 1 & \\ & & & & & & & & 1 \end{bmatrix} \begin{bmatrix} \bar{W}_{TA}^1 \\ \bar{W}_{TA}^2 \\ \bar{W}_{TA}^3 \\ \bar{W}_{TB}^1 \\ \bar{W}_{TB}^2 \\ S_{RM} \\ S_A \\ S_B \end{bmatrix}_k + \begin{bmatrix} 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & 1 & & & & & & \\ & & & 1 & & & & & \\ & & & & 1 & & & & \\ & & & & & -\beta_{TA} & & & \\ & & & & & & -\beta_{TB} & & \\ & & & & & & & 1 & \\ & & & & & & & & 1 \end{bmatrix} \begin{bmatrix} W_{TA} \\ W_{TB} \end{bmatrix}_k$$

- (Resource) constraints

$$\sum_i W_i(t) + \sum_i \sum_{n=t-\tau_i}^{n=1} \bar{W}_i^n(t) \leq 1$$

$$Ex(t) + Fu(t) \leq b$$

$$\begin{bmatrix} 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & 1 & & & & & & \\ & & & 1 & & & & & \\ & & & & 1 & & & & \\ & & & & & 1 & & & \\ & & & & & & 1 & & \\ & & & & & & & 1 & \\ & & & & & & & & 1 \end{bmatrix} \begin{bmatrix} W_{TA} \\ W_{TB} \end{bmatrix}_k + \begin{bmatrix} 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & 1 & & & & & & \\ & & & 1 & & & & & \\ & & & & 1 & & & & \\ & & & & & 1 & & & \\ & & & & & & 1 & & \\ & & & & & & & 1 & \\ & & & & & & & & 1 \end{bmatrix} \begin{bmatrix} \bar{W}_{TA}^1 \\ \bar{W}_{TA}^2 \\ \bar{W}_{TB}^1 \end{bmatrix}_k \leq [1]$$

# Disturbances

---

- Dynamic model with disturbances:  $x(t+1) = Ax(t) + Bu(t) + B_d d(t)$

# Disturbances

■ Dynamic model with disturbances:  $x(t+1) = Ax(t) + Bu(t) + B_d d(t)$

■ Deliveries and orders

- $D_{mt}$ : net delivery of material  $m$  during period  $t$

$$S_{m,t+1} = S_{m,t} + \sum \beta_i \bar{W}_{it}^{\tau_i} - \sum \beta_i W_{i,t} + D_{mt}$$

$$S_m(t+1) = S_m(t) + \sum \beta_i \bar{W}_i^{\tau_i}(t) - \sum \beta_i W_i(t) + D_m(t)$$

# Disturbances

- Dynamic model with disturbances:  $x(t+1) = Ax(t) + Bu(t) + B_d d(t)$

- Deliveries and orders

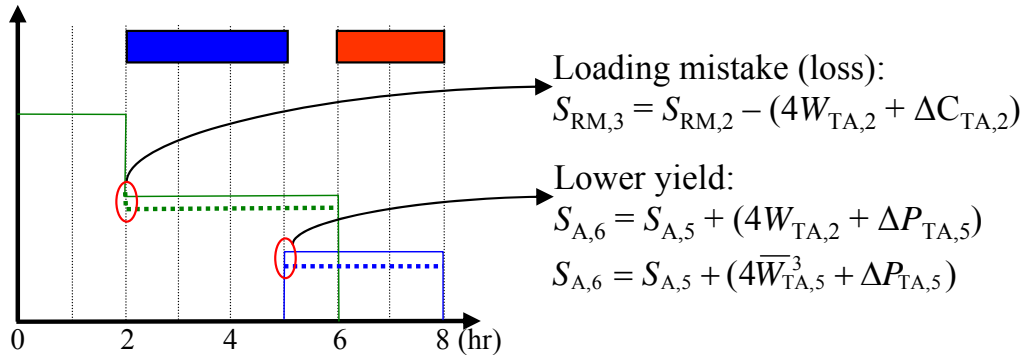
- $D_{mt}$ : net delivery of material  $m$  during period  $t$

$$S_{m,t+1} = S_{m,t} + \sum \beta_i \bar{W}_{it}^{\tau_i} - \sum \beta_i W_{i,t} + D_{mt}$$

$$S_m(t+1) = S_m(t) + \sum \beta_i \bar{W}_i^{\tau_i}(t) - \sum \beta_i W_i(t) + D_m(t)$$

- Production disturbances

- $\Delta C_{it}$ : consumption disturbance for task  $i$  starting at  $t$
  - $\Delta P_{it}$ : production disturbance for task finishing at  $t$



# Disturbances

- Dynamic model with disturbances:  $x(t+1) = Ax(t) + Bu(t) + B_d d(t)$

- Deliveries and orders

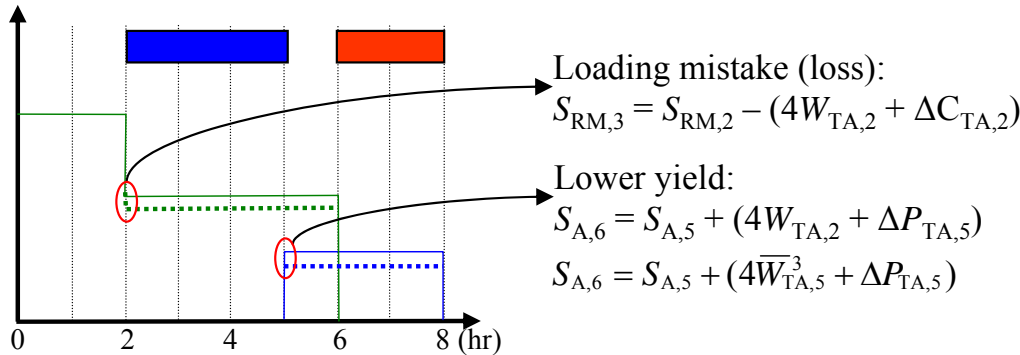
- $D_{mt}$ : net delivery of material  $m$  during period  $t$

$$S_{m,t+1} = S_{m,t} + \sum \beta_i \bar{W}_{it}^{\tau_i} - \sum \beta_i W_{i,t} + D_{mt}$$

$$S_m(t+1) = S_m(t) + \sum \beta_i \bar{W}_i^{\tau_i}(t) - \sum \beta_i W_i(t) + D_m(t)$$

- Production disturbances

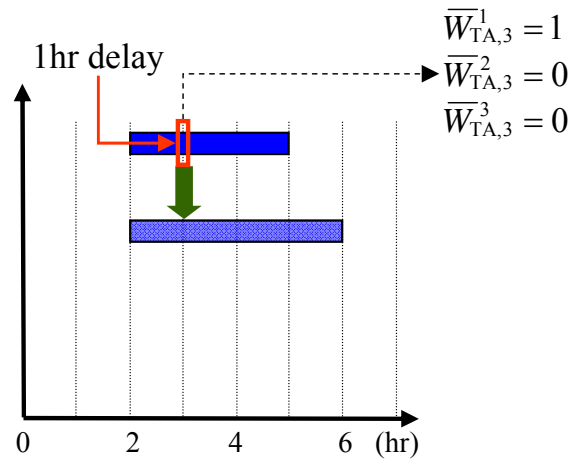
- $\Delta C_{it}$ : consumption disturbance for task  $i$  starting at  $t$
  - $\Delta P_{it}$ : production disturbance for task finishing at  $t$



$$S_{m,t+1} = S_{m,t} + \sum (\beta_i \bar{W}_{it}^{\tau_i} + \Delta P_{it}) - \sum (\beta_i W_{it} + \Delta C_{it}) + D_{m,t}$$

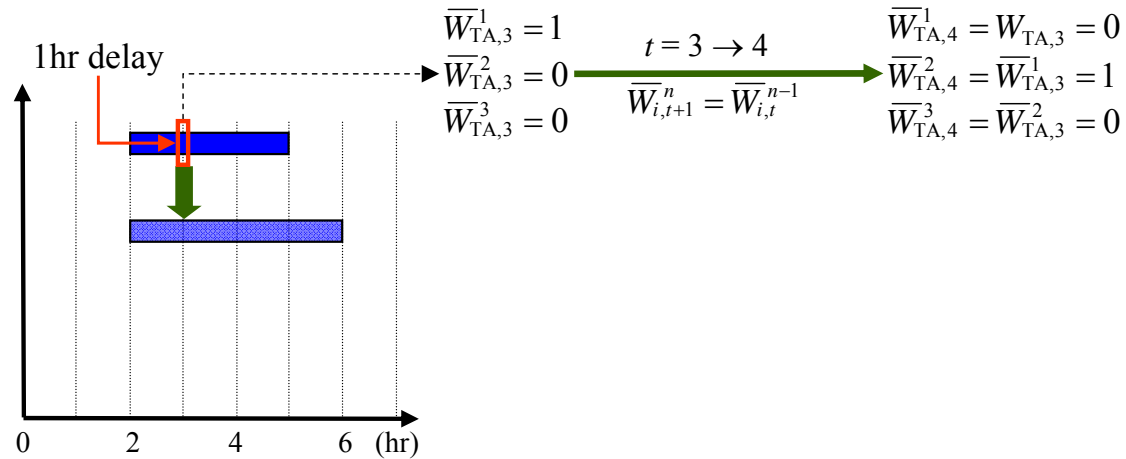
# Disturbances

- Dynamic model with disturbances
- Delays
  - $Y_{it}^n$ : 1 hr delay of task  $i$  after running for  $n$  periods



# Disturbances

- Dynamic model with disturbances
- Delays
  - $Y_{it}^n$ : 1 hr delay of task  $i$  after running for  $n$  periods

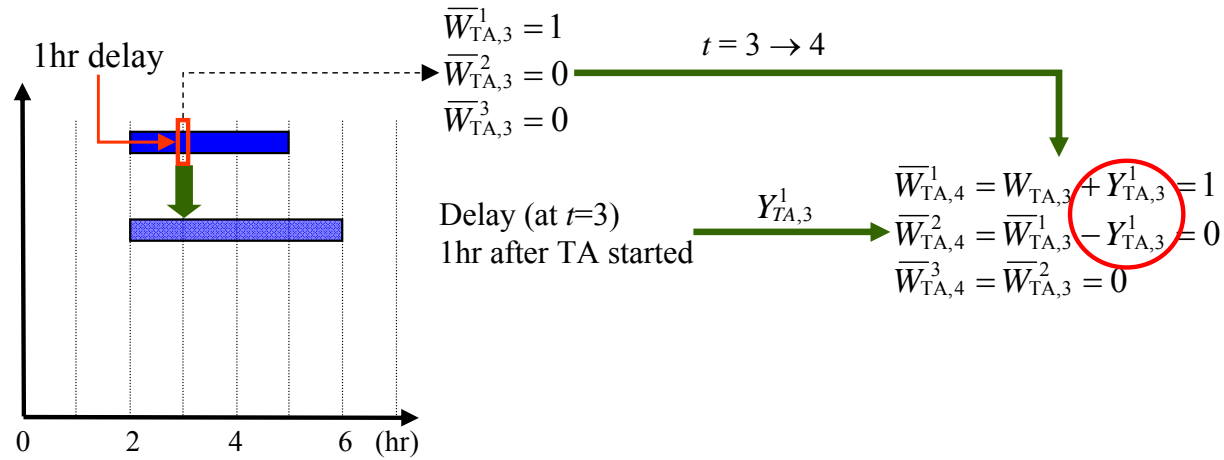


# Disturbances

- Dynamic model with disturbances

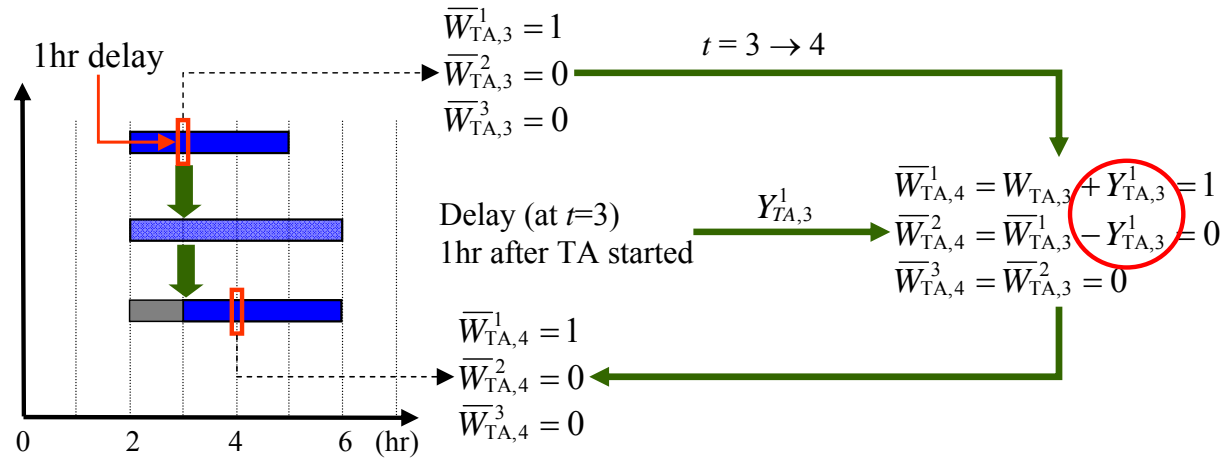
- Delays

- $Y_{it}^n$ : 1 hr delay of task  $i$  after running for  $n$  periods



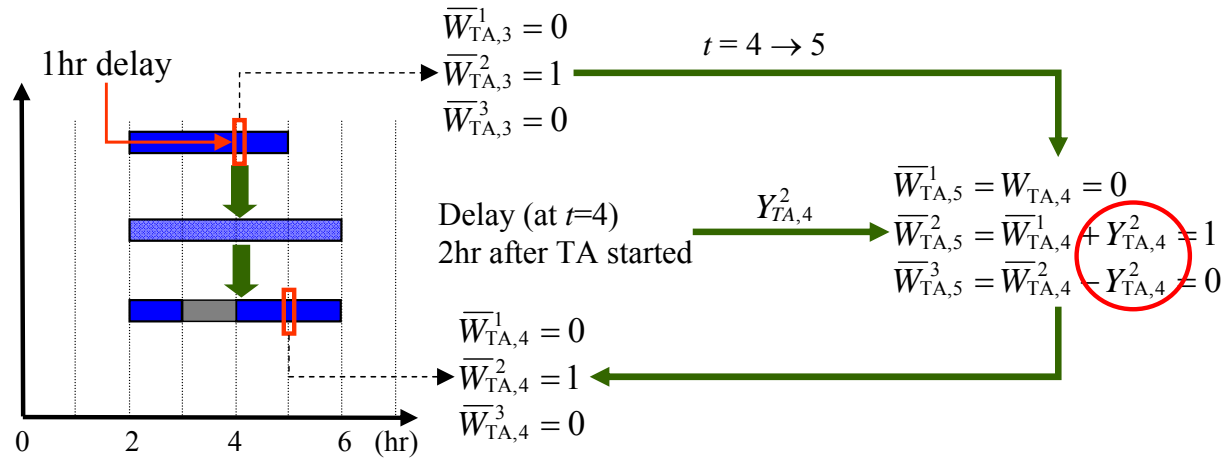
# Disturbances

- Dynamic model with disturbances
- Delays
  - $Y_{it}^n$ : 1 hr delay of task  $i$  after running for  $n$  periods



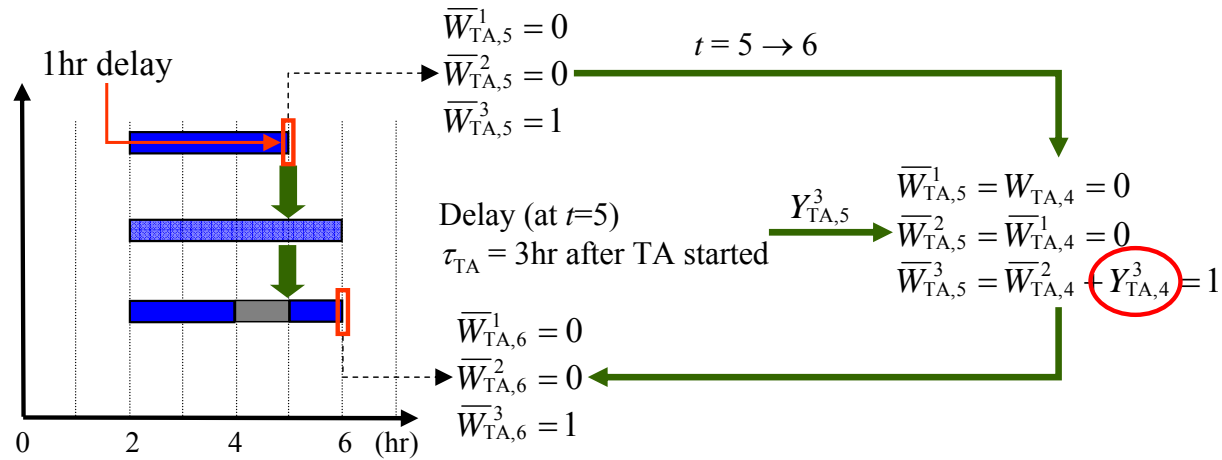
# Disturbances

- Dynamic model with disturbances
- Delays
  - $Y_{it}^n$ : 1 hr delay of task  $i$  after running for  $n$  periods



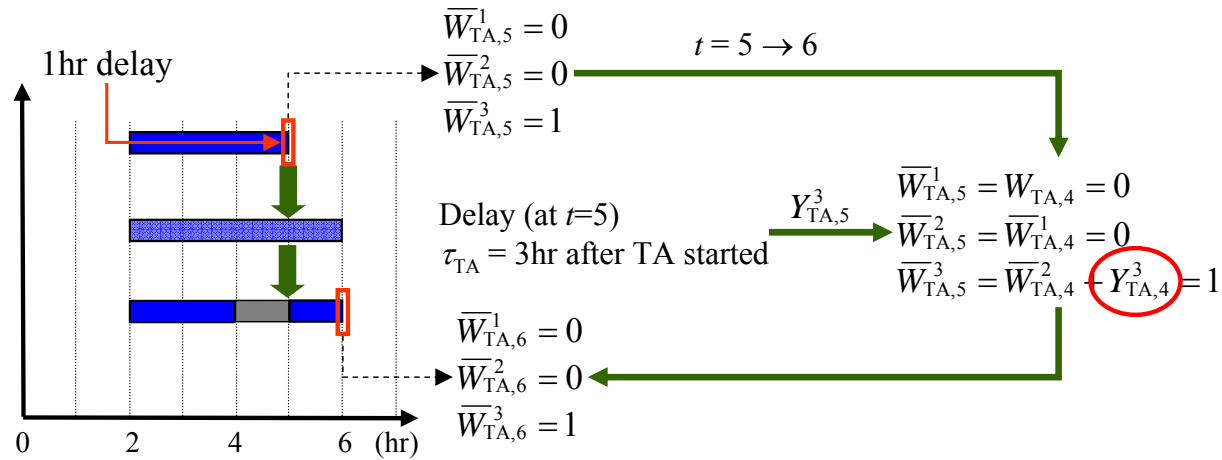
# Disturbances

- Dynamic model with disturbances
- Delays
  - $Y_{it}^n$ : 1 hr delay of task  $i$  after running for  $n$  periods



# Disturbances

- Dynamic model with disturbances
- Delays
  - $Y_{it}^n$ : 1 hr delay of task  $i$  after running for  $n$  periods



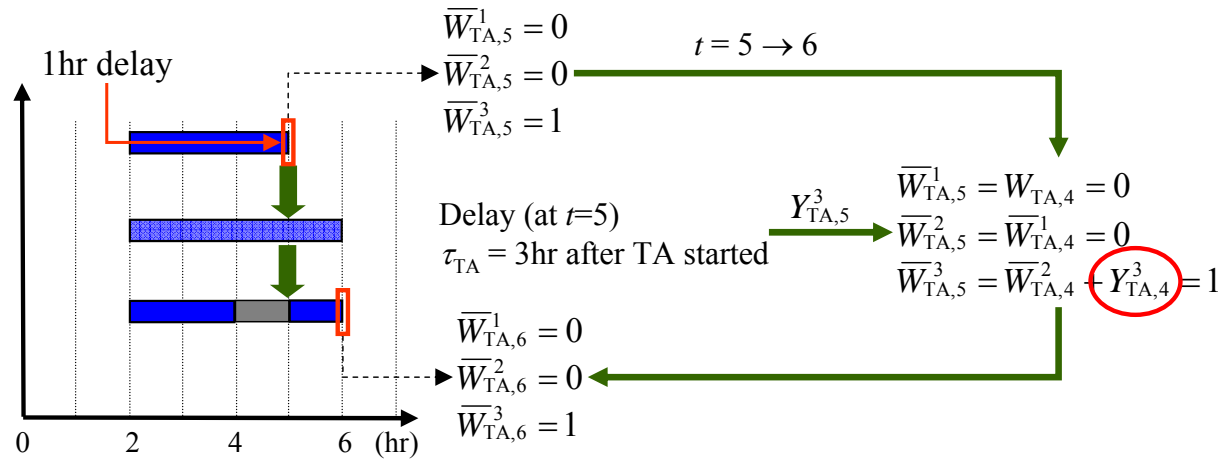
$$\bar{W}_{i,t+1}^n = \bar{W}_{i,t}^{n-1} + Y_{i,t}^n - Y_{i,t}^{n-1}$$

# Disturbances

- Dynamic model with disturbances

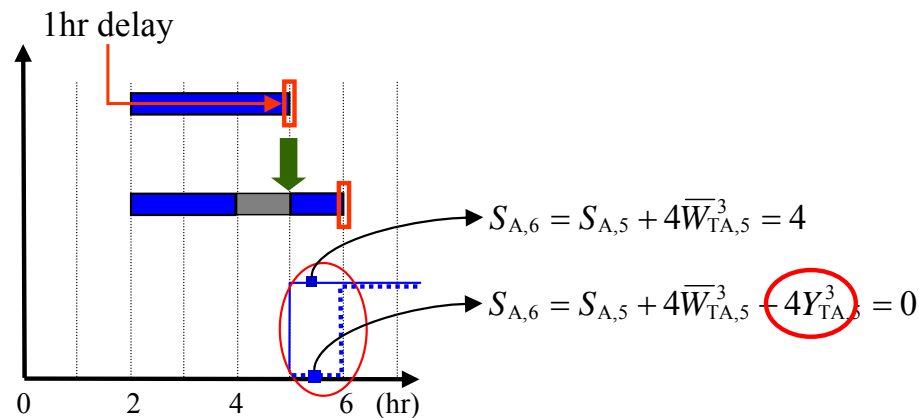
- Delays

- $Y_{it}^n$ : 1 hr delay of task  $i$  after running for  $n$  periods



$$\bar{W}_{i,t+1}^n = \bar{W}_{i,t}^{n-1} + Y_{i,t}^n - Y_{i,t}^{n-1}$$

- Inventory correction

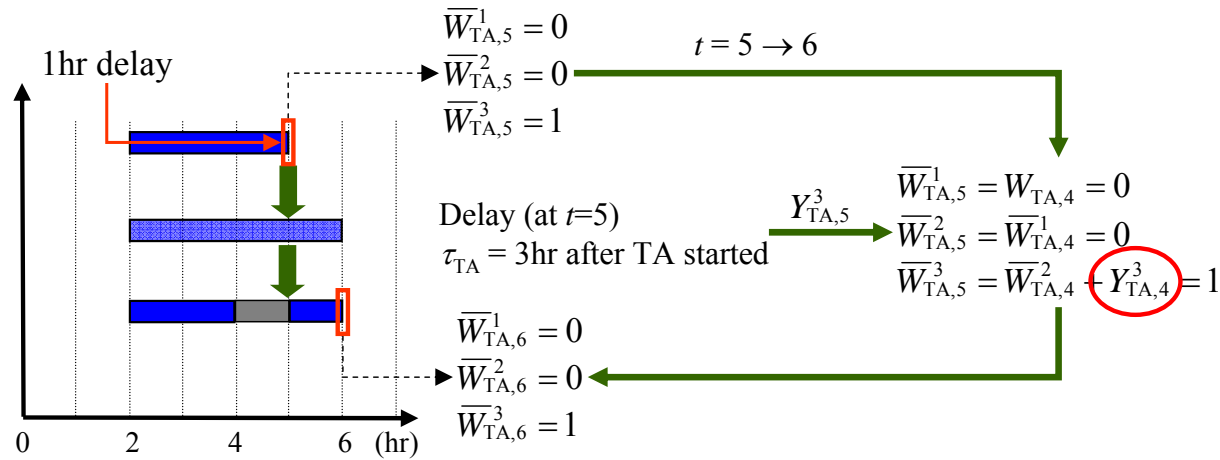


# Disturbances

- Dynamic model with disturbances

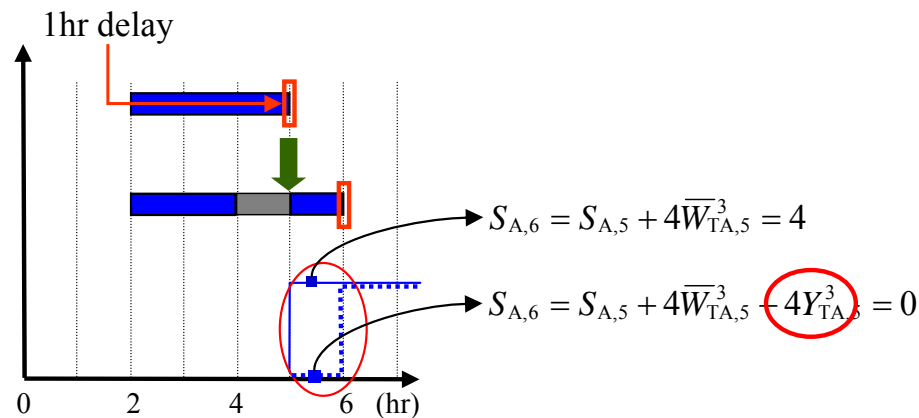
- Delays

- $Y_{it}^n$ : 1 hr delay of task  $i$  after running for  $n$  periods



$$\bar{W}_{i,t+1}^n = \bar{W}_{i,t}^{n-1} + Y_{i,t}^n - Y_{i,t}^{n-1}$$

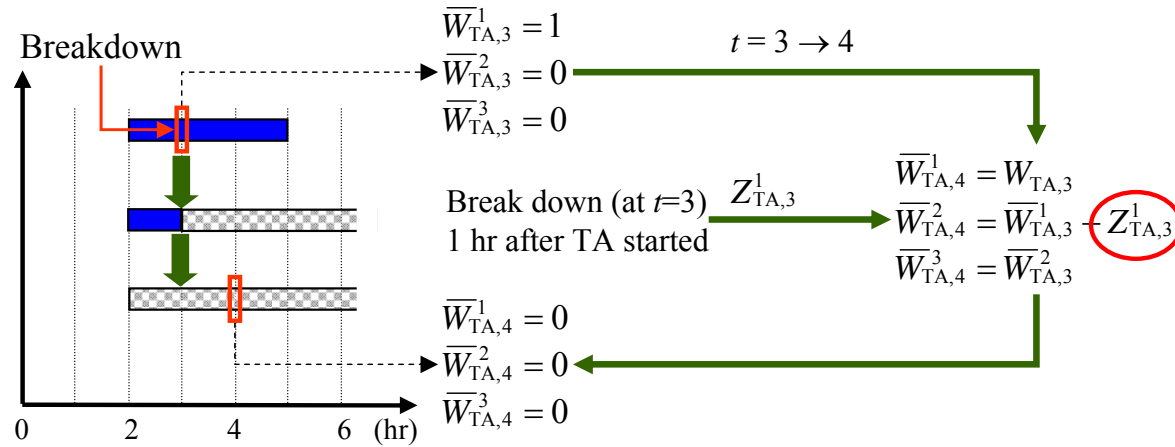
- Inventory correction



$$S_{m,t+1} = S_{m,t} + \sum \beta_i (\bar{W}_{it}^{\tau_i} - Y_{it}^{\tau_i}) - \sum \beta_i W_{i,t}$$

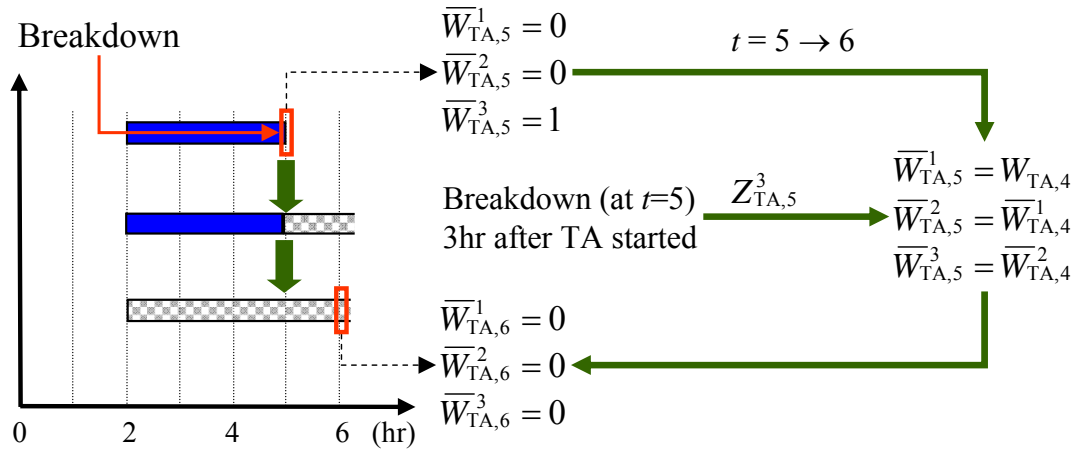
# Disturbances

- Dynamic model with disturbances
- Unit breakdowns
  - $Z_{it}^n$ : Unit breakdown after running task  $i$  for  $n$  periods



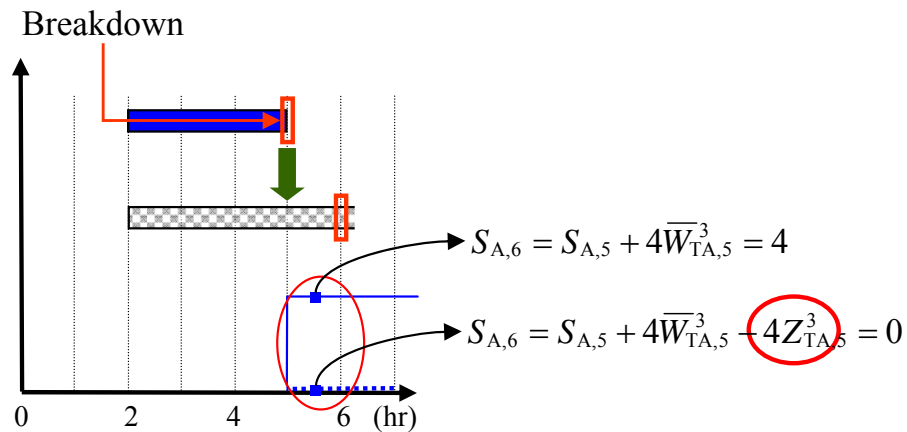
# Disturbances

- Dynamic model with disturbances
- Unit breakdowns
  - $Z_{it}^n$ : Unit breakdown after running task  $i$  for  $n$  periods



$$\bar{W}_{i,t+1}^n = \bar{W}_{i,t}^{n-1} - Z_{i,t}^{n-1} \quad n > 1$$

- Inventory correction



$$S_{m,t+1} = S_{m,t} + \sum \beta_i (\bar{W}_{it}^{r_i} - Z_{it}^{r_i}) - \sum \beta_i W_{i,t}$$

# Extensions

- Variable batchsizes,  $B_{it}$

- Inputs augmented with batchsizes:  $u = [W_i, B_i]$
- States augmented with batchsize-status:  $x = [S_m, \bar{W}_i^n, \bar{B}_i^n]$

- Dynamic model:  $\bar{W}_{i,t+1}^n = \bar{W}_{it}^{n-1}$

$$\bar{B}_{i,t+1}^n = \bar{B}_{it}^{n-1}$$

$$S_{m,t+1} = S_{mt} + \sum \bar{B}_{it}^{\tau_i} + \sum B_{it}$$

- Constraints:  $\sum_i W_{it} + \sum_i \sum_n \bar{W}_{it}^n \leq 1$

$$\beta_i^{\min} W_{it} - B_{it} \leq 0, \quad -\beta_i^{\max} W_{it} + B_{it} \leq 0$$

# Extensions

- Variable batchsizes,  $B_{it}$

- Inputs augmented with batchsizes:  $u = [W_i, B_i]$
- States augmented with batchsize-status:  $x = [S_m, \bar{W}_i^n, \bar{B}_i^n]$
- Dynamic model:  $\bar{W}_{i,t+1}^n = \bar{W}_{it}^{n-1}$   
 $\bar{B}_{i,t+1}^n = \bar{B}_{it}^{n-1}$   
 $S_{m,t+1} = S_{mt} + \sum \bar{B}_{it}^{\tau_i} + \sum B_{it}$
- Constraints:  $\sum_i W_{it} + \sum_i \sum_n \bar{W}_{it}^n \leq 1$   
 $\beta_i^{\min} W_{it} - B_{it} \leq 0, \quad -\beta_i^{\max} W_{it} + B_{it} \leq 0$

- Additional variables (e.g., flows, setups) and constraints (e.g., connectivity) handled similarly

# Extensions

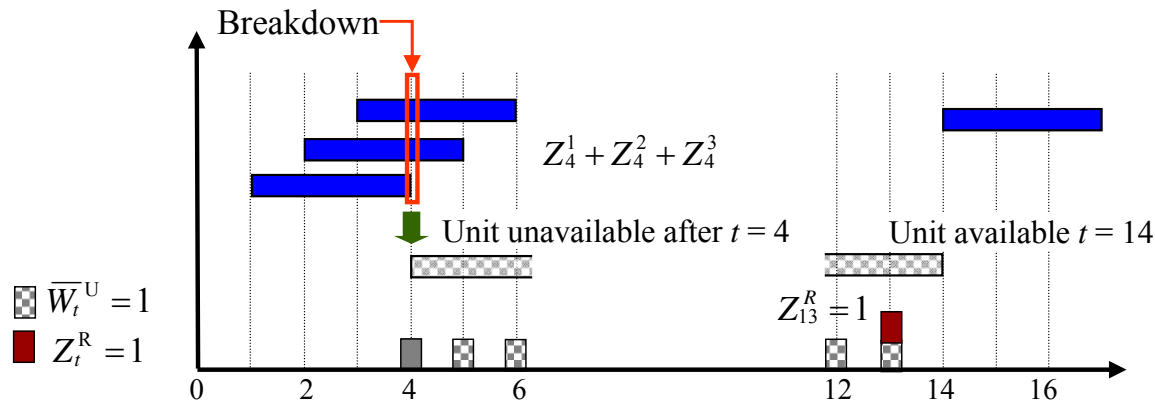
- Variable batchsizes,  $B_{it}$

- Inputs augmented with batchsizes:  $u = [W_i, B_i]$
- States augmented with batchsize-status:  $x = [S_m, \bar{W}_i^n, \bar{B}_i^n]$
- Dynamic model:  $\bar{W}_{i,t+1}^n = \bar{W}_{it}^{n-1}$   
 $\bar{B}_{i,t+1}^n = \bar{B}_{it}^{n-1}$   
 $S_{m,t+1} = S_{mt} + \sum \bar{B}_{it}^{\tau_i} + \sum B_{it}$
- Constraints:  $\sum_i W_{it} + \sum_i \sum_n \bar{W}_{it}^n \leq 1$   
 $\beta_i^{\min} W_{it} - B_{it} \leq 0, \quad -\beta_i^{\max} W_{it} + B_{it} \leq 0$

- Additional variables (e.g., flows, setups) and constraints (e.g., connectivity) handled similarly

- Processing unit availability

- Unit becomes unavailable after any breakdown: introduce unavailable state,  $\bar{W}_t^U$
- Unit becomes available after repair; introduce *repair* disturbance,  $Z_t^R$



$$\bar{W}_{t+1}^U = W_t^U + \sum_{n=1}^3 Z_t^n - Z_t^R$$

$$\sum W_{it} + \sum_i \sum_n \bar{W}_{it}^n + \sum_i \sum_n Z_{it}^n + \bar{W}_t^U \leq 1$$

# Conclusions

---

- Manufacturing node model for integration with inventory management
- Scheduling MIP model computationally more efficient

# Conclusions

---

- Manufacturing node model for integration with inventory management
- Scheduling MIP model computationally more efficient
- General state-space scheduling model
  - *MPC-friendly* scheduling model
  - Use existing <sup>1,2,3</sup> and develop new results for hybrid dynamical systems

<sup>1</sup> Bemporad & Morari (1999)

<sup>2</sup> Panek et al., (2008)

<sup>3</sup> Goebel et al., (2009)

# Conclusions

- Manufacturing node model for integration with inventory management
- Scheduling MIP model computationally more efficient
- General state-space scheduling model
  - *MPC-friendly* scheduling model
  - Use existing <sup>1,2,3</sup> and develop new results for hybrid dynamical systems
- *Standardize rescheduling*
  - Currently, unclear what rescheduling means
  - Keep the same state-space model; infer reformulation of the MIP model
  - Reverse transformation: state-space mode + disturbances → MIP model

<sup>1</sup> Bemporad & Morari (1999)

<sup>2</sup> Panek et al., (2008)

<sup>3</sup> Goebel et al., (2009)

# Conclusions

- Manufacturing node model for integration with inventory management
- Scheduling MIP model computationally more efficient
- General state-space scheduling model
  - *MPC-friendly* scheduling model
  - Use existing <sup>1,2,3</sup> and develop new results for hybrid dynamical systems
- *Standardize rescheduling*
  - Currently, unclear what rescheduling means
  - Keep the same state-space model; infer reformulation of the MIP model
  - Reverse transformation: state-space mode + disturbances → MIP model
- *What can we learn from process control?*
  - What is an optimal input/output trajectory? (Does it exist?)
  - Is stability relevant?
  - What are terminal constraints in scheduling?
  - ...

<sup>1</sup> Bemporad & Morari (1999)

<sup>2</sup> Panek et al., (2008)

<sup>3</sup> Goebel et al., (2009)

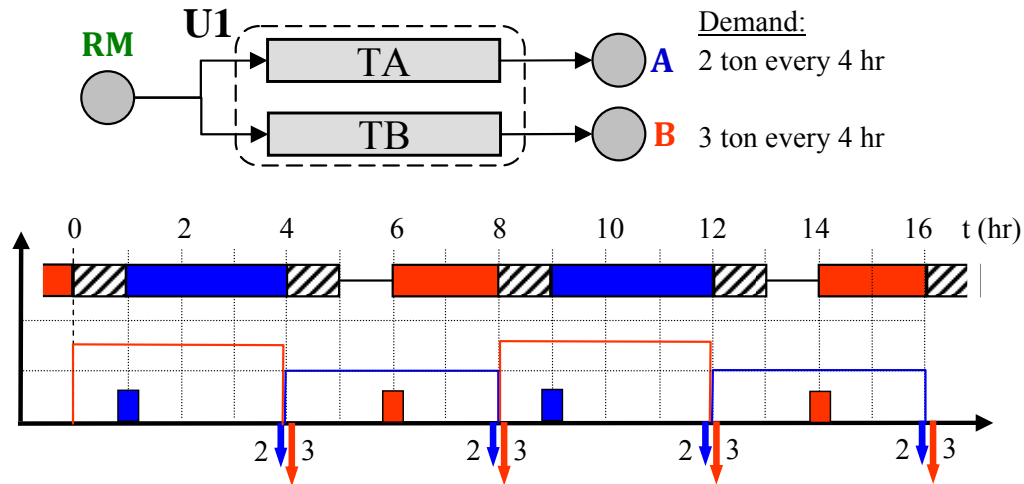
# Input & Output Trajectories

---

- Are there scheduling *set-points*?
  - Orders → inventories of products at certain time points (subset of states)

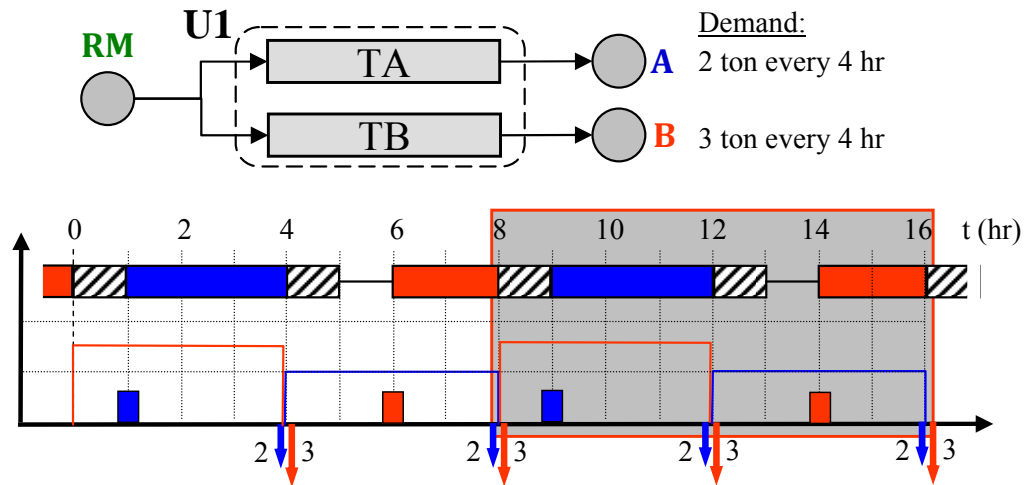
# Input & Output Trajectories

- Are there scheduling *set-points*?
  - Orders  $\rightarrow$  inventories of products at certain time points (subset of states)
- Are there optimal input/output set-point *trajectories*?



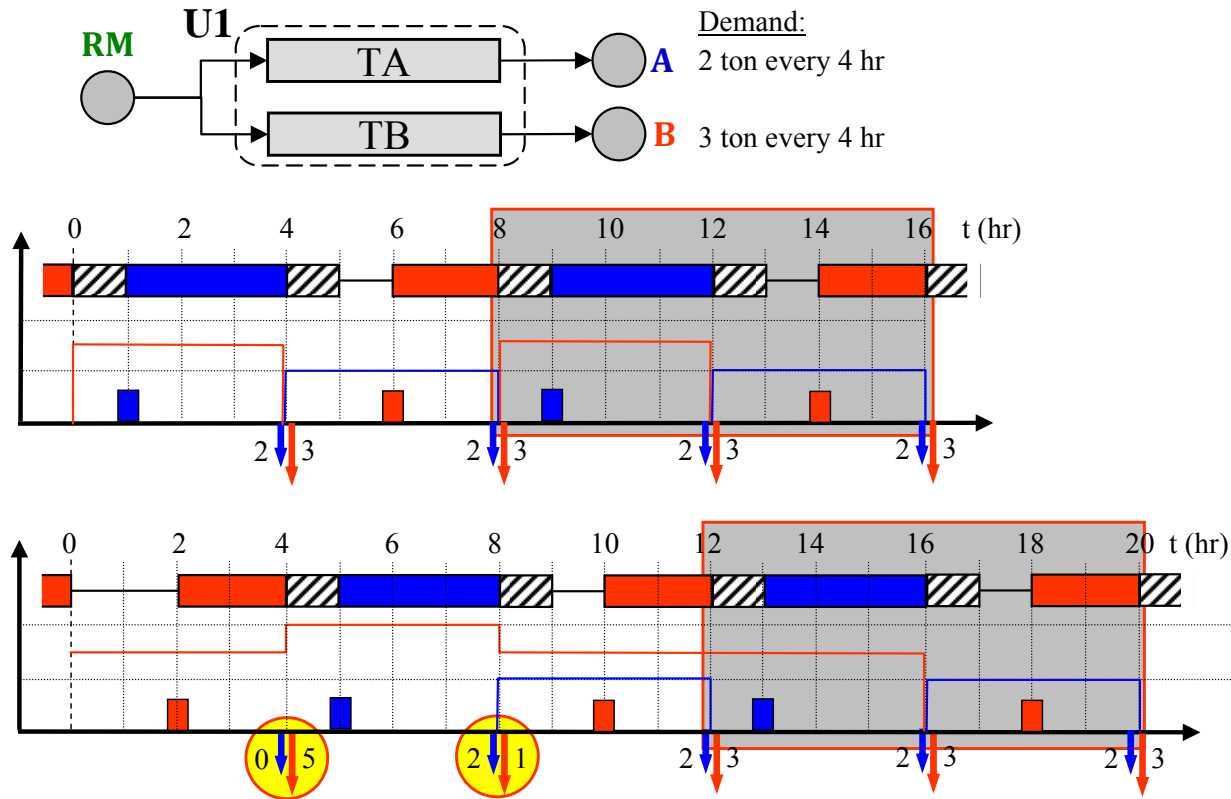
# Input & Output Trajectories

- Are there scheduling *set-points*?
  - Orders → inventories of products at certain time points (subset of states)
- Are there optimal input/output set-point *trajectories*?



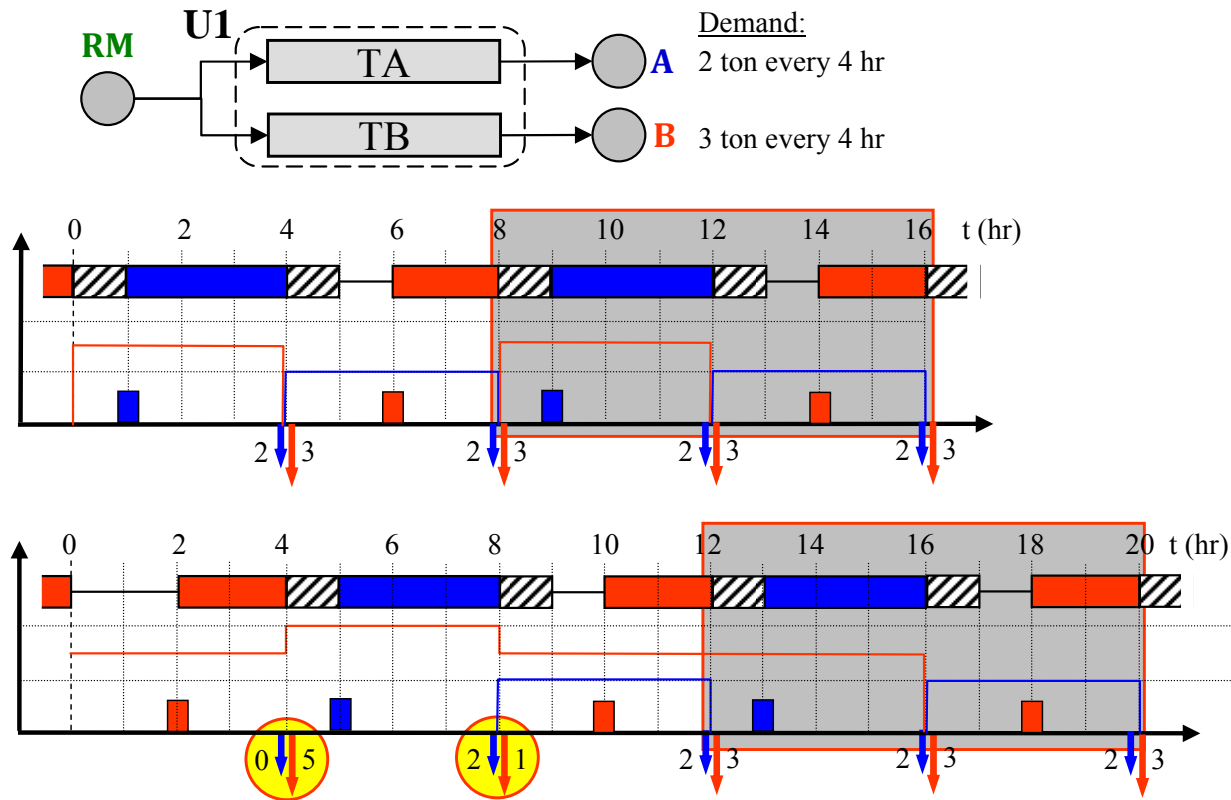
# Input & Output Trajectories

- Are there scheduling *set-points*?
  - Orders → inventories of products at certain time points (subset of states)
- Are there optimal input/output set-point *trajectories*?



# Input & Output Trajectories

- Are there scheduling *set-points*?
  - Orders → inventories of products at certain time points (subset of states)
- Are there optimal input/output set-point *trajectories*?

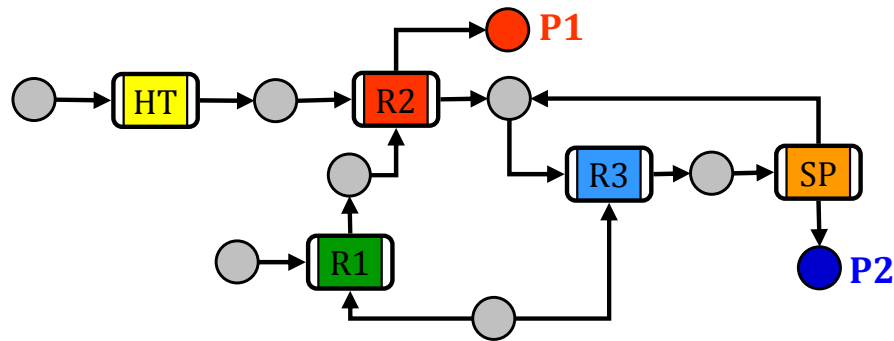


- Obvious solution:  $W_{it}^* = S_{mt}^* = 0$ 
  - This is what we typically do (**economic objective**)
- Can lead to closed-loop infeasible solutions or poor performance

# Instability

- Meet demand over 48-hr horizon at minimum cost
  - Solve scheduling MIP model in rolling horizon

(Kondili et al., 1993)

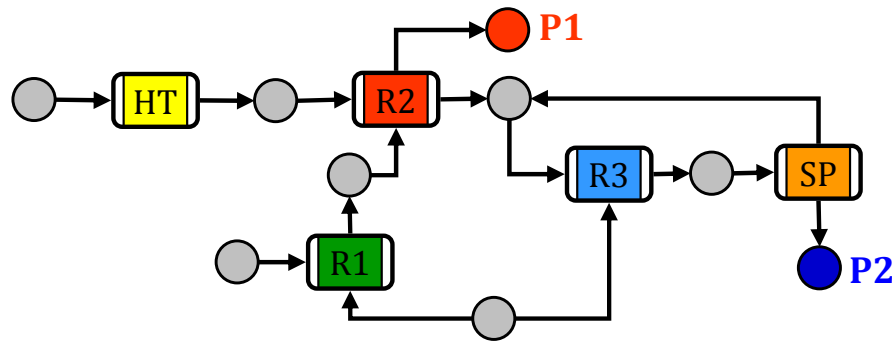


Demand  
P1 : 150 tons/day  
P2 : 450 tons/day

# Instability

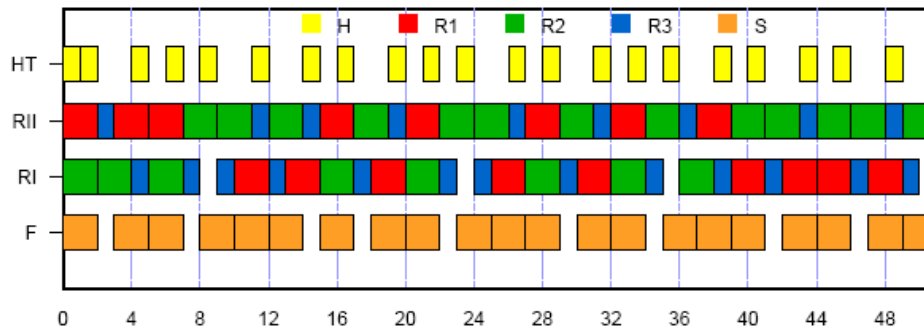
- Meet demand over 48-hr horizon at minimum cost
  - Solve scheduling MIP model in rolling horizon

(Kondili et al., 1993)



Demand  
P1 : 150 tons/day  
P2 : 450 tons/day

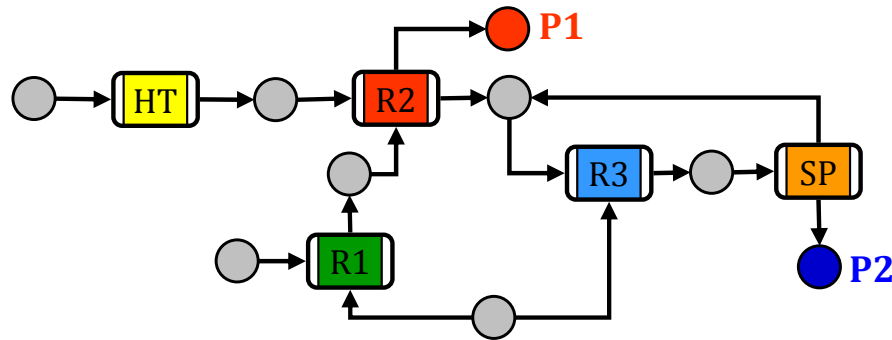
- Gantt chart of implemented solution (47 MIP runs)
  - All orders are satisfied perfectly



# Instability

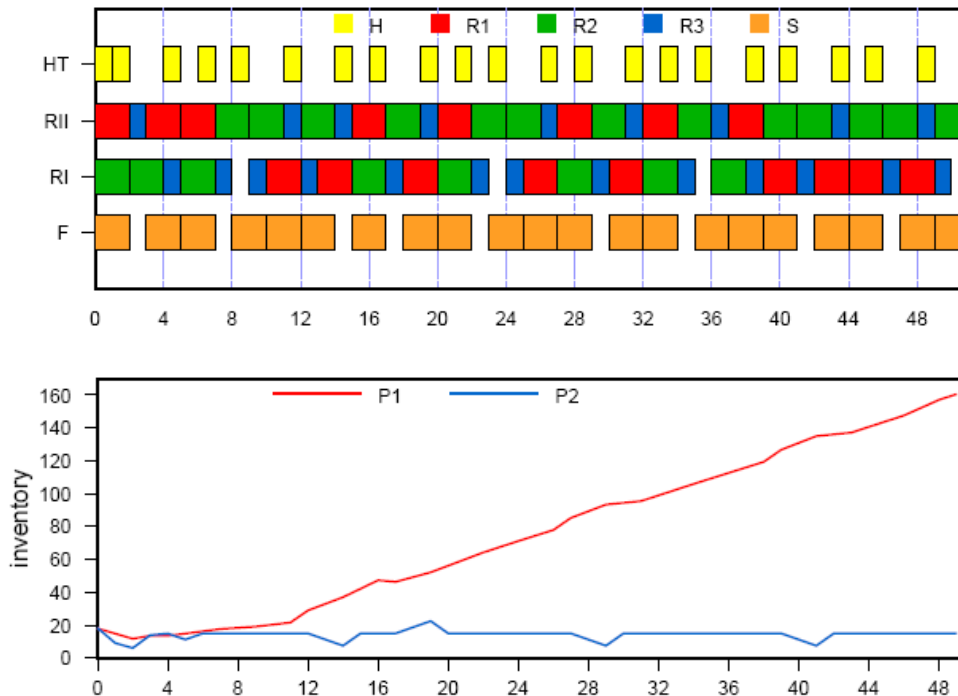
- Meet demand over 48-hr horizon at minimum cost
  - Solve scheduling MIP model in rolling horizon

(Kondili et al., 1993)



Demand  
P1 : 150 tons/day  
P2 : 450 tons/day

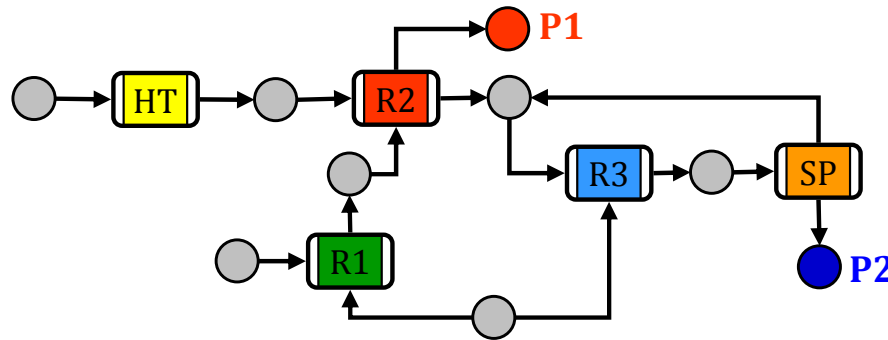
- Gantt chart of implemented solution (47 MIP runs)
  - All orders are satisfied perfectly



# Instability

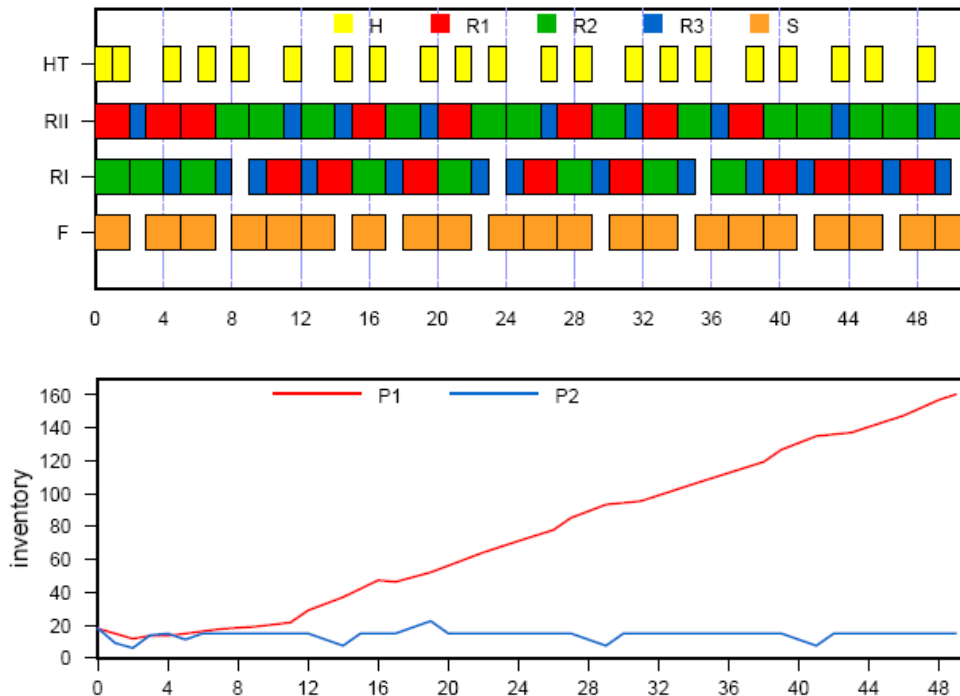
- Meet demand over 48-hr horizon at minimum cost
  - Solve scheduling MIP model in rolling horizon

(Kondili et al., 1993)

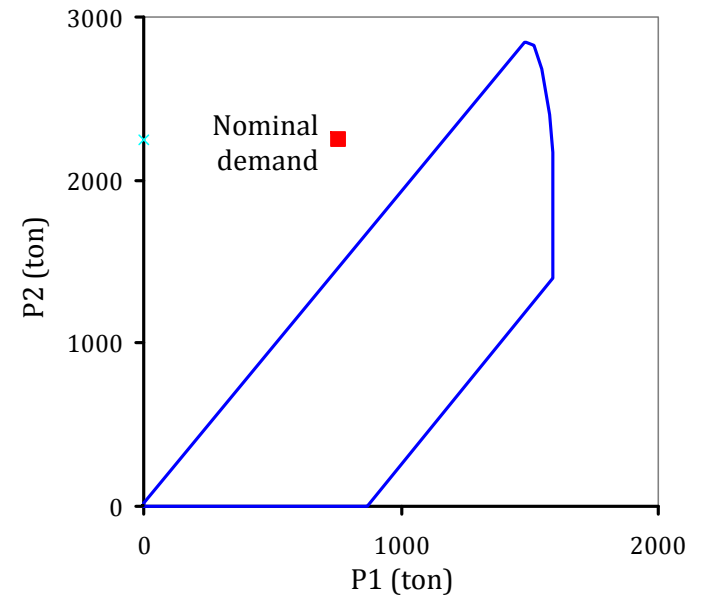


Demand  
 P1 : 150 tons/day  
 P2 : 450 tons/day

- Gantt chart of implemented solution (47 MIP runs)
  - All orders are satisfied perfectly



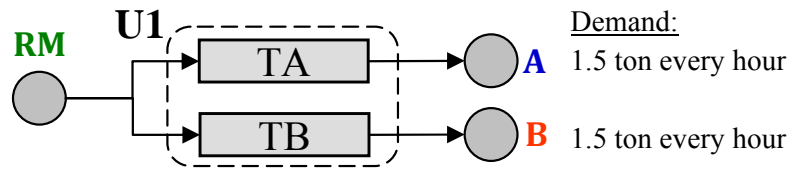
Projection of scheduling feasible region  
 (production targets for 5 days)



Sung & Maravelias (2007)

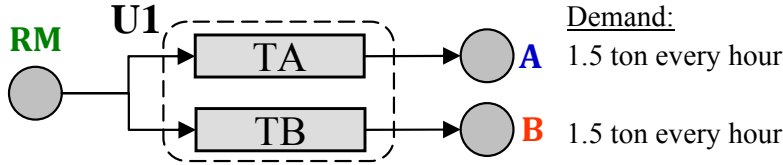
# Terminal Constraints

- Meet *nominal* demand

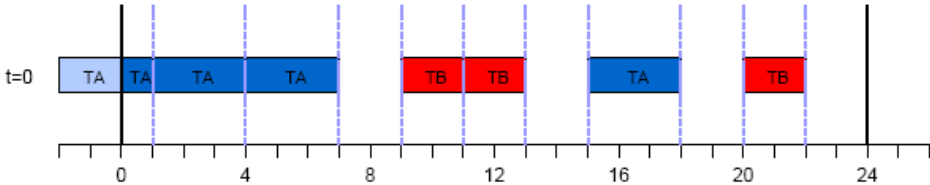


# Terminal Constraints

- Meet *nominal* demand

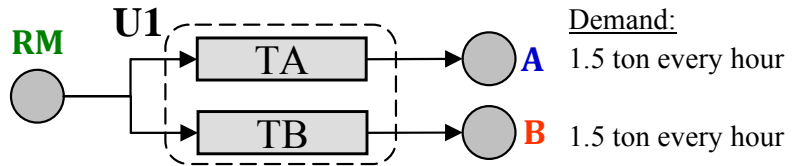


- Rolling horizon solutions ( $T = 24$ )

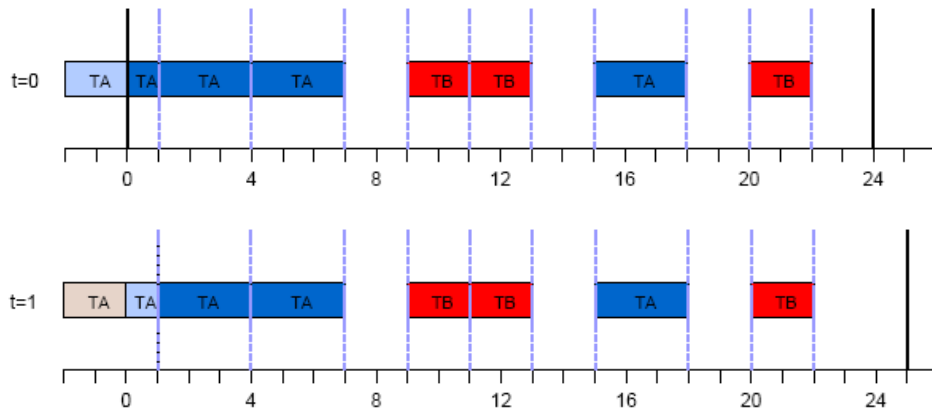


# Terminal Constraints

- Meet *nominal* demand



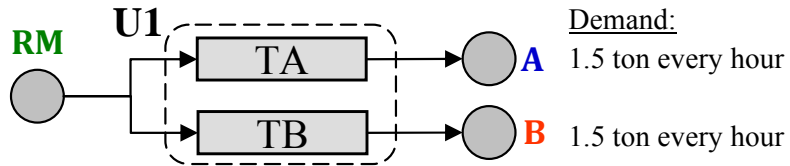
- Rolling horizon solutions ( $T = 24$ )



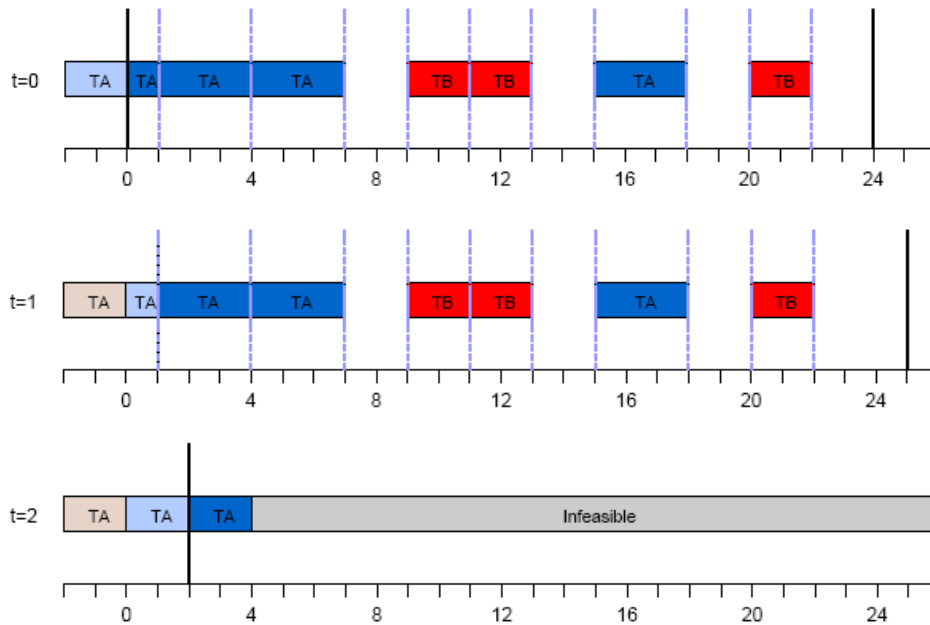
Many changeovers to keep inventories low

# Terminal Constraints

- Meet *nominal* demand



- Rolling horizon solutions ( $T = 24$ )

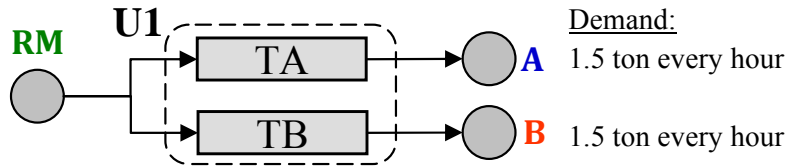


Many changeovers to keep inventories low

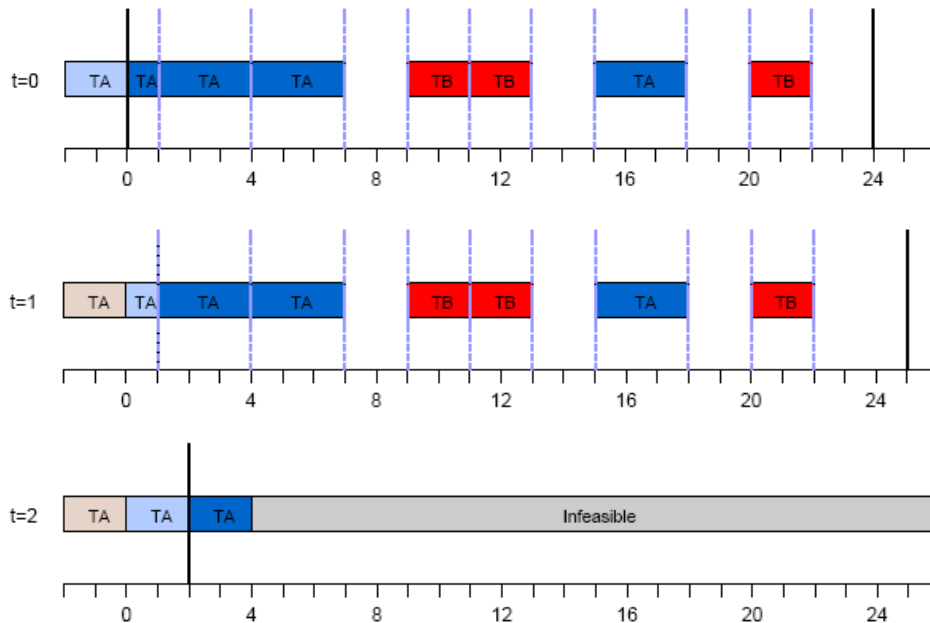
Inventory too low at  $t=25$   
 Cannot meet *new* demand

# Terminal Constraints

- Meet *nominal* demand



- Rolling horizon solutions ( $T = 24$ )



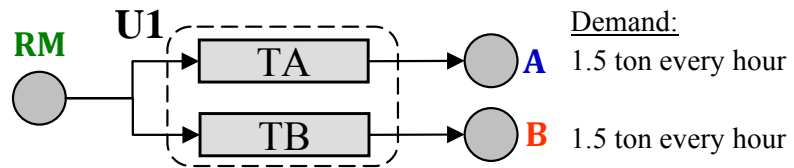
Many changeovers to keep inventories low

Inventory too low at  $t=25$   
Cannot meet *new* demand

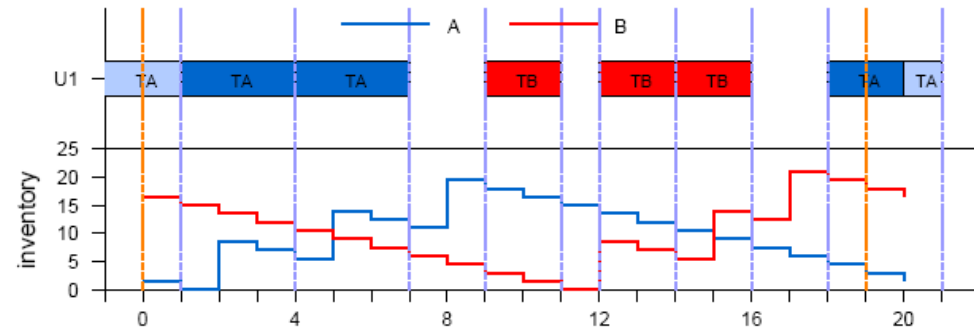
- Rolling horizon + Safety stock  $\Rightarrow$  Infeasible

# Terminal Constraints

- Meet *nominal* demand

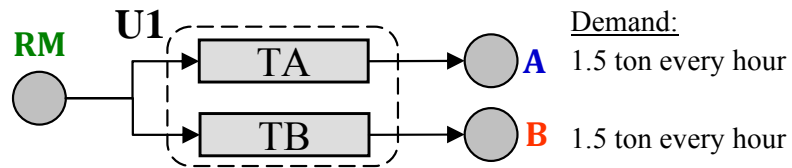


- Cyclic Solution (19-hr period)

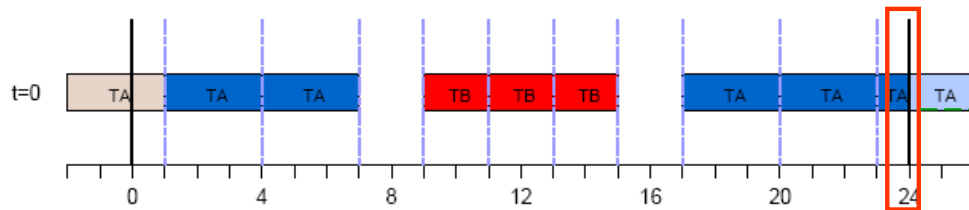


# Terminal Constraints

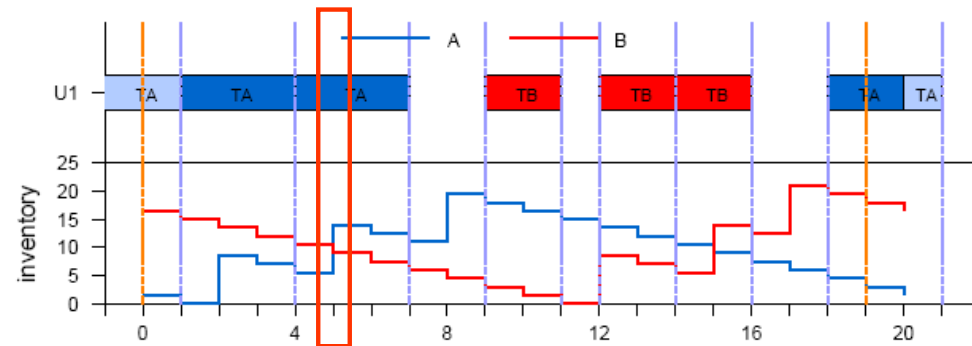
- Meet *nominal* demand



- Rolling horizon + terminal constraints ( $T = 24$ )
  - Go to any stage of the periodic solution

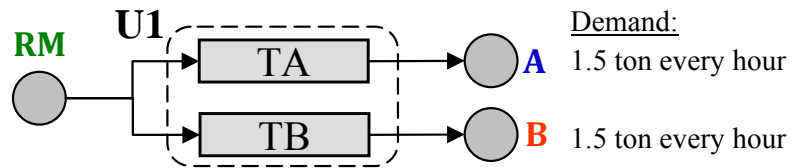


- Cyclic Solution (19-hr period)

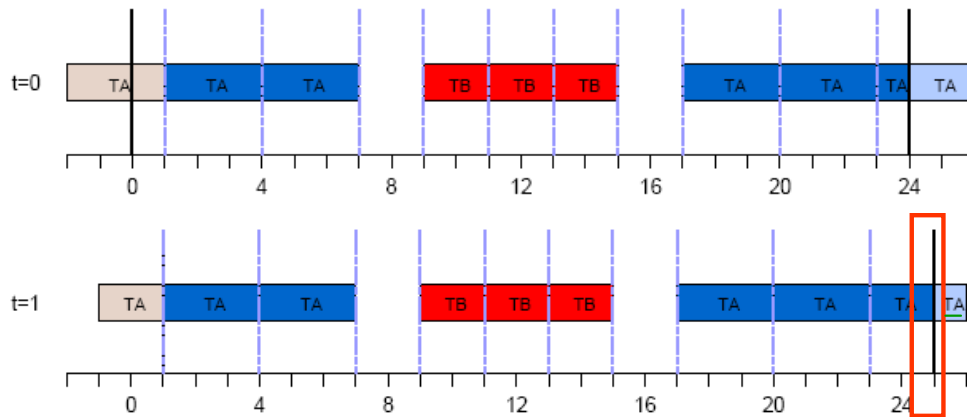


# Terminal Constraints

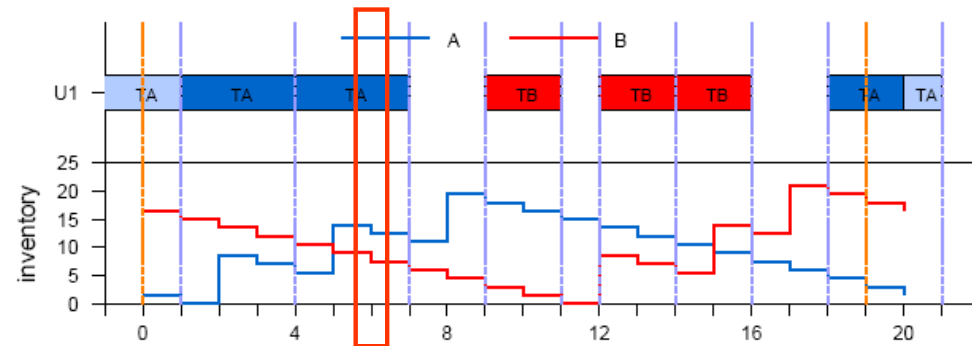
- Meet *nominal* demand



- Rolling horizon + terminal constraints ( $T = 24$ )
  - Go to any stage of the periodic solution

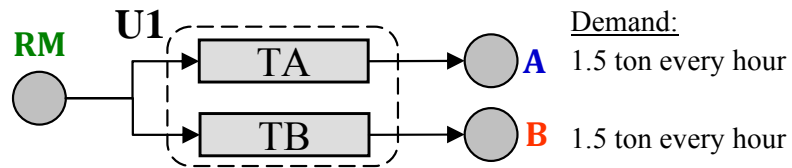


- Cyclic Solution (19-hr period)

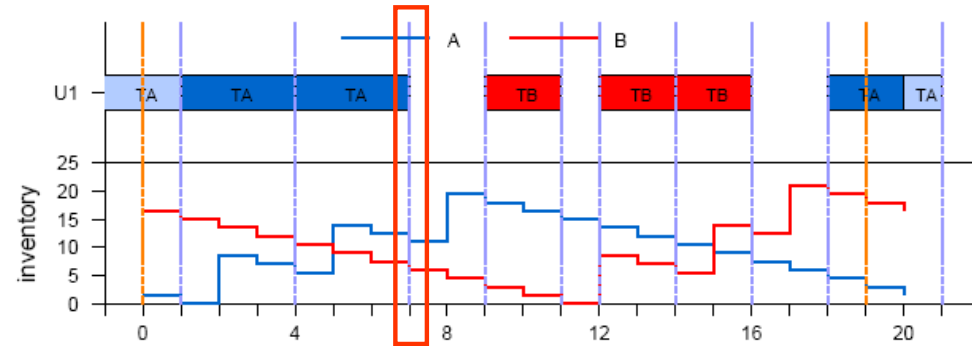


# Terminal Constraints

- Meet *nominal* demand

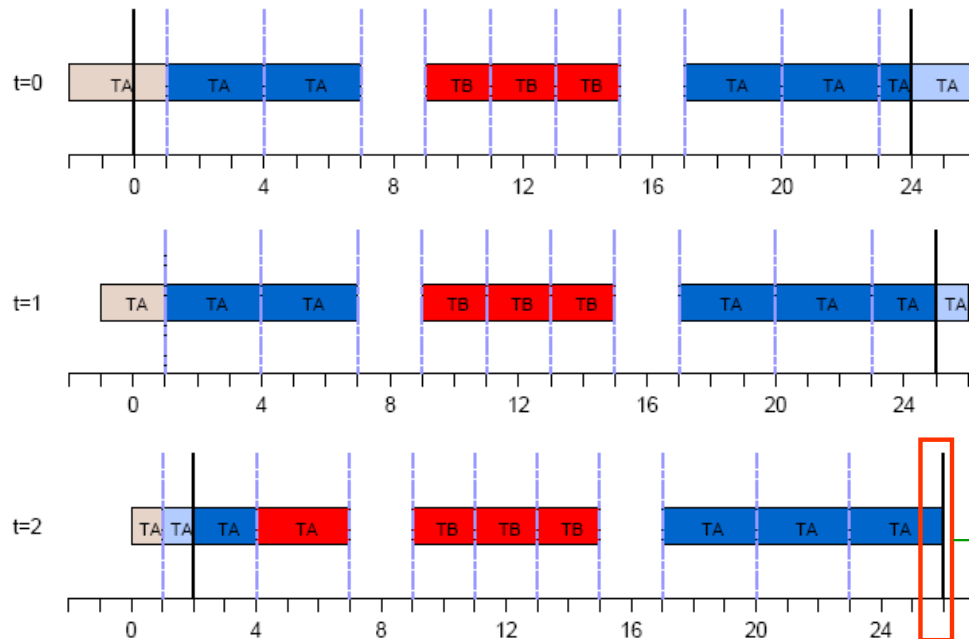


- Cyclic Solution (19-hr period)



- Rolling horizon + terminal constraints ( $T = 24$ )

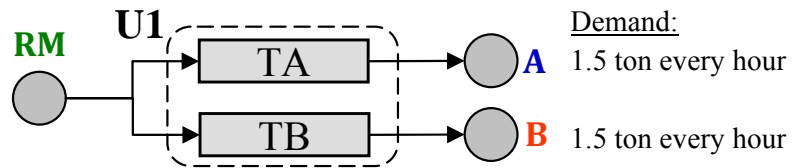
- Go to any stage of the periodic solution



- Terminal constraints ensure feasibility

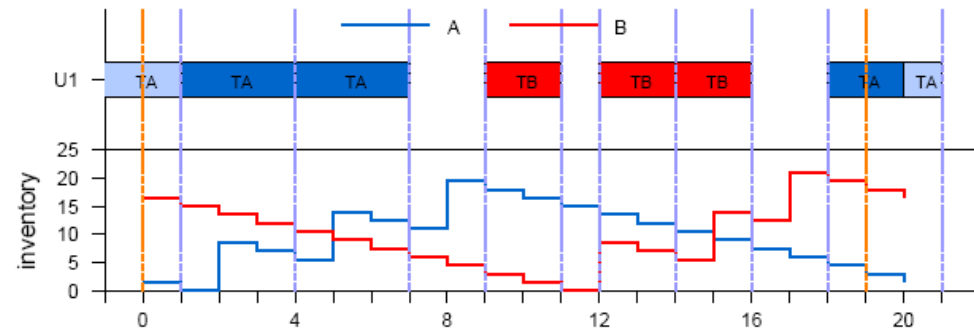
# Terminal Constraints

- Meet *nominal* demand



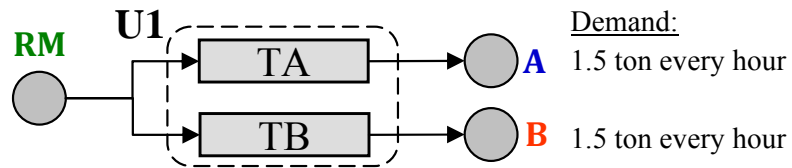
- Rolling horizon with terminal constraints and **shorter scheduling horizon** ( $T = 12$ )

- Cyclic Solution (19-hr period)

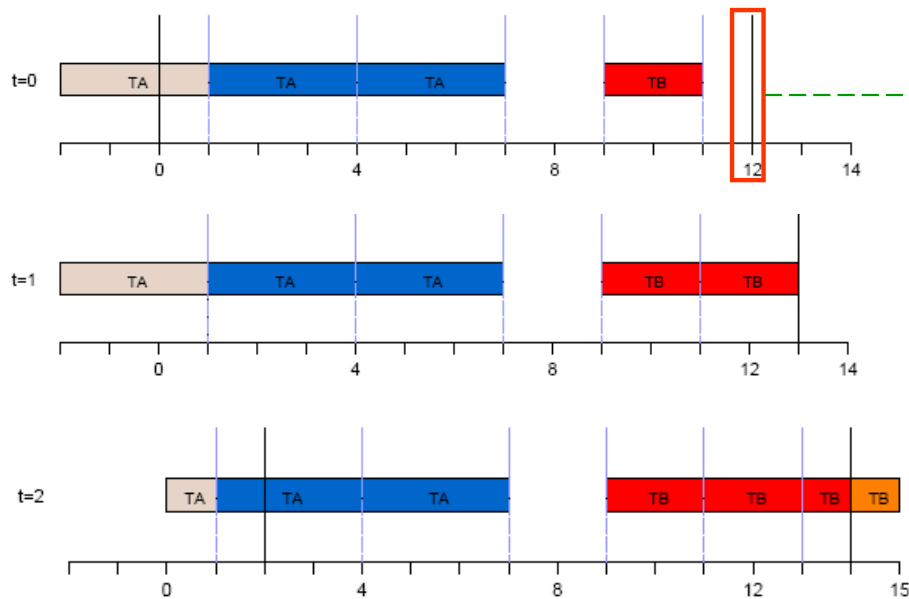


# Terminal Constraints

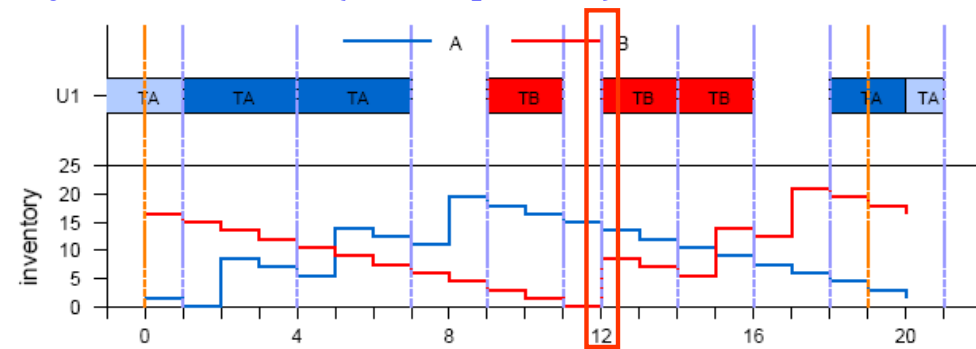
- Meet *nominal* demand



- Rolling horizon with terminal constraints and **shorter scheduling horizon ( $T = 12$ )**



- Cyclic Solution (19-hr period)



- Terminal constraints ensure feasibility
- Allow consideration of shorter scheduling horizon**

# Future Research Directions

## Distributed supply chain optimization (part I)

- Distributed restart for coupled unstable systems. Handle warm start infeasibility caused by unmodeled disturbances.
- Cooperative game theory. Explore using game theory methods for encouraging cooperation among competing players.
- Issues of scale. Hierarchical decomposition (Stewart et al., 2010).
- Optimizing economic performance. Economic MPC (Angeli et al., 2011; Diehl et al., 2011)

## Control theory in scheduling (part II)

# Future Research Directions

## Distributed supply chain optimization (part I)

- Distributed restart for coupled unstable systems. Handle warm start infeasibility caused by unmodeled disturbances.
- Cooperative game theory. Explore using game theory methods for encouraging cooperation among competing players.
- Issues of scale. Hierarchical decomposition (Stewart et al., 2010).
- Optimizing economic performance. Economic MPC (Angeli et al., 2011; Diehl et al., 2011)

## Control theory in scheduling (part II)

- Translate scheduling MIP models into state-space form (including objective function and disturbances)

# Future Research Directions

## Distributed supply chain optimization (part I)

- Distributed restart for coupled unstable systems. Handle warm start infeasibility caused by unmodeled disturbances.
- Cooperative game theory. Explore using game theory methods for encouraging cooperation among competing players.
- Issues of scale. Hierarchical decomposition (Stewart et al., 2010).
- Optimizing economic performance. Economic MPC (Angeli et al., 2011; Diehl et al., 2011)

## Control theory in scheduling (part II)

- Translate scheduling MIP models into state-space form (including objective function and disturbances)
- Design rescheduling algorithms with desired closed-loop properties

## Distributed supply chain optimization (part I)

- Distributed restart for coupled unstable systems. Handle warm start infeasibility caused by unmodeled disturbances.
- Cooperative game theory. Explore using game theory methods for encouraging cooperation among competing players.
- Issues of scale. Hierarchical decomposition (Stewart et al., 2010).
- Optimizing economic performance. Economic MPC (Angeli et al., 2011; Diehl et al., 2011)

## Control theory in scheduling (part II)

- Translate scheduling MIP models into state-space form (including objective function and disturbances)
- Design rescheduling algorithms with desired closed-loop properties
- Develop surrogate models for inventory management Sung and Maravelias (2009)

- D. Angeli, R. Amrit, and J. B. Rawlings. On average performance and stability of economic model predictive control. *IEEE Trans. Auto. Cont.*, 2011. Accepted for publication.
- B. J. Angerhofer and M. C. Angelides. System dynamics modelling in supply chain management: Research review. In *Simulation Conference Proceedings, 2000. Winter*, volume 1, pages 342–350, 2000.
- B. M. Beamon. Supply chain design and analysis: Models and methods. *Int. J. Prod. Econ.*, 55(3):281–294, 1998.
- M. W. Braun, D. E. Rivera, W. M. Carlyle, and K. G. Kempf. A model predictive control framework for robust management of multi-product, multiechelon demand networks. In *IFAC, 15th Triennial World Congress*, 2002.
- G. Cachon and S. Netessine. Game theory in supply chain analysis. *Tutorials in Operations Research: Models, Methods, and Applications for Innovative Decision Making*, 2006.
- G. Cachon and P. Zipkin. Competitive and cooperative inventory policies in a two stage supply chain. *Manage Sci.*, 45:936–953, 1999.
- A. J. Clark and H. Scarf. Optimal policies for a multiechelon inventory problem. *Manage Sci.*, 6:475–490, 1960.

## References II

- J. Dejonckheere, S. Disney, M. Lambrecht, and D. Towill. Measuring and avoiding the bullwhip effect: A control theoretic approach. *Eur. J. Oper. Res.*, 147(3):567–590, 2003.
- M. Diehl, R. Amrit, and J. B. Rawlings. A Lyapunov function for economic optimizing model predictive control. *IEEE Trans. Auto. Cont.*, 56(3):703–707, 2011.
- L. Dong and H. Lee. Optimal policies and approximations for a serial multiechelon inventory system with time-correlated demand. *Oper. Res.*, 51:969–980, 2003.
- W. B. Dunbar and S. Desa. Distributed model predictive control for dynamic supply chain management. In *Assessment and Future Directions of Nonlinear Model Predictive Control*. Springer, 2007.
- A. Federgruen. Centralized planning models for multiechelon inventory systems under uncertainty. *Handbooks Oper. Res. Manage. Sci.*, 4:133–173, 1993.
- A. Federgruen and P. Zipkin. Computational issues in an infinite-horizon, multiechelon inventory model. *Oper. Res.*, 32:818–836, 1984.
- M. Leng and M. Parlar. Game theoretic applications in supply chain management: A review. *INFOR*, 43(3):187–220, 2005.
- X. Li and T. E. Marlin. Robust supply chain performance via model predictive control. *Comput. Chem. Eng.*, 33(12, Sp. Iss. SI):2134–2143, Dec. 2009.

## References III

- P.-H. Lin, D. S.-H. Wong, S.-S. Jang, S. S. Shieh, and J.-Z. Chu. Controller design and reduction of bullwhip for a model supply chain system using z-transform analysis. *J. Proc. Cont.*, 14:487–499, September 2004.
- J. M. Maestre, D. Muñoz de la Peña, and E. F. Camacho. Distributed MPC: A supply chain case study. In *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, 2009.
- E. Mestan, M. Türkay, and Y. Arkun. Optimization of operations in supply chain systems using hybrid systems approach and model predictive control. *Ind. Eng. Chem. Res.*, 45:6493–6503, August 2006.
- M. Nagarajan and G. Sošić. Game-theoretic analysis of cooperation among supply chain agents: Review and extensions. *Eur. J. Oper. Res.*, 187(3):719–745, 2008.
- E. Perea López, I. E. Grossmann, B. E. Ydstie, and T. Tahmassebi. Dynamic modeling and decentralized control of supply chains. *Ind. Eng. Chem. Res.*, 40:3369–3383, June 2001.
- E. Perea López, B. E. Ydstie, and I. E. Grossmann. A model predictive control strategy for supply chain optimization. *Comput. Chem. Eng.*, 27(8-9):1201–1218, February 2003.
- S. Raghunathan. Impact of demand correlation on the value of and incentives for information sharing in a supply chain. *Eur. J. Oper. Res.*, 146(3):634–649, 2003.

## References IV

- P. Seferlis and N. F. Giannelos. A two-layered optimization-based control strategy for multi-echelon supply chain networks. *Comput. Chem. Eng.*, 28:1121–1129, 2004.
- N. Shah. Process industry supply chains: Advances and challenges. *Comput. Chem. Eng.*, 29:1225–1235, April 2005.
- K. Shang and J. Song. Newsvendor bounds and heuristic for optimal policies in serial supply chains. *Manage Sci.*, 49:618–638, 2003.
- B. T. Stewart, J. B. Rawlings, and S. J. Wright. Hierarchical cooperative distributed model predictive control. In *Proceedings of the American Control Conference*, Baltimore, Maryland, June 2010.
- C. Sung and C. Maravelias. A projection-based method for production planning of multiproduct facilities. *AIChE J.*, 55(10):2614–2630, 2009.
- D. R. Towill. Dynamic analysis of an inventory and order based production control system. *Int. J. Prod. Res.*, 6:671–687, 1982.
- A. White. Management of inventory using control theory. *Int. J. Tech. Manage.*, 17(7): 847–860, 1999.