OPTIMIZATION METHODS FOR BATCH SCHEDULING

Jaime Cerdá

Instituto de Desarrollo Tecnológico para la Industria Química
Universidad Nacional de Litoral - CONICET
Güemes 3450 – 3000 Santa Fe - Argentina

OUTLINE

- Problem definition
- Types of scheduling problems
- Types of scheduling methodologies
- Types of scheduling optimization approaches
- Overview of network-type discrete and continuous time models
- Comparison of network-type discrete and continuous time formulations (benchmarking examples)
- Overview of batch-oriented continuous time formulations
- Conclusions

MAIN REFERENCES

Méndez, C.A., Cerdá, J., Grossmann, I.E., Harjunkoski, I., Fahl, M. "State-of-the-Art Review of Optimization Methods for Short-Term Scheduling of Batch Processes". Submitted to Computers & Chemical Engineering (July, 2005).

Castro, P.M.; Barbosa-Póvoa, A.P.; Matos, H.A. & Novais, A.Q. (2004) *I&EC Research*, 43, 105 – 118.

Cerdá, J.; Henning, G.P. & Grossmann, I.E. (1997) *I&EC Research*, 36, 1695 – 1707.

Floudas, C.A.; Lin, X. (2004) *Computers and Chemical Engineering,* 28, 2109 – 2129.

Ierapetritou, M.G. & Floudas, C.A. (1998) *I&EC Research*, 37, 4341 – 4359.

Janak, S.L.; Lin, X. & Floudas, C.A. (2004) *I&EC Research,* 43, 2516 – 2533.

Kondili, E; Pantelides, C.C. & Sargent, W.H. (1993) *Computers and Chemical Engineering,* 2, 211 – 227.

Maravelias, C.T. & Grossmann, I.E. (2003) *I&EC Research*, 42, 3056 – 3074.

Méndez, C.A.; Henning, G.P. & Cerdá, J. (2001) *Computers and Chemical Engineering,* 25, 701 – 711.

Méndez, C.A. & Cerdá, J. (2003) *Comp. & Chem. Eng.,* 27, 1247 – 1259.

Pantelides, C.C. (1994) *Foundations of Computer-Aided Process Operations,* Cache publications, New York, 253 – 274.

Pinto, J.M. & Grossmann, I.E. (1995) *I&EC Research,* 34, 3037 – 3051.

Pinto, J.M. & Grossmann, I.E. (1997) *Computers and Chemical Engineering,* 21, 801 – 818.

Pinto, J.M. & Grossmann, I.E. (1998). *Annals of Operations Research*, 81, 433 – 466.

Reklaitis, G.V. (1992). Overview of scheduling and planning of batch process operations. *NATO Advanced Study Institute—Batch process systems engineering*. Turkey: Antalya.

INTRODUCTION

PROBLEM DEFINITION

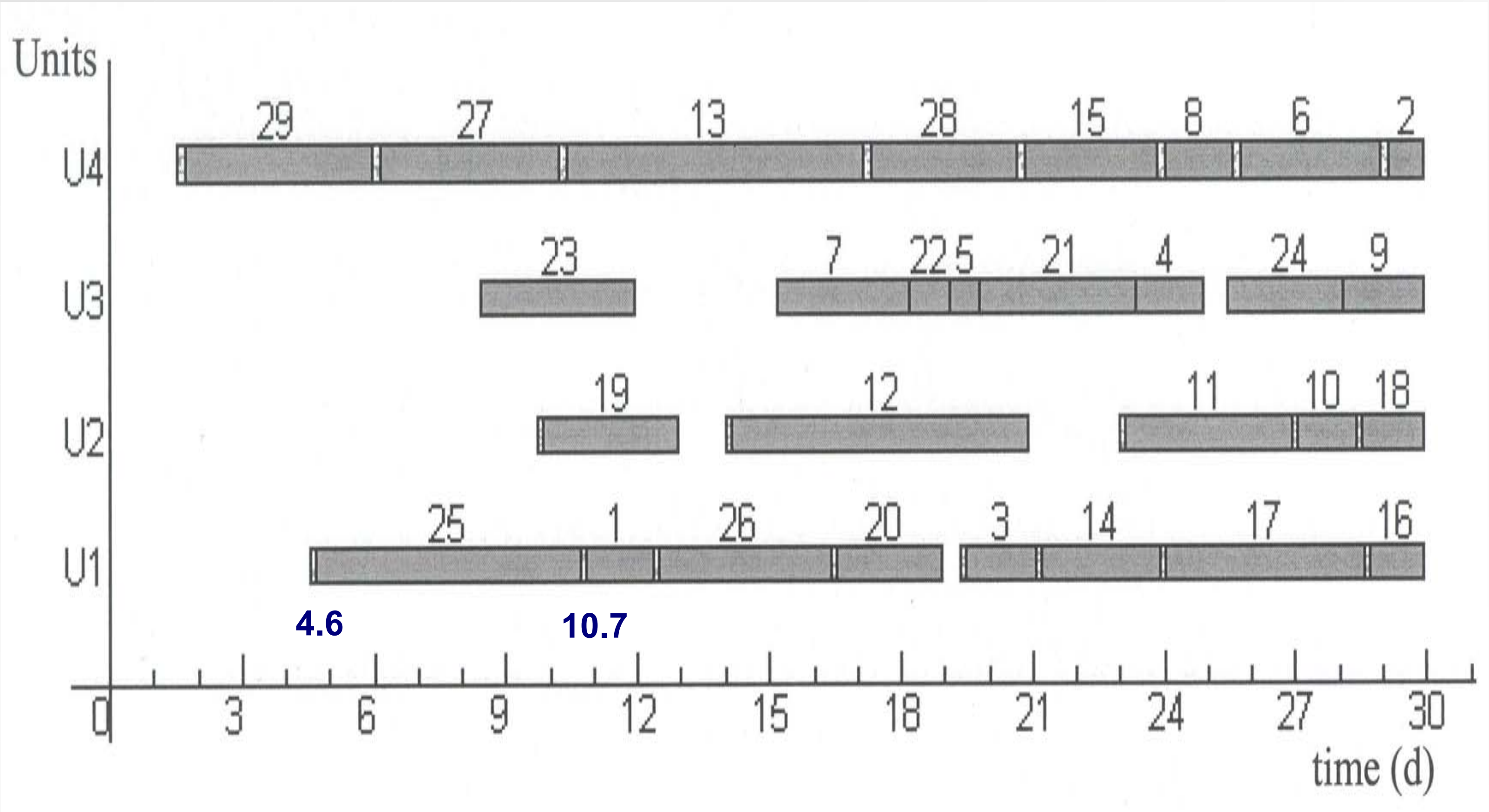
- Scheduling is a decision-making process thay plays an important role in most manufacturing and service industries
- The scheduling function aims to optimally allocate resources, available in limited supplies, to processing tasks over time.
- Each task requires certain amounts of specified resources for a specific time interval called the processing time
- Resources may be equipment units in a chemical plant, runways at an airport or crews at a construction site
- Tasks may be operations in a chemical plant, takeoffs and landings at an airport, activities in a construction project

SCHEDULING DECISIONS AND GOALS

- A proper allocation of resources to tasks enables the company to achieve its objectives
- The objectives may take many forms such as:
  - minimizing the time required to complete all the tasks (the makespan)
  - minimizing the number of orders completed after their committed due dates
  - maximizing customer satisfaction by completing orders in a timely fashion
  - maximizing plant throughput
  - maximizing profit or minimizing production costs
- Two eligible tasks cannot generally use the same required resource simultaneously but one at a time
- Scheduling decisions to be made include:
  - allocating resources to tasks
  - sequencing tasks allocated to the same resource item
  - task timing

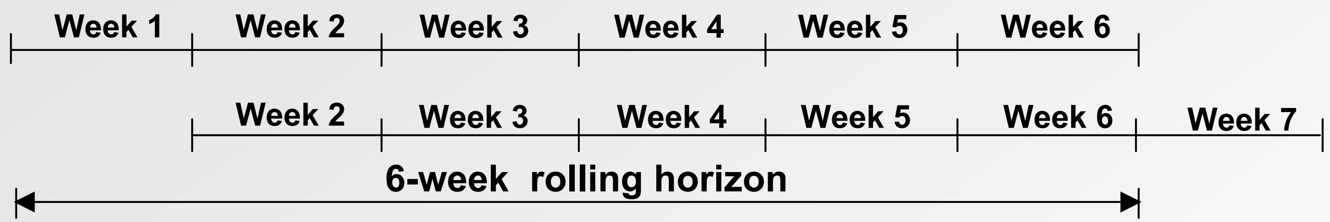
AN ILLUSTRATIVE EXAMPLE

29 Tasks - 4 Equipment Units - One-month Period Horizon



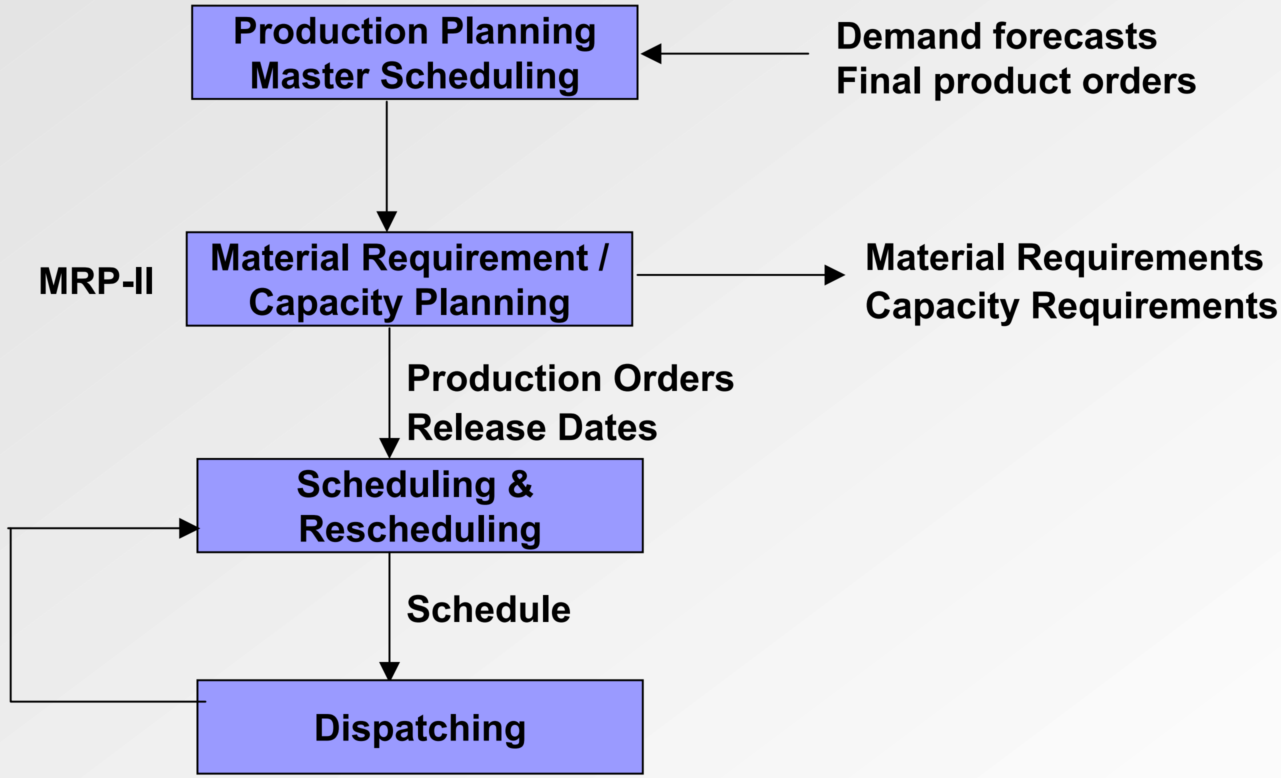
ROLLING TIME HORIZON

- The scheduling rolling horizon ranges from 2 to 6 weeks, depending on whether task processing times are on the order of a day or a week.



- The full schedule for a 6-week horizon might be updated once a week using updated order input and plant state.
- There will be frequent corrections to the schedule in midweek to account for unit breakdowns or late order arrivals
- The scheduling function has to interact with other decision-making systems used in the plant like the material requirement planning (the MRP system)
- The MRP system provides information on the weekly production order arrivals (product, arrival time, due date and order size), together with the tasks required to complete each order.

INTERACTION WITH THE MRP SYSTEM



- After the schedule has been developed, all raw materials and resources must be available at the specified times
- MRP-II aims to guarantee that the required raw materials and intermediates will be available in the right amounts at the right times, and the plant capacity is enough to process all the required productions orders

TYPES OF SCHEDULING PROBLEMS

- **STATIC vs. DYNAMIC PROBLEMS**

  In static problems, all the production orders and their arrival times are known beforehand

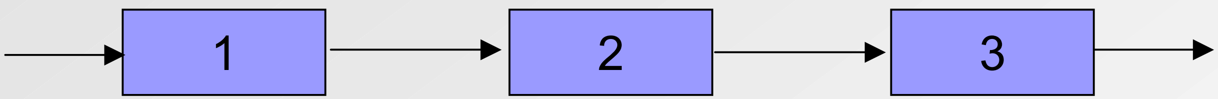
  In dynamic problems, new production orders can arrive at unexpected times while the schedule is being executed

- **FLOW SHOP vs. JOB SHOP PROBLEMS**

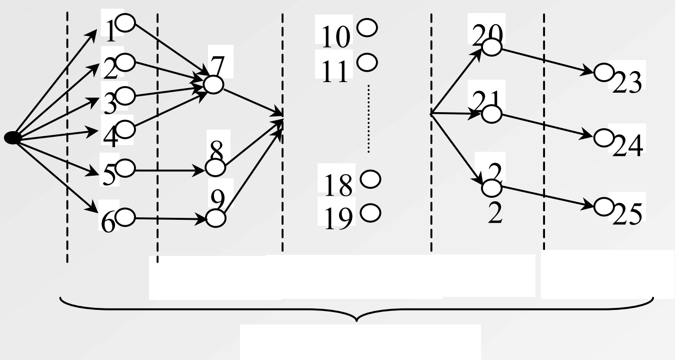
  Assume that the jobs require to perform multiple operations on different machines.

  - **Flow shop**: Every job consists of the same set of tasks to be performed in the same order. The units are accordingly arranged in production lines to minimize the movement of materials and manpower (multiproduct plant)
  - **Compound Flow shop**: Each unit in the series may be replaced by a set of parallel equipment items which may be identical or very different. Each job goes to one unit in the first stage, then it is transferred to one in the second stage and so on.
  - **Job shop**: Production orders have different routes (require different sequences of tasks) and some orders may even visit a given unit several times (multipurpose plants)

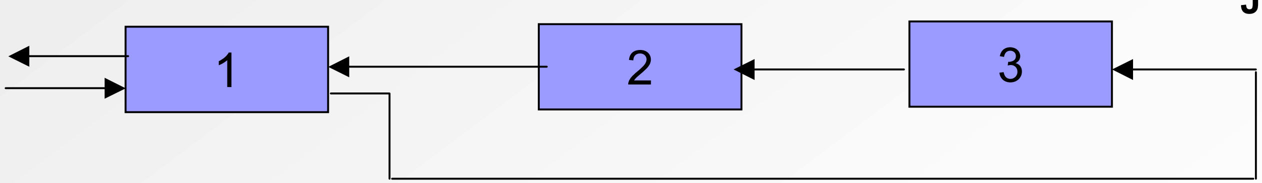
TYPES OF BATCH PRODUCTION FACILITIES



Flow-shop facility



Compound Flow-shop facility



Job-shop facility

TYPES OF SCHEDULING PROBLEMS -2

- **MAKE-TO-STOCK vs. MAKE-TO-ORDER PRODUCTION FACILITIES**

MAKE-TO-STOCK FACILITIES:

- A make-to-stock manufacturing facility opt to keep in stock items for which there is a steady demand and no risk of obsolescence.
- Items that are produced for inventory do not have tight due dates
- The lot size is determined by a trade-off between setup costs and inventory holding costs
- Make-to-stock manufacturing plants are referred to as "open shops"

MAKE-TO-ORDER FACILITIES:

- Make-to-order jobs have specified due dates and their sizes are determined by the customer
- Each order is unique and has a unique routing throughout the plant
- Make-to-order manufacturing facilities are referred to as "closed shops"
- Many manufacturing plants operate partly as a make-to-stock facility processing warehouse orders and partly as a make-to-order facility processing customer orders

TYPES OF SCHEDULING APPROACHES

- **HEURISTIC METHODS**
  - Basic Dispatching Rules
  - Composite Dispatching Rules
- **ALGORITHMS OF THE IMPROVEMENT TYPE**
  - Simulated Annealing
  - Genetic Algorithms
  - Tabu Search
- **OPTIMIZATION APPROACHES**
  - Discrete Time Models
  - Continuous Time Models

    Network-oriented Formulations

    Batch-oriented Formulations

HEURISTIC SCHEDULING METHODS

- **BASIC DISPATCHING RULES**

  - A basic dispatching rule is a rule that prioritizes all the jobs that are waiting for processing on a machine
  - The prioritization scheme may take into account jobs' attributes and machines' attributes as well as the current time
  - Whenever a machine has been freed, a dispatching rule inspects the waiting jobs and selects to process next the job with the highest priority
  - Dispatching rules can be classified into STATIC and DYNAMIC RULES.
    * A STATIC RULE is not time-dependent but just a function of the job data, the machine data or both (EDD-earliest due date first, SPT-shortest processing time first)
    * DYNAMIC RULES are time-dependent since they also take into account, in addition to the job and machine data, the current time (Example: MS-minimum slack time-first)
  - Dispatching rules can also be categorized into two classes: LOCAL and GLOBAL RULES
    * A LOCAL RULE uses only information related to either the queue or the machine / workcenter to which the rule is applied
    * A GLOBAL RULE may use information related to other machines, such as either the processing times of the jobs or the current queue length on the next machine

IMPROVEMENT ALGORITHMS

- **COMPOSITE DISPATCHING RULES**

  - Composite dispatching rules combine a number of basic dispatching rules
  - Each basic rule in the composite dispatching rule has its own scaling parameter that is chosen to properly scale the contribution of the basic rule to the final decision

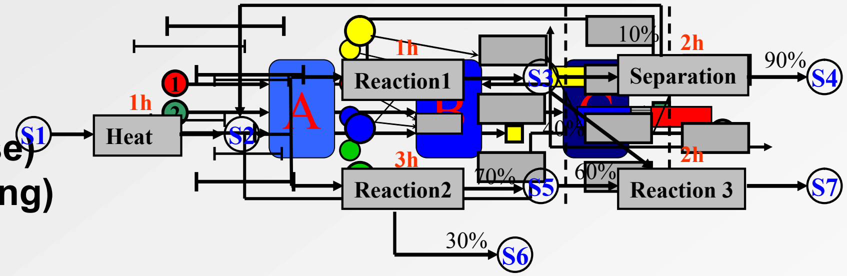
- **ALGORITHMS OF THE IMPROVEMENT TYPE**

  - Start with a complete schedule, which may be selected arbitrarily
  - Try to obtain a better schedule by manipulating the current schedule
  - Use local search procedures which do not guarantee an optimal solution
  - Attempt to find a better schedule than the current one in the neighborhood of the current one.
  - Two schedules are said to be neighbors if one can be obtained from the other through a well-defined modification scheme
  - The procedure either accepts or rejects a candidate solution as the next schedule to move to, based on a given acceptance-rejection criterion
  - The four elements of an improvement algorithm are: the schedule representation, the neighborhood design, the search process within the neighborhood and the acceptance-rejection criterion.

OPTIMIZATION APPROACHES

- **DISCRETE TIME MODELS OF THE NETWORK TYPE**
  - State-Task-Network (STN)-based discrete formulation
  - Resource-Task-Network (RTN)-based discrete formulation
- **CONTINUOUS TIME MODELS OF THE NETWORK TYPE**
  - Global Time Points
    * STN-based continuous time formulations
    * RTN-based continuous time formulations
  - Unit-Specific Time Events
    * STN-based unit-specific continuous time formulations
- **BATCH-ORIENTED CONTINUOUS TIME MODELS**
  - Time Slot-based formulations
  - Precedence-based formulations
    * Unit-specific immediate precedence-based models
    * Global direct precedence-based models
    * Global general precedence-based models

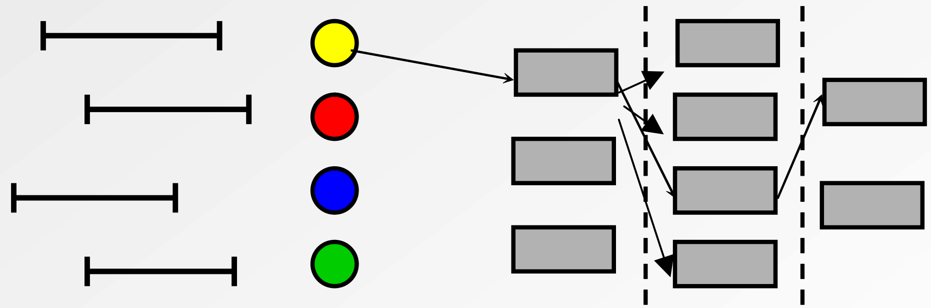
KEY ASPECTS IN BATCH SCHEDULING



(1) PROCESS TOPOLOGY:
- Single Stage (single unit or parallel units)
- Multiple Stage (multiproduct or multipurpose)
- Network (batch splitting and mixing, recycling)

(2) EQUIPMENT ASSIGNMENT
- Fixed (dedicated)
- Variable

(3) EQUIPMENT CONNECTIVITY
- Partial
- Full



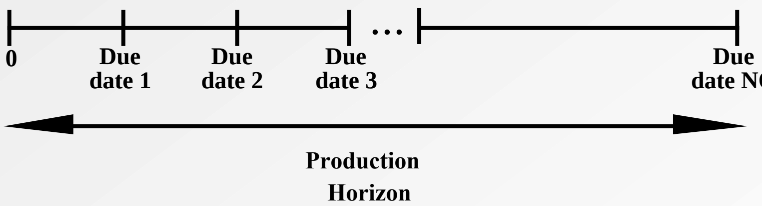
(4) INVENTORY STORAGE POLICIES
- Unlimited intermediate storage (UIS)
- Non-intermediate storage (NIS)
- Finite intermediate storage (FIS): Dedicated or shared storage units
- Zero wait (ZW)

(5) MATERIAL TRANSFER
- Instantaneous (neglected)
-Time consuming (no-resource, pipes, vessels)

KEY ASPECTS IN BATCH SCHEDULING

(6) BATCH SIZE:
- Fixed
- Variable (mixing and splitting operations)

(7) BATCH PROCESSING TIME
- Fixed
- Variable (unit / batch size dependent)



(8) DEMAND PATTERNS
- Due dates (single or multiple product demands)
- Scheduling horizon (fixed, minimum/maximum requirements)

(9) CHANGEOVERS
- None
- Unit dependent
- Sequence dependent (product or product/unit dependent)

(10) RESOURCE CONSTRAINTS
- None (only equipment)
- Discrete (manpower)
- Continuous (utilities)

KEY ASPECTS IN BATCH SCHEDULING

(11) TIME CONSTRAINTS

- None
- Non-working periods
- Maintenance
- Shifts

(12) COSTS

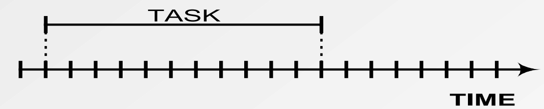
- Equipment
- Utilities (fixed or time dependent)
- Inventory
- Changeovers

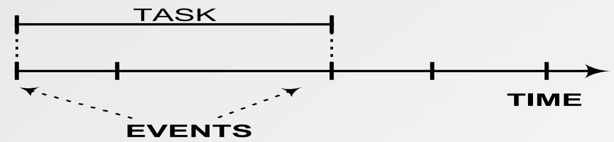
(13) Degree of certainty

- Deterministic
- Stochastic

ROAD-MAP FOR OPTIMIZATION APPROACHES

(A) TIME DOMAIN REPRESENTATION
- Discrete time
- Continuous time

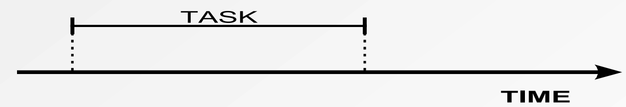


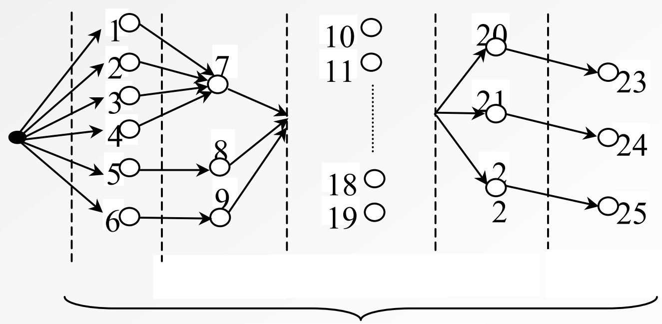


(B) EVENT REPRESENTATION

DISCRETE TIME
- Global time intervals

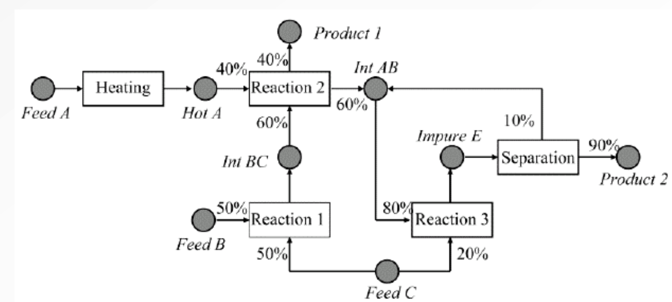
CONTINUOUS TIME
- Time slots
- Unit-specific direct precedence
- Global direct precedence
- Global general precedence
- Global time points
- Unit- specific time event





(C) MATERIAL BALANCES
- Lots (Order or batch oriented)
- Network flow equations (STN or RTN problem representation)

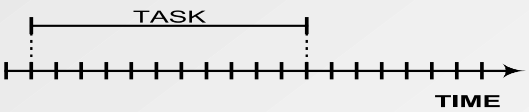
(D) OBJECTIVE FUNCTION
- Makespan
- Earliness/ Tardiness
- Profit
- Inventory
- Cost



DISCRETE TIME MODEL FEATURES

(A) TIME DOMAIN REPRESENTATION

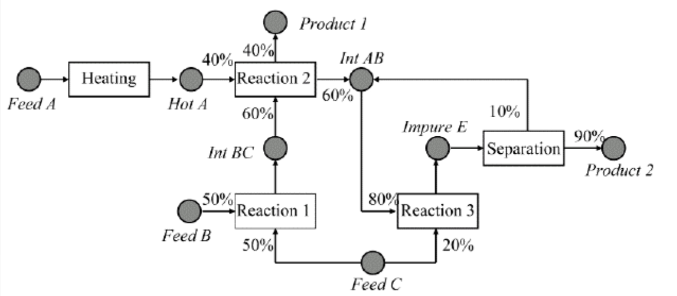
- DISCRETE TIME



(B) EVENT REPRESENTATION

DISCRETE TIME

- Global time interval



(C) MATERIAL BALANCES

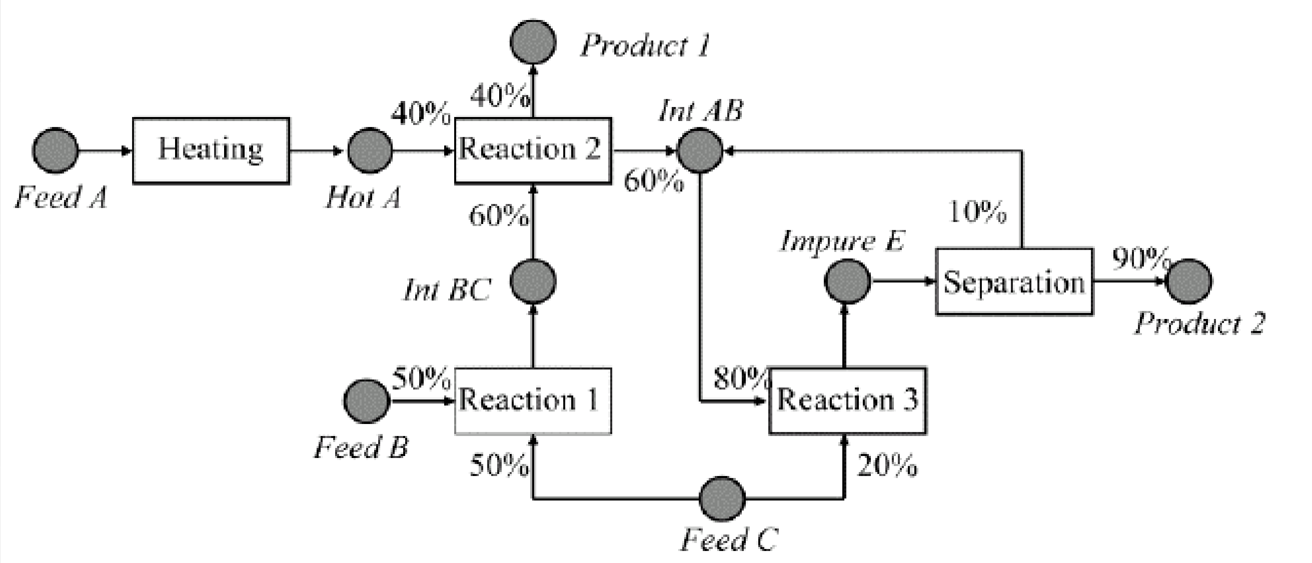
- Network flow equations (STN or RTN problem representation)

(D) OBJECTIVE FUNCTION

- Profit
- Cost

STATE-TASK NETWORK (STN) REPRESENTATION

The STN process representation is a directed graph consisting of three elements:



- **State nodes:** standing for the feeds, intermediates and final products and represented by circles.
- **Task nodes:** representing the process operations which transform material from one or more input states into one or more output states, and denoted by rectangles
- **Directed arcs:** linking states and tasks to indicate the flow of materials

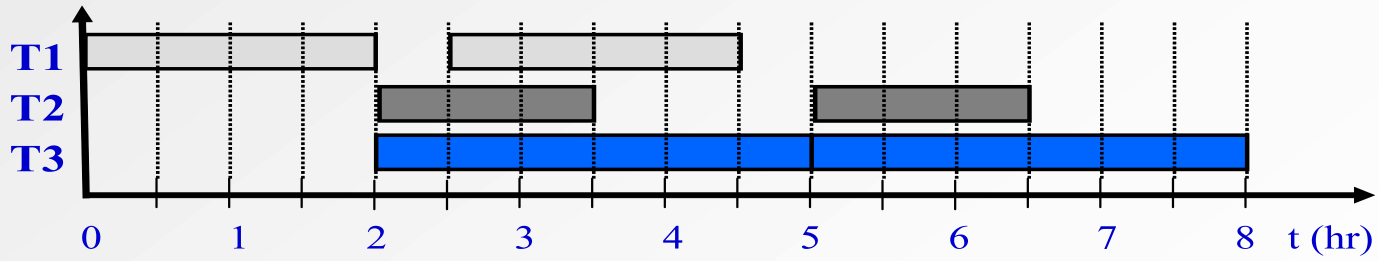
STN-BASED DISCRETE TIME FORMULATION

(Kondili et al., 1993; Shah et al., 1993)

DISCRETE TIME REPRESENTATION

- The time horizon is divided into a number of intervals of equal duration (uniform time grid).
- The uniform time grid is valid for all shared resources like equipment, utilities or manpower, i.e. global time intervals.
- Events of any type should occur at the interval boundaries.

- It can be regarded as events:
  - the start or the end of processing tasks
  - changes in the availability of any resource
  - changes in the resource requirement along the execution of a task



STN-BASED DISCRETE TIME FORMULATION

OTHER MAJOR ASSUMPTIONS

- The batch processing time for any task is constant, i.e. it does not change with the batch size.
- Dedicated storage tanks for each final/intermediate product are available.
- Every batch of state/product s is transferred to the assigned tank (or the next unit) immediately after finishing the processing task
- A processing unit cannot be used as a temporary storage device.
- A batch size changing with both the processing task and the assigned unit can be selected by the model.

INTERESTING PROBLEM FEATURES

- Product demands or bounds on product demands (in ton or cubic meters) are given
- Batches are to be generated and scheduled by solving the model.
- Alternative equipment units for a particular processing task can be available.
- The resource requirement may change along the task execution.
- Limited resource capacities are available

STN-BASED DISCRETE TIME FORMULATION

MAJOR MODEL VARIABLES

a) **BINARY VARIABLES**: $W_{i\,j\,t}$

task — unit — time interval

$W_{i\,j\,t}$ = 1 only if the processing of a batch undergoing task i in unit j $\in J_i$ is started at time point t.

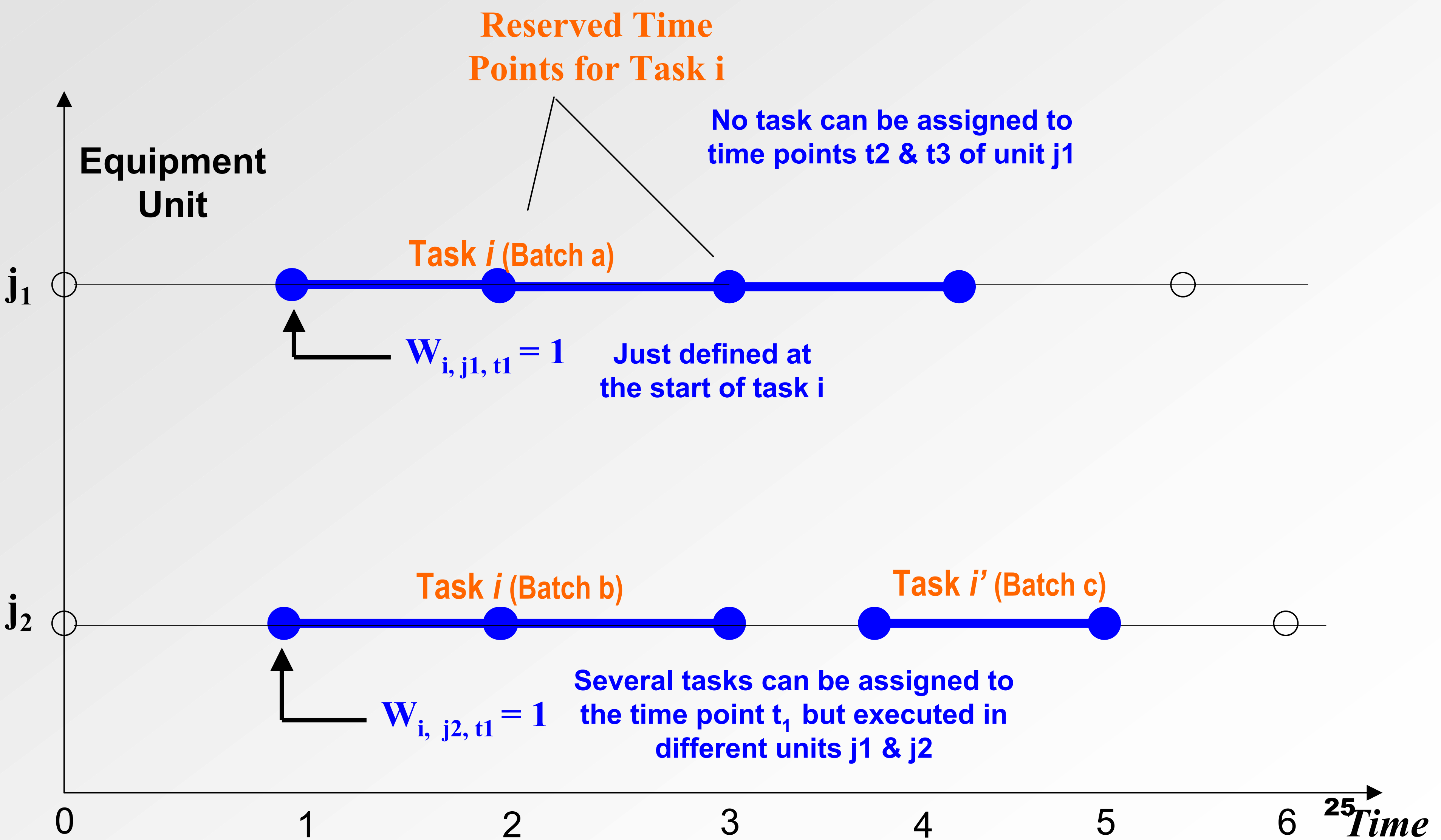
b) **CONTINUOUS VARIABLES**:

$B_{i\,j\,t}$ = size of the batch (i, j, t)

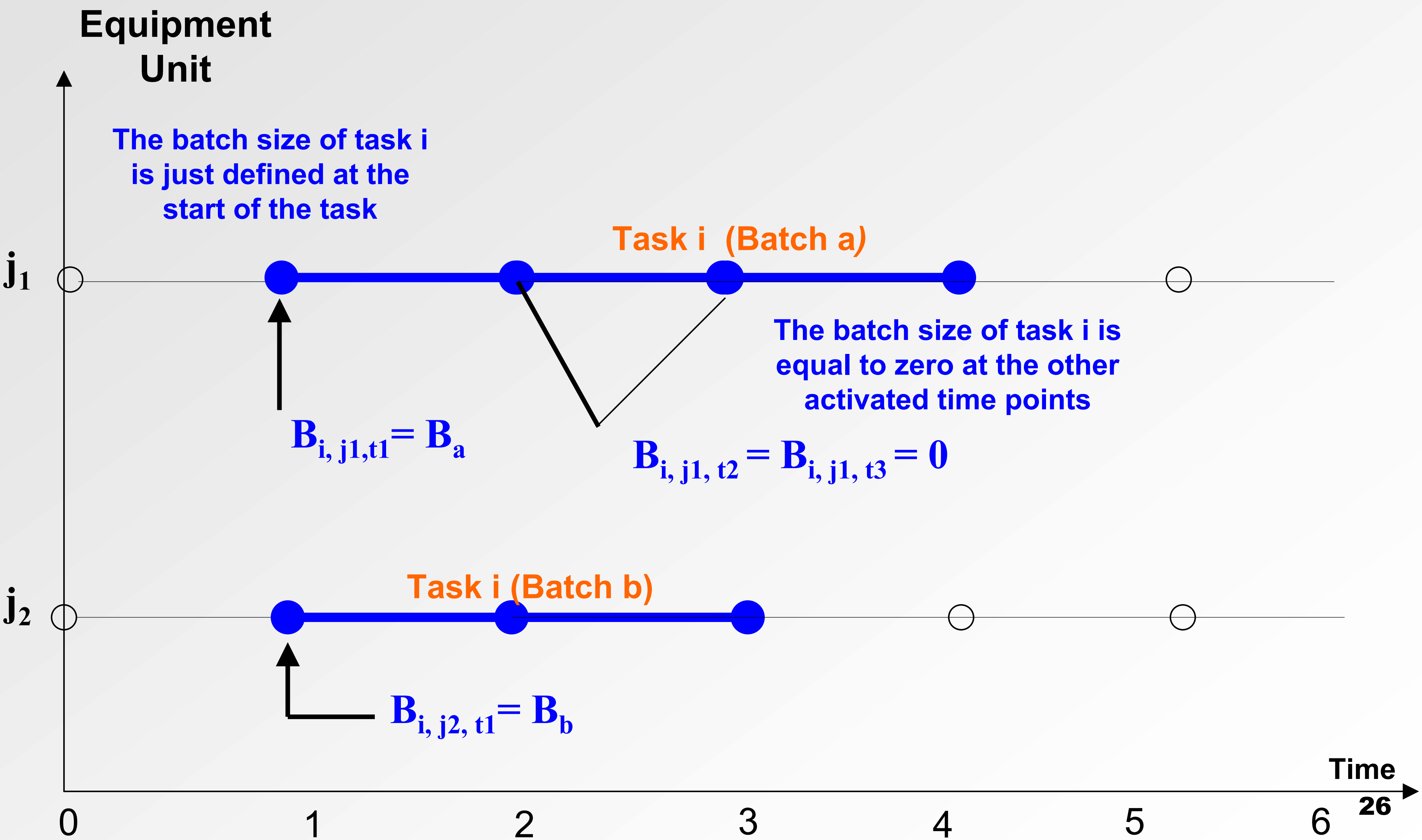
$S_{s\,t}$ = available inventory of state s$\in$S at time point t

$R_{r\,t}$ = requirement of resource *r* (different from equipment) at time point t

STN-BASED DISCRETE TIME FORMULATION



STN-BASED DISCRETE TIME FORMULATION



STN-BASED DISCRETE TIME FORMULATION

MAJOR CONSTRAINTS

- **ALLOCATION & SEQUENCING**

  At most a single task can be performed in a particular processing unit at any time point t.

- **BATCH SIZE**

  The size of a batch undergoing task *i* in unit $j \in J_i$ must be chosen within bounds.

- **MATERIAL BALANCES**

  The inventory of state s at time t is equal to that stored at time (t-1), plus the amount of s produced or received as raw material from external sources, minus the amount of s consumed in the process or delivered to the market during time interval t.

- **RESOURCE BALANCES**

  - **The total demand of resource r at time interval t is equal to the sum of the rth-resource requirements from tasks being executed at time t.**
  - **The overall resource requirement must never exceed the maximum rth-resource capacity**

- **CHANGEOVER TIMES**

  If unit *j* starts processing any task of family f at time t, no task *i'* of family f' can be initiated at least ($cl_{f'f} + pt_{i'j}$) units of time before time interval t.

STN-BASED DISCRETE TIME FORMULATION

(Kondili et al., 1993; Shah et al., 1993)

$$\sum_{i \in I_j} \sum_{t'=t-pt_{ij}+1}^{t} W_{ijt'} \leq 1 \qquad \forall j,t$$

ALLOCATION AND SEQUENCING

$$V_{ij}^{\min} W_{ijt} \leq B_{ijt} \leq V_{ij}^{\max} W_{ijt} \qquad \forall i, j \in J_i, t$$

BATCH SIZE

$$S_{st} = S_{s(t-1)} + \sum_{i' \in I_s^p} \rho_{is}^p \sum_{j \in J_i} B_{ij(t-pt_{is})} - \sum_{i' \in I_s^c} \rho_{is}^c \sum_{j \in J_i} B_{ijt} + \Pi_{st} - D_{st} \qquad \forall s,t$$

MATERIAL BALANCE

$$C_s^{\min} \leq S_{st} \leq C_s^{\max} \qquad \forall s,t$$

$$R_{rt} = \sum_i \sum_{j \in J_i} \sum_{t'=0}^{pt_{ij}-1} \left(\mu_{irt'} W_{ij(t-t')} + \nu_{irt'} B_{ij(t-t')} \right) \qquad \forall r,t$$

RESOURCE BALANCE

$$0 \leq R_{rt} \leq R_{rt}^{\max} \qquad \forall r,t$$

CHANGEOVER TIMES

$$\sum_{i \in I_j^f} W_{ijt} + \sum_{i \in I_j^{f'}} \sum_{t'=t-cl_{f'f}-pt_{ij}+1}^{t} W_{ijt'} \leq 1 \qquad \forall j, f, f', t$$

STN-BASED DISCRETE TIME FORMULATION

MAJOR ADVANTAGES

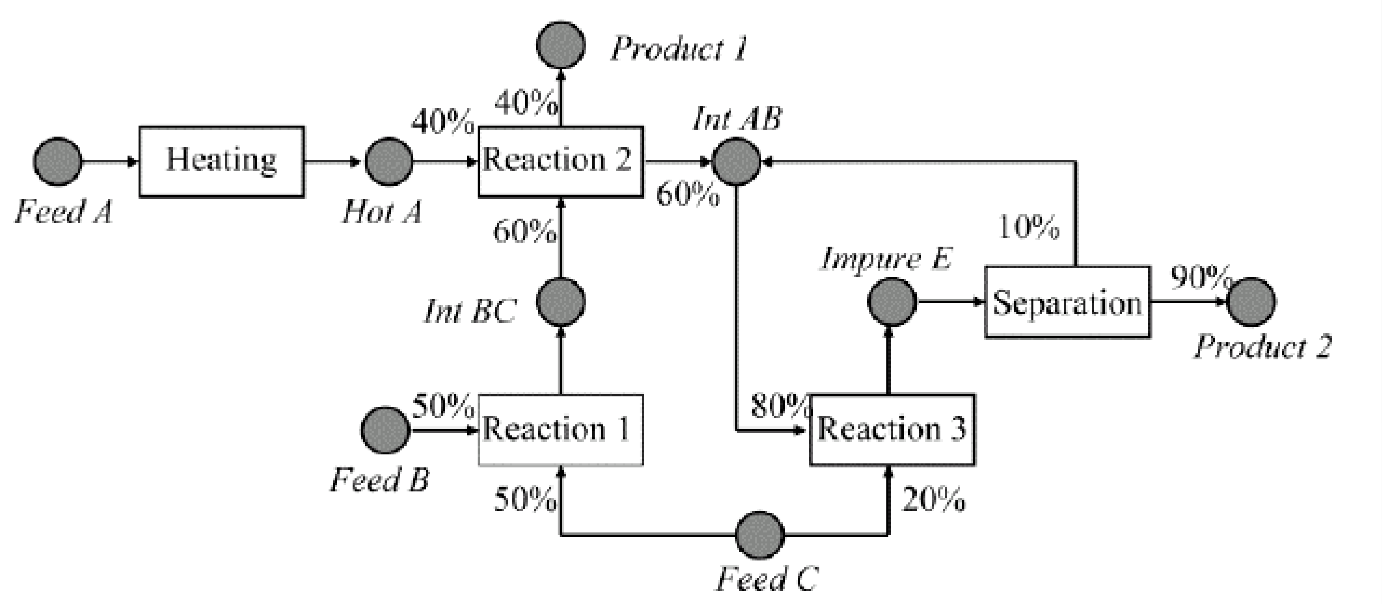
- Efficient handling of :
  - limited resource availabilities, only monitored at fixed, predefined time points
  - variable resource requirement along the task execution
  - other time-dependent aspects without compromising model linearity.
- Batch mixing and splitting are allowed
- No big-M constraint is required
- Good computational performance (lower integrality gap)
- Simple problem models accounting for a wide variety of scheduling features

MAJOR DISADVANTAGES

- Approximate processing times can lead to sub-optimal or infeasible solutions.
- The batch size B is a problem variable despite constant processing times.
- Handling of small sequence-dependent changeovers is rather awkward (very fine time discretization).
- Significant increase of the model size for longer time horizons

NETWORK-TYPE SCHEDULING EXAMPLE

STATE-TASK NETWORK REPRESENTATION (STN)



EQUIPMENT
-HEATER
- 2 REACTORS
-STILL

STN-BASED DISCRETE TIME FORMULATION

Task	Unit	Processing Time (h)
Heating	Heater	1
Reaction 1	Reactor 1 Reactor 2	2
Reaction 2		2
Reaction 3		1
Separation	Still	2

Number of Time Points: 10 (for H = 10 h)

Number of Binary Variables: 80

- **Heating: 1 x 1 x 10 = 10**
- **Reactions 1, 2 & 3: 3 x 2 x 10 = 60**
- **Separation: 1 x 1 x 10 = 10**

Number of Continuous Variables:

B (80), S (60)

Number of Constraints:

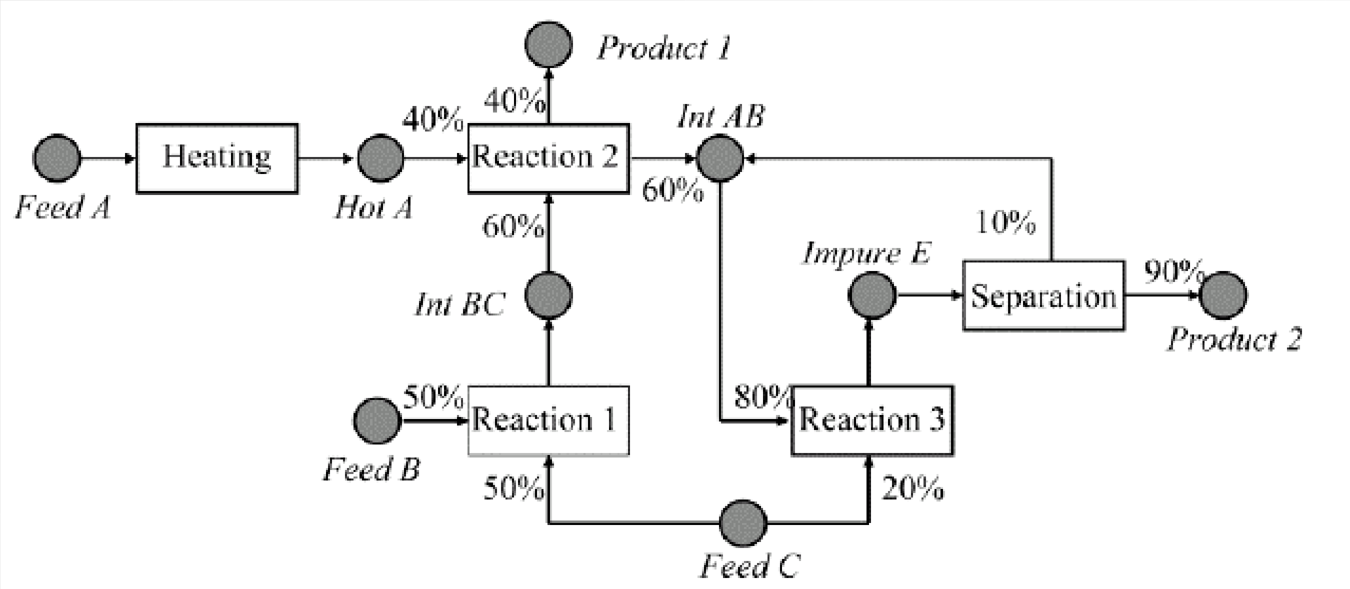
Allocation : 40 ; Batch size: 160

Material Balances: 60

Task	States Produced	States Consumed
Heating	Hot A	Raw Material A
Reaction 1	Int. AB (60%)+ P1 (40%)	Hot A (40%) + Int. BC (60%)
Reaction 2	Int. BC	Raw Materials B&C (50/50)
Reaction 3	Impure E	Raw Material C (20%) + Int. AB (80%)
Separation	P2 (90%) + Int. AB(10%)	Impure E

SCHEDULING EXAMPLE

STATE-TASK NETWORK REPRESENTATION (STN)



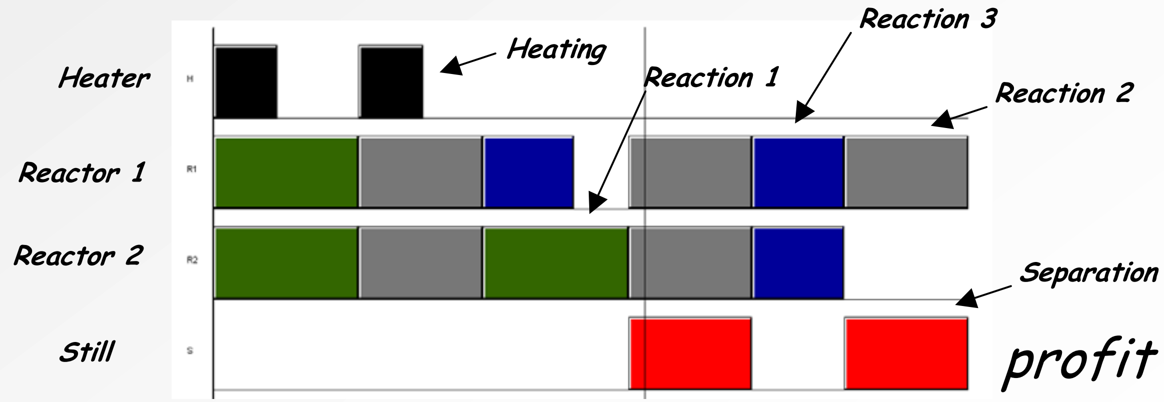
EQUIPMENT
-HEATER
- 2 REACTORS
-STILL

DECISIONS

Allocation

Sequencing

Timing



profit = 2805

RTN-BASED DISCRETE TIME FORMULATION

(Pantelides, 1994)

MAJOR FEATURES

- Similarly to the STN representation, it uses a predefined and fixed uniform time grid that is valid for all shared resources (global time intervals)
- Processing times are assumed to be independent of the batch size
- It is based on the Resource-Task-Network (RTN) concept
- All resources (equipment, materials, utilities) are treated in the same way
- Its major advantage with regards to the STN approach arises in problems involving identical equipment
- It requires to define just a single binary variable rather than multiple ones for a set of equipment units of similar type
- Each task can be allocated to just a single processing unit
- Task duplication is then required to handle alternative units and unit-dependent processing times
- Changeovers have to be considered as additional tasks.

RTN-BASED DISCRETE TIME FORMULATION

- **MAJOR MODEL VARIABLES**

a) **BINARY VARIABLES:** $W_{i\,t}$ (one less subscript)

b) **CONTINUOUS VARIABLES:** $B_{i\,t}$, $R_{i\,t}$

Since every task can be assigned to just a single unit, the subscript j can be eliminated.

- **MAJOR MODEL PARAMETERS**

$\mu_{i\,r\,t'}$ = **fixed amount of resource r produced/consumed by an instance of task i at time t' relative to the starting time interval t**

$\nu_{i\,r\,t'}$ = **coefficient in the term providing the amount of resource r produced / consumed by task i at time t' that is proportional to the batch size.**

When $r \in R$ stands for a processing unit, the meaning of parameters $\mu_{irt'}$ *and* $\nu_{irt'}$ is somewhat different.

RTN-BASED DISCRETE TIME FORMULATION

(Pantelides, 1994)

$$R_{rt} = R_{r(t-1)} + \sum_{i \in I_r} \sum_{t'=0}^{pt_i} \left(\mu_{irt'} W_{i(t-t')} + \nu_{irt'} B_{i(t-t')} \right) + \Pi_{rt} \qquad \forall r, t$$

$$0 \le R_{rt} \le R_{rt}^{\max} \qquad \forall r, t$$

RESOURCE BALANCE

$$V_{ir}^{\min} W_{it} \le B_{it} \le V_{ir}^{\max} W_{it} \qquad \forall i, r \in R_i^J, t$$

BATCH SIZE

If resource r corresponds to a processing unit and task i requires pt_i units of time, then:

$\mu_{irt'}$ = -1 for t' =0 (R_{rt} decreases by one if W_{it} = 1)

$\mu_{irt'}$ = +1 for t' =pt_i (R_{rt} increases by one if W_{it} = 1)

$\mu_{irt'}$ = 0 for any other t' (R_{rt} remains unchanged even if W_{it} = 1)

and: $\nu_{irt'}$ = 0 for any t'

RTN-BASED DISCRETE TIME FORMULATION

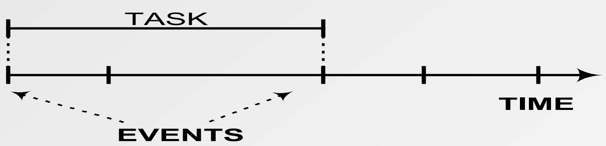
- MAJOR ADVANTAGES
  - Resource constraints are only monitored at predefined and fixed time points
  - All resources are treated in the same way
  - Saving in binary variables for problems involving identical equipment units
  - Efficient handling of limited resource availabilities
  - Good computational performance (lower integrality gap)
  - Very simple models and easy representation of a wide variety of scheduling features

- MAJOR DISADVANTAGES
  - Model size and complexity depend on the number of time intervals
  - Constant processing times independent of the batch size
  - Sub-optimal or infeasible solutions can be generated due to the use of approximate processing times
  - Changeovers have to be considered as additional tasks

NETWORK-TYPE GLOBAL TIME CONTINUOUS MODELS

(A) TIME DOMAIN REPRESENTATION

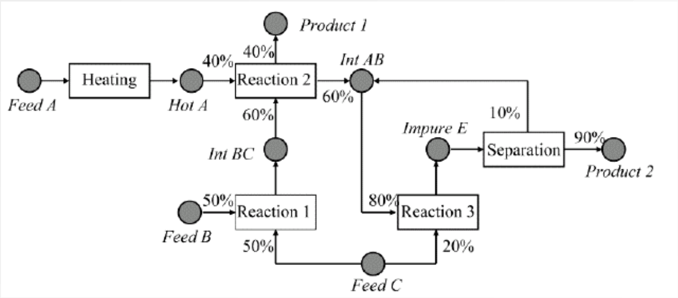
- CONTINUOUS TIME



(B) EVENT REPRESENTATION

CONTINUOUS TIME

- Global time points



(C) MATERIAL BALANCES

- Network flow equations (STN or RTN problem representation)

(D) OBJECTIVE FUNCTION

- Makespan
- Profit
- Cost

STN-BASED CONTINUOUS TIME FORMULATION

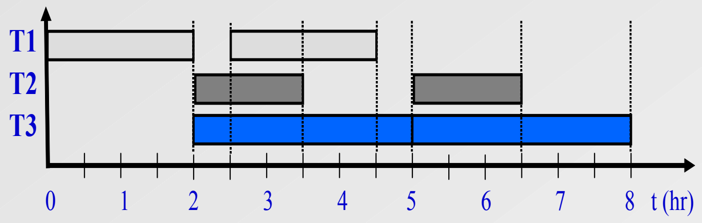
[Schilling & Pantelides (1996); Zhang & Sargent (1996); Mockus & Reklaitis (1999); Maravelias & Grossmann (2003)]

MAJOR FEATURES (Maravelias & Grossmann, 2003)

- A common time grid that is variable and valid for all shared resources (global time points)
- A predefined maximum number of time points (N) (a model parameter)
- The time points will occur at a priori unknown times (model decisions)
- Every event including the start and the end of a task must occur at a time point
- The start of several tasks can be assigned to the same time point *n* but at different units and, therefore, all must begin at the same time T_n.
- The end time of a task assigned to time point n does not necesarily occur exactly at T_n They can finish before except those tasks following a zero wait policy (ZW)
- For storage policies other than ZW, the equipment can be used as a temporary storage device from the end of the task to time T_n
- Each task can be allocated to just a single unit. Task duplication is required to handle alternative equipment units

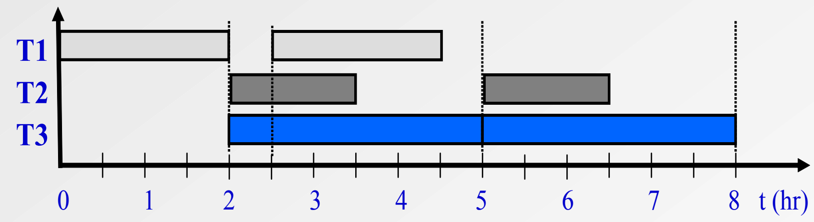
STN-BASED CONTINUOUS TIME FORMULATION

Continuous Time Representation I



Schilling & Pantelides, 1996

Continuous Time Representation II



Maravelias & Grossmann, 2003

- **MAJOR PROBLEM VARIABLES**

a) **BINARY VARIABLES:**

Ws_{in} = denotes allocation of the start of task i to time point n

Wf_{in} = denotes allocation of the end of task i to time point n

b) **CONTINUOUS VARIABLES:**

T_n = time for events allocated to time point n

$Tf_{i\,n}$ = end time of task i assigned to time point n

$Ts_{i\,n}$ = start time of task i assigned to time point n

$Bs_{i\,n}$ = batch size of task i at the start time point n

$Bp_{i\,n}$ = batch size of task i at the intermediate time point n

$Bf_{i\,n}$ = batch size of task i at the completion time point n

$S_{s\,n}$ = inventory of state s at time point n

$R_{r\,n}$ = availability of resource r at time point n

STN-BASED UNIT-SPECIFIC TIME-EVENT FORMULATION

MAJOR CONSTRAINTS

- **ALLOCATION CONSTRAINTS:**
  - At most a single task can be performed in unit j at the event time n
  - A task will be active at event time n only if it stars before or at event time n, and it finishes before time event n
  - All tasks that start must finish
  - An occurrence of task i can be started at event point n only if all previous instances of task i beginning earlier have finished before n
  - An occurrence of task i can finish at event point n only if it starts before n and ends not before time point n
- **BATCH SIZE CONSTRAINTS**
- **MATERIAL BALANCES**
- **TIMING AND SEQUENCING CONSTRAINTS**
- **STORAGE CONSTRAINTS**
- **RESOURCE CONSTRAINTS**

STN-BASED CONTINUOUS TIME FORMULATION
(GLOBAL TIME POINTS)

(Maravelias and Grossmann, 2003)

ALLOCATION CONSTRAINTS

$$\sum_{i \in I_j} Ws_{in} \le 1 \qquad \forall j,n$$

$$\sum_{i \in Ij} Wf_{in} \le 1 \qquad \forall j,n$$

$$\sum_{i \in Ij} \sum_{n' \le n} (Ws_{in'} - Wf_{in'}) \le 1 \qquad \forall j,n$$

$$\sum_{n} Ws_{in} = \sum_{n} Wf_{in} \qquad \forall i$$

BATCH SIZE CONSTRAINTS

$$V_i^{\min} Ws_{in} \le Bs_{in} \le V_i^{\max} Ws_{in} \qquad \forall i,n$$

$$V_i^{\min} Wf_{in} \le Bf_{in} \le V_i^{\max} Wf_{in} \qquad \forall i,n$$

$$V_i^{\min}\left(\sum_{n'<n} Ws_{in'} - \sum_{n' \le n} Wf_{in'}\right) \le Bp_{in} \le V_i^{\max}\left(\sum_{n'<n} Ws_{in'} - \sum_{n' \le n} Wf_{in'}\right) \qquad \forall i,n$$

$$Bs_{in-1} + Bp_{i(n-1)} = Bp_{in} + Bf_{in} \qquad \forall i,n>1$$

MATERIAL AND RESOURCE BALANCES

$$S_{sn} = S_{s(n-1)} - \sum_{i \in I_s^c} \rho_{is}^c Bs_{in} + \sum_{i \in I_s^p} \rho_{is}^p Bf_{in} \qquad \forall s,n>1$$

$$S_{sn} \le C_s^{\max} \qquad \forall s,n$$

$$R_{rn} = R_{r(n-1)} - \sum_{i} \mu_{ir}^c Ws_{in} + \nu_{ir}^c Bs_{in} + \sum_{i} \mu_{ir}^p Wf_{in} + \nu_{ir}^p Bf_{in} \qquad \forall r,n$$

TIMING AND SEQUENCING CONSTRAINTS

$$T_{n+1} \ge T_n \qquad \forall n$$

$$Tf_{in} \le T_n + \alpha_i Ws_{in} + \beta_i Bs_{in} + H(1 - Ws_{in}) \qquad \forall i,n$$

$$Tf_{in} \ge T_n + \alpha_i Ws_{in} + \beta_i Bs_{in} - H(1 - Ws_{in}) \qquad \forall i,n$$

$$Tf_{i(n-1)} \le T_n + H(1 - Wf_{in}) \qquad \forall i,n>1$$

$$Tf_{i(n-1)} \ge T_n - H(1 - Wf_{in}) \qquad \forall i \in I^{ZW}, n>1$$

$$Ts_{i'n} \ge Tf_{i(n-1)} + cl_{ii'} \qquad \forall j, i \in I_j, i' \in I_j, n$$

SHARED STORAGE TASKS

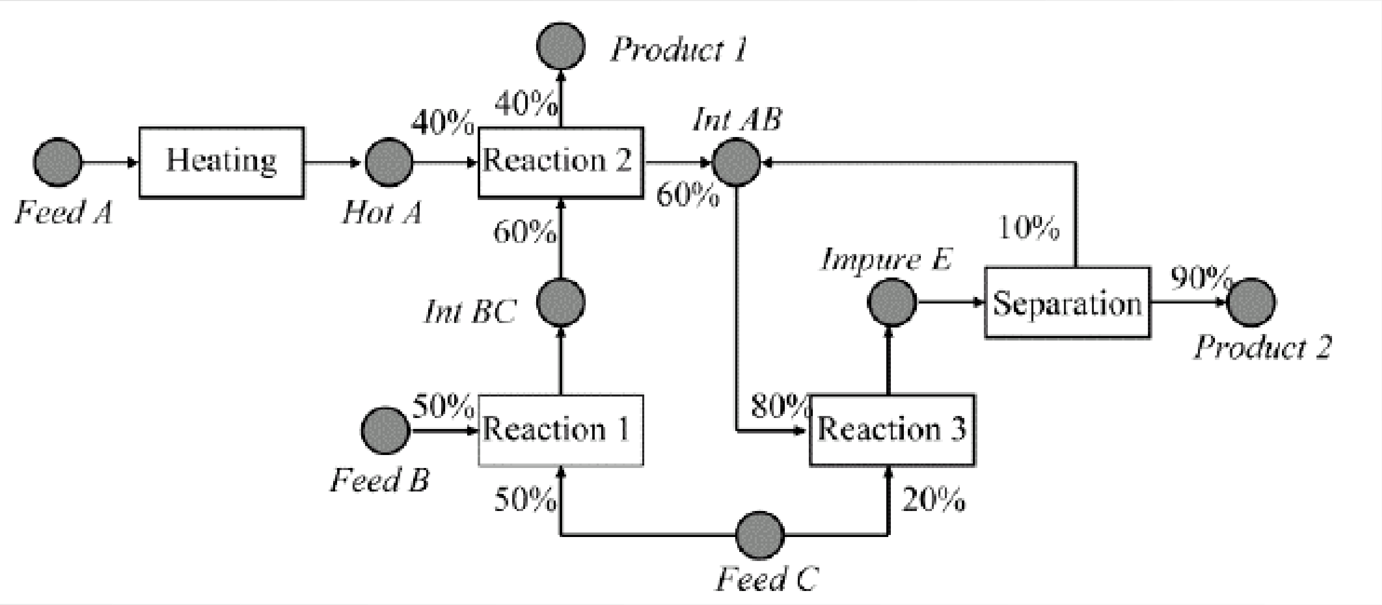
$$\sum_{s \in S_j} V_{jsn} \le 1 \qquad \forall j \in J^T, n$$

$$S_{sjn} \le C_j V_{jsn} \qquad \forall j \in J^T, s \in S_j, n$$

$$S_{sn} = \sum_{j \in J_s^T} S_{sjn} \qquad \forall s \in S^T, n$$

STN-BASED CONTINUOUS TIME FORMULATION

EXAMPLE



Problem Size (8 effective tasks, 8 time points) **it requires less time points**

a) Binary Variables: Ws (64) + Wf (64) = 128

b) Continuous Variables: T(64) + Tf(64) + Bs (64) + Bf (64) + Bp (64) + S (48) = 368

c) Constraints: Allocat (104) + BSize (568) + Time (264) + InvS (56) = 992

STN-BASED CONTINUOUS TIME FORMULATION

- **MAJOR ADVANTAGES**

  - Significant reduction in model size by predefining a minimum number of time points much lower than that required by discrete formulations
  - Handling of processing times which vary with the batch size
  - Consideration of a range of scheduling aspects
  - Monitoring of resource availabilities just at the time points

- **MAJOR DISADVANTAGES**

  - Need of computing the minimum number of time points
  - Model size and complexity both depending on the number of predefined time points
  - Suboptimal or infeasible schedules can be generated if the number of points is smaller than required

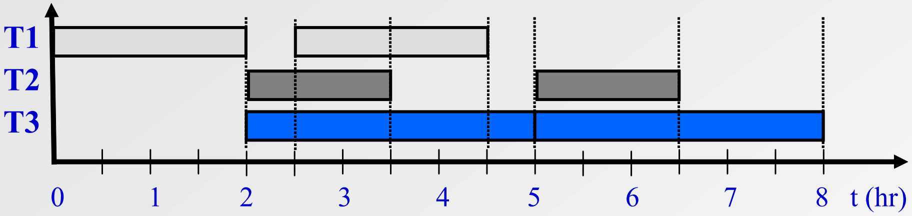
RTN-BASED CONTINUOUS TIME FORMULATION

(Castro et al., (2001, 2004))

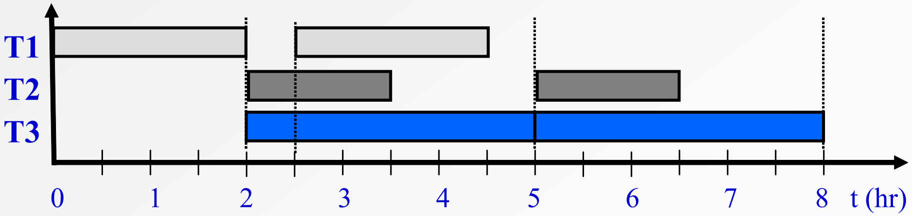
MAIN ASSUMPTIONS

- A common time grid for all shared resources
- The maximum number of time points is predefined
- The time at which each time point occurs is a model decision (continuous time domain)
- Tasks allocated to a certain time point *n* must start at the same time
- Only zero wait tasks must finish at a time point, others may finish before

Continuous Time Representation I



Continuous Time Representation II



ADVANTAGES

- Significant reduction in model size when the minimum number of time points is predefined
- Variable processing times
- Resource constraints are only monitored at each time point
- A wide variety of scheduling aspects can be considered in a very simple model

DISADVANTAGES

- Definition of the minimum number of time points
- Model size and complexity depend on the number of time points predefined
- Sub-optimal or infeasible solution can be generated if the number of time points is smaller than required

RTN-BASED CONTINUOUS FORMULATION
(GLOBAL TIME POINTS)

(Castro et al., 2004)

TIMING CONSTRAINTS

$$T_{n'} - T_n \geq \sum_{i \in I_r} \left(\alpha_i W_{inn'} + \beta_i B_{inn'} \right) \qquad \forall r \in R^J, n, n', (n < n')$$

$$T_{n'} - T_n \leq H \left(1 - \sum_{i \in I_r^{ZW}} W_{inn'} \right) + \sum_{i \in I_r^{ZW}} \left(\alpha_i W_{inn'} + \beta_i B_{inn'} \right) \qquad \forall r \in R^J, n, n', (n < n')$$

BATCH SIZE

$$V_i^{\min} W_{inn'} \leq B_{inn'} \leq V_i^{\max} W_{inn'} \qquad \forall i, n, n', (n < n')$$

RESOURCE BALANCE

$$R_{rn} = R_{r(n-1)} + \sum_{i \in I_r} \left[\sum_{n' < n} \left(\mu_{ir}^p W_{in'n} + \nu_{ir}^p B_{in'n} \right) - \sum_{n' > n} \left(\mu_{ir}^c W_{inn'} + \nu_{ir}^c B_{inn'} \right) \right] + \sum_{i \in I^S} \left(\mu_{ir}^p W_{i(n-1)n} - \mu_{ir}^c W_{i(n+1)} \right) \qquad \forall r, n > 1$$

$$R_r^{\min} \leq R_{rn} \leq R_r^{\max} \qquad \forall r, n$$

STORAGE CONSTRAINTS

$$V_i^{\min} W_{in(n+1)} \leq \sum_{r \in R_i^S} R_{rt} \leq V_i^{\max} W_{in(n+1)} \qquad \forall i \in I^s, n, (n \neq |N|)$$

$$V_i^{\min} W_{i(n-1)n} \leq \sum_{r \in R_i^S} R_{rt} \leq V_i^{\max} W_{i(n-1)n} \qquad \forall i \in I^s, n, (n \neq 1)$$

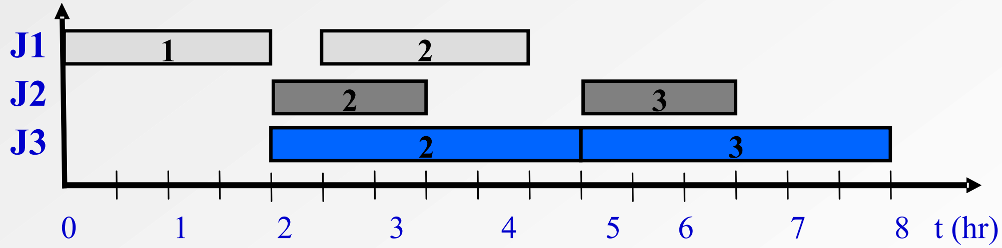
STN-BASED UNIT-SPECIFIC TIME-EVENT FORMULATION

(Ierapetritou and Floudas, 1998; Vin and Ierapetritou, 2000; Lin et al., 2002; Janak et al., 2004).

MAIN ASSUMPTIONS

- It is a STN-based formulation but the global time representation has been relaxed
- Different tasks assigned to the same event point but performed in different units can be started/finished at different times
- The number of event points is predefined (a model parameter)
- The time points will occur at a priori unknown times (model decisions)
- The start and the end of a task must occur at an event point
- Each task can be allocated to just a single unit. Task duplication is required to handle alternative equipment units
- It considers processing tasks i and storage tasks i^{st}

Event-Based Representation



STN-BASED UNIT-SPECIFIC TIME-EVENT FORMULATION

(Janak et al., 2004)

- **MAJOR PROBLEM VARIABLES**

A. **BINARY VARIABLES:** $Ws_{i\,n}$ (start) , $W_{i\,n}$ (active) , $Wf_{i\,n}$ (end)

8 tasks x 10 event points → 8 x 10 x 3 = 240

B. **CONTINUOUS VARIABLES:** $Bs_{i\,n}$ (start) , $B_{i\,n}$ (active) , $Bf_{i\,n}$ (end)

$Ts_{i\,n}$, $Tf_{i\,n}$, $Ts_{r\,n}$, $Tf_{r\,n}$, $S_{s\,n}$, $R_{i\,r\,n}$, $R^{A}_{r\,n}$

8 tasks x 10 event points x 1 resource → (8 x 10 x 6) + 30 = 510
6 states x 10 event points → 6 x 10 = 60 570

STN-BASED UNIT-SPECIFIC TIME-EVENT FORMULATION

MAJOR CONSTRAINTS

- **ALLOCATION CONSTRAINTS:**
  - At most a single task can be performed in unit j at the event time n
  - A task will be active at event time n only if it stars before or at event time n, and it finishes before time event n
  - All tasks that start must finish
  - An occurrence of task i can be started at event point n only if all previous instances of task i beginning earlier have finished before n
  - An occurrence of task i can finish at event point n only if it starts before n and ends not before time point n

- **BATCH SIZE CONSTRAINTS**
- **MATERIAL BALANCES**
- **TIMING AND SEQUENCING CONSTRAINTS**
- **STORAGE CONSTRAINTS**
- **RESOURCE CONSTRAINTS**

STN-BASED CONTINUOUS FORMULATION
(UNIT-SPECIFIC TIME EVENT)

(Janak et al., 2004)

ALLOCATION CONSTRAINTS

$$\sum_{i \in I_j} W_{in} \leq 1 \quad \forall j, n$$

$$\sum_{n' \leq n} Ws_{in'} - \sum_{n' < n} Wf_{in'} = W_{in} \quad \forall i, n$$

$$\sum_{n} Ws_{in} = \sum_{n} Wf_{in} \quad \forall i$$

$$\sum_{n} Ws_{in} \leq 1 - \sum_{n' < n} Ws_{in'} + \sum_{n' < n} Wf_{in'} \quad \forall i, n$$

$$Wf_{in} \leq \sum_{n' < n} Ws_{in'} - \sum_{n' < n} Wf_{in'} \quad \forall i, n$$

BATCH SIZE CONSTRAINTS

$$V_i^{min} W_{in} \leq B_{in} \leq V_i^{max} W_{in} \quad \forall i, n$$

$$B_{in} \leq B_{i(n-1)} - V_i^{\max}\left(1 - W_{i(n-1)} + Wf_{i(n-1)}\right) \quad \forall i, n > 1$$

$$B_{in} \geq B_{i(n-1)} - V_i^{\max}\left(1 - W_{i(n-1)} + Wf_{i(n-1)}\right) \quad \forall i, n > 1$$

$$Bs_{in} \leq B_{in} \quad \forall i, n$$

$$Bs_{in} \leq B_{in} + V_i^{\max} Ws_{in} \quad \forall i, n$$

$$Bs_{in} \geq B_{in} - V_i^{\max}\left(1 - Ws_{in}\right) \quad \forall i, n$$

$$Bf_{in} \leq B_{in} \quad \forall i, n$$

$$Bf_{in} \leq B_{in} + V_i^{\max} Wf_{in} \quad \forall i, n$$

$$Bf_{in} \geq B_{in} - V_i^{\max}\left(1 - Wf_{in}\right) \quad \forall i, n$$

MATERIAL BALANCE

$$S_{sn} = S_{s(n-1)} + \sum_{i \in I_s^p} \rho_{is}^p Bf_{i(n-1)} + \sum_{i^{st} \in I_s^{ST}} B_{i^{st}(n-1)} - \sum_{i \in I_s^c} \rho_{is}^c Bs_{in} - \sum_{i^{st} \in I_s^{st}} B_{i^{st}n} \quad \forall s, n$$

STORAGE CAPACITY

$$B_{i^{st}n} \leq C_s^{\max} \quad \forall s, i^{st} \in I_s^{st}, n$$

STN-BASED UNIT-SPECIFIC TIME EVENT FORMULATION

TIMING AND SEQUENCING CONSTRAINTS (PROCESSING TASKS)

$$Tf_{in} \geq Ts_{in} \qquad \forall i,n$$

$$Tf_{in} \leq Ts_{in} + H\,W_{in} \qquad \forall i,n$$

$$Ts_{in} \leq Tf_{i(n-1)} + H\left(1 - W_{i(n-1)} + Wf_{i(n-1)}\right) \qquad \forall i, n>1$$

$$Tf_{in'} - Ts_{in} \geq \alpha_i Ws_{in} + \beta_i B_{in} + H\left(1 - Ws_{in}\right) + H\left(1 - Wf_{in'}\right) + H\left(\sum_{n \leq n'' \leq n'} Wf_{in''}\right) \qquad \forall i,n,n',(n \leq n')$$

$$Tf_{in'} - Ts_{in} \leq \alpha_i Ws_{in} + \beta_i B_{in} + H\left(1 - Ws_{in}\right) + H\left(1 - Wf_{in'}\right) + H\left(\sum_{n \leq n'' \leq n'} Wf_{in''}\right) \qquad \forall i \in I^{ZW},n,n',(n \leq n')$$

$$Ts_{in} \geq Tf_{i(n-1)} \qquad \forall i, n>1$$

$$Ts_{in} \geq Tf_{i'(n-1)} + cl_{i'i} + H\left(1 - Wf_{i'(n-1)} - Ws_{in}\right) \qquad \forall i,i', i \neq i', j \in J_{ii'}, n>1$$

$$Ts_{in} \geq Tf_{i'(n-1)} + H\left(1 - Wf_{i'(n-1)}\right) \qquad \forall s, i \in I_s^c, i' \in I_s^p, j \in J_i, j' \in J_{i'}, j \neq j', n>1$$

$$Ts_{in} \leq Tf_{i'(n-1)} + H\left(2 - Wf_{i'(n-1)} - Ws_{in}\right) \qquad \forall s \in S^{ZW}, i \in I_s^c, i' \in I_s^p, j \in J_i, j' \in J_{i'}, j \neq j', n>1$$

STN-BASED UNIT-SPECIFIC TIME-EVENT FORMULATION

$$Tf_{i^{st}n} \geq Ts_{i^{st}n} \qquad \forall i^{st}, n$$
$$Ts_{i^{st}n} \geq Tf_{i(n-1)} - H\left(1 - Wf_{i(n-1)}\right) \qquad \forall s, i \in I_s^p, i^{st} \in I_s^{ST}, n > 1$$
$$Ts_{i^{st}n} \leq Tf_{i(n-1)} + H\left(1 - Wf_{i(n-1)}\right) \qquad \forall s, i \in I_s^p, i^{st} \in I_s^{ST}, n > 1$$
$$Ts_{in} \geq Tf_{i^{st}(n-1)} \qquad \forall s, i \in I_s^c, i^{st} \in I_s^{ST}, n > 1$$
$$Ts_{in} \leq Tf_{i^{st}(n-1)} + H\left(1 - Ws_{in}\right) \qquad \forall s, i \in I_s^c, i^{st} \in I_s^{ST}, n > 1$$
$$Ts_{i^{st}n} = Tf_{i^{st}(n-1)} \qquad \forall i^{st}, n > 1$$

TIMING AND SEQUENCING CONSTRAINTS (STORAGE TASKS)

$$R_{irn} = \mu_{ir}^c W_{in} + \nu_{ir}^c B_{in} \qquad \forall r, i \in I_r, n$$
$$\sum_{i \in I_r} R_{irn} + R_{rn}^A = R_r^{\max} \qquad \forall r, n = 1$$
$$\sum_{i \in I_r} R_{irn} + R_{rn}^A = \sum_{i \in I_r} R_{ir(n-1)} + R_{r(n-1)}^A \qquad \forall r, n > 1$$

RESOURCE BALANCE

$$Tf_{rn} \geq Ts_{rn} \qquad \forall r, n$$
$$Tf_{i(n-1)} \geq Ts_{rn} - H\left(1 - W_{i(n-1)} + Wf_{i(n-1)}\right) \qquad \forall r, i \in I_r, n > 1$$
$$Tf_{i(n-1)} \leq Ts_{rn} - H\left(1 - W_{i(n-1)}\right) \qquad \forall r, i \in I_r, n > 1$$
$$Ts_{rn} \geq Ts_{in} - H\left(1 - W_{in}\right) \qquad \forall r, i \in I_r, n$$
$$Ts_{rn} \leq Ts_{in} + H\left(1 - W_{in}\right) \qquad \forall r, i \in I_r, n$$
$$Ts_{rn} = Tf_{r(n-1)} \qquad \forall r, n > 1$$

TIMING AND SEQUENCING OF RESOURCE USAGE

STN-BASED DISCRETE TIME FORMULATION

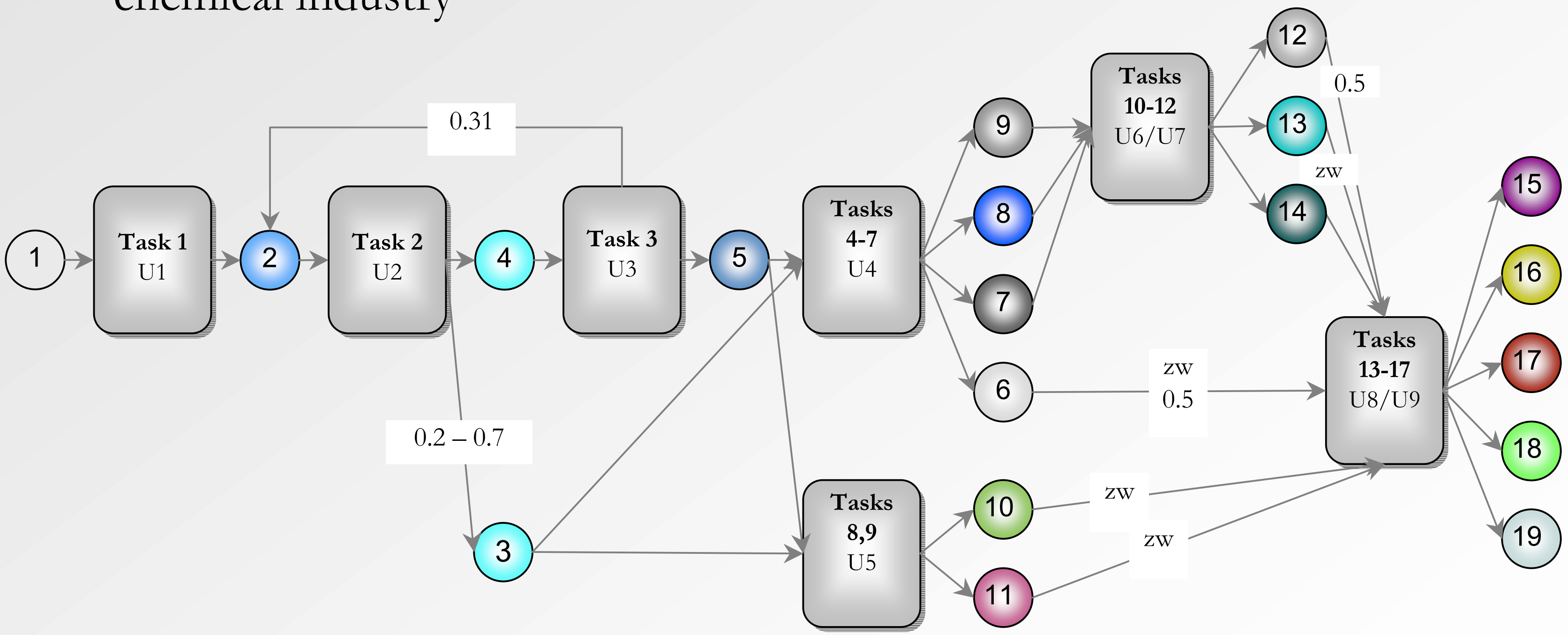
- **MAJOR ADVANTAGES**
  - More flexible time decisions
  - Less number of event points

- **MAJOR DISADVANTAGES**
  - Definition of event points
  - More complicated models
  - Model size and complexity depend on the number of time points predefined
  - Sub-optimal or infeasible solution can be generated if the number of time points is smaller than required
  - Additional tasks for storage and utilities

COMPARISON OF DISCRETE AND STN CONTINUOUS TIME FORMULATIONS

CASE STUDY: Westenberger & Kallrath (1995)

Benchmark problem for production scheduling in chemical industry



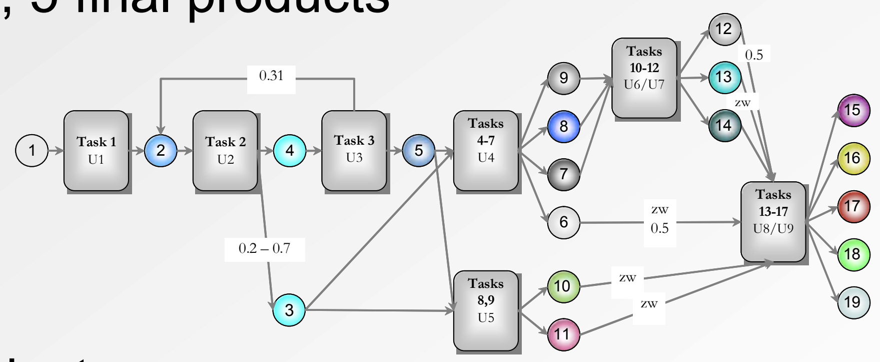
COMPARISON OF DISCRETE AND STN CONTINUOUS TIME FORMULATIONS

PROBLEM FEATURES

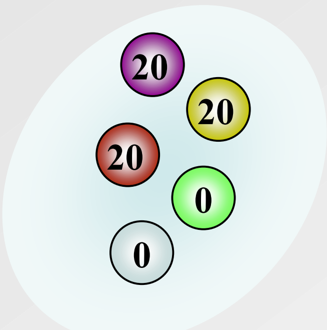
- The process includes flexible proportions of output states 3 & 4 (task 2), material recycles from task 3, and five final states (S15, S16, S17, S18, S19)
- There is enough stock of raw material (S1) and unlimited storage for the required raw material (S1) and the final products (S15-S19).
- Different intermediate storage modes are considered:
  - Zero-Wait transfer policy for states (S6, S10, S11, S13)
  - Finite dedicated intermediate storage (FIS) policy for the other intermediate states
- Problem data involves only integer processing times
- Two alternative problem objectives are considered:
  - Minimizing makespan (Case I)
  - Maximizing profit (Case II)
- **Options A,B:** product demands of 20 tons just for three final states have to be satisfied
- **Option C:** minimum product demands of (10, 10, 10, 5,10) tons for states (15, 16, 17, 18 and 19)

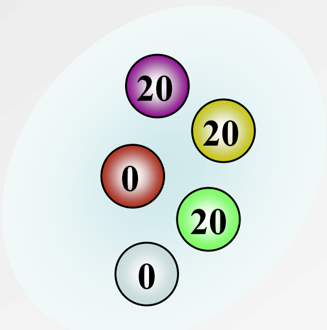
SUMMARY OF PROBLEM FEATURES

- 17 processing tasks, 19 states, 5 final products
- 9 production units
- 37 material flows
- Batch mixing / splitting
- Cyclical material flows
- Flexible output proportions
- Non-storable intermediate products
- No initial stock of final products
- Unlimited storage for raw material and final products
- Sequence-dependent changeover times



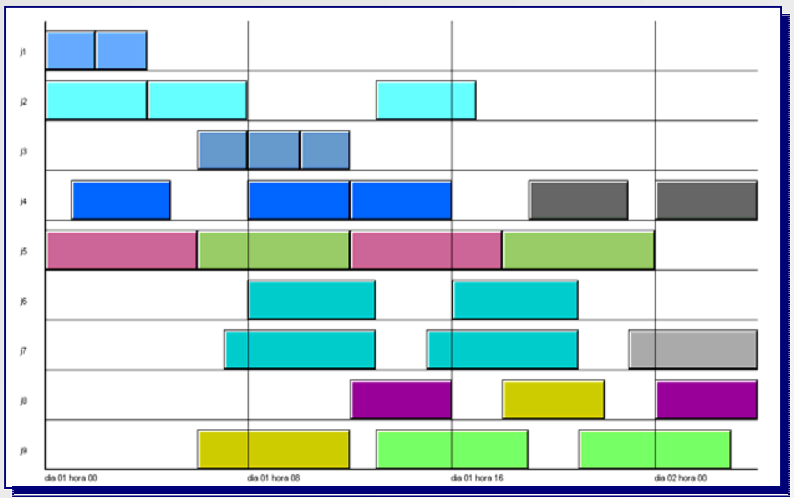
CASE I: MAKESPAN MINIMIZATION





Instance	*A*			*B*		
Formulation	*Discrete*	*Continuous*		*Discrete*	*Continuous*	
time points	30	8	9	30	7	8
binary variables	720	384	432	720	336	384
continuous variables	3542	2258	2540	3542	1976	2258
constraints	6713	4962	5585	6713	4343	4964
LP relaxation	9.9	24.2	24.1	9.9	25.2	24.3
objective	28	28	**28**	28	32	30
iterations	728	78082	27148	2276	58979	2815823
nodes	10	1180	470	25	1690	63855
CPU time (s)	1.34	108.39	51.41	4.41	66.45	3600.21
relative gap	0.0	0.0	0.0	0.0	0.0	0.067

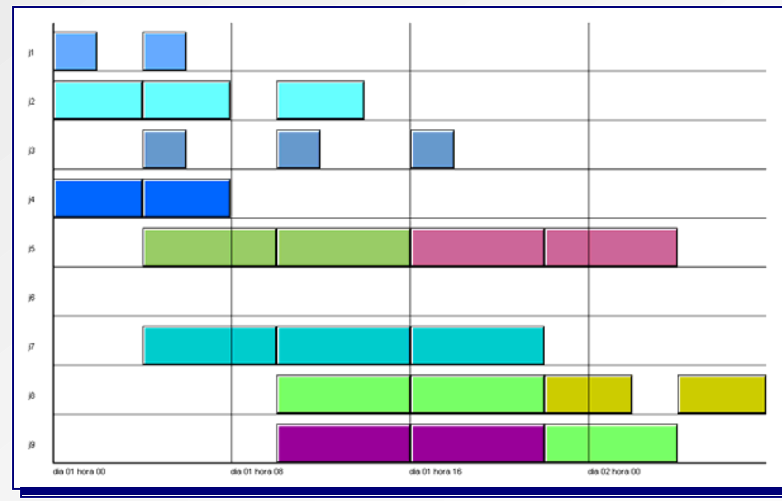
CASE I.B: MAKESPAN MINIMIZATION



Discrete model

Time intervals: 30

Makespan: 28



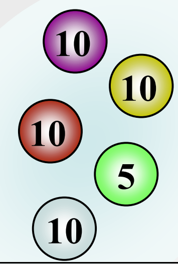
Continuous model

Time points: 7

Makespan: 32

CASE II.C : PROFIT MAXIMIZATION

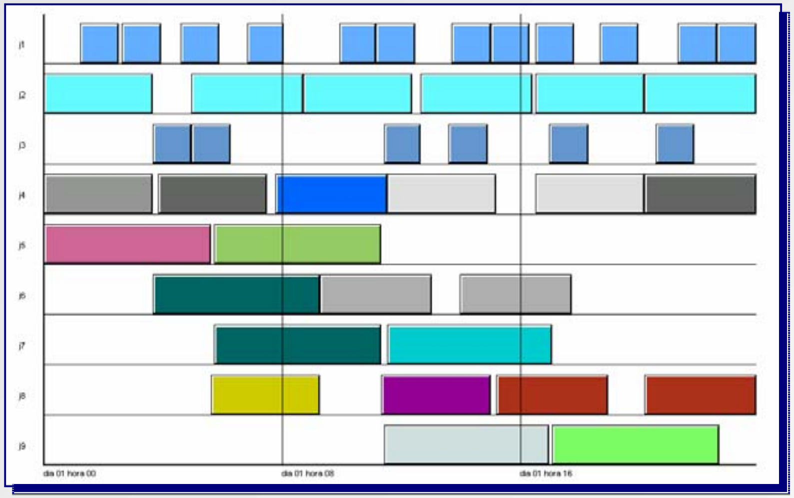
- H = 24 h



Formulation	Instance D: Discrete	Discrete LB	Discrete UB	Continuous
time points	240	24	24	14
binary variables	5760	576	576	672
continuous variables	28322	2834	2834	3950
constraints	47851	4794	4799	8476
LP relaxation	1769.9	1383.0	2070.9	1647.1
objective	1425.8	**1184.2**	**1721.8**	1407.4
iterations	449765	3133	99692	256271
nodes	5580	203	4384	1920
CPU time (s)	7202	6.41	58.32	258.54
relative gap	0.122	0.047	0.050	0.042

CASE II.C : PROFIT MAXIMIZATION

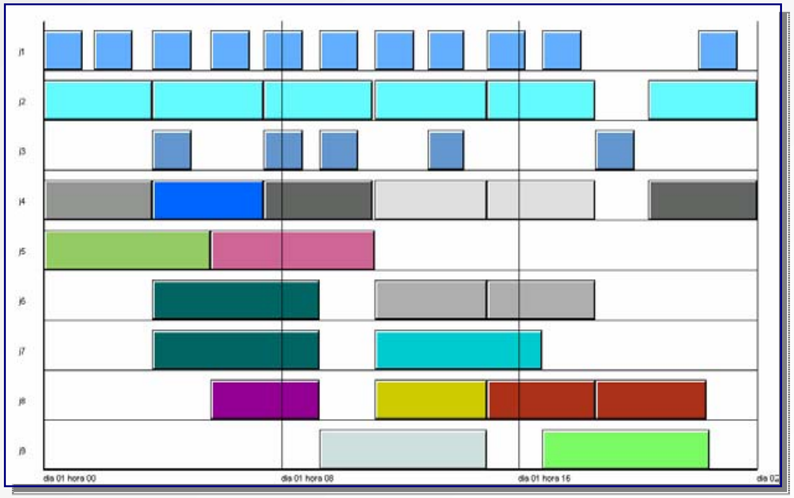
- H = 24 h



Discrete model

Time intervals: 240

Profit: **1425.8**



Continuous model

Time points: 14

Profit: **1407.4**

COMPARISON OF DISCRETE AND STN-CONTINUOUS FORMULATIONS

COMMENTS

- For Case I, instances comprising a larger number of demands were not possible to solve in a reasonable time
- Case I - Minimizing makespan:
  - Both formulations reach the same objective value of 28 h
  - 30 time points for the discrete model vs. 8 points for the continuous formulation
  - 1.34 s (discrete model) vs. 108 s (continuous model)
  - The number of time points is increased by one in each iteration until no improvement is achieved and the reported CPU time corresponds to the last iteration
- Case II - Maximizing profit:
  - A fixed horizon length of 24 hours was defined (longer periods cannot be solved in a reasonable time)
  - 240 time points for the discrete model vs. 14 points for the continuous formulation
  - The solution found through the discrete time model was slightly better
  - With 14 points the continuous approach is faster

COMPARING DISCRETE VS. STN-BASED CONTINUOUS MODELS

- **SOME PRELIMINARY CONCLUSIONS**

(1) Discrete time formulations are usually larger, but its simpler model structure tends to reduce the CPU time if a reasonable number of time points is proposed.

(2) Discrete time models may generate better and faster solutions than the continuous ones whenever the time discretization is a good approximation to the real data.

(3) The complex structure of continuous time models makes them useful only for problems that can be solved with a reduced number of time points (less than 15 time points).

(4) The model objective function selected may have a notable influence on the computational cost.

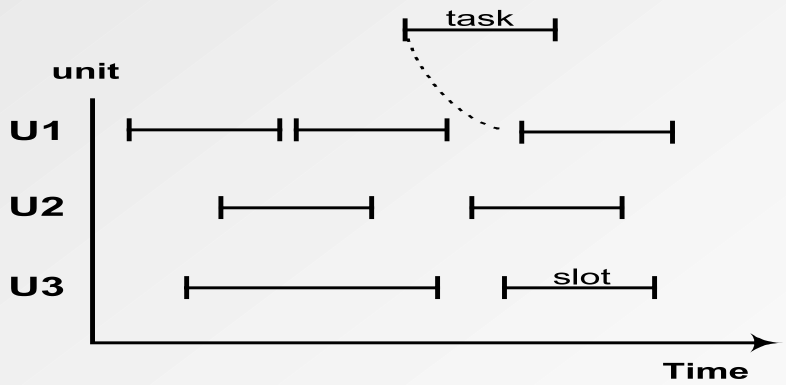
(5) Serious limitations for solving large-scale problem instances requiring a large number of fixed/variable time points were observed.

SLOT-BASED UNIT-SPECIFIC CONTINUOUS TIME FORMULATION

(Pinto and Grossmann (1995, 1996); Chen et. al. ,2002; Lim and Karimi, 2003)

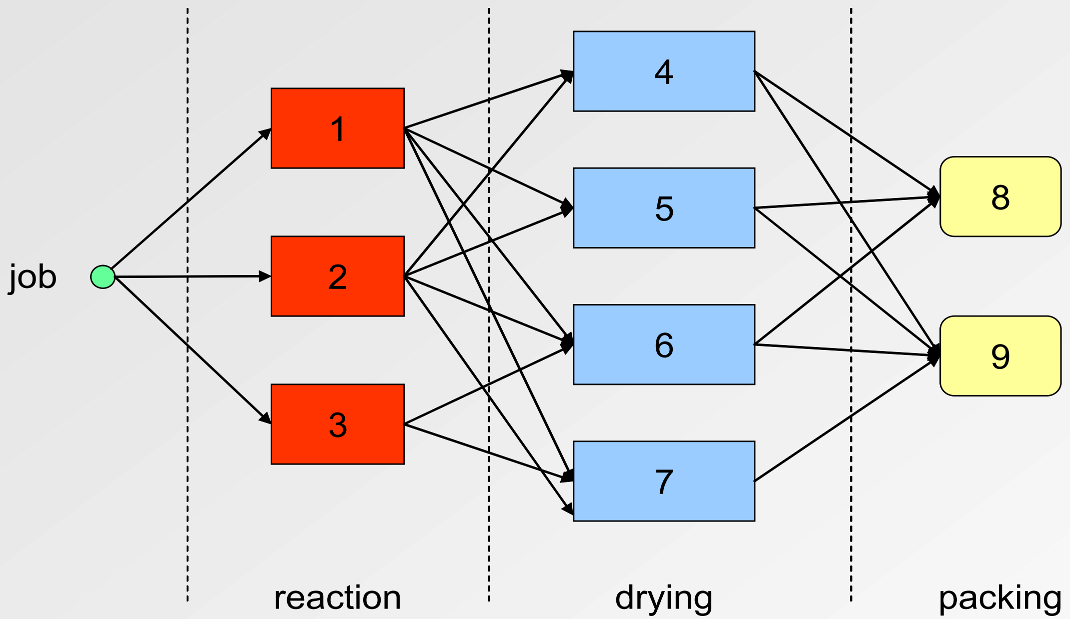
MAIN ASSUMPTIONS

- One of the first contributions on batch-oriented scheduling methodologies
- The notion of time slots stands for a set of predefined time intervals of unknown duration
- A different set of time slots is predefined for each processing unit
- Batches to be scheduled are defined a priori (problem data)
- Every batch is to be allocated to at most a single time slot
- No mixing and splitting operations are allowed
- It can be applied to a multistage sequential process with several parallel units at each stage
- Batches can start and finish at any time during the scheduling horizon



SLOT-BASED UNIT-SPECIFIC CONTINUOUS FORMULATION

A MULTISTAGE SEQUENTIAL BATCH PROCESS



- Neither the batch sizes nor the equipment capacities are model parameters
- A batch size feasibility test is not required
- Only batch processing times and setup times for each product at each stage are problem data
- Batch processing times can vary with the selected equipment unit

COMPARISON OF DISCRETE AND STN CONTINUOUS TIME FORMULATIONS

PROBLEM FEATURES

- The process includes flexible proportions of output states 3 & 4 (task 2), material recycles from task 3, and five final states (S15, S16, S17, S18, S19)
- There is enough stock of raw material (S1) and unlimited storage for the required raw material (S1) and the final products (S15-S19).
- Different intermediate storage modes are considered:
  - Zero-Wait transfer policy for states (S6, S10, S11, S13)
  - Finite dedicated intermediate storage (FIS) policy for the other intermediate states
- Problem data involves only integer processing times
- Two alternative problem objectives are considered:
  - Minimizing makespan (Case I)
  - Maximizing profit (Case II)
- **Options A,B:** product demands of 20 tons just for three final states have to be satisfied
- **Option C:** minimum product demands of (10, 10, 10, 5,10) tons for states (15, 16, 17, 18 and 19)

SLOT-BASED UNIT-SPECIFIC CONTINUOUS FORMULATION

- **MAJOR CONSTRAINTS**

BATCH ALLOCATION:

- The stage l of batch i must be allocated to just a single time slot

SLOT ALLOCATION:

- A time slot (j,k) can at most be assigned to a single task (stage l of batch i)

MATCHING CONSTRAINTS:

- If task (i,l) has been assigned to slot (j,k), then the start/end time of task (i,l) and the start/end time of slot (j,k) must be the same

SLOT SEQUENCING:

- The slot (k+1) at every unit j cannot be started before ending the slot (j,k). No overlap of time slots is permitted

STAGE SEQUENCING:

- The processing stage l+1 on batch i cannot be started before completing stage l

SLOT TIMING:

- The duration of slot (j,k) is given by the sum of the processing time & the setup time for the assigned task (i,l) , if $W_{ijkl} = 1$.

TIME-SLOT CONTINUOUS TIME FORMULATION

(Pinto and Grossmann (1995)

$$\sum_{j}\sum_{k\in K_j} W_{ijkl} = 1 \quad \forall i, l \in L_i$$

BATCH ALLOCATION

$$\sum_{i}\sum_{l\in L_i} W_{ijkl} \le 1 \quad \forall j, k \in K_j$$

SLOT ALLOCATION

$$Tf_{jk} = Ts_{jk} + \sum_{i}\sum_{l\in L_i} W_{ijkl}\left(p_{ijl} + su_{ijl}\right) \quad \forall j, k \in K_j$$

SLOT TIMING

$$Tf_{il} = Ts_{il} + \sum_{j}\sum_{k\in K_j} W_{ijkl}\left(p_{ijl} + su_{ijl}\right) \quad \forall i, l \in L_i$$

BATCH TIMING

$$Tf_{jk} \le Ts_{j(k+1)} \quad \forall j, k \in K_j$$

SLOT SEQUENCING

$$Tf_{il} \le Ts_{i(l+1)} \quad \forall j, k \in K_j$$

STAGE SEQUENCING

$$-M\left(1 - W_{ijkl}\right) \le Ts_{il} - Ts_{jk} \quad \forall i, j, k \in K_j, l \in L_i$$

$$M\left(1 - W_{ijkl}\right) \ge Ts_{il} - Ts_{jk} \quad \forall i, j, k \in K_j, l \in L_i$$

SLOT-BATCH MATCHING

SLOT-BASED CONTINUOUS TIME FORMULATION

ADVANTAGES

- Significant reduction in model size when a minimum number of time slots is predefined
- Good computational performance
- Simple model and easy representation for sequencing and allocation scheduling problems

DISADVANTAGES

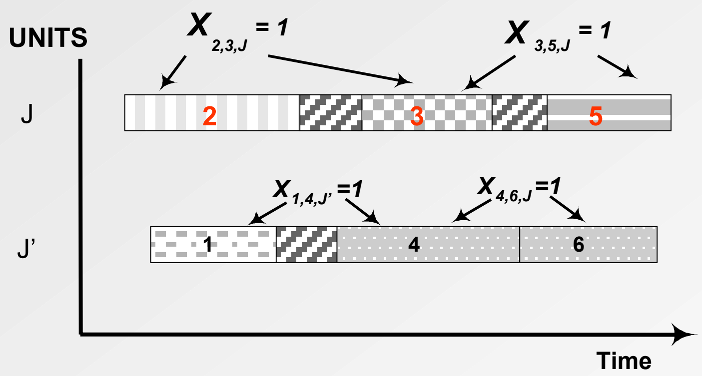
- Resource and inventory constraints are difficult to model
- Model size and complexity depend on the number of time slots predefined
- Sub-optimal or infeasible solution can be generated if the number of time slots is smaller than required

UNIT-SPECIFIC DIRECT PRECEDENCE CONTINUOUS MODEL

(Cerdá et al., 1997).

MAIN ASSUMPTIONS

- Batches to be scheduled are defined a priori
- No mixing and splitting operations are allowed (multistage sequential processes)
- Batches can start and finish at any time during the scheduling horizon



- The position of a batch in the processing sequence is defined in terms of its immediate predecesor & its immediate successor and the assigned unit
- Definition of time slots is not required
- Sequence-dependent setup times are explicitely considered
- A single-stage sequential batch process was studied

SLOT-BASED CONTINUOUS TIME FORMULATION

ADVANTAGES

- Significant reduction in model size when a minimum number of time slots is predefined
- Good computational performance
- Simple model and easy representation for sequencing and allocation scheduling problems

DISADVANTAGES

- Resource and inventory constraints are difficult to model
- Model size and complexity depend on the number of time slots predefined
- Sub-optimal or infeasible solution can be generated if the number of time slots is smaller than required

UNIT-SPECIFIC DIRECT-PRECEDENCE CONTINUOUS MODEL

MAJOR VARIABLES

A. BINARY VARIABLES: $Xf_{i\,j}$, $X_{i'\,i\,j}$

batch unit batch unit

batch

6 batches, 1 stage
2 units per stage
(6 x 2) + (6 x 5 x 2) = 72 variables
slot-based approach → 36

$Xf_{i\,j}$ = denotes that batch i is first processed in unit j

$X_{i'\,i\,j}$ = denotes that batch i is processed immediately after batch i' in unit j

B. CONTINUOUS VARIABLES: Ts_i , Tf_i

6 x 2 = 12 variables
Slot-based approach → 24
(3 slots per unit)

Ts_i, Tf_i = start/end time of batch i

C. MODEL PARAMETERS:

$tp_{i\,j}$ = processing time of batch i in unit j
$cl_{i'\,i\,j}$ = setup time between batches i' & i

UNIT-SPECIFIC DIRECT-PRECEDENCE CONTINUOUS MODEL

- **MAJOR CONSTRAINTS**

BATCH ALLOCATION AND SEQUENCING

- Only one batch can be first processed in a particular unit
- A batch is first processed or it has a single direct predecessor
- A batch has at most a single direct successor in the processing sequence
- A batch shares the same unit with its direct predecessor and its direct successor

BATCH TIMING AND SEQUENCING

- A batch i cannot be started before ending the processing of its direct predecessor i’
- The completion time of a batch can be computed from its starting time by adding both the sequence-dependent setup time and the unit-dependent processing time

UNIT-SPECIFIC DIRECT-PRECEDENCE CONTINUOUS MODEL

(Cerdá et al., 1997)

$$\sum_{i \in I_j} XF_{ij} = 1 \quad \forall j$$

FIRST BATCH IN THE PROCESSING SEQUENCE

$$\sum_{j \in J_i} XF_{ij} + \sum_{j \in J_i} \sum_{i' \in I_j} X_{i'ij} = 1 \quad \forall i$$

FIRST OR WITH ONE PREDECESSOR

$$\sum_{i' \in I_j} X_{ii'j} \le 1 \quad \forall i$$

AT MOST ONE SUCCESSOR

$$XF_{ij} + \sum_{i' \in I_j} X_{i'ij} + \sum_{\substack{j' \in J_i \\ j \ne j'}} \sum_{i' \in I_j} X_{ii'j'} \le 1 \quad \forall i, j \in J_i$$

SUCCESSOR AND PREDECESSOR IN THE SAME UNIT

$$Tf_i = Ts_i + \sum_{j \in J_i} tp_{ij} \left(XF_{ij} + \sum_{i' \in I_j} X_{i'ij} \right) \quad \forall i$$

PROCESSING TIME

$$Ts_i \ge Tf_{i'} + \sum_{j \in J_{ii'}} cl_{i'ij} \; X_{i'ij} - M \left(1 - \sum_{j \in J_{ii'}} X_{i'ij} \right) \quad \forall i, i'$$

SEQUENCING

UNIT-SPECIFIC DIRECT-PRECEDENCE CONTINUOUS MODEL

MAJOR ADVANTAGES

- Sequencing is explicitly considered in model variables
- Sequence-dependent changeover times and costs are easy to implement

MAJOR DISADVANTAGES

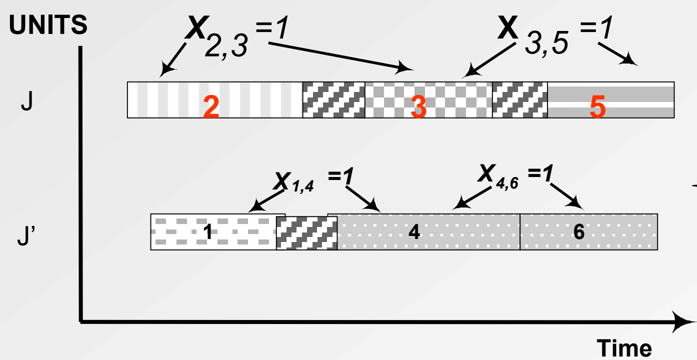
- Larger number of sequencing variables compared with the slot-based approach
- Resource and material balances are difficult to model

GLOBAL DIRECT PRECEDENCE CONTINUOUS FORMULATION

(Méndez et al., 2000; Gupta and Karimi, 2003)

MAIN ASSUMPTIONS

- Batches to be scheduled are defined a priori
- No mixing and splitting operations are allowed
- Batches can start and finish at any time during the scheduling horizon



Allocation variables

$W_{2,J} = 1; W_{3,J} = 1; W_{5,J} = 1$

$W_{1,J'} = 1; W_{4,J'} = 1; W_{6,J'} = 1$

- The formulation is still based on the immediate or direct precedence notion
- The batch location in the processing sequence is given in terms of the immediate predecessor

GLOBAL DIRECT PRECEDENCE CONTINUOUS FORMULATION

- Allocation and sequencing decisions are separately handled through two different sets of binary variables
- The general precedence notion is not associated to a specific unit, i.e. it is a global one
- No time slots are predefined
- Sequence-dependent setup times are explicitely considered

MAJOR PROBLEM VARIABLES (SINGLE-STAGE BATCH PROCESS)

A. BINARY VARIABLES: $W_{i\,j}$ (batch, unit) , $X_{i'\,i}$ (batch, batch) $Xf_{i\,j}$ (batch, unit)

6 batches, 1 stage
2 units per stage
(6 x 5) + (6 x 2 x 2) = 54 variables
slot-based approach ⟶ 36

$X_{i'\,i}$ = denotes that batch i' is processed before batch i in the same unit
$Xf_{i\,j}$ = denotes that batch i is first processed in unit j
$W_{i\,j}$ = denotes that batch i is processed in unit j but not in first place

GLOBAL DIRECT PRECEDENCE CONTINUOUS FORMULATION

B. CONTINUOUS VARIABLES: Ts_i , Tf_i

6 x 2 = 12 variables
Slot-based approach ⟶ 24
(3 slots per unit)

Ts_i, Tf_i = start/end time of batch i

C. MODEL PARAMETERS: $tp_{i\,j}$ = processing time of batch i in unit j
$cl_{i'\,i\,j}$ = setup time between batches i' & i

MAJOR CONSTRAINTS

- ALLOCATION CONSTRAINTS
  - At most one batch *i* can be the first processed in unit *j*
- ALLOCATION-SEQUENCING MATCHING
  - Whenever a pair of batches are related through the immediate precedence relationship, both batches must be allocated to the same unit
- SEQUENCING CONSTRAINTS
  - Every batch should be either the first processed or directly preceded by another batch
  - Every batch has at most only one successor
- TIMING CONSTRAINTS
  - The ending time of batch *i* can be computed from its starting time and the sum of its processing time and the setup time in the allocated unit
  - A batch can be started after its direct predecessor has been completed

GLOBAL DIRECT PRECEDENCE CONTINUOUS FORMULATION

(Méndez et al., 2000)

$$\sum_{i \in I_j} XF_{ij} \leq 1 \quad \forall j$$

AT MOST ONE FIRST BATCH IN THE PROCESSING SEQUENCE

$$\sum_{j \in J_i} XF_{ij} + \sum_{j \in J_i} W_{ij} = 1 \quad \forall i$$

ALLOCATION CONSTRAINT

$$XF_{ij} + W_{ij} \leq W_{i'j} - X_{ii'} + 1 \quad \forall i, i', j \in J_{ii'}$$

$$XF_{ij} + W_{ij} \leq 1 - X_{ii'} \quad \forall i, i', j \in \left(J_i - J_{ii'}\right)$$

SEQUENCING-ALLOCATION MATCHING

$$\sum_{j \in J_i} XF_{ij} + \sum_{i'} X_{i'i} = 1 \quad \forall i$$

FIRST OR WITH ONE PREDECESSOR

$$\sum_{i'} X_{ii'} \leq 1 \quad \forall i$$

AT MOST ONE SUCCESSOR

$$Tf_i = Ts_i + \sum_{j \in J_i} tp_{ij}\left(XF_{ij} + W_{ij}\right) \quad \forall i$$

$$Ts_{i'} \geq Tf_i + \sum_{j \in J_i} \left(cl_{ii'j} + su_{i'j}\right) W_{i'j} - M\left(1 - X_{ii'}\right) \quad \forall i$$

TIMING AND SEQUENCING

GLOBAL DIRECT PRECEDENCE CONTINUOUS FORMULATION

MAJOR ADVANTAGES

- Sequencing is explicitly considered in model variables
- Changeover times and costs are easy to implement
- Lower number of sequencing variables compared with the immediate precedence approach

MAJOR DISADVANTAGES

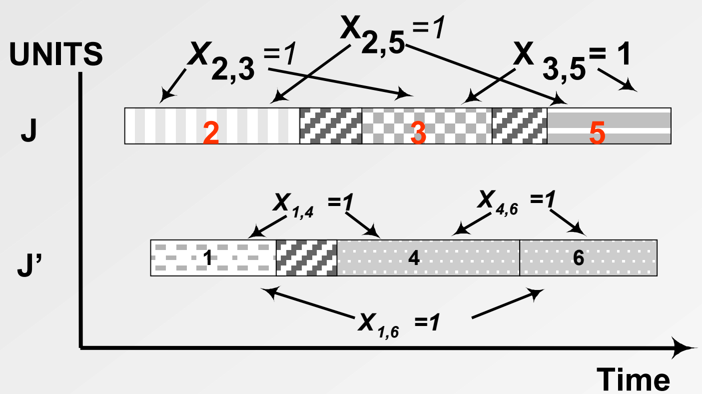
- Resource and material balances are difficult to model
- Large number of sequencing variables compared with the slot-based approach

GLOBAL GENERAL PRECEDENCE CONTINUOUS FORMULATION

(Méndez et al., 2001; Méndez and Cerdá, 2003; Méndez and Cerdá, 2004))

MAIN ASSUMPTIONS

- No mixing and splitting operations are allowed
- Batches can start and finish at any time during the scheduling horizon
- Batches to be scheduled are defined a priori



Allocation variables

$Y_{2,J}=1$; $Y_{3,J}=1$; $Y_{5,J}=1$

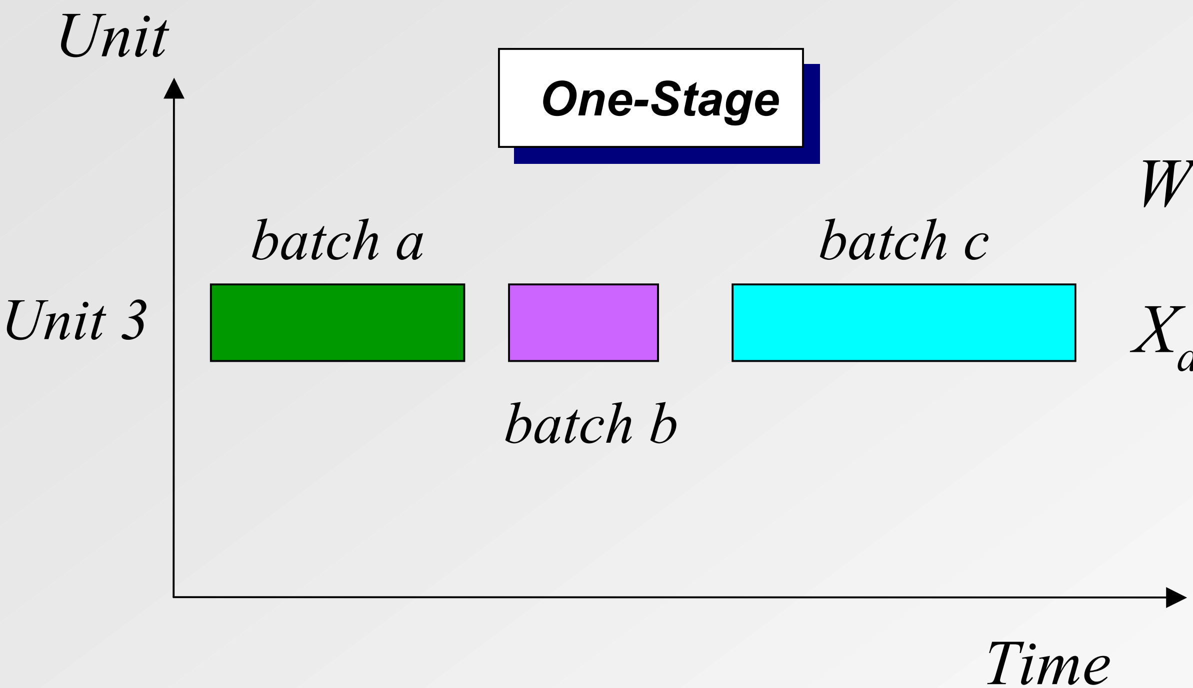
$Y_{1,J'}=1$; $Y_{4,J'}=1$; $Y_{6,J'}=1$

- The generalized precedence notion extends the immediate precedence concept
- The batch location is given in terms of not only the immediate predecessor but also of all the batches processed before in the same unit
- The general precedence notion is not related to a specific unit but a global one
- No time slots are predefined

GLOBAL GENERAL PRECEDENCE CONTINUOUS FORMULATION

INTERESTING FEATURES OF THE APPROACH

- Allocation and sequencing decisions are divided into two different sets of binary variables
- Only one sequencing variable is required to define the relative location of any pair of batch tasks that can be allocated to the same resource
- Different types of shared renewable resources such as processing units, storage tanks, utilities and manpower can be treated in the same way
- Such renewable resources can be efficiently handled through the same set of sequencing variables without compromising the solution optimality
- Sequence-dependent setup times are explicitely considered



$$W_{a3} = W_{b3} = W_{c3} = 1$$

$$X_{ab} = X_{ac} = X_{bc} = 1$$

Batches "a" and "b" are the (direct/non-direct) predecessors of batch "c"

GLOBAL GENERAL PRECEDENCE CONTINUOUS FORMULATION

MAJOR VARIABLES (SINGLE-STAGE BATCH PROCESS)

A. BINARY VARIABLES: $W_{i\,j}$, $X_{i'\,i}$ $i' < i$

$W_{i\,j}$: batch, unit

$X_{i'\,i}$: batch, batch

6 batches, 1 stage
2 units per stage
(6 x 5)/2 + (6 x 2) = 27 variables
slot-based approach → 36

$X_{i'\,i}$ = denotes that batch i' is processed before batch i in the same unit

B. CONTINUOUS VARIABLES: Ts_i , Tf_i

Ts_i, Tf_i = start/end time of batch i

6 x 2 = 12 variables
Slot-based approach → 24
(3 slots per unit)

C. MODEL PARAMETERS: $tp_{i\,j}$ = processing time of batch i in unit j
$cl_{i'\,i\,j}$ = setup time between batches i' & i

GLOBAL GENERAL PRECEDENCE CONTINUOUS FORMULATION

MAJOR PROBLEM VARIABLES (MULTISTAGE BATCH PROCESS)

A. BINARY VARIABLES: $W_{i\,l\,j}$, $X_{i'\,l'\,i\,l}$ $\quad i' < i$

$W_{i\,l\,j}$: batch, stage, unit

$X_{i'\,l'\,i\,l}$: batch, stage, batch, stage

6 batches, 3 stages
2 units per stage
3 x (6 x 5)/2 + (6 x 2 x 3) = 81 variables
slot-based approach → 108

$X_{i'\,l'\,i\,l} = 1$, if task (i',l') is processed before task (i,l) in the same unit

$X_{i'\,l'\,i\,l} = 0$, if task (i',l') is processed after task (i,l) in the same unit

B. CONTINUOUS VARIABLES: $Ts_{i\,l}$, $Tf_{i\,l}$

Ts_{il}, Tf_{il} = start/end time of batch task (i,l)

6 x 3 x 2 = 36 variables
Slot-based approach → 72
(3 slots per unit)

C. MODEL PARAMETERS:

$tp_{i\,l\,j}$ = processing time of batch task (i,l) in unit j

$cl_{i'\,l'\,i\,l\,j}$ = setup time between batch tasks (i',l') & (i,l)

GLOBAL GENERAL PRECEDENCE CONTINUOUS FORMULATION

MAJOR CONSTRAINTS

- **ALLOCATION CONSTRAINTS**

  Every batch task should be allocated to only one processing unit

- **SEQUENCING CONSTRAINTS**

  - Defined for every pair of tasks that can be allocated to the same unit
  - Assume that tasks (i,l) & (i',l') were allocated to the same unit. If task (i,l) is processed before, then task (i',l') starts after completing task (i,l).
  - Otherwise, task (i',l') is ended before starting task (i,l)

- **TECHNOLOGICAL CONSTRAINT**

  - The stage (l+1) of batch i can be started only if stage l has been completed

- **TIMING CONSTRAINTS**

  - The completion time of task (i,l) can be computed from its starting time by adding to it both the task processing time and the sequence-dependent setup time

GLOBAL GENERAL PRECEDENCE CONTINUOUS FORMULATION

(Méndez and Cerdá, 2003)

$$\sum_{j \in J_{il}} W_{ilj} = 1 \quad \forall i, l \in L_i$$

ALLOCATION CONSTRAINT

$$Tf_{il} = Ts_{il} + \sum_{j \in J_{il}} tp_{ilj} W_{ilj} \quad \forall i, l \in L_i$$

PROCESSING TIME

$$Ts_{i'l'} \geq Tf_{il} + cl_{il,i'l'} + su_{i'l'} - M\left(1 - X_{il,i'l'}\right) - M\left(2 - W_{ilj} - W_{i'l'j}\right) \quad \forall i, i', l \in L_i, l' \in L_{i'}, j \in J_{il,i'l'}$$

SEQUENCING CONSTRAINTS

$$Ts_{il} \geq Tf_{i'l'} + cl_{i'l',il} + su_{il} - M\, X_{il,i'l'} - M\left(2 - W_{ilj} - W_{i'l'j}\right) \quad \forall i, i', l \in L_i, l' \in L_{i'}, j \in J_{il,i'l'}$$

$$Ts_{il} \geq Tf_{i(l-1)} \quad \forall i, l \in L_i, l > 1$$

STAGE PRECEDENCE

GLOBAL GENERAL PRECEDENCE CONTINUOUS FORMULATION

ADVANTAGES

- General sequencing is explicitly considered in model variables
- Changeover times and costs are easy to implement
- Lower number of sequencing decisions
- Sequencing decisions can be extrapolated to other resources

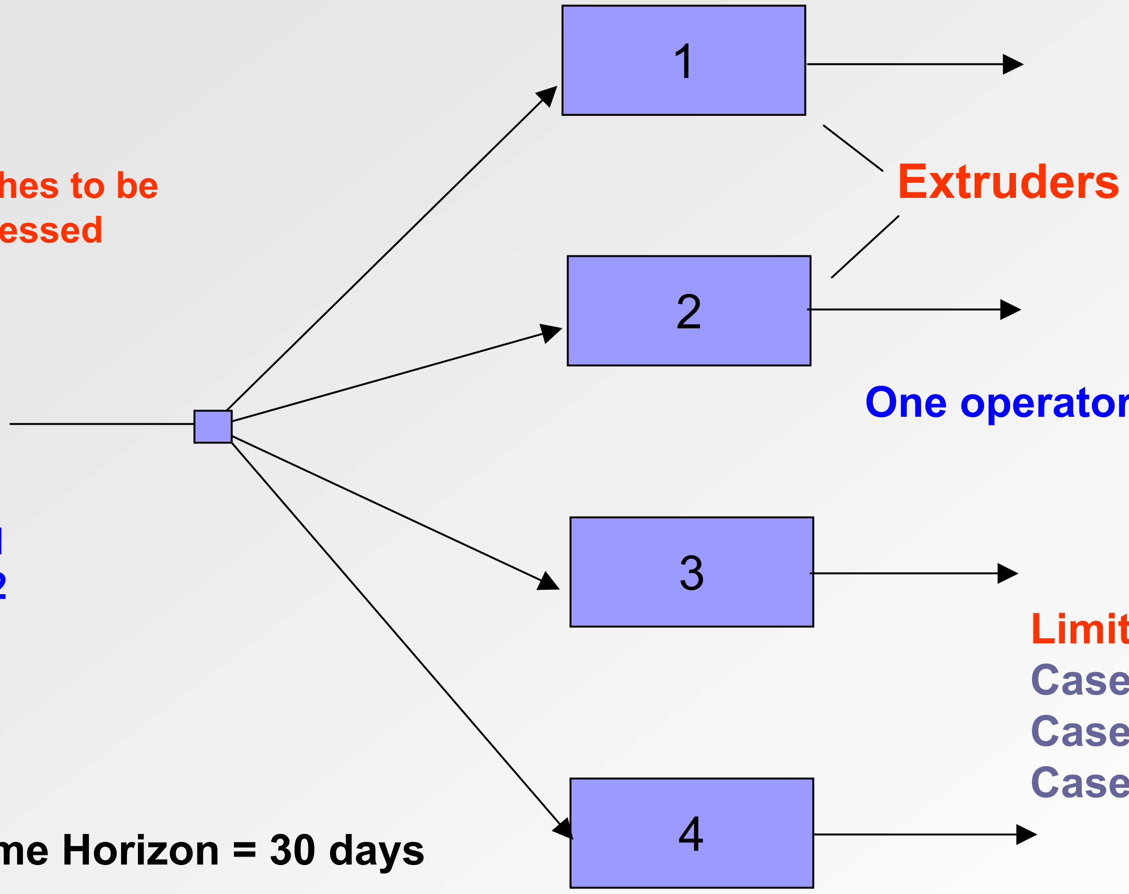
DISADVANTAGES

- Material balances are difficult to model

CASE STUDY FOR COMPARING BATCH-ORIENTED APPROACHES

Batches to be Processed

B1
B2
B3
....
....
B11
B12



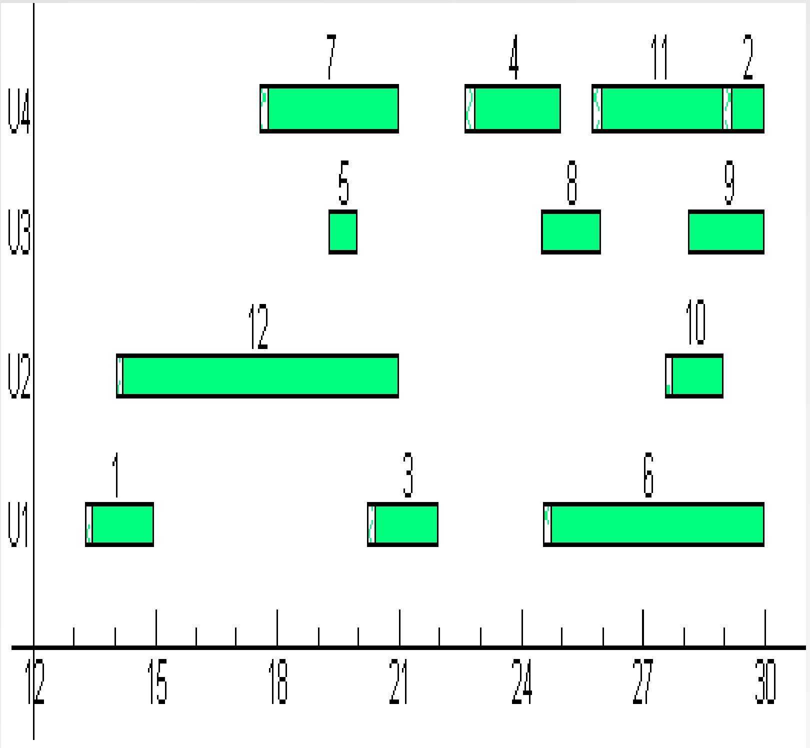
One operator crew per extruder

Limited Manpower:
Case 1: 4 operator crews
Case 2: 3 operator crews
Case 3: 2 operator crews

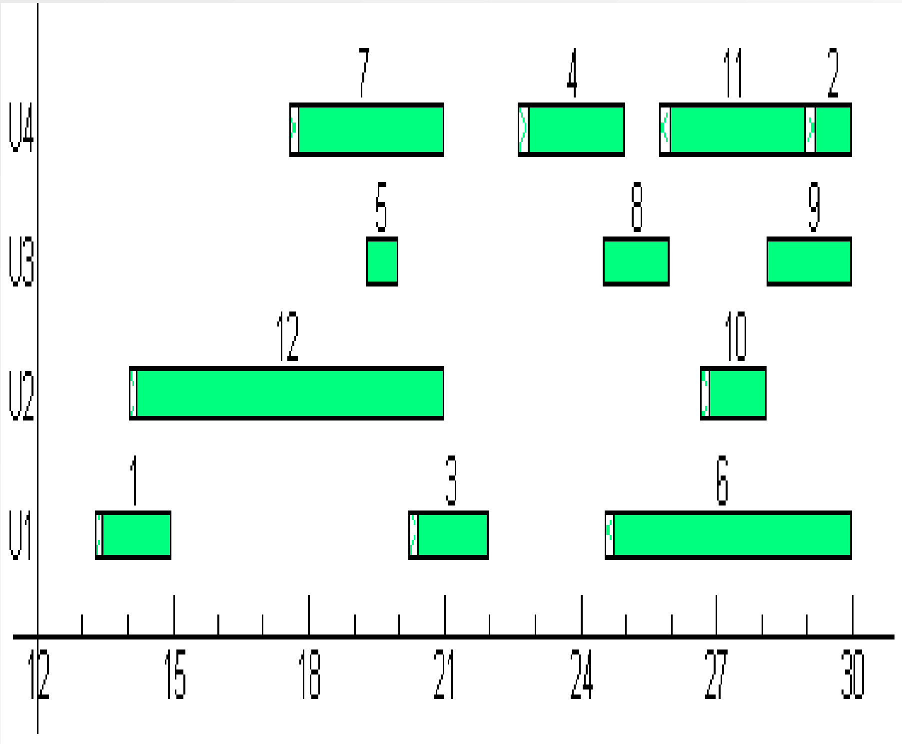
Time Horizon = 30 days

CASE STUDY FOR COMPARING BATCH-ORIENTED APPROACHES-2

OPTIMAL SCHEDULES



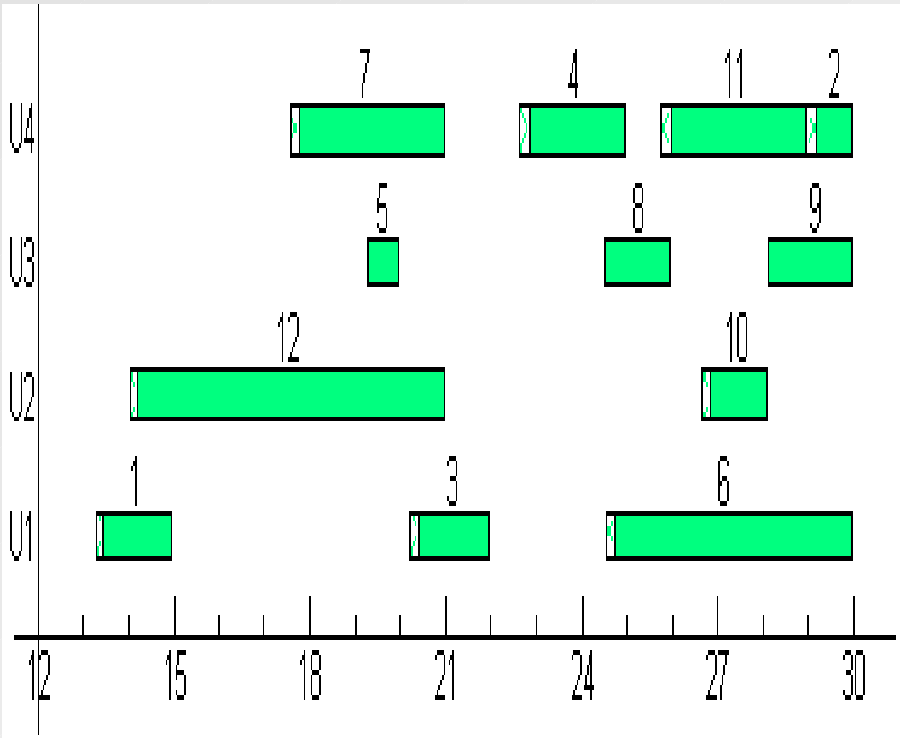
(a) without manpower limitation



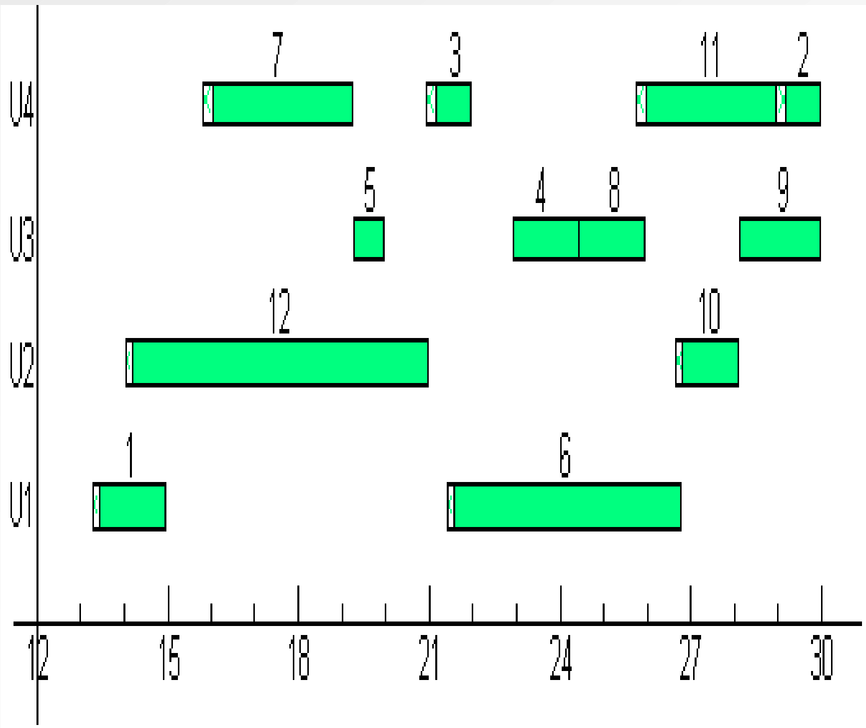
(b) 3 operator crews

CASE STUDY FOR COMPARING BATCH-ORIENTED APPROACHES-3

OPTIMAL SCHEDULES



(b) 3 operator crews



(c) 2 operator crews

CASE STUDY FOR COMPARING BATCH-ORIENTED APPROACHES-4

Case Study	Event representation	Binary vars, cont. vars, constraints	Objective function	CPU time
2.a	Time slots & preordering	100, 220, 478	1.581	67.74 (*113.35*)*
	General precedence	82, 12, 202	1.026	0.11[b]
2.b	**Time slots & preordering**	**289, 329, 1156**	**2.424**	**2224** (*210.7*)*
	General precedence	**127, 12, 610**	**1.895**	**7.91**[b]
2.c	Time slots & preordering	289, 329, 1156	8.323	76390 (*927.16*)*
	General precedence	115, 12, 478	7.334	35.87[b]

BATCH REACTIVE SCHEDULING

Roslöf, Harjunkoski, Björkqvist, Karlsson and Westerlund (2001); Méndez and Cerdá (2003, 2004)

- An industrial environment is dynamic in nature and the proposed schedule must usually be updated in midweek because of unexpected events.
- Different types of unexpected events may happen like:
  - changes in batch processing/setup times
  - unit breakdown/startup
  - late order arrivals and/or orders cancellations
  - reprocessing of batches
  - delayed raw material shipments
  - modifications in order due dates and/or customer priorities
- To prevent rescheduling actions from disrupting a smooth plant operation, limited changes in batch sequencing and unit assignment are just permitted.
- The goal is to meet all the production requirements still to achieve under the new conditions by making limited low-cost changes in batch sequencing and assignment.

BATCH REACTIVE SCHEDULING

Roslöf, Harjunkoski, Björkqvist, Karlsson and Westerlund (2001); Méndez and Cerdá (2003, 2004)

- A minimum deterioration of the problem objective (minimum makespan, minimum tardiness) is pursued.
- **PROBLEM DATA**
  - the schedule in progress
  - the present plant state
  - current inventory levels
  - present resource availabilities
  - current time data
  - unexpected events
  - allowed rescheduling actions
  - The criterion to be optimized
- **RESCHEDULING ACTIONS**

  Proper adjustments to the current schedule may include:
  - simultaneous local reordering of old batches at some equipment units
  - reassignment of certain old batches to alternative equipment items due to unexpected unit failures
  - insertion of new batches
  - batch time shifting

BATCH REACTIVE SCHEDULING CONTINUOUS APPROACH

Méndez and Cerdá (2003, 2004)

- It is based on the global general precedence notion
- It is a batch-oriented iterative approach
- It allows simultaneous insertion and reallocation of new/old batches as well as the resequencing of old batches at each iteration
- It considers sequence-dependent changeovers and limited renewable resources
- A limited number of rescheduling actions can be applied in order to reduce the problem size as much as the scheduler wants

- At each iteration of the rescheduling algorithm, two steps are sequentially executed:
  - the assignment step during which new batches are inserted and a limited number of old batches can be simultaneously reallocated
  - the sequencing step where neighboring batches in the same queue can exchange locations, and the procedure is repeated until no improvement in the objective function is observed

REACTIVE SCHEDULING

Allocation variables: Y_{5A} ; Y_{5B}

Sequencing variables: X_{15} ; X_{25} ; X_{35} ; X_{45}

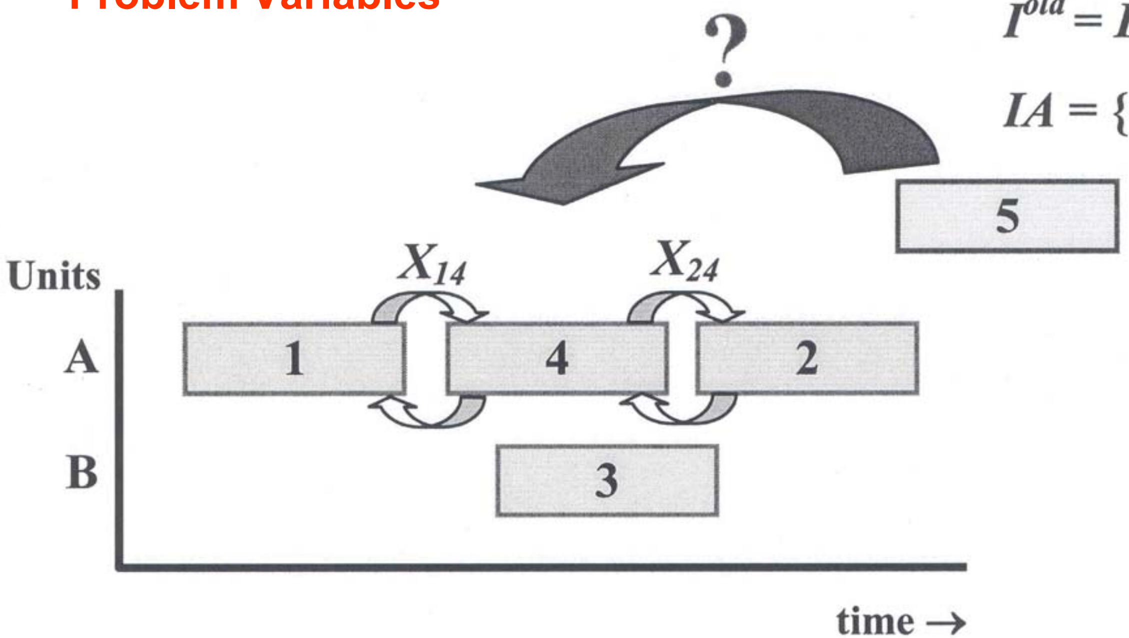
Problem Variables

$n_i = 1$

$I^{new} = \{ 5 \}$

$I^{old} = IS = \{ 1, 2, 3, 4 \}$

$IA = \{\Phi\}$



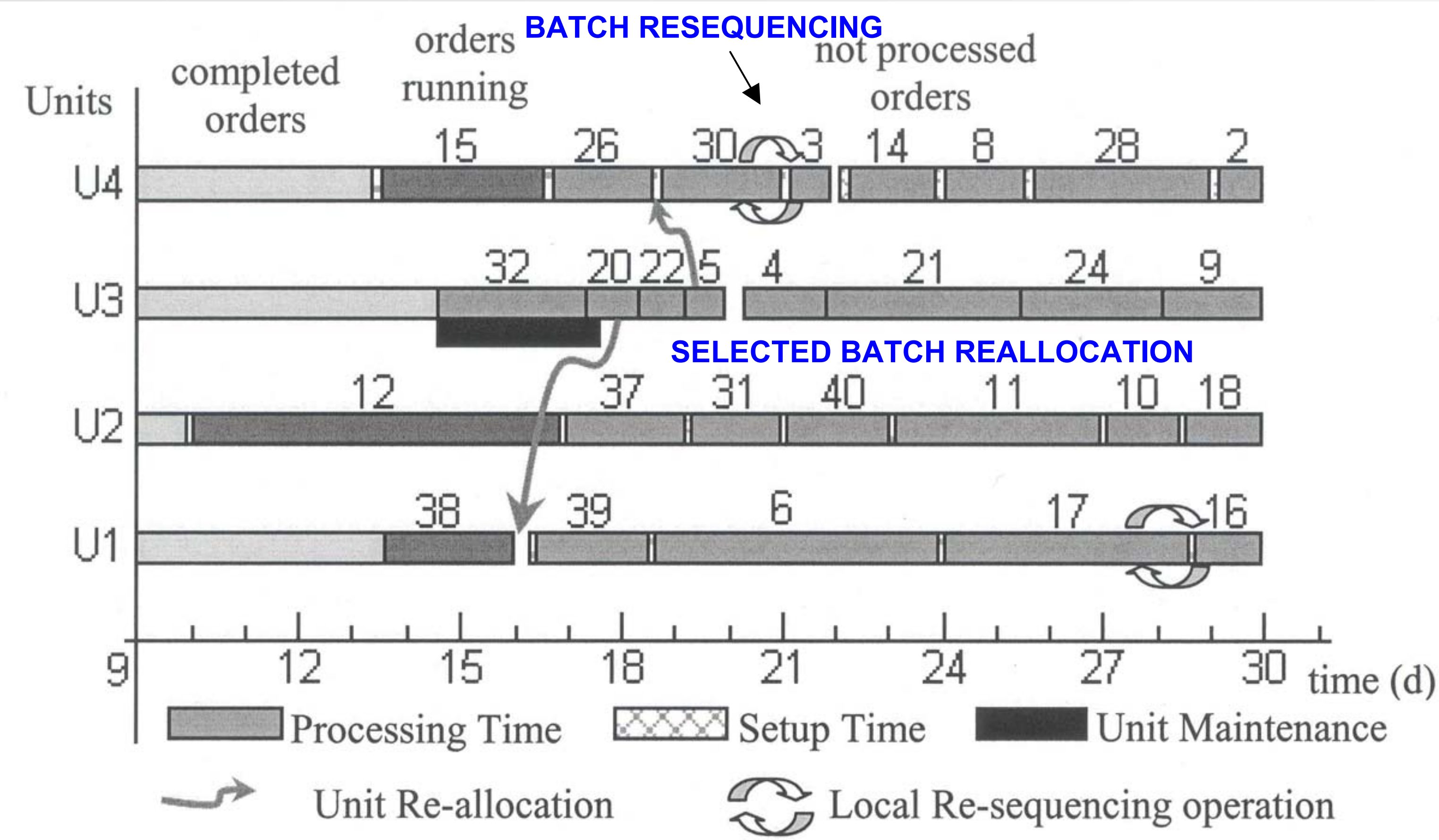
New batch to be inserted

REACTIVE SCHEDULING EXAMPLE

- **A single-stage multiproduct batch plant with four units working in parallel**
- **Forty batches are to be processed within a 30-day scheduling horizon.**
- **An unexpected 3-day maintenance period for unit U_3 at time t = 14.6 d makes necessary to perform a rescheduling process**
- **At the rescheduling time, 25 batches are still to be processed**

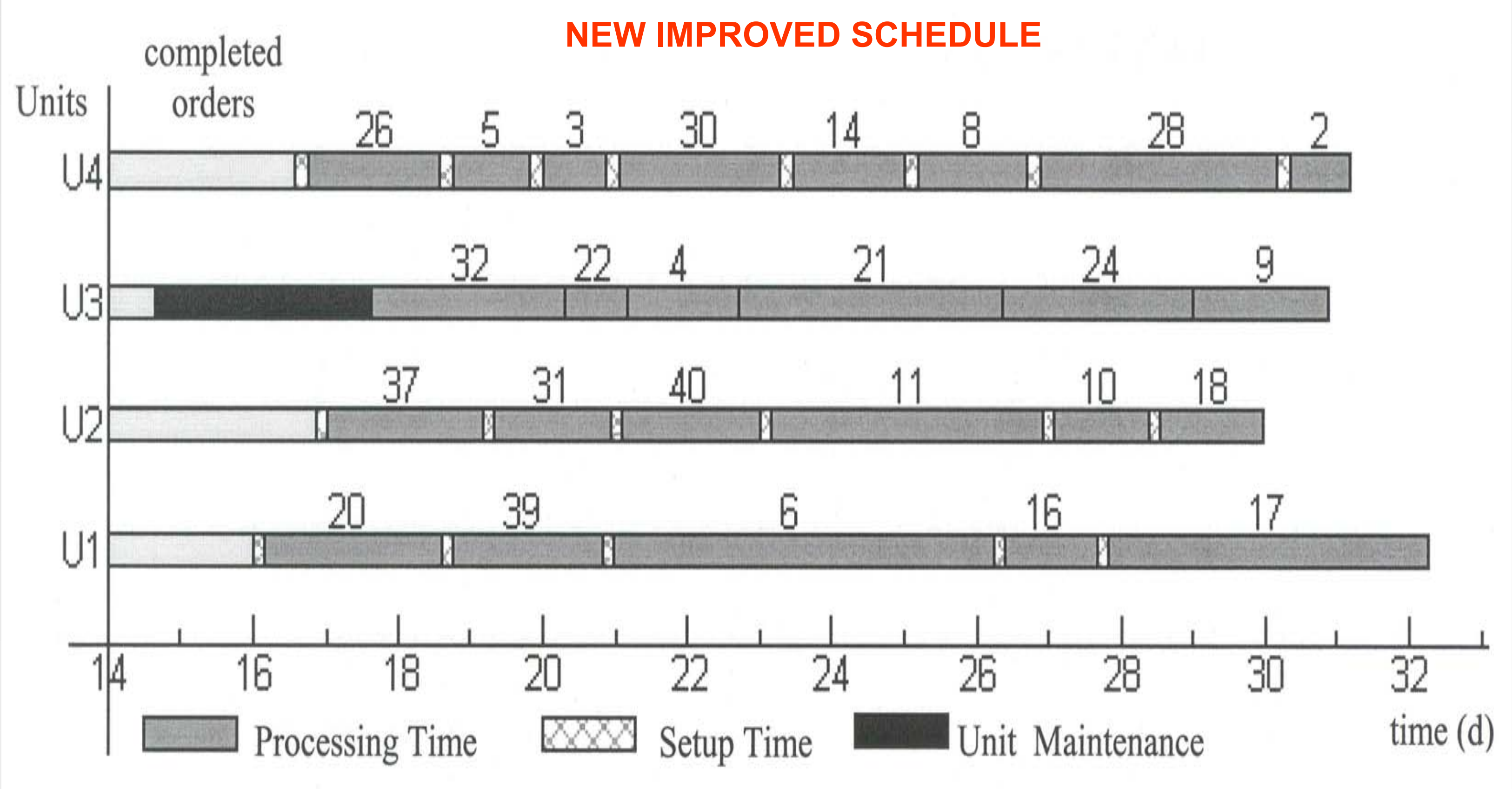
REACTIVE SCHEDULING EXAMPLE

UNEXPECTED 3-DAY MAINTAINANCE OF UNIT U3



SCHEDULE IN PROGRESS

REACTIVE SCHEDULING EXAMPLE



	Total Tardiness
Only batch time shifting in unit U3:	13.55 d
New improved schedule:	8.84 d

CONCLUSIONS

- Current optimization models are able to solve moderate-size batch processes
- Small examples can be solved to optimality
- Discrete-time models are computationally more effective than continuous-time models of the network type
- Difficult selection of the number of time or event points in network-oriented continuous time formulations
- Network-oriented continuous-time models become quickly computationally intractable for scheduling of medium complexity process networks.
- Problems with more than 150 time intervals are difficult to solve using discrete time models
- Problems with more than 15 time or event points appear intractable using network-oriented continuous time models.
- Depending on the objective function, different computational performances are observed
- Batch-oriented continuous approaches are computationally more efficient but usually require to first solve the batching problem (a decomposition approach)
- Combining other approaches with mathematical programming (hybrid methods) for solving large scale problems looks very promising