Using MILP and CP for the Scheduling of Batch Chemical Processes

Ignacio E. Grossmann
Christos Maravelias

Department of Chemical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213

Motivation

- Increased interest in optimal scheduling of batch scheduling by chemical/pharmaceutical industry

- *Hybrid Methods* that combine Constraint Programming (CP) with Mixed-Integer Linear Programming (MILP) to overcome combinatorial explosion in scheduling

Exploit respective strengths through two subproblems:
Assignment (MILP) and Sequencing (CP)

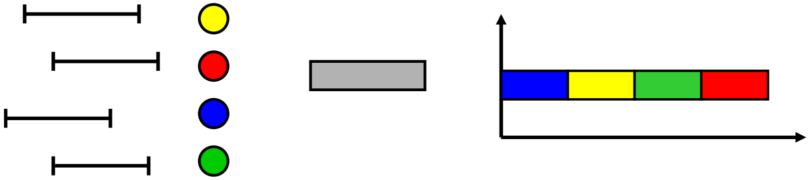
•Two classes of scheduling problems:

- Single stage parallel units
- Continuous-time State Task Network

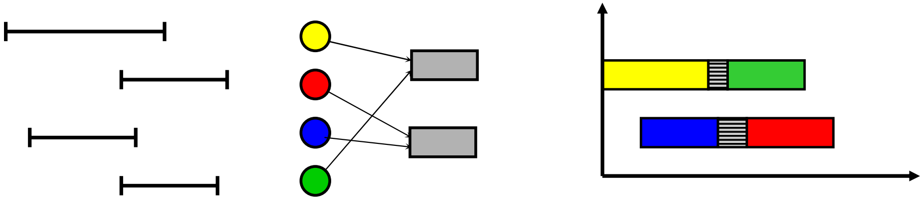
Goal: Generalization and unified treatment

Variety of Batch Processing Problems

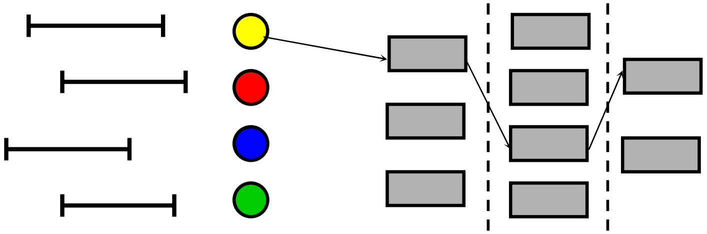
One unit



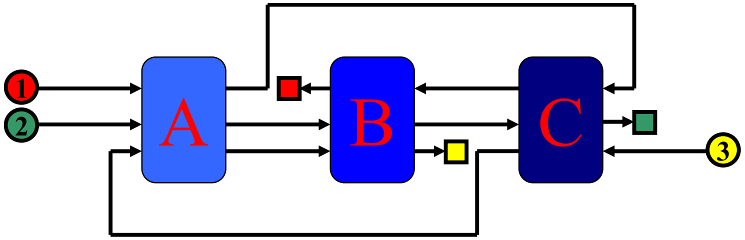
Single-stage-Parallel units



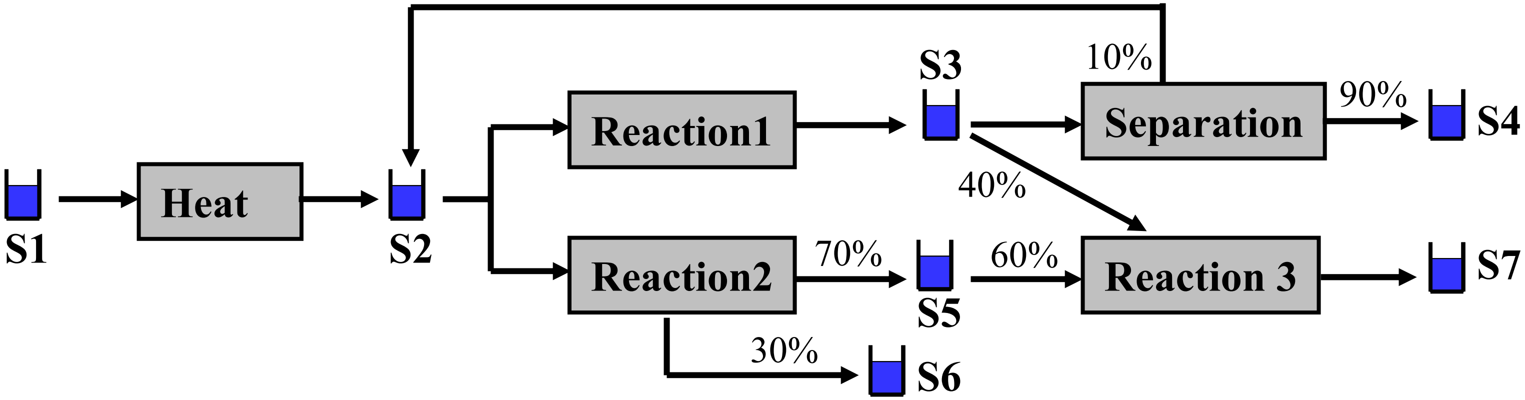
Flow-shop (Multi-stage)



Job-shop



Multipurpose Plant



S1
Heat
S2
Reaction1
Reaction2
S3
10%
Separation
90%
S4
40%
70%
S5
60%
Reaction 3
S7
30%
S6

Constraint vs. Mixed-integer Programming

Mixed-integer Programming

- *Intelligent* search strategy for general purpose models

  Computationally effective for optimization problems with many feasible solutions

  Not effective for feasibility problems and *sequencing* problems

Constraint Programming

- *Fast algorithms* for special problems

  Computationally effective for *highly constrained, feasibility* and *sequencing* problems

  Not effective for optimization problems with *complex structure* and *many feasible solutions*

Basic Idea

Decompose problem into two parts:

1. Use MILP for high-level optimization decisions *(assignment)*
2 Use CP for low-level decisions *(sequencing)*

MILP/CP Hybrid Models

Jain, Grossmann (2001)

MILP:

$$
\begin{aligned}
(M1):\min\ & c^T x \\
s.t.\ \ & Ax + By + Cv \leqslant a \\
& A'x + B'y + C'v \leqslant a' \\
& x \in \{0,1\}^n,\ \ y \in \{0,1\}^m,\ v \in R^p
\end{aligned}
$$

Complicating rows

Complicating variables
(not in objective function)

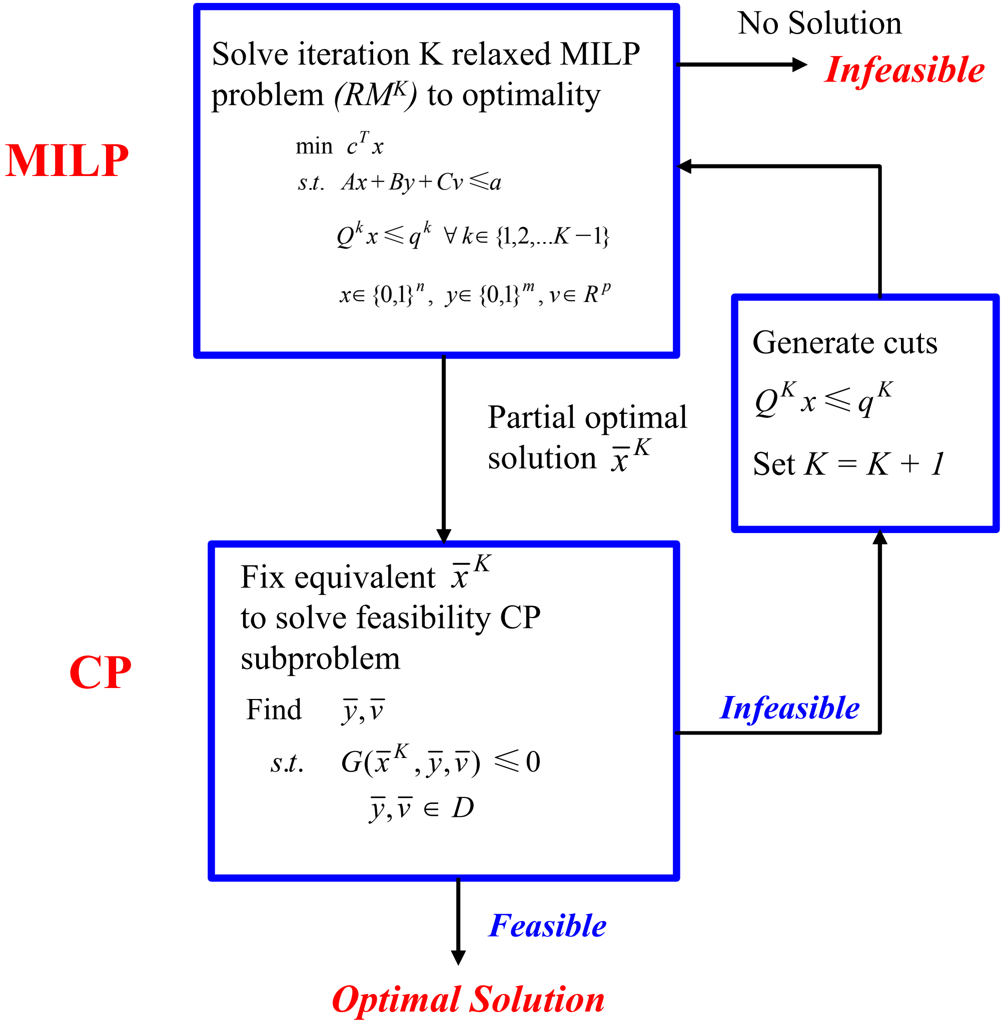
CP:

$$
\begin{aligned}
(M2):\min\ \ & f(\bar{x}) \\
s.t.\ \ & G(\bar{x},\bar{y},\bar{v}) \leqslant 0 \\
& \bar{x},\bar{y},\bar{v} \in D
\end{aligned}
$$

Hybrid:

$$
\begin{aligned}
&\left.\begin{aligned}
(M3):\min\ & c^T x \\
s.t.\ \ & Ax + By + Cv \leqslant a
\end{aligned}\right\}\ \textit{MILP (optimality)} \\
&\left.\begin{aligned}
& x \Leftrightarrow \bar{x} \\
& \bar{G}(\bar{x},\bar{y},\bar{v}) \leqslant 0
\end{aligned}\right\}\ \textit{CP (feasibility)} \\
& x \in \{0,1\}^n,\ \ y \in \{0,1\}^m,\ v \in R^p \\
& \bar{x},\bar{y},\bar{v} \in D
\end{aligned}
$$

Decomposition Hybrid Model



MILP

Solve iteration K relaxed MILP problem *(RMK)* to optimality

$$\begin{aligned} &\min \ c^T x \\ &s.t. \ \ Ax + By + Cv \leqslant a \\ &\quad Q^k x \leqslant q^k \ \ \forall k \in \{1,2,...K-1\} \\ &\quad x \in \{0,1\}^n, \ y \in \{0,1\}^m, \ v \in R^p \end{aligned}$$

No Solution

Infeasible

Partial optimal solution $\bar{x}^K$

CP

Fix equivalent $\bar{x}^K$ to solve feasibility CP subproblem

$$\begin{aligned} &\text{Find} \ \ \bar{y}, \bar{v} \\ &s.t. \ \ G(\bar{x}^K, \bar{y}, \bar{v}) \leqslant 0 \\ &\qquad \bar{y}, \bar{v} \in D \end{aligned}$$

Infeasible

Generate cuts

$Q^K x \leqslant q^K$

Set $K = K + 1$

Feasible

Optimal Solution

$$\sum_{i \in T^k} x_i - \sum_{i \in F^k} x_i \leqslant B^k - 1$$

$$T^k = \{i \mid x_i^k = 1\}, \ F^k = \{i \mid x_i^k = 0\}$$

No-good Cuts (weak)

Balas & Jeroslow (1972)

Theorem: *If the MILP/CP decomposition method is applied to solve problem (M3) with the no-good cuts, the method converges to the optimal solution or proves infeasibility in a finite number of iterations.*

Remark.

Method can be implemented as a branch-and-cut method

- Branch-and-bound on MILP subproblem
- At integer nodes derive cuts from CP subproblems

Two major issues:

How to partition/decompose problem?

How to derive effective cuts?

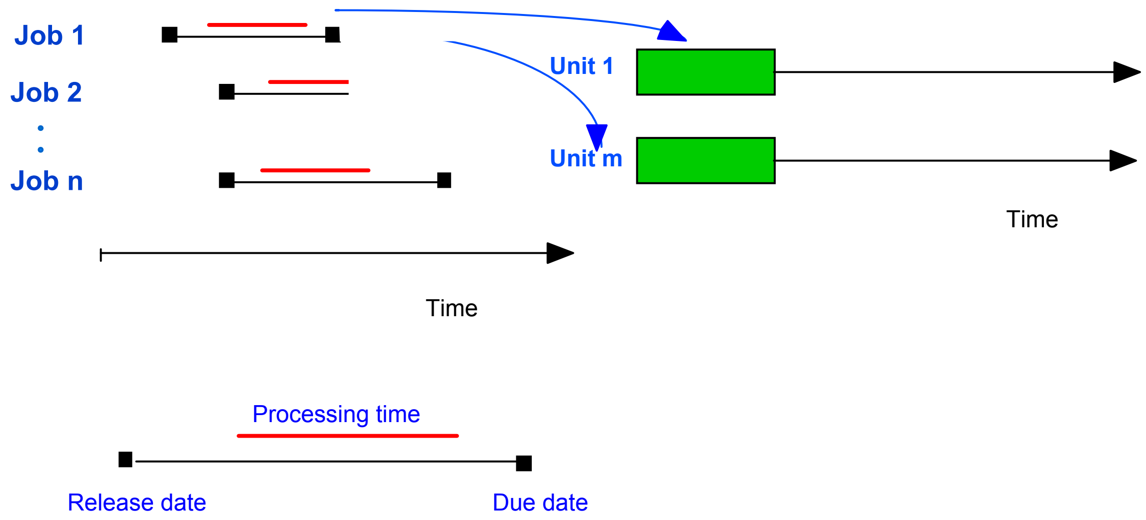
Scheduling of parallel units (Single Stage)

(Jain and Grossmann, 2001)

Given: n jobs/orders (release dates, processing times, due dates)

m units (cost different for each unit/machine)

Find schedule that minimizes cost and meets all due dates



MILP Optimization Model

$$x_{im} = \begin{cases} 1 \; if \; task \; i \; to \; unit \; m \\ 0 \; otherwise \end{cases} \qquad ts_i = start \; time \; task \; i$$

$$\min \sum_{i \in I} \sum_{m \in M} C_{im} x_{im}$$

Cost processing

$$s.t. \quad ts_i \geq r_i$$

Earliest start

$$ts_i \leq d_i - \sum_{m \in M} p_{im} x_{im} \quad \forall i \in I$$

Latest start

$$\sum_{m \in M} x_{im} = 1 \quad \forall i \in I$$

Assign to only one unit

$$\sum_{i \in I} x_{im} p_{im} \leq \max_i \{d_i\} - \min_i \{r_i\} \quad \forall m \in M$$

Assignment constraints

Sequencing tasks in each unit

$Let\ y_{ii'} = 1\ if\ task\ i\ before\ task\ i'\ on\ given\ unit$

$If\ x_{im}\ AND\ x_{i'm}\ true\ then\ y_{ii'}\ OR\ y_{i'i}\ are\ true$

$$y_{ii'} + y_{i'i} \geq x_{im} + x_{i'm} - 1 \quad \forall i, i' \in I, i > i', m \in M$$

$If\ \ y_{ii'} = 1\ then\ ts_{i'} \geq ts_i + p_{im}$

$$ts_{i'} \geq ts_i + \sum_{m \in M} p_{im} x_{im} - M(1 - y_{ii'}) \quad \forall i, i' \in I, i \neq i'$$

Big-M Constraint

$$x_{im} = \{0,1\},\ y_{ii'} = \{0,1\}, \qquad ts_i \geq 0$$

Hybrid Optimization Model

Assignment orders to units: Mixed-integer linear programming

$$x_{im} = \begin{cases} 1 \; if \;\; job \; i \; to \; unit \; m \\ 0 \; otherwise \end{cases}$$

$$\min \sum_{i \in I} \sum_{m \in M} C_{im} x_{im}$$

$$s.t. \;\; ts_i \geq r_i$$

$$ts_i \leq d_i - \sum_{m \in M} p_{im} x_{im} \quad \forall i \in I$$

$$\sum_{m \in M} x_{im} = 1 \quad \forall i \in I$$

$$\sum_{i \in I} x_{im} p_{im} \leq \max_i \{d_i\} - \min_i \{r_i\} \;\; \forall m \in M$$

Sequencing of orders in each unit

Constraint Programming

$$if\ (x_{im} = 1)\ then\ \ (z_i = m)\ \forall i \in I, m \in M$$

$$i.start \geq r_i \ \ \forall i \in I$$

$$i.start \leq d_i - p_{z_i} \ \ \forall i \in I$$

$$i.duration = p_{z_i}$$

$$i\ requires\ t_{z_i} \ \ \forall i \in I$$

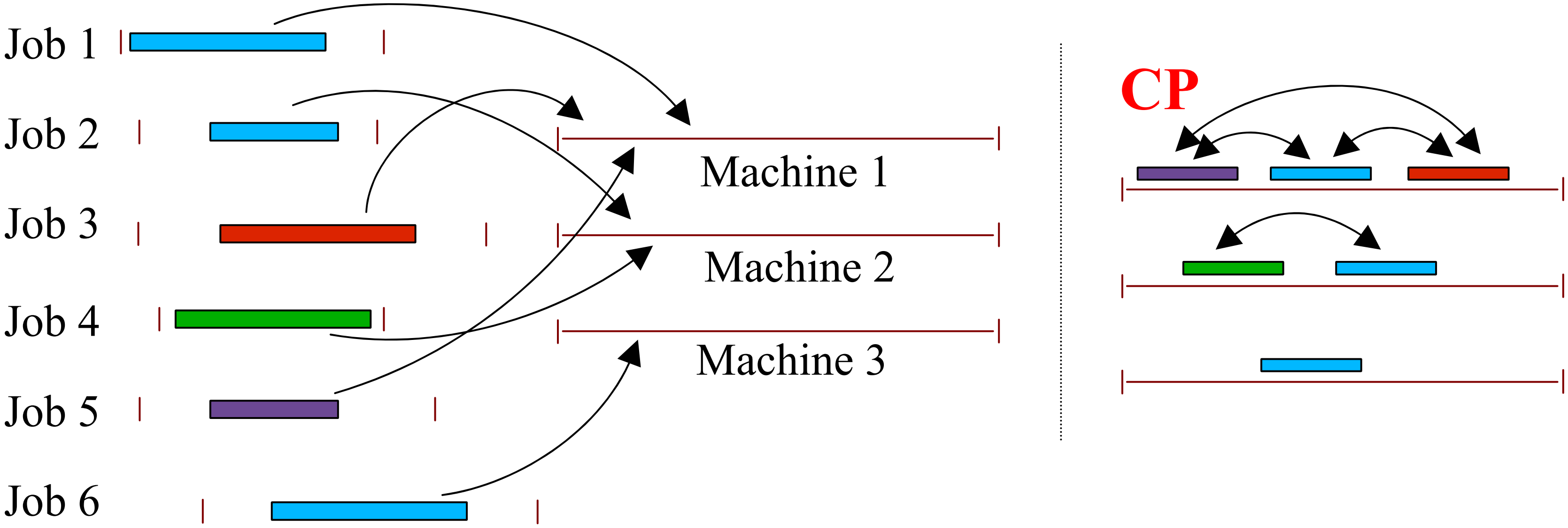
Global constraint *(implicit)*

$$x_{im} \in \{0,1\}, ts_i \geq 0 \ \ \forall i \in I, m \in M$$

$$z_i \in M\ \forall i \in I; i.start \in D, i.duration \in D \ \ \forall i \in I$$

Decomposition Strategy

MILP



Job 1
Job 2
Job 3
Job 4
Job 5
Job 6

Machine 1
Machine 2
Machine 3

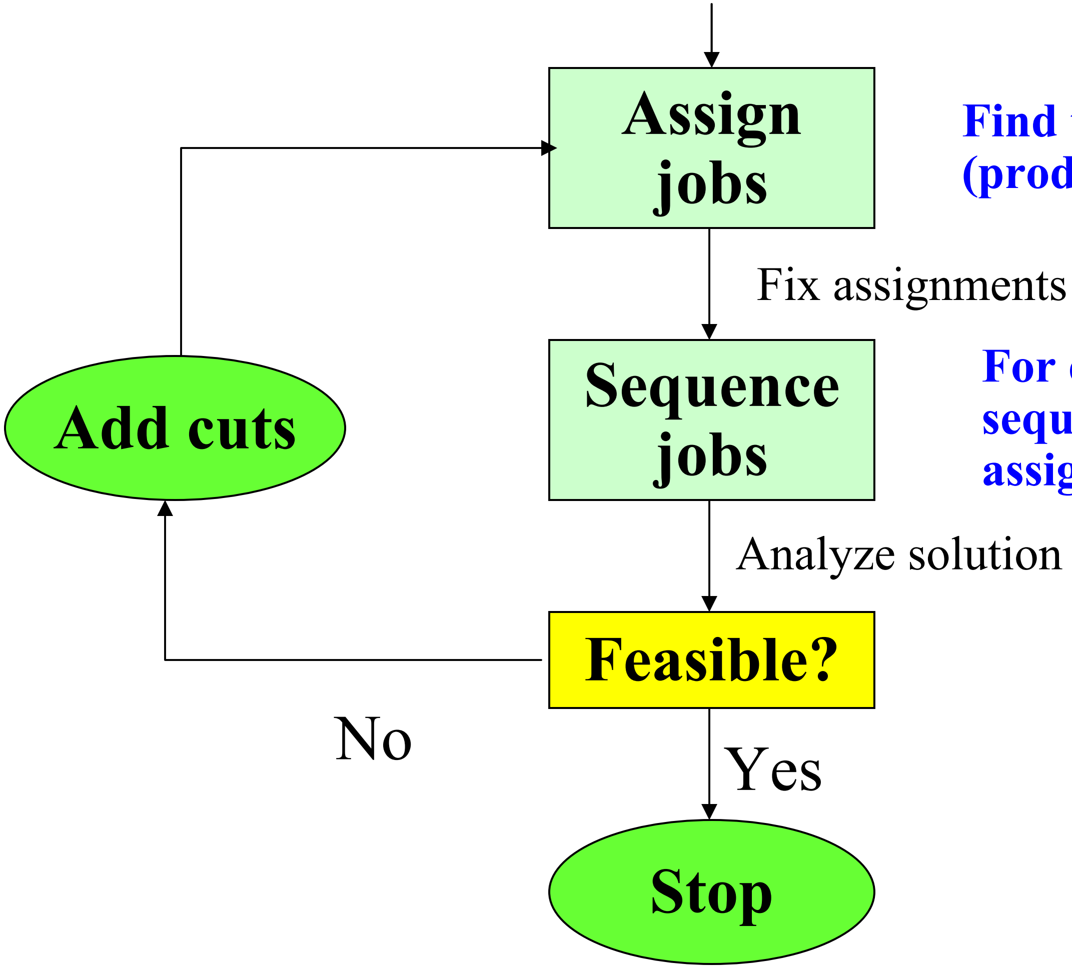
CP

Assignment

Sequencing

Decomposition Strategy

Separate problem into assignment and sequencing problems



Assign jobs

Fix assignments

Sequence jobs

Analyze solution

Feasible?

No

Add cuts

Yes

Stop

Find the minimum assignment (production) cost MILP

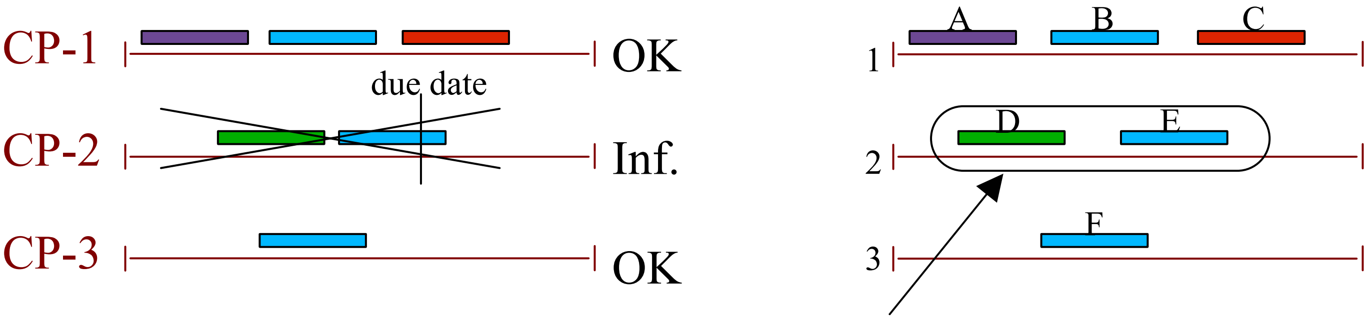
For each unit find a feasible sequence for the chosen assignment CP

Cuts

Cuts for infeasible assignments:

$$\sum_{i \in I_m^k} x_{im} \leq B_k^m - 1 \quad \forall m \in M$$

$$I_m^k = \left\{ i \,\middle|\, x_{im}^k = 1 \right\}, \; B_k^m = \left| I_m^k \right|$$



Cut out this assignment

$$(x_{D2} + x_{E2} \leq 1)$$

- No sequence can be found in machine 2 in the assignment of is infeasible and can be cut out
- The cuts also exclude large number of supersets

Computational Results

CPU times with integer data *(Jain and Grossmann, 2001)*

CPLEX 6.5/Sun Ultra 60

	2 machines		3 machines		3 machines		5 machines		5 machines	
	3 jobs		7 jobs		12 jobs		15 jobs		20 jobs	
Problem	**Set 1**	**Set 2**	**Set 1**	**Set 2**	**Set 1**	**Set 2**	**Set 1**	**Set 2**	**Set 1**	**Set 2**
MILP	0.04	0.04	0.31	0.27	926.3	199.9	1784.7	73.3	18142.7	102672.3
CLP	0	0.02	0.04	0.14	3.84	0.38	553.5	9.28	68853.5	2673.9
Hybrid	0.02	0.01	0.52	0.02	4.18	0.02	2.25	0.04	14.13	0.41

CPU times with arbitrary rational numbers *(Harjunkoski et al., 2002)*

CPLEX 6.5/Sun Ultra 60

	2M, 3J		3M, 7J		3M, 12J		5M, 15J		5M, 20J	
Problem	**Set 1**	**Set 2**	**Set 1**	**Set 2**	**Set 1**	**Set 2**	**Set 1**	**Set 2**	**Set 1**	**Set 2**
Hybrid	0.00	0.01	0.51	0.02	5.36	0.03	0.64	0.92	36.63	4.79

Recent related work

Bockmayr, Pisaruk (2003)

Cuts: Monotone constraints

Branch and cut

Cardinality cuts for scheduling problem

Sadykov, Wolsey (2003)

Investigated several algorithms

Most efficient:

MIP/CP branch and cut algorithm valid inequalities

Column generation method with strengthened MIP/CP subproblems

Hooker (2004)

Single Stage Parallel Units with Resource Constraints

Cost, Makespan, Tardiness minimization

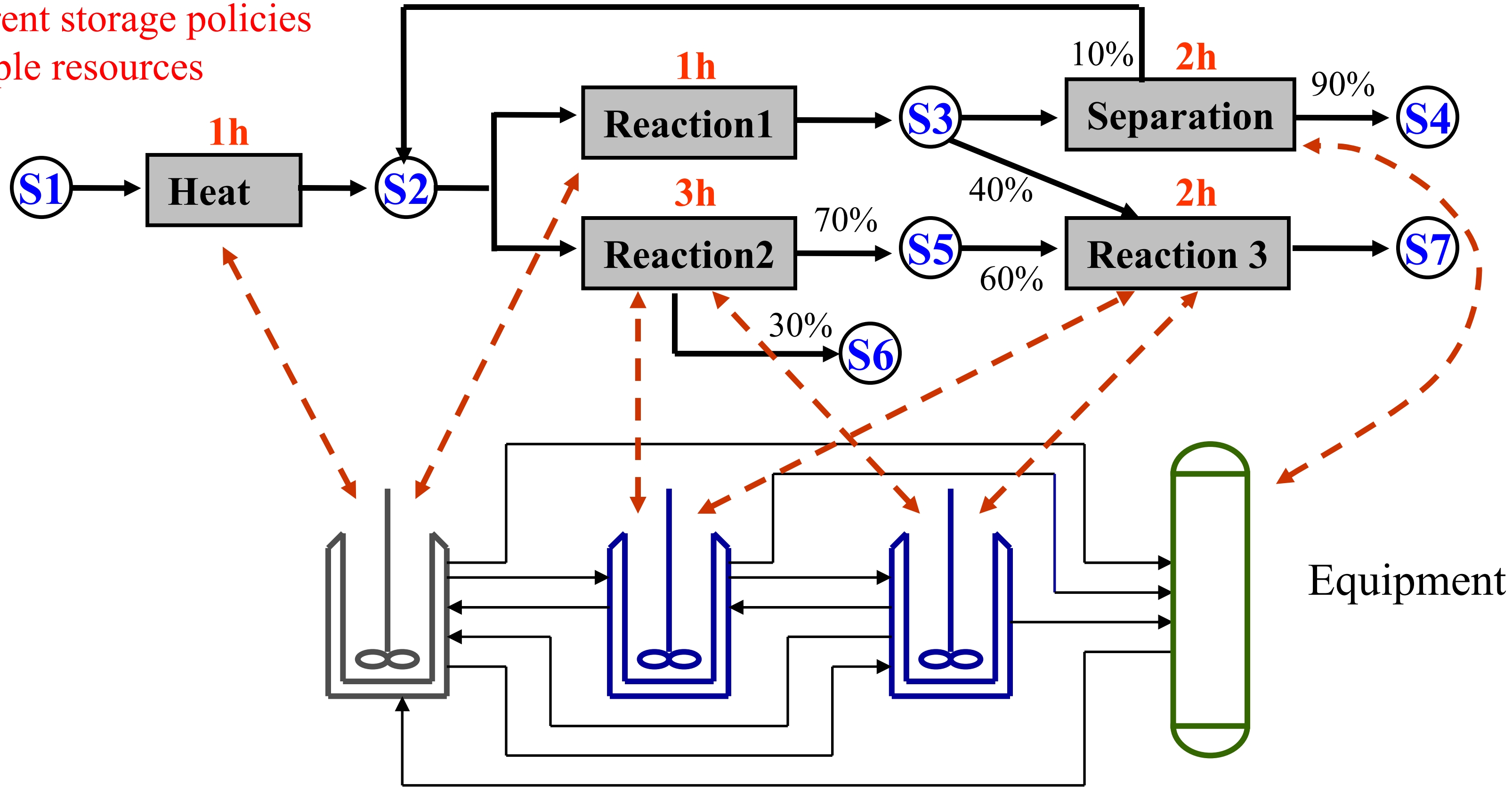
Decompose: MILP Assignment/CP Resource Constrained

Benders cuts

State Task Network (STN)

- ➢ Complex batch plant configurations
- ➢ Variable batch sizes, processing times
- ➢ Different storage policies
- ➢ Multiple resources

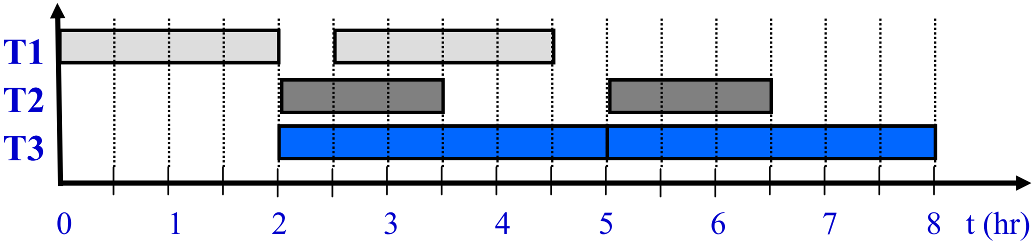
States ○ Tasks ▭



- Kondili, Pantelides & Sargent (1993); Shah, Pantelides & Sargent (1993): STN – Discrete
- Pantelides (1994): RTN – Discrete.

Discrete-time STN Model

Partition into fixed time intervals



Variables:

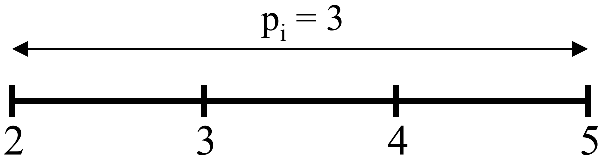
W_{ijt} = 1 if unit *j* starts processing task *i* at the beginning of time period *t*; 0 otherwise.

B_{ijt} = Amount of material which starts undergoing task *i* in unit *j* at the beginning of period *t*.

S_{st} = Amount of material stored in state s, at the beginning of period t.

U_{ut} = Demand of utility *u* over time interval *t*.

Task *i* starts at *t*=2 in unit *j*



$W_{ij2} = 1,\ B_{ij2} \neq 0$

$W_{ijt} = 0,\ B_{ijt} = 0,\ t \neq 2$

Allocation Constraints:

$$\sum_{i \in I_j} \sum_{\hat{t}=t}^{t-p_i+1} W_{ij\hat{t}} \le 1 \quad \forall j,t$$

Capacity limitations:

$$W_{ijt} V_{ij}^{\min} \le B_{ijt} \le W_{ijt} V_{ij}^{\max} \quad \forall i,t,j \in K_i \qquad 0 \le ST_{st} \le C_s \quad \forall s,t$$

Material balances:

$$ST_{st} = ST_{st-1} + \sum_{i \in \bar{T}_s} \bar{\rho}_{is} \sum_{j \in K_i} B_{ij,t-p_{is}} - \sum_{i \in T_s} \rho_{is} \sum_{j \in K_i} B_{ijt} + R_{st} - D_{st} \quad \forall s,t$$

Availability of utilities:

$$U_{ut} = \sum_{t} \sum_{j \in K_i} \sum_{\theta=0}^{p_i-1} (\alpha_{ui\theta} W_{ijt-\theta} + \beta_{ui\theta} B_{ijt-\theta}) \quad \forall u,t \qquad 0 \le U_{ut} \le U_{ut}^{\max} \quad \forall u,t$$

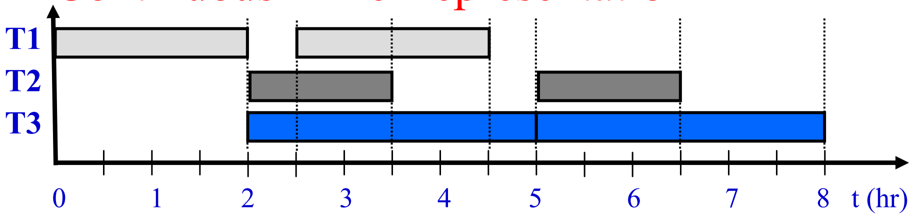
Objective Function:
Profit

$$Z = \sum_{s} \sum_{t=1}^{H} C_{st}^{D} D_{st} - \sum_{s} \sum_{t=1}^{H} C_{st}^{R} R_{st} + \sum_{s} C_{s,H+1} S_{s,H+1} - \sum_{u} \sum_{t=1}^{H} C_{ut} U_{ut}$$

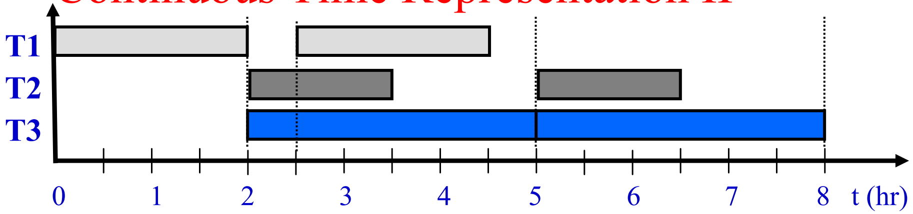
Continuous Time Models

Motivation: Fewer Intervals arbitrary fixed /variable processing times

Continuous Time Representation I



Continuous Time Representation II



- Fewer intervals
- Variable proc. Times
- Unknown time points

Zhang & Sargent (1995); Schilling & Pantelides (1996): RTN – Continuous
Mockus & Reklaitis (1999): STN – Continuous
Ierapetritou & Floudas (1998): Continuous Event-Based Formulation

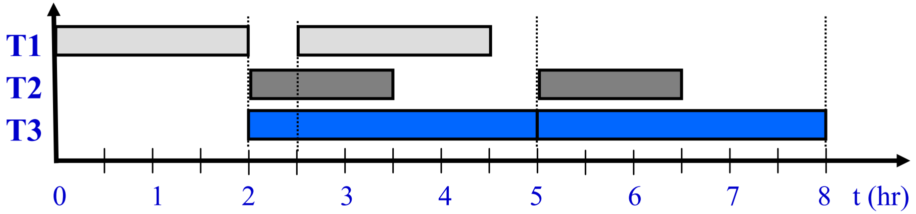
Goal: ***(Maravelias, Grossmann, 2003*)**

New Continuous Time Model for Scheduling of Multipurpose Batch Plants

- Accommodate multiple objective functions ***(max profit, min makespan)***
- **Combined MILP/Constraint Programming**

M-G Continuous-time STN Model

1. Common mixed continuous-time representation ⇒ Fewer time points

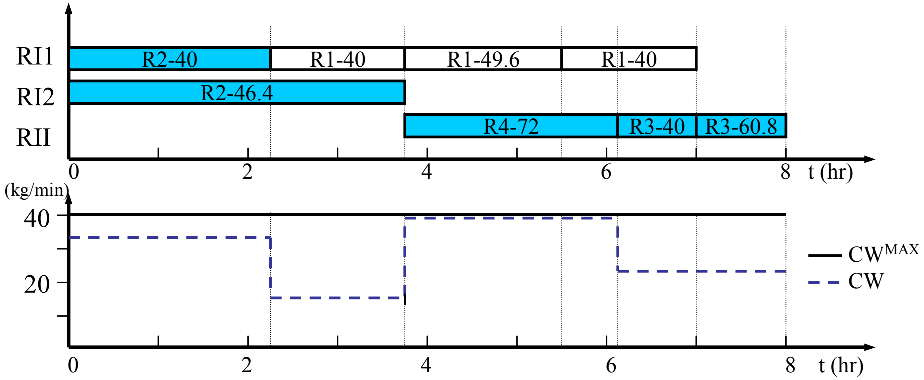


- If produce ZW-state finish **at** time point
- Else finish **at or before** time point

2. Unit-task decoupling and New Assignment Constraints ⇒ Fewer binaries

(Ierapetritou, Floudas, 1998)

3. Utility Constraints



4. Addition of New Valid Inequalities ⇒ Tighter LP Relaxation

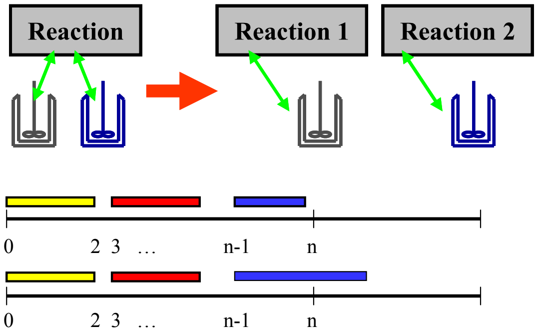
M-G Continuous-time STN Model

Assignment Constraints

Ws_{in} = *1* if task *i* starts at time point *n*
Wp_{in} = *1* if task *i* is being processed at time point *n*
Wf_{in} = *1* if task *i* finishes at or before time point *n*

Zs_{jn} = *1* if a task in *I(j)* starts in unit *j* at time point *n*
Zp_{jn} = *1* if a task in *I(j)* is being processed in unit *j* at time point *n*
Zf_{jn} = *1* if a task in *I(j)* finishes in unit *j*, at or before time point *n*

Task-Unit Decoupling: Define new tasks



$$Zs_{jn} \Leftrightarrow \underset{i \in I(j)}{\vee} Ws_{in} \quad \forall j, \forall n$$

$$Zf_{jn} \Leftrightarrow \underset{i \in I(j)}{\vee} Wf_{in} \quad \forall j, \forall n$$

$$Zp_{jn} = \sum_{n'<n} Zs_{jn'} - \sum_{n' \le n} Zf_{jn'} \quad \forall j, \forall n$$

Logic Condition:

If a task is assigned to start in unit j at time point n, then equipment j at time point n is not processing any other task

$$Zs_{jn} \Rightarrow \neg Zp_{jn}$$

Integer assignment constraints

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

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

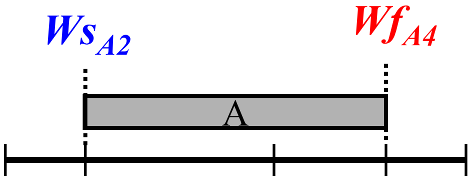
$$\sum_{i \in I(j)} Ws_{in} \le 1 \quad \forall j, \forall n \qquad \sum_{i \in I(j)} Wf_{in} \le 1 \quad \forall j, \forall n$$

$$Wf_{i0} = 0 \ \forall i \qquad Ws_{|N|} = 0 \ \ \forall i$$

M-G Continuous-time STN Model

Timing Constraints

T_n	Time that corresponds to time point *n* (i.e. start of period *n*; finish of period *n-1*)
Ts_{in}	Start time of task *i* that starts at time point *n*
Tf_{in}	Finish time of task *i* that starts at time point *n*
D_{in}	Duration of task *i* that starts at time point *n*



Tn=Ts	0	2	6	8	10
n	1	2	3	4	5
D	0	6	0	0	0
Tf	0	8	8	8	8

$$Ts_{in} = T_n \quad \forall i, \forall n$$

$$\begin{pmatrix} Ws_{in} \\ D_{in} = \alpha_i + \beta_i Bs_{in} \\ Tf_{in} = Ts_{in} + D_{in} \end{pmatrix} \vee \begin{pmatrix} \neg Ws_{in} \\ D_{in} = 0 \\ Tf_{in} = Tf_{in-1} \end{pmatrix}, \quad \forall i, \forall n$$

$$\begin{pmatrix} Wf_{in} \\ Tf_{in} \le T_n, \quad i \notin ZW \\ Tf_{in} = T_n, \quad i \in ZW \end{pmatrix} \vee \begin{pmatrix} \neg Wf_{in} \\ Tf_{in} \ge Tf_{in-1} \end{pmatrix}, \quad \forall i, \forall n$$

MIP Constraints (big-M)

$$D_{in} = \alpha_i Ws_{in} + \beta_i Bs_{in} \quad \forall i, \forall n$$

$$Tf_{in} \le Ts_{in} + D_{in} + H(1 - Ws_{in}) \quad \forall i, \forall n$$

$$Tf_{in} \ge Ts_{in} + D_{in} - H(1 - Ws_{in}) \quad \forall i, \forall n$$

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

$$Tf_{in-1} \ge T_n - H(1 - Wf_{in}) \quad \forall i \in ZW(i), \forall n$$

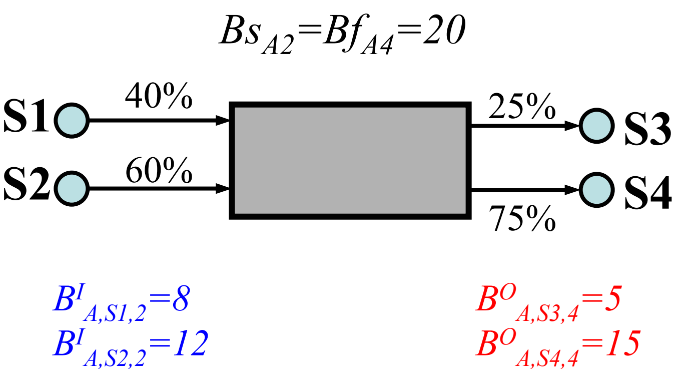
$$Ts_{in} = T_n \quad \forall i, \forall n$$

$$Tf_{in} - Tf_{in-1} \le H \cdot Ws_{in} \ \forall i, \forall n$$

$$Tf_{in} - Tf_{in-1} \ge D_{in} \ \forall i, \forall n$$

M-G Continuous-time STN Model

Batch Size Constraints and Mass Balances



$$B_i^{MIN} Ws_{in} \le Bs_{in} \le B_i^{MAX} Ws_{in} \quad \forall i, \forall n$$

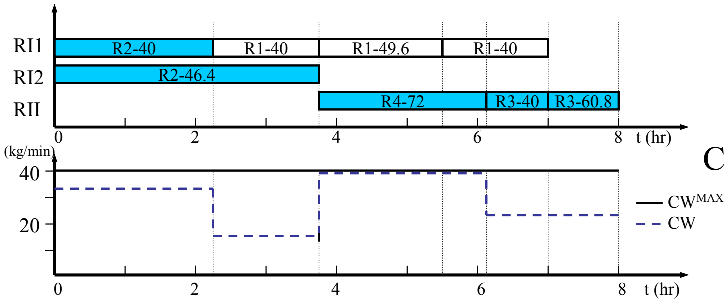
$$B_i^{MIN} Wf_{in} \le Bf_{in} \le B_i^{MAX} Wf_{in} \quad \forall i, \forall n$$

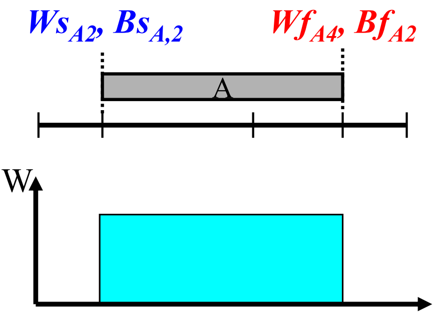
$$B^I_{isn} = \rho_{is} Bs_{in} \quad \forall i, \forall n, \forall s \in SI(i) \qquad B^O_{isn} = \rho_{is} Bf_{in} \quad \forall i, \forall n, \forall s \in SO(i)$$

$$S_{sn} + SS_{sn} = S_{s,n-1} + \sum_{i \in O(s)} B^O_{isn} - \sum_{i \in I(s)} B^I_{isn} \quad \forall s, \forall n > 1$$

$$S_{sn} \le C_s \quad \forall s, \forall n$$

Utility Constraints





$$R^I_{irn} = \gamma_{ir} Ws_{in} + \delta_{irs} Bs_{in} \quad \forall i, \forall r, \forall n$$

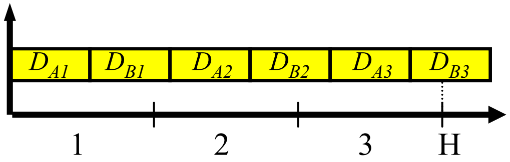
$$R^O_{irn} = \gamma_{ir} Wf_{in} + \delta_{irs} Bf_{in} \quad \forall i, \forall r, \forall n$$

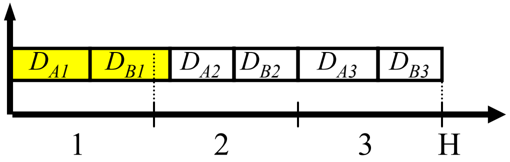
$$R_{rn} = R_{rn-1} - \sum_i R^O_{irn-1} + \sum_i R^I_{irn} \quad \forall r, \forall n$$

$$R_{rn} \le R_r^{MAX} \quad \forall r, \forall n$$

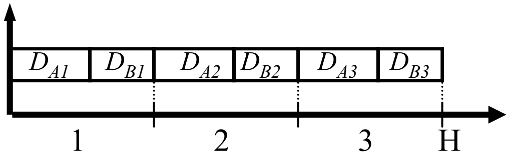
M-G Continuous-time STN Model

Addition of new valid inequalities ⇒ Tighter LP relaxation





$$\sum_{i\in I(j)}\sum_{n} D_{in} \le H \quad \forall j$$



$$\sum_{i\in I(j)}\sum_{n'\ge n} D_{in'} \le H - T_n \quad \forall j, \forall n$$

$$\sum_{i\in I(j)}\sum_{n'\le n} (\alpha_i Wf_{in'} + \beta_i Bf_{in'}) \le T_n \quad \forall j, \forall n$$

⇒ **Smaller Bs'**
Lower Production

Lower Profit

MILP Formulation (MGP)

$$\max Z = \sum SS_{sn}\zeta_s$$

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

$$\sum_n Ws_{in} = \sum_n Wf_{in} \quad \forall i$$

Novel Assignment Constraints

$$D_{in} = \alpha_i Ws_{in} + \beta_i Bs_{in} \quad \forall i, \forall n$$

$$Tf_{in} \le Ts_{in} + D_{in} + H(1 - Ws_{in}) \quad \forall i, \forall n$$

$$Tf_{in} \ge Ts_{in} + D_{in} - H(1 - Ws_{in}) \quad \forall i, \forall n$$

$$Ts_{in} = T_n \quad \forall i, \forall n$$

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

$$Tf_{in-1} \ge T_n - H(1 - Wf_{in}) \quad \forall i \in ZW(i), \forall n$$

Finish time of task *i*

Big-M

$$B_i^{MIN} Ws_{in} \le Bs_{in} \le B_i^{MAX} Ws_{in} \quad \forall i, \forall n$$

$$B_i^{MIN} Wf_{in} \le Bf_{in} \le B_i^{MAX} Wf_{in} \quad \forall i, \forall n$$

$$Bs_{in-1} + Bp_{in-1} = Bp_{in} + Bf_{in} \quad \forall i, \forall n$$

$$B^I_{isn} = \rho_{is} Bs_{in} \quad \forall i, \forall n, \forall s \in SI(i)$$

$$B^O_{isn} = \rho_{is} Bf_{in} \quad \forall i, \forall n, \forall s \in SO(i)$$

$$S_{sn} = S_{s,n-1} + \sum_{i\in O(s)} B^O_{isn} - \sum_{i\in I(s)} B^I_{isn} + SP_{sn} - SS_{sn} \quad \forall s, \forall n > 1$$

Mass balances

$$R^I_{irn} = \gamma_{ir} Ws_{in} + \delta_{irs} Bs_{in} \quad \forall i, \forall r, \forall n$$

$$R^O_{irn} = \gamma_{ir} Wf_{in} + \delta_{irs} Bf_{in} \quad \forall i, \forall r, \forall n$$

$$R_{rn} = R_{rn-1} - \sum_i R^O_{irn-1} + \sum_i R^I_{irn} \quad \forall r, \forall n$$

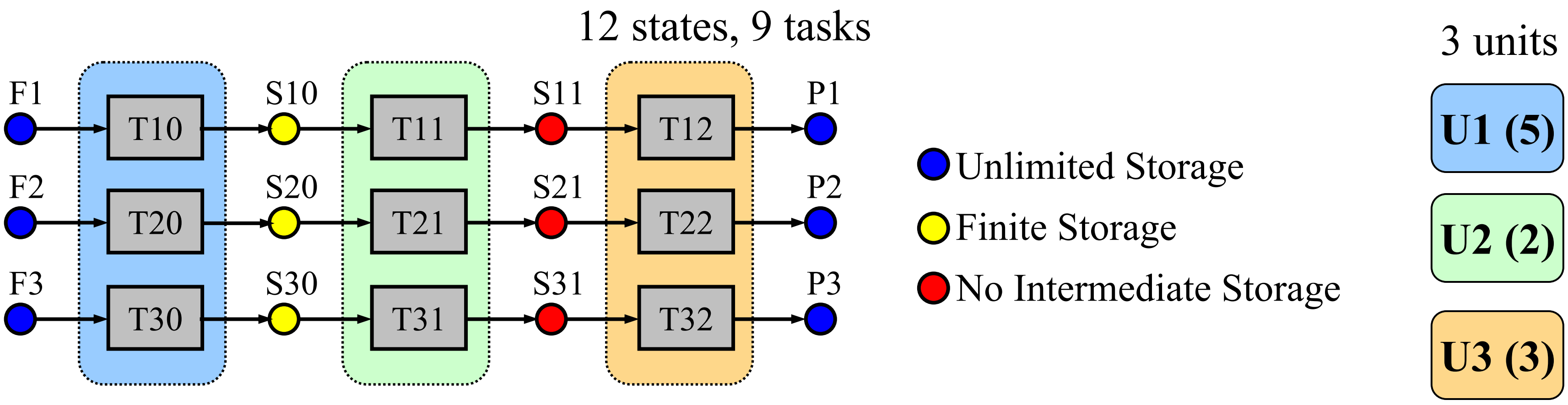
Utility Constraints

$$\sum_{i\in I(j)}\sum_{n'\ge n} D_{in'} \le H - T_n \quad \forall j, \forall n$$

$$\sum_{i\in I(j)}\sum_{n'\le n}(Wf_{in'} fd_i + Bf_{in'} vd_i) \le T_n \quad \forall j, \forall n$$

Novel Tightening Constraints

Example (12,9,3)



Maximize Production

	MILP STN Model						
Horizon (hrs)	n	0-1	Cont.	Constr.	LP Relax	Obj	CPU s
15	10	180	1163	2473	14	12	58.5
20	12	216	1395	2965	18	16*	> 7200
25	15	270	1743	3703	24	22	> 7200

GAMS/CPLEX7.5/Pentium III 933MHz

Continuous-time State Task Network MILP

Computationally Limited

Observations

1. MILP involves big-M constraints from logic conditions/disjunctions

$$\boxed{Wf_{in} \Rightarrow Tf_{in} = T_n \quad \forall i, \forall n} \quad => \quad \begin{array}{l} Tf_{in} \leqslant T_n + H(1 - Wf_{in}) \quad \forall i, \forall n \\ Tf_{in} \geqslant T_n - H(1 - Wf_{in}) \quad \forall i, \forall n \end{array}$$

2. To set up MILP model need to postulate number of intervals and iterate on them for optimal solution

3. Unlike the single stage parallel scheduling problem there is
   no obvious partitioning of constraints in MILP
   no obvious cuts for decomposition

Decomposition: **Aggregated MILP master problem for assignment**
CP with material flows for sequencing

Aggregated MILP for Assignment

Type and number of tasks are unknown

Binary Z_{ic} = *1* if copy *c* of task *i* takes place ($c=1,2..N_i$)

$\Rightarrow$ Define activity for CP only if $Z_{ic}=1$

Proposition: The set of mixed-integer constraints in (MP) corresponds to a linear combination of constraints of problem (MGP)

$$\begin{aligned}
&\sum_{i\in I(j)}\sum_{c} D_{ic} Z_{ic} \le H \quad \forall j\\
&B_i^{MIN} Z_{ic} \le B_{ic} \le B_i^{MAX} Z_{ic} \quad \forall i, \forall c\\
&S_s = S_0 + \sum_i \sum_c \rho_{is}^{O} B_{ic} - \sum_i \sum_c \rho_{is}^{I} B_{ic} \quad \forall s\\
&S_s \ge d_s \quad \forall s \in FP\\
&S_s \le C_s \quad \forall s \in INT\\
&Z_{ic+1} \le Z_{ic} \quad \forall i, \forall c < |C|
\end{aligned} \qquad \text{(MP)}$$

Corollary: Optimizing a given objective function over (MP) provides a bound to the optimization over the constraints of (MGP).

Max Profit: Upper bound

Min Makespan: Lower bound

CP subproblem

For given tasks (Z_{ic}=1) from aggregated MILP

optimize objective function

$$B_i^{MIN} \le B_{ic} \le B_i^{MAX} \quad \forall i, \forall c$$

$$B_{ics}^{I} = \rho_{is}^{I} B_{ic} \quad \forall i, \forall c, \forall s$$

$$B_{ics}^{O} = \rho_{is}^{O} B_{ic} \quad \forall i, \forall c, \forall s$$

$$R_{ic} = \alpha_i + \beta_i B_{ic} \quad \forall i, \forall c$$

$$\sum_i \sum_c B_{ics}^{O} \ge d_s \quad \forall s \in FP \qquad \text{(CP)}$$

Task[i,c] requires Unit[j] $\forall j, \forall i \in I(j), \forall c$

Task[i,c] requires R_{ic} *Utility[r]* $\forall i, \forall c$

Task[i,c] consumes B^{I}_{ics} *State[s]* $\forall i, \forall c, \forall s$

Task[i,c] produces B^{O}_{ics} *State[s]* $\forall i, \forall c, \forall s$

Task[i,c].end $\le MS$ $\forall i, \forall c$

Task[i,c] precedes Task[i,c+1] $\forall i, \forall c<|C|$

Tasks ⇒ Activities
Units ⇒ Unary Resources
Utilities ⇒ Discrete Resources
States ⇒ Reservoirs

Challenges for Decomposition for STN

1. **CP subproblem requires optimization**

   Maximization Profit/Minimization Makespan

2. **Variable duration (D_{ic})** (function batch size)

   Constraints in MP for B_{ic}, D_{ic} ⇒ Continuous variables – discretization

3. **Very difficult to create effective cuts**

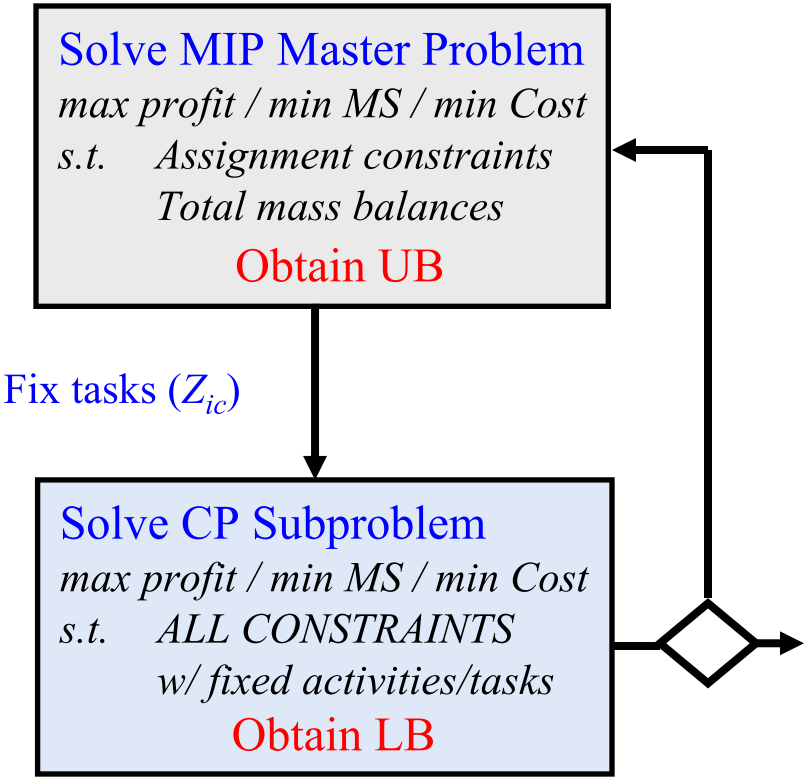
   Process network interconnected ⇒ No specific cuts

   Difficult to "locate" the source of infeasibility

   Many iterations

Proposed Hybrid Decomposition

$Z_{ic} = 1$ if copy c of task i is carried out



Solve MIP Master Problem

max profit / min MS / min Cost

s.t. Assignment constraints

Total mass balances

Obtain UB

Fix tasks (Z_{ic})

Solve CP Subproblem

max profit / min MS / min Cost

s.t. ALL CONSTRAINTS

w/ fixed activities/tasks

Obtain LB

$$\sum_{i\in I(j)}\sum_{c} D_{ic}Z_{ic} \le H \quad \forall j$$
$$B_i^{MIN} Z_{ic} \le B_{ic} \le B_i^{MAX} Z_{ic} \quad \forall i, \forall c$$
$$S_s = S_0 + \sum_i\sum_c \rho_{is}^{O} B_{ic} - \sum_i\sum_c \rho_{is}^{I} B_{ic} \quad \forall s$$
$$S_s \ge d_s \quad \forall s \in FP$$
$$S_s \le C_s \quad \forall s \in INT$$
$$Z_{ic+1} \le Z_{ic} \quad \forall i, \forall c < |C|$$

Integer Cuts

$$B_i^{MIN} \le B_{ic} \le B_i^{MAX} \quad \forall i, \forall c$$
$$B_{ics}^{I} = \rho_{is}^{I} B_{ic} \quad \forall i, \forall c, \forall s$$
$$B_{ics}^{O} = \rho_{is}^{O} B_{ic} \quad \forall i, \forall c, \forall s$$
$$R_{ic} = \alpha_i + \beta_i B_{ic} \quad \forall i, \forall c$$
$$\sum_i\sum_c B_{ics}^{O} \ge d_s \quad \forall s \in FP$$

Task[i,c] requires Unit[j] $\quad \forall j, \forall i \in I(j), \forall c$

Task[i,c] requires R_{ic} *Utility[r]* $\quad \forall i, \forall c$

Task[i,c] consumes B^{I}_{ics} *State[s]* $\quad \forall i, \forall c, \forall s$

Task[i,c] produces B^{O}_{ics} *State[s]* $\quad \forall i, \forall c, \forall s$

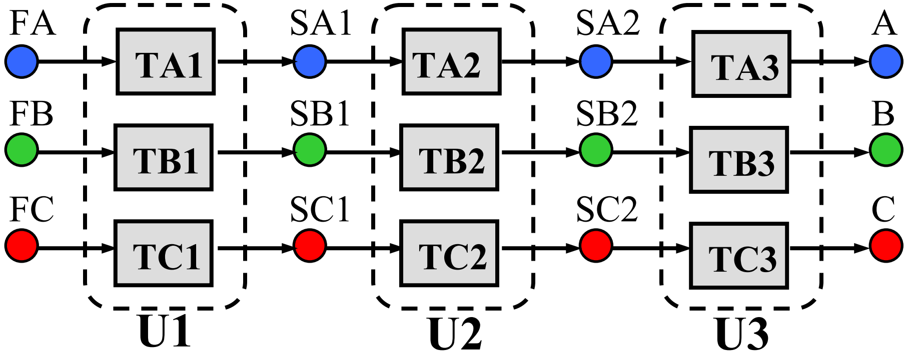
Task[i,c].end $\le MS$ $\quad \forall i, \forall c$

Task[i,c] precedes Task[i,c+1] $\quad \forall i, \forall c<|C|$

Preprocessing

GOAL: Determine Earliest Start Time (EST) and Latest Finish Time (LFT)

Example



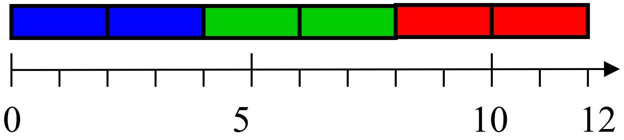
$H = 12\ hr$

$D_{Ai} = D_{Bi} = D_{Ci} = 2$

$\mathbf{EST_{U2} = 2\ hr}$

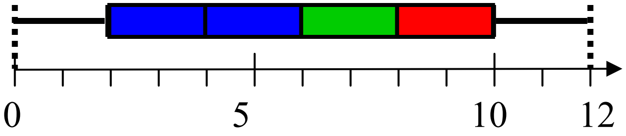
$\mathbf{LFT_{U2} = 10\ hr}$

Assignment Constraint: $\sum_{i \in I(j)} \sum_{c} D_{ic} Z_{ic} \leq H \quad \forall j$



⇒ Many redundant solutions

Using EST_{U2} and LFT_{U2}: Domain reduction



⇒ Rule-out infeasible/unrealistic solutions

$$\sum_{i \in I(j)} \sum_{c} D_{ic} Z_{ic} \leq H \quad \forall j \quad \Rightarrow \quad \begin{array}{l} \sum_{i \in I(j)} \sum_{c} D_{ic} Z_{ic} \leq LFT_j - EST_j \quad \forall j \\ \sum_{c} D_{ic} Z_{ic} \leq LFT_i - EST_i \quad \forall i \end{array}$$

Integer Cuts I

(1) $$\sum_{(i,c)\in B^k} Z_{ic} - \sum_{(i,c)\in N^k} Z_{ic} \le \left| B^k \right| - 1$$

B^k = {(*i,c*) | Z_{ic} = 1 in the current iteration *k*}
N^k = {(*i,c*) | $Z_{ic}=0$ in the current iteration *k*}

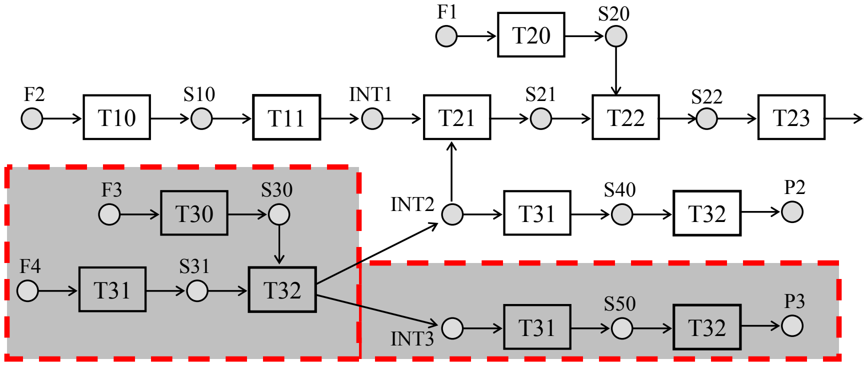
Very weak: Exclude only one assignment

(2) $$\sum_{(i,c)\in B^k} Z_{ic} \le \left| B^k \right| - 1$$

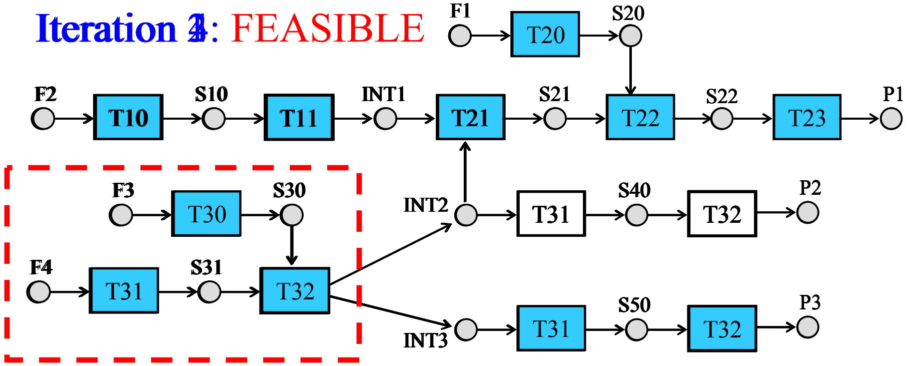
Stronger: Exclude the current assignment *k* and any superset of assignment *k*
i.e. any assignment *l* for which $B^l \supseteq B^k$.

Can be used for *Minimization of makespan and constant processing times.*

Integer Cuts II



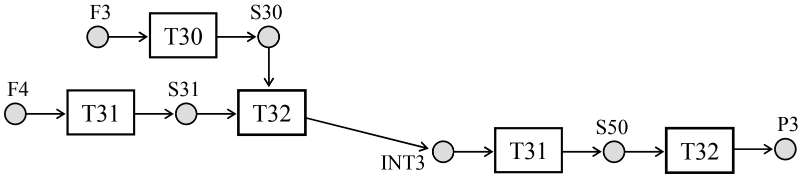
Iteration 4: FEASIBLE



Main Idea:

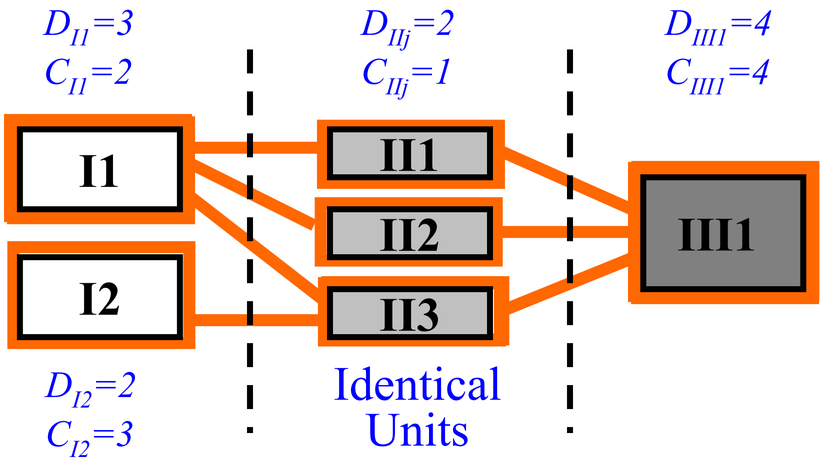
- Decompose plant network into “subnetworks”
- Pre-process each subnetwork separately ⇒ Create cuts for each subnetwork
- Add cuts to cut-pool ⇒ Start Master problem with many cuts

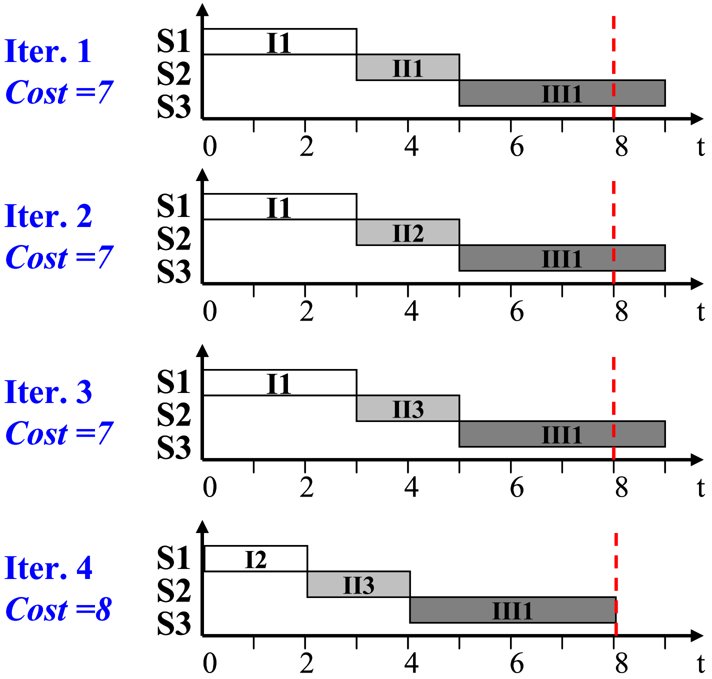
SUBNETWORKS: Individual Products
Individual machines (Shifting Bottleneck Heuristic)



Integer Cuts III

Integer Cuts II



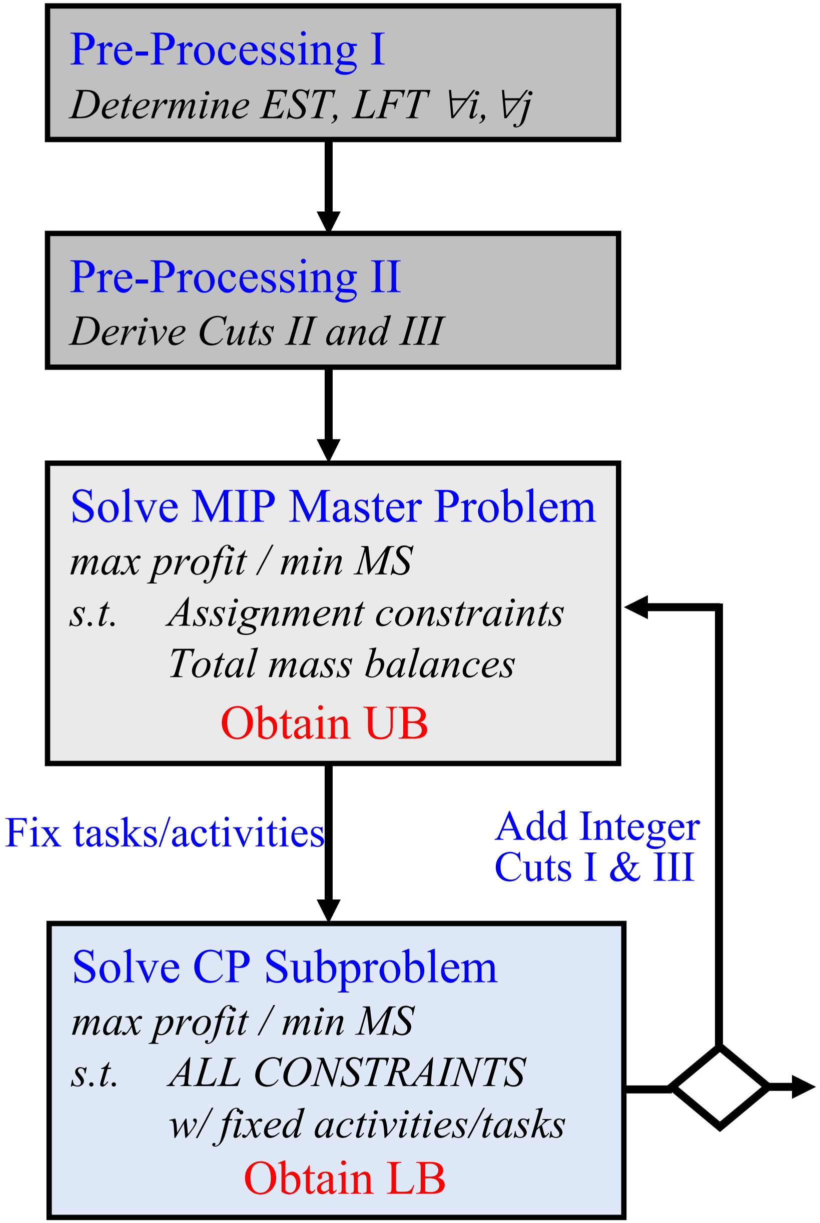


Need to exclude all configurations with I1 assignment

Main Idea:

- Define classes of identical equipment
- Define new binaries for classes of equipment
- Produce cuts in terms of the new binaries that exclude all “similar” configurations

Proposed Algorithm



Pre-Processing I

Determine EST, LFT $\forall i, \forall j$

Pre-Processing II

Derive Cuts II and III

Solve MIP Master Problem

max profit / min MS

s.t. Assignment constraints

Total mass balances

Obtain UB

Fix tasks/activities

Add Integer Cuts I & III

Solve CP Subproblem

max profit / min MS

s.t. ALL CONSTRAINTS

w/ fixed activities/tasks

Obtain LB

$$\sum_{i\in I(j)}\sum_{c} D_{ic}Z_{ic} \le LFT_j - EST_j \quad \forall j$$

$$\sum_{c} D_{ic}Z_{ic} \le LFT_i - EST_i \quad \forall i$$

$$B_i^{MIN} Z_{ic} \le B_{ic} \le B_i^{MAX} Z_{ic} \quad \forall i, \forall c$$

$$S_s = S_0 + \sum_i\sum_c \rho^{O}_{is} B_{ic} - \sum_i\sum_c \rho^{I}_{is} B_{ic} \quad \forall s$$

$$S_s \ge d_s \quad \forall s \in FP$$

$$S_s \le C_s \quad \forall s \in INT$$

$$Z_{ic+1} \le Z_{ic} \quad \forall i, \forall c < |C|$$

Integer Cuts I, II and III

Remarks

Convergence:

- Algorithm yields optimal solution
- Finite convergence
- Termination Criterion: Bounds depend on objective function

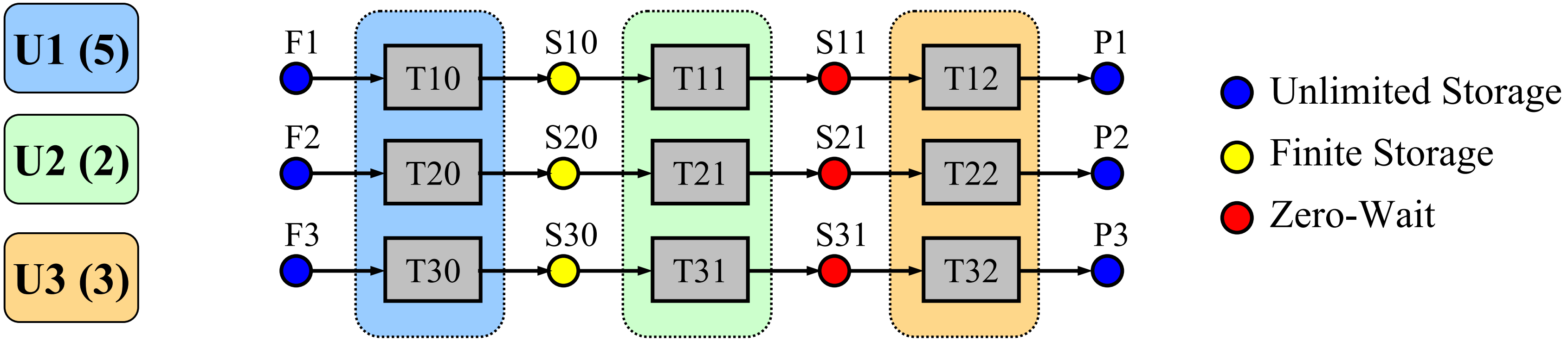
Implementation:

- ILOG's OPL Studio 3.5: MILP: CPLEX 7.5, CP: ILOG's Solver and Scheduler
- Solution of Subproblem: Optimization vs. (Successive) Feasibility problems
- Discretization needed for continuous processing times

Special cases:

- Sequence-dependent changeovers at no additional computational cost
- Release and due dates at no additional computational cost

Example (12,9,3) Revisited



Maximize Production

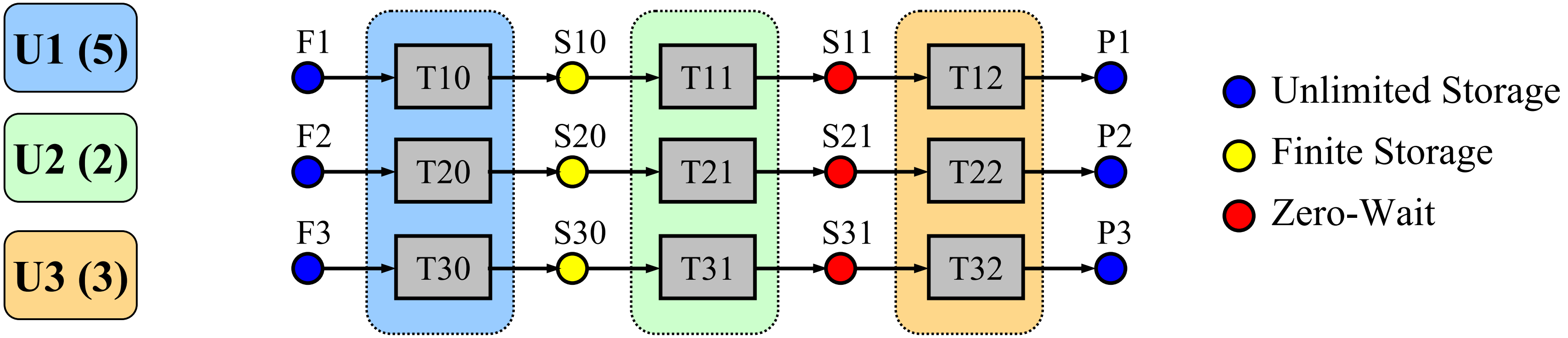
	MILP STN Model							Hybrid MILP/CP		
Horizon	n	0-1	Cont.	Constr.	LP Relax	Obj	CPU s	# iters	Obj	CPU s
15	10	180	1163	2473	14	12	58.5	3	12	0.32
20	12	216	1395	2965	18	16*	> 7200	3	16	0.37
25	15	270	1743	3703	24	22	> 7200	3	22	1.05

Minimize Makespan

	MILP STN Model							Hybrid MILP/CP		
Demand	n	0-1	Cont.	Constr.	LP Relax	Obj	CPU s	# iters	Obj	CPU s
4 / 5 / 6	12	216	1393	2938	12	19	32077	1	19	0.23
5 / 6 / 8	13	234	1509	3220	15	-	> 36000	1	23	21.02
5 / 8 / 10	15	270	1741	3712	18	-	> 36000	1	27	0.64

Orders of magnitude reduction in computational time!

Example (12,9,3): Variable Processing Times



		MILP STN Model ♦				MILP/CP Hybrid Scheme ♦♦		
Profit	**H (hours)**	**Obj ($\$10^3$)**	**Time points**	**Nodes**	**CPU sec**	**Obj ($\$10^3$)**	**# iter's**	**CPU sec**
Pr1	**15**	**11.9***	**12**	**215,214**	**36,000**	**12.0**	**11**	**2.69**
Pr2	**20**	**12.0 ****	**12**	**175,799**	**36,000**	**16.5**	**85**	**104.51**
Pr3	**25**	**16.0**	**11**	**1,669**	**356.0**	**20.5**	**32**	**36,000**
Makespan	**Demand (tons)**	**Obj (hr)**				**Obj (hr)**		
Ms1	**4/5/6**	**- *****	**13**	**98,400**	**36,000**	**19.7**	**29**	**10.02**
Ms2	**5/6/8**	**- *****	**14**	**58,500**	**36,000**	**23.8**	**29**	**32.62**
Ms3	**5/8/10**	**- *****	**15**	**52,400**	**36,000**	**28.2**	**49**	**1,878.7**

♦ Statistics of the MILP that gave the best solution in 36,000 CPU-s.

♦♦ Best solutions with 5 copies for each task.

* Optimality not proved: Gap 0.05%

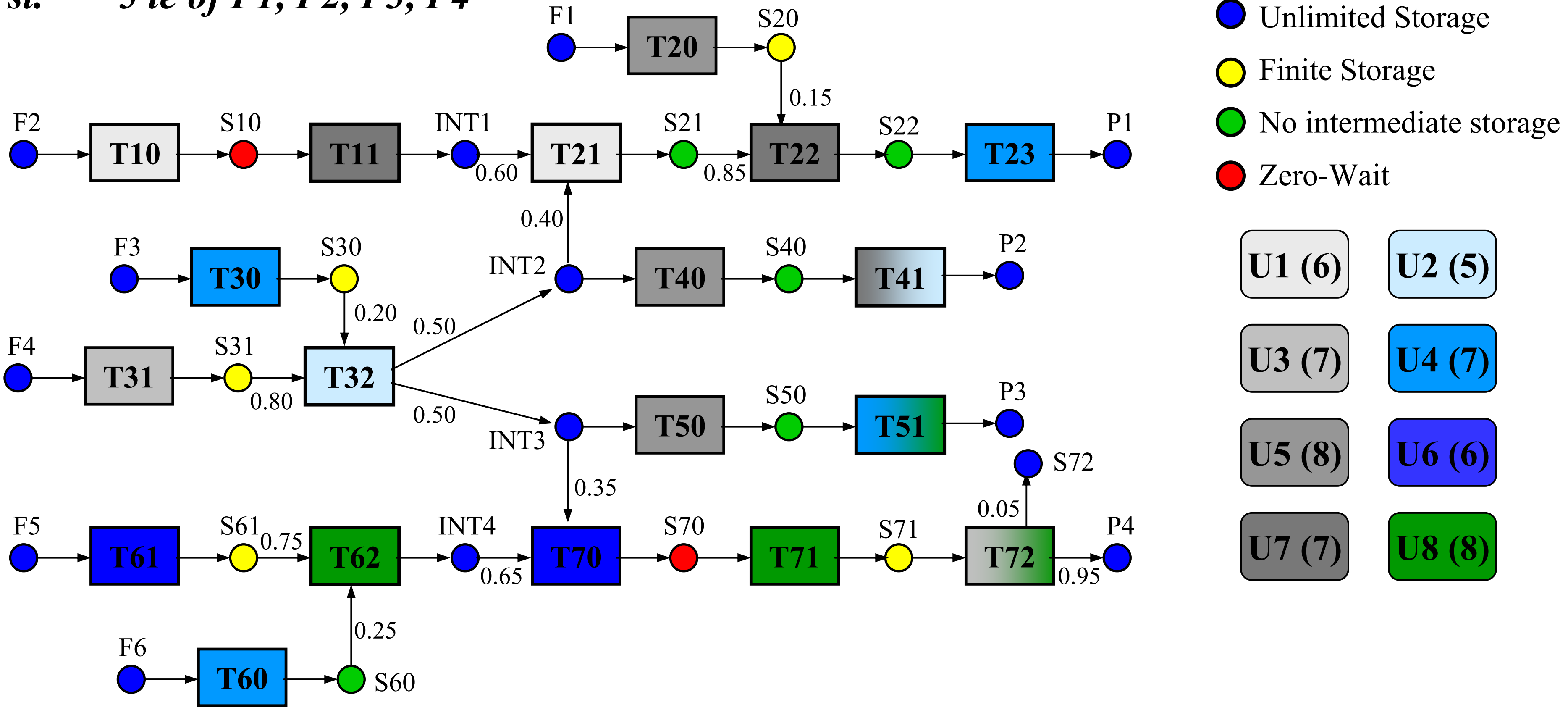
** Optimality not proved: Gap 32.9%

*** No integer solution found

Example (19, 27,8)

min MS

st. ***5 te of P1, P2, P3, P4***

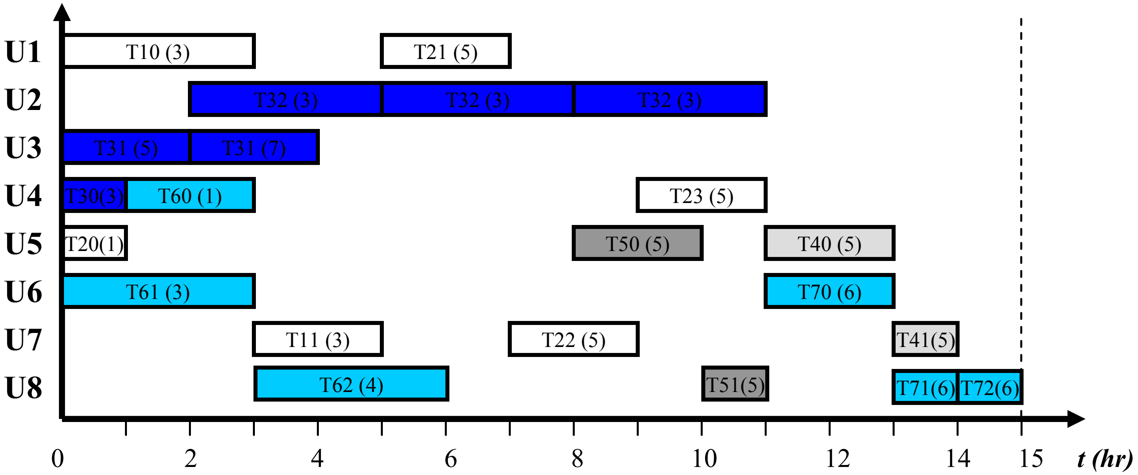


10 time points => 400 binaries
4000 continuous
6000 constraints

Intractable MILP

Example (19,27,8)

Without changeover times



P1 P2 P3 P4 Spl.

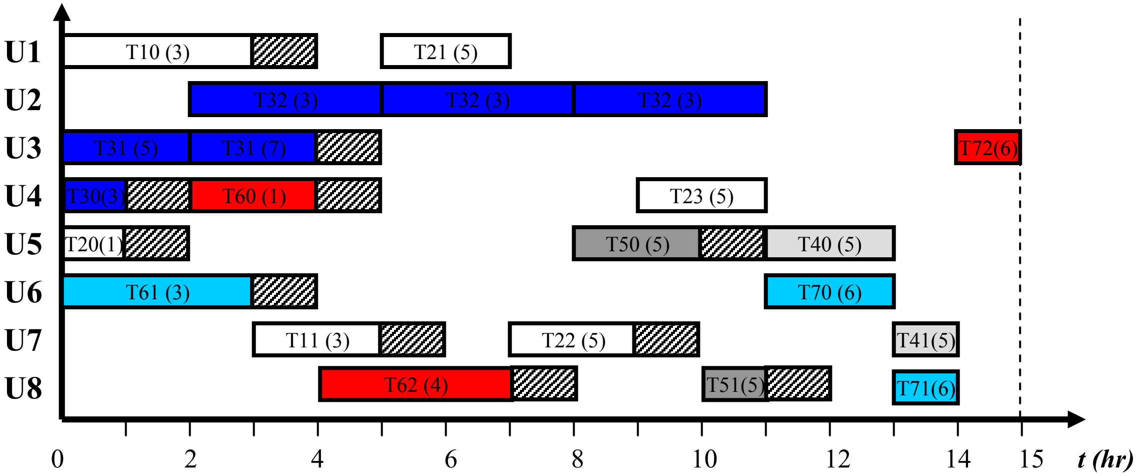
○ ● ● ● ●

MS = 15 hours

9 iterations

4.4 CPU seconds!!

With changeover times of 1 hr



P1 P2 P3 P4 Spl. Ch.

○ ● ● ● ● ●

MS = 15 hours

9 iterations

4.8 CPU seconds!!

▨ Changeover Time

Questions

1. **Can the General Continuous-time MILP/CP model be effectively applied to specialized cases?**

    multistage batch plants

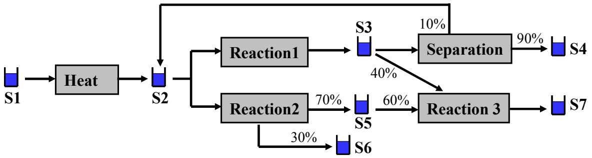
    single stage parallel units

2. **Does it pay to apply a different algorithm than CP to the sequencing subproblem?**

Special cases

Plant Topology

Multipurpose Batch Plant



Master MIP Problem

Z_{ic} = *1* if batch *c* of task *i* is carried out
B_{ic} = batch size of batch c of task *i*
S_s = inventory level of state *s*

$$\sum_{i \in I(j)} \sum_{c} D_{ic} Z_{ic} \le MS - ST_j - EST_j \quad \forall j$$

$$B_i^{MIN} Z_{ic} \le B_{ic} \le B_i^{MAX} Z_{ic} \quad \forall i, \forall c$$

$$S_s = S_0 + \sum_i \sum_c \rho^O_{is} B_{ic} - \sum_i \sum_c \rho^I_{is} B_{ic} \quad \forall s$$

$$S_s \ge d_s \quad \forall s \in FP$$

$$S_s \le C_s \quad \forall s \in INT$$

CP Subproblem

$$B_i^{MIN} \le B_{ic} \le B_i^{MAX} \quad \forall i, \forall c$$

$$R_{ic} = \alpha_i + \beta_i B_{ic} \quad \forall i, \forall c$$

$$\sum_i \sum_c B^O_{ics} \ge d_s \quad \forall s \in FP$$

Task[i,c] requires Unit[j] $\forall j, \forall i \in I(j), \forall c$

Task[i,c] requires R_{ic} *Utility[r]* $\forall i, \forall c$

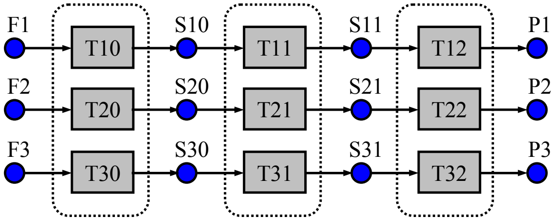
Task[i,c] consumes $\rho^c_{is} B_{is}$ *State[s]* $\forall i, \forall c, \forall s$

Task[i,c] produces $\rho^p_{is} B_{ics}$ *State[s]* $\forall i, \forall c, \forall s$

Task[i,c].end $\le MS$ $\forall i, \forall c$

General Multi-stage Plant

- No shared utilities
- No recycle streams;
- No batch splitting/mixing



$$\sum_{i \in I(j)} \sum_{c} D_{ic} Z_{ic} \le MS - ST_j - EST_j \quad \forall j$$

$$B_i^{MIN} Z_{ic} \le B_{ic} \le B_i^{MAX} Z_{ic} \quad \forall i, \forall c$$

$$S_s = S_0 + \sum_i \sum_c 1 \cdot B_{ic} - \sum_i \sum_c 1 \cdot B_{ic} \quad \forall s$$

$$S_s \ge d_s \quad \forall s \in FP$$

$$S_s \le C_s \quad \forall s \in INT$$

$$B_i^{MIN} \le B_{ic} \le B_i^{MAX} \quad \forall i, \forall c$$

$$\sum_i \sum_c B^O_{ics} \ge d_s \quad \forall s \in FP$$

Task[i,c] requires Unit[j] $\forall j, \forall i \in I(j), \forall c$

Task[i,c] consumes $1 \cdot B_{is}$ *State[s]* $\forall i, \forall c, \forall s$

Task[i,c] produces $1 \cdot B_{ics}$ *State[s]* $\forall i, \forall c, \forall s$

Task[i,c].end $\le MS$ $\forall i, \forall c$

Special Cases

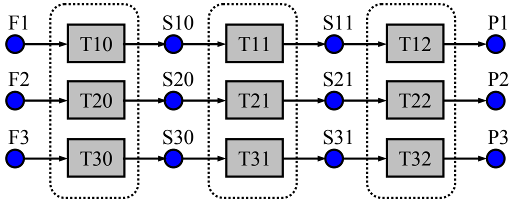
Plant Topology

Master MIP Problem

CP Subproblem

Multi-stage: unlimited storage

- No need to account for states



$$\sum_{i\in I(j)}\sum_{c} D_{ic} Z_{ic} \le MS - ST_j - EST_j \quad \forall j$$

$$B_i^{MIN} Z_{ic} \le B_{ic} \le B_i^{MAX} Z_{ic} \quad \forall i, \forall c$$

$$S_s = S_0 + \sum_i \sum_c 1 \cdot B_{ic} - \sum_i \sum_c 1 \cdot B_{ic} \quad \forall s$$

$$S_s \ge d_s \quad \forall s \in FP$$

$$B_i^{MIN} \le B_{ic} \le B_i^{MAX} \quad \forall i, \forall c$$

$$\sum_{i\in O(s)} \sum_{c} B_{is} \ge d_s \quad \forall s \in FP$$

Task[i,c] requires Unit[j] $\forall j, \forall i \in I(j), \forall c$

Task[i,c] precedes Task[i',c] $\forall (i,i') \in A, \forall c$

Task[i,c].end $\le MS$ $\forall i, \forall c$

Multi-stage: demand in orders

- Fixed batches & batch-sizes
- Drop *c* index, *B* variables
- *Task*→(*order*,*stage*,*unit*): *i*→(*o*,*k*,*j*)
- Add assignment constraint

$$\sum_{o\in O(j)} D_{okj} Z_{okj} \leqslant (MS - ST_k) - EST_k \, \forall j \in J(k)$$

$$\sum_{j\in J(k)} Z_{okj} = 1 \quad \forall o, \forall k$$

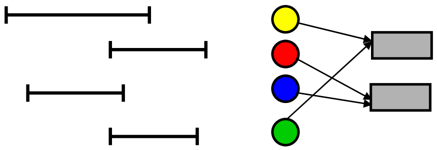
Task[o,k,j] requires Unit[j] $\forall o, \forall k, \forall j | Z_{okj}=1$

Task[o,k,j] preceeds Task[o,k-1,j'] $\forall o, \forall k$

Task[o,k,j].end $\le MS$ $\forall o, \forall k, \forall j | Z_{okj}=1$

Single-stage

- Drop stage index *k*



$$\sum_{o\in O(j)} D_{oj} Z_{oj} \leqslant \max_{o\in O}\{d_o\} - \min_{o\in O}\{r_o\} \forall j$$

$$\sum_{j} Z_{oj} = 1 \quad \forall o$$

Preprocessing: Better Cover Cuts

Task[o,j].start $\ge r_o$ $\forall o, \forall j$

Task[o,,j].end $\le d_o$ $\forall o, \forall j$

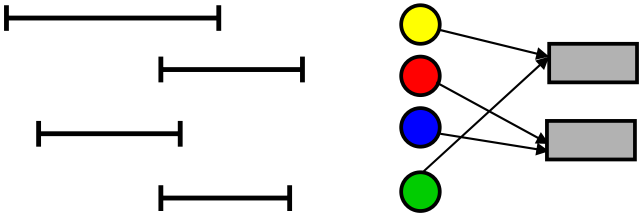
Task[o,j] requires Unit[j] $\forall o, \forall k, \forall j | Z_{ok}=1$

Task[o,k,j].end $\le MS$ $\forall o, \forall k, \forall j | Z_{ok}=1$

Parallel Machines Revisited

Minimization of Assignment Cost in Parallel Machine Problem

N jobs (with release and due times) have to be scheduled on M machines, minimizing cost



Computational Results for examples of Jain & Grossmann (2001)

Problem		Obj	MILP	CP	Jain & Grossmann		Bockmayr & Pisaruk		Proposed Approach*		
M	N		Time	Time	Iter's	Time	Nodes	Time	Iter's	PP Cuts	Time
3	12	101	220	3.8	31	12.7	78	0.5	2	108	0.6
3	12	83	1.8	0.4	1	0.5	149	0.9	1	20	0.2
5	15	115	180.4	553.5	18	5.1	144	0.7	3	303	1.5
5	15	102	61.8	9.3	1	4.3	177	0.9	1	113	0.2
5	20	158	>20000	68853.5	31	41.5	924	4.0	2	581	2.7
5	20	140	106.3	2673.9	6	162	38303	245.8	1	302	0.3

* In Mosel using XPRESS Solver for Master Problem and Carlier & Pinson algorithm for subproblem

Preprocessing by pairs of jobs

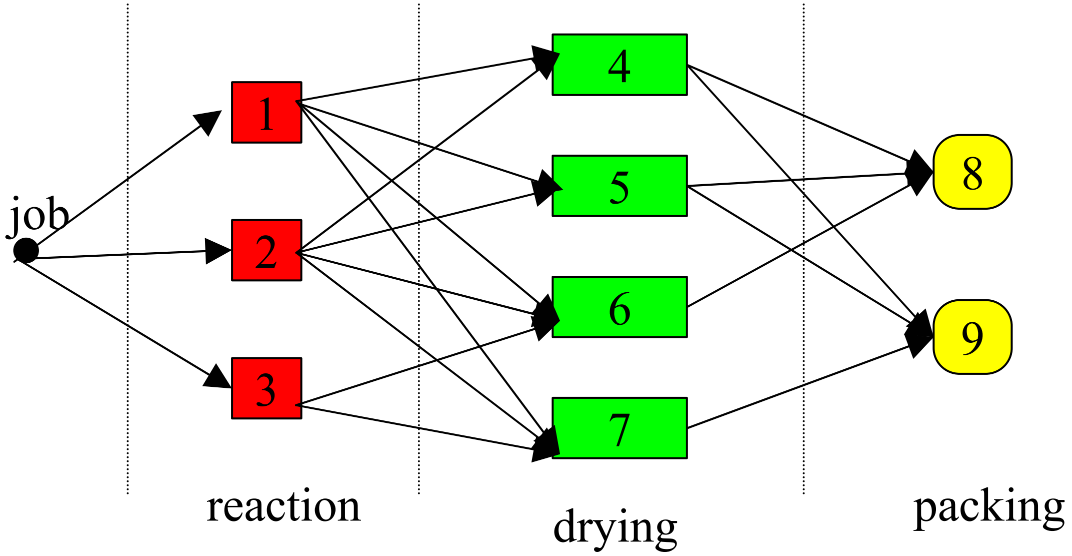
Multistage Scheduling Problem

Minimization of cost of multi-stage problem for orders with release and due times

N orders to be processed sequentially on K stages, where each stage consists of M_k units.

Each order i has release r_i and due d_i time, and processing cost c_{ij} and processing time pt_{ij} on unit j.

Objective: minimize processing costs subject to meeting the release and due times.

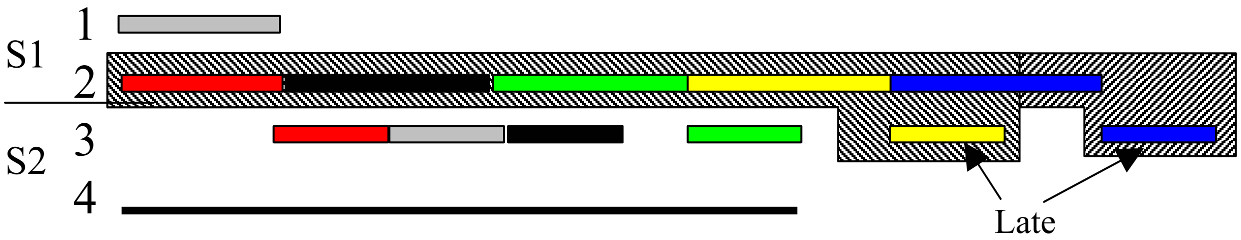


3 stages, 9 units

Problem-specific algorithms to solve subproblem

1. Use CP or MILP (fixed assignment) to derive CUT-H ***Harjunkoski, Grossmann (2002)***

Cut-H: tracing bottleneck paths that yield violation of due dates



cut 1: $y_{A2}+y_{B2}+y_{C2}+y_{D2}+y_{E2}+y_{A3}\leq 5$

cut 2: $y_{B2}+y_{C2}+y_{D2}+y_{E2}+y_{D3}\leq 4$

2. Sequencing subproblem is a *traditional* job-shop scheduling problem

⇒ Efficient algorithms

Apply Shifting Bottleneck Procedure (Adams, Balas, Zawack, 1988) to solve the subproblem

Note: ***Cuts not guaranteed to be valid, but rarely fail in practice***

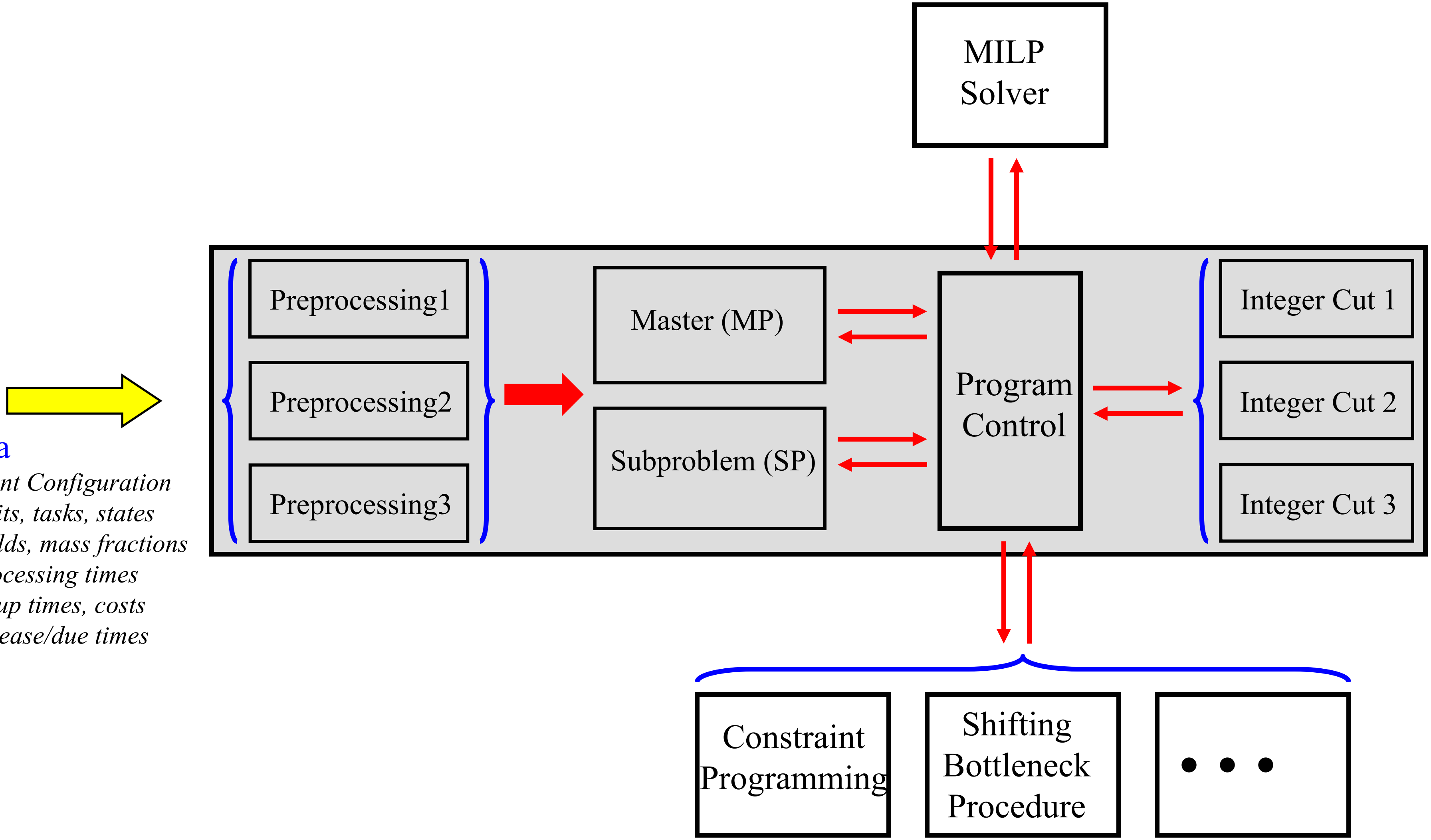
Computational Results

CP/Bottleneck cuts **Shifting bottleneck**

CPLEX 6.5/OPL Studio3.5/933MHz XPRESS-MP 14.2/1GHz Pentium III

	Obj	**MIP**	**CP**	MIP/CP: CUT H		**MIP/SBP**	
		CPU s	**CPU s**	Iter’s/Cuts	CPU s	Iter’s/Cuts	**CPU s**
P1D1	39	**0.1**	**0.1**	5/4	0.1	2/1	**0.2**
P1D2	112	**0.1**	**0.1**	3/2	0.1	2/1	**0.2**
P2D1	153	**4.4**	**0.1**	16/19	0.7	4/4	**1.3**
P2D2	188	**0.8**	**0.3**	4/3	0.1	1/0	**0.2**
P3D1	56	**446.6**	**4.2**	29/45	8.7	26/32	**16.9**
P3D2	1113	**0.4**	**1375.0**	5/5	0.4	2/2	**1.1**
P4D1	149	**27.3**	**447.2**	17/22	28.8	12/15	**7.8**
P4D2	946	**1.4**	**359.0**	20/26	11.0	6/14	**9.8**
P5D1	111	**4041.7**	**293.0**	43/56	318.4	19/25	**17.2**
P5D2	704	**2767.8**	**712.6**	2/2	13.7	2/1	**0.4**

MILP/CP Framework for Batch Scheduling



Data

- *Plant Configuration*
- *Units, tasks, states*
- *Yields, mass fractions*
- *Processing times*
- *Setup times, costs*
- *Release/due times*

Conclusions

- **Hybrid MILP/CP exploits respective strengths of methods**
  - Natural decomposition of assignment/sequencing
  - Challenge: integer cuts
- **Successful applications in batch scheduling problems**
  - Single stage, parallel lines

    Clean partition of MILP/CP constraints
  - STN model

    Aggregated MILP, preprocessing, various integer cuts are key

Order magnitude reduction in CPU time!

- **General STN model unifies treatment to specialized cases**