Recent Advances in Chemical Production Scheduling

Notation, Models, and Solution Methods

In Honor of Ignacio Grossmann's 65th Birthday

Christos T. Maravelias

Department of Chemical and Biological Engineering

University of Wisconsin – Madison

2014 Annual AIChE Meeting

November 16-21, Atlanta, GA

Outline

- Introduction
- New MIP scheduling models
- Solution methods
  - Tightening methods
  - Multi-grid models
  - Reformulations and branching strategies
- Concluding thoughts

Preliminaries

- Widespread applications in the chemical industries
  - Batch process scheduling (e.g., pharma, food industry, fine chemicals)
  - Continuous process scheduling (e.g., polymerization)
  - Transportation and delivery of crude oil
- Scheduling in PSE
  - First publications in early 1980s; focused on *sequential* facilities (Rippin, Reklaitis)
  - Problems in *network* structures addressed in early 1990s (Pantelides et al.)
- Very challenging problem: Small problems can be very hard
  - Most *Open* problems in MIPLIB are scheduling related
    - Railway scheduling: 1,500 constraints, 1,083 variables, 794 binaries
    - Production planning: 1,307 constraints, 792 variables, 240 binaries

Problem Statement

Given are:

a) Production facility data; e.g., unit capacities, unit connectivity, etc.
b) Production recipes; i.e., mixing rules, processing times/rates, utility requirements, etc.
d) Production costs; e.g., raw materials, utilities, changeover, etc.
e) Material availability; e.g., deliveries (amount and date) of raw materials.
f) Resource availability; e.g., maintenance schedule, resource allocation from planning, etc.
g) Production targets or orders with due dates.

(a, b, d: **Facility and recipe data**; e, f, g: **Input from other planning functions**)

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

Alternative objective functions are the minimization of tardiness or lateness (minimization of backlog cost) or the minimization of earliness (minimization of inventory cost) or the maximization of profit.

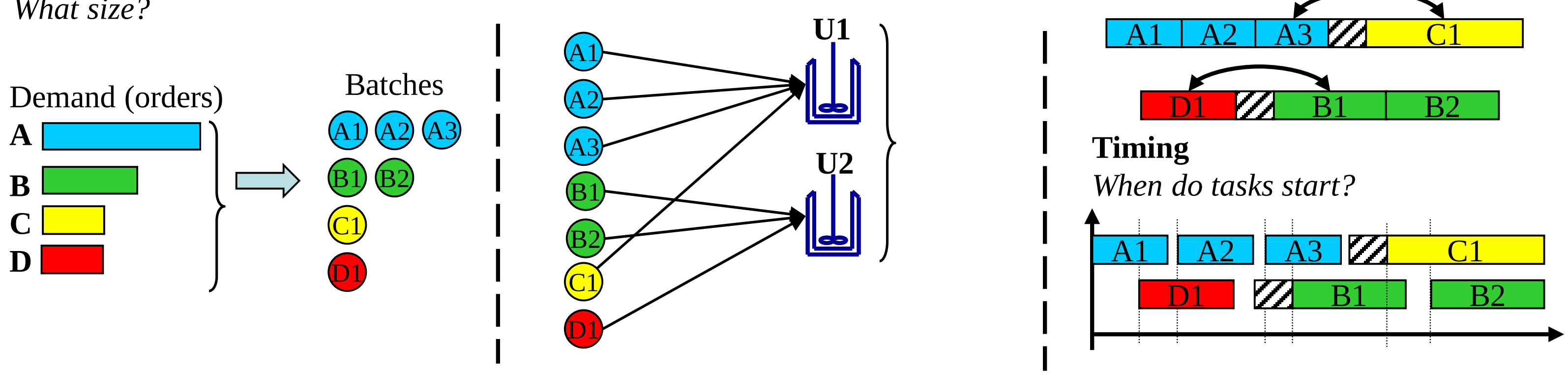
In the general problem, we seek to optimize our objective by making four types of decisions:

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

Task selection (batching)
How many tasks/batches?
What size?

Task-resource Assignment
What resources each task requires?

Sequencing (for unary resources)
In what sequence are batches processed?



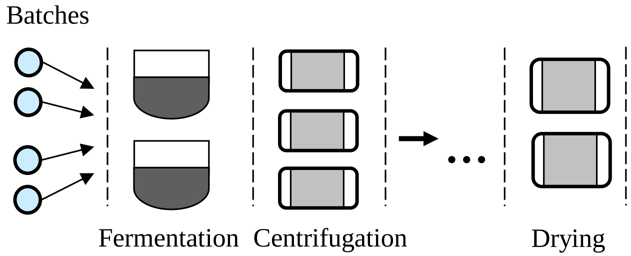
Problem Classes

α / β / γ

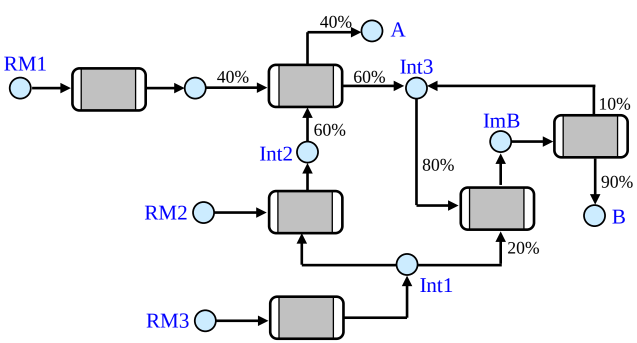
- α Production environment
- β Processing characteristics
- γ Objective functions

- Discrete manufacturing
  - A *job* (e.g., circuit chip) moves through *operations* consisting of parallel *machines*
  - Each job is not split into multiple jobs; jobs are not *merged*
- Chemical production: *tasks* involve fluids
  - *Fluids* coming from different batches can be mixed; fluids of different types can be mixed to be converted to a new fluid; output of a task (stored in a vessel) can be used in multiple downstream tasks.
  - No mixing/splitting restrictions may be *added* (e.g., quality control)

- ***Sequential*** environment
  - Problems similar to discrete manufacturing due to material handling restrictions



- ***Network*** environment
  - Bathes and materials are split
  - Recycle streams, etc.
  - Problems different from discrete manufacturing

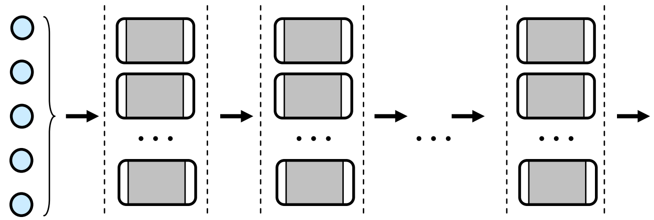


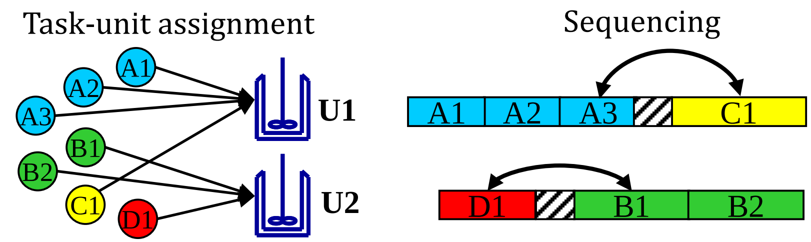
Maravelias (2011), *AIChE Journal*

From *Problems* to *Models* (1990-2008)

Batch-based formulations

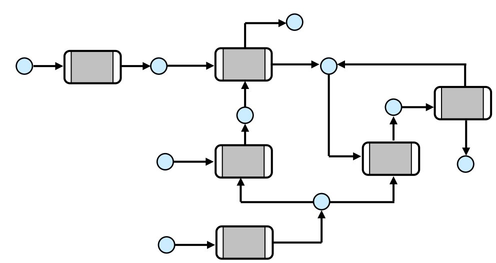
- Exploit sequential process structure
- *Batch-centric* approach
  - Batches are assigned to units
  - Sequencing constraints for batches in the same unit
- Batching problem is solved prior to scheduling
- Other common assumptions:
  - No storage constraints
  - No utility constraints





Material-based formulations

- Handle *network* environments (mixing, splitting, recycle)
- *Material-based* approach
  - Tasks consume/produce materials & resources
  - Material & resource balances over time
- Consider batching and scheduling simultaneously
- Consider wide range of processing constraints
  - Storage & utility constraints



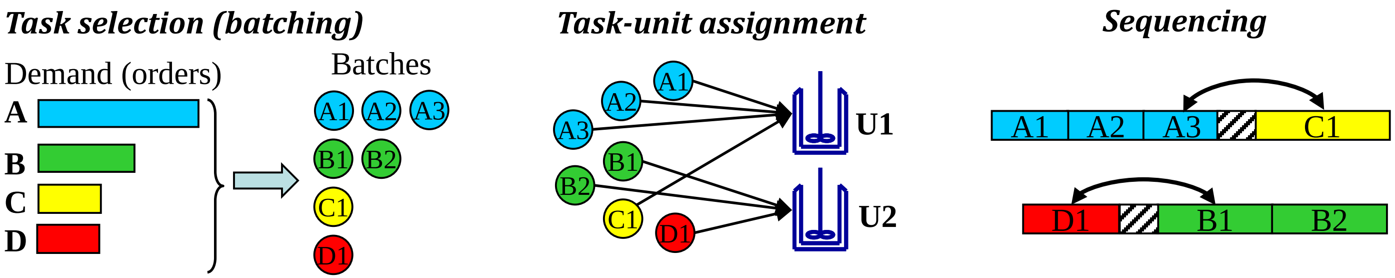
Can we formulate models for general problems in sequential environments?

Can we formulate models applicable to all/combined environments?

General Models for Sequential Environments

1) Simultaneous batching, assignment and sequencing [1]

- Variable number of batches ⇒ variable batchsizes ⇒ variable processing times
- Introduce new *selection* variables; if a batch is selected then it is assigned and sequenced



2) Simultaneous batching, assignment, sequencing + general *storage* constraints [2]

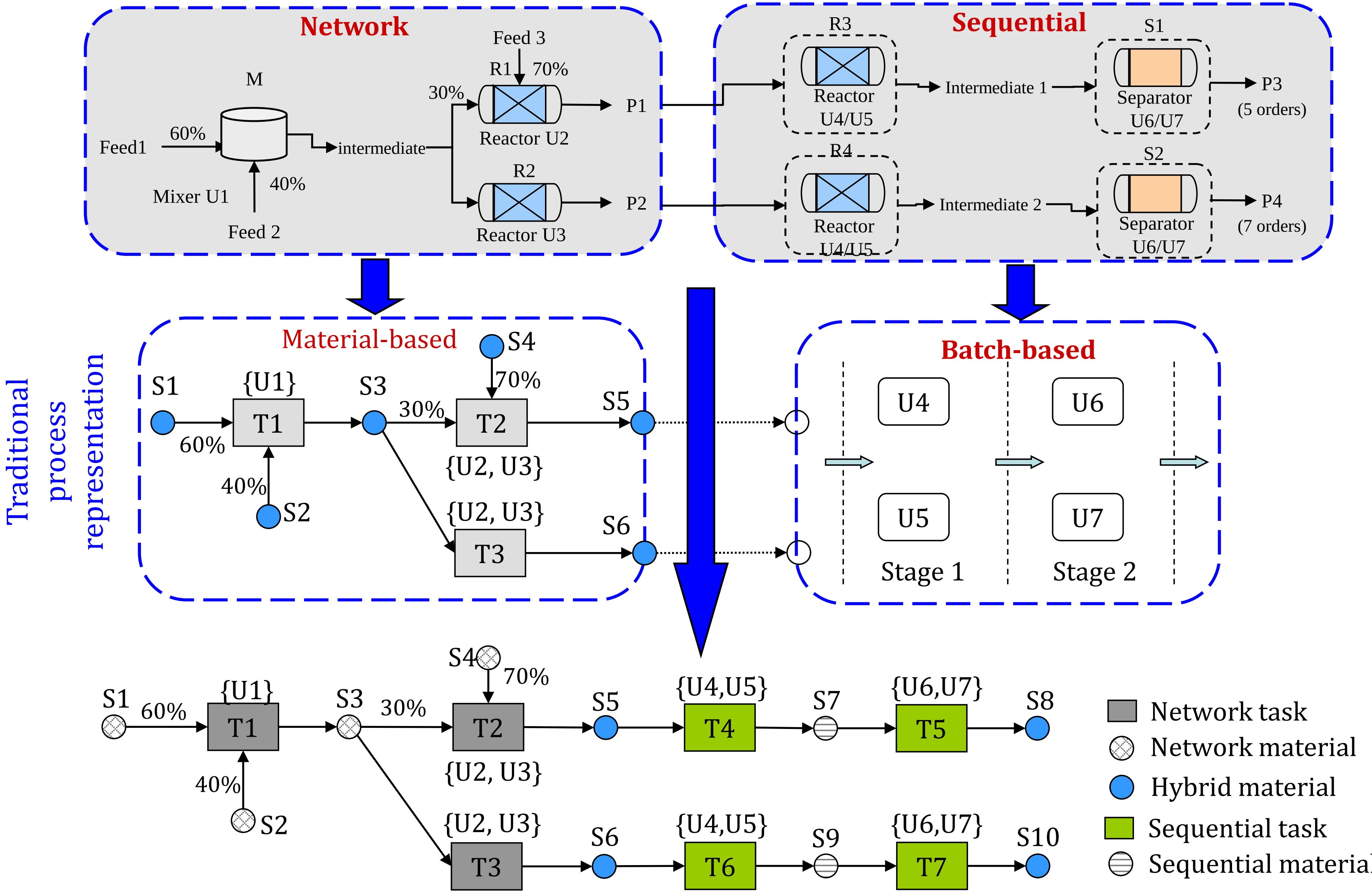
- Consider timing (waiting in units & tanks) and capacity (tank number & size) constraints
- Storage tanks modeled as additional resources

3) Simultaneous batching, assignment, sequencing + storage and *utility* constraints [3]

- Storage tanks modeled as resources
- Adopt common (discrete) time grid to monitor utilities
- Express resource balance constraints for units, tanks, and utitilities

[1] Prasad & Maravelias, *Computers and Chemical Engineering*, 32 (6), 2008
[2] Sundaramoorthy & Maravelias, *Industrial and Engineering Chemistry Research*, 47 (17), 2008
[3] Sundaramoorthy & Maravelias, *Industrial and Engineering Chemistry Research*, 48 (13), 2009

Unified Framework



Sundaramoorthy & Maravelias, AIChE, 2011

Remarks

Sequential environments

- Simultaneous batching, assignment, sequencing
- Storage policies, shared storage vessels, general resource constraints

Network environments*

- Non-simultaneous and multiple material transfers[1]
- Resource-constrained material transfers and changeover activities[2]
- . . .

Combined environments[3] *

- Upstream sequential followed by downstream network, followed by continuous processing

All of the above[4] *

* Material-based models

[1] Gimenez et al., *Computers and Chemical Engineering*, 33 (9), 2009
[2] Gimenez et al., *Computers and Chemical Engineering*, 33 (10), 2009
[3] Sundaramoorthy & Maravelias, *AIChE J.*, 57(3), 2011
[4] Velez & Maravelias, *Industrial & Engineering Chemistry Research*, 52(9), 2013

Material-Based Models

Problem Statement

Given are a set of tasks $i \in \mathbf{I}$, processing units $j \in \mathbf{J}$, materials (states) $k \in \mathbf{K}$, and resources $r \in \mathbf{R}$

- Task i in unit j has processing time τ_{ij} and variable batchsize in $[\beta_j^{min}\ \beta_j^{max}]$
- Each task can consume/produce multiple materials; conversion coefficient ρ_{ik}
- Material k can be produced/consumed by multiple tasks; is stored in a dedicated tank

- Tasks trigger changes in:
  (1) unit utilization, (2) material inventories, (3) resource utilization/availability
- Tasks are mapped onto one or more time grids

Modeling of Time

- *Early* Discrete[1,2,3,4] : horizon η divided into ***uniform*** periods of known length δ
  - Processing times are approximated
  - Large-scale MIP models
- Continuous[5,6]: horizon η divided into periods of unknown (variable) length
  - Exact processing times
  - Smaller MIP models
- Continuous time models have been studied extensively since 1995[7,8,9,10,11,12]
  unit-specific time grids; wide range of constraints, etc.
- Discrete time models have two key advantages
  - Are easy to modify and extend to account for various features (used widely in industry)
  - Extensions do not lead to harder problems

[1] Kondili et al., *Computers and Chemical Engineering*, 17, 1993
[2] Shah et al., *Computers and Chemical Engineering*, 17, 1993
[3] Pantelides, *2nd Conference on Foundations of Computer Aided Process Operations*, 1994
[4] Bassett et al., *AIChE J.*, 42(12), 1996
[5] Zhang & Sargent, *Computers and Chemical Engineering*, 1996
[6] Schilling & Pantelides, *Computers and Chemical Engineering*, 20, 1996
[7] Ierapetritou & Floudas, *Industrial and Engineering Chemistry Research*, 37, 1998
[8] Mockus & Reklaitis, *Industrial and Engineering Chemistry Research*, 38, 1999
[9] Castro et al., *Industrial and Engineering Chemistry Research*, 40(9), 2011
[10] Maravelias & Grossmann, *Industrial and Engineering Chemistry Research*, 24, 2003
[11] Sundaramoorthy & Karimi, *Chemical Engineering Science*, 60, 2005
[12] Gimenez et al., *Computers and Chemical Engineering*, 22, 2009

Major Types of Scheduling Constraints

1. A unit can process at most one task at a time

- X_{ijt} = 1 if task i starts in unit j at time t
- Basic difference between discrete- and continuous-time formulations

$$\sum_{i\in\mathbf{I}_j}\sum_{n'\geq n-\tau_{ij}+1}^{n} X_{ijn'} \leq 1,\ \ \forall j,n$$

2. Batch-sizes are within the unit capacity

- B_{ijt} = batch-size of task i in unit j starting at time t
- If a task is carried out then its batch-size is within a lower and upper bound

$$\beta_j^{min} X_{ijn} \leq B_{ijn} \leq \beta_j^{max} X_{ijn}, \qquad \forall i, j \in \mathbf{J}_i, n$$

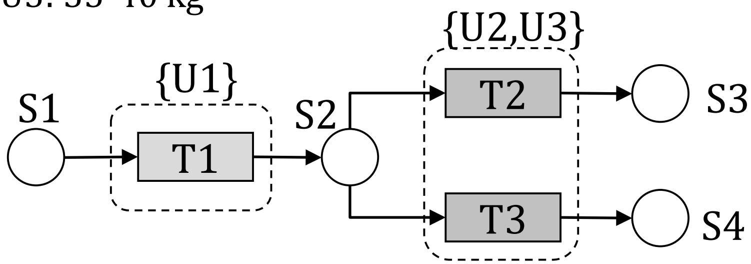
3. Material balance constraints

- S_{st} = inventory of material s at time t

$$S_{kn} = S_{k,n-1} + \sum_{i\in\mathbf{I}_k^+} \rho_{ik} \sum_{j\in\mathbf{J}_i} B_{ij,n-\tau_{ij}} + \sum_{i\in\mathbf{I}_k^-} \rho_{ik} \sum_{j\in\mathbf{J}_i} B_{ijn} + \xi_{kn} \leq \gamma_k,\ \ \forall k,n$$

Tightening Methods: Motivating Example

Capacities:
U1: 25-60 kg
U2: 40-50 kg
U3: 35-40 kg



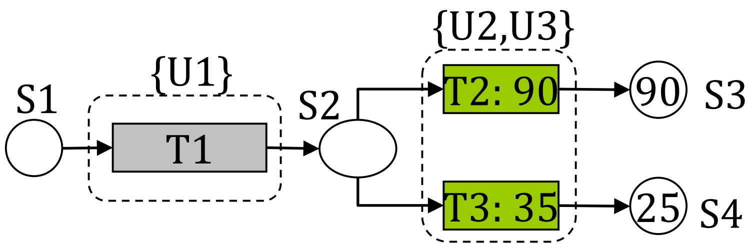
Demand: 90 kg S3 and 25 kg S4

	# of Batches			Min. Cost
	T1	T2	T3	($)
LP-Relaxation				
No tightening	1.9	1.8	0.5	76.7
With tightening*	3	2	1	105
Optimal solution	3	2	1	105

- Calculation of some bounds for specific models using auxiliary LPs & MIPs
  Burkard & Hatzl, 2006; Janak & Floudas, 2008
- ***Can we generalize to all networks and models and calculate bounds fast?***

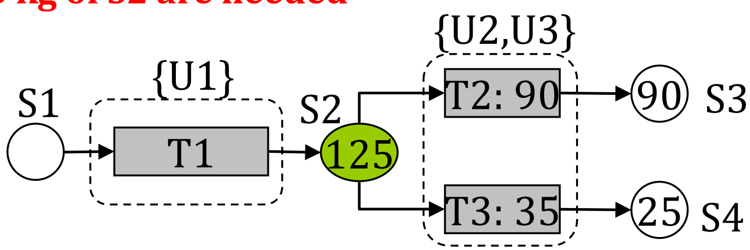
1. Demand ⇒ number of batches:

- T2 must produce **90 kg in at least 2 batches**
- T3 has to produce 25 kg, but the minimum capacity is 35 kg, so **T3 must produce 35 kg in 1 batch**



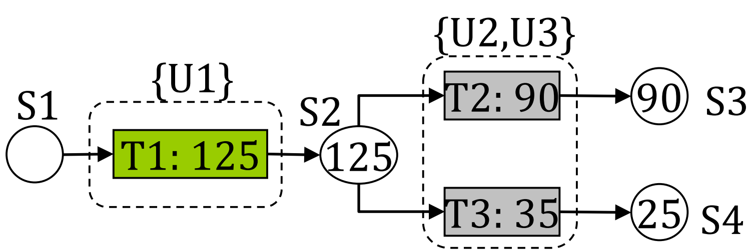
2.Number of batches ⇒ intermediate demand:

- **125 kg of S2 are needed**



3.Intermediate demand ⇒ number of batches:

- T1 has to produce at least 125 kg;
- Capacity of U1 is 25-60 kg, so **3 batches are required**

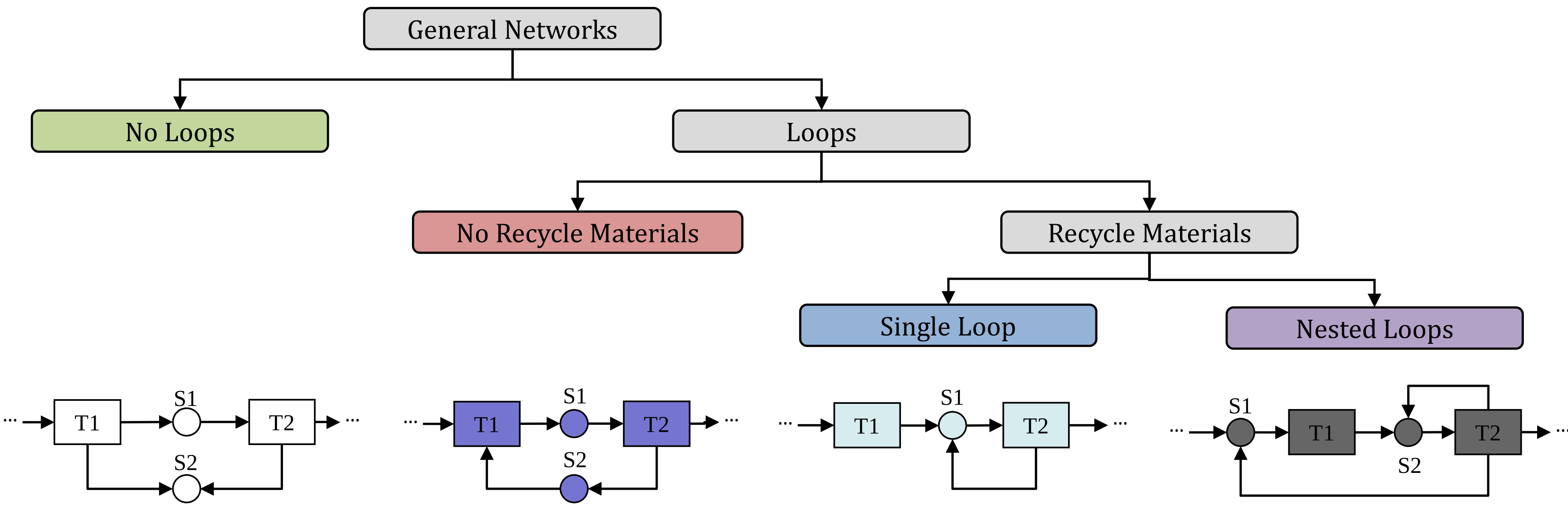


Tightening Methods: Remarks

Challenges

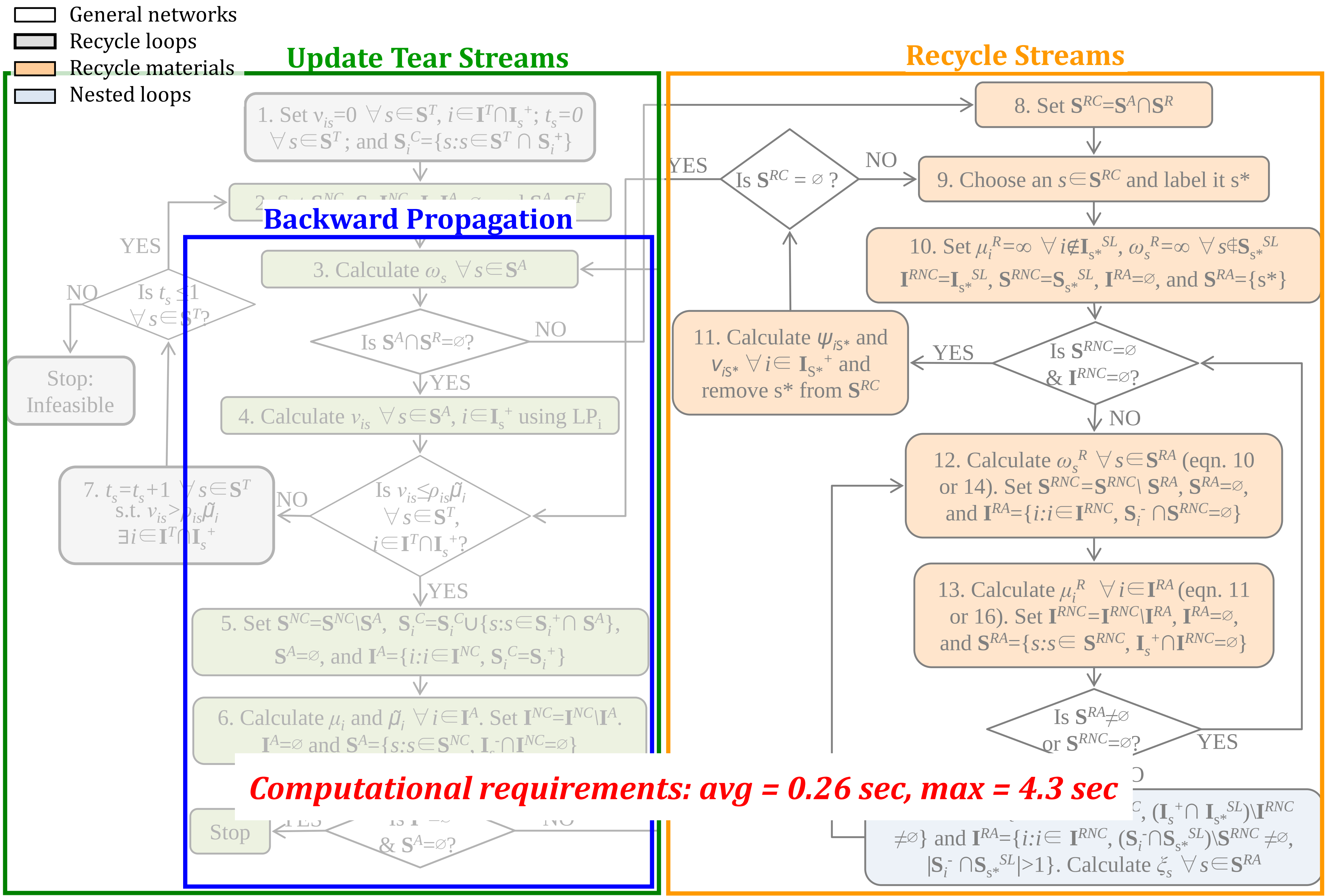
- A material can be produced by many tasks
- A task can be carried out in many units (of unequal capacity)
- ***Recycle streams***

Algorithm depends on network structure



- Use tear streams; iterate until bounds converge
- Use combination of backward and forward propagation for nested loops

Tightening Methods: The Algorithm



Computational requirements: avg = 0.26 sec, max = 4.3 sec

Tightening Constraints

The algorithm calculates:

ω_s: minimum required amount of material *s*

μ_i^1: minimum cumulative production of task *i*

Problem data

β_j^{max} = maximum capacity of unit *j*

ρ_{is}^+ = fraction of material *s* produced by task *i*

Variable in all time-indexed formulations

X_{ijt} = 1 if task *i* starts in unit *j* at time *t*

Tightening Constraints

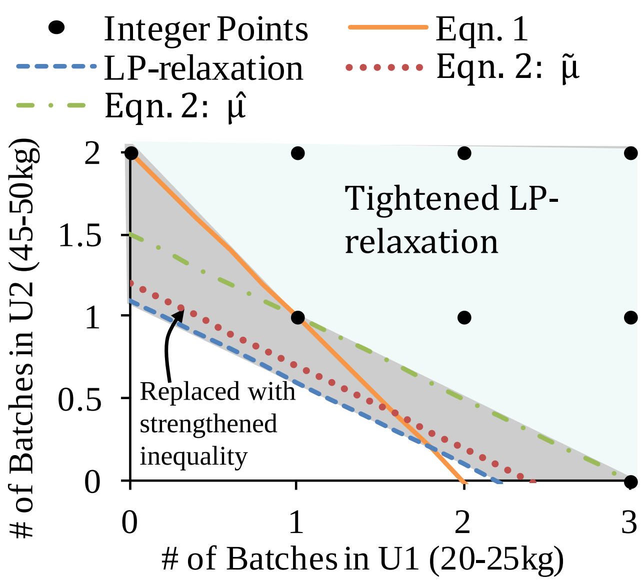
1. Number of batches of task *i*

$$\sum_{j \in \mathbf{J}_i, n} X_{ijn} \geq \left\lceil \mu_i^1 / \max_{j \in \mathbf{J}_i} \{\beta_j^{max}\} \right\rceil \quad \forall i$$

2A. Cumulative production of task *i*

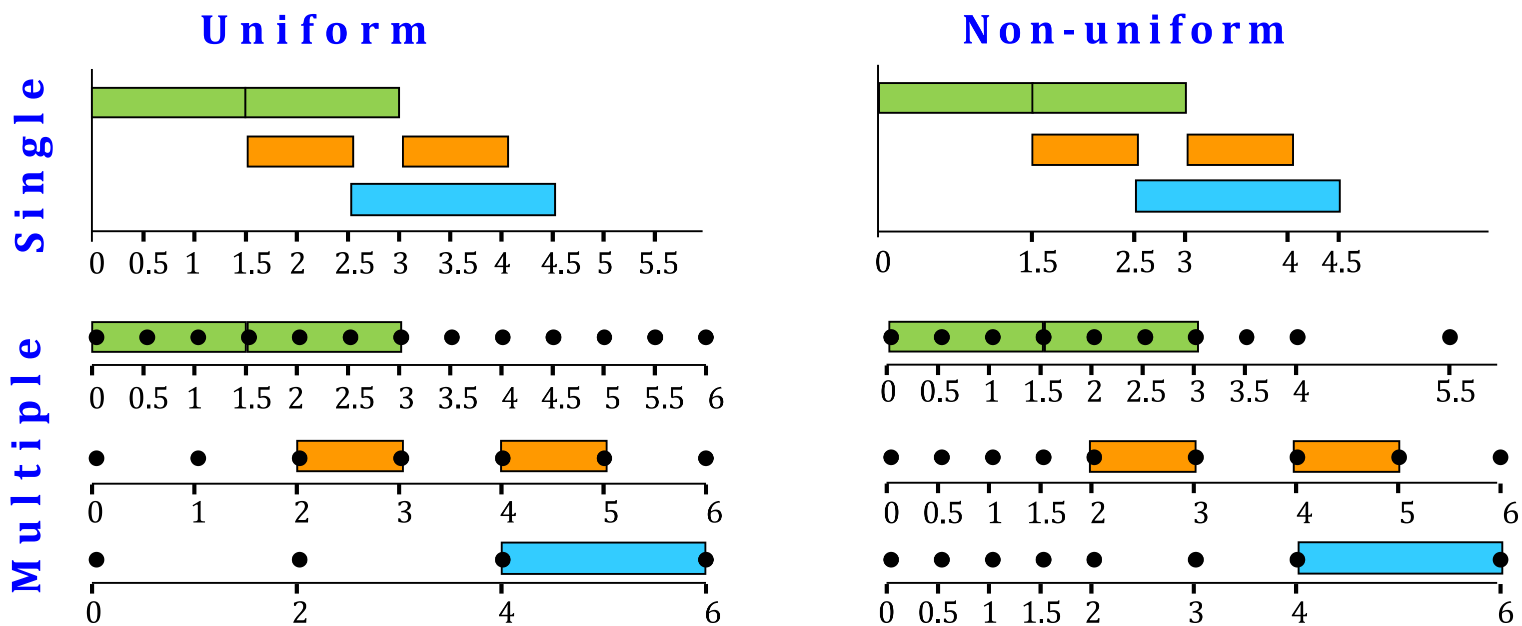
$$\sum_{j \in \mathbf{J}_i, n} \beta_j^{max} X_{ijn} \geq \tilde{\mu}_i \ \forall i$$

1-2 orders of magnitude enhancement



Multiple Non-uniform Discrete Time Grids

- Continuous-time models: common and unit-specific grids
- Discrete-time models thought of as **single and uniform grid**
- Discrete-time models can have **non-uniform and multiple** grids:
  task-, unit-, material-specific grids; and varying accuracy grids



Developed[1,2]:

1) Algorithms to determine specific discrete-time grids from problem data

2) Methods to determine approximate grids based on desired level of accuracy

3) Feasibility and optimality results

[1] Velez & Maravelias, *Computers and Chemical Engineering*, 53, 70-85, 2013

[2] Velez & Maravelias, *Computers and Chemical Engineering*, 72, 233-254, 2015

Computational Challenge

Equivalent schedules

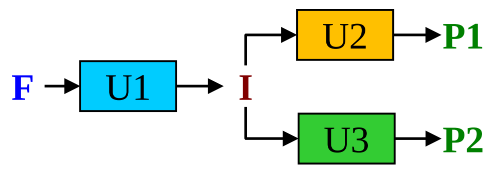
- Formed by shifting tasks earlier or later
- Have the same number of batches
- Have the same objective value

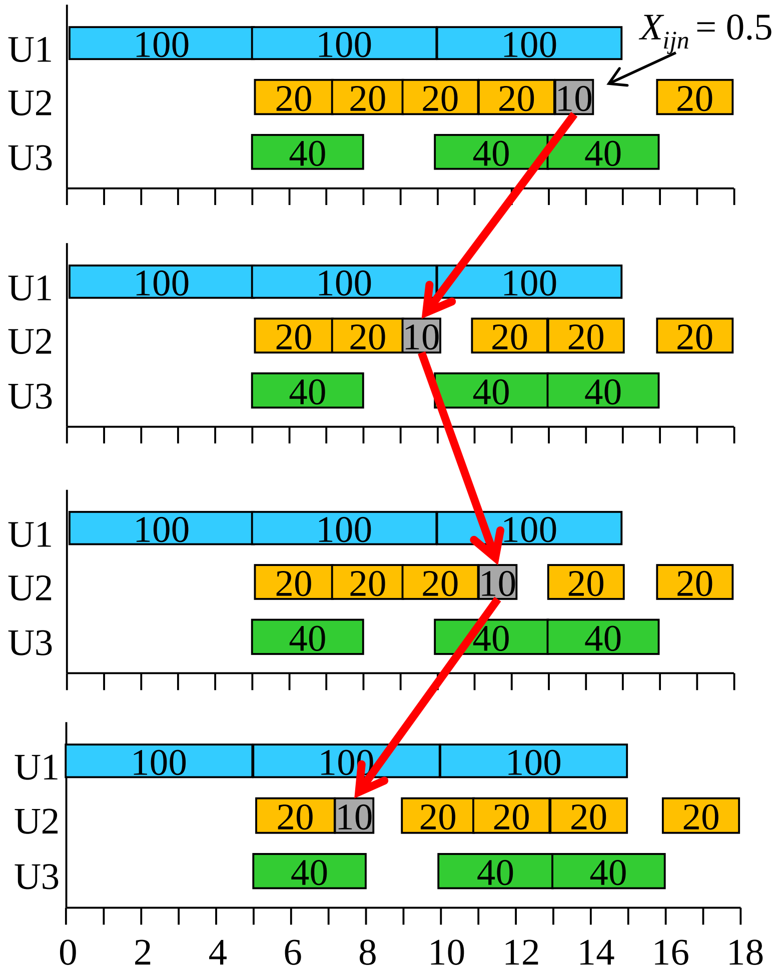
In the B&B algorithm:

- Branching on X_{ijn} leads to equivalent schedules
- There are millions of equivalent fractional solutions
- Bound does not improve

The *number of batches* is a key feature

- Leads to schedules with different objectives





… ~600 more equivalent schedules

Reformulation

Introduce New Variable & Define Equation

- N_{ij} = number of batches of each task i executed on unit j

$$N_{ij} = \sum_n X_{ijn}$$

- Tried various reformulations using SOS1 binaries ($\sum_{b \in \mathbf{B}_{ij}} bZ_{ijb} = \sum_n X_{ijn}$)

Branching Alternatives

- No priorities
- High priority on N_{ij}
- High priorities on X_{ijn}
- Specific N_{ij} priorities calculated from LP-relaxation at each node
- Various priorities with and without strong branching with SOS1 variables

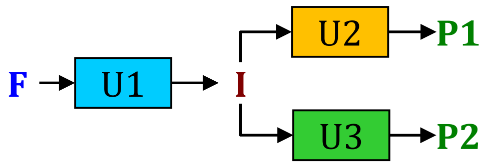
Summary of Results

- Using N_{ij} was faster than Z_{ijb} by a factor of ~2
- Using priorities on X_{ijn} was worse than using no priorities
- For cost/makespan, using or not using priorities on N_{ij} gave similar results
- For profit, using priorities on N_{ij} was faster than no priorities by 12%

- Using priorities on N_{ij} gave the best results
- Results presented for N_{ij} reformulation without priorities

Branching on N_{ij}

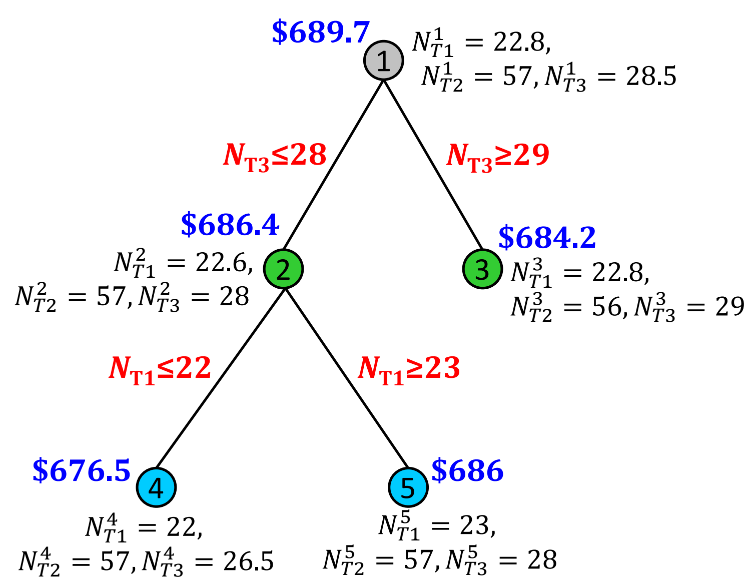
Example: Maximize profit over η = 120 hr (δ = 1 hr)



	T1	T2	T3
Proc. Time (hr)	5	2	3
Cost ($/batch)	1	1	1
Capacity (kg)	100	20	10
Profit ($/kg)	0	0.5	0.2

- LP-Relaxation: $689.7
- Optimal Solution: $686
- Integrality gap: $3.7

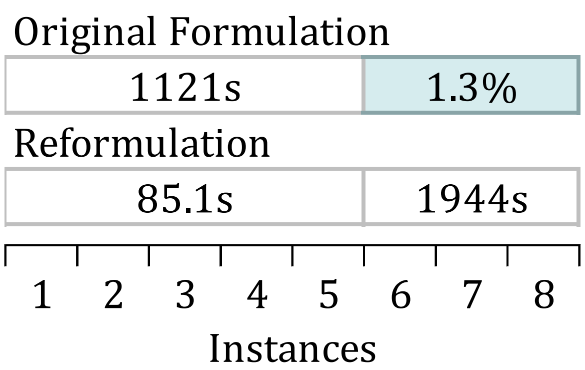
- Original formulation
  - 10 hours, 10 million nodes
  - Bound improves from $689.7 to $689.5
  - Only closes ~5% of integrality gap
- Reformulation
  - Route node: $z^{LP} = 689.7$; $N_{T3} = 28.5$
  - Branching once on N_{T3} improves bound to $686.4 (closes gap by 89%)
  - Branching twice improves bound to $686
  - Closes 100% of integrality gap in 5 nodes
  - CPLEX solves to optimality in <1 second

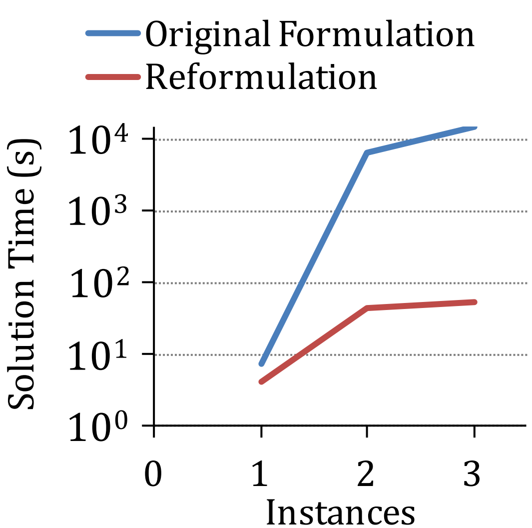


Computational Results – STN

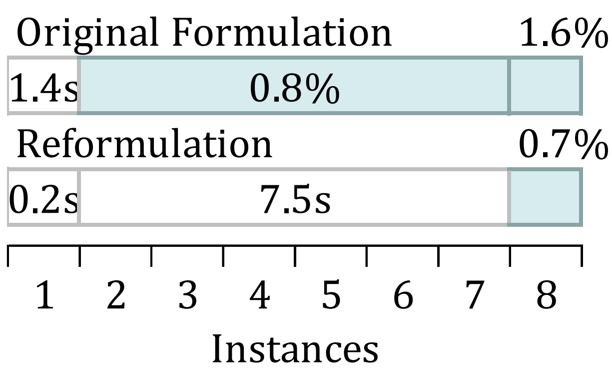
- Testing library: 8 instances; 120-hr horizon ; 1-hr time step
- Settings: 3 hour resource limit ; default CPLEX settings

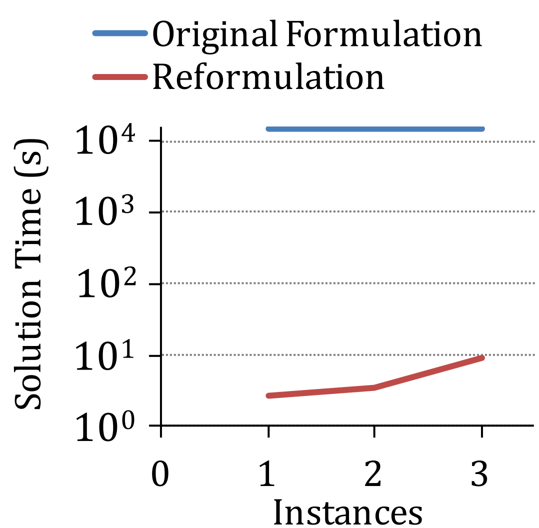
Makespan Minimization



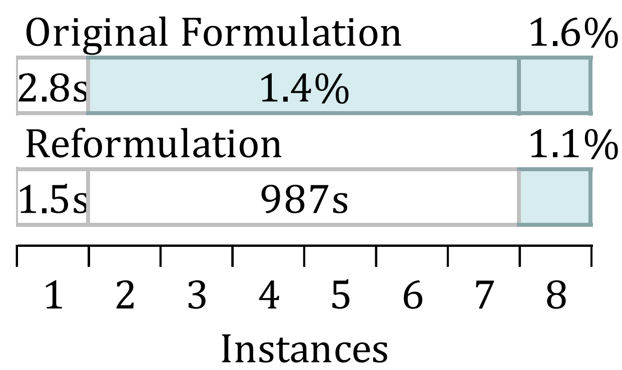


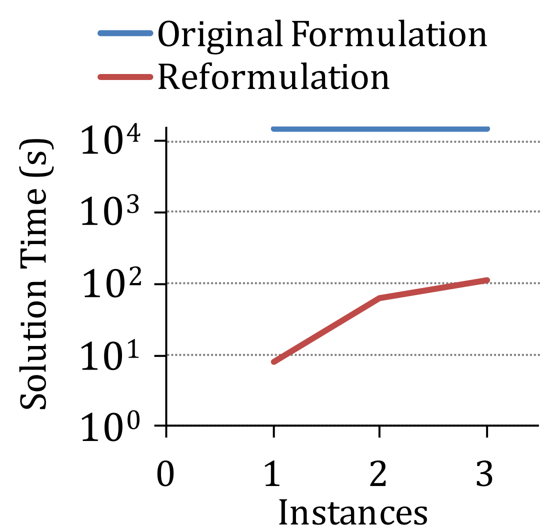
Cost Minimization



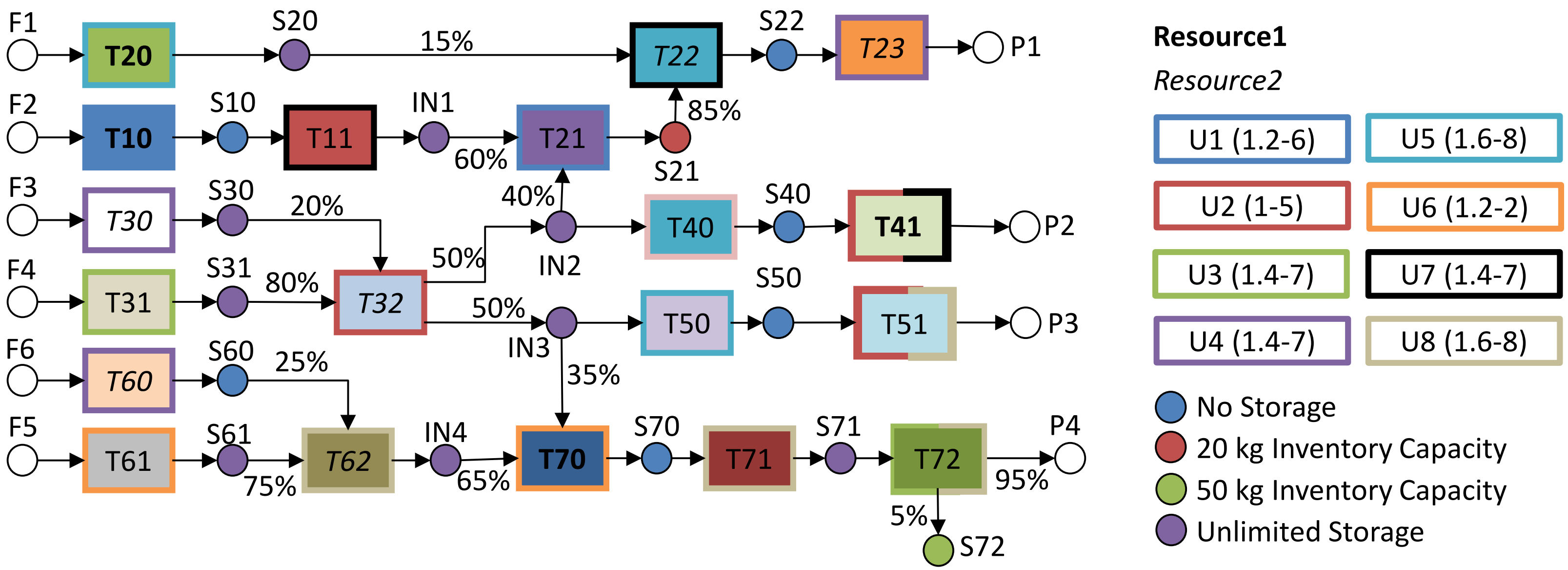


Profit Maximization

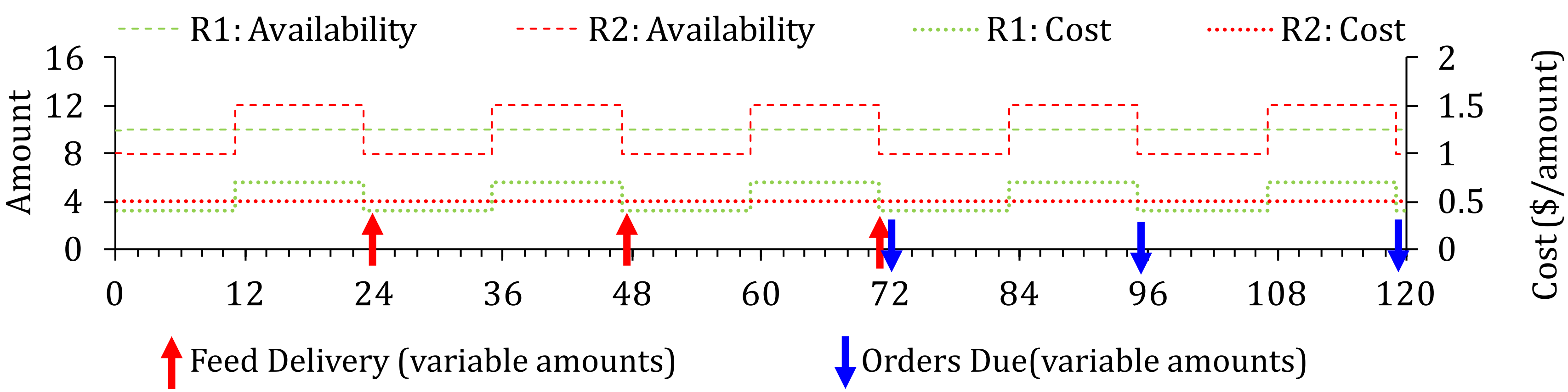




Example



Utility availability & cost, deliveries, and orders:

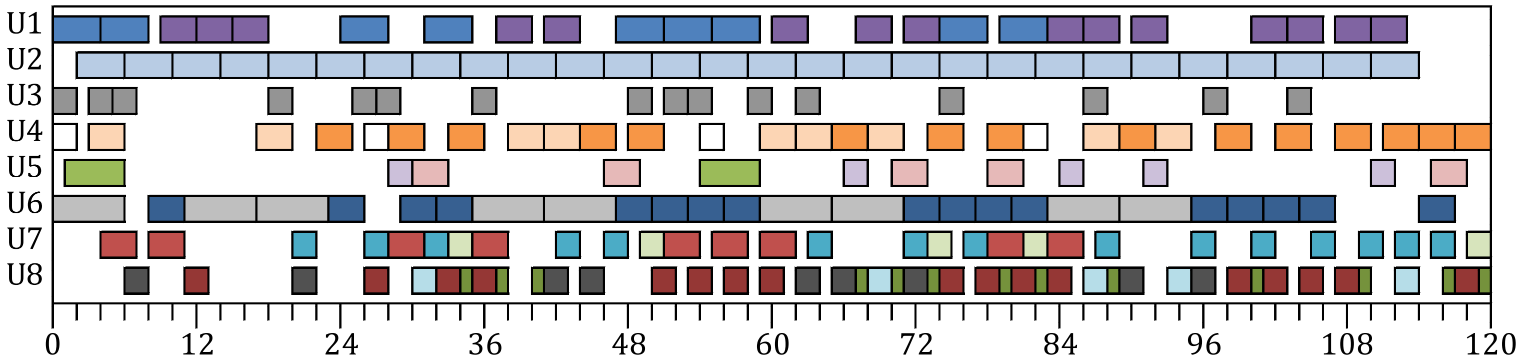


Objective: max *Profit*, while filling customer orders

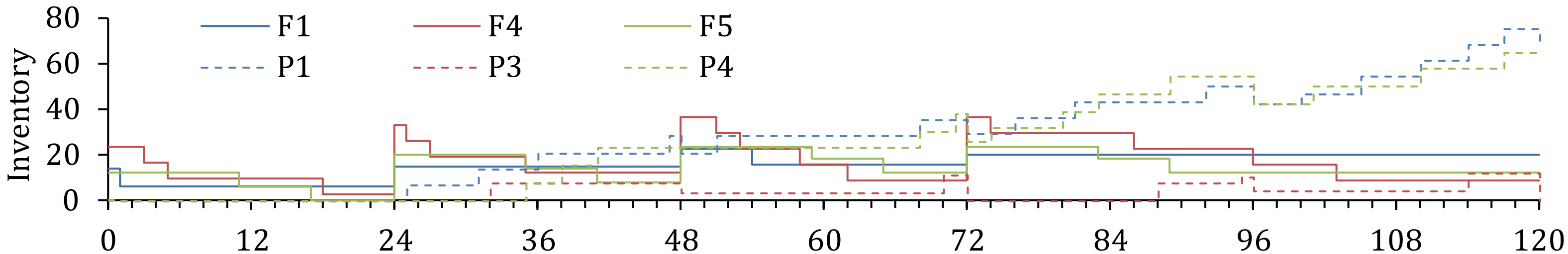
Papageorgiou & Pantelides, *Industrial and Engineering Chemistry Research*, 35, 1996

Example

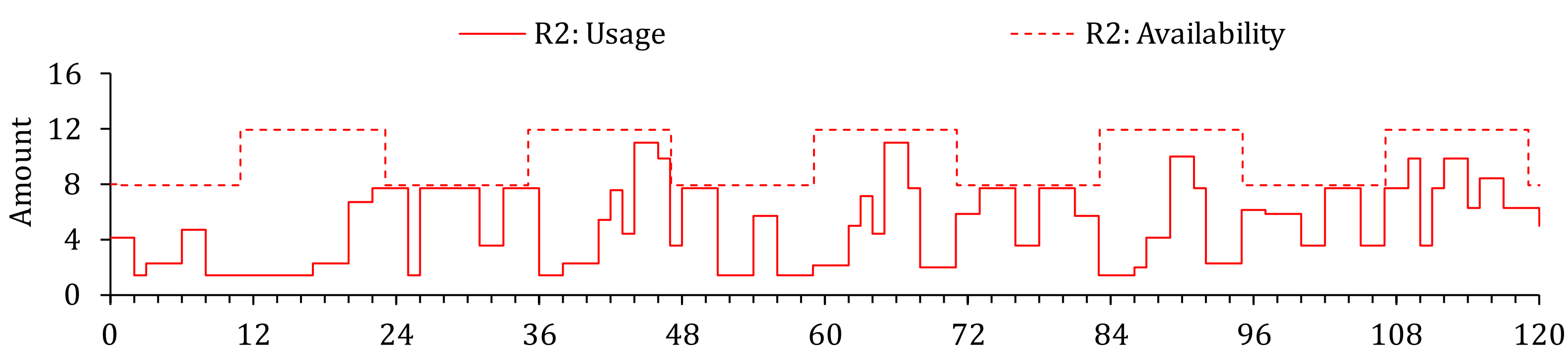
- Original model runs out of memory after 2.5 days (0.6% optimality gap)
- Reformulation is solved in 3.5 minutes



A. Gantt Chart



B. Inventory Profile



C. Resource R2 Profile

Concluding Thoughts

- **New scheduling models**
  - **Sequential environment**
    Simultaneous batching and scheduling with storage and utility constraints
  - **General scheduling model**
    Problems in all/combined production environments
    under wide range of processing characteristics and constraints

- **Solution methods**
  - **Constraint propagation for tightening constraints**
  - **New models with multiple nonuniform discrete grids**
  - **Reformulation and branching methods**
  - **Orders of magnitude reduction in computational requirements**
  - **Can be applied to all time-indexed MIP scheduling formulations**

Research supported by National Science Foundation: CBET-1066206

Happy Birthday!!!