A Novel Global Optimization Approach to the Multiperiod Blending Problem

Scott Kolodziej

Advisor: Dr. Ignacio Grossmann

Department of Chemical Engineering

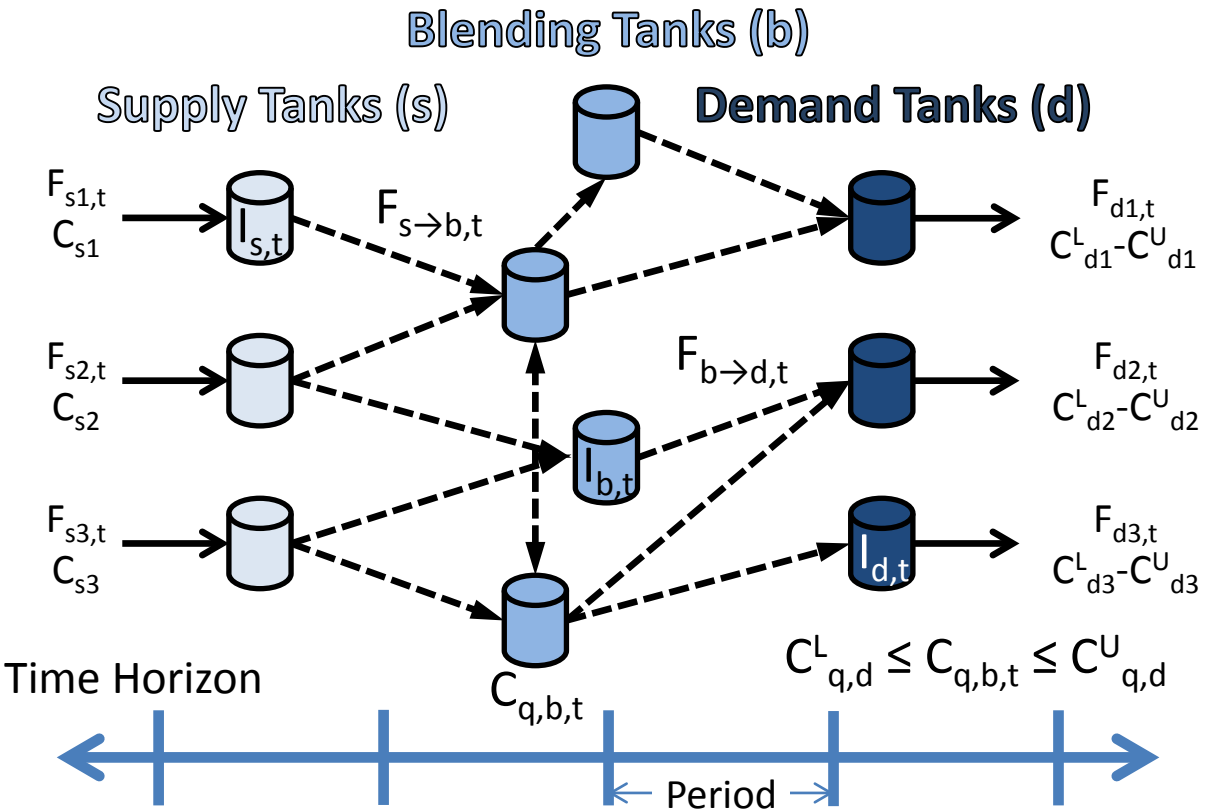
Carnegie Mellon University

ExxonMobil

Problem Definition

Given Information

- **Time Horizon (Periods)**
- **Network Topology**
- **Supply Tanks**
  - Amount entering
  - Concentrations
- **Demand Tanks**
  - Amount withdrawn
  - Concentration limits
- **Initial Conditions**
  - Inventories
  - Concentrations
- **Economic Costs**
  - Network flow costs
  - Raw material costs
  - Profit for meeting demand



Determine

- Flows between which tanks in which time periods
- Inventories and concentrations for all tanks in each time period
- **Maximum total profit** of blending operation

Mathematical Model

Objective: Maximize Profit

- Mass Balances
  - Overall Flows
  - Individual Components (Blending)
- Flow/Inventory Bounds
- Operational Constraints
- Demand Specifications

} Process Constraints

Variables

- Flows, Concentrations, and Inventories **(Continuous)**
- Existence/Nonexistence of Streams **(Binary)**

Complicating **nonconvex bilinearities F·C and I·C** appear in the individual component mass balances

Requires **global optimization** techniques

Resulting model is an **MINLP**

Radix-Based Discretization

Given that $C = \sum_{k=p}^{P} \sum_{j=0}^{9} 10^k \cdot j \cdot z_{j,k}$ the bilinear product $u = F \cdot C$ can be replaced by this set of constraints using exact linearization:

F is **disaggregated** but still **continuous**

$$u = \sum_{k=p}^{P} \sum_{j=0}^{9} 10^k \cdot j \cdot \hat{F}_{j,k}$$

$$\hat{F}_{j,k} \leq F^U \cdot z_{j,k} \qquad \forall\, k \in \{p, \ldots, P\}, j \in \{0, \ldots, 9\}$$

$$\sum_{j=0}^{9} \hat{F}_{j,k} = F \qquad \forall\, k \in \{p, \ldots, P\}$$

$$\sum_{j=0}^{9} z_{j,k} = 1 \qquad \forall\, k \in \{p, \ldots, P\}$$

$$z_{j,k} \in \{0, 1\} \qquad \forall\, k \in \{p, \ldots, P\}, j \in \{0, \ldots, 9\}$$

Teles, Castro, & Matos (2011)

Discretization **replaces bilinearity** with **mixed-integer linear constraints**

The **MINLP** model can be reformulated as an **MILP approximation**

Radix-Based Discretization

Teles, Castro, & Matos (2011)

$$C = \sum_{k=p}^{P} \sum_{j=0}^{9} 10^k \cdot j \cdot z_{j,k}$$

0-1 Binary Variable (one for each digit)

k selects over powers of 10

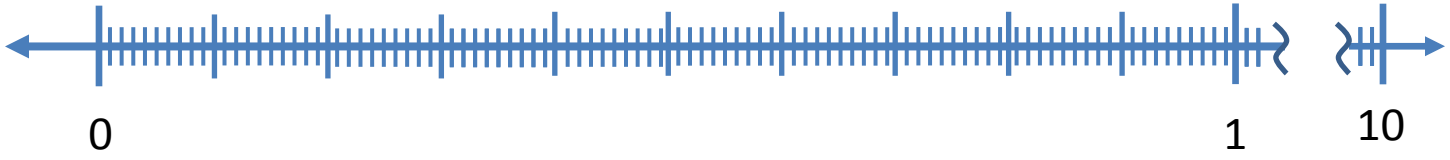
Radix (or base)

j selects over set of digits

$$\sum_{j=0}^{9} z_{j,k} = 1 \quad \forall\, k \in \{p, .., P\}$$

Ensures only one digit for each decimal place

C Axis

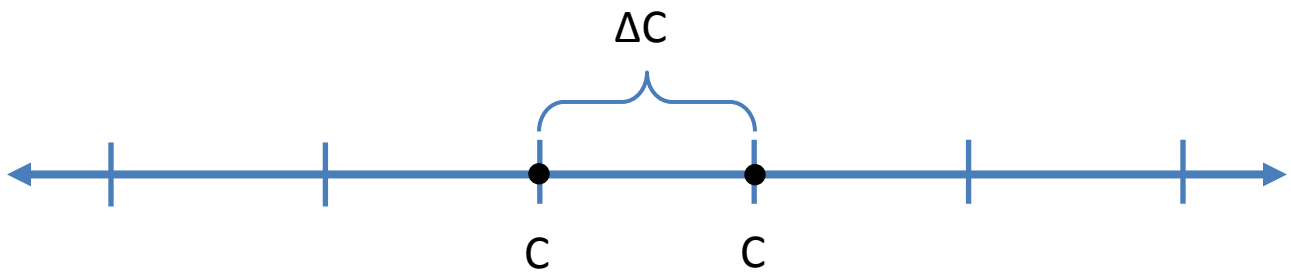


Increment	1	0.1	0.01
Range	0-9	0-9.9	0-9.99
Significant Digits	1	2	3
Binary Variables	10	20	30

Linear increase in binary variables for **order of magnitude** increase in precision

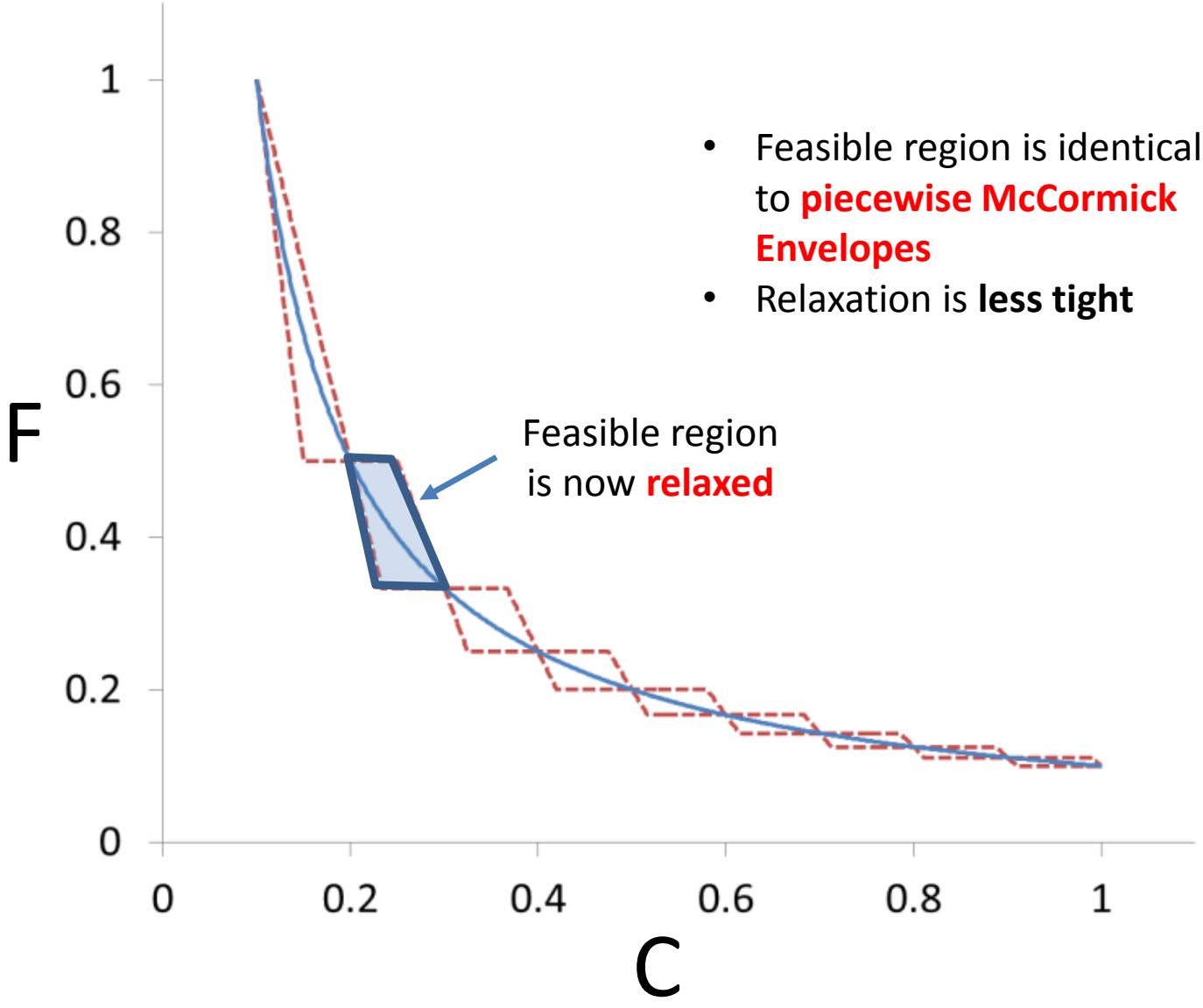
Lower Bounding

- RBD yields an **upper bound**
- What about a **lower bound**?



- We introduce ΔC as a **slack variable**
  - ΔC ranges between 0 and 10^p (the size of the "gaps")
  - **Relaxed RBD constraints** can be added to "fill the gap" and **relax the original problem**

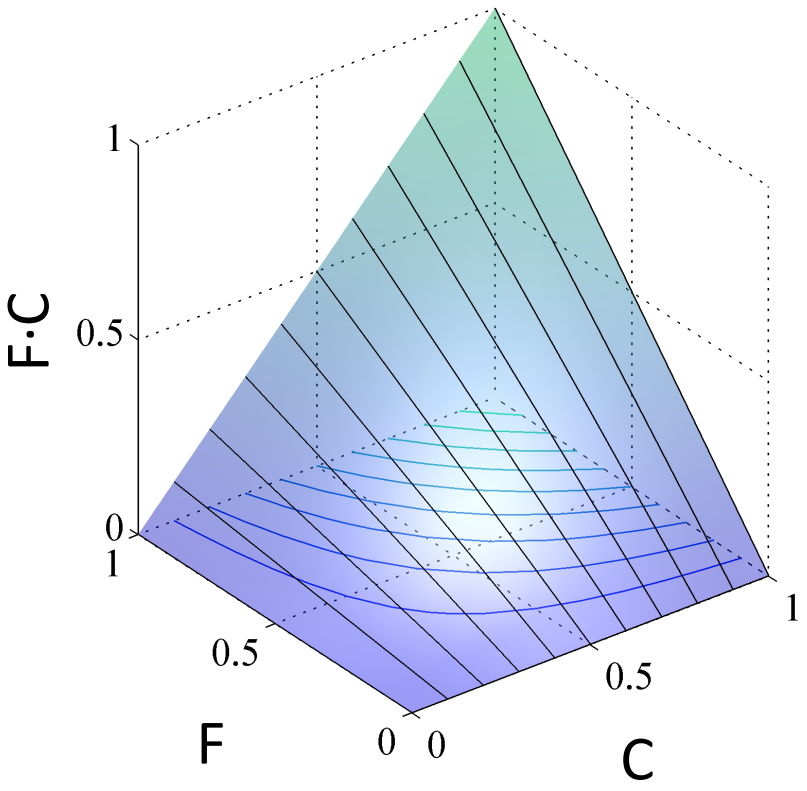
Lower Bounding



- Feasible region is identical to **piecewise McCormick Envelopes**
- Relaxation is **less tight**

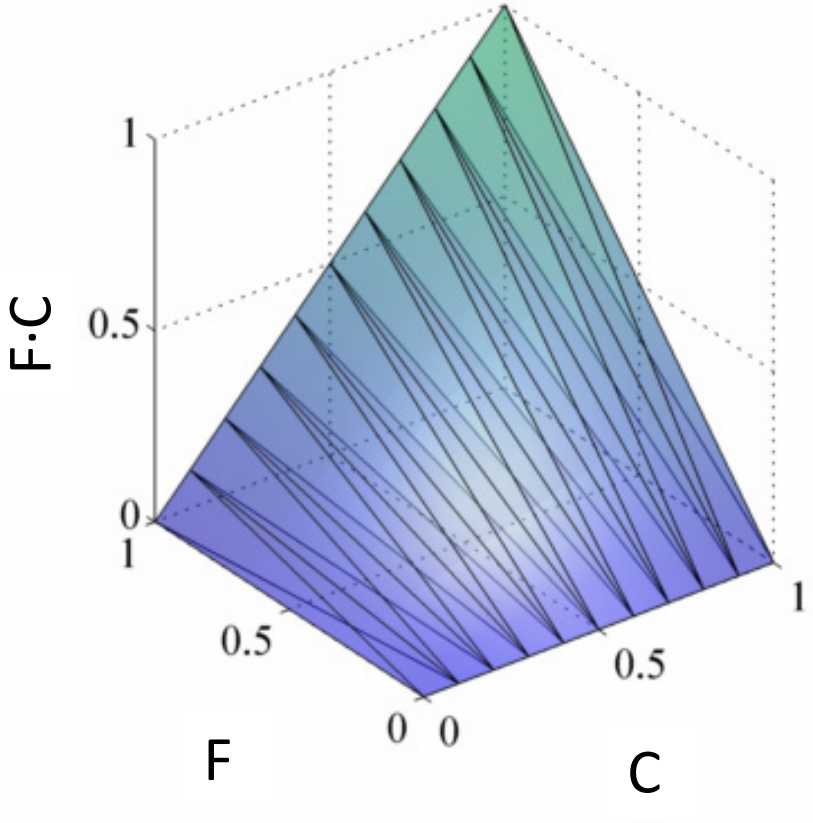
Upper and Lower Bounding

Upper Bounding



Underestimates the feasible region

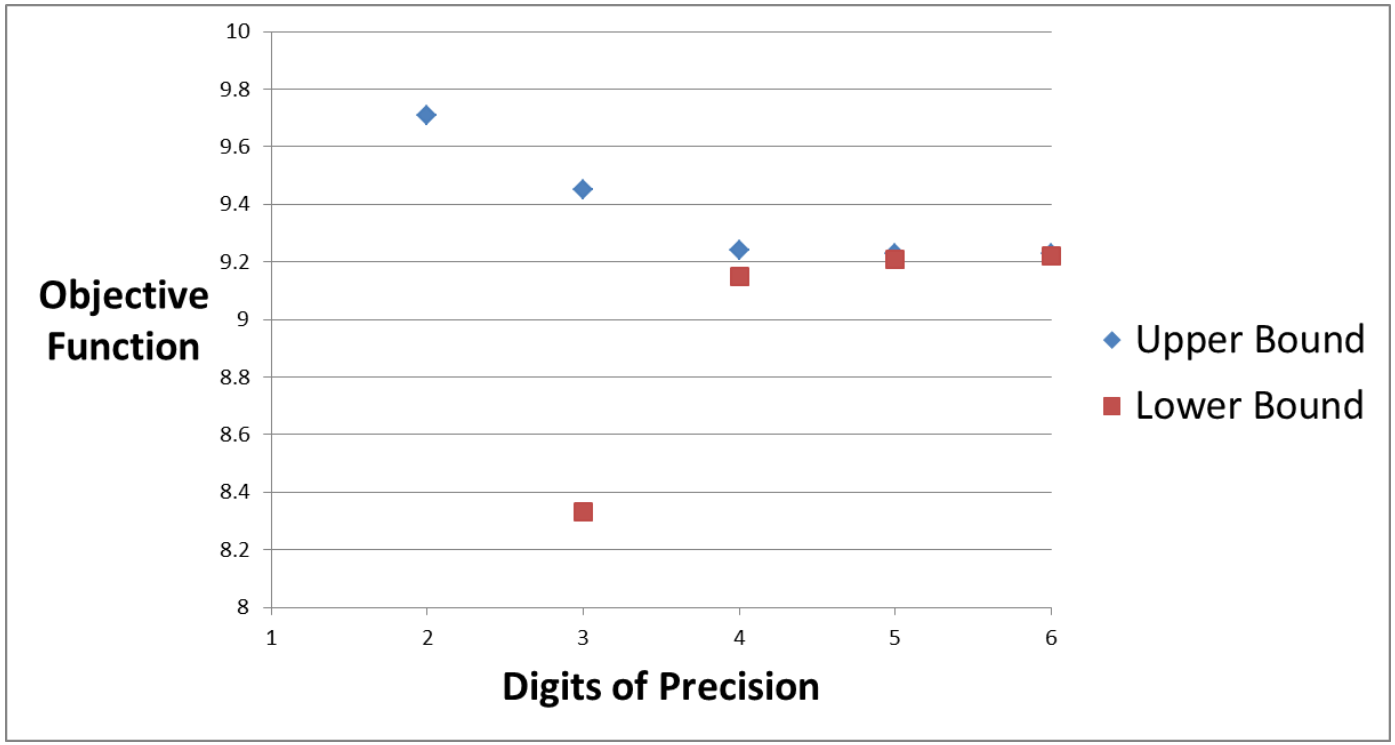
Lower Bounding (Relaxation)



Overestimates the feasible region

Global Optimization Algorithm

- As **precision increases**, **UB → LB**
- We can start with **low precision** and slowly **increase precision** until **gap < ε**
- Currently **work in progress**



Global Optimization Algorithm

- Preliminary Computational Results

Tanks	6	8	8	8	8	8	8
Time Periods	3	3	3	3	4	4	4
Global Optimization Wall Time (s)	2.1	4753	2588	426	338	>7200	1525
MINLP Wall Time (s)	21.2	>7200	>7200	>7200	>7200	>7200	>7200
Upper Bound	13.3594	45.2843	7.3936	13.5268	53.9627	9.2266	20.0363
Lower Bound	13.3594	45.2608	7.3818	13.5268	53.9257	9.2051	19.9686
Relative Gap	0	5.19×10^{-4}	1.60×10^{-3}	0	6.86×10^{-4}	2.34×10^{-3}	3.39×10^{-3}
Iterations	2	4	4	4	4	4	4

Generally **much faster** than **MINLP**

Conclusions and Future Work

Conclusions

- MINLP approach takes **too long**
- Effective **radix-based MILP approach** reviewed and expanded
  - **Order of magnitude reductions** in CPU time observed
- **Upper** and **lower bounds** were introduced using **radix-based discretization**

Future Work

- Utilize **upper/lower bounding** formulations to **globally optimize**
- Large scale instances may require **decomposition techniques**
  - Utilize **Lagrangean decomposition** to decompose larger problems into easier to solve **subproblems**