New Paradigms for Process Optimization Modeling and Solution Strategies

L. T. Biegler
Chemical Engineering Dept.
Carnegie Mellon University

June, 2017

Overview

- **Introduction and Motivation**
  - Why Optimize?
- **Fundamentals of Nonlinear Programming**
  - Problem formulation and KKT conditions
  - NLP algorithms and optimization models
- **Large-scale Optimization Case Study**
  - Water Contaminant Source Detection
  - High Performance Computing
- **Multi-level Optimization**
  - MPCC Formulations
  - ASU Case Study
- **Conclusions and Future Steps**

Why Process Optimization?

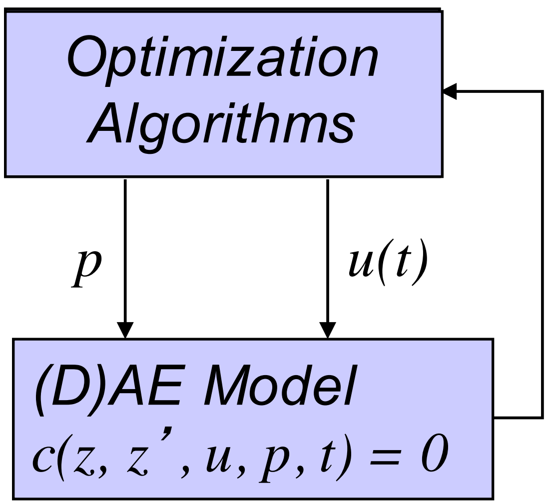
is

r

.,

Why Process Optimization?

- Equipment and Flowsheet Design
- Process Operations, Transients and Upsets
- Parameter Estimation and Model Discrimination



Optimization: find the best solution to this process within constraints.

Objective Function: quantitative indicator of good solution, e.g., cost, yield, profit…, or multiple objectives!

Decision Variables: variables that influence process behavior and can be adjusted

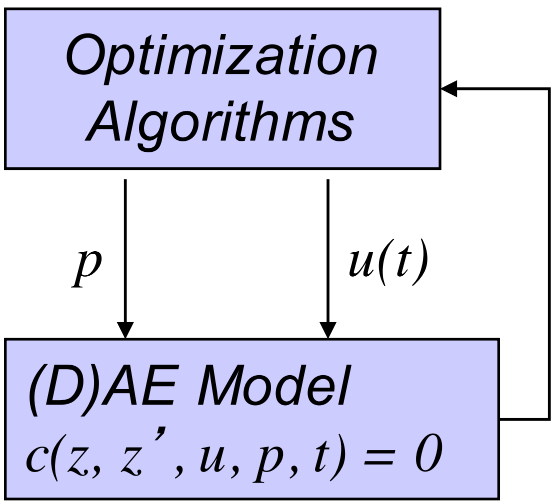
Often done by trial and error (through case study).

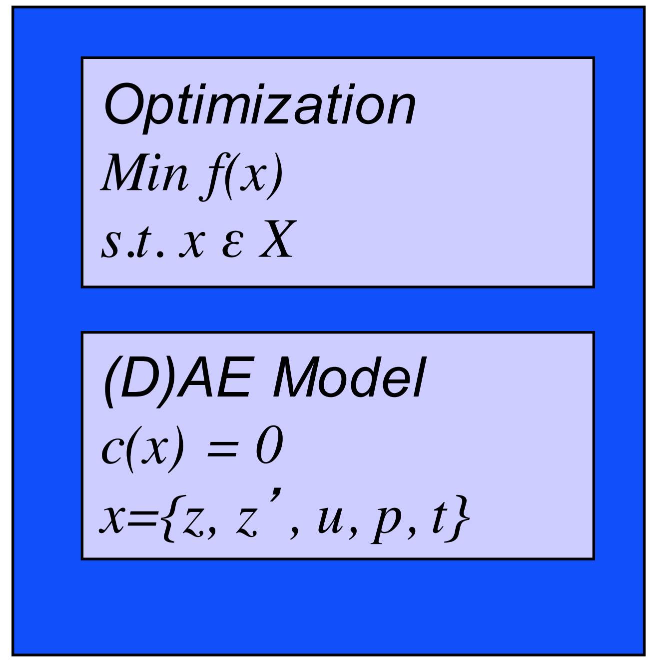
Systematic approach to this task?
- make this task as efficient as possible.

- Optimization Gives Better Results than with “Experience”
- Consistent Results among all Practitioners

Why Process Optimization?

- Equipment and Flowsheet Design
- Process Operations, Transients and Upsets
- Parameter Estimation and Model Discrimination





- Optimization Gives Better Results than with “Experience”
- Consistent Results among all Practitioners
- Reduce Solution Time by Orders of Magnitude
- Support and Enhance Process Understanding

Constrained Optimization Problems With Smooth Functions (Nonlinear Programming)

Problem: $\quad Min_x \ f(x)$

$s.t. \quad g(x) \leq 0$

$h(x) = 0$

where:

- $f(x)$ - scalar objective function
- x - n vector of variables
- $g(x)$ - inequality constraints, m vector
- $h(x)$ - meq equality constraints.

Sufficient Condition for Global Optimum

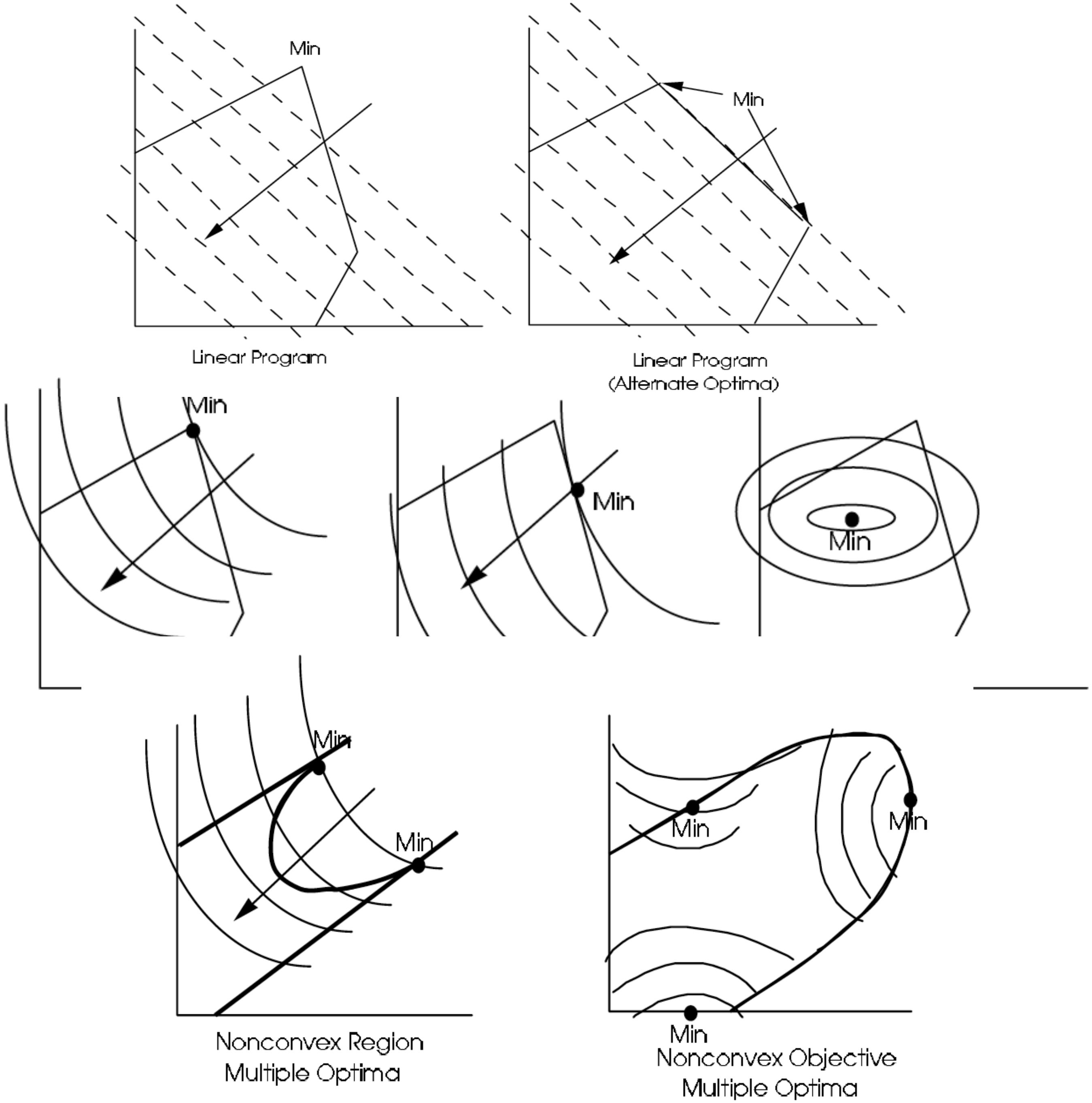
- $f(x)$ must be *convex*, and
- feasible region must be convex,

  i.e. $g(x)$ are all *convex*

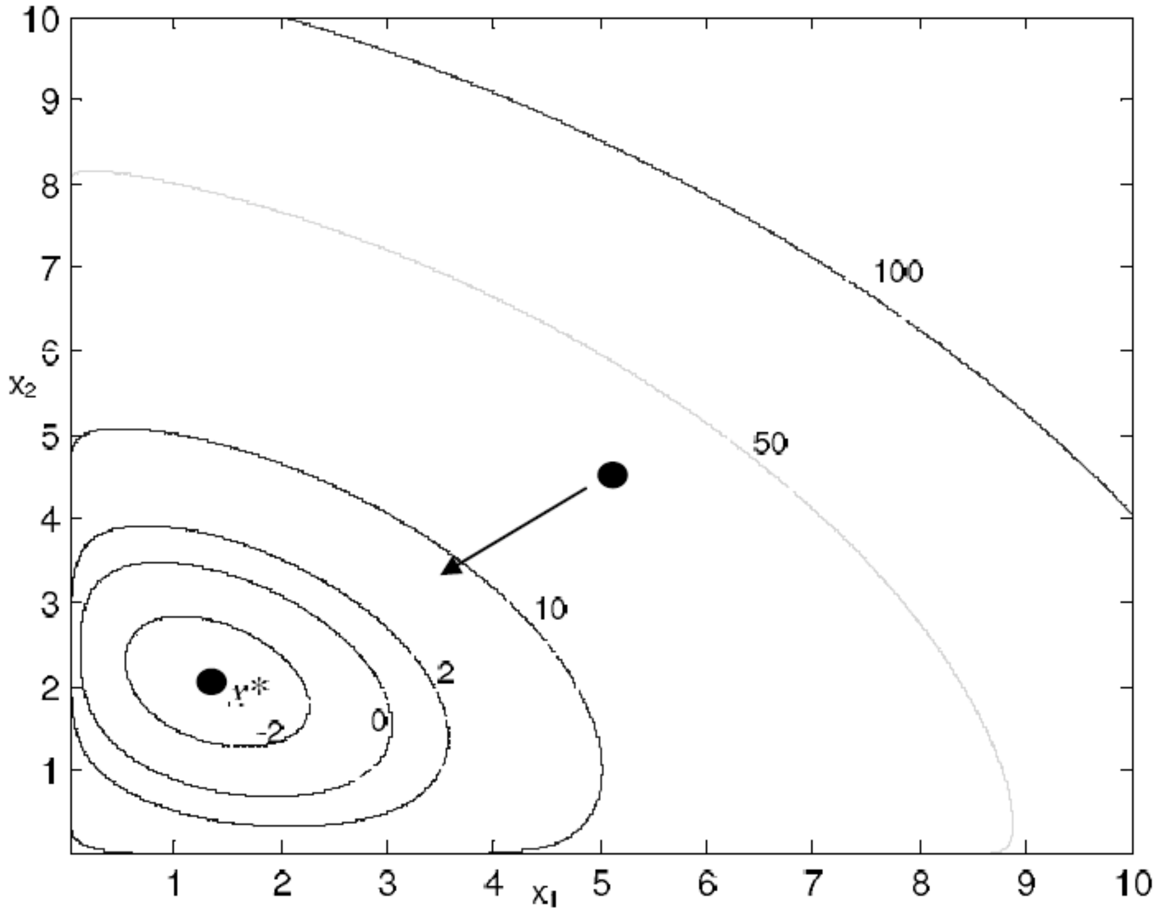
  $h(x)$ are all *linear*

Except in special cases, there is no guarantee that a local optimum is global if sufficient conditions are violated.

Characterization of Constrained Optima

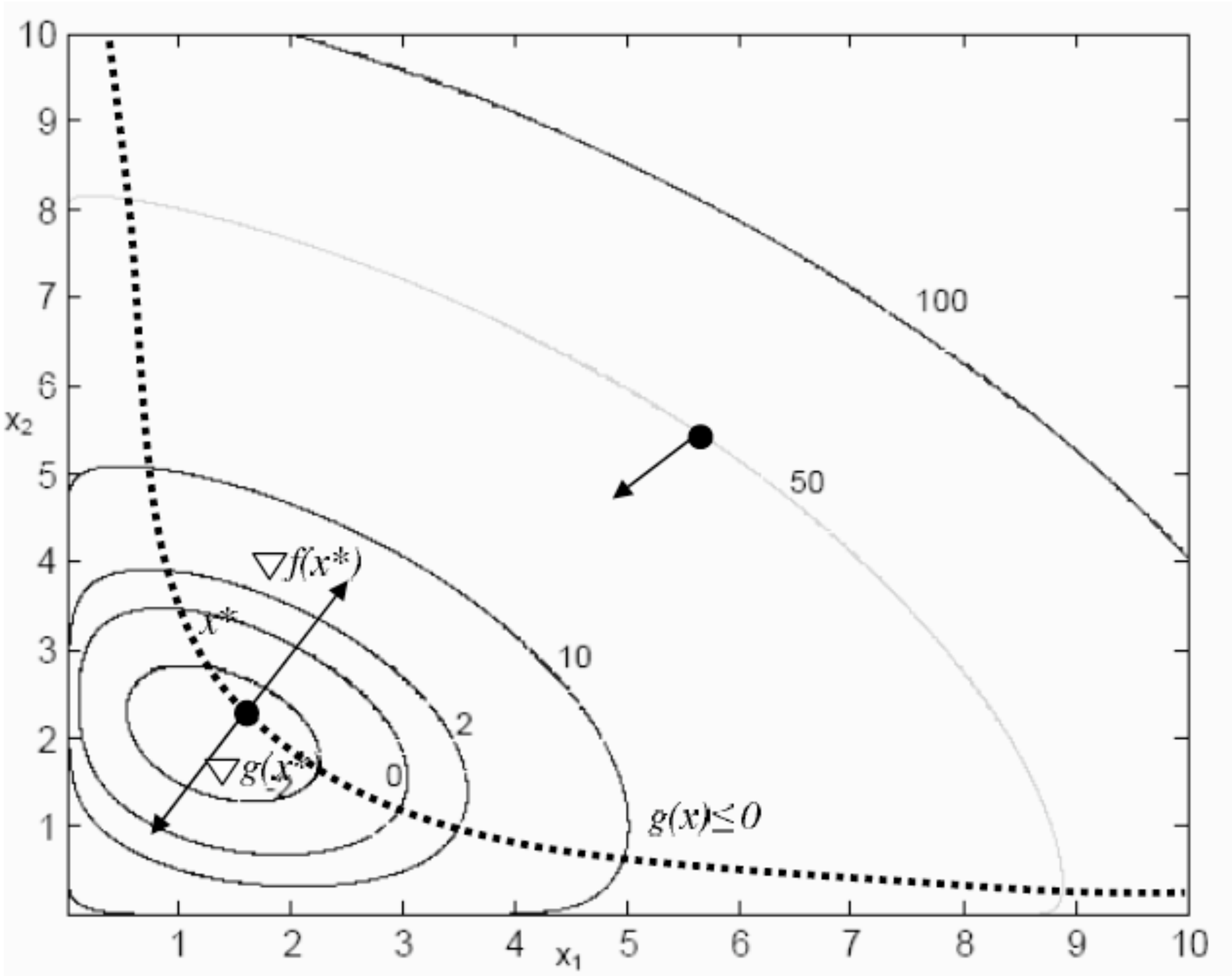


What conditions characterize a (locally) optimal solution?



Unconstrained Local Minimum Necessary Conditions	Unconstrained Local Minimum Sufficient Conditions
$\nabla f(x^*) = 0$	$\nabla f(x^*) = 0$
$p^T \nabla^2 f(x^*)\, p \geq 0 \quad \text{for } p \in \Re^n$	$p^T \nabla^2 f(x^*)\, p > 0 \quad \text{for } p \in \Re^n$
(positive semi-definite)	(positive definite)

Optimal solution for inequality constrained problem

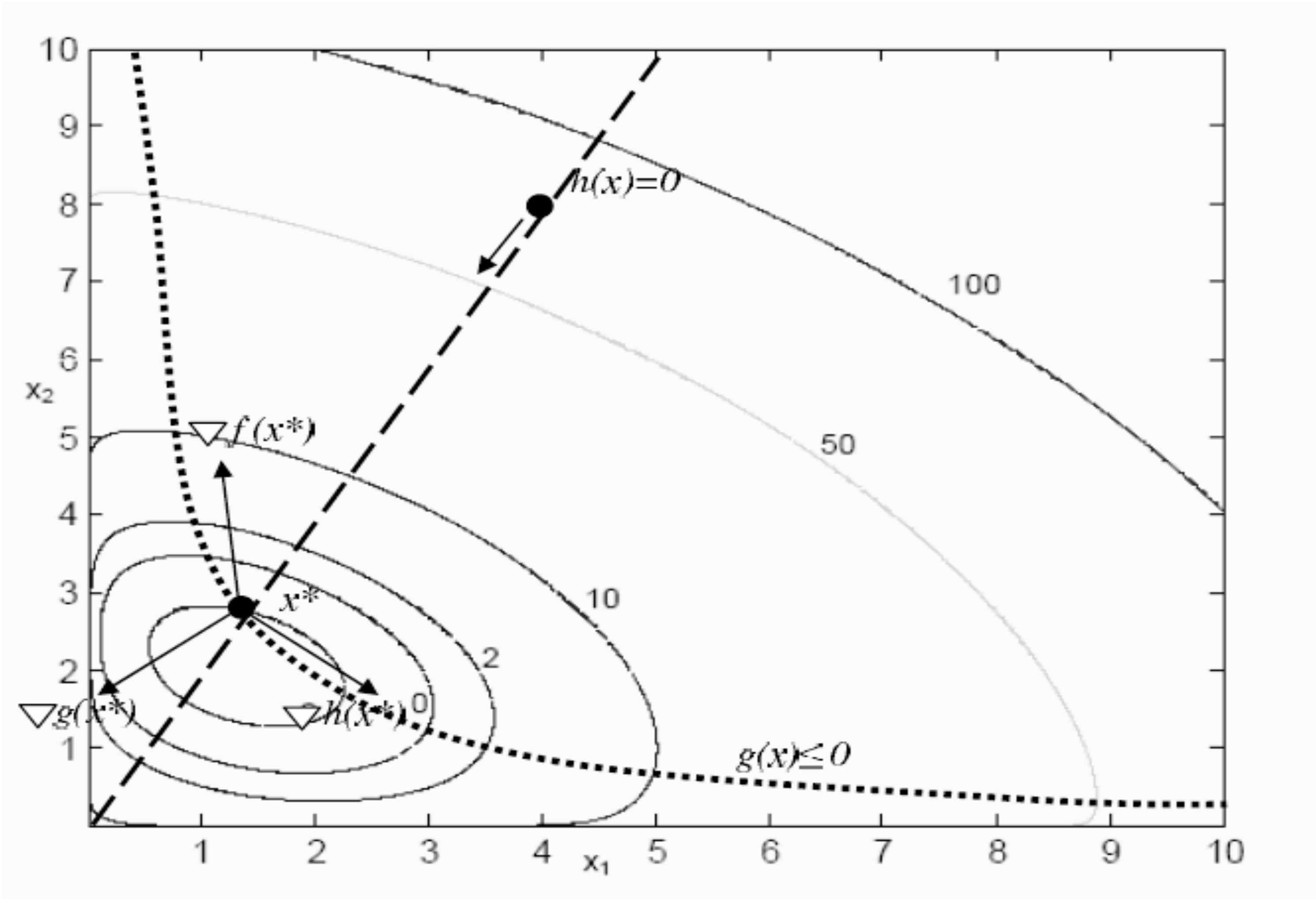


$$
\begin{array}{ll}
\textbf{Min} & f(x) \\
\text{s.t.} & g(x) \leq 0
\end{array}
$$

<u>Analogy</u>: Ball rolling down valley pinned by fence

<u>Note</u>: Balance of forces (∇f, ∇g_1)

Optimal solution for general constrained problem



Problem: Min $f(x)$

s.t. $g(x) \le 0$

$h(x) = 0$

<u>Analogy</u>: Ball rolling on rail pinned by fences

Balance of forces: $\nabla f, \nabla g_1, \nabla h$

Optimality conditions for local optimum

Necessary First Order Karush Kuhn - Tucker Conditions

$\nabla L(x^*, u, v) = \nabla f(x^*) + \nabla g(x^*)\, u + \nabla h(x^*)\, v = 0$
(Balance of Forces)
$u \geq 0$ (Inequalities act in only one direction)
$g(x^*) \leq 0,\ \ h(x^*) = 0$ (Feasibility)
$u_j\, g_j(x^*) = 0$ (Complementarity: either $g_j(x^*) = 0$ or $u_j = 0$)
u, v are "weights" for "forces," known as KKT multipliers, shadow prices, dual variables

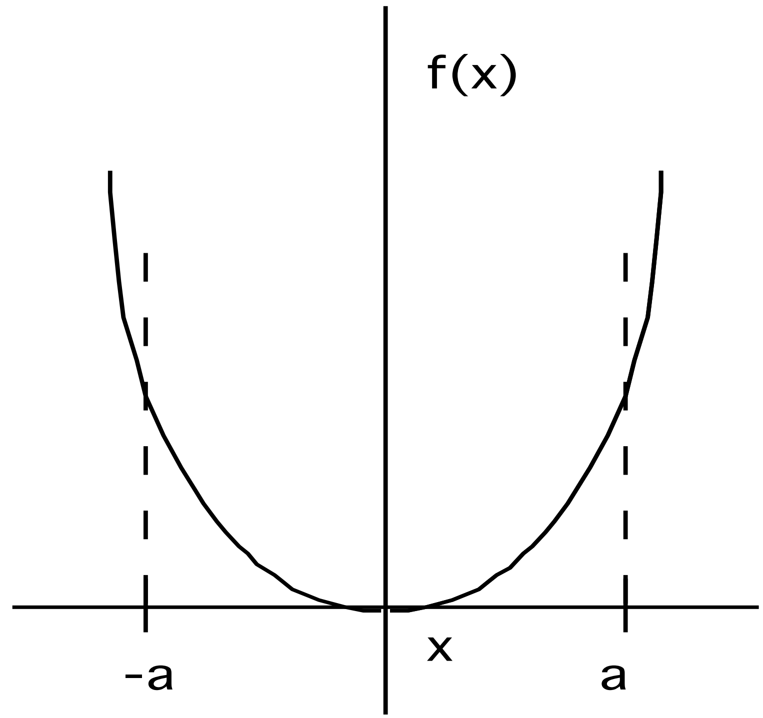
"To guarantee that a local NLP solution satisfies KKT conditions, a constraint qualification is required. E.g., the *Linear Independence Constraint Qualification* (LICQ) requires active constraint gradients, $[\nabla g_A(x^*)\ \ \nabla h(x^*)]$, to be linearly independent. Also, under LICQ, KKT multipliers are uniquely determined."

Necessary (Sufficient) Second Order Conditions

- Positive curvature in "constraint" directions.
- $p^T \nabla^2 L(x^*)\, p \geq 0$ $\ (p^T \nabla^2 L(x^*)\, p > 0)$
  where p are the constrained directions: $\nabla g_A(x^*)^T p = 0,\ \nabla h(x^*)^T p = 0$

Single Variable Example of KKT Conditions

Min $(x)^2$ s.t. $-a \le x \le a, a > 0$
$x^* = 0$ is seen by inspection



Lagrange function :
$L(x, u) = x^2 + u_1(x-a) + u_2(-a-x)$

First Order KKT conditions:
$\nabla L(x, u) = 2\,x + u_1 - u_2 = 0$
$u_1\,(x-a) = 0$
$u_2\,(-a-x) = 0$
$-a \le x \le a$ $\qquad u_1, u_2 \ge 0$

Consider three cases:

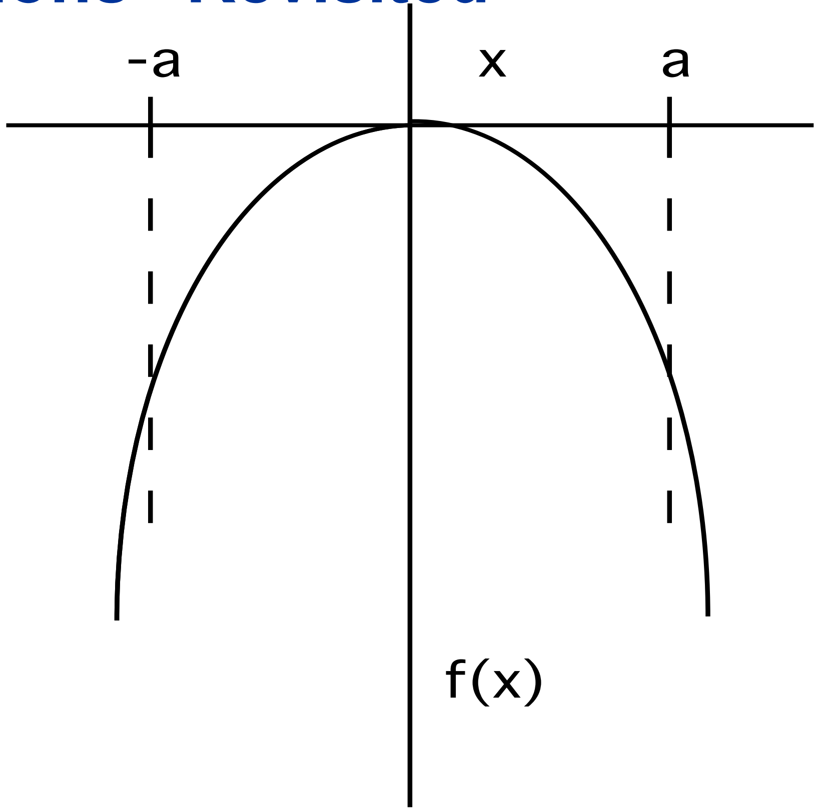
- $u_1 \ge 0,\ u_2 = 0$ — Upper bound is active, $x = a, u_1 = -2a, u_2 = 0$
- $u_1 = 0,\ u_2 \ge 0$ — Lower bound is active, $x = -a, u_2 = -2a, u_1 = 0$
- $u_1 = u_2 = 0$ — Neither bound is active, $u_1 = 0, u_2 = 0,\ x = 0$

Second order conditions $(x^*, u_1, u_2 = 0)$

$$\nabla_{xx} L\,(x^*, u^*) = 2$$
$$p^T \nabla_{xx} L\,(x^*, u^*)\,p = 2\,(\Delta x)^2 > 0$$

Single Variable Example of KKT Conditions - Revisited

Min $-(x)^2$ s.t. $-a \le x \le a, a > 0$
$x^* = \pm a$ is seen by inspection



Lagrange function :
$L(x, u) = -x^2 + u_1(x-a) + u_2(-a-x)$

First Order KKT conditions:
$\nabla L(x, u) = -2x + u_1 - u_2 = 0$
$u_1 (x-a) = 0$
$u_2 (-a-x) = 0$
$-a \le x \le a \qquad u_1, u_2 \ge 0$

Consider three cases:

- $u_1 \ge 0,\ u_2 = 0$ Upper bound is active, $x = a, u_1 = 2a, u_2 = 0$
- $u_1 = 0,\ u_2 \ge 0$ Lower bound is active, $x = -a, u_2 = 2a, u_1 = 0$
- $u_1 = u_2 = 0$ Neither bound is active, $u_1 = 0, u_2 = 0,\ x = 0$

Second order conditions $(x^*, u_1, u_2 = 0)$

$$\nabla_{xx} L(x^*, u^*) = -2$$
$$p^T \nabla_{xx} L(x^*, u^*)\, p = -2(\Delta x)^2 < 0$$

Interpretation of Second Order Conditions

For $x = a$ or $x = -a$, we require the allowable direction to satisfy the active constraints exactly. Here, any point along the allowable direction, x^* must remain at its bound.

For this problem, however, there are no nonzero allowable directions that satisfy this condition. Consequently the solution x^* is defined entirely by the active constraint. The condition:

$$p^T \nabla_{xx} L(x^*, u^*, v^*)\, p > 0$$

for the allowable directions, is *vacuously* satisfied - because there are *no* allowable directions that satisfy $\nabla g_A(x^*)^T p = 0$. Hence, *sufficient* second order conditions are satisfied.

As we will see, sufficient second order conditions are satisfied by linear programs as well.

Nonlinear Programming Strategies

Problem: $Min_x \ f(x)$

s.t. $g(x) \leq 0$

$h(x) = 0$

KKT Conditions:

Stationarity $\nabla f(x) + \nabla g(x)u + \nabla h(x)v = 0$

Feasibility $h(x) = 0, g(x) \leq 0$

Complementarity $0 \leq u \perp g(x) \leq 0$

Almost like solving nonlinear equations
Except for Complementarity

Active Set Methods
(SQP, SLP, GRG)

Penalty/Barrier Methods
(IPOPT, KNITRO, LOQO)

Successive Quadratic Programming (SQP)

Motivation:

- Take KKT conditions, expand in Taylor series about current point.
- Take Newton step (QP) to determine next point.

Derivation – KKT Conditions

$$\nabla_x L\,(x^*, u^*, v^*) = \nabla f(x^*) + \nabla g_A(x^*)\, u^* + \nabla h(x^*)\, v^* = 0$$

$$h(x^*) = 0$$

$g_A(x^*) = 0,$ where g_A are the active constraints.

Newton - Step

$$\begin{bmatrix} \nabla_{xx} L & \nabla g_A & \nabla h \\ \nabla g_A^T & 0 & 0 \\ \nabla h^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta u \\ \Delta v \end{bmatrix} = - \begin{bmatrix} \nabla_x L\left(x^k, u^k, v^k\right) \\ g_A\left(x^k\right) \\ h\left(x^k\right) \end{bmatrix}$$

Requirements:

- $\nabla_{xx} L$ must be calculated and should be bounded
- need to find correct active set g_A
- need to choose good (and bounded) estimates of u^k, v^k

Basic SQP Algorithm

0. Guess x^0, Set $B^0 = I$ (Identity). Evaluate $f(x^0)$, $g(x^0)$ and $h(x^0)$.
1. At x^k, evaluate $\nabla f(x^k)$, $\nabla g(x^k)$, $\nabla h(x^k)$.
2. If k > 0, update B^k using the BFGS Formula.
3. Solve:

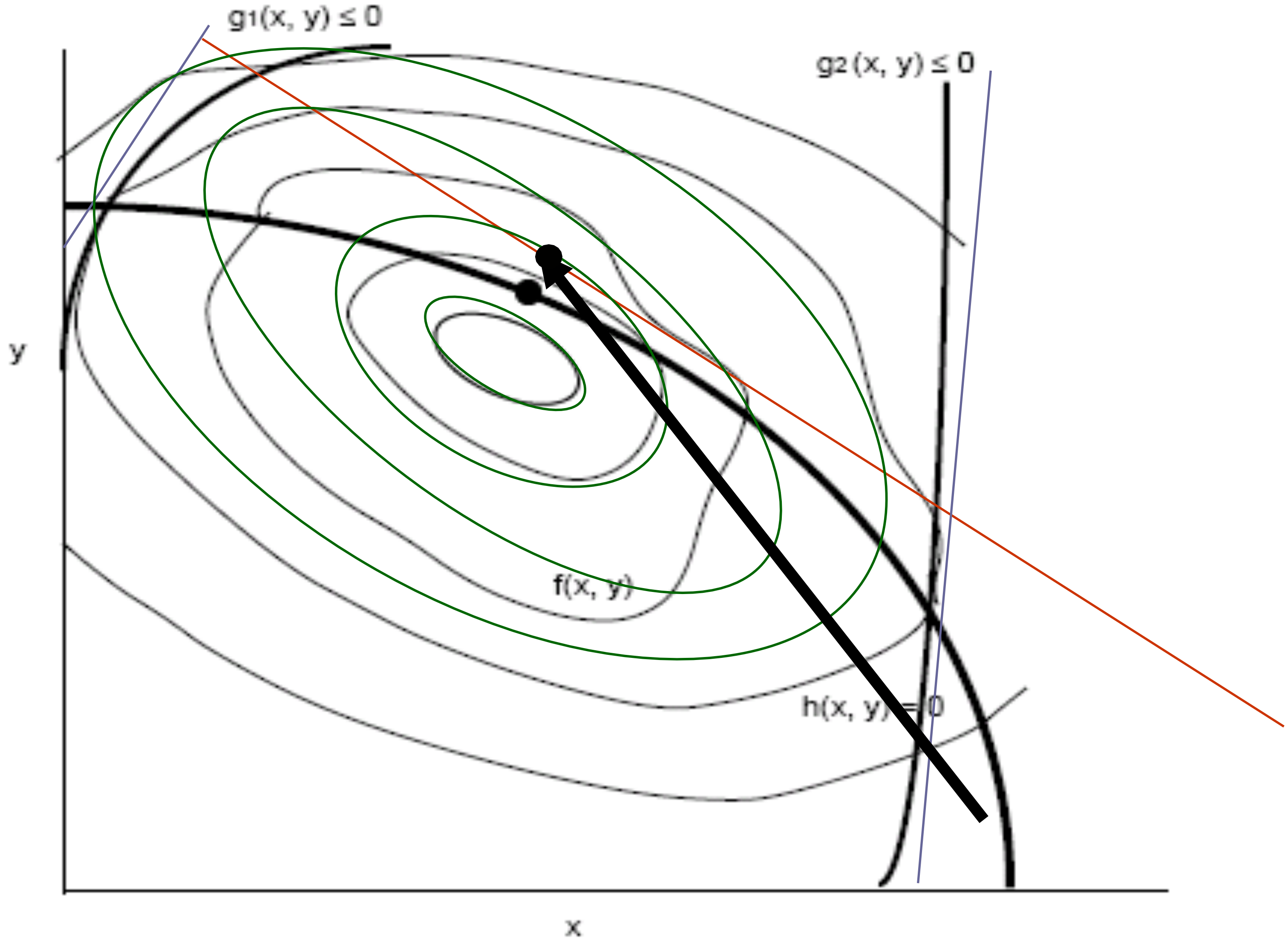
$$\begin{aligned} &Min_d \;\; \nabla f(x^k)^T d + 1/2\, d^T B^k d \\ s.t.\;\; &g(x^k) + \nabla g(x^k)^T d \le 0 \\ &h(x^k) + \nabla h(x^k)^T d = 0 \end{aligned}$$

   If KKT error less than tolerance: $\|\nabla L(x^*)\| \le \varepsilon$, $\|h(x^*)\| \le \varepsilon$, $\|g(x^*)_+\| \le \varepsilon$. STOP, else go to 4.

4. Find α so that $0 < \alpha \le 1$ and $\psi(x^k + \alpha d) < \psi(x^k)$ sufficiently

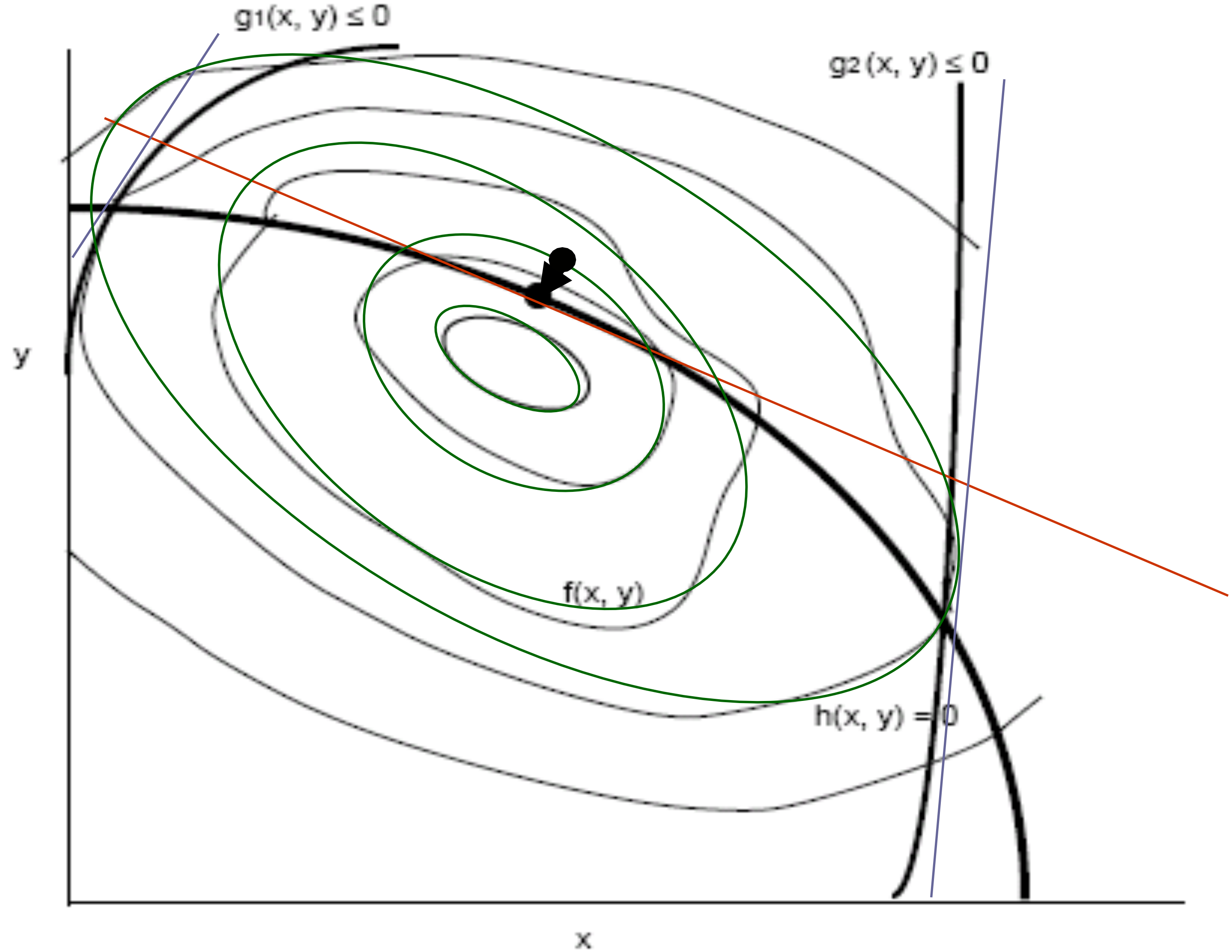
   (Each trial requires evaluation of $f(x)$, $g(x)$ and $h(x)$).

5. $x^{k+1} = x^k + \alpha d$. Set k = k + 1 Go to 2.



SQP – Set up and solve quadratic program

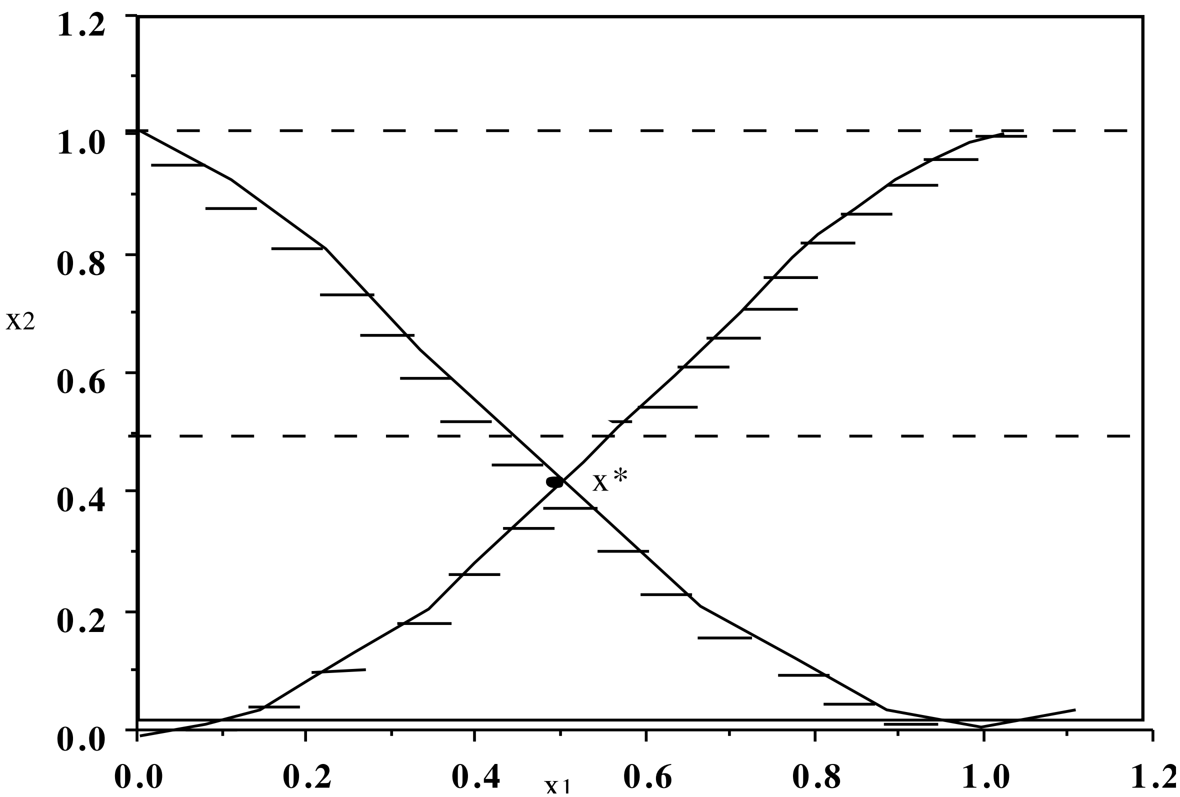
- Evaluate functions and gradients at current point
- Extended Newton method on KKT conditions



SQP – Set up and solve quadratic program

- Evaluate functions and gradients at current point
- Extended Newton method on KKT conditions

SQP Test Problem



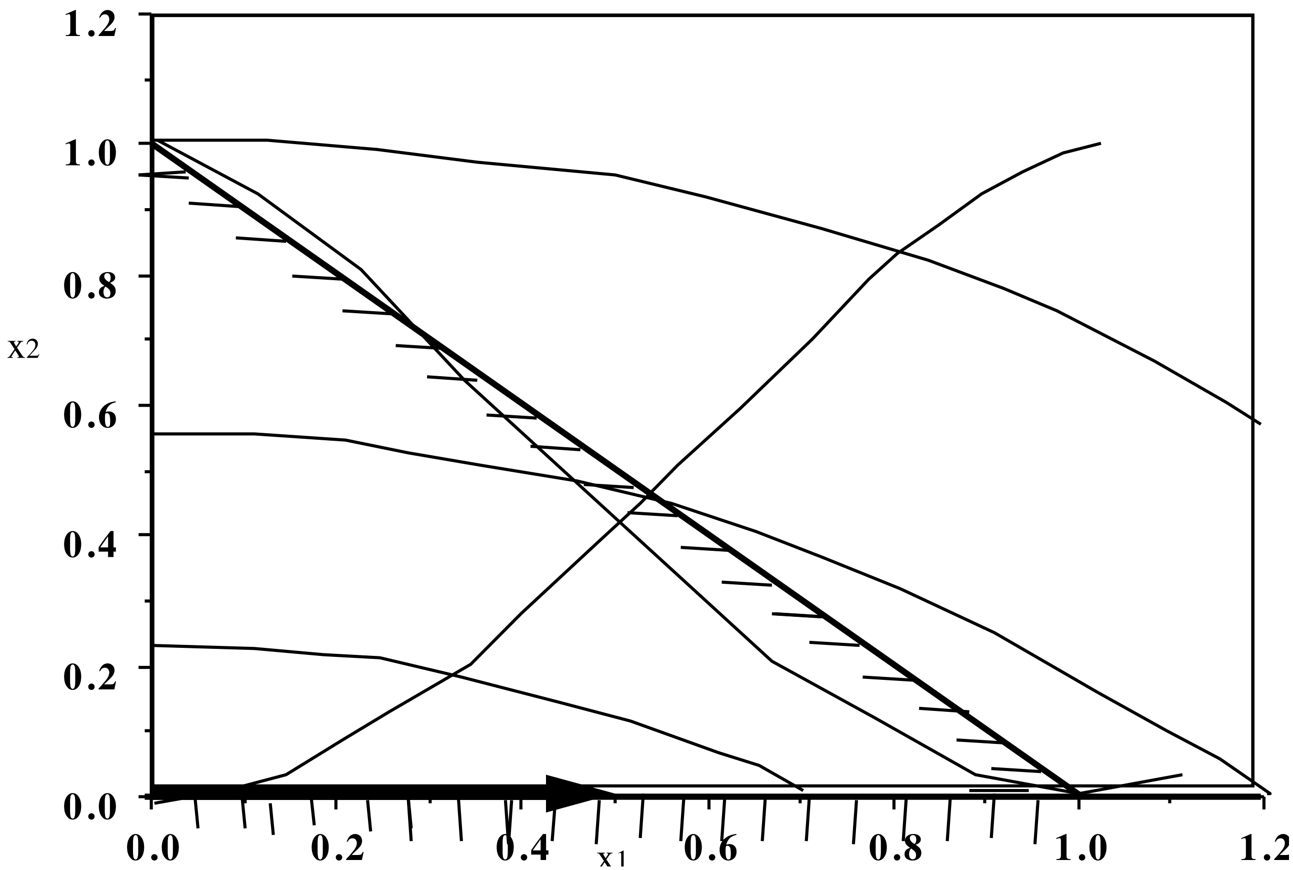
$$\text{Min} \quad x_2$$

$$\text{s.t.} \quad -x_2 + 2\,x_1^{\,2} - x_1^{\,3} \leq 0$$

$$-x_2 + 2\,(1-x_1)^2 - (1-x_1)^3 \leq 0$$

$$x^* = [0.5, 0.375].$$

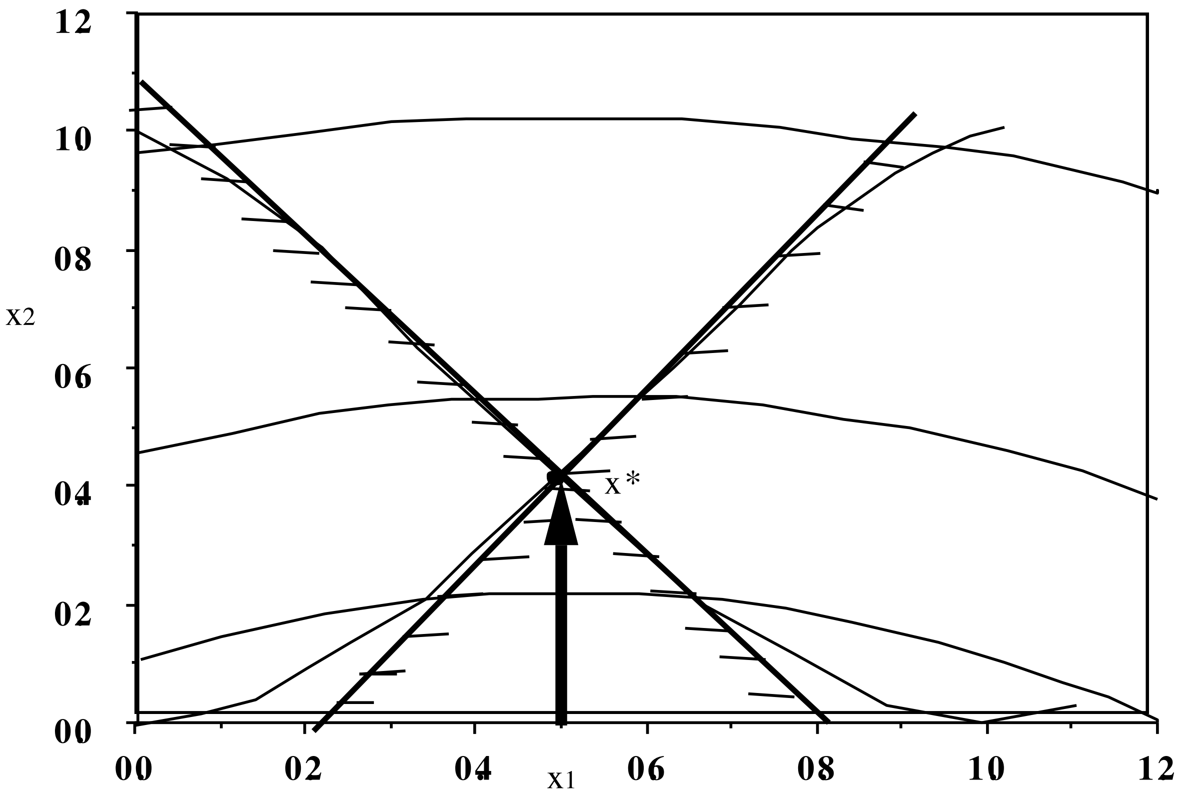
SQP Test Problem – First Iteration



Start from the origin ($x_0 = [0, 0]^T$) with $B^0 = I$, form:

$$\begin{aligned} &Min \quad d_2 + 1/2\,(d_1^2 + d_2^2) \\ &s.t. \quad d_2 \geq 0 \\ &\qquad d_1 + d_2 \geq 1 \\ &\qquad d = [1, 0]^T. \text{ with } \mu_1 = 0 \text{ and } \mu_2 = 1. \end{aligned}$$

SQP Test Problem – Second Iteration



From $x_1 = [0.5, 0]^T$ with $B^1 = I$
(no update from BFGS possible), form:

$$\begin{aligned}
&Min \quad d_2 + 1/2\,(d_1^2 + d_2^2)\\
&s.t. \quad -1.25\, d_1 - d_2 + 0.375 \le 0\\
&\qquad\quad 1.25\, d_1 - d_2 + 0.375 \le 0
\end{aligned}$$

$d = [0, 0.375]^T$ with $\mu_1 = 0.5$ and $\mu_2 = 0.5$

$x^* = [0.5, 0.375]^T$ is optimal

Barrier Methods for Large-Scale Nonlinear Programming

Original Formulation

$$\min_{x \in \Re^n} f(x)$$
$$\text{s.t} \quad c(x) = 0$$
$$x \geq 0$$

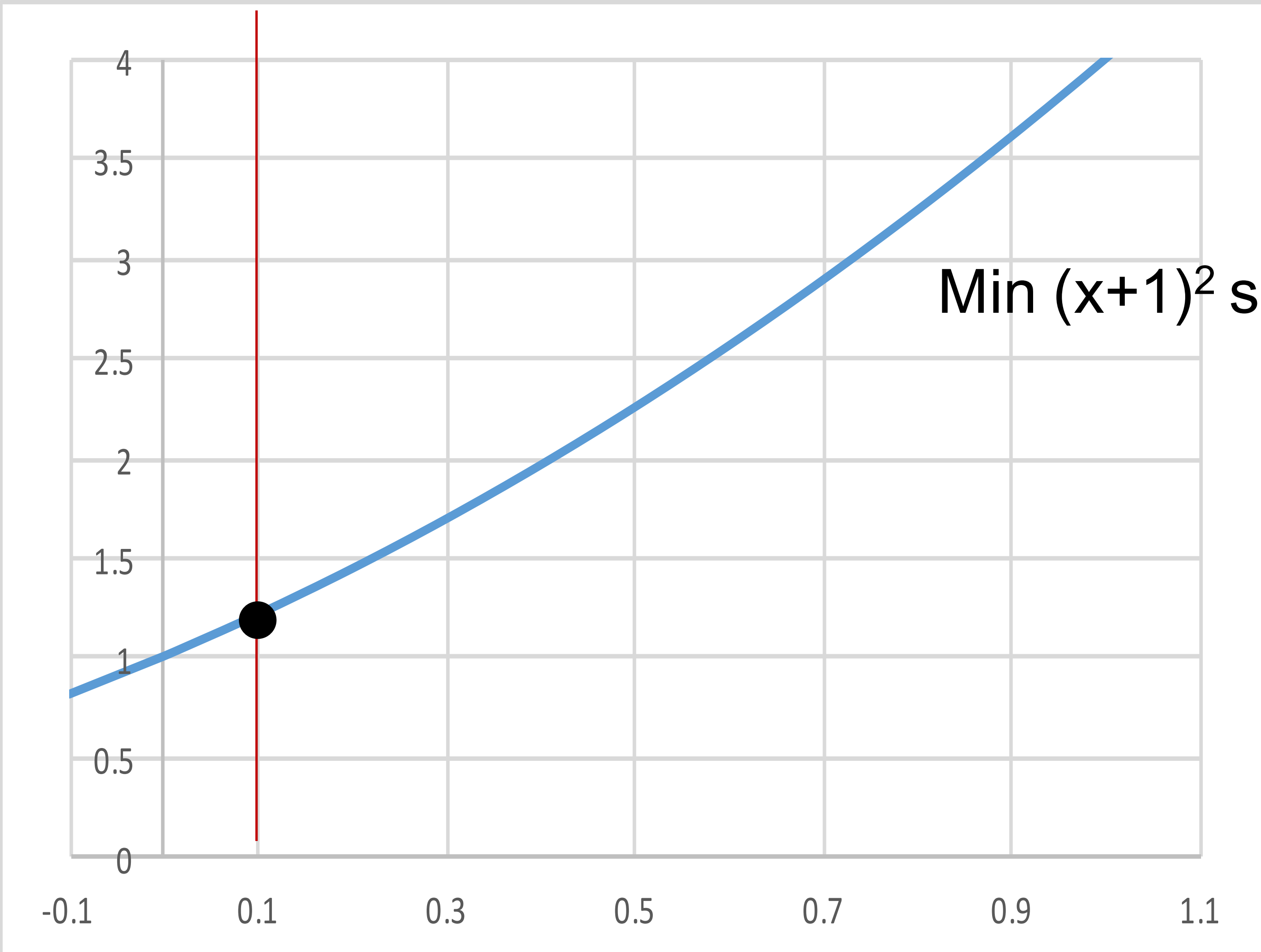
Can generalize for
$$a \leq x \leq b$$

Barrier Approach

$$\min_{x \in \Re^n} \varphi_\mu(x) = f(x) - \mu \sum_{i=1}^{n} \ln x_i$$
$$\text{s.t} \quad c(x) = 0$$

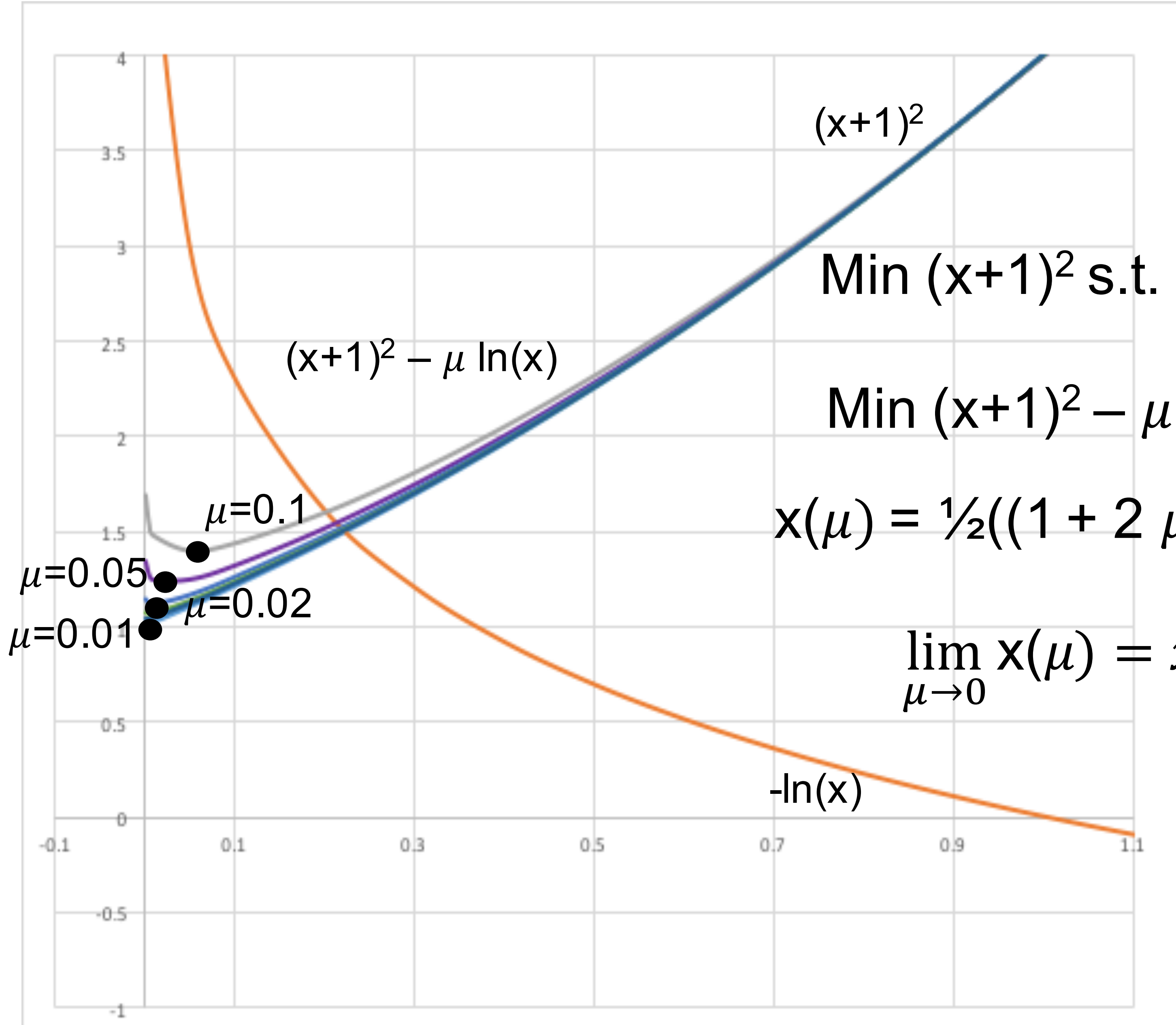
$\Rightarrow$As $\mu \rightarrow 0, \quad x^*(\mu) \rightarrow x^*$ Fiacco and McCormick (1968)

Example: Properties of Barrier Methods



Min $(x+1)^2$ s.t. $x \geq 0$

Example: Properties of Barrier Methods



Min $(x+1)^2$ s.t. $x \geq 0$

Min $(x+1)^2 - \mu \ln(x)$

$x(\mu) = \frac{1}{2}((1 + 2\mu)^{1/2} - 1)$

$$\lim_{\mu \to 0} x(\mu) = x^*$$

Solution of the Barrier Problem

⇒ **Newton Directions (KKT System)**

$$\begin{aligned} \nabla f(x) + A(x)\lambda - \nu &= 0 \\ X\nu - \mu e &= 0 \\ c(x) &= 0 \end{aligned}$$

$e^T = [1, 1, 1 \ldots],\ X = diag(x)$

$A = \nabla c(x),\ W = \nabla_{xx} L(x, \lambda, \nu)$

⇒ **Solve**

$$\begin{bmatrix} W & A & -I \\ A^T & 0 & 0 \\ V & 0 & X \end{bmatrix} \begin{bmatrix} d_x \\ d_\lambda \\ d_\nu \end{bmatrix} = -\begin{bmatrix} \nabla f + A\lambda - \nu \\ c \\ X\nu - \mu e \end{bmatrix}$$

Solution of the Barrier Problem

⇒Newton Directions (KKT System)

$$\begin{aligned} \nabla f(x) + A(x)\lambda - v &= 0 \\ Xv - \mu e &= 0 \\ c(x) &= 0 \end{aligned}$$

$e^T = [1,1,1\ldots],\ X = diag(x)$

$A = \nabla c(x),\ W = \nabla_{xx} L(x,\lambda,v)$

⇒ Reducing the System

$$d_v = \mu X^{-1} e - v - X^{-1} V d_x$$

$$\begin{bmatrix} W + \Sigma & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} d_x \\ \lambda^+ \end{bmatrix} = - \begin{bmatrix} \nabla \varphi_\mu \\ c \end{bmatrix}$$

$$\Sigma = X^{-1} V$$

IPOPT Code – www.coin-or.org

IPOPT Algorithm – Features

Line Search Strategies for Globalization

- l_2 exact penalty merit function
- augmented Lagrangian merit function
- Filter method (adapted and extended from Fletcher and Leyffer)

Hessian Calculation

- BFGS (full/LM and reduced space)
- SR1 (full/LM and reduced space)
- Exact full Hessian (direct)
- Exact reduced Hessian (direct)
- Preconditioned CG

Algorithmic Properties

Globally, superlinearly convergent (Wächter and B., 2005)

Easily tailored to different problem structures

Freely Available

Eclipse License and COIN-OR distribution: **http://www.coin-or.org**

IPOPT 3.x: rewritten in C++

Solved on thousands of test problems and applications

Evolution of Gradient-Based (NLP) Algorithms & Tasks

’80s: Flowsheet optimization
~ 100 variables and constraints

’90s: Static real-time optimization (RTO)
over 100 000 variables & constraints

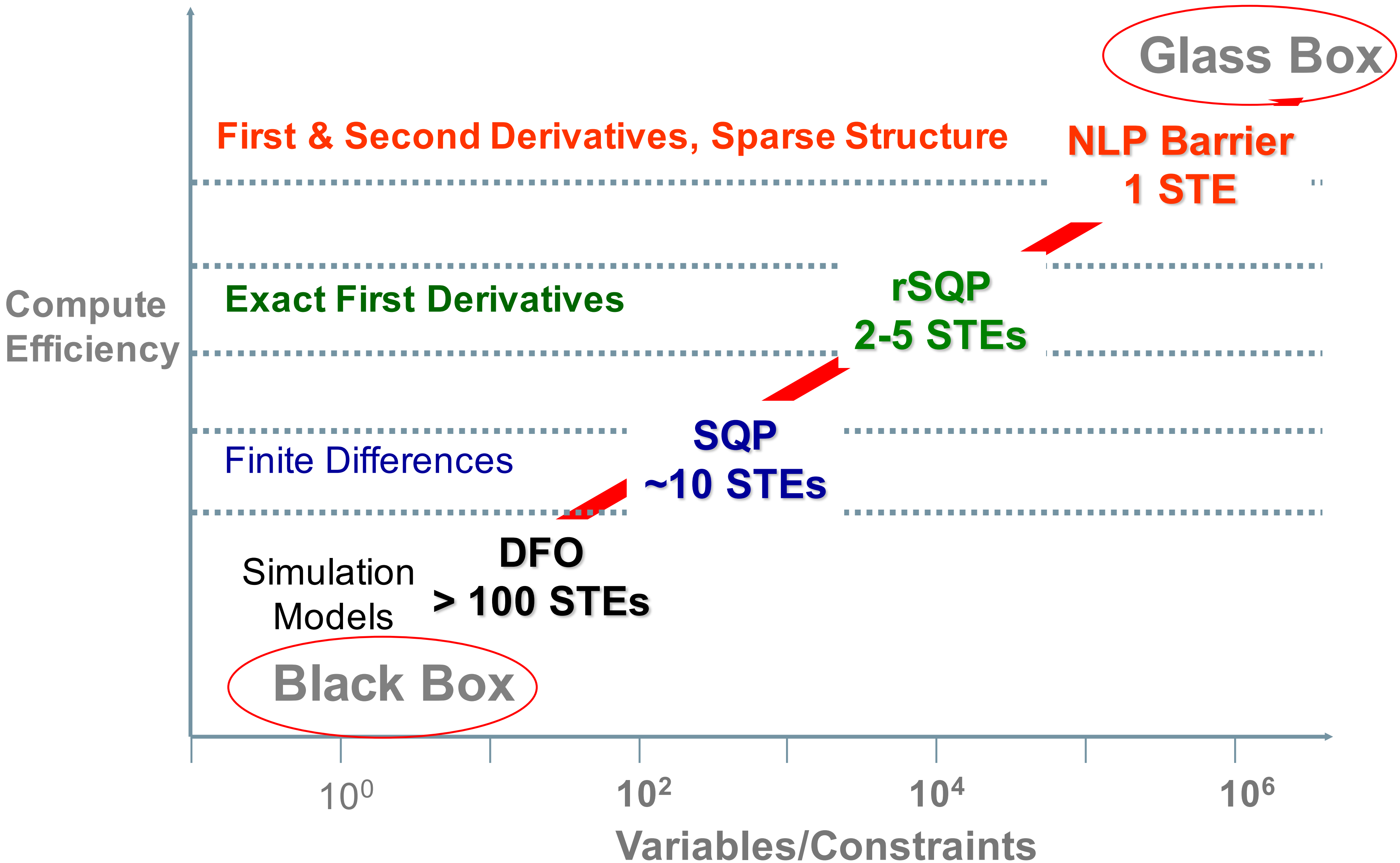
’00s: Simultaneous dynamic optimization
over 1 000 000 variables and constraints

‘10s: Sensitivity-based dynamic on-line
optimization for large NLPs: < 1 CPUs

SQP → rSQP → Barrier (IPOPT) → sIPOPT

The most efficient NLP tools now handle millions of variables and constraints with modest computational effort

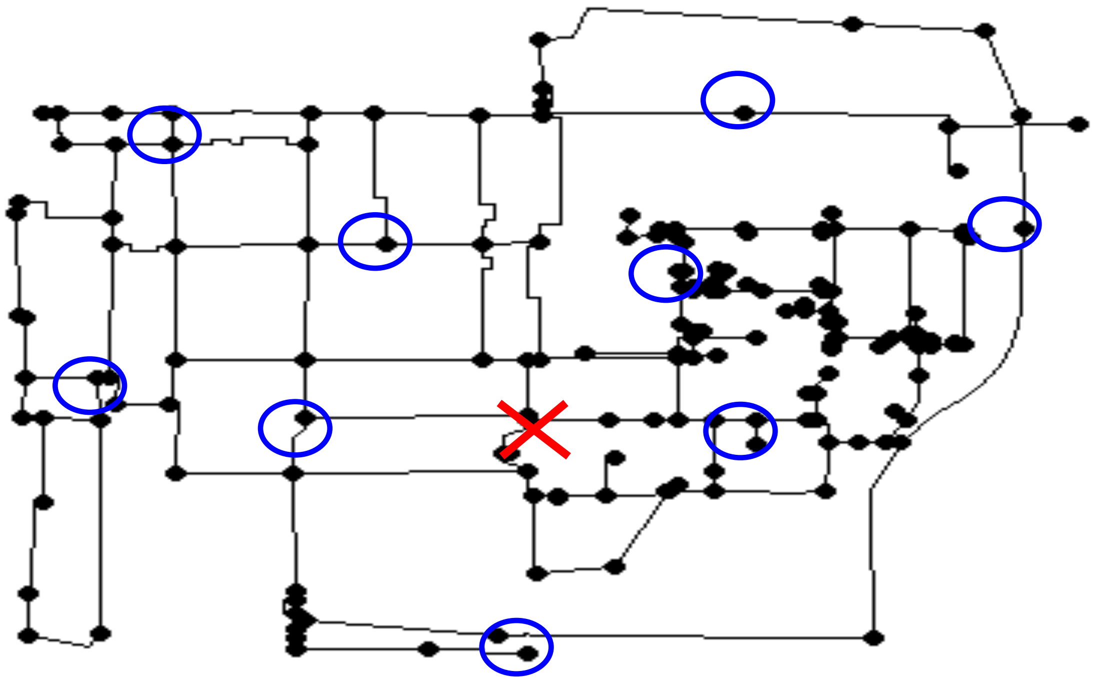
Process Optimization Environments and NLP Solvers



Equation-Oriented Utopia for Process Optimization

- Glass Box Models - Exact Jacobians/Hessians and sparse equation structure
- Fast Newton-based NLP solvers
- NLP sensitivity (post-optimality and interpretation, multi-level opt., …)
- EO-Modeling Enables:
  – Efficient MINLP Strategies
  – Deterministic Global Optimization
  – Robust and Stochastic Optimization for Uncertainty

Early Warning Detection System Municipal Water Networks
(Laird, B., 2005, 2006)

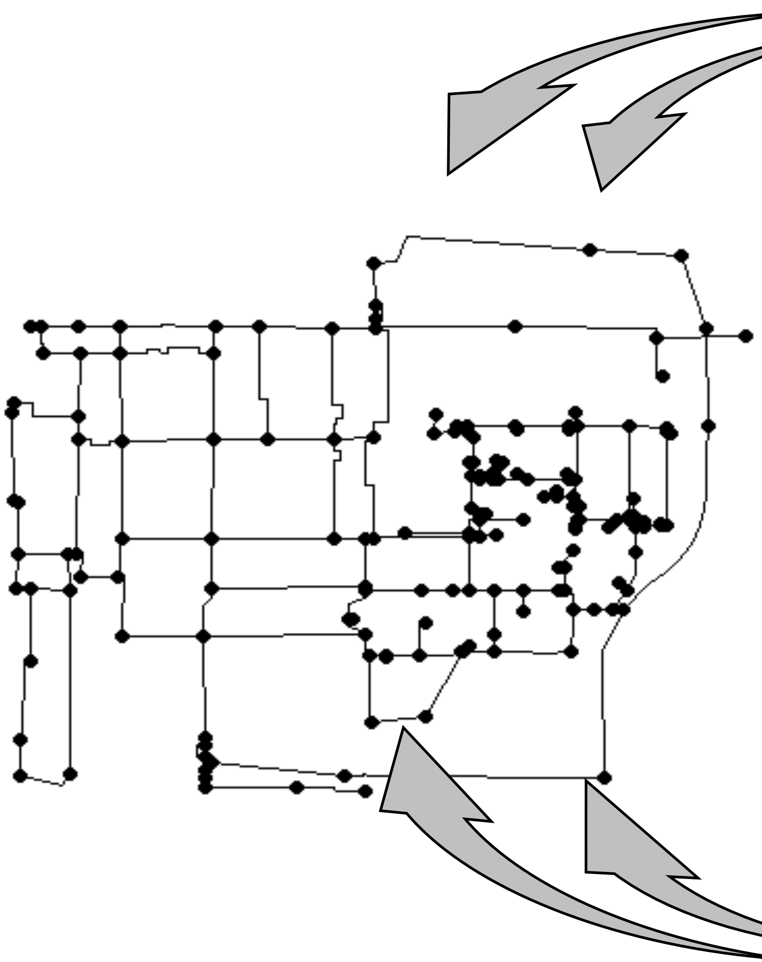


•Installed sensors provide an early warning of contamination

•System provides only a coarse measure of contamination time and location

•Desired: Accurate and fast time & location information

Water Quality Model



Pipes, Valves, Pumps

- Collapsed Node Models
- Plug Flow
- Complete Mixing
- No Reaction
- Known Sources Contaminant Free
- Time Dependent Mass Injections at All Nodes (Negligible Flow rates)
- Decoupled Hydraulics and Water Quality Calculations

Storage Tanks, Junctions

Equation-Oriented Optimization Formulation

Node Concentrations & Injection Terms Only

$$\min_{m(t),\bar{c}(x,t),\hat{c}(t)} \Psi = \sum_{r\in\Theta_s}\sum_{k\in\mathcal{N}_s} \frac{1}{2}\int_0^{t_f} w_k(t)\left(\hat{c}_k(t)-\hat{c}_k^{\star}(t)\right)^2\,\delta(t-t_r)\,dt+\frac{\rho}{2}\int_0^{t_f} m_k(t)^2 dt$$

Only Constraints with Spatial Dependence

$$\left.\begin{aligned} \frac{\partial \bar{c}_i(x,t)}{\partial t}+u_i(t)\frac{\partial \bar{c}_i(x,t)}{\partial x} &= 0,\\ \bar{c}_i(x{=}\mathcal{I}_i(t),t) &= \hat{c}_{k_i(t)}(t),\\ \bar{c}_i(x,t{=}0) &= 0, \end{aligned}\right\}\forall i\in\mathcal{P},$$

Pipe Boundary Concentrations

$$\hat{c}_k(t)=\frac{\left(\displaystyle\sum_{i\in\Gamma_k(t)} Q_i(t)\,\bar{c}_i(x{=}\mathcal{O}_i(t),t)\right)+m_k(t)}{\left(\displaystyle\sum_{i\in\Gamma_k(t)} Q_i(t)\right)+Q_k^{ext}(t)+Q_k^{inj}(t)},\qquad \forall k\in\mathcal{J},$$

$$\left.\begin{aligned} V_k(t)\frac{d\hat{c}_k(t)}{dt} &= \left(\sum_{i\in\Gamma_k(t)} Q_i(t)\,\bar{c}_i(x{=}\mathcal{O}_i(t),t)\right)+m_k(t)-\left[\left(\sum_{i\in\Gamma_k(t)} Q_i(t)\right)+Q_k^{ext}(t)+Q_k^{inj}(t)\right]\hat{c}_k(t),\\ \hat{c}_k(t{=}0) &= 0, \end{aligned}\right\}\forall k\in\mathcal{S},$$

Injection Terms Only

$$m_k(t)\geq 0,\qquad \forall k\in\mathcal{N}.$$

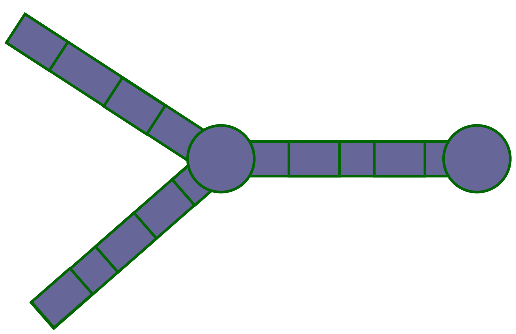
Pipeline Simulation Techniques

Eulerian

Discretize in time and space

Track concentration at fixed points or volumes

Local process for simulation, but global treatment needed for simultaneous optimization

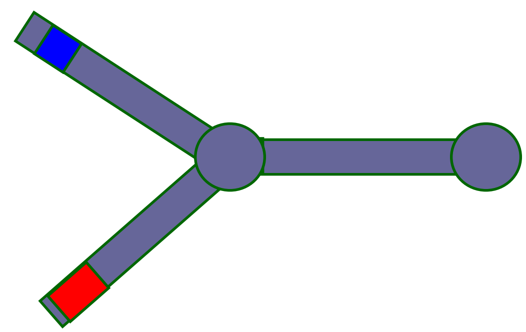


Lagrangian

Discretize in time alone

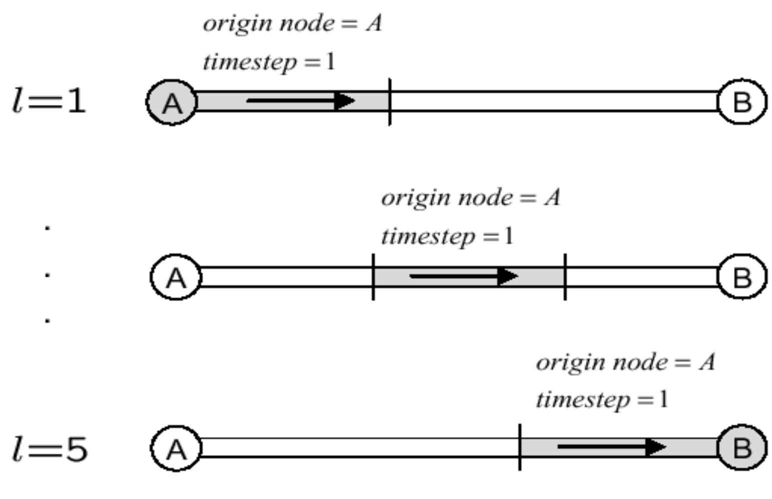
Track concentration of elements as they move

Algorithmic in nature



Review of methods by Rossman and Boulos, 1996.

Origin Tracking Algorithm



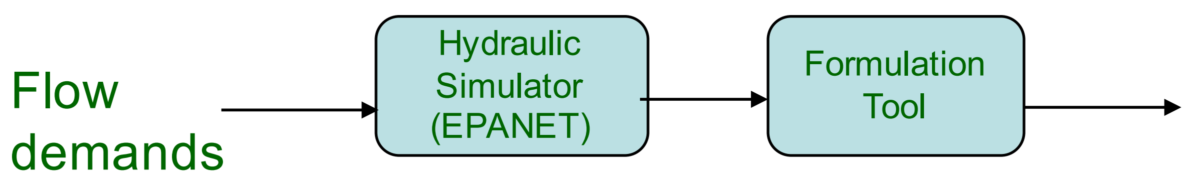
$$\bar{c}(x{=}\mathcal{I}_i(t_1), t_1) = \hat{c}_A(t_1)$$
$$\bar{c}(x{=}\mathcal{O}_i(t_1), t_1) = 0$$

$$\bar{c}(x{=}\mathcal{I}_i(t_5), t_5) = \hat{c}_A(t_5)$$
$$\bar{c}(x{=}\mathcal{O}_i(t_5), t_5) = \hat{c}_A(t_1)$$

Known Hydraulics – Function of Time
Pipe Network PDEs Linear in Concentration
Pipe by Pipe PDEs

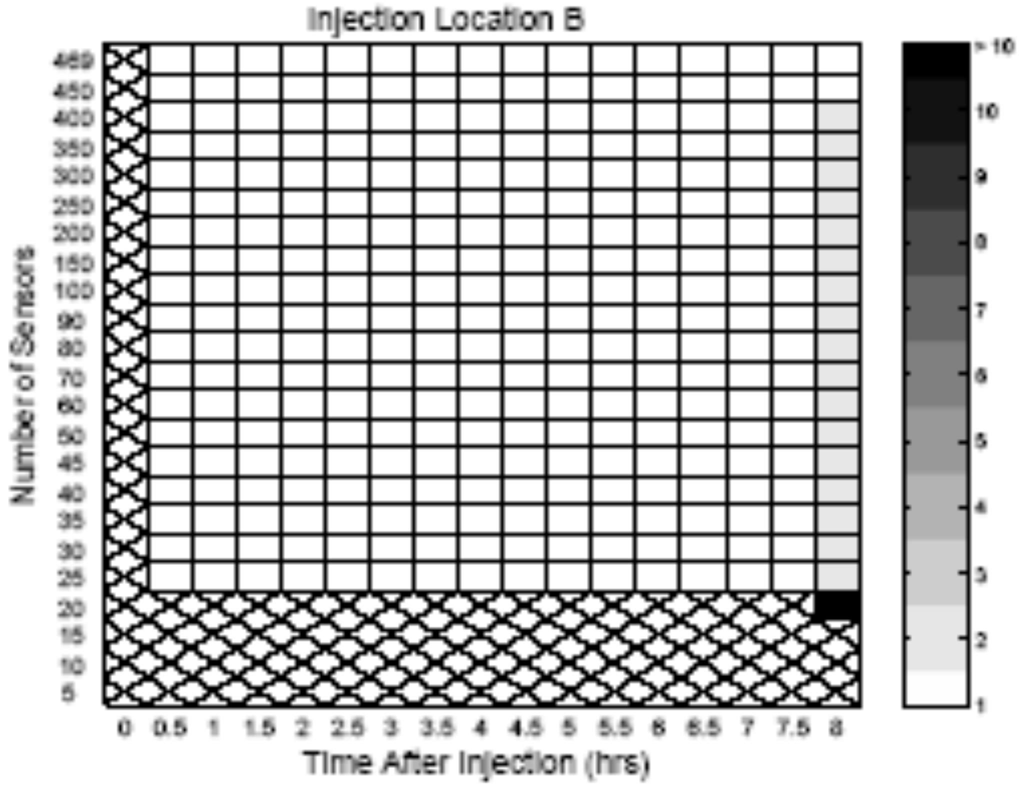
- Efficient for Large Networks
- Convert PDEs to DAEs with variable time delays

Removes Need to Discretize in Space



$$\begin{aligned} \min_{\bar{c},c,m} \quad & f(c,m) \\ \text{s.t.} \quad & \bar{c} - Pc = 0, \\ & \bar{N}\bar{c} + Nc + Mm = 0, \\ & m \geq 0 \end{aligned}$$

Municipal Source Detection Example



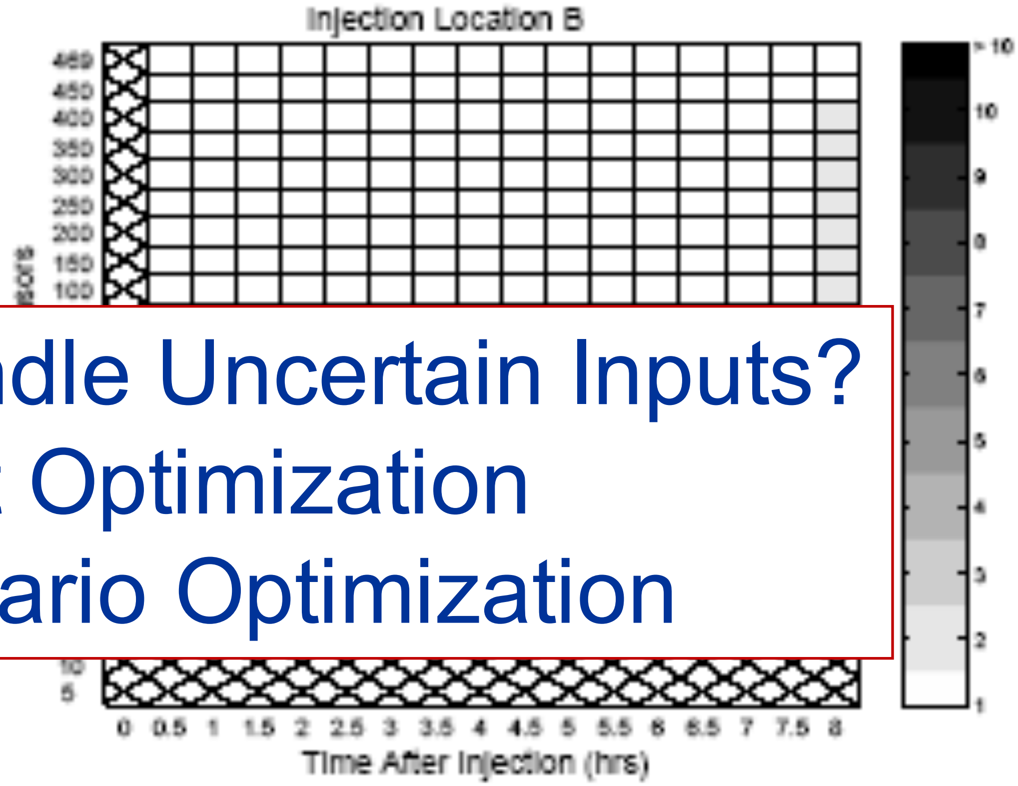
Algorithm successful on over 10(networks

Links to existing water flow network simulator ➔ variable time delays

Solution time < 2 CPU minutes for ~ 250,000 variables, ~45,000 degrees of freedom ➔ Effective in a real time setting

Can impose unique solutions through an extended MIQP formulation (post-processing phase)

Municipal Source Detection Example



How do we handle Uncertain Inputs?
Robust Optimization
Multi-Scenario Optimization

Alg
networks

Links to existing water flow network simulator ➔ variable time delays

Solution time < 2 CPU minutes for ~ 250,000 variables, ~45,000 degrees of freedom ➔ Effective in a real time setting

Can impose unique solutions through an extended MIQP formulation (post-processing phase)

Multi-scenario Optimization

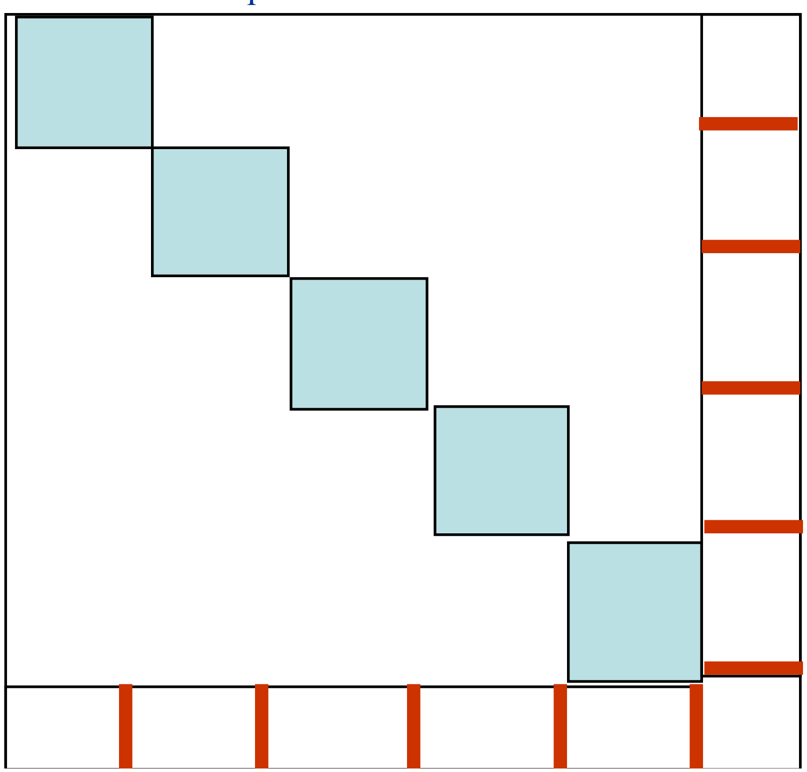
$$\begin{aligned}
&Min\ f_0(d) + \Sigma_i f_i(d, x_i)\\
&s.t.\ h_i(x_i, d) = 0,\ i = 1, \ldots N\\
&\quad g_i(x_i, d) \le 0,\ i = 1, \ldots N\\
&\quad r(d) \le 0
\end{aligned}$$

Variables:

x: state (z) and decision (y) variables for each scenario

d: common variables (e. g. equipment parameters) used

δ_i: substitute for d in each period and add $\delta_i = d$



Composite NLP

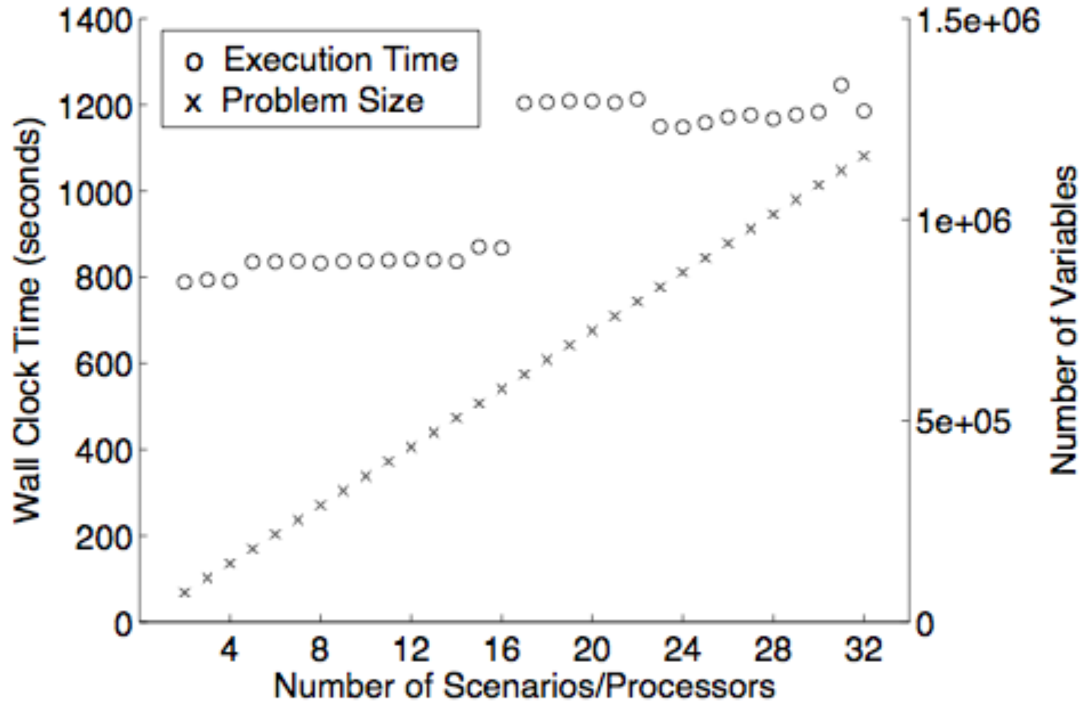
$$\begin{aligned}
&Min\ \ \Sigma_i\ (f_i(\delta_i, x_i) + f_0(\delta_i)/N)\\
&s.t.\ h_i(x_i, \delta_i) = 0,\ i = 1, \ldots N\\
&\quad g_i(x_i, \delta_i) + s_i = 0,\ i = 1, \ldots N\\
&\quad 0 \le s_i,\ \underline{d - \delta_i = 0},\ i = 1, \ldots N\\
&\quad r(d) \le 0
\end{aligned}$$

Parallel Schur-Complement Scalability

- Multi-scenario Optimization
  - Single Optimization over many scenarios, performed on paralle' cluster

  Water Network Case Study
  - 1 basic model
    - Nominal design optimization
  - 32 scenarios (operating data)
    - Form individual blocks

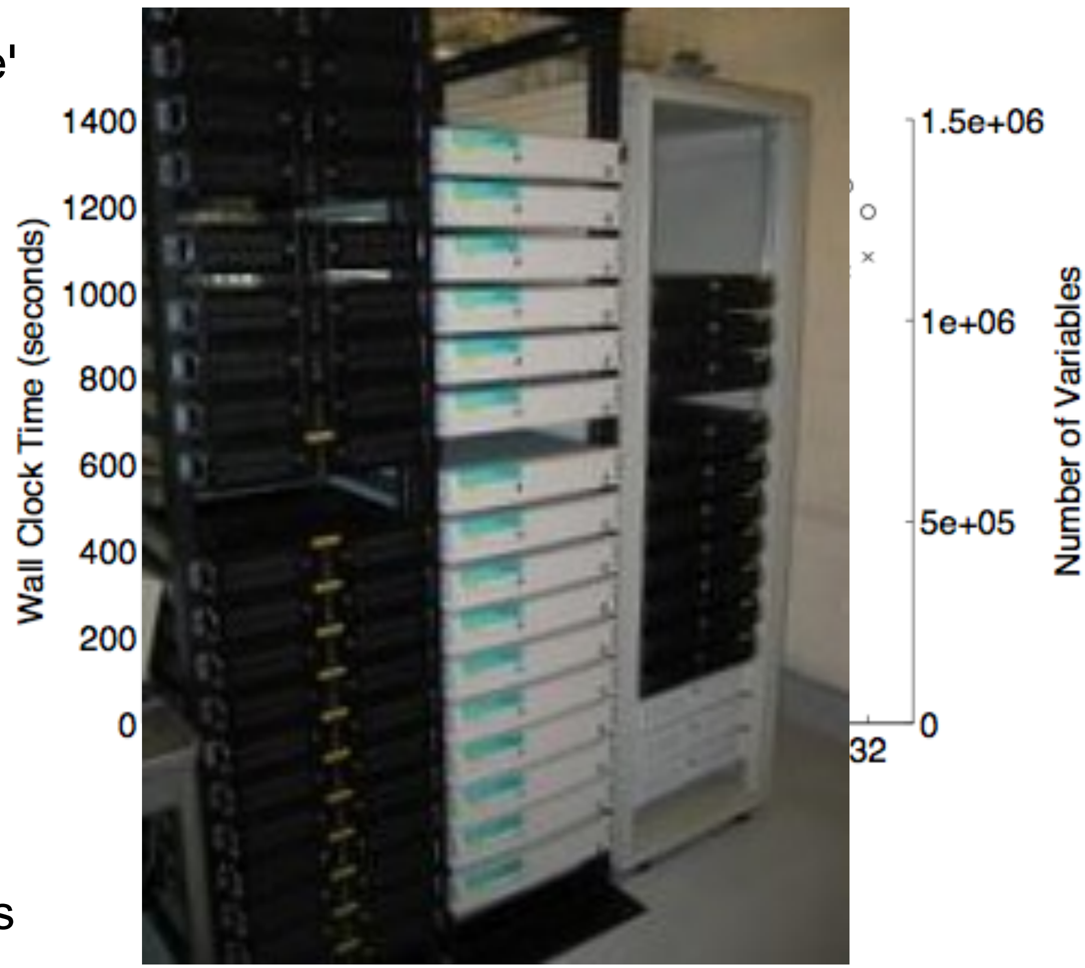
- Determine Injection time profiles as common variables
- Characteristics
  - 36,000 variables per scenario
  - 600 common variables
  - Solution with 1.2 x 10^6 variables
  (20 CPU min)



Parallel Schur-Complement Scalability

- Multi-scenario Optimization
  - Single Optimization over many scenarios, performed on paralle' cluster

Water Network Case Study
- 1 basic model
  - Nominal design optimization
- 32 scenarios (operating data)
  - Form individual blocks



- Determine Injection time profiles as common variables
- Characteristics
  - 36,000 variables per scenario
  - 600 common variables
  - Solution with 1.2 x 10^6 variables (20 CPU min)

Decomposition in Parallel Structures

$$\begin{bmatrix} K_1 & & & & A_1 \\ & K_2 & & & A_2 \\ & & \ddots & & \vdots \\ & & & K_N & A_N \\ A_1^T & A_2^T & \dots & A_N^T & D \end{bmatrix}$$

Optimization under Uncertainty
Multi-Scenario → Schur Complement

$$\begin{bmatrix} W_{u_1u_1} & f_{u_1} & & & & & & & & \\ f_{u_1}^T & 0 & -I & & & & & & & \\ & -I & W_{x_2x_2} & W_{x_2u_2} & f_{x_2} & & & & & \\ & & W_{u_2x_2} & W_{u_2u_2} & f_{u_2} & & & & & \\ & & f_{x_2}^T & f_{u_2}^T & 0 & -I & & & & \\ & & & & -I & W_{x_3x_3} & W_{x_3u_3} & f_{x_3} & & \\ & & & & & W_{u_3x_3} & W_{u_3u_3} & f_{u_3} & & \\ & & & & & f_{x_3}^T & f_{u_3}^T & 0 & \ddots & \\ & & & & & & & \ddots & \ddots & -I \\ & & & & & & & & -I & W_{x_Nx_N} \end{bmatrix}$$

Dynamic Optimization
Block Tridiagonal → Cyclic Reduction
Nicholson et al. (2016)

$$\begin{bmatrix} \boxed{K_1} & & & & A_1 \\ & \boxed{K_2} & & & A_2 \\ & & \ddots & & \vdots \\ & & & \boxed{K_N} & A_N \\ A_1^T & A_2^T & \dots & A_N^T & D \end{bmatrix}$$

Dynamic Optimization under Uncertainty
Block Tridiagonal with Multi-Scenario →
Nested Cyclic Reduction and Schur Complement
Gondzio and Grothey (2011)

The Next 15 – 25 Years
Giga-scale Process Optimization

Enabling Tools:

- Structured NLPs with billions of variables
- Friendly, Powerful and Intelligent Optimization Modeling Environments
- Distributed Optimization Solvers with Exploitable Large-scale Structures
- Integrated with Advanced Computation Environments (Multi-core CPUs, GPUs…)

Applications:

- Dynamic Global Network Models
- Electric Grid
- Gas and Oil Pipelines
- Enterprise-Wide Dynamic, Real-time Optimization

Extend Equation-Oriented Optimization to (Many) Discrete Decisions?

Min Overall Objective
s.t. Conservation Laws
    Performance Equations
    Constitutive Equations
    Phase and Chemical Equilibrium
    Heat Integration
    Process/Product Specifications

→ Consider MPCCs derived from Bi-level Optimization

Extend Equation-Oriented Optimization to (Many) Discrete Decisions?

Min Overall Objective
s.t. Conservation Laws
    Performance Equations
    Constitutive Equations
    Process/Product Specifications

Minimize Gibbs Free Energy
(Phase/Chem. Equilibrium)

Minimize Utilities
(Heat Integration)

→ Consider MPCCs derived from Bi-level Optimization

Bilevel Problems and NLP Reformulations

$$\min_{x,y} \; f(x, y)$$

$$\text{s.t. } g(x, y) \le 0, \; h(x, y) = 0$$

$$\boxed{\begin{array}{l} \min_{y} \; \bar{f}(x, y) \\ \text{s.t. } \bar{g}(x, y) \le 0, \; \bar{h}(x, y) = 0 \end{array}}$$

Formulation Guidelines

- Attempt to define regular, convex inner minimization problem (optimistic bilevel problems, Dempe, 2002)
- Require connected feasible regions for inner problem variables (no exclusive ORs!)
- How can this problem be solved?

Mathematical Programs with Complementarity Constraints (MPCC)

$$\min_{x,y} \; f(x, y)$$

$$\text{s.t. } g(x, y) \leq 0, \; h(x,y) = 0$$

$$\boxed{\begin{array}{l} \nabla_y \bar{f}(x, y) + \nabla_y \bar{g}(x, y)u + \nabla_y \bar{h}(x, y)v = 0 \\ \bar{g}(x, y) \leq 0, \; \bar{h}(x,y) = 0 \\ \boxed{0 \leq u \perp \bar{g}(x, y) \leq 0} \end{array}}$$

Formulation Guidelines

•Substitute optimality conditions as constraints for bilevel problem ➔ need to solve "singular system"

•Poorly posed optimization problem, constraint qualifications violated

Solving MPCCs through NLP Reformulation – ℓ_1 penalty

$$\min_{x,y}\ f(x,y) \boxed{- u^T \bar{g}(x,y)}$$

$$\text{s.t. } g(x,y) \le 0,\ h(x,y) = 0$$

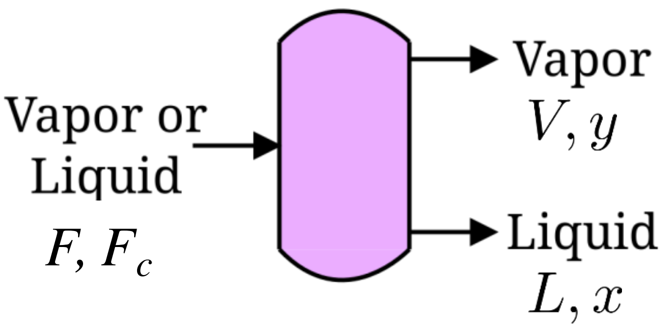
$$\boxed{\begin{array}{l} \nabla_y \bar{f}(x,y) + \nabla_y \bar{g}(x,y)u + \nabla_y \bar{h}(x,y)v = 0 \\ \bar{g}(x,y) \le 0,\ \bar{h}(x,y) = 0 \\ \boxed{0 \le u,\ \ \bar{g}(x,y) \le 0} \end{array}}$$

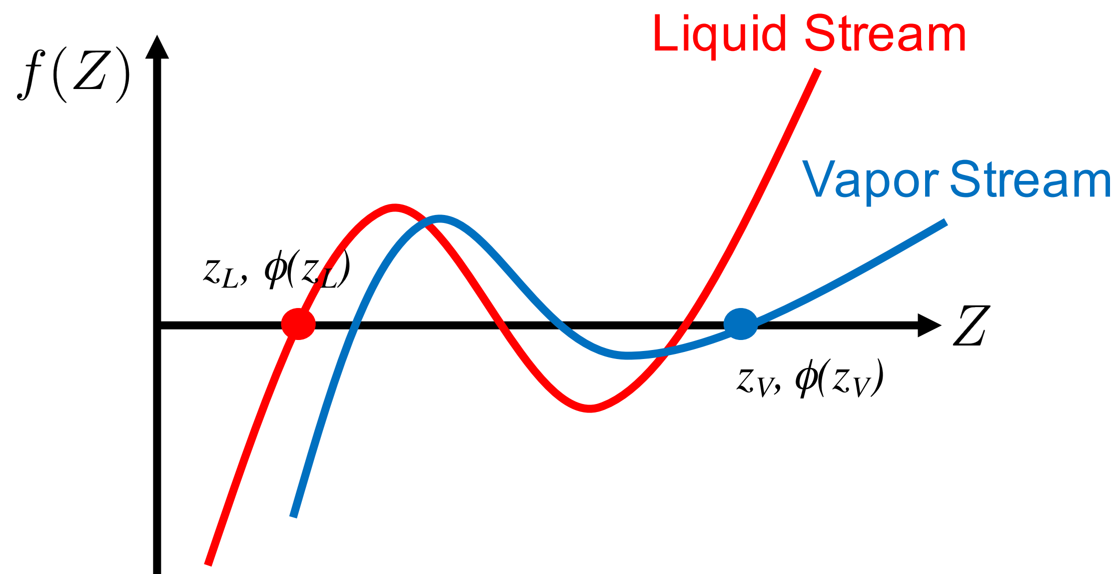
Formulation Guidelines

- Replace complementarity by penalty term and simple inequalities
- Well-posed optimization problem, constraint qualifications satisfied

Phase Equilibrium thru Complementarity

$$Z^3 - (1 + B - uB)Z^2 + (A + wB^2 - uB - uB^2)Z - AB - wB^2 - wB^3 = 0$$





$$F = L + V$$

$$F_c = Lx_c + Vy_c, \quad \forall c \in \{Comps\}$$

$$FH^F + Q = LH^L + VH^V$$

$$y_c = \quad K_c(T, P, x, y)x_c$$

$$0 \leq x_c, y_c \leq 1$$

$$0 \leq L, V \leq F$$

$$K_c = \phi_c^L / \phi_c^V, \quad \forall c \in \{Comps\}$$

$$f(z_L) = 0 \qquad f(z_V) = 0$$

$$f'(z_L) \geq 0 \qquad f'(z_V) \geq 0$$

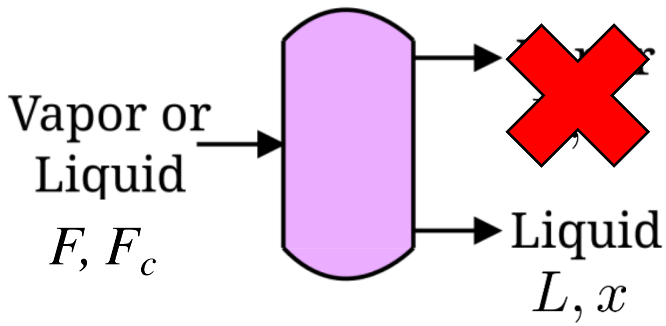
$$f''(z_L) \leq 0 \qquad f''(z_V) \geq 0$$

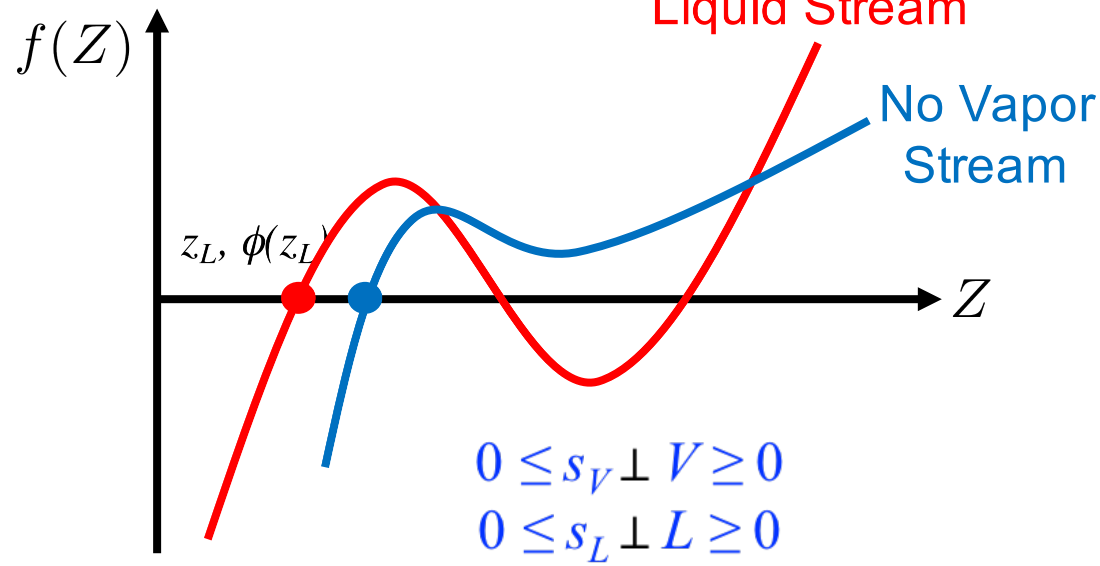
Mass Balance + Necessary KKT conditions

Kamath, Grossmann, B., Comp. Chem. Eng. 2011; Dowling, B., Comp. Chem. Eng., 2014

CEOS Phase Equilibrium thru Complementarity

$$Z^3 - (1 + B - uB)Z^2 + (A + wB^2 - uB - uB^2)Z - AB - wB^2 - wB^3 = 0$$





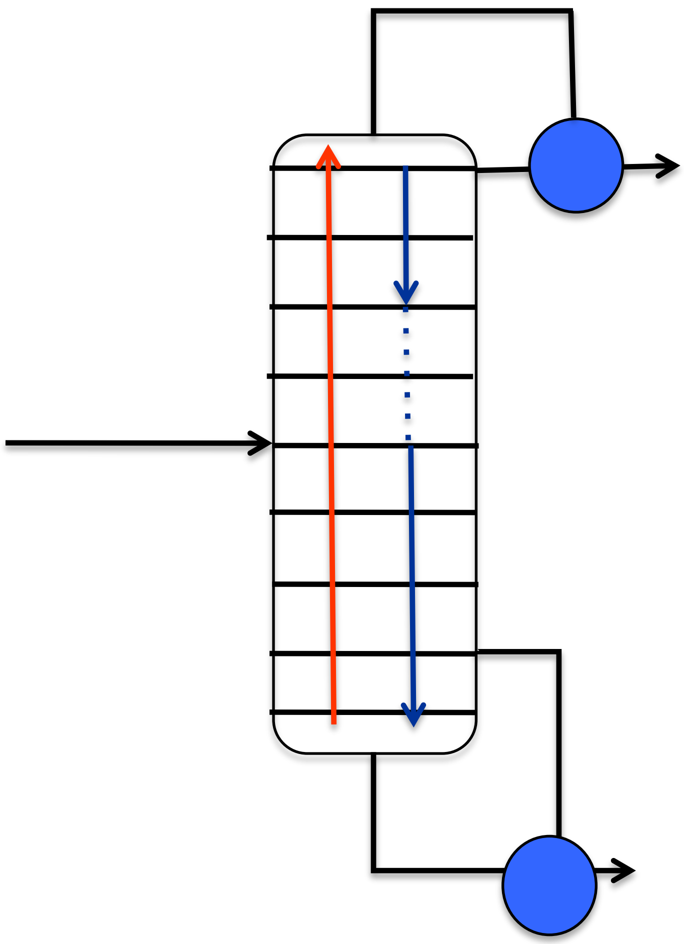
$$0 \le s_V \perp V \ge 0$$
$$0 \le s_L \perp L \ge 0$$
$$-s_L \le \beta - 1 \ge s_V$$

$$F = L + V$$
$$F_c = Lx_c + Vy_c, \quad \forall c \in \{Comps\}$$
$$FH^F + Q = LH^L + VH^V$$
$$y_c = \beta\ K_c(T, P, x, y)x_c$$
$$0 \le x_c, y_c \le 1$$
$$0 \le L, V \le F$$

$$K_c = \phi_c^L / \phi_c^V, \quad \forall c \in \{Comps\}$$

$f(z_L) = 0$	$f(z_V) = 0$
$f'(z_L) \ge 0$	$f'(z_V) \ge 0$
$f''(z_L) \le 0$	$f''(z_V) \ge 0$
$f''(z_L) \le M s_L$	$f''(z_V) \ge -M s_V$

Distillation: Complementarity Formulation
(Raghunathan, B, 2002; Kamath et al., 2010)



- Consists of Mass, Equilibrium, Summation and Heat (MESH) equations
- Continuous Variable Optimization
  - number of trays
  - feed location
  - reflux ratio

- **When phases disappear, MESH fails.**
- Reformulate phase minimization,
  - embed complementarity
  - Model dry trays, Vaporless trays

- How does this extend to distillation optimization?

Distillation Optimization (MESH Model)

Minimize Reboiler Duty

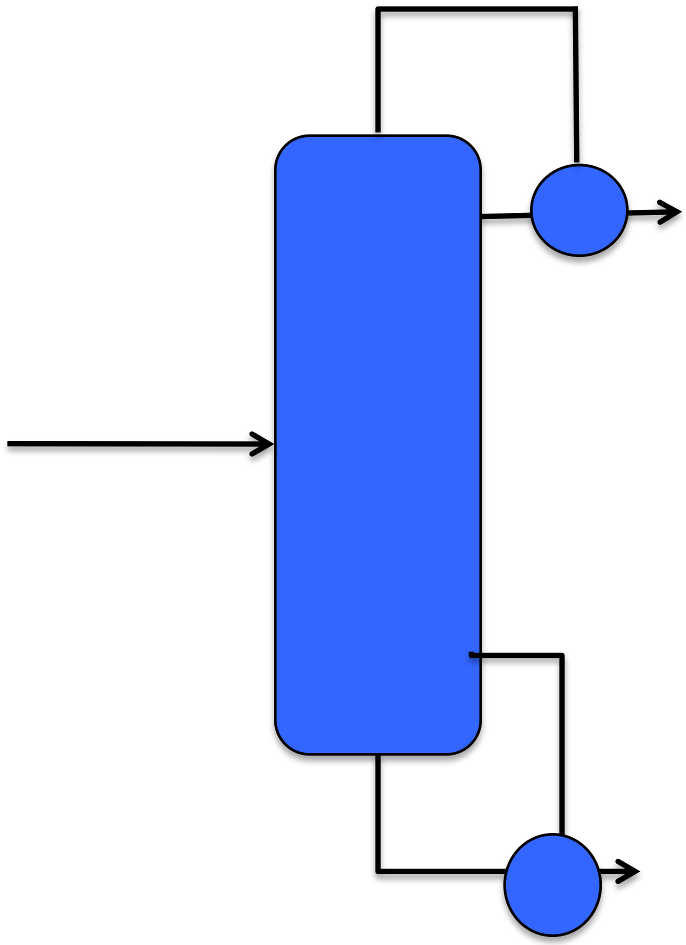
s.t. Top/ Bottom Product Specifications

Mass

Equilibrium

Heat

Summation



- Three component Separation (nC6, nC7, nC9)
- Fixed: 20 trays, Feed = 10
- Reboil and Reflux as decisions

Distillation Optimization (MESH Model)

Minimize Reboiler Duty

s.t. Top/ Bottom Product Specifications

$$(L_i + DL + rd)x_{ij} + DVy_{ij} = V_{i-1}y_{i-1,j} \quad i \in CON$$

$$L_i x_{ij} + V_i y_{ij} = L_{i+1}x_{i+1,j} + V_{i-1}y_{i-1,j} + \sum_k f_{ik} Fd_k xf_{dj} + g_i \cdot rd \cdot x_{ij} \quad i \in COL$$

$$Bx_{ij} + V_i y_{ij} = L_{i+1}x_j + \sum_k f_{ik} Fd_k xf_{dj} \quad i \in REB$$

$$\boxed{\begin{aligned} & y_{ij} = \beta_i K_{ij} x_{ij} \\ & -s_i^V \le \beta_i - 1 \le s_i^L \\ & 0 \le L_i \perp s_i^L \ge 0 \\ & 0 \le V_i \perp s_i^V \ge 0 \end{aligned}}$$

$$(L_i + DL + rd)hl_i + DV \cdot hv_i = V_{i-1}hv_{i-1} + Q_c \quad i \in CON$$

$$L_i hl_i + V_i hv_i = L_{i+1}hl_{i+1} + V_{i-1}hv_{i-1} + \sum_k f_{ik} Fd_k shf_{dj} + g_i \cdot rd \cdot hl_i \quad i \in COL$$

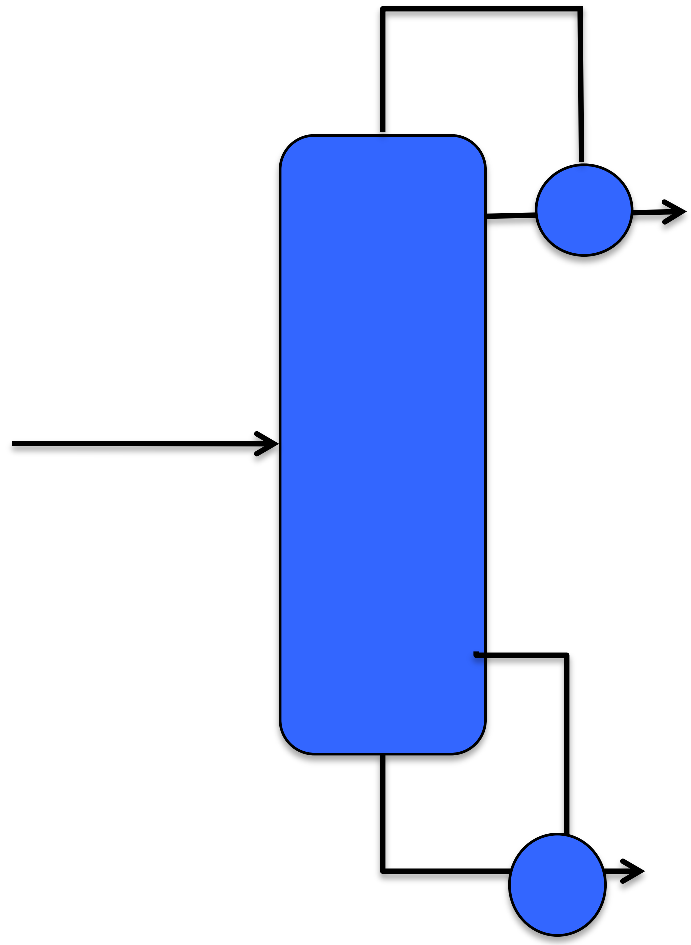
$$B \cdot hl_i + V_i hv_i = L_{i+1}hl_{i+1} + \sum_k f_{ik} Fd_k shf_{dj} + Q_H \quad i \in REB$$

$$\sum_j x_{ij} - \sum_j y_{ij} = 1$$

$$R_{total} = R \cdot D$$

$$L_{N_{max}} = (1 - rdf)R_{total}$$

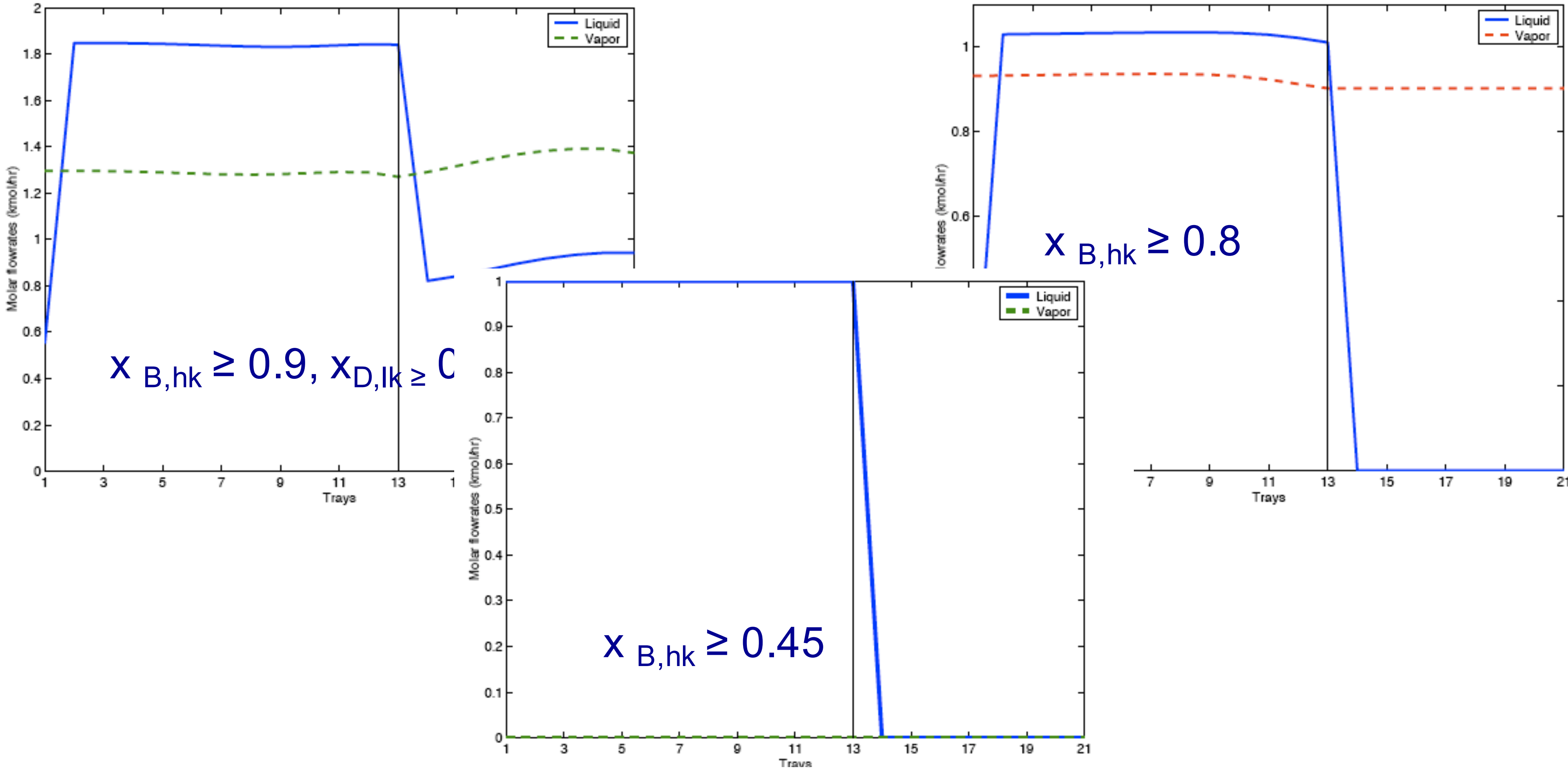
$$rd = rdf \cdot R_{total}$$



- Three component Separation (nC6, nC7, nC9)
- Fixed: 20 trays, Feed = 10
- Reboil and Reflux as decisions

Distillation Results – Min Heat Duty

(Kamath et al., 2010)



Case	Product purity constraints	Optimal reboiler duty (MW)
1	$x_{\text{top},n\text{-hexane}} \geq 0.9$, $x_{\text{bottom},n\text{-nonane}} \geq 0.9$	28.14
2	$x_{\text{bottom},n\text{-nonane}} \geq 0.8$	19.337
3	$x_{\text{bottom},n\text{-nonane}} \geq 0.45$	0.0

Glass Box Optimization: Air Separation Units (Dowling, B., 2015)

Boiling pts (1 atm.)

- Oxygen: 90 K
- Argon: 87.5 K
- Nitrogen: 77.4 K

Feedstock (air) is free: dominant cost is compression energy

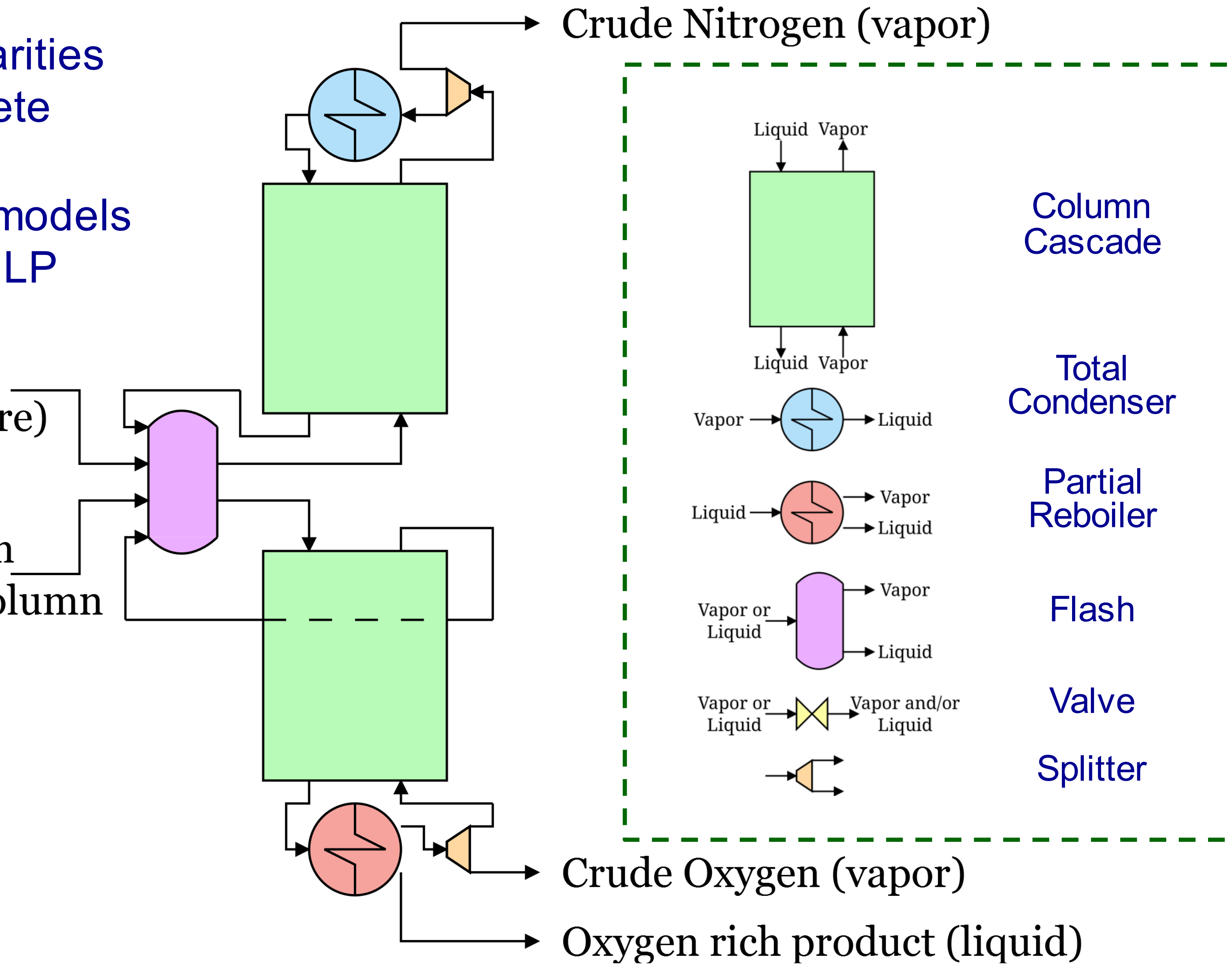
Multicomponent distillation with tight heat integration

Nonideal Phase Equilibrium: Cubic Equations of State

Phase conditions not known *a priori*

ASU Superstructure – Building the Column

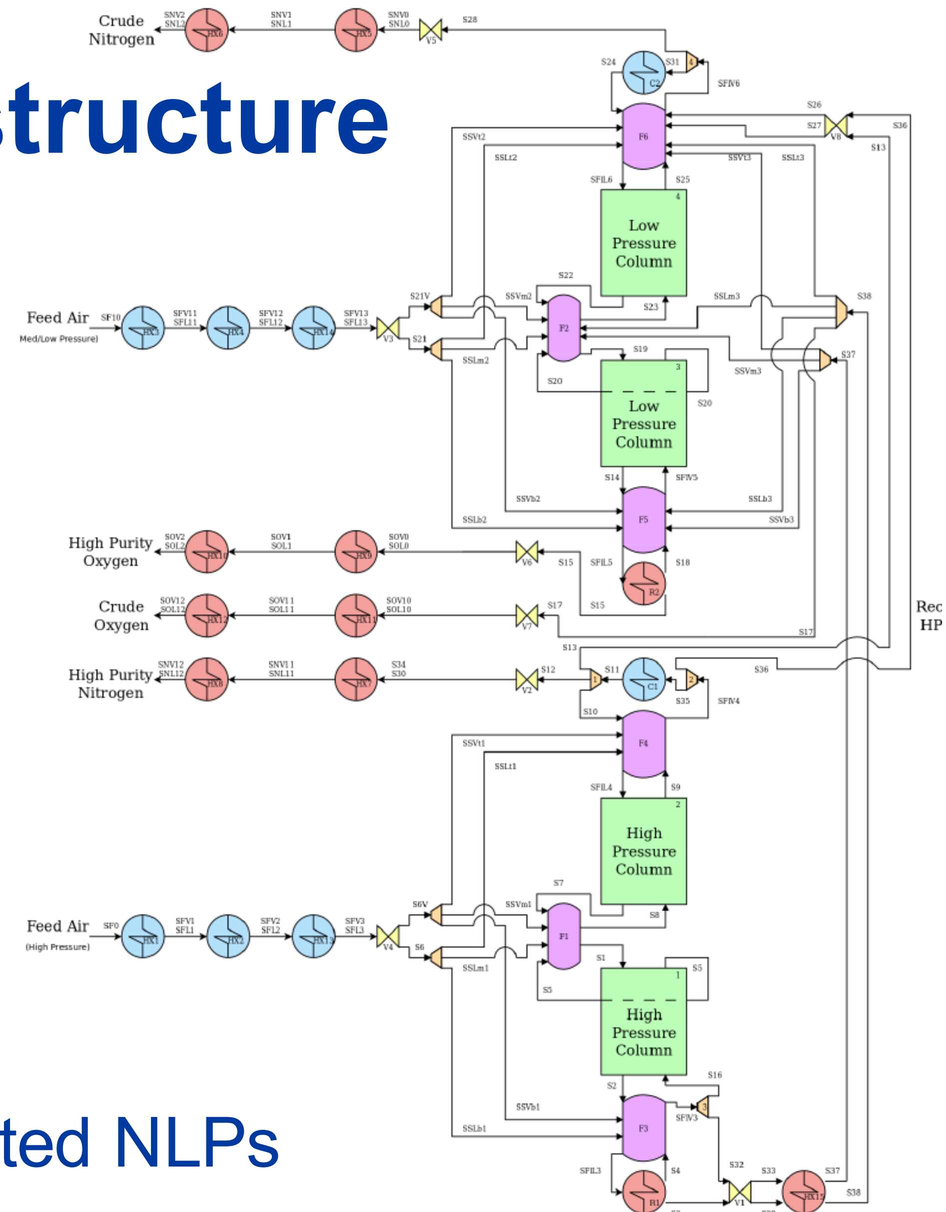
- Complementarities replace discrete decisions
- Tray bypass models
- Continuous NLP



ASU NLP Superstructure

min ASU Compression Energy (kWh / kg O_2 product)

s.t.
Thermodynamics Module
Unit Operation Models
Cascade Model
Flowsheet Connectivity
Heat Integration
O_2 product purity $\geq$ 95 mol%

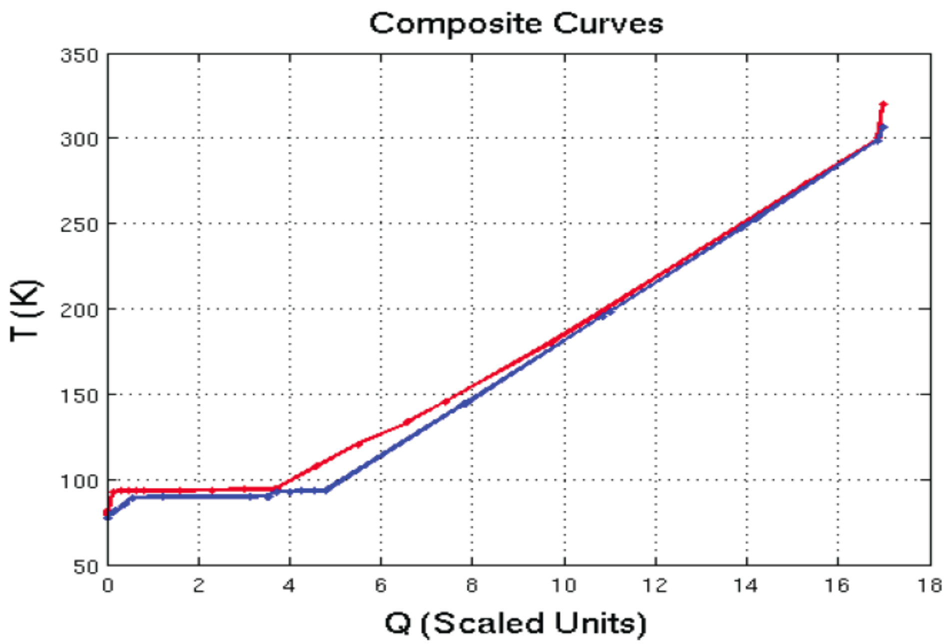


- Glass Box strategy with nested NLPs
- Determines Final ASU Design:
  Number of Trays / HX integration / P, T, Flows

ASU Optimization

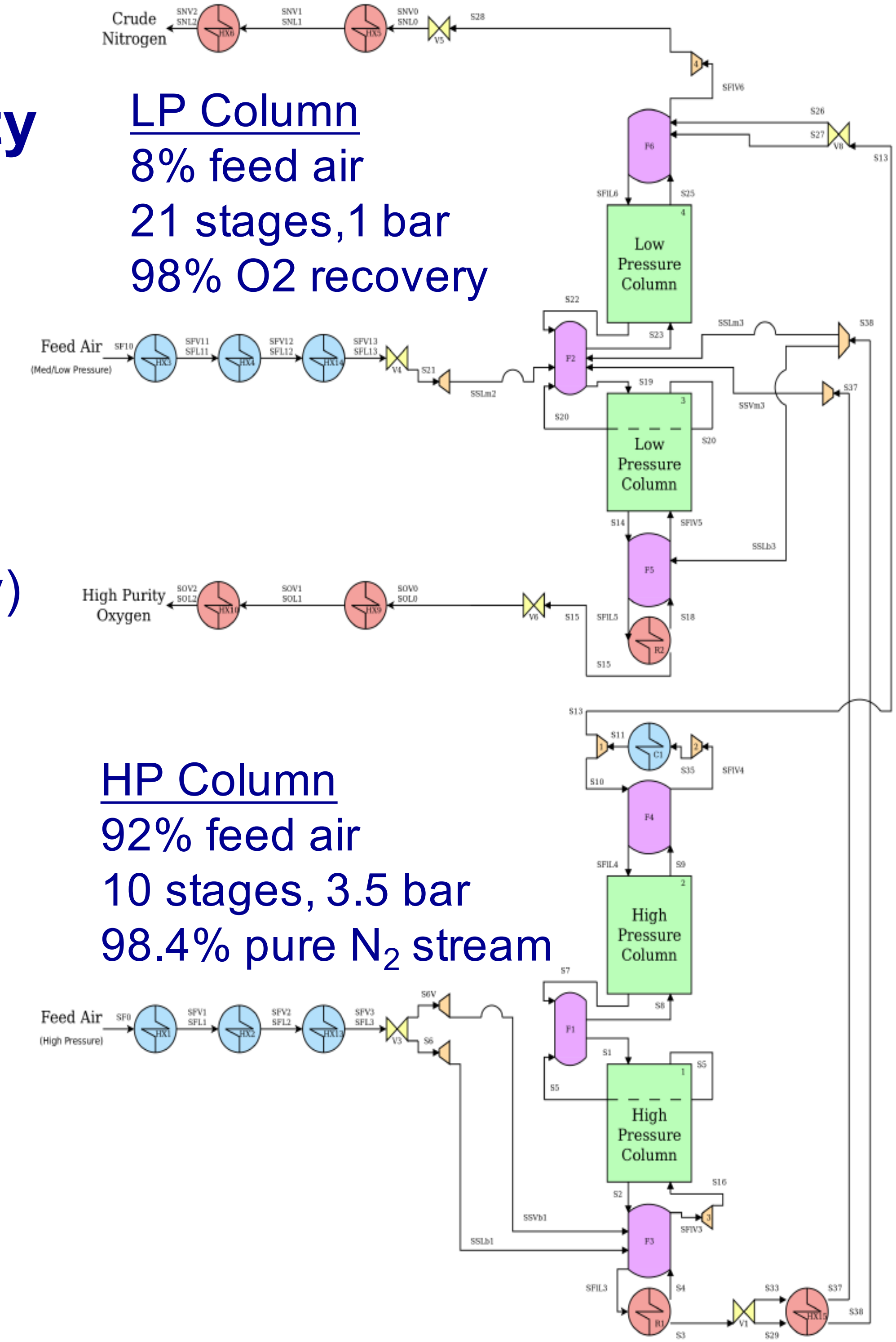
ΔT_{min} = 1.5 K, 95% O2 purity

- Balanced Reboiler/Condenser
- No heating and cooling, only power
- Typical NLP: 15534 variables, 261 degrees of freedom
- MPCC bootstrapping process ➔ 15 CPU min (CONOPT/ GAMS)
- 0.196 kWh/kg (86% comp efficiency)
- <u>Superior to Commercial Designs</u>



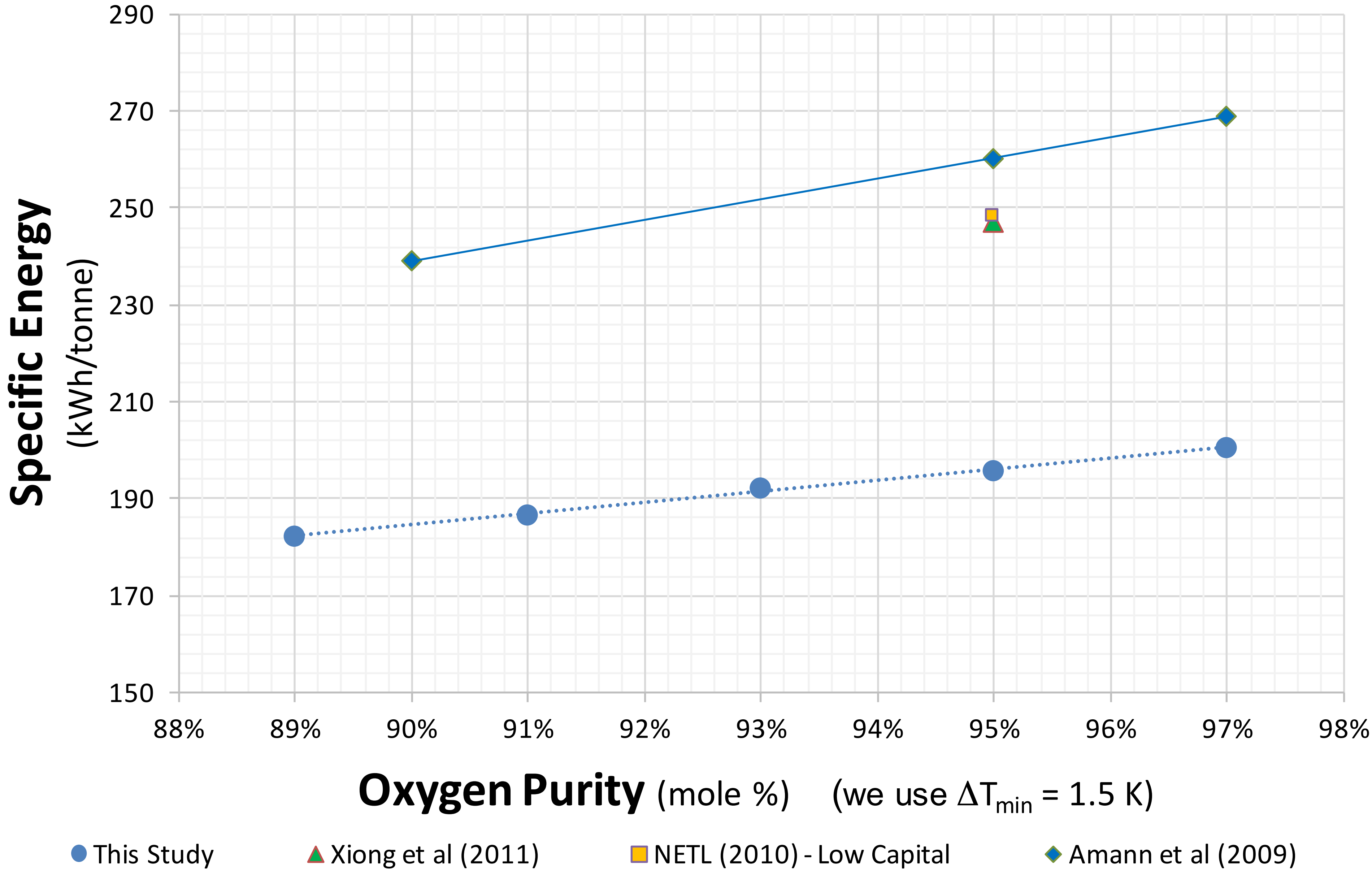
<u>LP Column</u>
8% feed air
21 stages,1 bar
98% O2 recovery

<u>HP Column</u>
92% feed air
10 stages, 3.5 bar
98.4% pure N_2 stream



Process Optimization Based on O_2 Purity

(~25% energy savings in literature comparison)



The Next 15 – 25 Years
Multi-level Optimization Models

Enabling Tools

- Complementarity modeling platforms
- Advanced algorithms for MPCCs, discrete optimization and global optimization
- Global optimization models distributed locally through multi-level optimization

Applications

- Discrete decisions in time and space through complementarities
- Phase changes and complex phenomena in pipelines, complex heat exchangers and reservoir models
- Enterprise-wide distributed Advanced Control, Real-time Optimization, Integration to Scheduling and Planning

Summary and Conclusions

- **Optimization is much more than glorified case studies on simulation models**
  - Exploit sparse, structured models
  - Embrace nonsmoothness, and physical transitions
  - Gain insights on optimal solution

- **Solving larger optimization problems is not more expensive**
  - Equation-based Modeling Utopia is close to reality
  - Fast NLP tools, sensitivity
  - Powerful optimization platforms, even for on-line optimization

- **Multilevel Optimization Extends Process Modeling**
  - Extend domain of solving problems
  - Systematic exploration of discontinuous phenomena

- **Huge Potential for Process Optimization Applications**
  - Ongoing work (EO-Process Modeling, Surrogate Optimization) within PyOMO Optimization Framework and NETL/Sandia/LBNL