The Tree-Sparse Interior Approach to Multistage Stochastic Programming

Marc C. Steinbach, ZIB, Berlin
http://www.zib.de/steinbach

2001-08-28

Outline

- Basic Ideas
- Generic Framework
- Specialization
- Computational Experience

Basic Ideas

Aim

Solve convex dynamic stochastic programs on scenario trees

Approach

Standard interior method combined with problem specific sparse linear algebra

Advantages

- good convergence (theory & practice)
- handles smooth convex problems directly
- extends to nonconvex problems (nonlinear IPM or SQP)

"Separation of Difficulties"

- interior method iterates on inequalities and nonlinearities
- linear algebra deals with sparsity

Basic Ideas

Interior Methods in Stochastic Programming

Birge, Qi (1988)
Lustig, Mulvey, Carpenter (1991)
Birge, Holmes (1992)
Jessup, Yang, Zenios (1994)
Bahn, du Merle, Goffin, Vial (1995)
Berger, Mulvey, Rothberg, Vanderbei (1995)
Czyzyk, Fourer, Mehrotra (1995)
Yang, Zenios (1997)

Schweitzer (1998)
Berkelaar, Dert, Oldenkamp, Zhang (1999)
Gondzio, Kouwenberg (2000)
Blomvall, Lindberg (2001)

Why Yet Another Approach?

- new concept for full exploitation of sparsity
- most approaches above are for two-stage SLP

Concept

Hierarchical Sparse Structure

- Generic block-sparse structure: induced by scenario tree
- Problem-specific substructure: sparsity inside blocks

Generic Tree-Sparse Structure: Modeling Level

- view as control problem yields complete classification
  - of problem types
  - of constraints within problem classes
- associated regularity conditions
- predetermined block pivot order

$\longrightarrow$ integrated natural modeling and solution framework

Problem-Specific Substructure: Implementation Level (later)

Generic Framework

Interior Approach

Convex Program

$$\min_y \phi(y) \qquad \text{s.t.} \qquad Ay = a, \quad y \in [b_l, b_u], \quad By \in [r_l, r_u]$$

Smooth cost $\phi \in C^2$, convex on polyhedron Y: $H \equiv \nabla^2 \phi \geq 0$

KKT Matrix

$$\begin{bmatrix} H + \Phi & A^* & B^* \\ A & & \\ B & & -\Psi^{-1} \end{bmatrix} \longrightarrow \begin{bmatrix} \bar{H} & A^* \\ A & 0 \end{bmatrix}$$

Here

$\Phi \geq 0$: barrier matrix for bounds

$\Psi \geq 0$: barrier matrix for ranges

$\bar{H} = H + \Phi + B^* \Psi B$ careful: sparsity?

Generic Framework

Basic Classification

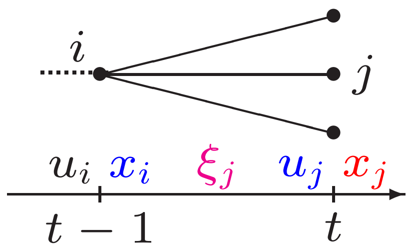
Dynamics: three basic forms, increasing generality

Explicit with Incoming Control

Transition equation: $x_j = G_j x_i + E_j u_j + h_j$

Interpretation: decision u_j effective immediately

Typical occurence: financial models

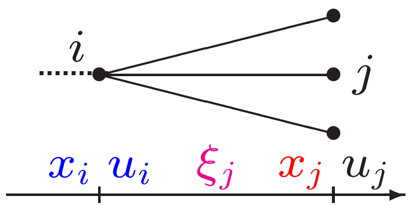


Explicit with Outgoing Control

Transition equation: $x_j = G_j x_i + E_j u_i + h_j$

Interpretation: decision u_j effective after delay

Typical occurrence: engineering models

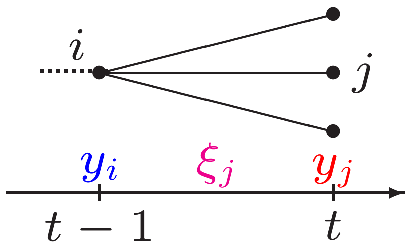


Implicit

Transition equation: $P_j y_j = G_j y_i + h_j$

No state/control distinction, no natural interpretation

Essentially the standard formulation: catchall



Problem Classes

Explicit Dynamics, Incoming Control

$$\min_{u,x} \quad \sum_{j\in V} p_j[\phi_{ij}(x_i, u_j) + \phi_j(x_j)] \qquad \text{(objective)}$$

$$\text{s.t.} \quad x_j = G_j x_i + E_j u_j + h_j \qquad \forall j \in V, \qquad \text{(dynamics)}$$

$$((x_i, u_j), x_j) \in Y_{ij} \times X_j \qquad \forall j \in V, \qquad \text{(local)}$$

$$\sum_{j\in V} p_j[D_j u_j + F_j x_j + e_j] = 0. \qquad \text{(global)}$$

- objective & global constraints: expectations
- objective & local constraints: coupling as in dynamics ($\longrightarrow$ structure of $\bar{H}$)
- local feasible sets polyhedral ($\longrightarrow$ subhierarchy of constraints)

Problem Classes

Explicit Dynamics, Outgoing Control

$$\min_{x,u} \quad \sum_{j \in V} p_j \phi_j(x_j, u_j) \qquad \text{(objective)}$$

$$\text{s.t.} \quad x_j = G_j x_i + E_j u_i + h_j \qquad \forall j \in V, \qquad \text{(dynamics)}$$

$$(x_j, u_j) \in Y_j \qquad \forall j \in V, \qquad \text{(local)}$$

$$\sum_{j \in V} p_j [F_j x_j + D_j u_j + e_j] = 0. \qquad \text{(global)}$$

- objective & global constraints: expectations
- objective & local constraints: coupling as in dynamics ($\longrightarrow$ structure of $\bar{H}$)
- local feasible sets polyhedral ($\longrightarrow$ subhierarchy of constraints)

Generic Framework

Problem Classes

Implicit Dynamics

$$
\begin{aligned}
\min_{y} \quad & \sum_{j \in V} p_j \phi_j(y_j) & & \text{(objective)} \\
\text{s.t.} \quad & P_j y_j = G_j y_i + h_j & \forall j \in V, & \quad \text{(dynamics)} \\
& y_j \in Y_j & \forall j \in V, & \quad \text{(local)} \\
& \sum_{j \in V} p_j [F_j y_j + e_j] = 0. & & \text{(global)}
\end{aligned}
$$

- objective & global constraints: expectations
- objective & local constraints: coupling as in dynamics ($\longrightarrow$ structure of $\bar{H}$)
- local feasible sets polyhedral ($\longrightarrow$ subhierarchy of constraints)

Generic Framework

Subhierarchies

Incoming Control

$$D_j^u u_j + e_j^u = 0 \qquad u_j \in [b_{lj}^u, b_{uj}^u]$$

$$F_j^x x_j + e_j^x = 0 \qquad x_j \in [b_{lj}^x, b_{uj}^x]$$

$$F_{ij}^c x_i + D_j^c u_j + e_j^c = 0 \qquad F_{ij}^r x_i + D_j^r u_j \in [r_{lj}^u, r_{uj}^u]$$

$$F_j^r x_j \in [r_{lj}^x, r_{uj}^x]$$

Outgoing Control

$$F_j^x x_j + e_j^x = 0 \qquad x_j \in [b_{lj}^x, b_{uj}^x]$$

$$D_j^u u_j + e_j^u = 0 \qquad u_j \in [b_{lj}^u, b_{uj}^u]$$

$$F_j^c x_j + D_j^c u_j + e_j^c = 0 \qquad F_j^r x_j + D_j^r u_j \in [r_{lj}, r_{uj}]$$

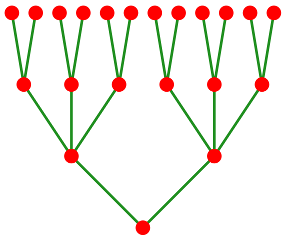
Implicit Dynamics

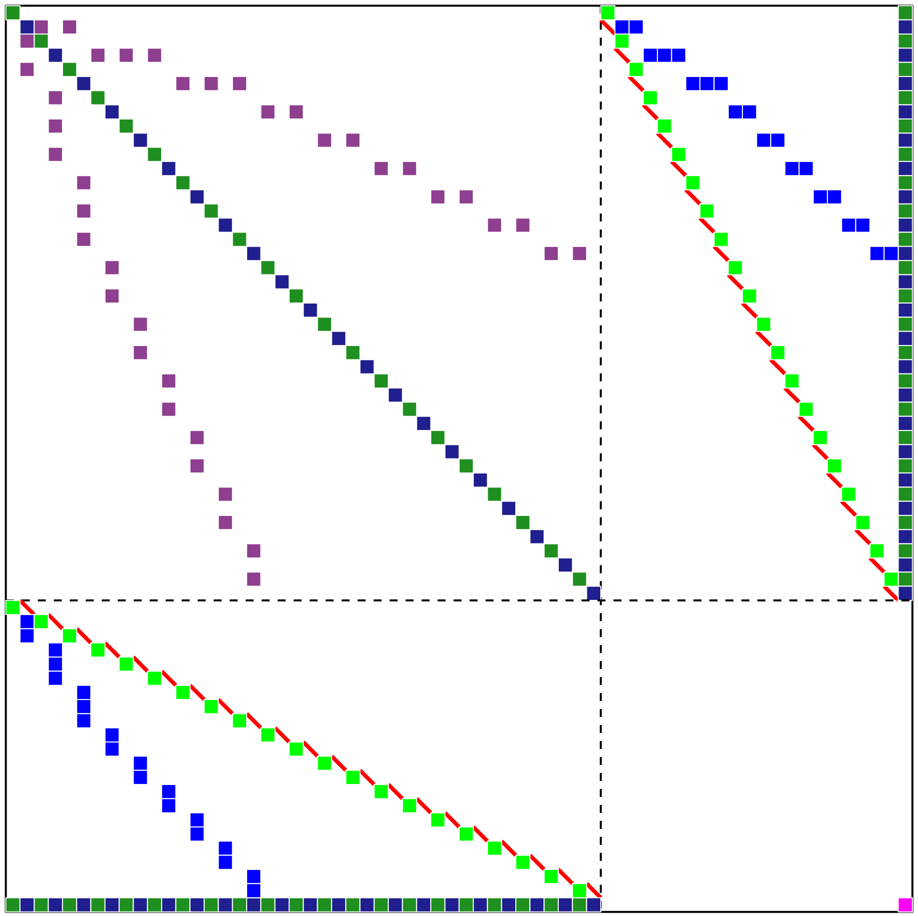
$$F_j^y y_j + e_j^y = 0 \qquad y_j \in [b_{lj}, b_{uj}] \qquad F_j^r y_j \in [r_{lj}, r_{uj}]$$

Generic Framework

Incoming Control: Sparsity

KKT matrix

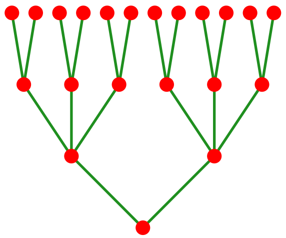


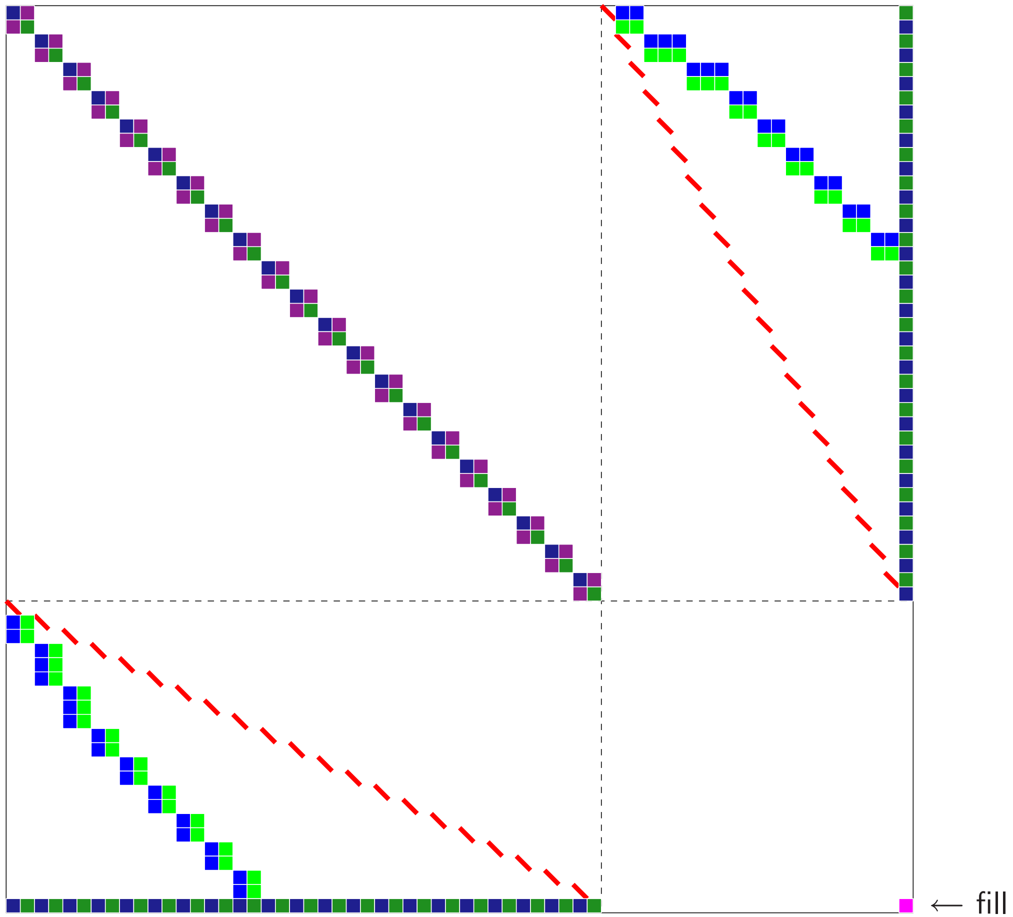


Generic Framework

Outgoing Control: Sparsity

KKT matrix

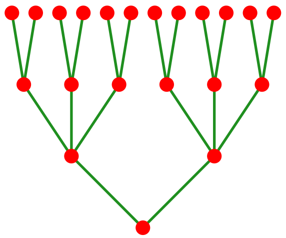


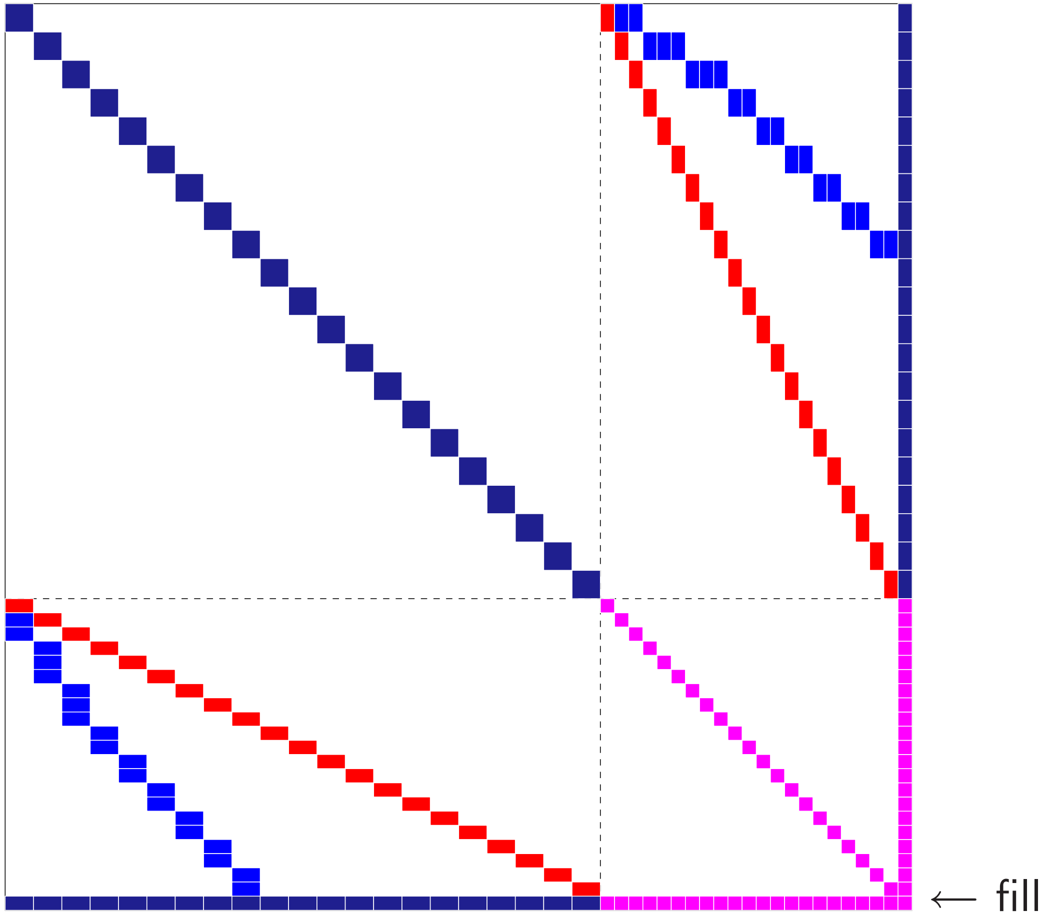


Generic Framework

Implicit Dynamics: Sparsity

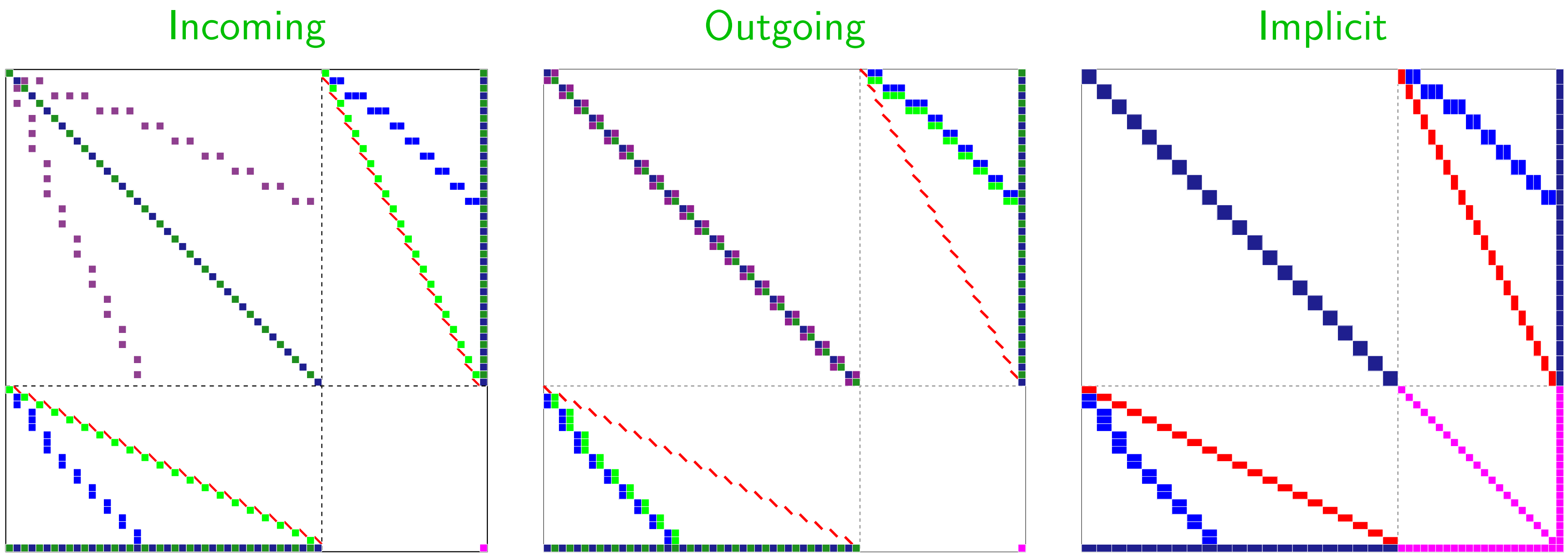
KKT matrix

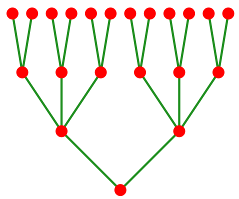




Generic Framework

Overview





Generic Framework

Algorithm

View as Control Problem

KKT system = linearized KKT conditions of barrier problem (Newton step)

= equality-constrained tree-sparse QP

= LQ control problem on tree + local/global constraints

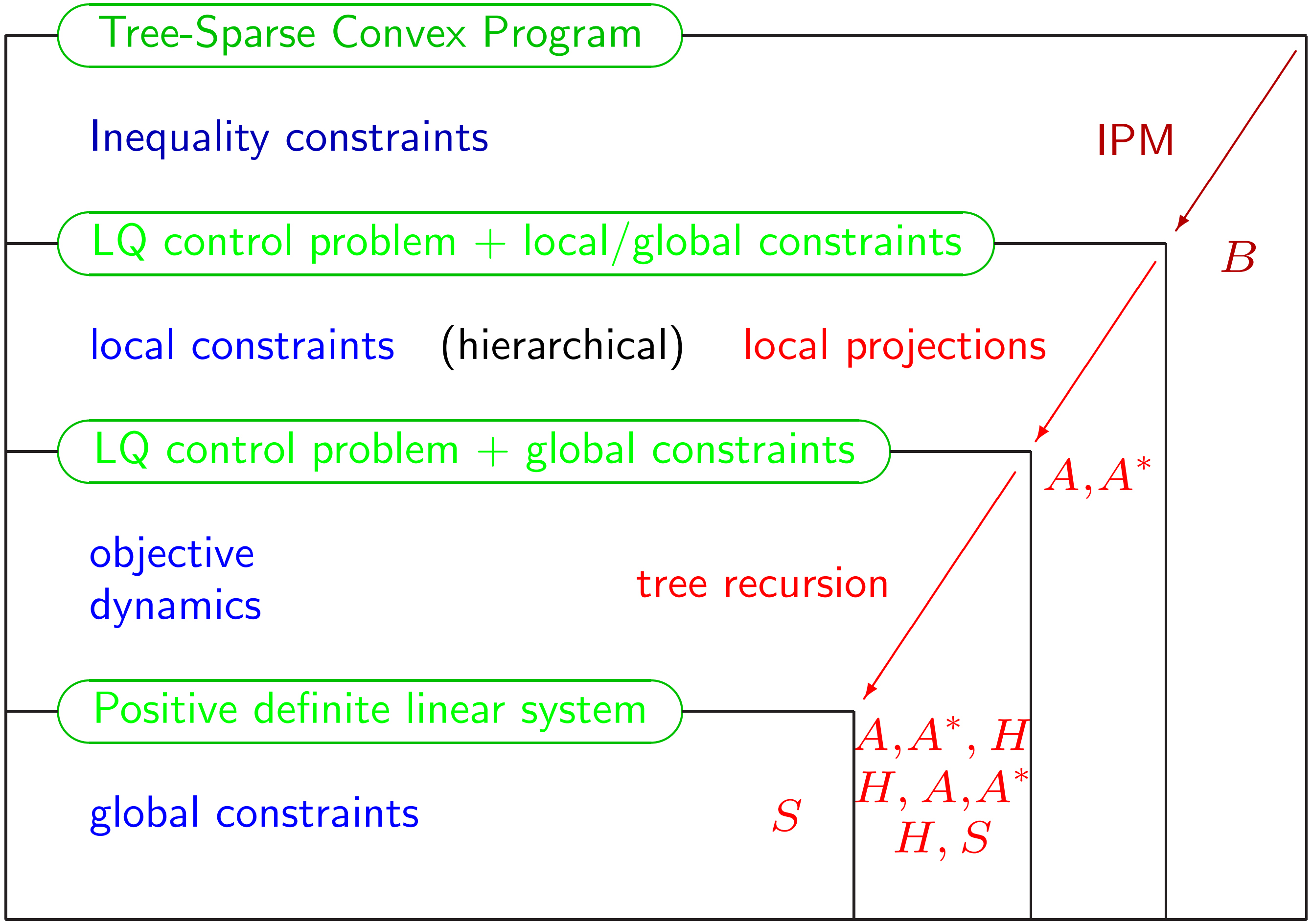
= Riccati equation on tree + local/global constraints

General Approach

- direct factorization of KKT matrix
- symmetric (primal/dual)
- identical operations in all nodes
- control-theoretic interpretation

Generic Framework

Algorithm



Generic Framework

Memory complexity: $O(n^2 N)$
Runtime complexity: $O(n^3 N)$

$$N = \text{tree size } (\# \text{ nodes})$$
$$n = \text{largest local dimension}$$

Summary

- conceptually clean and simple
- allows arbitrary constraints (local and global)
- efficient for dense or small blocks
- parallelizes perfectly ($\longrightarrow$ J. Blomvall: next lecture)
- equivalent: allows efficient usage of disk storage

Other Interior Approaches

Two-Stage Linear SP

$$\begin{aligned} \min_y \quad & f_0^* y_0 + \sum_{j=1}^N f_j^* y_j \\ \text{s.t.} \quad & P_0 y_0 = h_0, \quad y_0 \geq 0, \\ & P_j y_j = h_j + G_j y_0, \quad y_j \geq 0, \qquad j = 1, \ldots, N. \end{aligned}$$

$\longrightarrow$ Special case of implicit dynamics variant

Theorem. The Birge and Holmes (92) modification of the Birge and Qi (88) block-angular factorization for the two-stage SLP is equivalent to the implicit tree-sparse factorization applied to the (equivalent) problem with an additional quadratic penalty term in the root:

$$\min_y \frac{1}{2} \|P_0 y_0 - h_0\|_2^2 + f_0^* y_0 + \sum_{j=1}^N f_j^* y_j$$

Other Interior Approaches

Comparison Two-Stage Linear

authors	year	equiv	KKT	remarks
Birge, Qi	88	$\times$	normal	
Lustig, Mulvey, Carpenter	91	—	normal	OB1 + variable splitting
Birge, Holmes	92	$\times$	normal	
Jessup, Yang, Zenios	94	$\times$	normal	
Bahn, du Merle, Goffin, Vial	95	—	normal*	analytic ctr cutting plane
Czyzyk, Fourer, Mehrotra	95	$(\times)$	augm	
Yang, Zenios	97	$\times$	normal	
Berkelaar, Dert, Oldenkamp, Zhang	99	$(\times)$	augm*	homogeneous self-dual

Comparison Multistage Convex

authors	year	dyn	KKT	remarks
Berger, Mulvey, Rothberg, Vanderbei	95	impl*	augm	LOQO + tree dissection
Schweitzer	98	impl	augm	global Schur complement
Gondzio, Kouwenberg	00	—	—	nested Benders
Blomvall, Lindberg	01	out	augm	

Problem-Specific Sparsity

Generic Framework

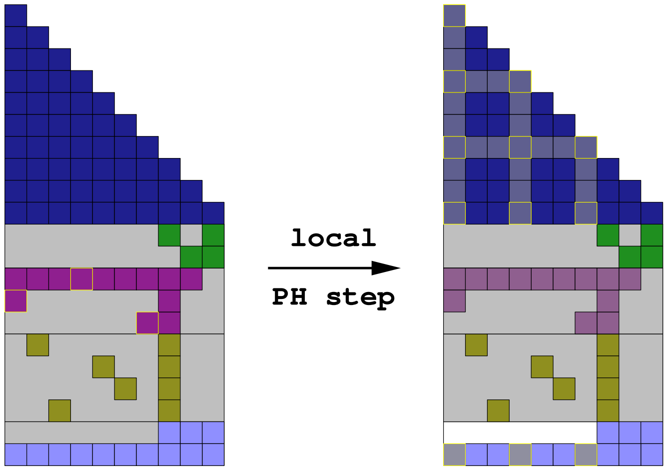
- determines block-level operations
- cannot address sparse substructure

Implementation of Tree-Sparse KKT Solver

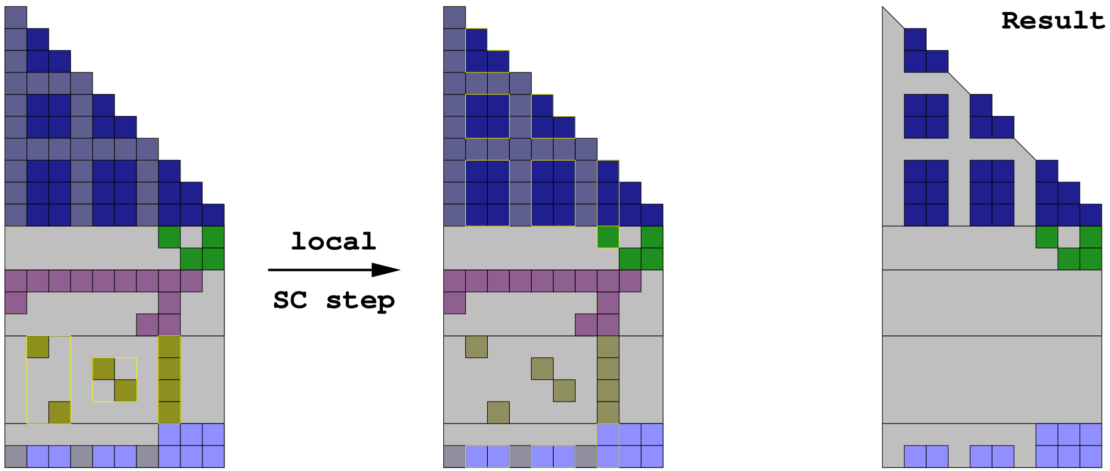
- dense implementation of general deterministic outgoing case (1994)
- generic implementation of basic tree-sparse variants (1997+)
  - tree with node pointers
  - nodes with pointers to dimension arrays and entry arrays
- dense storage/operations by default
- allows sparse data structures and node operations

Specialization

Example 1. Financial Model (Ainassaari, Kallio, Ranne 1997)

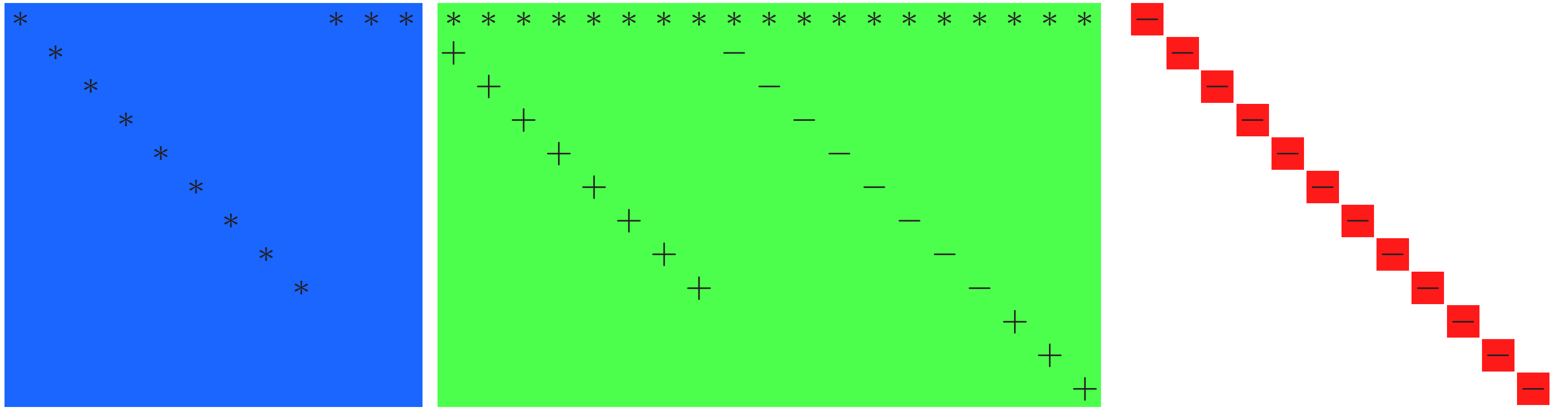


Savings:
Memory 50%
Operations 70%



Specialization

Example 2. DEVA (Frauendorfer 1995)



Savings:

Memory 35% for $n = 12$ (40% for $n \to \infty$)

Operations 60% for $n = 12$ (70% for $n \to \infty$)

Specialization

Problem-Specific Sparsity

Specialized Implementation

+ substantially better memory and runtime efficiency but
– time-consuming
– tedious
– error-prone
– especially for large systems

Work in Progress: Code Generator (A. Hutanu, July 2002)

- perform recursive fill analysis a priori (static)
- determine sparse data structures
- generate linear algebra operations
- currently semi-automatic: no pivoting

Computational Experience

Overview

Problems Considered

1. Simplified portfolio management problems
2. Dynamic Mean-Variance Analysis (DEVA, commercial model)
3. Tracking problem in process engineering
4. Energy minimization in distillation process (work in progress)

Computational Experience

Financial Application

Cooperation: K. Frauendorfer, Univ. St. Gallen (since 1997)

DEVA: Dynamic Mean-Variance Analysis (Frauendorfer 1995, F. & Siede 2000)

- Direct generalization of Markowitz approach to multiple periods
- Risk minimization for target wealth: efficient frontier

Model

- international portfolio: stocks & bonds, forward contracts, cash (riskless)
- transaction costs, cash flow, shortselling & market restrictions
- virtual reduction of investment $\longrightarrow$ approximate downside variance

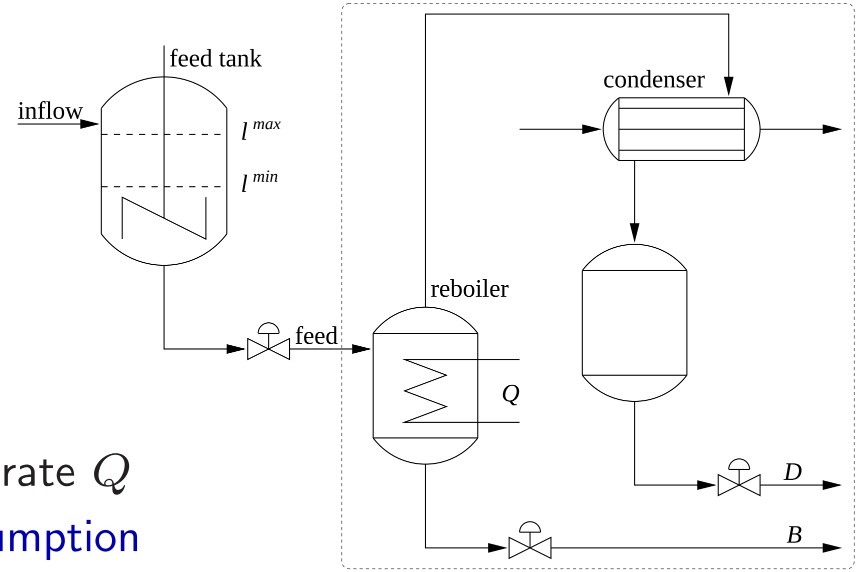
Swiss Insurance Company

- Quarterly reallocation of pension funds
- Test runs started Jan 1999; application since Oct 1999

Computational Experience

Process Engineering Application

Distillation process
with random inflow



Description

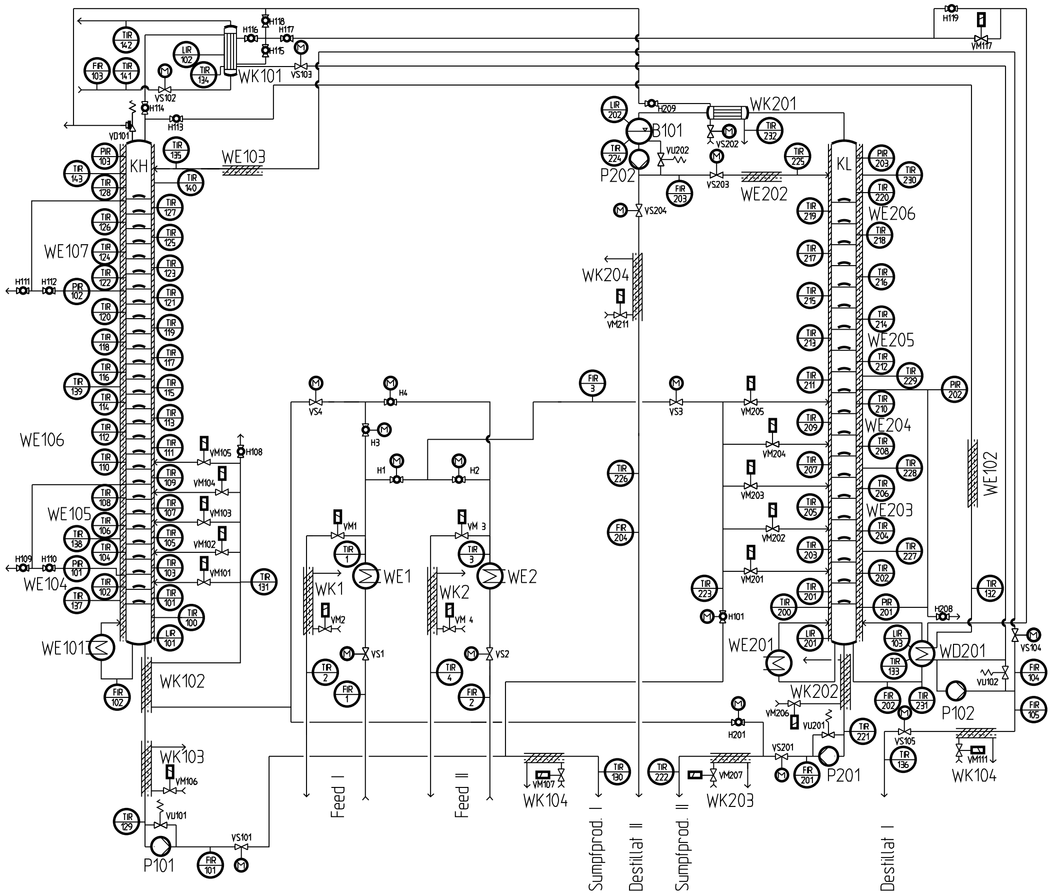
- Separation of methanol/water mixture
- Autocorrelated stochastic inflow rate
- Controls: feed extraction rate F, heating rate Q
- Goal: Minimize accumulated energy consumption over some planning horizon
- Deterministic constraints: feed tank must not run dry or spill over

Simplification

- Decouple system dynamics and stochasticity
  - Optimize nominal operation based on expected inflow (off-line)
  - Minimize expected tracking error w.r.t. nominal extraction (on-line)

Computational Experience

Pilot system at TU Berlin



Computational Experience

Work in Progress

Cooperation: R. Henrion (WIAS Berlin), G. Wozny (TU Berlin)
Support: DFG Research Center Real-Time Optimization of Large Systems

Integrated Model

- process dynamics and uncertainty
- process largely simplified: flash unit system

Nonlinear Dynamics

- differential-algebraic equations (DAE)
- time grid much too coarse for direct discretization

DAE System

Feed tank

$$\frac{d}{dt} M_F = \xi - F$$

Computational Experience

Reboiler

$$
\begin{aligned}
0 &= F - V_R - B \\
M_R^L \tfrac{d}{dt} X_{1R} &= F Z_1 - V_R Y_{1R} - B X_{1R} \\
M_R^L \tfrac{d}{dt} H_R^L &= F H_F - V_R H_R^V - B H_R^L + Q = M_R^L \Big(\frac{\partial H_R^L}{\partial X_{1R}} \tfrac{d}{dt} X_{1R} + \frac{\partial H_R^L}{\partial T_R} \tfrac{d}{dt} T_R \Big) \\
H_R^L &= X_{1R} h_1^L(T_R) + X_{2R} h_2^L(T_R) \\
H_R^V &= Y_{1R} h_1^V(T_R) + Y_{2R} h_2^V(T_R) \\
Y_{iR} &= \gamma_i(T_R, X_{1R}, X_{2R}) X_{iR} P_i^{\mathsf{vp}}(T_R)/P_R, \quad i = 1, 2 \\
1 &= X_{1R} + X_{2R}, \qquad 1 = Y_{1R} + Y_{2R}
\end{aligned}
$$

Total condenser

$$
\begin{aligned}
P_R - P_C &= \frac{1}{2} \zeta_C \left(\frac{Y_{1R} M_1 + Y_{2R} M_2}{Y_{1R} \bar{V}_1^V(T_R) + Y_{2R} \bar{V}_2^V(T_R)} \right) \left(\frac{Y_{1R} \bar{V}_1^V(T_R) + Y_{2R} \bar{V}_2^V(T_R)}{A_C^{FA}} V_R \right)^2 \\
0 &= V_R - D \\
M_C^L \tfrac{d}{dt} X_{1C} &= V_R Y_{1R} - D X_{1C}
\end{aligned}
$$

Computational Experience

Problem Overview

Problem	objective	constraints	dynamics	end cond.	remarks
Portfolio	quadratic	linear	in or impl	global	
DEVA	quadratic	linear	incoming	global	degenerate[1)]
Tracking	least squares	linear	outgoing	global or local	perfect recombination[2)]
Distillation	linear	nonlinear DAE	outgoing	global or local	perfect recombination[2)]

[1)] Optimal set is polyhedron of dimension $2n\,|V| - (n+1)\,|L|$
$n = \#$ assets, $m = \#$ forwards, total $\#$ variables $= (3n + 2m + 1)\,|V|$

[2)] Recombination leads to groups of equivalent scenarios
(with regular space discretization and depending on limits and terminal condition)

None of the problems has relatively complete recourse

Computational Experience

Computations

Problem	states controls node vars	nodes scenarios branches	rows columns dof	iter	CPU
DEVA	13	10687	138912	125	16:22
12 assets	24	10647	395419		
	37	39×273	256487		
DEVA	11	28348	311829	178	45:10
10 assets	20	28314	878788		
	31	33×858	566959		
Tracking	1	97656	97656	27	0:57
local	1	78125	117187	55	1:55
	2	5^7	19531		
Tracking	1	488281	488282	71	16:16
global	1	390625	585937	120	27:30
	2	5^8	97655		

THE END

You've gone too far!