Fast Algorithms for Nonlinear Multistage Stochastic Programs

Marc Steinbach
www.zib.de/steinbach

2005-10-20

Outline

• Introduction
• Algorithms
• Application
• Conclusions
Example 1: Distillation Process

Separation of methanol/water mixture with stochastic inflow

**Goal**
Tracking a desired (optimal) extraction profile without exceeding tank level bounds

**Basic Assumption**
Probabilistic model of inflow process at hand

Cooperation: G. Wozny (TU Berlin), R. Henrion (WIAS Berlin) - Support: DFG
Introduction

Example 1: Distillation Process

Scenario Tree Construction

\[ 0 \quad 1 \quad 2 \quad \cdots \quad T \]

\[ \text{max} \]

\[ \text{min} \]
Introduction

Example 1: Distillation Process

Scenario Tree Construction

distributions independent of decisions!
Introduction

Example 1: Distillation Process

Scenario Tree $\mathcal{T} = (V, E)$
Introduction

Example 1: Distillation Process

Scenario Tree $\mathcal{T} = (V, E)$: Notation

Node probability $p_j \quad \sum_{j \in L_t} p_j = 1$ for $t = 0, \ldots, T$
Introduction

Example 1: Distillation Process

Discrete Optimization Problem

$x, u$: Tank level, extraction

$\hat{u}$: Nominal extraction

$$\min_{(x,u)} \sum_{t=0}^{T-1} \sum_{j \in L_t} p_j |u_j - \hat{u}_t|^2$$

subject to

$$x_j = x_{\pi(j)} - u_{\pi(j)} + \xi_j \quad j \in V$$

+ terminal condition (stochastic)
+ simple bounds

Problem Size

$$\dim(x, u) \quad 2$$

$$|V| \quad 10^5 \ldots 10^6$$
Example 1: Integrated Process Dynamics

Differential variables

\[
x = \begin{pmatrix} M_F \\ X_{1R} \\ X_{1C} \\ T_R \end{pmatrix}
\]

- molar holdup in feed tank
- liquid mole fraction of methanol at reboiler
- liquid mole fraction of methanol at total condenser
- temperature at reboiler

Algebraic variables

\[
z = \begin{pmatrix} V_R \\ Y_{1R} \end{pmatrix}
\]

- molar vapor flowrate
- vapor mole fraction of methanol at reboiler

Control variables

\[
u = \begin{pmatrix} F \\ Q \end{pmatrix}
\]

- molar feed flowrate
- heating power at reboiler

Relaxed DAE-IVP for inconsistent iterates \((x_j, z_j)\) [Schulz, Bock, St. 1998]

\[
B(x) \dot{x} = f(x, z, u_i) + \xi_j e_1, \quad x(\tau_i) = x_i,
\]

\[
0 = g(x, z) - g(x_i, z_i) e^{-\beta(\tau - \tau_i)}, \quad z(\tau_i) = z_i.
\]

Problem Size

\[
\dim(x, z, u) = 8 \\
|V| = 10^5 \ldots 10^6
\]
Introduction

Example 1: Pilot System at TU Berlin
**Example 2: Portfolio Management**

**DEVA (Dynamic Expectation-Variance Analysis)**

- Portfolio with $n$ assets
- Invested amounts $x \in \mathbb{R}^n$, total wealth $e^*x$, $e = (1 \ldots 1)$
- Random returns $r \in \mathbb{R}^n$, new wealth $r^*x = e^* \text{Diag}(r)x$
- Investment horizon $[0, T + 1]$
- Rebalancing at $t = 1 \ldots T$
- DEVA: 
  - Minimize variance of total wealth ("risk")
  - subject to prescribed expectation at $T + 1$

**Cooperation:** K. Frauendorfer (ior/cf-HSG, U St. Gallen, Switzerland)

**Refs:** Frauendorfer 1995; St. 1998; Frauendorfer, Siede 2000; St. 2001
Introduction

Example 2: Portfolio Management

DEVA Base Model

\[
\begin{align*}
\min_x & \quad \sum_{j \in L} p_j x_j^* (\Sigma_j + \bar{r}_j \bar{r}_j^*) x_j \\
\text{subject to} & \quad e^* x_0 = 1 & \text{initial wealth} \\
& \quad e^* x_j = r^*_j x_{\pi(j)} & \text{rebalancing} \\
& \quad \sum_{j \in L} p_j \bar{r}_j^* x_j = \rho & \text{target wealth}
\end{align*}
\]

Problem Size

\[
\begin{array}{ll}
\text{dim}(x) & 10 \ldots 20 \\
|V| & 10^4 \ldots 10^5
\end{array}
\]

Algorithm

\[
\begin{array}{lll}
\text{IfU} & O(|V|^2) & O(|V|^3) \\
\text{ZIB} & O(|V|) & O(|V|)
\end{array}
\]

Algorithms

Problem Class: Overall Structure

Tree topology $T = (V, E)$

Dynamic Optimization Problem

$$\min_y \sum_{j \in V} \text{cost}_j(y_j)$$

subject to

$$\text{dyn}_j(y_{\pi(j)}, y_j) = 0 \quad j \in V$$

$$\sum_{j \in V} \text{glob}_j(y_j) = 0$$

$$B_j y_j \geq b_j \quad j \in V$$

Special Case: Control Problems

$x_j = \text{dyn}_j(x_{\pi(j)}, u_{\pi(j)})$ or $x_j = \text{dyn}_j(x_{\pi(j)}, u_j)$

Structure: Nonlinear coupling: father-son
Algorithms

Problem Class: Overall Structure

Tree topology \( T = (V, E) \)

Non-Markovian Dynamic Optimization Problem

\[
\min_y \sum_{j \in V} \text{cost}_j(y_0, \ldots, y_{\pi(j)}, y_j)
\]

subject to

\[
\text{dyn}_j(y_0, \ldots, y_{\pi(j)}, y_j) = 0 \quad j \in V
\]

\[
\sum_{j \in V} \text{glob}_j(y_0, \ldots, y_{\pi(j)}, y_j) = 0
\]

\[
B_j y_j \geq b_j \quad j \in V
\]

Special Case: Control Problems

\( x_j = \text{dyn}_j(x_0, u_0, \ldots, x_{\pi(j)}, u_{\pi(j)}) \) or \( x_j = \text{dyn}_j(u_0, x_0, \ldots, u_{\pi(j)}, x_{\pi(j)}, u_j) \)

Structure: Nonlinear coupling: all ancestors (path to root)
Algorithms

Overall Concept

Integrated Modeling and Solution Approach
- Tree-sparse NLP formulation: Exposing the rich structure
- Generic iterative algorithms: Preserving the structure
- Highly specialized linear algebra: Exploiting the structure

Generic Iterations
- SQP: Sequentiel Quadratic Programming
- IPM: Interior Point Method (primal-dual)
- Subproblem: Hierarchically structured KKT system
**Algorithms**

**KKT Solver**

**Algebraic Structure:** Tree-Sparse

- NLP formulation: *Hierarchical classification of restrictions*
- Generic iterations . . .
- Linear algebra: *Hierarchical structure of KKT matrix*
  1. Superstructure: primal-dual (optimization)
  2. Block structure – coarse: *tree topology* (discretization)
  3. Block structure – fine: *hierarchy of restrictions* (modeling)
  4. Sub-block structure: problem specific (*implementation*)

**Why classification of restrictions?**

- KKT factorization with linear complexity $O(|V|)$
- Most general regularity assumptions
- Result: three problem types + associated factorizations
Algorithms

Overview KKT coarse structure

Implicit
\[ \text{dyn}(y_{\pi(j)}, y_j) = 0 \]

Outgoing
\[ x_j = \text{dyn}(x_{\pi(j)}, u_{\pi(j)}) \]

Incoming
\[ x_j = \text{dyn}(x_{\pi(j)}, u_j) \]
Algorithms

Classification of Restrictions

General
Optimization Problem

- constraints
  - equalities
  - inequalities
   - bound
   - range
Algorithms

Classification of Restrictions

General Optimization Problem

Dynamic Optimization Problem

- **equalities**
  - dynamic
  - local
  - global

- **inequalities**
  - bound

- **constraints**
  - range
Algorithms

Earlier Solution Algorithms

Stochastic Optimization
Birge & Qi 1988, . . . (two-stage), Schweitzer 1998

Constraints

Equalities

Dynamic

Local

Global

Bound

State

Control

Inequalities

Range

State

Mixed
Algorithms

Fill-in for Example 1: standard form, Birge & Qi

Hessian

dynamics
DAE

global constr.
cycling cond.

node $i$ primal
node $j$ primal
node $j$ dual
global
Algorithms

Fill-in for Example 1: tree-sparse control form – outgoing with local constraints

Hessian

dynamics

diff. eqns

local constr.
alg. eqns

global constr.
cycling cond.
Algorithms

Fill-in for Example 1: comparison

* 8 periods, $5^7 \times 1 = 78,125$ scenarios, 175,781 nodes
  1,249,998 variables, 1,054,687 constraints
## Algorithms

### Hierarchic KKT Solver

#### Algorithmic Scheme

<table>
<thead>
<tr>
<th>Elimination</th>
<th>Support</th>
<th>Method</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>local constraints</td>
<td>single nodes</td>
<td>local projections</td>
<td>primal</td>
</tr>
<tr>
<td>dynamics &amp; objective</td>
<td>father &amp; son*</td>
<td>tree recursion</td>
<td>primal-dual</td>
</tr>
<tr>
<td>global constraints</td>
<td>arbitrary node set</td>
<td>Schur complement</td>
<td>dual</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lagrange relaxation</td>
<td></td>
</tr>
</tbody>
</table>

*possibly further ancestors
Algorithms

Control-Theoretic Interpretation

- Tree-sparse NLP
- Nonlinearities & Inequalities
- Tree-Sparse KKT System
- SQP or IPM

Algorithms

Control-Theoretic Interpretation

- Tree-sparse NLP
- Nonlinearities & Inequalities
- LQ Regulator Problem + Local/Global Constraints
  - Local Constraints
  - Local Projections
- LQ Regulator Problem + Global Constraints
  - Objective
  - Dynamics
  - Tree Rescursion
- Negative Definite Linear System
  - Global Constraints
  - Direct

SQP or IPM

\[ \begin{align*}
\min_{x \in \mathbb{R}^n} & \quad \frac{1}{2} \|Ax - b\|^2 \\
\text{s.t.} & \quad \Phi x = \phi, \quad B x = \beta, \quad C x = \gamma
\end{align*} \]
How to exploit block substructure?

- No implementation efficient on all problems
- Algorithm manually adaptable for specific problems
- Straightforward, but tedious & error-prone

→ Impractical: need software tool for custom code

Required Functionality

\[
\{ \text{access to operations on} \} \quad \{ \text{vector matrix inverse} \}
\]

Realization

- No local projections yet
Example: entry types in DEVA

±  value ±1 in every node and every problem instance
+  identical value in every node
*  different value in every node
## Finance Problems

<table>
<thead>
<tr>
<th>Problem</th>
<th>class</th>
<th>objective</th>
<th>constr.</th>
<th>dynamics</th>
<th>final cond.</th>
<th>granularity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stock/bond</td>
<td>LP</td>
<td>linear</td>
<td>linear</td>
<td>implicit</td>
<td>—</td>
<td>very fine</td>
</tr>
<tr>
<td>Asset mgmt</td>
<td>CP</td>
<td>convex</td>
<td>linear</td>
<td>implicit</td>
<td>—</td>
<td>fine</td>
</tr>
<tr>
<td>DEVA</td>
<td>QP</td>
<td>quadratic</td>
<td>linear</td>
<td>incoming</td>
<td>global</td>
<td>medium</td>
</tr>
</tbody>
</table>

## Chemical Engineering Problems

<table>
<thead>
<tr>
<th>Problem</th>
<th>class</th>
<th>objective</th>
<th>constr.</th>
<th>dynamics</th>
<th>final cond.</th>
<th>granularity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reactor</td>
<td>NLP</td>
<td>linear</td>
<td>ODE</td>
<td>outgoing</td>
<td>—</td>
<td>very fine</td>
</tr>
<tr>
<td>Tracking</td>
<td>QP</td>
<td>least squares</td>
<td>linear</td>
<td>outgoing</td>
<td>global/local</td>
<td>very fine</td>
</tr>
<tr>
<td>Distillation</td>
<td>NLP</td>
<td>linear</td>
<td>DAE</td>
<td>outgoing</td>
<td>global/local</td>
<td>fine</td>
</tr>
</tbody>
</table>
* No dense version available; CPLEX presolve eliminates all rows + columns
Applications

Portfolio Management

How it all began: single-period model (Markowitz 1952: *Portfolio Selection* )

\[
\begin{align*}
\text{min } & \quad x^* \Sigma x \\
\text{s.t.} & \quad e^* x = 1 \\
& \quad \tilde{r}^* x = \rho \\
& \quad x \geq 0
\end{align*}
\]

\[
\begin{align*}
\text{min } & \quad \sum_{j \in L} p_j x^*_j (\Sigma_j + \tilde{r}_j \tilde{r}^*_j) x_j \\
\text{s.t.} & \quad e^* x_0 = 1 \\
& \quad e^* x_j = r^*_j x_{\pi(j)} \quad j \in V \setminus \{0\} \\
& \quad \sum_{j \in L} p_j \tilde{r}_j^* x_j = \rho
\end{align*}
\]
Applications

Portfolio Management

**DEVA Application Model** (+ riskless asset, cash flow, transaction costs, . . . )

\[
\min_{(u,x)} \sum_{j \in L} p_j \left( \frac{x_j^c}{x_j} \right) = \left( \frac{(r_{T+1}^c)^2}{r_{T+1}^c} \right) \left( \frac{r_{T+1}^c}{\bar{r}_j} \sum_j + \bar{r}_j \bar{r}_j^* \right) \left( \frac{x_j^c}{x_j} \right)
\]

unter

\[
x_j^c = r_t^c x_{\pi(j)}^c - (e + c)^* v_j + (e - d)^* w_j + \phi_t
\]

\[
x_j = \text{Diag}(r_j)x_{\pi(j)} + v_j - w_j
\]

\[
\sum_{j \in L} p_j (r_{T+1}^c x_j^c + \bar{r}_j^* x_j) = \rho
\]

\[
u_j \geq 0, \quad x_j \in [x_j^{\min}, x_j^{\max}], \quad B_j x_j \geq 0
\]

**Special Property**

- Investment reduction below 100% possible
- Equivalent: 100% investment with lower semi-variance as risk measure
Applications

Application Problem

Strategic Portfolio Management (Pension Fund)
- Quarterly rebalancing
- Tests 01/1999; routine application 10/1999; commercial software with GUI 2003

Portfolio (12 assets + cash)
- 4 bonds: CHF, EUR, USD, GBP
- 8 stocks: Switzerland, France, Germany, UK, Netherlands, North America, Japan, Emerging Markets

Typical Problem
- 27,225 scenarios, 767,000 variables, 280 MB
- PC 2.4 GHz: 2:00 min, 69 iterations

Large Problem
- 53,235 scenarios, 1,971,000 variables, 854 MB
- PC 2.4 GHz: 16:20 min, 187 iterations
Applications

Application Problem

Strategic Portfolio Management (Pension Fund)

- Quarterly rebalancing
- Tests 01/1999; routine application 10/1999; commercial software with GUI 2003

Portfolio (12 assets + cash)

- 4 bonds: CHF, EUR, USD, GBP
- 8 stocks: Switzerland, France, Germany, UK, Netherlands, North America, Japan, Emerging Markets

Typical Problem

- 27,225 scenarios, 767,000 variables, 280 MB
- PC 2.4 GHz: 2:00 min, 69 iterations (highly degenerate solutions!)

Large Problem

- 53,235 scenarios, 1,971,000 variables, 854 MB
- PC 2.4 GHz: 16:20 min, 187 iterations (highly degenerate solutions!)
Applications

Current Work

DREWAG Model

Refs: Eichhorn, Römisch, Wegner-Specht 2005 (2×)

- Mean-risk optimization of electricity portfolios
- Municipal power utility (price taker)
- Combined heat and power production facility
- Electricity futures, spot market, supply contracts (EEX)
- Uncertain electricity & heat demands, spot & future prices
- Horizon = 1 year with hourly discretization
- Polyhedral multiperiod risk measures
- Four scenario trees: input tree, future tree, trading day tree, contract tree
  various trading windows and delays of decisions
  (day ahead trading . . . ⇒ nonstandard nonanticipativity constraints)
Applications

Current Work

Upcoming Project: VKW Model
Refs: Steinberger 2004; Steinberger, Zinner 2004

- Related planning problem of Vorarlberger Kraftwerke, Austria
- Hydroelectric plants and pump storage plants
- Production = 11% of demand, trading = 89% (Illwerke, ENBW, EEX)
- Horizon 1–2 years, discretization 1 hour or 15 minutes
- Deterministic LP model: up to 2 GB under CPLEX
- Crucial in SP algorithm:
  Data management, storage efficiency, structure exploitation
- Theoretical framework:
  Generalized nonanticipativity & embedding into single scenario tree [St. 2001]
Conclusions

- Robust and efficient solution method for nonlinear stochastic programs
  - global scenario coupling (→ mean-risk)
  - arbitrary nonanticipativity constraints
- Natural classification of restrictions
- Natural block level KKT factorization
  - Fixed elimination scheme (block pivots)
  - Linear complexity (storage, runtime)
- Exploitation of problem specific sub-structure: implementation
  - Static fill-in analysis
  - Automatic generation of custom sparse solvers (Huțanu 2002)
- Strategic Portfolio Management
  - Custom code with full structure exploitation
  - Practical application since 1999
  - Commercial software with GUI since 2003
Conclusions

KKT Matrix Factorization

Fine Block Structure
Conclusions

KKT Matrix Factorization

Fine Block Structure
Conclusions

KKT Matrix Factorization

Projection: Control Constraints (children)
Conclusions

KKT Matrix Factorization

Projection: State Constraints (children)
Conclusions

KKT Matrix Factorization

Projection: Combined Mixed Constraints (children)
Conclusions

KKT Matrix Factorization

Projection: Control Constraints (parent)
Conclusions

KKT Matrix Factorization

Projection: Combined State Constraints (parent)
Conclusions

KKT Matrix Factorization

Projection: Combined Mixed Constraints (parent)
Conclusions

KKT Matrix Factorization

Projection: Dynamics (children)
Conclusions

KKT Matrix Factorization

Recursion: Control (children)
Conclusions

KKT Matrix Factorization

DP Recursion: Dynamics (parent)
Conclusions

KKT Matrix Factorization

DP Recursion: Control (parent)