NTT DATA Mathematical Systems Inc.

Implementation issues of Interior-Point Method for real-world NLP problems

29 Mar. 2019
NTT DATA Mathematical Systems, Inc.
Takahito Tanabe (tanabe@msi.co.jp)

Trusted Global Innovator
NTT DATA Group

Our Company Overview

Our Company Profile

We call ourselves
MSI

Name	NTT DATA Mathematical Systems Inc.
Office location	Shinanomachi, Shinjuku-ku, Tokyo
History	**Founded in 1982** as Mathematical Systems Inc. Joined NTT DATA group in February, 2012 Changed name to NTT DATA Mathematical Systems Inc. in September, 2013
Common Stock	56 million yen
Financial Information	Net Sales ： 1485 million yen Ordinary Income ： 150 million yen (April 1,2017 to March 31,2018)
Number of Employees	**110** (as of April 1, 2018)
Business	Packaged software development and sales Analysis and Consulting services Entrusted development of software

Technical staff : about 87

Background
-Scientists ： 65%
-Engineers ： 10%

Degree
-Master ： 67%
-Ph.D. ： 14%

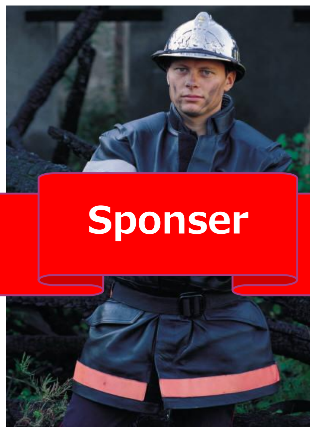
Our Standpoint

Solve real world problem for business practitioners

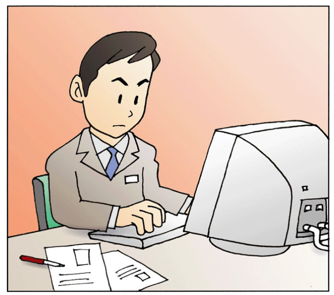
⇒ Our packaged softwares are 'stock in trade' for this purpose

Nuorium Optimizer (NuOpt) is one of them

Business Practitioners	**MSI**	**Academia**



Sponser



Business requirements ⟷ software / mathematical models ⟷ algorithms / theories

Our mission and solutions

Our Mission

Solve real-world problems using mathematical engineering and computer science

Our Solutions

- Data Mining / Machine Learning
- Text Mining
- Mathematical Optimization
- Technical and Scientific Computing
- Simulations

Our solutions and application

Our Solution

Data Mining / Machine Learning
- demand forecast
- image classification
- outliner analysis
- data-fusion
- Bayesian network
- recommendation
- sparse modeling

Text Mining
- call center log analysis
- patent mining
- nurse's record analysis
- text classification
- chat-bot

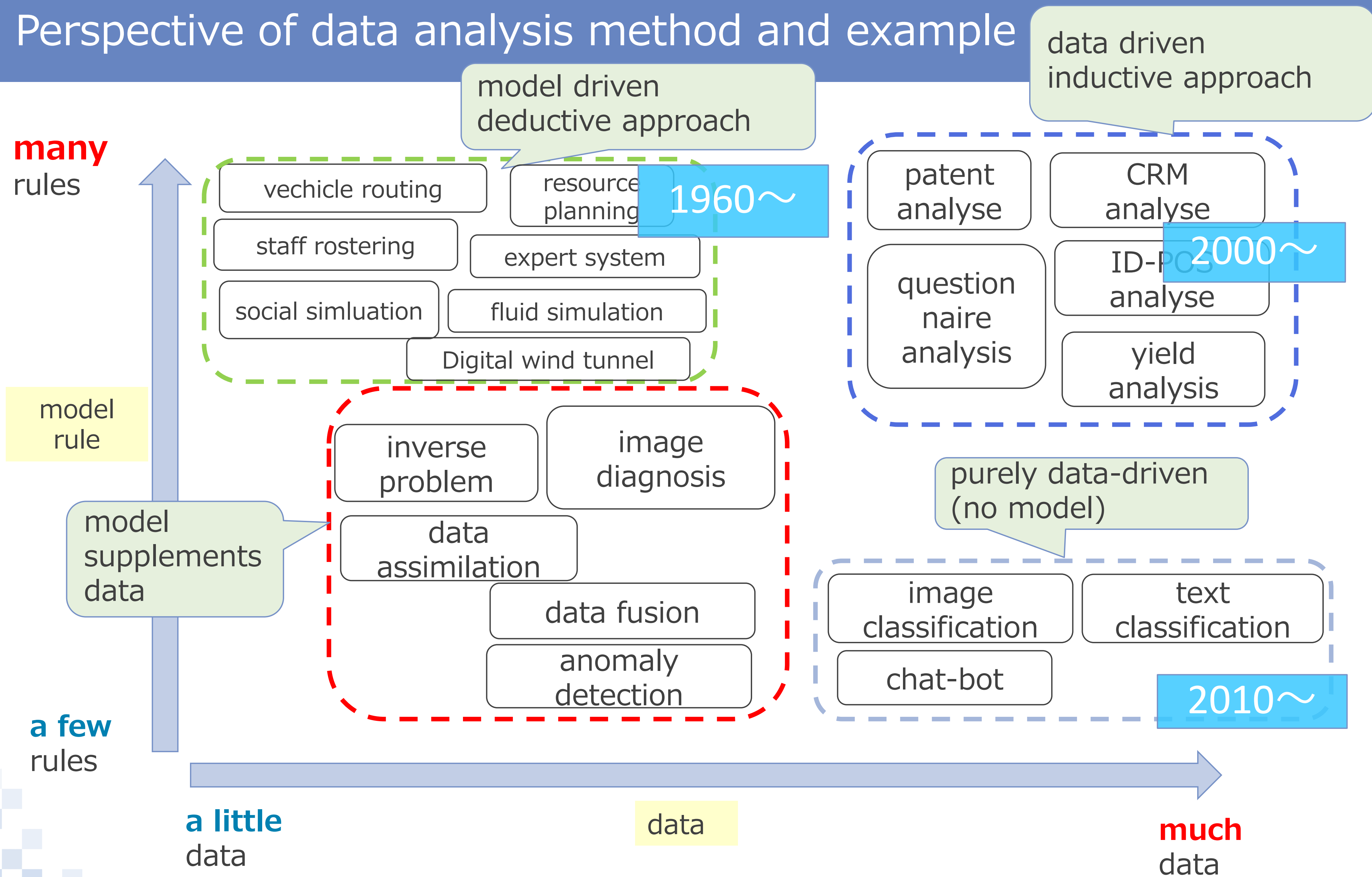
Numerical Optimization
- resource management
- financial engineering
- production scheduling
- staff rostering
- logistics optimization
- resource planning
- energy management

¼ of total sales

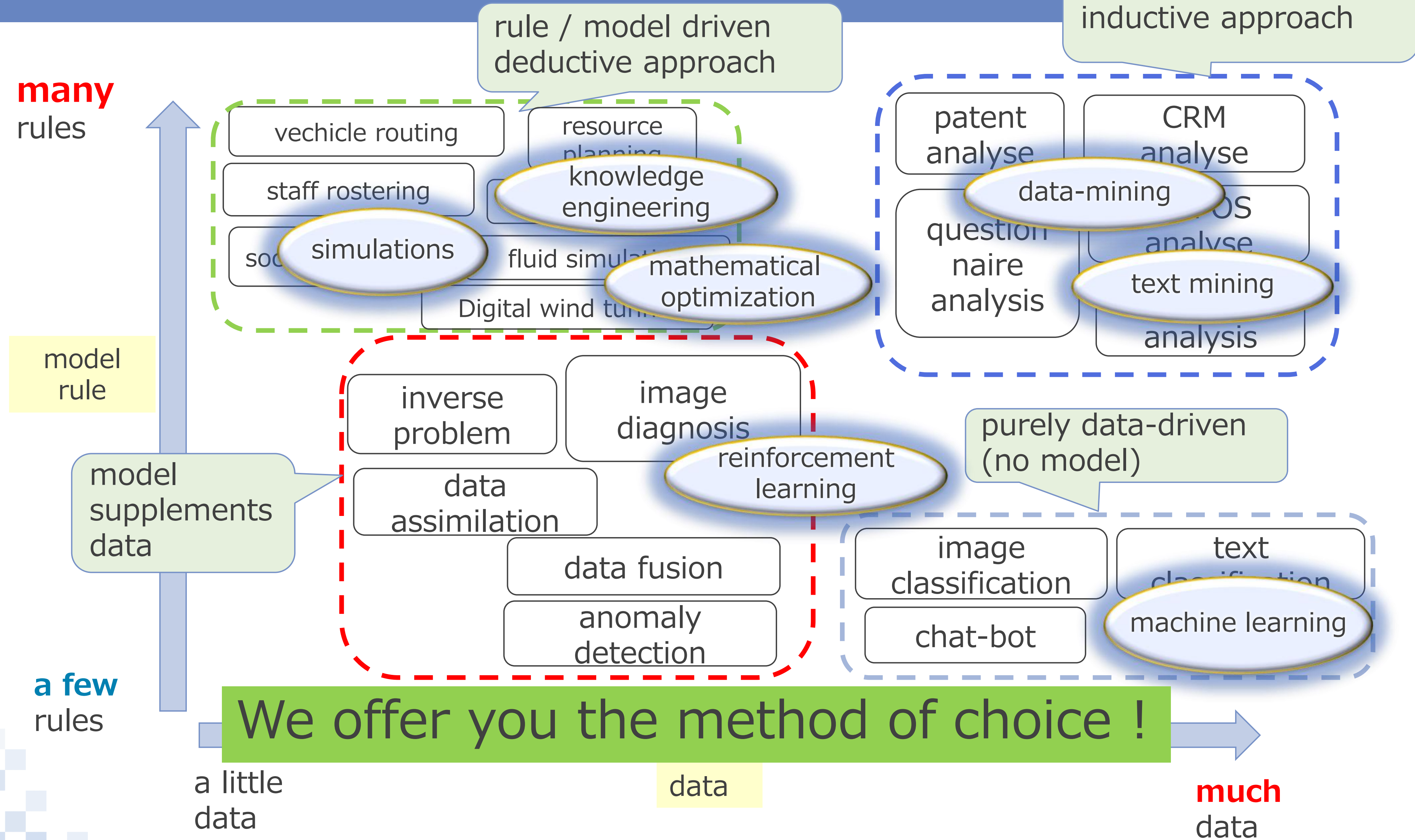
Technical and Scientific Computing
- inverse problem analysis
- semantic web analysis
- computational geometry
- image processing
- reverse engineering

Simulations
- agent simulation
- social system simulation
- traffic simulation
- facility management
- montecarlo simulation

Perspective of data analysis method and example



Perspective of data analysis method and example



Implementation of IPM for Nonlinear Programming

Implementation of IPM for NLP in NUOPT

- Solving Barrier KKT by Newton method

$$\text{minimize} \quad f(x), \quad x \in \mathbf{R}^n,$$
$$\text{s. t.} \quad g_E(x) = 0, \quad g_E(x) \in \mathbf{R}^{m_E},$$
$$g_I(x) \ge 0, \quad g_I(x) \in \mathbf{R}^{m_I},$$
$$x \ge 0$$

$$g_E(x) = 0,$$
$$g_I(x) - s = 0,$$
$$\nabla f(x) - A(x)^t y - z = 0,$$
$$y - w = 0,$$
$$Xz = \mu,$$
$$Sw = \mu,$$
$$x \ge 0, s \ge 0, z \ge 0, w \ge 0$$

$$A(x) \equiv \begin{bmatrix} A_E \\ A_I \end{bmatrix} \equiv \begin{bmatrix} \nabla g_E(x) \\ \nabla g_I(x) \end{bmatrix}$$

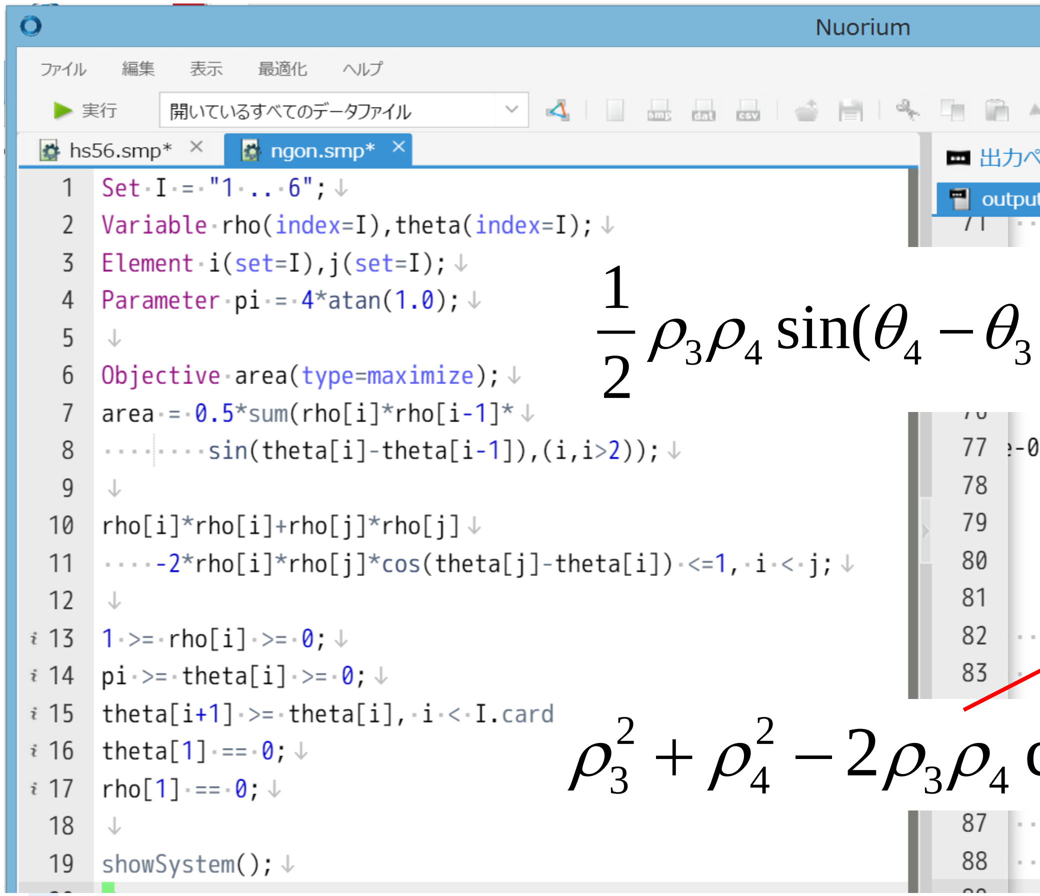
- Supervised by Merit Function (to ensure global convergence)

$$F(x,\mu) = f(x) - \mu \sum \log(x_i) + \rho \sum |g_i(x)|$$

(Yamashita 1992)

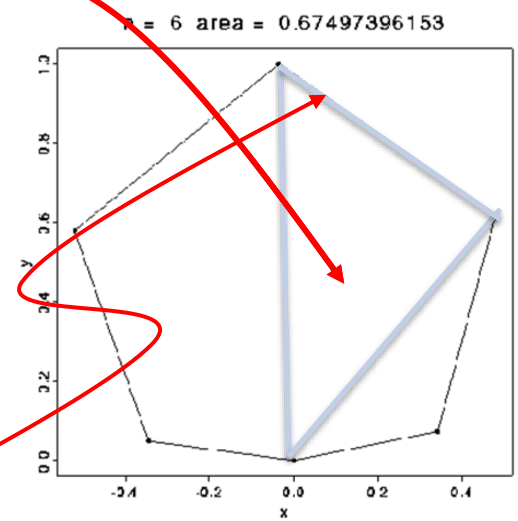
My contribution to NUOPT

- Sparse Direct Solver
- Algebraic Modeling Language (SIMPLE) with Automatic Differentiation Feature



Maximize area of hexagon with edge length <= 1

$$\frac{1}{2}\rho_3\rho_4\sin(\theta_4-\theta_3)$$



$$\rho_3^2+\rho_4^2-2\rho_3\rho_4\cos(\theta_4-\theta_3)$$

NLP test problems

- Hock & Schttkowski 41

$$\begin{aligned} &\text{mininimize} \quad && 2 - x_1 x_2 x_3 \\ &s.t. && x_1 + 2x_2 + 2x_3 - x_4 = 0 \\ & && 0 \le x_i \le 1, \quad (i = 1, \cdots, 3) \\ & && 0 \le x_4 \le 2 \end{aligned}$$

- Hock & Schttkowski 45

$$\begin{aligned} &\text{mininimize} \quad && 2 - \frac{1}{120} x_1 x_2 x_3 x_4 x_5 \\ &s.t. && 0 \le x_i \le i, \quad (i = 1, \cdots, 5) \end{aligned}$$

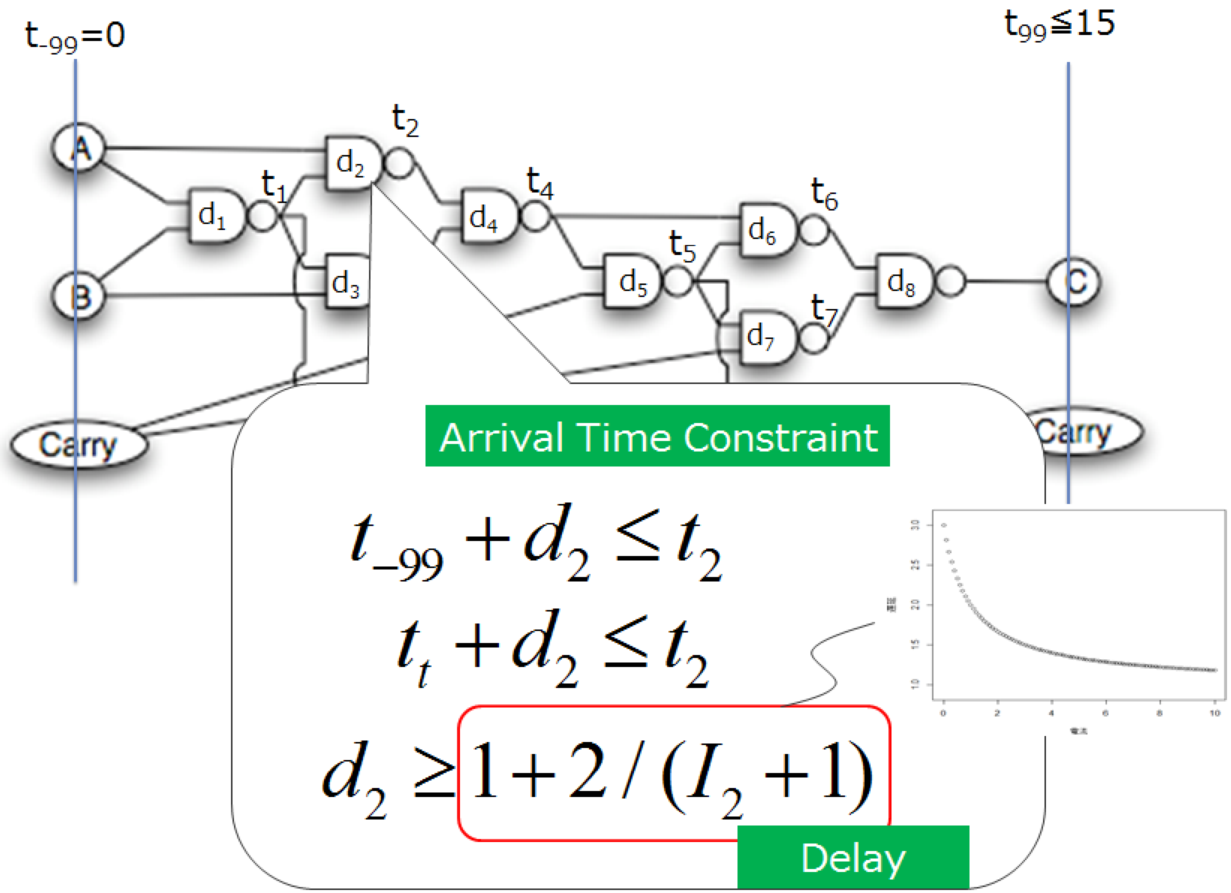
Initial Goal

Our Design Policy

- Stay general as possible.
- Achieve adequate performance for LP,QP

- Large Convex Optimization （n～20000）

Circuit Optimization (1993-)



$$t_{-99} + d_2 \le t_2$$

$$t_t + d_2 \le t_2$$

$$d_2 \ge 1 + 2/(I_2+1)$$

Real World NLP #2

- Nonlinear Mixing Problem (n～5000)

Nonlinear Blend Problem (1999-)

Real World NLP #3

- Pooling Problem（n～20000）

LNG plant scheduling at Osaka Gas Company (2008-)

- Determine LNG plant operation schedule (MINLP) (30days, daily, 18tanks)
- LNG density control is CRUCIAL

Volume conservation (tank i):

$$v_i^{t+1} - v_i^t = S^t \cdot \delta_i^t - \sum_{j \in N(i)} m_{i \to j}^t + \sum_{j \in N(i)} m_{j \to i}^t - D \cdot p_i^t$$

import ‘S’ unit of LNG or not (binary)

amount of LNG move (continuous)

output ‘D’ unit of LNG or not (binary)

Weight conservation (tank i):

$$q_i^{t+1} v_i^{t+1} - q_i^t v_i^t = Q^t S^t \cdot \delta_i^t - \sum_{j \in N(i)} q_i^t m_{i \to j}^t + \sum_{j \in N(i)} q_j^t m_{j \to i}^t - q^t D \cdot p_i^t$$

density of LNG in tank

density of imported LNG

Our Design Policy and Real World NLP

- Stay general as possible.
- Achieve adequate performance for LP,QP

⇒ **Proved Right ! because**

- Real World NLP is
  - Large and Sparse
  - Contains many linear constraints
  - Not too complicated ('bi-linear' is typical)
  - Hessian Matrix is mostly Diagonal (a few cross-term)

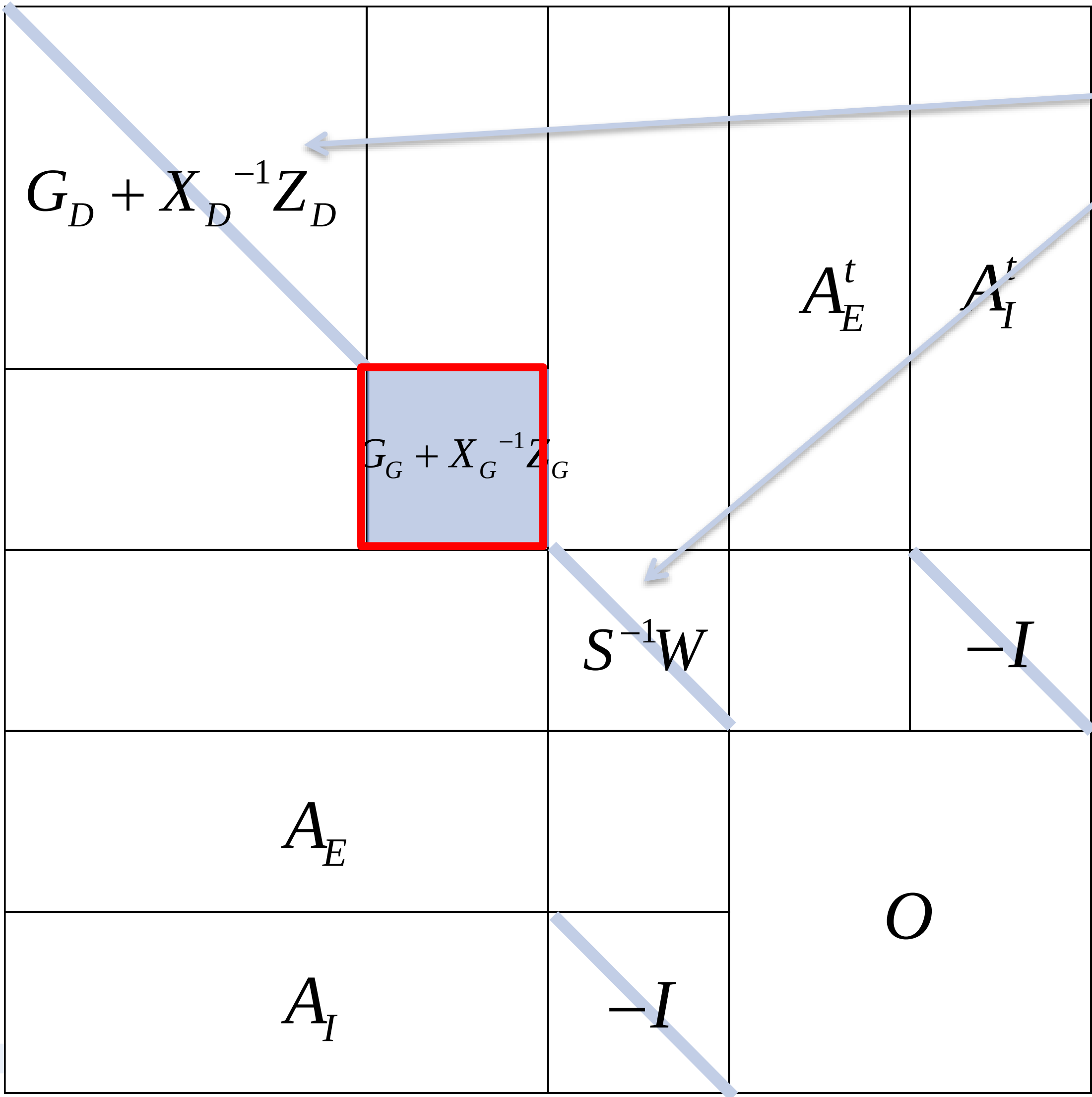
LP/QP like !

Direct Solver for IPM

$$G \equiv \nabla_x^2 f - \sum_i y_i \nabla_x^2 g_i$$

$$
\begin{array}{llll}
\text{minimize} & f(x), & & x \in \mathbf{R}^n, \\
\text{s. t.} & g_E(x) = 0, & & g_E(x) \in \mathbf{R}^{m_E}, \\
 & g_I(x) \geq 0, & & g_I(x) \in \mathbf{R}^{m_I}, \\
 & x \geq 0 & &
\end{array}
$$

$$
\begin{pmatrix}
G + X^{-1}Z & & A_E^t & A_I^t \\
 & S^{-1}W & & -I \\
A_E & & O & \\
A_I & -I & &
\end{pmatrix}
\begin{pmatrix}
\Delta x \\ \Delta s \\ \Delta y_E \\ \Delta y_I
\end{pmatrix}
= b
$$



1. Pivot Upper left Diagonal Part

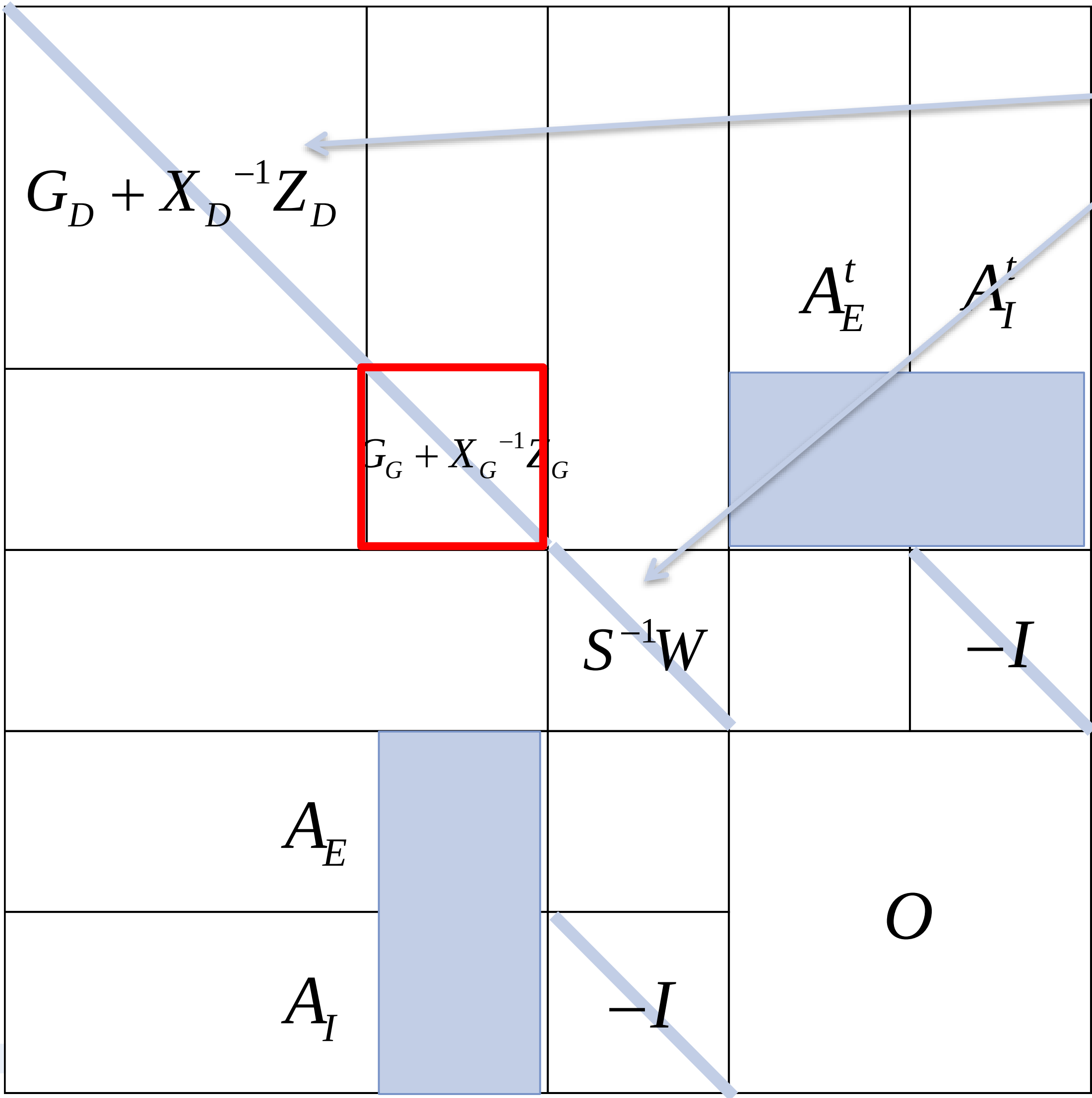
Except:

Non-Diagonal Hessian Part

Dense Column

Free Variable

Direct Solver for IPM



1. Pivot Upper left
   Diagonal Part

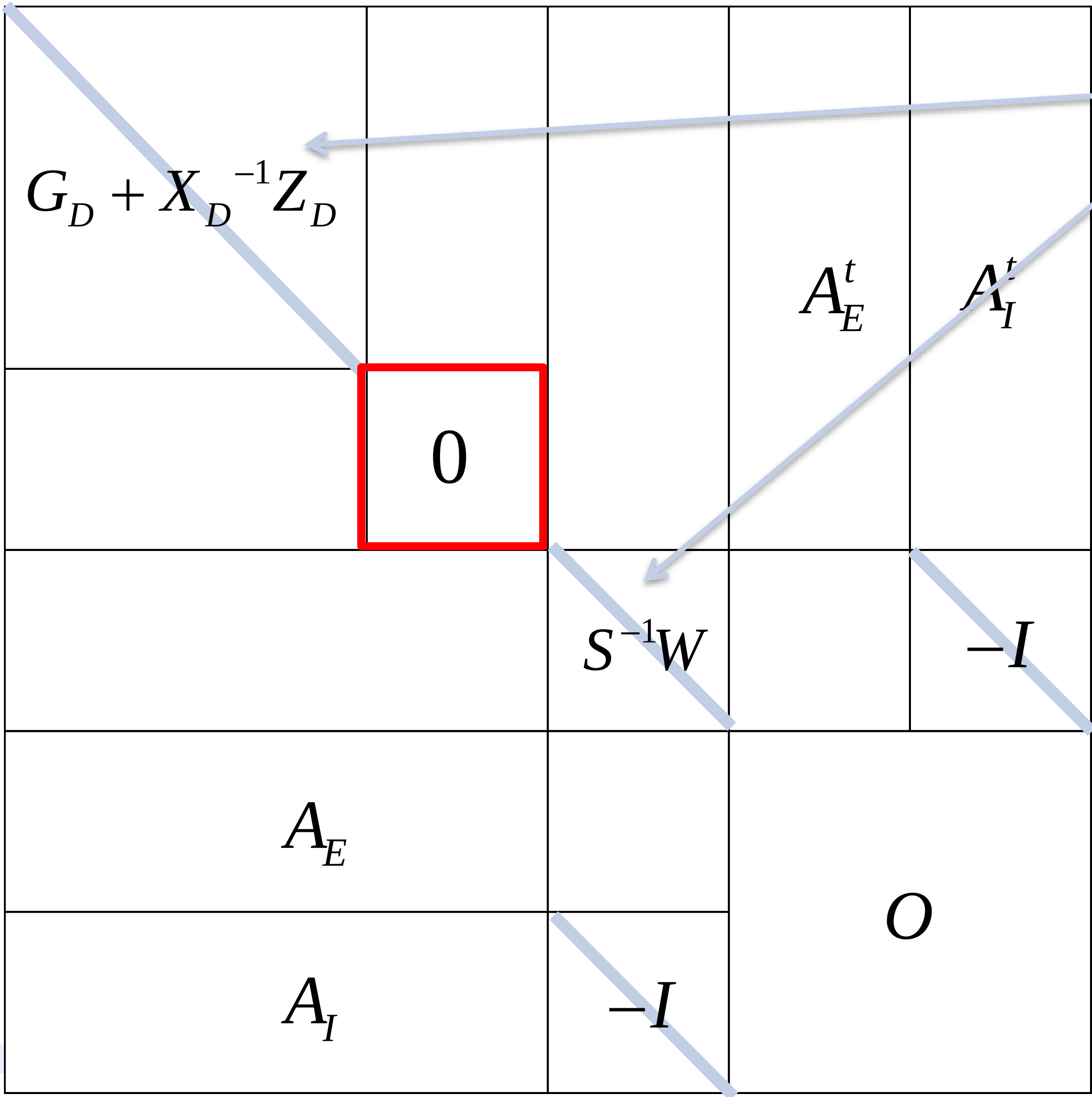
Except:

Non-Diagonal Hessian Part

Dense Column

Free Variable

Direct Solver for IPM



1. Pivot Upper left Diagonal Part

Except:

Non-Diagonal Hessian Part

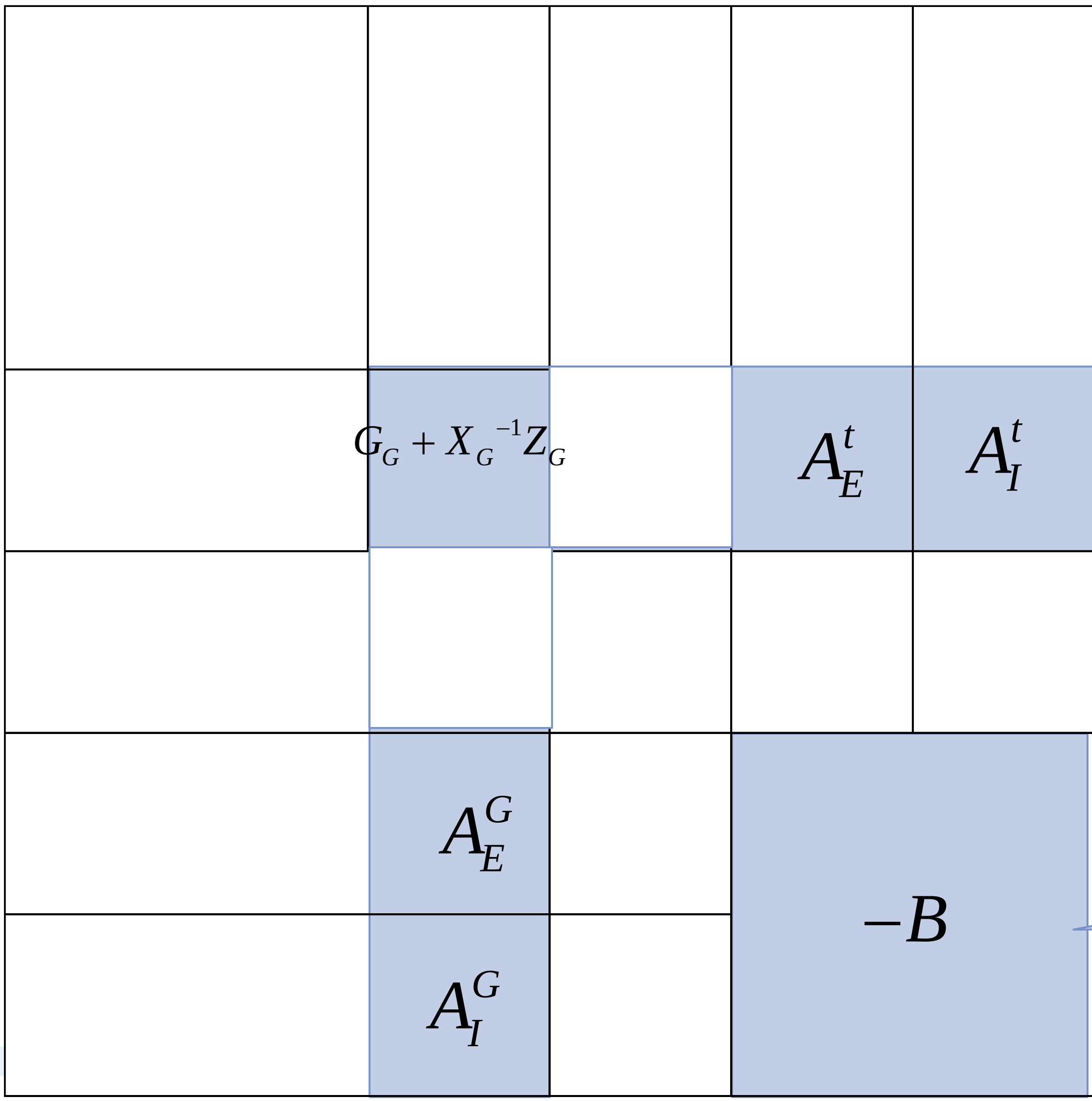
Dense Column

Free Variable

l30 (from netlib)
n:15380 m:2702
#free variable:1880

FREE VARIABLE TREATMENT	ITERATION	TIME
ON	20	2.0 sec
OFF	+151	+13.2 sec

Direct Solver for IPM



2. Factorize Remaining Sparse Indefinite Matrix

Use 2x2 pivot if required.

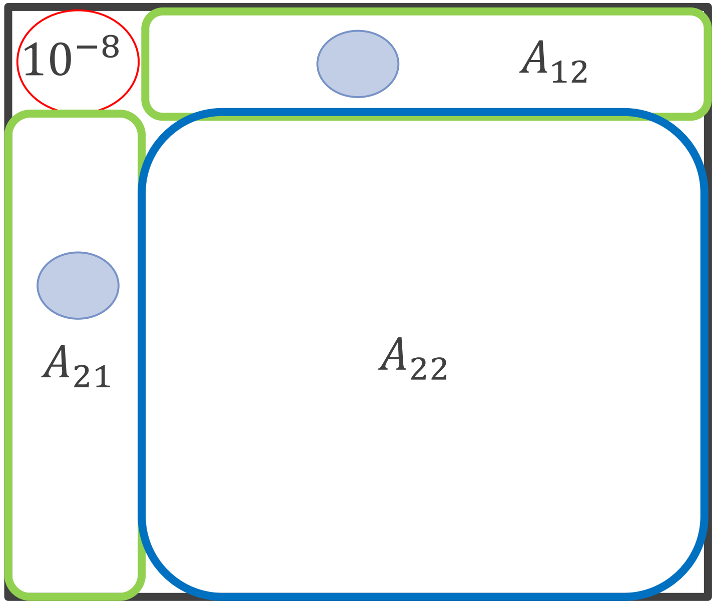
- Reduce fill
- Avoid Numerical Breakdown

$$B \equiv A^D (G_D + X_D^{-1} Z_D)^{-1} (A^D)^t + SW^{-1}$$

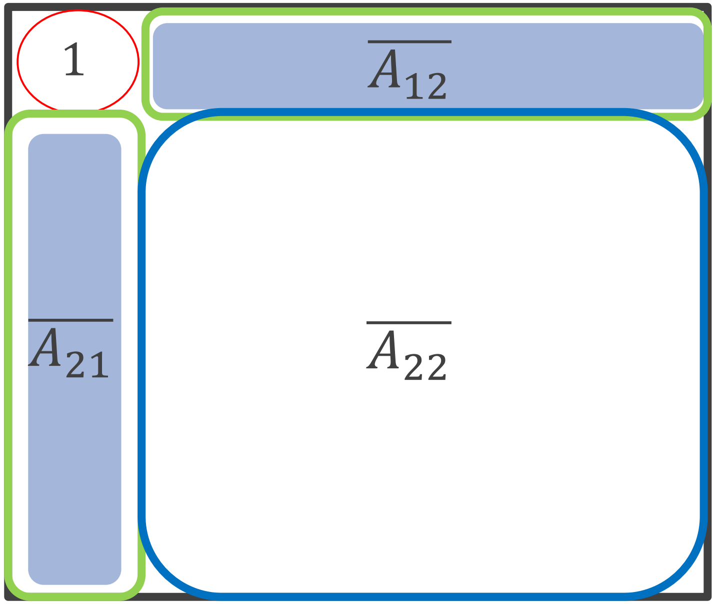
Normal Equation

Avoiding Numerical breakdown ⇔ Reducing Fill-in

$$A \equiv \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} I & O \\ A_{21}A_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} A_{11} & O \\ O & A_{22} - A_{21}A_{11}^{-1}A_{12} \end{pmatrix} \begin{pmatrix} I & A_{11}^{-1}A_{12} \\ O & I \end{pmatrix}$$



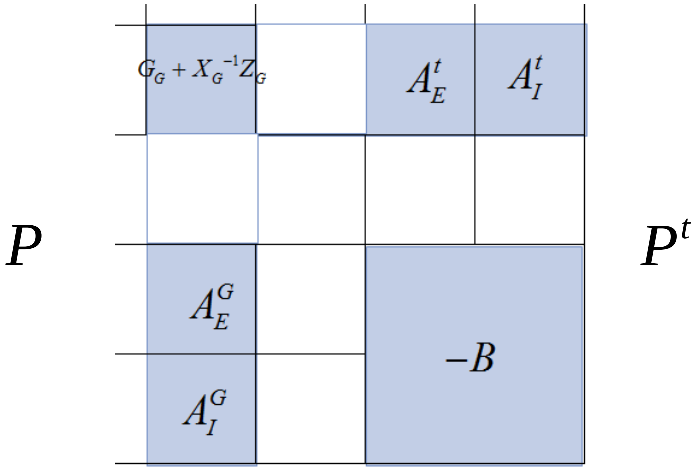
Numerically Unstable



Lose Sparsity

a. fill-reduce ordering (MLF)
b. multifrontal factorization with pivot selection (Duff 1983)

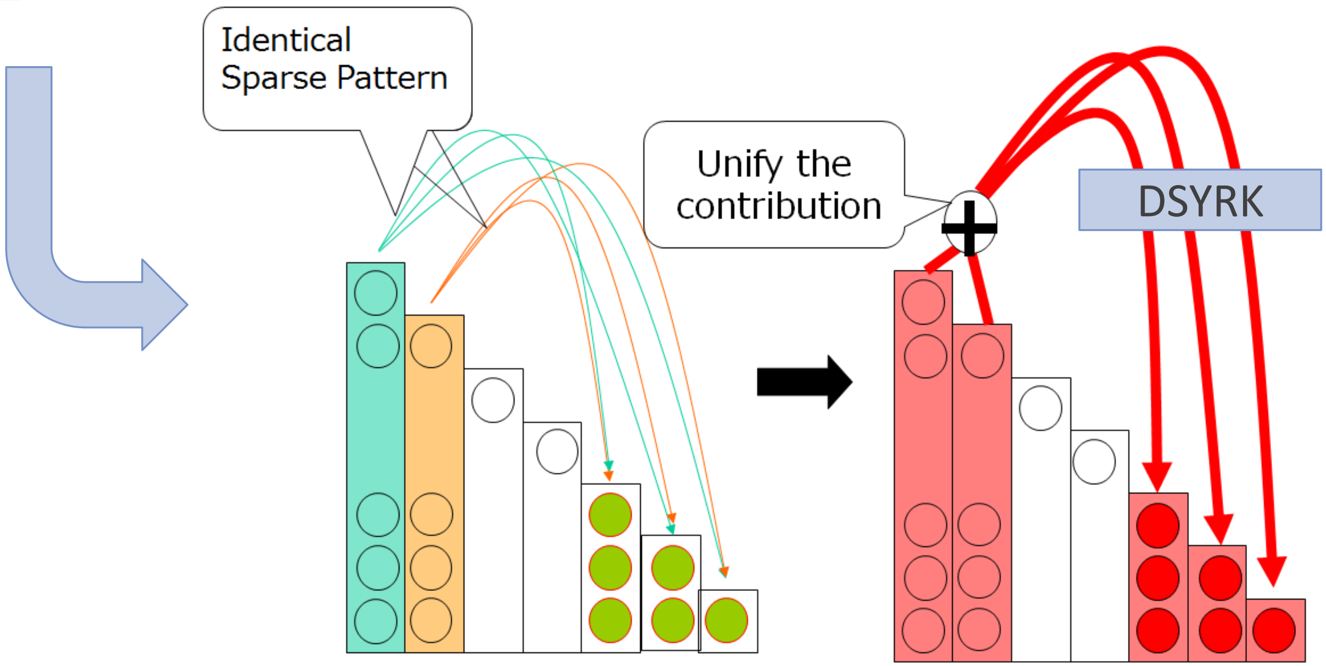
Direct Solver for IPM



3. Freeze pivot order and

'Supernodal Right Looking'

- Use dense kernel (MKL) DSYRK (level3)



Direct Solver for IPM

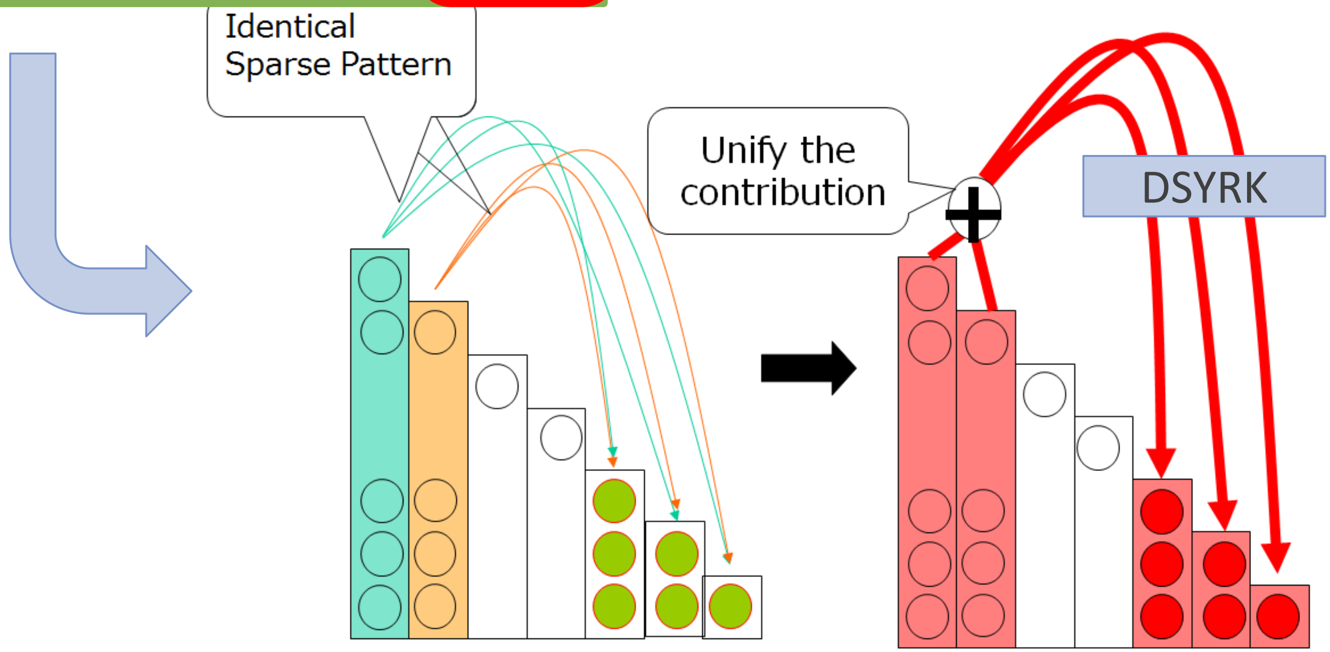
Result of LP

	n	m	nonzero	Freeze OFF(sec)	Freeze ON(sec)
DFL001	12230	6072	41873	124.7	23.3
SETCOV	394980	619	1572850	62.7	36.1
RECOM	300000	10061	900000	81.2	39.5

3. Freeze pivot order and

'Supernodal Right Looking'

- Use dense kernel (MKL) DSYRK (level3)



Observations: Direct Solvers for IPM

Become arbitrary large/small

- Finding Good Diagonal Pivot is important

  Sometimes $G + X^{-1}Z$ is more stable than $X^{-1}Z$

  ⇒ Be careful of Free variable (expecially for LP)

  ⇒ Utilize information from the modeling language

  ⇒ Dense column treatment is part of Pivot selection strategy.

- 'Freezing pivot' strategy works for most of the real problems.

  ⇒ You can share 1-2 pivot sequence almost all of the iteration.

- Sparse Indefinite Factorization is important but difficult.

  ⇒ Our code is based on MA27, left room for improvement.

Initial Point Strategy for NLP

- Few paper published
- Seek the general strategy works for LP/QP/NLP

$$
\begin{array}{llll}
\text{minimize} & f(x), & & x \in \mathbf{R}^n, \\
\text{s. t.} & g_E(x) = 0, & g_E(x) \in \mathbf{R}^{m_E}, & \\
& g_I(x) - s = 0, & g_I(x), s \in \mathbf{R}^{m_I}, & \\
& x_U \geq x \geq x_L, & g_U \geq s \geq g_L &
\end{array}
$$

NLP formulation is different
(no bound on primal)

Ref.

(2004) M.Gertz,J.Nocedal,A.Sartenar, A starting point strategy for nonlinear interior methods. *Applied Mathematics Letters* **17**:8, 945-952.

(2004) R.Waltz,Advances in Interior-Point Methods
Nonlinear Optimization, International Conference on Continuous Optimization, ICCOPT

Our conclusion

- Primal variables

  Compute Newton step $\Delta x, \Delta s$ to solve:

LP only

$$\text{minimize } \frac{1}{2}(\|x - x_L\|_2^2 + \|x_U - x\|_2^2 + \|s - g_L\|_2^2 + \|g_U - s\|_2^2) + f(x)$$

$$s.t. \quad g_E(x) = 0, \; g_I(x) - s = 0$$

and set $x \leftarrow x_0 + \Delta x, \; s \leftarrow s_0 + \Delta s \; (x_0, s_0 : \text{user specified})$

- Dual variables

$$\text{minimize } \|\nabla f(x_0) - A(x_0)^t y\|_2$$

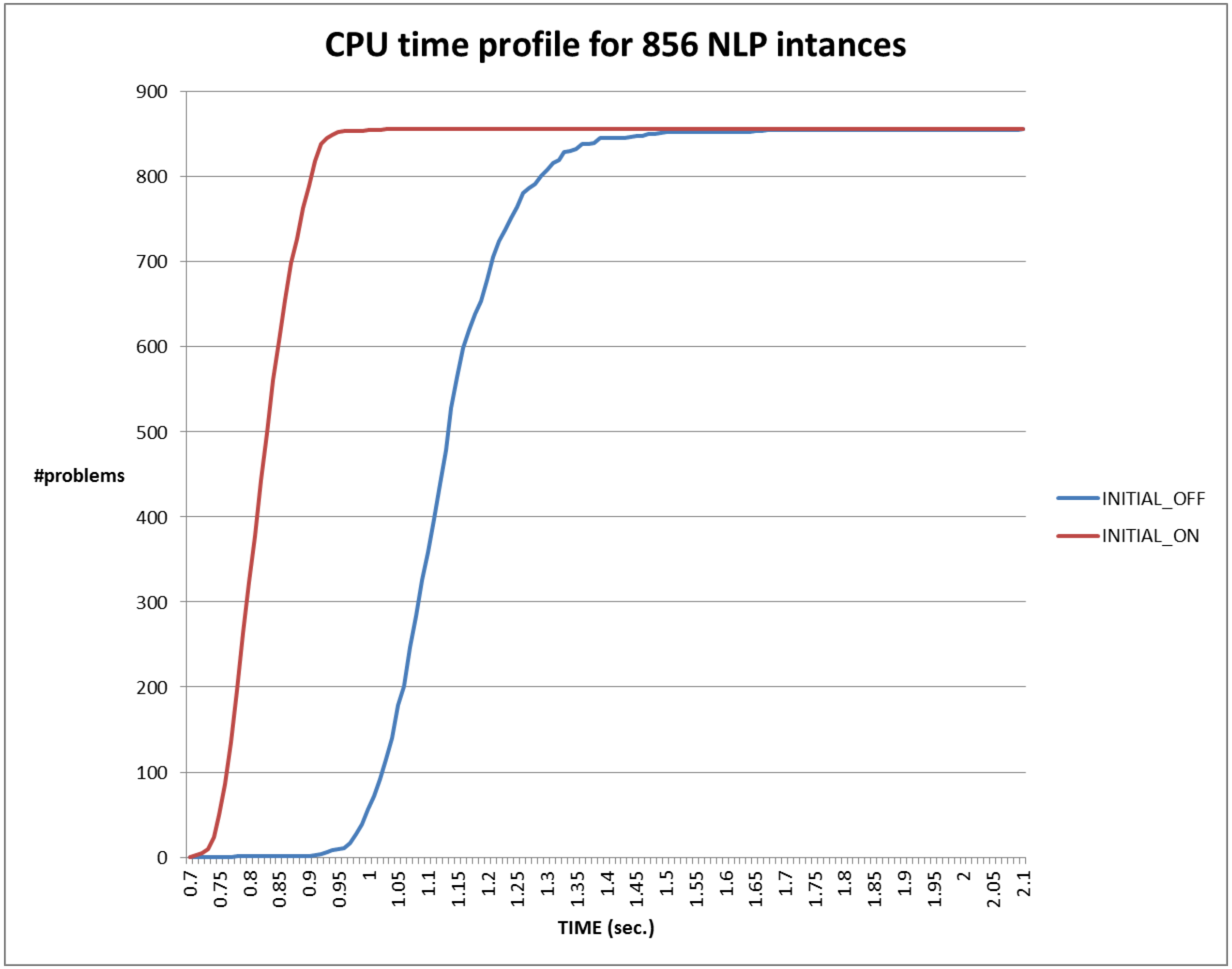
and set $z \leftarrow \nabla f(x) - A^t y, \; w \leftarrow y_I$

$$z \equiv z_{L:} - z_U, \quad z_L \cdot z_U = 0$$

$$w \equiv w_{L:} - w_U, \quad w_L \cdot w_U = 0$$

A Test of Initial Strategy

- 856 instances of NLP
- #Vars～ 6000-7000 #Cons ～10000



Interaction with Scaling

- Scaling (NLP only)

  Scale x, s by $\|\Delta x\|_\infty$ and scale $|f(x)|$ around 1

Newton step of

$$\text{minimize} \ \frac{1}{2}(\|x - x_L\|_2^2 + \|x_U - x\|_2^2 + \|s - g_L\|_2^2 + \|g_U - s\|_2^2)$$

$$s.t. \quad g_E(x) = 0, \ g_I(x) - s = 0$$

- Test

  { Scaling-on,Scaling-off } × { Initial Point-on, Initial Point-off }

A Result (Scaling & Initial Point Strategy)

Total Time (#856 runs)		Scaling	
		ON	OFF
Iinitial Point	ON	708sec	778sec
	OFF	971sec	788sec

Scaling is harmful without Initial Point Strategy

Similar Story

"Mehrotra's corrector step may be harmful without careful choice of starting point."

$$x \cdot \Delta z + z \cdot \Delta x = -Xz$$ predictor step

$$x \cdot \Delta z^{Corr} + z \cdot \Delta x^{Corr} = -(Xz - \mu) - \Delta x \Delta z$$ corrector step

(2009) J.Nocedal,A.Wachter,R.Waltz,
Adaptive Barrier Strategies for Nonlinear Interior Methods,
SIAM Journal on Optimization 19(4):1674-1693

Computing Derivatives ?

Direct Solver for IPM

$$G \equiv \nabla_x^2 f - \sum_i y_i \nabla_x^2 g_i$$

$$\begin{pmatrix} G + X^{-1}Z & & A_E^t & A_I^t \\ & S^{-1}W & & -I \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta s \end{pmatrix} =$$

Model Parameter Fitting for Financial Engineering(2005)

- Yield curve fitting for Japanese Pension fund consultant farm

minimize $\sum_j w_j (p_j - \hat{p}_j)^2$

subject to $\hat{p}_j = \sum_i c_{ij} \exp(-m_{ij} s_{ij})$

$$s_{ij} = \beta_0 + \beta_1 \frac{1-\exp\left(-\frac{m_{ij}}{\tau_1}\right)}{\frac{m_{ij}}{\tau_1}} + \beta_2 \left\{ \frac{1-\exp\left(-\frac{m_{ij}}{\tau_1}\right)}{\frac{m_{ij}}{\tau_1}} - \exp\left(-\frac{m_{ij}}{\tau_1}\right) \right\} + \beta_3 \left\{ \frac{1-\exp\left(-\frac{m_{ij}}{\tau_2}\right)}{\frac{m_{ij}}{\tau_2}} - \exp\left(-\frac{m_{ij}}{\tau_2}\right) \right\}$$

constraints:

$\beta_0 \geq 0$

$\beta_0 + \beta_1 \geq 0$

$\tau_{\min} \leq \tau_1 \leq \tau_{\max}$

$\tau_{\min} \leq \tau_2 \leq \tau_{\max}$

Model Parameter Fitting for Financial Engineering(2000)

- Decomposition of rating transition probability matrix for Japanese Bank

Variable	X	$X \in \mathbf{R}^{n \times n}$ (n:# of ratings (～20))
minimize	$\left\| X^{12} - A_{year} \right\|$	$A_{year} \in \mathbf{R}^{n \times n}$
subject to	$\sum_{j \in S} X_{ij} = 1$	$i \in S$
	$X_{ij} \geq 0$	$i, j \in S$

Automatic Differentiation Viewpoint (Computing Derivatives)

How do we express the Nonlinear Function on Computers?

$$f(x, y) = x + \sin(x \cdot y)$$

```
t = x * y
u = sin(t)
f = x + u
```

recursive application of predetermined operator/function

$$f(x) = \sum_{i=0}^{n} a_i x^i$$

```
f = a[n]
for ( i in {1..n} ) {
      f = x * f
      f = f + a[n-i]
}
```

Jacobian calculation

$$F(x) : \mathbf{R}^n \to \mathbf{R}^m \qquad \Delta F = J \Delta x \quad , J \in \mathbf{R}^{n \times m}$$

expressed in p steps

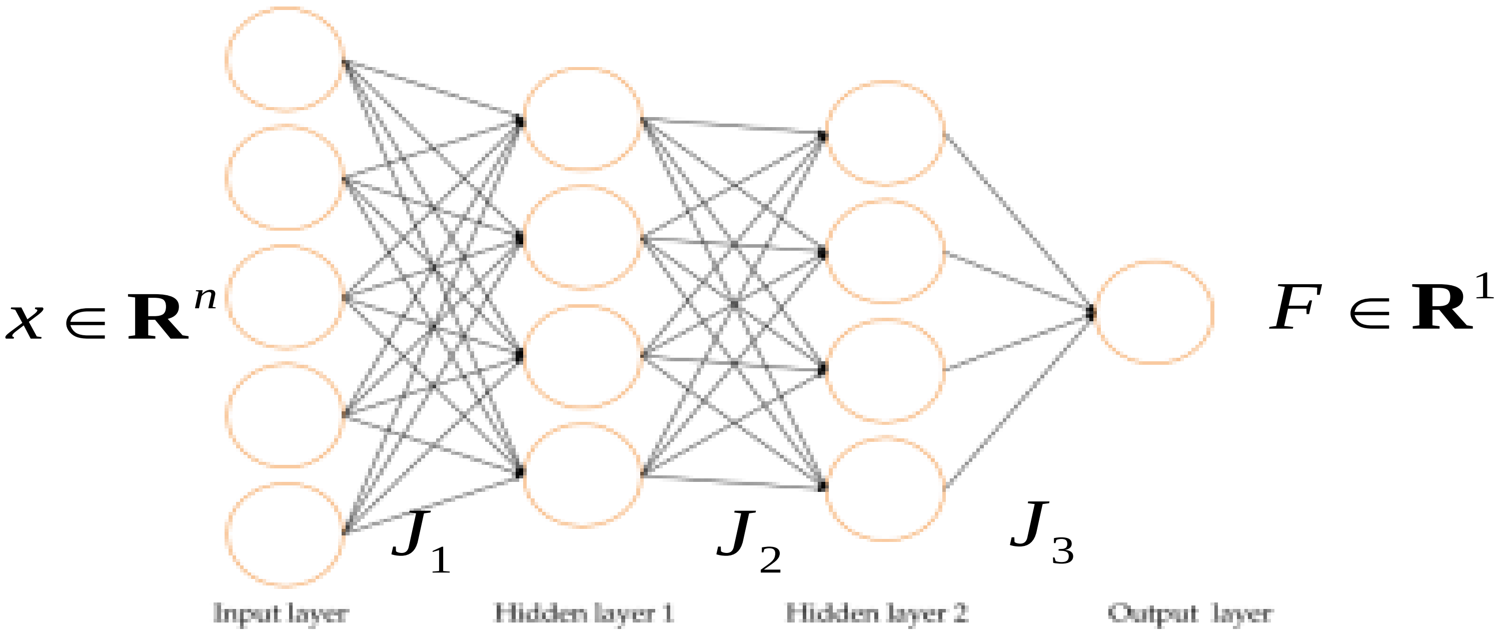
$$
\begin{aligned}
u_0 &= x \\
u_1 &= F_1(u_0) : \mathbf{R}^n \to \mathbf{R}^{s_1} \\
u_2 &= F_2(u_1) : \mathbf{R}^{s_1} \to \mathbf{R}^{s_2} \\
&\cdots \\
u_{p-1} &= F_{p-1}(u_{p-2}) : \mathbf{R}^{s_{p-2}} \to \mathbf{R}^{s_{p-1}} \\
F &= F_p(u_{p-1}) : \mathbf{R}^{s_{p-1}} \to \mathbf{R}^{s_m}
\end{aligned}
$$

$$
\begin{aligned}
\Delta u_0 &= \Delta x \\
\Delta u_1 &= J_1 \Delta u_0 &&, J_1 \in \mathbf{R}^{n \times s_1} \\
\Delta u_2 &= J_2 \Delta u_1 &&, J_2 \in \mathbf{R}^{s_1 \times s_2} \\
&\cdots \\
\Delta u_{p-1} &= J_{p-1} \Delta u_{p-2} &&, J_{p-1} \in \mathbf{R}^{s_{p-2} \times s_{p-1}} \\
\Delta F &= J_p \Delta u_{p-1} &&, J_p \in \mathbf{R}^{s_p \times s_m}
\end{aligned}
$$

$$J = J_p \cdot J_{p-1} \cdots J_2 \cdot J_1$$

Calculate the Product of Partial Jacobian matrix

Layered Neural Network（input >> output)



$$J = J_3 \, J_2 \, J_1$$

Calculation order (Backward propagation)

In real world NLP

(Partial) Jacobian is sparse

$$Area = \sum_i \frac{1}{2} \rho_i \rho_{i+1} \sin(\theta_{i+1} - \theta_i)$$

$$\sum_{j \in N_i} a_{ij} \cdot \phi_j - a_{ii} \cdot \phi_i = \rho_i$$

$$\sum_{j,(i,j) \in IJ} x_{ij} = a_i \quad , \quad \sum_{i,(i,j) \in IJ} x_{ij} = b_j$$

$J = J_p \cdot J_{p-1} \cdot \cdots \cdot J_2 \cdot J_1$
optimal calculation order is unclear (heuristics required)

$$u_0 = x$$
$$u_1 = F_1(u_0) : \mathbf{R}^n \to \mathbf{R}^{s_1}$$
$$u_2 = F_2(u_1) : \mathbf{R}^{s_1} \to \mathbf{R}^{s_2}$$
$$\cdots$$
$$u_{p-1} = F_{p-1}(u_{p-2}) : \mathbf{R}^{s_{p-2}} \to \mathbf{R}^{s_{p-1}}$$
$$F = F_p(u_{p-1}) : \mathbf{R}^{s_{p-1}} \to \mathbf{R}^{s_m}$$

sparse

$$\Delta u_0 = \Delta x$$
$$\Delta u_1 = J_1 \Delta u_0 \quad , J_1 \in \mathbf{R}^{n \times s_1}$$
$$\Delta u_2 = J_2 \Delta u_1 \quad , J_2 \in \mathbf{R}^{s_1 \times s_2}$$
$$\cdots$$
$$\Delta u_{p-1} = J_{p-1} \Delta u_{p-2} \quad , J_{p-1} \in \mathbf{R}^{s_{p-2} \times s_{p-1}}$$
$$\Delta F = J_p \Delta u_{p-1} \quad , J_p \in \mathbf{R}^{s_p \times s_m}$$

Our Current Implementation

- Expand First

W1[i,j] = sum(X[i,k]*X[k,j],k);

↓

W1[1,1] = X[1,1]*X[1,1]+X[1,2]*X[2,1]+X[1,3]*X[3,1]+…
W1[2,1] = X[2,1]*X[1,1]+X[2,2]*X[2,1]+X[2,3]*X[3,1]+…
W1[3,1] = X[3,1]*X[1,1]+X[3,2]*X[2,1]+X[3,3]*X[3,1]+…
W1[4,1] = X[4,1]*X[1,1]+X[4,2]*X[2,1]+X[4,3]*X[3,1]+…

↓

u1 = X[1,1]*X[1,1]
u2 = X[1,2]*X[2,1]
u3 = X[1,3]*X[3,1]
u4 = u1 + u2
u5 = u4 + u3
. . . .

Lose the structure information

Alternative Implementation (in progress)

- Jacobian Calculation can be expressed in Modeling Language

```
W1[i,j] = sum(X[i,k]*X
W2[i,j] = sum(W1[i,k]*
W3[i,j] = sum(W2[i,k]*W2[k,j],k);
W1[i,j] = sum(W2[i,k]*W3[k,j],k);
f = sum(pow(W1[i,j]-Ayear[i,j],2),(i,j));
```

User Defined Model

$$J = J_p \cdot J_{p-1} \cdots J_2 \cdot J_1$$

heuristics to find the calculation order

Jacobian Calculation Model (generated)

```
// Function definition
prod0[i,k,j] = X[i,k]*X[k,j];
   ..
// Partial Jacobian Element definition
AA[i,k,j,i,k] = X[k,j];
AB[i,k,j,k,j] = X[i,k];
AC[i,k,j,i,k,k,j] = 1;
   ..
// Jacobian Multiplication
BF[i13,j13,i3,j3] = sum(AT[i13,j13,i12,j12]*BE[i12,j12,i3,j3],(i12,j12));
BG[i13,j13] = sum(AX[i15,j15]*BB[i15,j15,i13,j13],(i15,j15));
            ..
// Final Jacobian Evaluation
CP[i1,j1] = sum(CE[i5,k5,j5]*AY[i5,k5,j5,i1,j1],(i5,k5,j5));
```

Dense kernel (gemm)

Summary

- NLP problems in real-world example.
  - Introduction of our package ‘NuOpt’
- Interior Point Method for NLP
  - Sparse Direct Solver
  - Initial Point Strategy
- Automatic Differentiation Implementation
  - How to utilize the structure (unfinished)

NTT DATA

NTT DATA Mathematical Systems Inc.

Trusted Global Innovator

NTT DATA Group