Rapid Prototyping of Decomposition Algorithms

Alexander Meeraus AMeeraus@gams.com

Michael Bussieck MBussieck@gams.com

Lutz Westermann LWestermann@gams.com

GAMS Development Corporation

GAMS Software GmbH

www.gams.com

GOR GAMS

Bad Honnef,

November 16, 2012

Agenda

- Introduction
- High Performance Prototypes
- Generic Algorithms

Algebraic Modeling Languages

What's that?

http://en.wikipedia.org/wiki/Algebraic_modeling_language

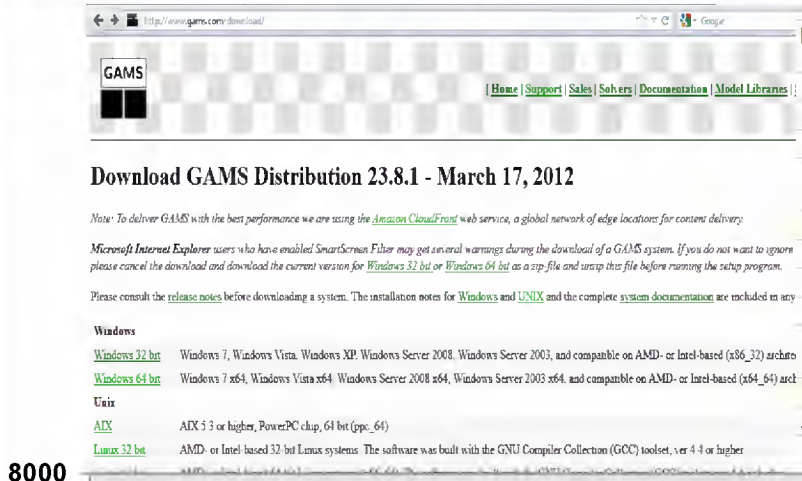
- High-level **computer programming languages** for the formulation of **complex mathematical optimization problems**
- **Notation similar to algebraic notation**: Concise and readable definition of problems in the domain of optimization
- **Do not solve problems directly**, but ready-for-use links to state-of-the-art algorithms

General Algebraic Modeling System

- Roots: World Bank, 1976
- Went commercial in 1987
- GAMS Development Corporation (Washington, Houston)
- GAMS Software GmbH (Köln, Braunschweig)

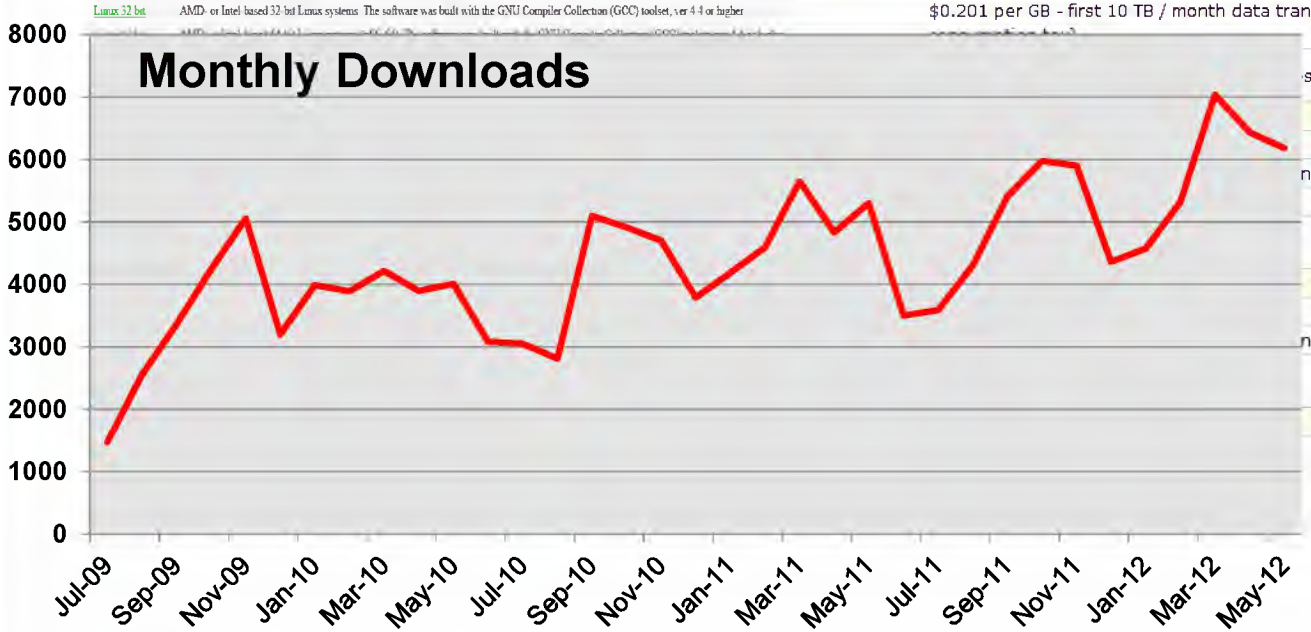
- Broad academic & commercial user community and network

Monthly System Downloads



Amazon CloudFront
Download Usage Report »

			$67.04
United States			
	$0.120 per GB - first 10 TB / month data transfer out	197.126 GB	23.66
	$0.0100 per 10,000 HTTPS Requests	3 Requests	0.01
	$0.0075 per 10,000 HTTP Requests	52,154 Requests	0.04
			23.71
Europe			
	$0.120 per GB - first 10 TB / month data transfer out	212.982 GB	25.56
	$0.0120 per 10,000 HTTPS Requests	1 Request	0.01
	$0.0090 per 10,000 HTTP Requests	16,456 Requests	0.01
			25.58
Asia Pacific (Tokyo) Region			
	$0.201 per GB - first 10 TB / month data transfer out (includes	23.800 GB	4.78
	s consumption tax).	4,676 Requests	0.01
			4.79
	nsfer out	39.512 GB	7.51
		1 Request	0.01
		18,087 Requests	0.02
			7.54
	nsfer out	21.656 GB	5.41
		1,535 Requests	0.01
			5.42



INFORMS Lanchester Prize 1973

1973	Frederick W. Lanchester Prize: Winner [-hide]

Citation: Multi-Level Planning: Case Studies in Mexico, edited by Louis M. Goreux and Alan S. Manne.
This book is a bench mark quantitative study of policy oriented issues in a growing economy. In the modern tradition of Professor W. Leontief's input-output analysis, a team of researchers from several institutions employed advanced mathematical programming approaches to study in depth the problems of interdependency among national economic choices. This monograph on multi-level planning is impressive in its dedication to developing and testing large-scale models based on available statistical data. The team's decision a half-decade ago to give special emphasis to the agricultural and energy sectors was prophetic in anticipating many of today's critical world-wide problems. Beyond its substantial contribution to empirical analysis, the book also enhances conceptual understanding of multi-level national planning as well as demonstrates the benefits to strategic policy analysis of continuing technical innovations in operations research.

Model Structure

4 (c) Regional farmer employment accounting rows:

$$-RESr + 3 \sum_{d \in r} \sum_{q} dFLq + \sum_{d \in r} \sum_{t} dFLt = 0, \quad \text{each } r$$

$$-\left[\begin{array}{c}\text{Regional farmer}\\ \text{employment}\\ \text{activity}\end{array}\right] + 3\left[\begin{array}{c}\text{Sum over districts}\\ \text{and quarters of}\\ \text{quarterly farmer}\\ \text{employment}\end{array}\right]^{37}$$

$$+\left[\begin{array}{c}\text{Sum over districts}\\ \text{and months of}\\ \text{monthly farmer employment}\end{array}\right]^{37} = 0$$

1 (d) Total employment accounting row in man-years:

$$-12LMAN + \sum_{t} LMANt = 0$$

$$-12\left[\begin{array}{c}\text{Total employment}\\ \text{in man-years}\end{array}\right] + \left[\begin{array}{c}\text{Sum over months of}\\ \text{total employment}\\ \text{in man-months}\end{array}\right] = 0$$

12 (e) Total monthly employment accounting rows in man-months:

$$-2.2LMANt + \sum_{d} dDLt + \sum_{d} dFLq + \sum_{d} dFLt = 0,$$

each t and q such that $t \in q$

$$-2.2\left[\begin{array}{c}\text{Total}\\ \text{employment}\\ \text{in month } t\end{array}\right]^{38} + \left[\begin{array}{c}\text{Sum over districts of}\\ \text{day labor employment}\\ \text{in month } t\end{array}\right]$$

$$+\left[\begin{array}{c}\text{Sum over districts of}\\ \text{quarterly farmer}\\ \text{employment in the}\\ \text{quarter containing}\\ \text{month } t\end{array}\right] + \left[\begin{array}{c}\text{Sum over districts}\\ \text{of monthly farmer}\\ \text{employment}\end{array}\right] = 0$$

[37] In irrigation districts the quarterly contract device is used for farmers, but in non-irrigated districts farmers are assumed to be available on a monthly basis, so that seasonal migration to irrigated areas may occur.

[38] The activities for hiring farmers and day laborers are stated in units of tens of man-days per month (or quarter), and there are 22 working days per month; hence the conversion factor of 2.2 is required in the first term of this equation.

Model Data

Table 3
Sequence of standard operations for cotton cultivation (days of unskilled labor, machinery services, and draft animal services required per hectare by month)

Cultivation month and operation		Mechanized		Partially mechanized			Non-mechanized	
		Unskilled labor	Machinery	Unskilled labor	Machinery	Animals	Unskilled labor	Animals
1st	Preparatory tasks		0.12		0.12		1.0	2.0
	Fallow		0.5		0.5		3.0	6.0
	Cross-plowing						2.5	5.0
	Harrowing		0.2		0.2		0.5	1.0
	Land levelling		0.25		0.25		1.0	2.0
	Canal cleaning	1.0		1.0			1.0	
2nd	Irrigation ditches	1.0	0.2	1.0	0.2		2.0	2.0
	Forming borders [a]		0.2		0.2		2.0	
	Linking borders [b]	1.0		1.0				
	Water application	2.0		2.0			2.0	
	Harrowing		0.2		0.2		2.0	4.0
	Seeding and fertilization	0.2	0.2	0.2	0.2		4.0	
	Maintenance of field works		0.2	0.2			2.0	
3rd	Thinning plants	4.0		4.0			4.0	
	Cultivation		0.2	2.0		4.0	2.0	4.0
	Weeding	6.0		6.0			6.0	
	Applications of insecticides (2) [c]							

Matrix Generator

```
Y(248)'X(248)
  IF (X(248),LT.0.5,AND,X(248),GT.,00) Y(248)'Z(248,:)*(1+X(248))  CUMPPB
Y(249)'X(249)
  IF (X(249),LT.0.5,AND,X(249),GT.,00) Y(249)'Z(249,:)*(1+X(249))  CUMPIB
Y(250)'X(250)
  IF (X(250),LT.0.5,AND,X(250),GT.,00) Y(250)'Z(250,1)*(1+X(250))  CUMNEI
Y(251)'X(251)
  IF (X(251),LT.0.5,AND,X(251),GT.,00) Y(251)'Z(251,1)*(1+X(251))  CUMONE
Y(252)'X(252)
  IF (X(252),LT.0.5,AND,X(252),GT.,00) Y(252)'Z(252,1)*(1+X(252))  CUMTRO
Y(253)'X(253)
  IF (X(253),LT.0.5,AND,X(253),GT.,00) Y(253)'Z(253,1)*(1+X(253))  CUMTRR
Y(254)'X(254)
  IF (X(254),LT.0.5,AND,X(254),GT.,00) Y(254)'Z(254,1)*(1+X(254))  CUMFCU
Y(255)'Y(266)+Y(267)                                               CUMPIV
Y(256)'X(256)                                                      CUMLCT
  IF (X(256),LT.0.5,AND,X(256),GT.,00) Y(256)'Z(256,1)*(1+X(256))
Y(257)'X(257)                                                      CUMLCG
  IF (X(257),LT.0.5,AND,X(257),GT.,00) Y(257)'Z(257,1)*(1+X(257))
Y(258)'X(258)                                                      CUMOLS
  IF (X(258),LT.0.5,AND,X(258),GT.,00) Y(258)'Z(258,1)*(1+X(258))
Y(259)'X(259)                                                      CU B-
  IF (X(259),LT.0.5,AND,X(259),GT.,00) Y(259)'Z(259,1)*(1+X(259))
Y(260)'X(260)                                                      C-  -
  IF (X(260),LT.0.5,AND,X(260),GT.,00) Y(260)'Z(260,1)*(1+X(260))
Y(261)'Y(83)                                                       EXPORT
Y(262)'X(262)                                                      NETDII
  IF (X(262),LT.0.5,AND,X(262),GT.,00) Y(262)'Z(262,1)*(1+X(262))
Y(263)'X(263)                                                      NETDFI
  IF (X(263),LT.0.5,AND,X(263),GT.,00) Y(263)'Z(263,1)*(1+X(263))
Y(264)'X(264)                                                      WARRMT
  IF (X(264),LT.0.5,AND,X(264),GT.,00) Y(264)'Z(264,1)*(1-X(264))
Y(265)'X(265)                                                      NETTRN
  IF (X(265),LT.0.5,AND,X(265),GT.,00) Y(265)'Z(265,1)*(1-X(265))
Y(266)'X(266)                                                      OFFCUR
  IF (X(266),LT.0.5,AND,X(266),GT.,00) Y(266)'Z(266,1)*(1-X(266))
Y(267)'X(267)                                                      OFFCAP
```

Matrix Generator Input

```
    25     1     8     0     0     0     1      AGGREGAT
       0.12                    0.0165
ALA ALG ALV ARO AZU CAR CEG CHV FRI GAR JIT JON MAI MAT MEL P
PLU SAL SAN SOR SOT SOY TRI
    0.0286
     99999
AZU AZU    -0.25        1.0         0.0070     2627020.
JIT JIT    -0.4         1.0         0.1150     174752.
PEP PEP    -0.6         1.0         0.0590     19.
PLU PLU    -1800.       1.0         0.5770     85209.
CCC
CHI         1     -0.2
CHV          0.1500     14.459     1.0
FOR         6     -0.3
SOR          0.0630    285.818     1.0
CEG          0.0930      0.665     1.0
ALV          0.0100    226.109     1.0
ALA          0.0400    179.019     1.0
GAR          0.0990      1.427     1.0
MAI          0.0860     77.997     1.0
FEC         4     -0.3
FRI          0.1830     33.001     1.0
ARO          0.1220    126.197     1.0
PAP          0.0930     27.138     1.0
GAR          0.0990      0.158     1.0
GRA         2     -0.1
MAI          0.0860    192.804     1.0
TRI          0.0800    343.979     1.0
FRU         2     +2.0
SAN          0.0780     10.850     1.0
MEL          0.0680      6.9350      1.0
OLE         4     -1.2
SAL          0.0830    193.910     1.0
JON          0.2410      9.224     1.0
CAR          0.1550     72.490     1.0
SOY          0.1600     57.220     1.0
END
ALA          .02         0.0
```

MPS File – Column Section

```
X,ASGHC2   B,AS,,C2        -1,00000
X,ASGHC2   A,TRA            6,98400
X,ASGHC3   D,,,GH,N         0,33500
X,ASGHC3   R,,,GHC3         1,00000
X,ASGHC3   B,AS,,C3        -1,00000
X,ASGHC3   A,TRA            6,98400
X,ASGHAS   D,,,GH,N         0,20600
X,ASGHAS   R,,,GHAS         1,00000
X,ASGHAS   B,AS,,AS        -1,00000
X,ASGHAS   A,TRA            6,98400
X,ASGHS1   D,,,GH,P         0,15000
X,ASGHS1   R,,,GHS1         1,00000
X,ASGHS1   B,AS,,S1        -1,00000
X,ASGHS1   A,TRA            6,98400
X,ASGHCN   R,,,GHCN         1,00000
X,ASGHCN   B,AS,,CN        -1,00000
X,ASGHCN   A,TRA            6,98400
X,ASKSC1   D,,,KS,N         0,26000
X,ASKSC1   R,,,KSC1         1,00000
X,ASKSC1   B,AS,,C1        -1,00000
X,ASKSC1   A,TRA            7,56000
X,ASKSC2   D,,,KS,N         0,31000
X,ASKSC2   R,,,KSC2         1,00000
X,ASKSC2   B,AS,,C2        -1,00000
X,ASKSC2   A,TRA            7,56000
X,ASKSC3   D,,,KS,N         0,33500
X,ASKSC3   R,,,KSC3         1,00000
X,ASKSC3   B,AS,,C3        -1,00000
X,ASKSC3   A,TRA            7,56000
X,ASKSAS   D,,,KS,N         0,20600
X,ASKSAS   R,,,KSAS         1,00000
X,ASKSAS   B,AS,,AS        -1,00000
X,ASKSAS   A,TRA            7,56000
X,ASKSS1   D,,,KS,P         0,15000
X,ASKSS1   R,,,KSS1         1,00000
X,ASKSS1   B,AS,,S1        -1,00000
X,ASKSS1   A,TRA            7,56000
X,ASKSCN   R,,,KSCN         1,00000
X,ASKSCN   B,AS,,CN        -1,00000
```

MPS Revision File

```
     BRANCH     *          MAJERR

     NEXT

     REVISE     REV5       TAPE14


                                                  ***** CARD READ SUMMARY ****

HEADER, CARD NO     1    $NAME          REV4
HEADER, CARD NO     2    $COLUMNS
HEADER, CARD NO     3    $  MODIFY
HEADER, CARD NO     6    $RHS
HEADER, CARD NO     7    $  MODIFY
HEADER, CARD NO    18    $ENDATA
HEADER, CARD NO    19    $NAME          REV1
HEADER, CARD NO    20    $COLUMNS
HEADER, CARD NO    21    $  MODIFY
HEADER, CARD NO    42    $ENDATA
HEADER, CARD NO    43    $NAME          REV2
HEADER, CARD NO    44    $COLUMNS
HEADER, CARD NO    45    $  MODIFY
HEADER, CARD NO    51    $ENDATA
HEADER, CARD NO    52    $NAME          REV4
HEADER, CARD NO    53    $RHS
HEADER, CARD NO    54    $  MODIFY
HEADER, CARD NO    68    $ENDATA
HEADER, CARD NO    69    $NAME          REV5
HEADER, CARD NO    70    $RHS
HEADER, CARD NO    71    $  MODIFY
        CARD NO    72    $     RHS1     CL4,V,01     5.03328
        CARD NO    73    $     RHS1     CL4,V,02     5.03328
        CARD NO    74    $     RHS1     CL4,V,03     5.03328
        CARD NO    75    $     RHS1     CL4,V,04     5.03328
        CARD NO    76    $     RHS1     CL4,V,05     5.03328
        CARD NO    77    $     RHS1     CL4,V,06     5.03328
        CARD NO    78    $     RHS1     CL4,V,07     5.03328
        CARD NO    79    $     RHS1     CL4,V,08     5.03328
        CARD NO    80    $     RHS1     CL4,V,09     5.03328
        CARD NO    81    $     RHS1     CL4,V,10     5.03328
        CARD NO    82    $     RHS1     CL4,V,11     5.03328
        CARD NO    83    $     RHS1     CL4,V,12     5.03328
        CARD NO    84    $     RHS1     CL4,V,TO    60.39936
HEADER, CARD NO    85    $ENDATA
```

MPS Output

DATE 07/30/76 TIME 22.12.21

PRINT OPTION = COMPLETE OUTPUT W/SPECIAL
NAME = CENTRAL OBJ = OBJ RHS = RHS1
DIR = MAXIMIZE OBJ = BND = LIMITS
RNG =

COLUMNS

APEX-III 1.000 PAGE

ITIVE = 28.18489
1.0000 RPSRHS = 1.0000
0.0000 RPCHRMS = 0.0000

NUMBER	NAME	TYPE	STATUS	COL ACTIVITY	OBJ COEF	D UPPER	MARGINAL
101	CBE1V..	PL	LOWER	.	-47.80000	+INF	-6.46851
102	CBE2F..	PL	ACTIVE	.00087	-701.00000	+INF	.
103	CBE3C..	PL	ACTIVE	.	-10330.60000	+INF	.
104	CBE4F..	PL	LOWER	.	-2429.70000	+INF	-912.25118
105	CBE5C..	PL	LOWER	.	-9410.00000	+INF	-2342.38642
106	CBE6C..	PL	ACTIVE	.	-5118.00000	+INF	.
107	CBE7C..	PL	ACTIVE	.06067	-15.20000	+INF	.
108	CSG.V..	PL	ACTIVE	.	-231.57000	+INF	.
109	CSG.F..	PL	ACTIVE	.00226	-231.57000	+INF	.
110	CPD.V..	PL	ACTIVE	.	-139.67000	+INF	.
111	CPD.F..	PL	ACTIVE	.00002	-139.67000	+INF	.
112	CPD.C..	PL	ACTIVE	.00045	-139.67000	+INF	.
113	CEG.V..	PL	ACTIVE	.	-76.71000	+INF	.
114	CEG.F..	PL	ACTIVE	.00025	-76.71000	+INF	.
115	CEG.C..	PL	ACTIVE	.00128	-76.71000	+INF	.
116	COA.CX.	PL	ACTIVE	.07685	12.91000	+INF	.
117	COF.CX.	PL	LOWER	.	180.74000	+INF	-87.19134
118	COC.CX.	PL	LOWER	.	167.83000	+INF	-256.39963
119	COS.CX.	PL	ACTIVE	.06968	121.35000	+INF	.
120	COL.CX.	PL	ACTIVE	.00225	91.66000	+INF	.
121	CWS.CX.	PL	ACTIVE	.00606	109.74000	+INF	.
122	CWL.CX.	PL	ACTIVE	.00748	77.46060	+INF	.

WB Old Slide 1

PLANNING PROBLEM AND OBJECTIVES INITIALLY OFTEN

UNSTRUCTURED

ILL-DEFINED

CONFLICTING

UNCERTAIN

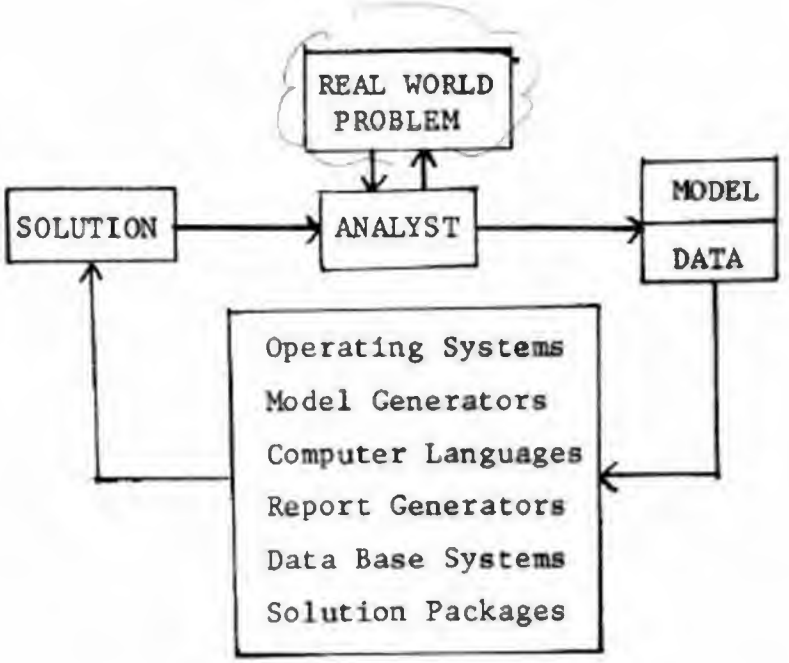
CHANGING

EMOTIONAL

MATHEMATICAL MODEL USED TO RECOGNIZE AND FORMULATE PROBLEMS, DEFINE ISSUES AND EXPLORE SOLUTION SPACE

WB Old Slide 2

PRESENT TECHNOLOGY



RESULT: - Drain of resources (technical, time, money)

- Essentially no documentation

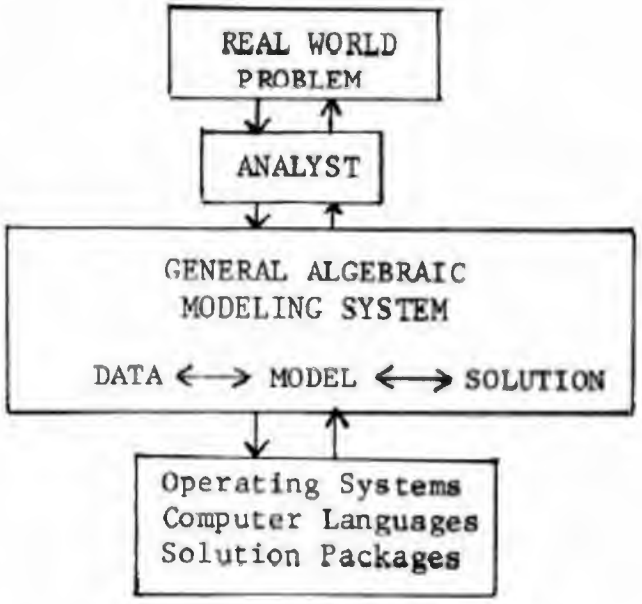
WB Old Slide 3

MAJOR CONSTRAINTS :

- COST
- SKILLS
- TIME
- TOOLS
- DOCUMENTATION
- TRUST
- .
- .
- .

WB Old Slide 4

IDEAL TECHNOLOGY



RESULT:
- Limited drain of resources
- Same representation of models for humans and machines
- Model representation is also model documentation

MP 1982

Mathematical Programming Study 20 (1982) 1–29
North-Holland Publishing Company

ON THE DEVELOPMENT OF A GENERAL ALGEBRAIC MODELING SYSTEM IN A STRATEGIC PLANNING ENVIRONMENT*

Johannes BISSCHOP** and Alexander MEERAUS

Development Research Center, The World Bank, Washington, DC 20433, U.S.A.

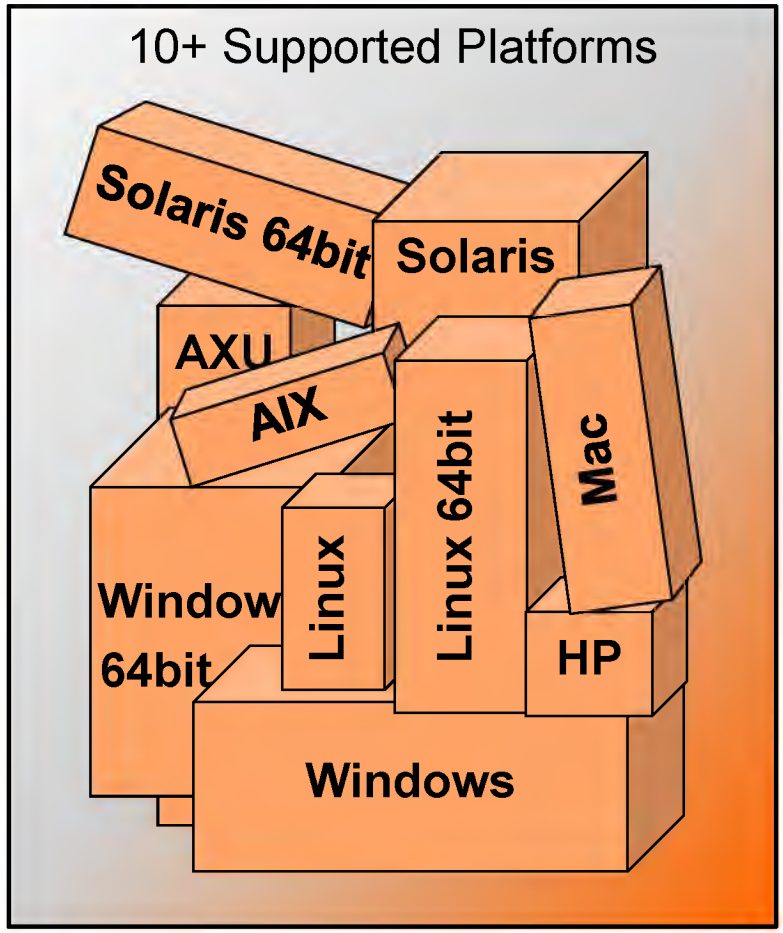
Received 18 March 1980
Revised manuscript received 8 May 1981

Modeling activities at the World Bank are highlighted and typified. Requirements for successful modeling applications in such a strategic planning environment are examined. The resulting development of a General Algebraic Modeling System (GAMS) is described. The data structure of this system is analyzed in some detail, and comparisons to other modeling systems are made. Selected aspects of the language are presented. The paper concludes with a case study of the Egyptian Fertilizer Sector in which GAMS has been used as a modeling tool.

Key words: Algebraic Modeling System, Modeling Language, Strategic Planning, Applications.

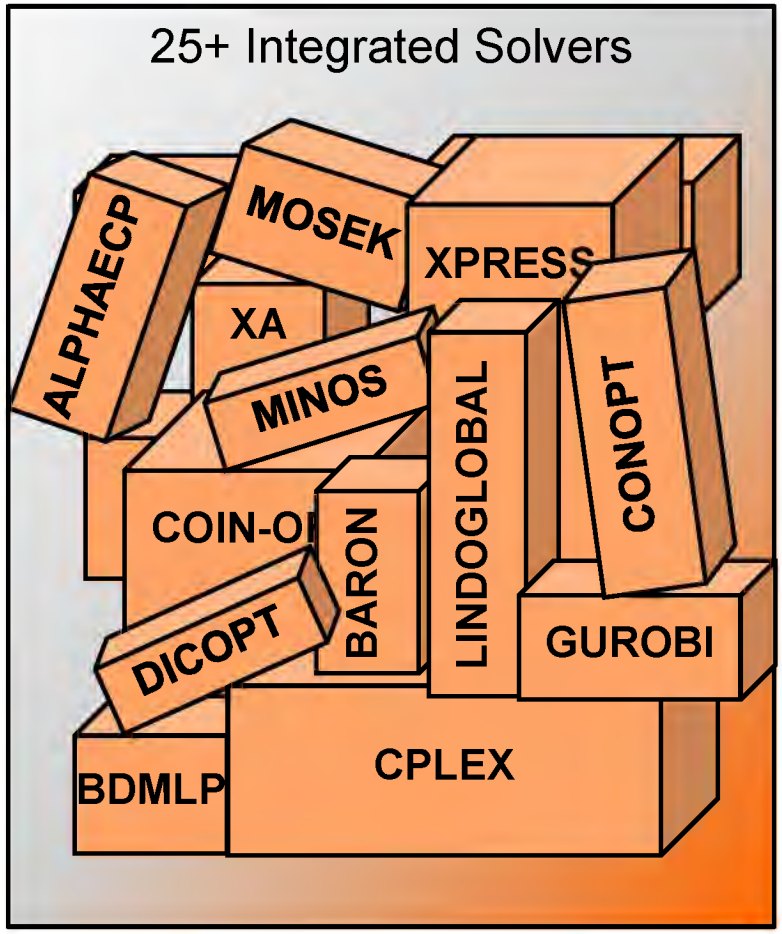
GAMS' Fundamental concepts

- **Platform independence**
- Hassle-free switch of solution methods
- Open architecture and interfaces to other systems
- Balanced mix of declarative and procedural elements



GAMS' Fundamental concepts

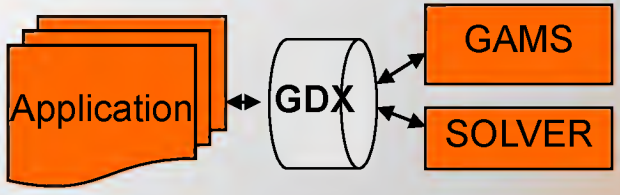
- Platform independence
- **Hassle-free switch of solution methods**
- Open architecture and interfaces to other systems
- Balanced mix of declarative and procedural elements



GAMS' Fundamental concepts

- Platform independence
- Hassle-free switch of solution methods
- **Open architecture and interfaces to other systems**
- Balanced mix of declarative and procedural elements

Binary Data Exchange



- Fast exchange of data
- Syntactical check on data before model starts
- Data Exchange at any stage (Compile and Run-time)
- Platform Independent
- Direct GDX interfaces and general API
- Scenario Management Support
- Full Support of Batch Runs

GAMS' Fundamental concepts

- Platform independence
- Hassle-free switch of solution methods
- Open architecture and interfaces to other systems
- **Balanced mix of declarative and procedural elements**

Declaration of..

- **Sets**
- **Parameters**
- **Variables**
- **Equations**
- **Models**
- **…**

Procedural Elements like…

- **loops**
- **if-then-else**
- **…**

Solver Prototypes in GAMS

- DICOPT (Grossmann)
- BARON (Sahinidis)
- SBB (Drud)
- NLPEC
- …
- *EMP*

New Modeling and Solution Concepts

- Breakouts of traditional MP classes
  - Extended Nonlinear Programs
  - Chance Constraints
  - CVaR Constraints
  - Robust Programming
  - Bilevel Programs
  - Generalized Disjunctive Programs
  - Multi Agent Equilibrium

- Limited support with common model representation
- No conventional syntax
- Incomplete/experimental solution approaches
- Lack of reliable/any software

What now?

Do not:

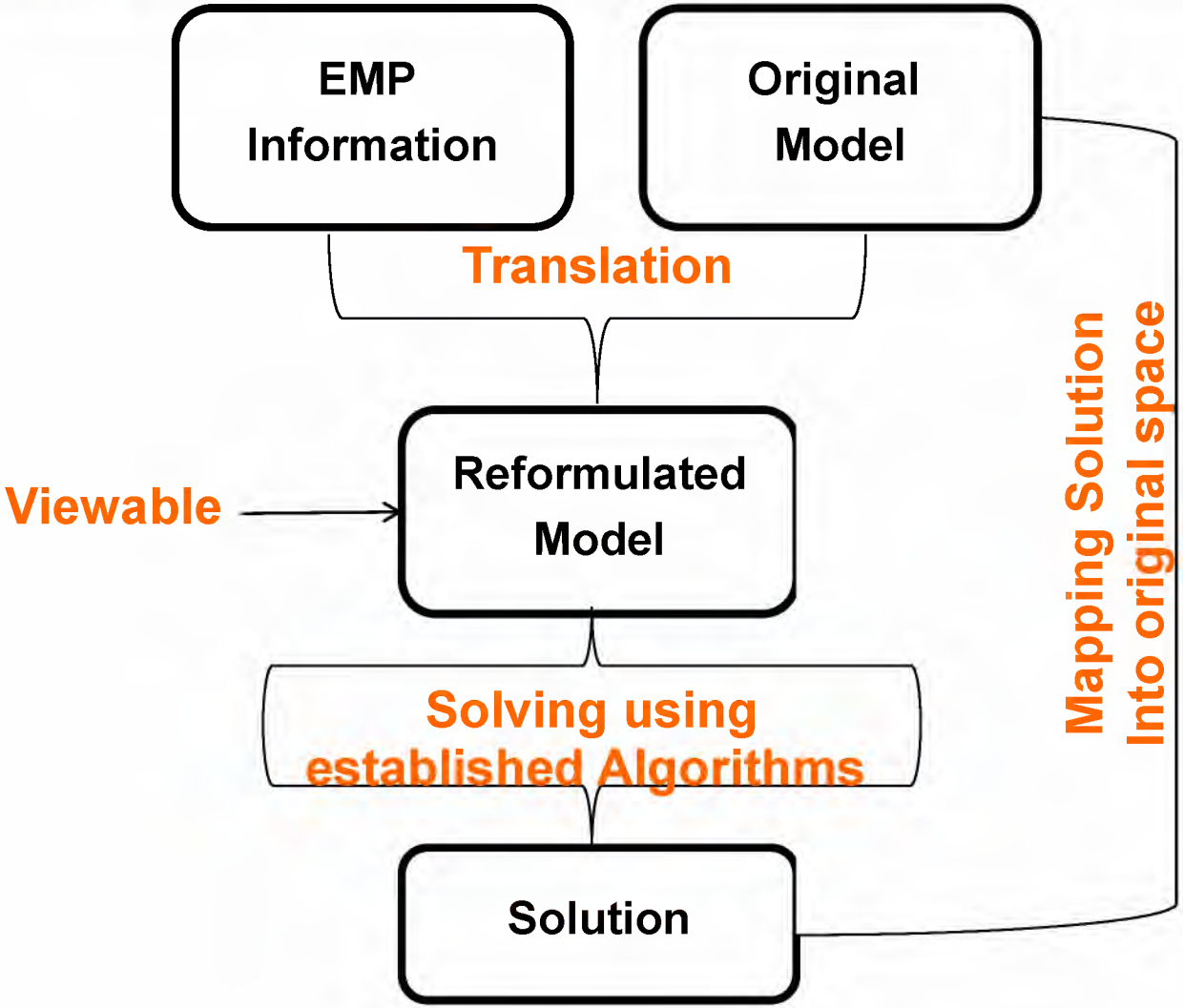
- overload existing GAMS notation right away !
- attempt to build new solvers right away !

But:

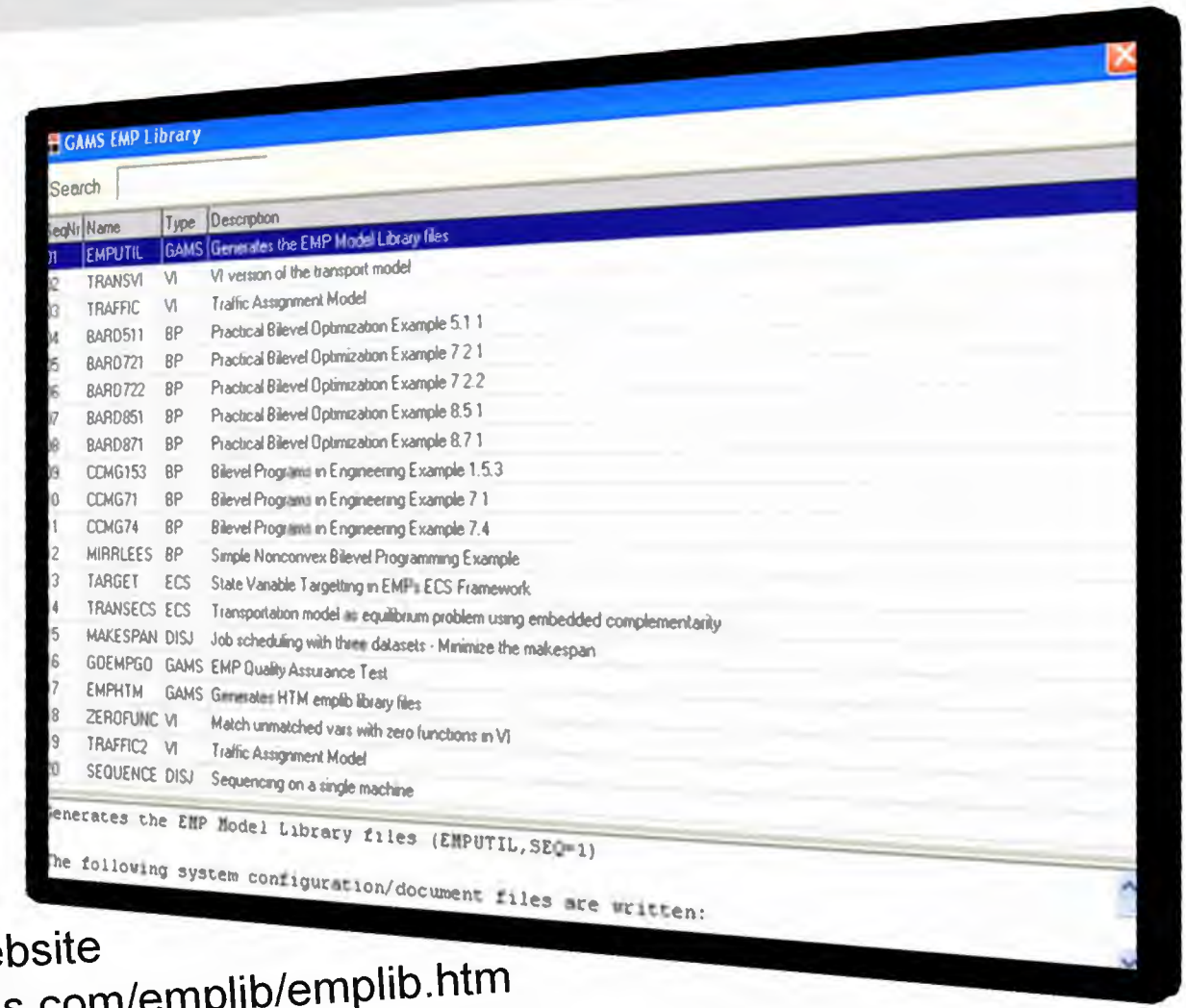
- Use existing language features to specify additional model features, structure, and semantics
- Express extended model in symbolic (source) form and apply existing modeling/solution technology
- Package new tools with the production system

→ **Extended Mathematical Programming (EMP)**

JAMS: a GAMS EMP Solver



EMP Library



- Distributed with GAMS
- Available on website http://www.gams.com/emplib/emplib.htm

Agenda

- Introduction
- High Performance Prototypes
- Generic Algorithms

Simple Transport Model

```
Sets
   i factories                                                  /f1*f3/
   j distribution centers                                       /d1*d5/

Parameter
   capacity(i) /f1 500, f2 450, f3 650/
   demand(j)   /d1 160, d2 120, d3 270, d4 325, d5 700 /
   prodcost  unit production cost                               /14/
   price     sales price                                        /24/
   wastecost cost of removal of overstocked products /4/

Table transcost(i,j) unit transportation cost
       d1    d2    d3    d4    d5
  f1   2.49  5.21  3.76  4.85  2.07
  f2   1.46  2.54  1.83  1.86  4.76
  f3   3.26  3.08  2.60  3.76  4.45;
```

Simple Transport Model – Cont.

```
Variables
   ship(i,j)    shipments
   product(i)   units produced
   received(j)  unit received
   sales(j)     sales (actually sold)
   waste(j)     overstocked products
   profit
Positive variables ship,product,sales,waste;

Equations
   obj
   production(i)
   receive(j)
   selling(j)
   market(j);

obj.. profit =e= sum(j, price*sales(j)) - sum((i,j), transcost(i,j)*ship(i,j))
                 - sum(j, wastecost*waste(j)) - sum(i,prodcost*product(i));

production(i).. product(i) =e= sum(j, ship(i,j));
product.up(i) = capacity(i);

receive(j).. received(j) =e= sum(i, ship(i,j));
selling(j).. sales(j) =e= received(j) - waste(j);
market(j)..  sales(j) =l= demand(j);

model transport /all/;
solve transport maximizing profit using lp;
```

→ t_det.gms

Benders Decomposition for 2-Stage SP

```
Set
   s scenarios /lo,mid,hi/

* Stochastic demand plus probabilities
Table ScenarioData(s,*)
      d1   d2   d3   d4   d5 prob
lo   150  100  250  300  600 0.25
mid  160  120  270  325  700 0.50
hi   170  135  300  350  800 0.25;
```

$$\min c^T x + \sum_{\omega} p(\omega) d_\omega^T y_\omega$$

$$Ax = b$$

$$T_\omega x + W_\omega y_\omega = h_\omega$$

$$x \geq 0, y_\omega \geq 0$$

$$\min d_\omega^T y_\omega$$

$$W_\omega y_\omega = h_\omega - T_\omega \overline{x}^\nu$$

$$y_\omega \geq 0$$

$$\min c^T x + \theta$$

$$Ax = b$$

$$\theta \geq \sum_{\omega \in \Omega} p_\omega \left(-\overline{\pi}_\omega^\ell [T_\omega x + W_\omega \overline{y}_\omega^\ell - h_\omega] \right), \ell = 1, \ldots, \nu - 1$$

$$x \geq 0$$

Benders Decomposition for 2-Stage SP

based on paper by E. Kalvelagen

```
masterobj..
    zmaster =e=  theta -sum((i,j), transcost(i,j)*ship(i,j))
                       - sum(i,prodcost*product(i));

receive(j)..        received(j) =e= sum(i, ship(i,j));

production(i)..     product(i) =e= sum(j, ship(i,j));
product.up(i) = capacity(i);

optcut(dyniter)..  theta =l= cutconst(dyniter) +
                             sum(j, cutcoeff(dyniter,j)*received(j));
```

```
subobj..
   zsub =e= sum(j, price*sales(j)) - sum(j, wastecost*waste(j));

selling(j)..  sales(j) + waste(j) =e= received.l(j);

market(j)..   sales(j) =l= demand(j);
```

Benders Decomposition for 2-Stage SP

based on paper by E. Kalvelagen

```
loop(iter$(not done),
* solve subproblems
   dyniter(iter) = yes;
   loop(s,
       demand(j) = ScenarioData(s,j);
       solve subproblem max zsub using lp;
       objsub(s) = zsub.l;
       cutconst(iter) = cutconst(iter) + p(s)*sum(j,market.m(j)*demand(j));
       cutcoeff(iter,j) = cutcoeff(iter,j) + p(s)*selling.m(j);
   );
   lowerbound = max(lowerbound, objmaster + sum(s, p(s)*objsub(s)));

* convergence test
   if( (upperbound-lowerbound) < 0.001*(1+abs(upperbound)),
      done = 1;
   else
* solve masterproblem
      solve masterproblem max zmaster using lp;
      upperbound = zmaster.l;
      objmaster = zmaster.l - theta.l;
   );
);
```

→ t_gams.gms

Benders GAMS Implementation

- GAMS Implementation (solver Cplex)
  - 17 iterations: (3+1)*17 = 68 small models
    - 6.1 secs (all default)
    - 5.8 secs (minimize listing file size)
    - 4.6 secs (GAMS stays in memory)
    - 0.5 secs (communicate with solver through memory)

```
option limrow=0, limcol=0, solprint=silent,
       solvelink=%Solvelink.LoadLibrary%;
```

- Grid computing
- Smart update of sub-model (Scenario Solver/GUSS)
- Object Oriented API (e.g. .NET)

Parallel Power – GAMS Grid Facility

```
subproblem.solvelink = %Solvelink.AsyncGrid%;
loop(s,
  demand(j) = sDemand(s,j);
  solve subproblem max zsub using lp;
  h(s) = subproblem.handle;
);


Repeat
  loop(s$handlecollect(h(s)),
    objsub(s) = zsub.l;
    display$handledelete(h(s)) 'trouble deleting handles';
    h(s) = 0 );
    display$sleep(card(h)*0.02) 'sleep for some time';
until card(h)=0;
```

Parallel Power – GAMS Grid Facility

```
loop(iter$(not done),
* solve subproblems
   dyniter(iter) = yes;
* Submission loop
   loop(s,
        demand(j) = ScenarioData(s,j);
        solve subproblem max zsub using lp;
        h(s) = subproblem.handle;
   );
* Collection loop
  repeat
     loop(s$handlecollect(h(s)),
        objsub(s) = zsub.l;
        cutconst(iter) = cutconst(iter) + p(s)*sum(j,market.m(j)*sDemand(s,j));
        cutcoeff(iter,j) = cutcoeff(iter,j) + p(s)*selling.m(j);
        display$handledelete(h(s)) 'trouble deleting handles' ;
        h(s) = 0 );
     display$sleep(card(h)*0.02) 'was sleeping for some time';
  until card(h) = 0;
  lowerbound = max(lowerbound, objmaster + sum(s, p(s)*objsub(s)));

* convergence test
```

→ t_grid.gms

GUSS: Gather-Update-Solve-Scatter

```
market(j).. sales(j)=l= demand(j);
subobj..    zsub     =e= sum(j, price*sales(j)) -
                         sum(j,wastecost*waste(j));

loop(s,
  demand(j) = sDemand(s,j);
  solve subproblem max zsub using lp;
  objsub(s) = zsub.l;
);


set dict / s.       scenario. ''
           demand.  param.    sDemand
           zsub.    level.    Objsub   /

solve subproblem max zsub using lp scenario dict;
```

GUSS: Gather-Update-Solve-Scatter

```
* GUSS setup
Set dict / s.       scenario. ''
           demand. param.    sDemand
           market. marginal. sMarket
           selling.marginal. sSelling
           zsub.   level.    objsub /;

loop(iter$(not done),
* solve subproblems
   dyniter(iter) = yes;
   solve subproblem max zsub using lp scenario dict;
   cutconst(iter) = cutconst(iter)+sum(s,p(s)*sum(j,sMarket(s,j)*sDemand(s,j)));
   cutcoeff(iter,j) = cutcoeff(iter,j) + sum(s,p(s)*sSelling(s,j));

   lowerbound = max(lowerbound, objmaster + sum(s, p(s)*objsub(s)));

* convergence test
   if( (upperbound-lowerbound) < 0.001*(1+abs(upperbound)),
      done = 1;
   else
* solve masterproblem
      solve masterproblem max zmaster using lp;
      upperbound = zmaster.l;
      objmaster = zmaster.l - theta.l;
   );
);
```

→ t_guss.gms

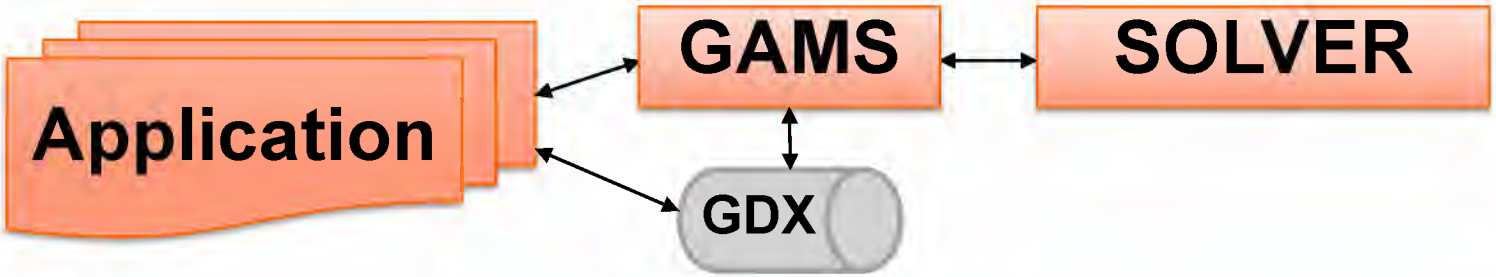
GUSS: Gather-Update-Solve-Scatter

- 100 scenarios, 38 iterations: (100+1)*38 = 3838 models

Setting	Solve time (secs)
Solvelink=0 (default)	294.36
Solvelink=%Solvelink.CallModule%	239.57
Solvelink=%Solvelink.LoadLibrary%	16.21
GUSS	9.18

- Updates model data instead of matrix coefficients/rhs
- Hot start (keeps the model hot inside the solver and uses solver's best update mechanism)
- Saves model generation and solver setup time
- A priori knowledge of all scenario data

Calling GAMS from your Application



Creating Input for GAMS Model

→ Data handling using GDX API

Callout to GAMS

→GAMS option settings using Option API

→Starting GAMS using GAMS API

Reading Solution from GAMS Model

→ Data handling using GDX API

Low level APIs → Object Oriented API

- Low level APIs
  - GDX, OPT, GAMSX, GMO, …
  - High performance and flexibility
  - Automatically generated imperative APIs for several languages (C, Delphi, Java, Python, C#, …)

- Object Oriented GAMS API
  - Additional layer on top of the low level APIs
  - Object Oriented
  - Written by hand to meet the specific requirements of different Object Oriented languages

Features of the object oriented API

- No modeling capability, model is still written in GAMS
- Prepare input data and retrieve results in a convenient way → *GAMSDatabase*
- Control GAMS execution → *GAMSJob*
- Scenario Solving: Feature to solve multiple very similar models in a dynamic and efficient way.
  →*GAMSModelInstance*
- Seamless integration of GAMS into other programming environments

GAMSModelInstance etc.

GAMSJob

- Manages the execution of a GAMS program given by GAMS model source

creates

GAMSCheckpoint

- Captures the state of a `GAMSJob`

initializes

GAMSModelInstance

- A single mathematical model generated by a GAMS solve statement

modifies

GAMSModifier

- Marks elements of a `GAMSModelInstance` to be modifiable

Object Oriented GAMS API

```
do
{
    masteri.Solve(GAMSModelInstance.SymbolUpdateType.BaseCase);
    if (1 < iter)
        upperbound = masteri.SyncDB.GetVariable("zmaster").FirstRecord().Level;
    objmaster = masteri.SyncDB.GetVariable("zmaster").FirstRecord().Level - theta.FirstRecord().Level;
    received.Clear();
    foreach (GAMSVariableRecord r in masteri.SyncDB.GetVariable("received"))  {
        received.AddRecord(r.Keys).Value = r.Level;
        cutcoeff.AddRecord(iter.ToString(), r.Keys[0]);
    }
    cutconst.AddRecord(iter.ToString());
    double objsub = 0.0;
    foreach (GAMSSetRecord s in data.OutDB.GetSet("s"))
    {
        demand.Clear();
        foreach (GAMSSetRecord j in data.OutDB.GetSet("j"))
            demand.AddRecord(j.Keys).Value = scenarioData.FindRecord(s.Keys[0], j.Keys[0]).Value;
        subi.Solve(GAMSModelInstance.SymbolUpdateType.BaseCase);
        double probability = scenarioData.FindRecord(s.Keys[0], "prob").Value;
        objsub += probability * subi.SyncDB.GetVariable("zsub").FirstRecord().Level;
        foreach (GAMSSetRecord j in data.OutDB.GetSet("j"))
        {
            cutconst.FindRecord(iter.ToString()).Value += probability *
                subi.SyncDB.GetEquation("market").FindRecord(j.Keys).Marginal * demand.FindRecord(j.Keys).Value;
            cutcoeff.FindRecord(iter.ToString(), j.Keys[0]).Value += probability *
                subi.SyncDB.GetEquation("selling").FindRecord(j.Keys).Marginal;
        }
    }
    lowerbound = Math.Max(lowerbound, objmaster + objsub);
    iter++;
    if (iter == maxiter + 1)
        throw new Exception("Benders out of iterations");
} while ((upperbound - lowerbound) >= 0.001 * (1 + Math.Abs(upperbound)));
```


➔ Benders….sln

Object Oriented GAMS API

Setting	Solve time (secs)
Solvelink=0 (default)	294.36
Solvelink=%Solvelink.CallModule%	239.57
Solvelink=%Solvelink.LoadLibrary%	16.21
GUSS	9.18
C# Application	2.43

Excursus: SP with EMP

```
model transport /all/;
solve transport maximizing profit using lp;

file emp / '%emp.info%' /; put emp '* problem %gams.i%'/;
$onput
jrandvar demand('d1') demand('d2') demand('d3') demand('d4') demand('d5')
    0.25     150           100           250           300           600
    0.5      160           120           270           325           700
    0.25     170           135           300           350           800
stage 2 demand sales waste profit obj selling market
$offput
putclose emp;

Set scen Scenarios / s1*s3 /;
Parameter
   sc_demand(scen,j)        demand by scenario
   sc_sales(scen,j)         sales by scenario
   sc_waste(scen,j)         waste by scenario
   sc_profit(scen)          profit by scenario;

Set dict / scen          .scenario.''
           demand        .randvar. sc_demand
           sales         .level.   sc_sales
           waste         .level.   sc_waste
           profit        .level.   sc_profit /;

option emp=de;
solve transport maximizing profit using emp scenario dict;
display ship.l, sc_demand, sc_sales, sc_waste, sc_profit;
```

→ t_emp.gms

ACSOM

- Advanced Collaborative System Optimization Modeler
- Explore high-dimensional Pareto surface for configuration of (expensive) military vehicles
- Collaboration between:
  - General Dynamics Land Systems (GDLS)
  - Industrial & Systems Engineering, Wayne State University
  - GAMS Development Corp

Abrams M1A2


Remote weapons station
Loader's Armor Gun Shield
Loader's thermal sight
tank/infantry telephone
Thermal sight goggles
Rear protecting unit slat armor
Thermal sight components
Abrams Reactive Armore Tiles

The Whole-system Design Problem

21 SUBSYSTEMS

- AFES
- Armor
- Chassis Structure
- Crew Station
- Defensive Armament
- Dismountable
- ECS
- Fuel
- Hit Avoidance
- Lighting
- Mission Equipment (Shooter)
- Mission Station
- Mission Structure
- NBC
- Platform Electronics
- Power Distribution & Mgmt
- Propulsion
- Signature Management
- Suspension
- Turret Structure
- Water Management

Options per subsystem	Theoretically Possible Subsystem Combinations
2	2,092,152
3	10,460,353,203

Approved for Public Release, Distribution Unlimited, GDLS approved, log 2008-09, dated 03/19/08 4

Considers All to Make the Whole

Full Spectrum

Numerous Subsystems with a multitude of options for each

Force Protection Approach?

Two, Three, Four Crew Members?

Auto loader ?
- Configuration/Approach?

Hull Material?
- Aluminum?
- Steel?
- Titanium?

Which Core Data Network?

Hull Design?

Power ?
- What Type of Engine?
- What Type Transmission?

What Type of Suspension System?
- Torsion bars?
- HSUs?

Passive?
Fully Active?
Track?

Which Servo Motor Controller Architecture?

Approved for Public Release, Distribution Unlimited, GDLS approved, log 2008-09, dated 03/19/08 6

Algorithms

- Suite of Algorithms
  - Complete Enumeration
  - Acsom 1.2 algorithm
  - Integer Cut
  - WSU algorithms: ACE, UPSA
- Developed in
  - VB.Net
  - GAMS
    - GAMS Models and Algorithms
    - .NET API for recursive ACE algorithm
  - Cplex MIP Solver
  - MySQL

Aims For GAMS Models

- High Performance
- Use multiple cores
  - For current crop of laptops
  - For high-performance machines
  - Be prepared for advancing technologies (more parallelism)

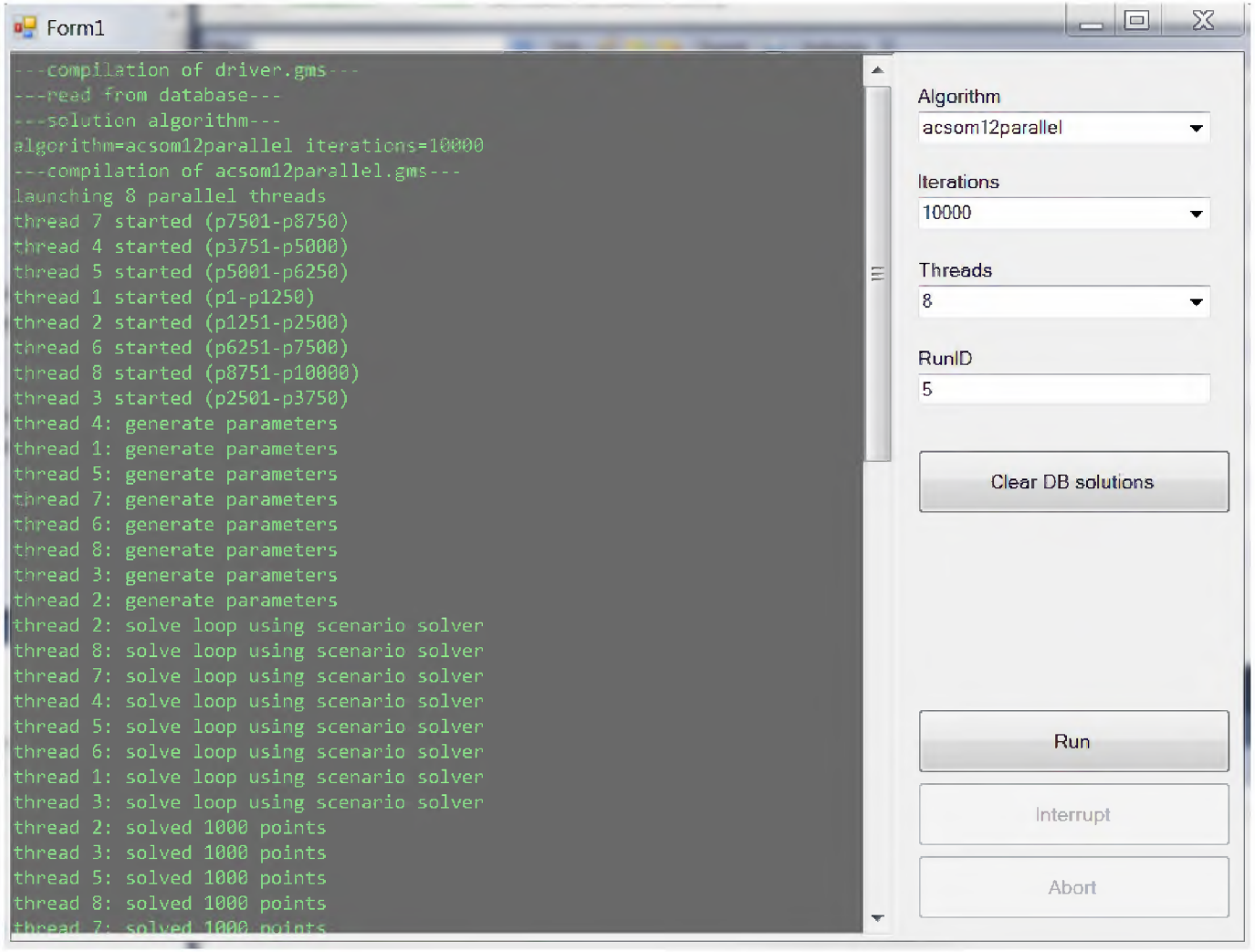
Modeling System Lessons

- Focus on whole system
  - High level modeling language can help improve design
  - Although slower than traditional programming languages, by using better design we can achieve higher performance
  - Examples:
    - Use parallel instances of scenario solver where possible
    - Use bulk SQL in one spot instead of record level SQL scattered over application

Example

- Reimplementation of Acsom 1.2 algorithm
- Old system:
  - VB.Net
  - MPL (single point)
  - Cplex
  - SQL throughout
- New System:
  - GAMS (complete algorithm; streamlined design)
  - Multiple parallel instances of scenario solver
  - Cplex
  - Bulk SQL just once
- Performance improvement
  - On the same hardware:
    - From hours and even days for large problems
    - To minutes, up to an hour

Test Environment



yetanothermathprogrammingconsultant.blogspot.de/2012/04/parallel-gams-jobs-2.html
Most Visited Getting Started Latest Headlines Yet Another Math Pro... Bookmarks

Yet Another Math Programming Consultant

So I am now a full time math programming consultant... I will try to post my (technical) notes here. Keeping a searchable list of them will make this useful for me in my daily life.

Sunday, April 15, 2012

Parallel GAMS jobs (2)

In http://yetanothermathprogrammingconsultant.blogspot.com/2012/04/parallel-gams-jobs.html I described a simple approach I suggested to a client allowing to run multiple scenarios in parallel.

For a different client we needed to run a randomized algorithm that solves many small MIP models. They are so small that using multiple threads inside the MIP solver does not give much performance boost (much of the time is spent outside the pure Branch & Bound part – such as preprocessing etc.). However as the MIP problems are independent of each other we could generate all the necessary data in advance and then call the scenario solver (http://www.gams.com/modlib/adddocs/gusspaper.pdf). This will keep the generated problem in memory, and does in-core updates, so we don't regenerate the model all the time.

When running the algorithm with n=100,000 MIP models we see the following performance. Note that besides the MIP models there is also a substantial piece of GAMS code that implements other parts of the algorithm.

Implementation	number of MIP models	solve time	rest of algorithm	total time
Traditional GAMS loop (call solver as DLL)	100,000	1068 sec	169 sec	1237 sec
Scenario Solver	100,000	293 sec	166 sec	459 sec

To get more performance I tried to run the scenario solver in parallel. That is not completely trivial as the solver has a number glitches (e.g. scratch files with fixed, hard coded names). I also run parts of the GAMS algorithm in parallel, but some parts had to be done in the master model after merging the results.

Implementation	number of MIP models	Worker threads	parallel sub-problem time	rest of algorithm (serial)	total time
Parallel + Scenario Solver	100,000	4	116 sec	67 sec	183 sec

The implementation does not win the beauty contest, but it could be developed quickly. For these larger

About Me

ERWIN KALVELAGEN

Mathematical Programming Consultant

View my complete profile

Home Page

Amsterdam Optimization Modeling Group LLC

Recent Comments

Since you are writing a wrapper. You might as well... - Friday, April 06, 2012 - Srikrishna Sridhar

Hi, Erwin. I´m working in a Vehicle Route Problem... - Friday, April 06, 2012 - HENRY

Dear M. Kalvelagen, We thank you for pointing out... - Tuesday, March 27, 2012 - Frédéric Gardi

Hi, Erwin!! I´m working on a vrp capacitated, and... - Sunday, March 25, 2012 - Juan Pablo

Agenda

- Introduction
- High Performance Prototypes
- Generic Algorithms

Indexed Model – Scalar Model

- **Model** = Model Instance

- Indexed Models
  - Algebraic Modeling Systems were invented to support this type of modeling

- Scalar Models:
  - Variables x(1), x(2), x(3), …
  - Equations e(1), e(2), e(3), …
  - Algorithm development
  - Automatic translation of model instance into other AML scalar: GAMS/Convert (AMPL, Lingo, Minopt, …)

Model Translation

lation server

```
Variables
MODEL:
[Obj
Posi
[e1]
Equa
[e2]
[e3]
[e4]
cost
[e5]
supp
[e6]
dema
@Fre
Mode
End
Solve tr
```

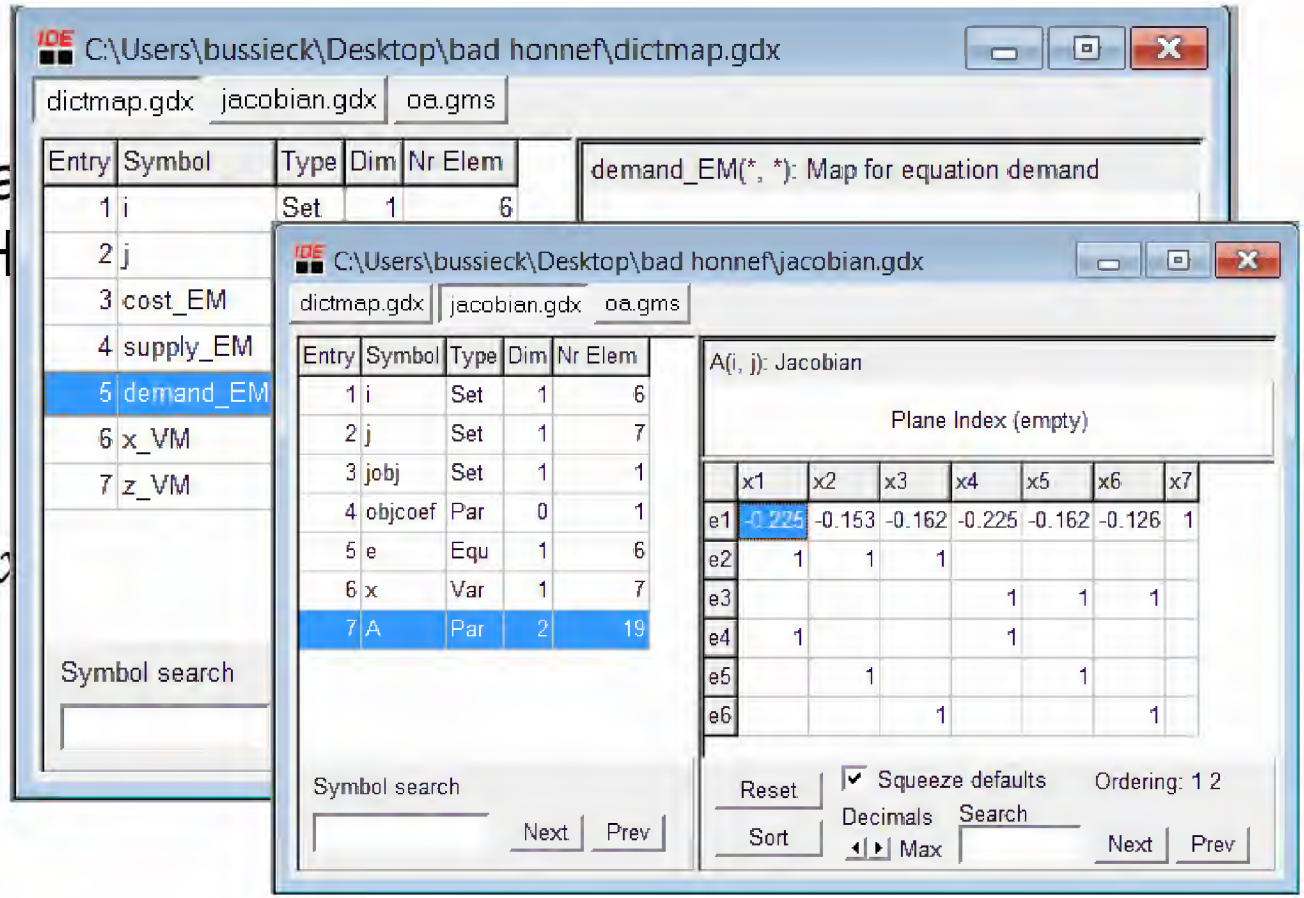
```
var x1 >= 0;
var x2 >= 0;
var x3 >
var x4 >
var x5 >
var x6 >

minimize
      + 0

subject

e2:     x

e3:     x

e4:     x

e5:     x

e6:     x
```

```
Variables  x1,x2,x3,x4,x5,x6,x7;

Positive Variables  x1,x2,x3,x4,x5,x6;

Equations  e1,e2,e3,e4,e5,e6;


e1..  - 0.225*x1 - 0.153*x2 - 0.162*x3 - 0.225*x4 - 0.162*x5 - 0.126*x6 + x7
      =E= 0;

e2..    x1 + x2 + x3 =L= 350;

e3..    x4 + x5 + x6 =L= 600;

e4..    x1 + x4 =G= 325;

e5..    x2 + x5 =G= 300;

e6..    x3 + x6 =G= 275;

Model m / all /;

Solve m using LP minimizing x7;
```

- No exe
- No file creation, i.e.

GAMS/Convert

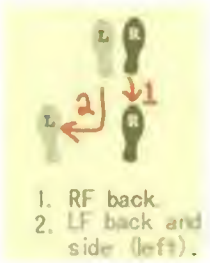
- Mapping Between Indexed and Scalar Model
  - DictMap:
- Access to Gra
  - Jacobian/H
- For example:
  - Non-linear
  - OA Cut at x

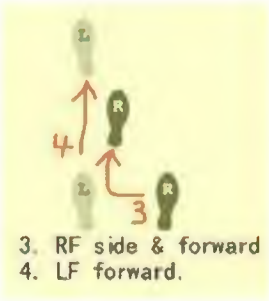


Research Applications

- Tom Rutherford:
  - EPEC:
    Equilibrium Problems with Equilibrium Constraints

- Francisco Trespalacios, Ignacio Grossmann:
  - **Basic Step** for Disjunctive (Non-linear) Programs
  - https://aiche.confex.com/aiche/2012/webprogram/Paper278434.html





The Problem of the Day

Mixed Integer Nonlinear Program (MINLP)

$$\begin{cases} \underset{x,y}{\text{minimize}} & f(x,y) \\ \text{subject to} & c(x,y) \leq 0 \\ & x \in X,\ y \in Y \text{ integer} \end{cases}$$

- f, c smooth (convex) functions
- X, Y polyhedral sets, e.g. $Y = \{y \in [0,1]^p \mid Ay \leq b\}$
- $y \in Y$ integer $\Rightarrow$ hard problem
- f, c *not* convex $\Rightarrow$ very hard problem

Outer Approximation (Duran and Grossmann, 1986)

Motivation: avoid *solving huge number* of NLPs

- Exploit MILP/NLP solvers: decompose integer/nonlinear part

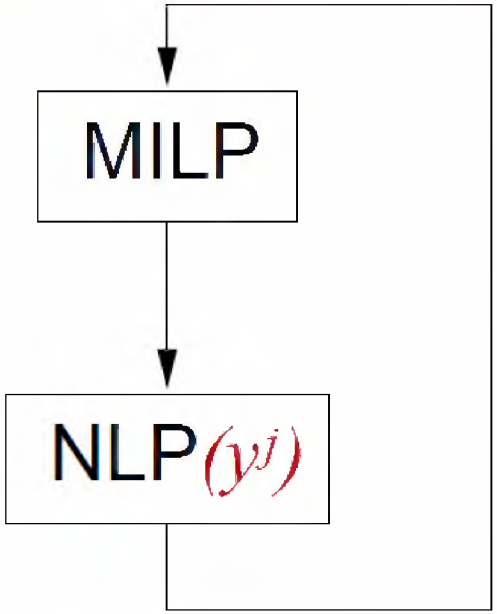
Key idea: reformulate MINLP as MILP (implicit)

- Solve alternating sequence of MILP & NLP

NLP subproblem y_j fixed:

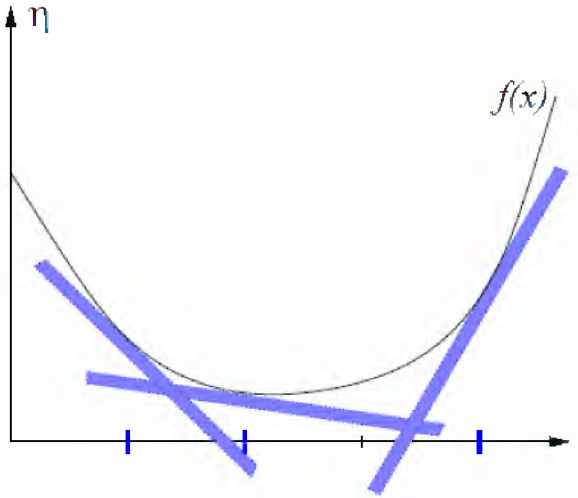
$$\text{NLP}(y_j) \begin{cases} \underset{x}{\text{minimize}} & f(x, y_j) \\ \text{subject to} & c(x, y_j) \leq 0 \\ & x \in X \end{cases}$$

Main Assumption: f, c are convex



Outer Approximation (Duran and Grossmann, 1986)

- let (x_j, y_j) solve NLP(y_j)
- linearize f, c about $(x_j, y_j) =: z_j$
- new objective variable $\eta \geq f(x, y)$
- MINLP $(P) \equiv$ MILP (M)

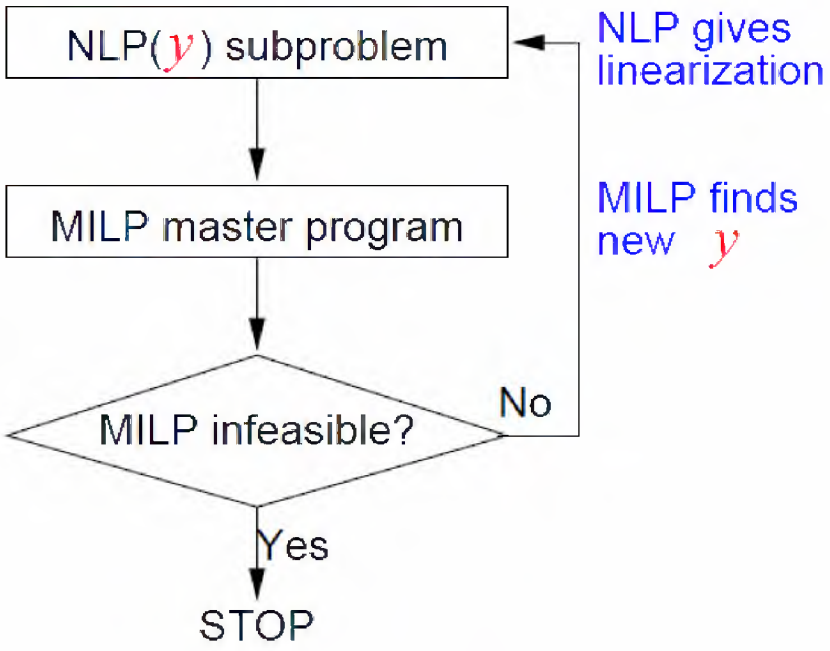


$$(M) \begin{cases} \underset{z=(x,y),\eta}{\text{minimize}} & \eta \\ \text{subject to} & \eta \geq f_j + \nabla f_j^T (z - z_j) \quad \forall y_j \in Y \\ & 0 \geq c_j + \nabla c_j^T (z - z_j) \quad \forall y_j \in Y \\ & x \in X,\ y \in Y \text{ integer} \end{cases}$$

SNAG: need *all* $y_j \in Y$ linearizations

Outer Approximation (Duran and Grossmann, 1986)

(M_k): lower bound (underestimate convex f, c)
NLP(y_j): upper bound U (fixed y_j)



$\Rightarrow$ stop, if lower bound $\geq$ upper bound

Simple OA Implementation in GAMS

- *Simple*: Convex + Binary variable only
- User help to identify
  - Non-linear constraints
  - Binary variables
- OA Implementation:
  - Scalar Model for (r)MINLP
    - Solve with fixed binary variables
    - Get gradients for OA cuts (Jacobian)
  - Scalar Model for MIP (lower bound)
    - Add OA Cuts
    - Cut off discrete solutions (MIP cuts)
  - Rewrite GAMS source code inserts for scalar model

Tools for Algorithm Development

- Matrix tools:
  - Cholesky
  - Eigenvalue
  - Eigenvector
  - Invert
- Others
  - Gdxrank (sorting)
  - csdp (SDP Solver)

Summary

- Rapid Prototype Development
  - GAMS Language Elements
  - High-performance (GUSS, Grid, …)
  - Other Execution Systems (.NET API)
  - Extended Mathematical Programming (EMP)
  - Scalar Models plus Toolbox to Build Algorithms Independent of a Particular Index Model

Thank You !

USA

GAMS Development Corp.
1217 Potomac Street, NW
Washington, DC 20007
USA

Phone: +1 202 342 0180
Fax: +1 202 342 0181

http://www.gams.com
sales@gams.com
support@gams.com

Europe

GAMS Software GmbH
Eupener Str. 135-137
50933 Cologne
Germany

Phone: +49 221 949 9170
Fax: +49 221 949 9171

http://www.gams.com
info@gams.de
support@gams.com