GAMS

Model Development – Using CHP as an example

Michael Bussieck mbussieck@gams.com

Jan-Hendrik Jagla jhjagla@gams.com

Lutz Westermann lwestermann@gams.com

GAMS Software GmbH

www.gams.de

GAMS Development Corporation

www.gams.com

Agenda

- GAMS – Basic Syntax
- Excursus: GDX
- Building a Model: CHP Generation Plant

Agenda

- GAMS – Basic Syntax
- Excursus: GDX
- Building a Model: CHP Generation Plant

GAMS Syntax: Declaration

- Sets

```
Sets
       i           canning plants    / seattle, san-diego /
       h           hours             / 1*24 /
       work(h)     hours of work     / 9*12, 14*17 /;
```

- Parameters

```
Parameters
       a(i)    capacity of plant i in cases
         /      seattle      350
                san-diego    600   /;

Table d(i,j)   distance in thousands of miles
                     new-york       chicago       topeka
      seattle           2.5            1.7           1.8
      san-diego         2.5            1.8           1.4   ;

Parameter
      pay(i,h) Payment per city and hour in $
               / seattle.9*12   9
                 seattle.14*17 11 /;
```

- Scalars

```
Scalar f  freight in dollars per case per thousand miles  /90/ ;
```

GAMS Syntax: Data Assignment using Sets

- General

```
Parameter c(i,j)  transport cost in thousands of dollars per case;
          c('seattle','chicago') = f * d('seattle','chicago') / 1000;
          c(i,j)                 = f * d(i,j)                 / 1000;
```

- Sum

```
Parameter daypay(i) Payment for a complete workday in $;
          daypay(i) = sum(h$work(h),pay(i,h));
          daypay(i) = sum(work(h),pay(i,h));
          daypay(i) = sum(work,pay(i,work));
```

- Product

```
scalar prodcap Product of all capacities;
       prodcap = prod(i,a(i));
```

- Minimum/Maximum

```
Scalar maxdem Maximum of all demands;
       maxdem = smax(j,b(j));

Scalar mindist Minimum of all distances;
       mindist = smin((i,j),d(i,j));
```

GAMS Syntax: Defined Elements of a Set

- Ord() and Card()

```
Set lasth(h) Last hour of the day;
    lasth(h) = (ord(h) = card(h));
    lasth(h)$(ord(h) = card(h)) = yes;
```

- Sameas(,)

```
Scalar demXny Demand in all markets except for New-York;
       demXny = sum(j$(not sameas(j,'new-york')), b(j));
```

GAMS Syntax: Variables

- Free (-∞ to ∞)

```
Variables           Z           Total transportation costs in thousands of dollars ;
```

- Positive (0 to ∞)

```
Positive Variable   X(i,j)      Shipment quantities in cases;
```

- Negative (-∞ to 0)

```
Negative Variable   Y(h)        Resource consumption;
```

- Integer (0, 1, 2, …)

```
Integer Variable    OUT(h)      Output;
```

- Binary (0 or 1)

```
Binary Variable     PRODUCE(i)  Decision whether to produce or not;
```

GAMS Syntax: Variables

- Semi continuous (0 or above certain value)

```
Semicont Variable SHIP(i,j)   Ship at least 100 tons;
                  SHIP.lo(i,j) = 100;
```

- Semi integer (0 or integer above certain value)

```
Semiint Variable  OUTP(i)     Produce at leat 12 units;
                  OUTP.lo(i) = 12;
```

- Special Ordered Sets Type 1 (Only one member in a set of variables can have nonzero value)

```
SOS1 Variable     PRODUCE(i)  Prodeuce at one location only;
```

- Special Ordered Sets Type 2 (Only two adjacent members in a set of variables can have nonzero value)

```
SOS2 Variable     WORKSCED(h) Scedule work so that 2 hours in series are assigned;
```

GAMS Syntax: Equations

- Definition

```
Equations
     cost          define objective function
     supply(i)     observe supply limit at plant i
     demand(j)     satisfy demand at market j ;
```

- Declaration

```
cost ..          Z  =e=  sum((i,j), c(i,j)*X(i,j));
supply(i) ..     sum(j, X(i,j))  =l=  a(i);
```

GAMS Syntax: Model Definition

- Model

```
Model transport /all/;
```

- Solver selection

```
option lp=coincbc;
```

- GAMS options

```
Option reslim  = 60;
Option iterlim = 100;
```

- Solver options

```
$onecho > cplex.opt
lpmethod = 4
$offecho
```

- Solve

```
Solve transport using lp minimizing z;
```

GAMS Syntax: Procedural Elements

- For

```
scalar scen;
for(scen=1 to 10 by 0.5,
   f      = 10*scen;
   c(i,j) = f * d(i,j) / 1000;
   Solve transport using lp minimizing Z;
   Display Z.l;);
```

- While

```
Scalar scen /1/;
while(scen<=10,
   f      = 10*scen;
   c(i,j) = f * d(i,j) / 1000;
   Solve transport using lp minimizing Z;
   scen = scen + 0.5;
);
```

- Loop
- If… else…

```
loop(h,
   if(work(h),
      pay(i,h)  = 0.6*pay(i,h);
   else
      pay(i,h)  = 1.5*pay(i,h);
   );
);

pay(i,h)  = 0.6*pay(i,h)$work(h)  + 1.5*pay(i,h)$(not work(h));
```

Mathematical Functions

Function	*Description*
`errorf(x)`	Integral of the standard normal distribution from $-\infty$ to x
`exp(x)`	Exponential, e^x
`log(x)`	Natural logarithm, $\log_e x$
`log10(x)`	Common logarithm, $\log_{10} x$
`normal(x,y)`	Random number normally distributed with mean x and standard deviation y
`uniform(x,y)`	Random number with uniform distribution between x and y
`abs(x)`	Absolute Value of x, i.e. $\lvert x \rvert$
`ceil(x)`	Ceiling of x. Smallest integer $\geq x$
`floor(x)`	Floor of x. Largest integer $\leq x$
`mapval(x)`	Mapping function. Assigns unique numbers to special values.
`max(x,y,..)`	Largest value among all arguments.
`min(x,y,..)`	Smallest value among all arguments
`mod(x,y)`	Remainder. $x - y * \texttt{trunc}(x/y)$
`power(x,y)`	Integer power. x^y, where y must be an integer
`round(x)`	round x to the nearest integer
`round(x,y)`	Rounds x to y decimal places right (+) or left (-) to the decimal point
`sign(x)`	Returns 1 if $x > 0$, -1 if $x < 0$, and 0 if $x = 0$
`sqr(x)`	Square of x. x^2
`sqrt(x)`	Square root of x. $\sqrt{x}$
`trunc(x)`	`sign(x)*floor(abs(x))`
`arctan(x)`	$\tan^{-1} x$. Result in radians
`cos(x)`	$\cos x$; x in radians
`sin(x)`	$\sin x$; x in radians

*not exhaustive

Compile Time vs. Execution Time

- Compile time arguments…
  - start with $
  - are executed when compiling a GAMS file
  - are e.g. $if, $set, $goto, $exit, $call …

- Execution time arguments…
  - are executed during the execution of the compiled GAMS file
  - are e.g. if, execute, solve, loop, …

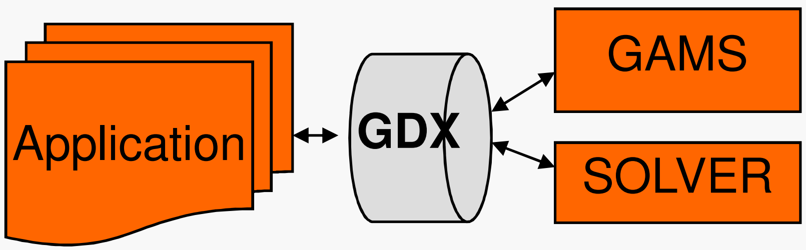
NOTE: When reading a model from top to bottom, we can see an execution time command before a compile time command, but the latter will be executed first.

Agenda

- GAMS – Basic Syntax
- Excursus: GDX
- Building a Model: CHP Generation Plant

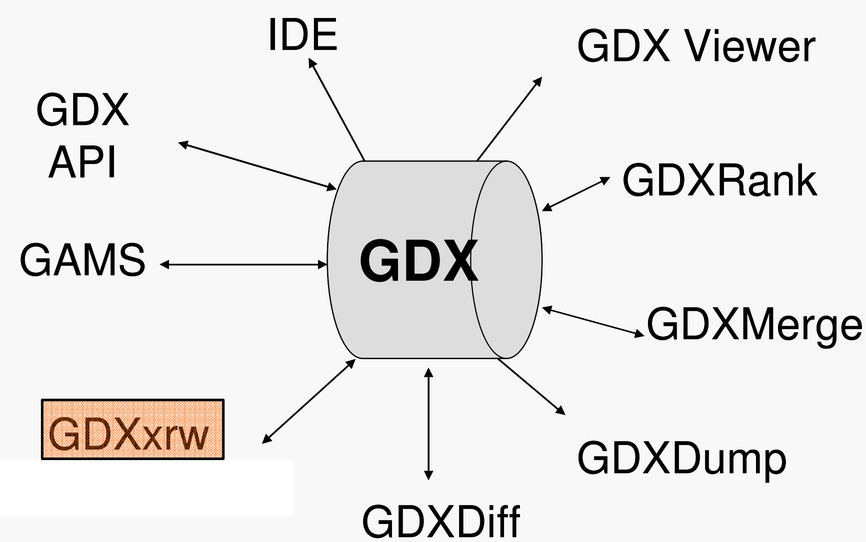
Gams Data eXchange

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 Excel connectivity
- General API
- Scenario Management Support

GDX Tools



Using GDXXRW to read from Excel

N19

	A	B	C	D	E	F	G	H	I
1	**Transportation**		New-York	Chicago	Topeka	Latitude	Longitude		
2	**Matrix**		NY	IL	KS				
3	Seattle	WA	2.5	1.7	1.8	47.45	122.3		
4	San-Diego	CA	2.5	1.8	1.4	32.82	117.13		
5									
6									
7									

```
Parameter d(i,j) distance in thousands of miles;
$call GDXXRW dist.xls par=d rng=dist!A1 rdim=1 cdim=1
$if errorlevel 1 $abort "Problem with file dist.xls!"
$gdxin dist
$load d
```

GAMS

Using GDXXRW to write to Excel

```
execute_unload 'ship' x;
execute 'GDXXRW ship.gdx var=x rng=ship!A1 rdim=1 cdim=1';
```

	A	B	C	D	E
1		new-york	chicago	topeka	
2	seattle	50	300		
3	san-diego	275		275	
4					
5					

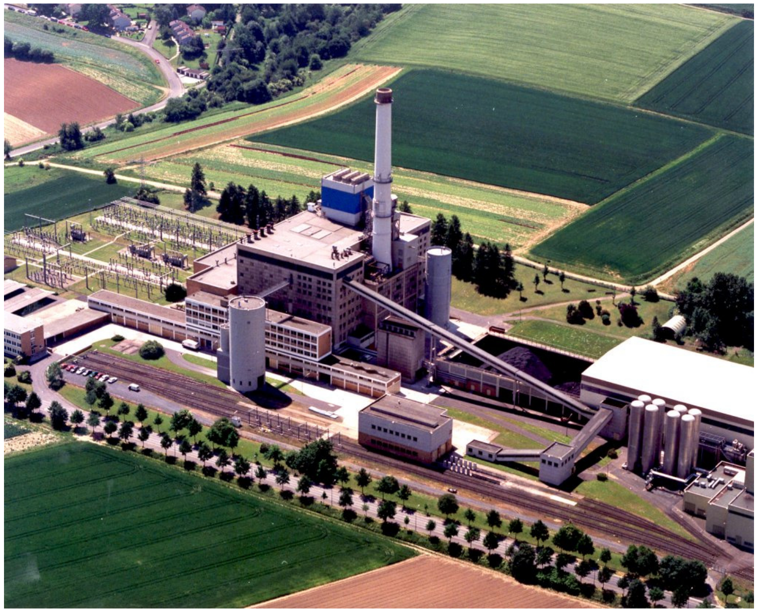
ship | Sheet1 | Sheet2 | Sheet3

Ready

Agenda

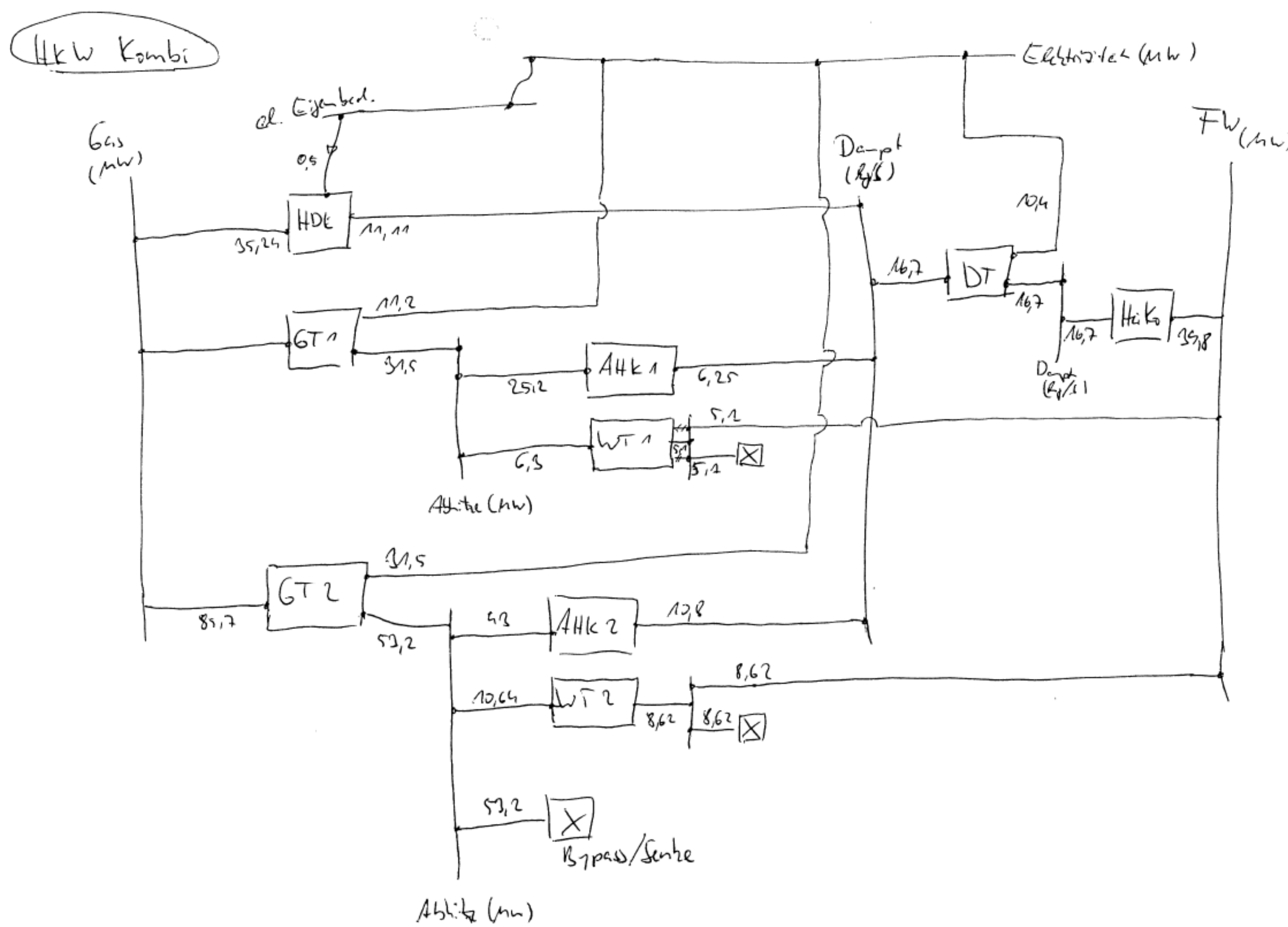
- GAMS – Basic Syntax
- Excursus: GDX
- Building a Model: CHP Generation Plant

Combined Heat and Power (CHP) Plant



- Produces heat and electricity in combination
- Certain demand of heat and electricity has to be satisfied
- Electricity can be traded at energy exchange
- Excess heat cannot be released, the demanded amount has to be generated exactly
- Cogeneration is subsidized by government

Process Model



Process Model

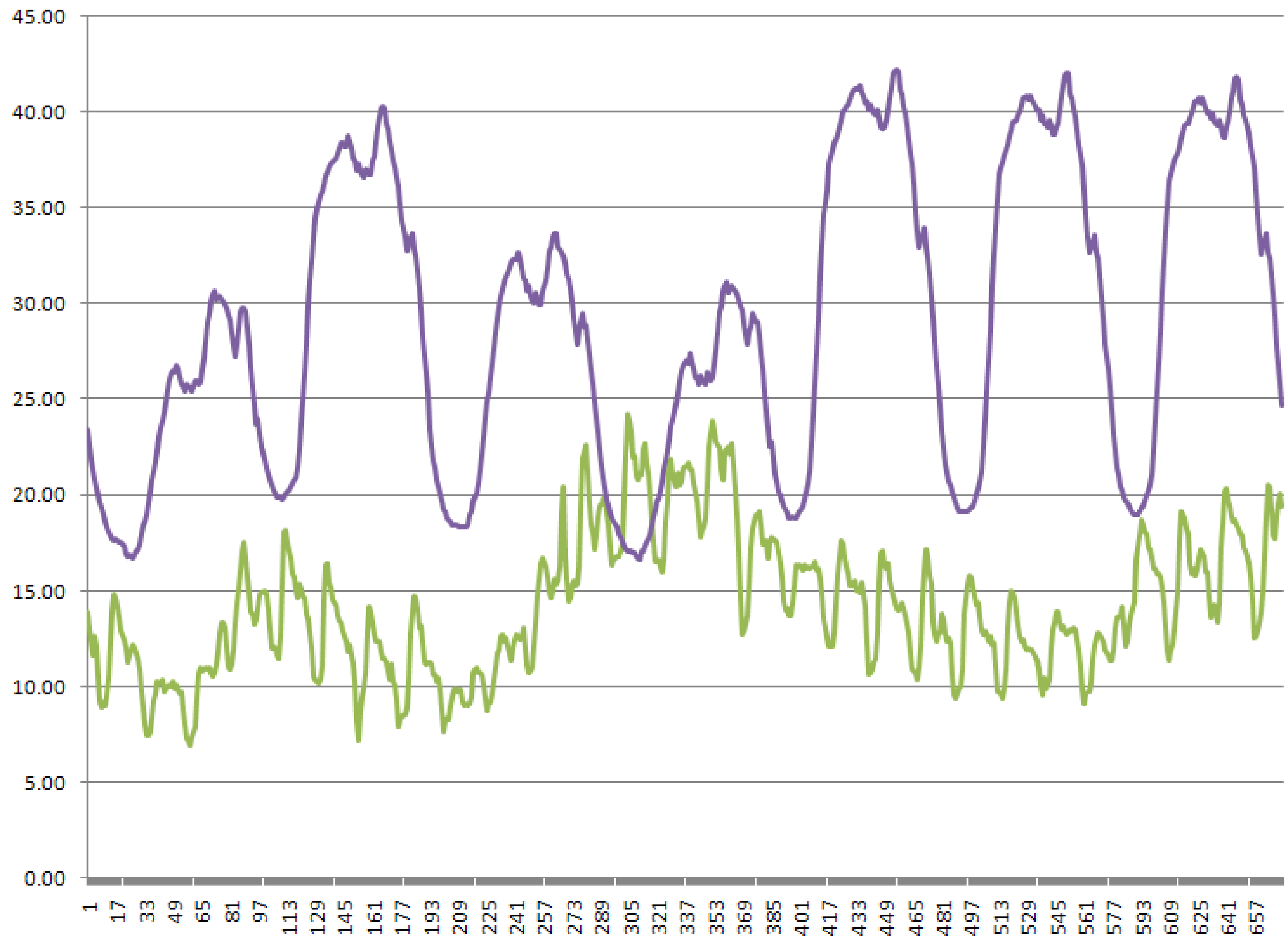
- External Inputs/Output
- Intermediate Commodities

	A	B	C	D	E	F	G	H	I	J
1		HTB	GT	HRB	HE	ST	HC			
2		1	1	1	1	1	1			
3	Gas	-39.6	-85.67							MWh
4	WasteHeat		53.2	-42.56	-10.64					MWh
5	Steam			10.72		-16.7				kg/s
6	Exhaust					16.7	-16.7			kg/s
7	Heat	25			8.6184		39.84			MWh
8	Electricity		29.55			10.39				MWh

Processes with minimum utilization level

	A	B	C	D	E	F	G	H	I	J	K	L
1		HTB	HTB	GT	GT	HRB	HE	ST	HC			
2		0	1	0	1	1	1	1	1			
3	Gas	0	-39.6	-54.69	-85.67							MWh
4	WasteHeat			36.43	53.2	-42.56	-10.64					MWh
5	Steam					10.72		-16.7				kg/s
6	Exhaust							16.7	-16.7			kg/s
7	Heat	0	25				8.6184		39.84			MWh
8	Electricity			13.77	29.55			10.39				MWh

Demand Electricity/Heat



Demand Electricity/Heat

	A	B	C	D
1		HeatDemand	ElectricityDemand	EXPrice
2	h1	13.93	23.48	44.78
3	h2	12.98	22.64	36.14
4	h3	12.35	21.91	31.02
5	h4	11.60	21.35	27.96
6	h5	12.58	20.81	26.61
7	h6	12.08	20.34	25.55
8	h7	10.41	19.97	16.00
9	h8	9.13	19.66	19.17
10	h9	8.88	19.38	26.00
11	h10	9.19	19.02	37.62
12	h11	8.96	18.66	46.83
13	h12	9.52	18.38	57.79
14	h13	10.43	18.14	53.98
15	h14	12.38	17.92	46.04
16	h15	13.97	17.74	41.19
17	h16	14.83	17.68	38.51
18	h17	14.64	17.69	46.46
19	h18	14.19	17.65	64.10
20	h19	13.61	17.56	71.51
21	h20	12.92	17.57	67.81
22	h21	12.46	17.35	55.24
23	h22	12.02	16.96	50.46
24	h23	11.27	16.79	57.59
25	h24	11.59	16.90	45.75

GAMS

Modeling Task

- Find cost minimal solution
  - Satisfy demand
  - Buy or make electricity
  - Subsidize cogeneration
  - Technical feasible schedule of plants
  - Investment decisions (new/upgraded power plants)
  - Economical aspects (e.g. shared ownership of plants)
  - …

Model Formulation Scheme

Minimize [Fuel costs]
+ [Costs/returns from electricity trading]
– [Bonus for cogeneration]

s.t. [Matter input] = [Matter output]

[Generated electricity] + [Purchased electricity]
= [Demand for electricity]

[Generated heat] = [Demand for heat]

Exercise 1: Add Steam Generator (SG)

- At maximum utilization:
  - Output: Steam 11 kg/s
  - Input: Coal 35.24 MWh
    Electricity 0.5 MWh

- At minimum utilization:
  - Output: Steam 5.5 kg/s
  - Input: Coal 17.62 MWh
    Electricity 0.25 MWh

- Coal costs: 12.23 $/MWh

- Reduces Output of HTB: 25 MWh → 18 MWh

Exercise 2: Add Heat Bypass (HB)

- Consumes up to 53.2 MWh wasteheat
- Cools it down at costs of 4 $ per MWh
- No relevant output

Exercise 3: Add Heat Storage Tank (HST)

- Maximum capacity of 50 MW heat
- At most 15 MW per hour input
- At most 12 MW per hour output
- “Pump” heat into tank costs 0.05 $ per MW
- 2% of stored heat gets lost per h

GAMS

Heat Storage Tank

Before:

$$dem_{Heat}(h) = GEN_{Heat}(h)$$

After:

$$dem_{Heat}(h) = GEN_{Heat}(h) + HOUT(h) - HIN(h)$$

$$HLVL(h) = HLVL(h-1) \cdot 0.98 - HOUT(h) + HIN(h)$$

$$HLVL(h) \leq 50$$

$$HIN(h) \leq 15$$

$$HOUT(h) \leq 12$$

Exercise 4: Limiting Number of GT Starts

- Startup costs:
  - GT: 500 $
  - SG: 1000$
- GT may be turned on not more than 8 times during modeled time frame

$$ONOFF(h,p)=1 \wedge ONOFF(h-1,p)=0 \Rightarrow STARTUP(h,p)=1$$

$$\rightarrow STARTUP(h,p) \geq ONOFF(h,p)-ONOFF(h-1,p)$$

$$\sum_h STARTUP(h,'GT') \leq 8$$

Exercise 5: Add “cool down” time for GT

- GT has to stay off for at least 8 hours when shut down

$$ONOFF(h,p)=0 \wedge ONOFF(h-1,p)=1 \Rightarrow SHUTDOWN(h,p)=1$$

$$\rightarrow SHUTDOWN(h,'GT') \geq ONOFF(h-1,'GT') - ONOFF(h,'GT')$$

$$SHUTDOWN(h,'GT')=1$$

$$\Rightarrow STARTUP(h_2,'GT')=0 \mid h \leq h_2 < h+8$$

$$\rightarrow \sum_{h_2 \mid h_2 \geq h \wedge h_2 < h+8} STARTUP(h_2,'GT') \leq 1 - SHUTDOWN(h,'GT')$$

Demo: Calling GAMS from MS Excel

	SG	SG	HTB	GT	GT	HRB	HE	HB	ST	HC	
	0	1	1	0	1	1	1	1	1	1	
Gas			-39.6	-54.69	-85.67						MWh
Coal	-17.62	-35.24									MWh
WasteHeat				36.43	53.2	-42.56	-10.64	-53.2			MWh
Steam	5.5	11				10.72			-16.7		kg/s
Exhaust									16.7	-16.7	kg/s
Heat			18				8.6184			39.84	MWh
Electricity	-0.25	-0.5		13.77	29.55				10.39		MWh

GAMS Directory:	c:\program files\gams22.8\
Working Directory:	c:\tmp\
Solver:	CPLEX

- COINIPOPT
- CONOPT
- CPLEX
- KNITRO
- LINDOGLOBAL
- MINOS
- MOSEK
- XPRESS

SOLV

GAMS on the Web

Download	www.gams.de	www.gams.com

Help and Support

Support Wiki	http://support.gams-software.com
Interfaces Wiki	http://interfaces.gams-software.com
McCarl‘s News	http://www.gams.com/maillist/newsletter.htm
User Group	http://www.gams.com/maillist/gams_l.htm
Google Group	http://groups.google.de/group/gamsworld

Search all GAMS Websites http://www.gams.com/search.htm

Contacting GAMS

<u>Europe</u>

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

Phone: +49 221 949 9170
Fax: +49 221 949 9171
http://www.gams.de

info@gams.de
support@gams-software.com

<u>USA</u>

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