



# From Model to App

---

**Develop and Deploy your GAMS Models**

**Robin Schuchmann**

GAMS Software GmbH

Software used in this talk:

GAMS: [www.gams.com/download/](http://www.gams.com/download/)

GAMS MIRO Desktop: [www.gams.com/miro/download.html](http://www.gams.com/miro/download.html)

# GAMS at a Glance

# Company

- Roots: World Bank, 1976
- Went commercial in 1987
- Locations
  - GAMS Development Corporation (USA)
  - GAMS Software GmbH (Germany)
- Product
  - The **G**eneral **A**lgebraic **M**odeling **S**ystem

# Modeling Basics

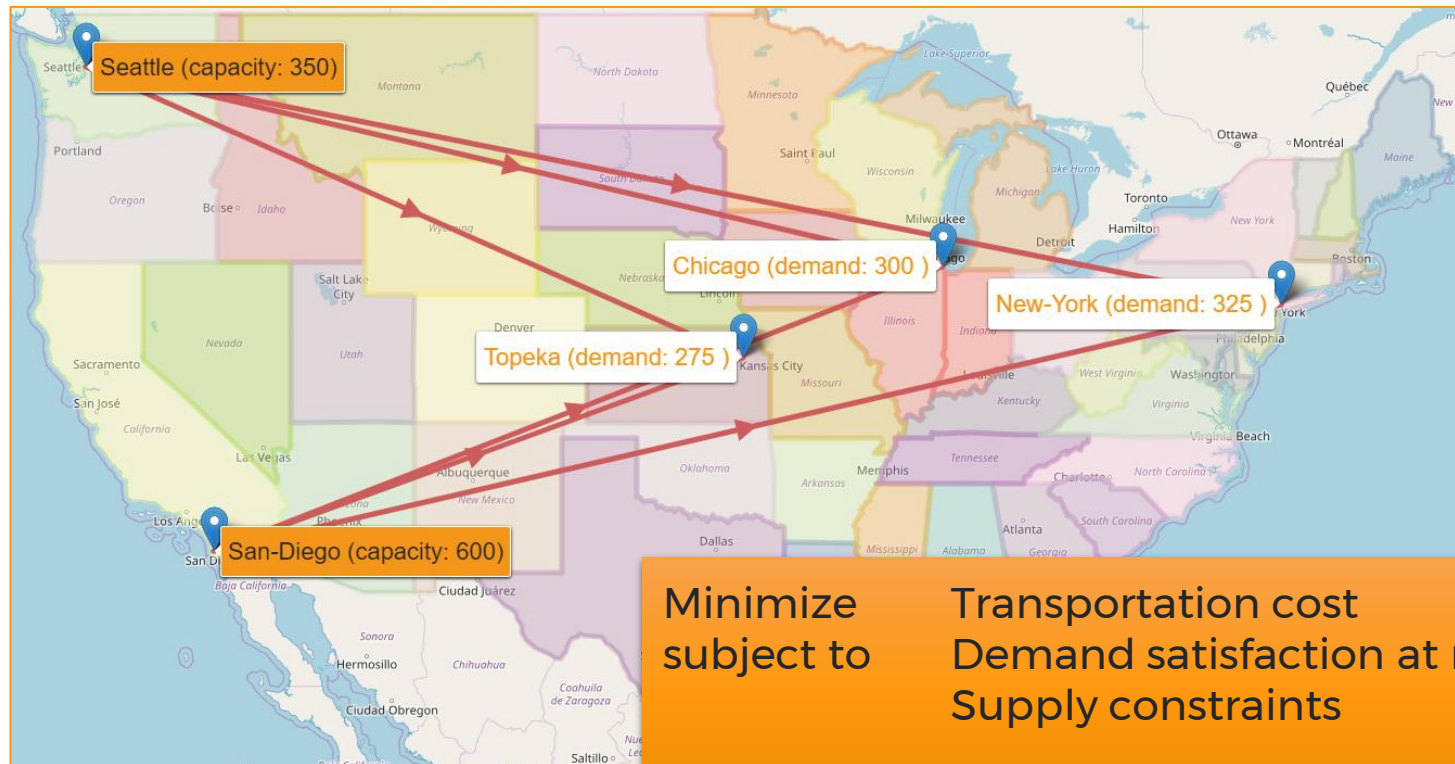
Developing a GAMS Model

# A Simple Transportation Problem

Canning Plants (supply)

shipments  
(Number of cases)

Markets (demand)



Minimize  
subject to

Transportation cost  
Demand satisfaction at markets  
Supply constraints

Freight: \$90 case / thousand miles

# Modeling Basics

Indices:

$i$  = plants

$j$  = markets

Given Data:

$a_i$  = supply of commodity of plant  $i$  (in cases)

$b_j$  = demand for commodity at market  $j$

$c_{ij}$  = cost per unit shipment between plant  $i$  and market  $j$

Decision Variables:

$x_{ij}$  = amount of commodity to ship from plant  $i$  to market  $j$

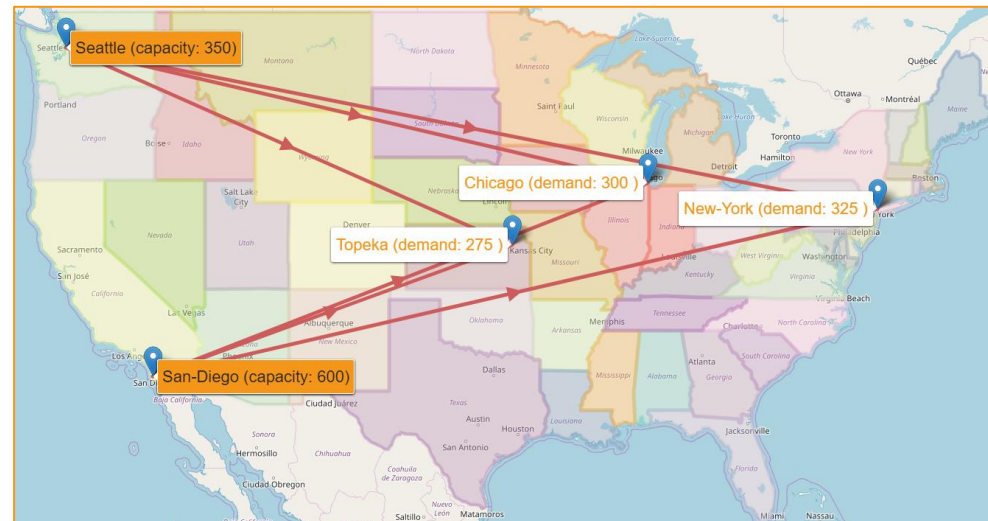
where  $x_{ij} \geq 0$ , for all  $i, j$

Constraints:

Observe supply limit at plant  $i$ :  $\sum_j x_{ij} \leq a_i$  for all  $i$  (cases)

Satisfy demand at market  $j$ :  $\sum_i x_{ij} \geq b_j$  for all  $j$  (cases)

Objective Function: Minimize  $\sum_i \sum_j c_{ij} x_{ij}$  (\$K)



# Modeling Basics

Indices:

$i$  = plants  
 $j$  = markets

Given Data:

$a_i$  = supply of commodity of plant  $i$  (in cases)  
 $b_j$  = demand for commodity at market  $j$   
 $c_{ij}$  = cost per unit shipment between plant  $i$  and market  $j$

Decision Variables:

$x_{ij}$  = amount of commodity to ship from plant  $i$  to market  $j$   
where  $x_{ij} \geq 0$ , for all  $i, j$

Constraints:

Observe supply limit at plant  $i$ :  $\sum_j x_{ij} \leq a_i$  for all  $i$  (cases)  
Satisfy demand at market  $j$ :  $\sum_i x_{ij} \geq b_j$  for all  $j$  (cases)  
Objective Function: Minimize  $\sum_i \sum_j c_{ij} x_{ij}$  (\$K)

Set

```
i canning plants / seattle, san-diego /  
j markets / new-york, chicago, topeka /;
```

Parameter

```
a(i) capacity of plant i in cases  
b(j) demand at market j in cases  
c(i,j) transport cost in thousands of dollars per case ;
```

Variable

```
x(i,j) shipment quantities in cases  
z total transportation costs in thousands of dollars ;
```

Equation

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

```
cost .. z =e= sum((i,j), c(i,j)*x(i,j));  
supply(i) .. sum(j, x(i,j)) =l= a(i);  
demand(j) .. sum(i, x(i,j)) =g= b(j);
```

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

```
solve transport using lp minimizing z;
```

# Modeling Basics

## Indices:

$i$  = plants  
 $j$  = markets

## Given Data:

$a_i$  = supply of commodity of plant  $i$  (in cases)  
 $b_j$  = demand for commodity at market  $j$   
 $c_{ij}$  = cost per unit shipment between plant  $i$  and market  $j$

## Decision Variables:

$x_{ij}$  = amount of commodity to ship from plant  $i$  to market  $j$   
where  $x_{ij} \geq 0$ , for all  $i, j$

## Constraints:

Observe supply limit at plant  $i$ :  $\sum_j x_{ij} \leq a_i$  for all  $i$  (cases)  
Satisfy demand at market  $j$ :  $\sum_i x_{ij} \geq b_j$  for all  $j$  (cases)  
Objective Function: Minimize  $\sum_i \sum_j c_{ij} x_{ij}$  (\$K)



## Set

```
i canning plants / seattle, san-diego /  
j markets / new-york, chicago, topeka /;
```

## Parameter

```
a(i) capacity of plant i in cases  
b(j) demand at market j in cases  
c(i,j) transport cost in thousands of dollars per case ;
```

## Variable

```
x(i,j) shipment quantities in cases  
z total transportation costs in thousands of dollars ;
```

[www.gams.com/products/solvers/](http://www.gams.com/products/solvers/)

## Equation

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

```
cost .. z =e= sum((i,j), c(i,j)*x(i,j));  
supply(i) .. sum(j, x(i,j)) =l= a(i);  
demand(j) .. sum(i, x(i,j)) =g= b(j);
```

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

```
Option LP = CBC;
```

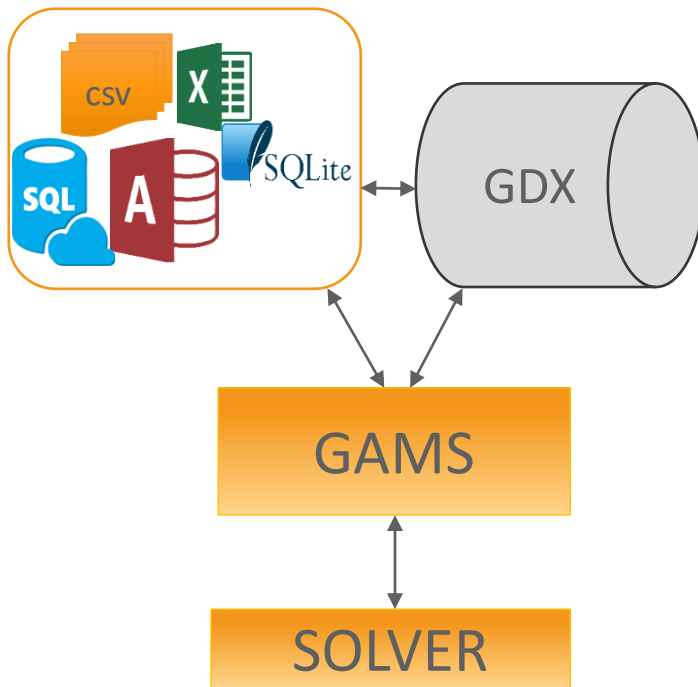
```
solve transport using lp minimizing z;
```

## Independence of

- ✓ Model and Operating System
- ✓ Model and Solver



# Independence of Model and Data



```
Set
  i canning plants / seattle, san-diego /
  j markets        / new-york, chicago, topeka /;

Parameter
  a(i)  capacity of plant i in cases
  b(j)  demand at market j in cases
  c(i,j) transport cost in thousands of dollars per case ;

Variable
  x(i,j) shipment quantities in cases
  z      total transportation costs in thousands of dollars ;

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

cost ..      z =e= sum((i,j), c(i,j)*x(i,j));
supply(i) .. sum(j, x(i,j)) =l= a(i);
demand(j) .. sum(i, x(i,j)) =g= b(j);

Model transport / all /;

solve transport using lp minimizing z;
```

## Independence of

- ✓ Model and Operating System
- ✓ Model and Solver
- ✓ Model and Data

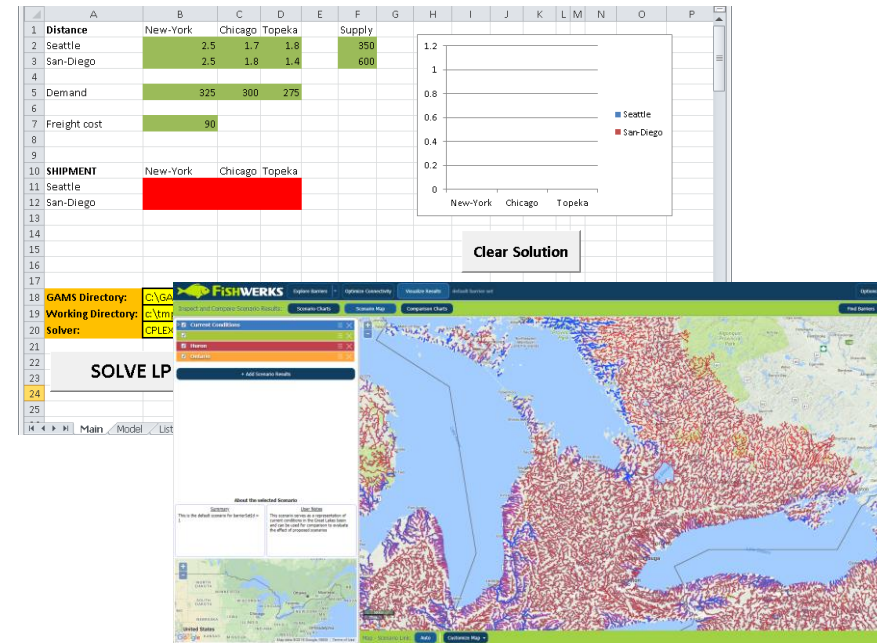
# Independence of Model and User Interface

## APIs

- Expert level APIs: high performance and flexibility
  - Object Oriented APIs (Python, Java, C++, ...)  
to develop applications
- Programming required to build applications

## GAMS MIRO

- Deployment environment for GAMS models
- Configuration instead of programming



## Independence of

- ✓ Model and Operating System
- ✓ Model and Solver
- ✓ Model and Data
- ✓ Model and User Interface

**What is GAMS MIRO?**

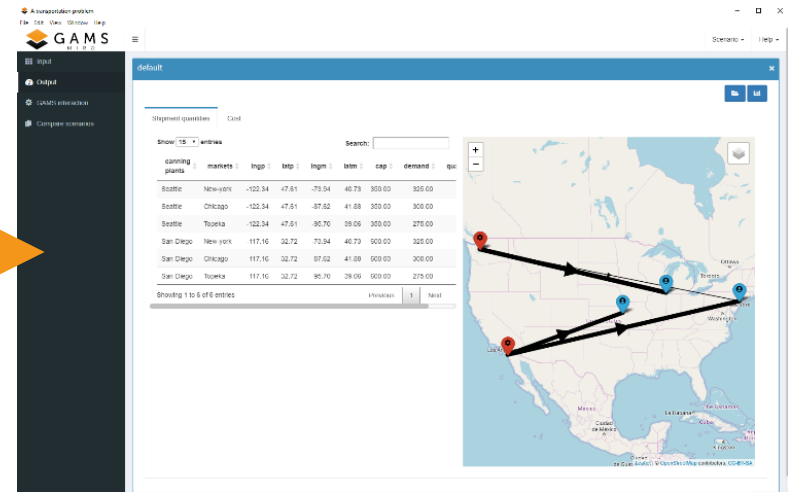
# GAMS MIRO

## Model Interface with Rapid Orchestration

```

11 This transportation is described as detail on
12 Homotopy, 6.4; Chapter 2.9 GAMS/Transport, in GAMS: A User's Guide.
13 The Bibliography Entry, Belmont City, California, 1995.
14
15 The line numbers will not match those in the book because of line
16 comments.
17
18 Expanded linear programming, transportation problem, scheduling
19 M2Text
20
21 Parameter
22 1 'existing plants' / seattle, hawkins /
23 2 'markets' / new-york, chicago, topeka /;
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

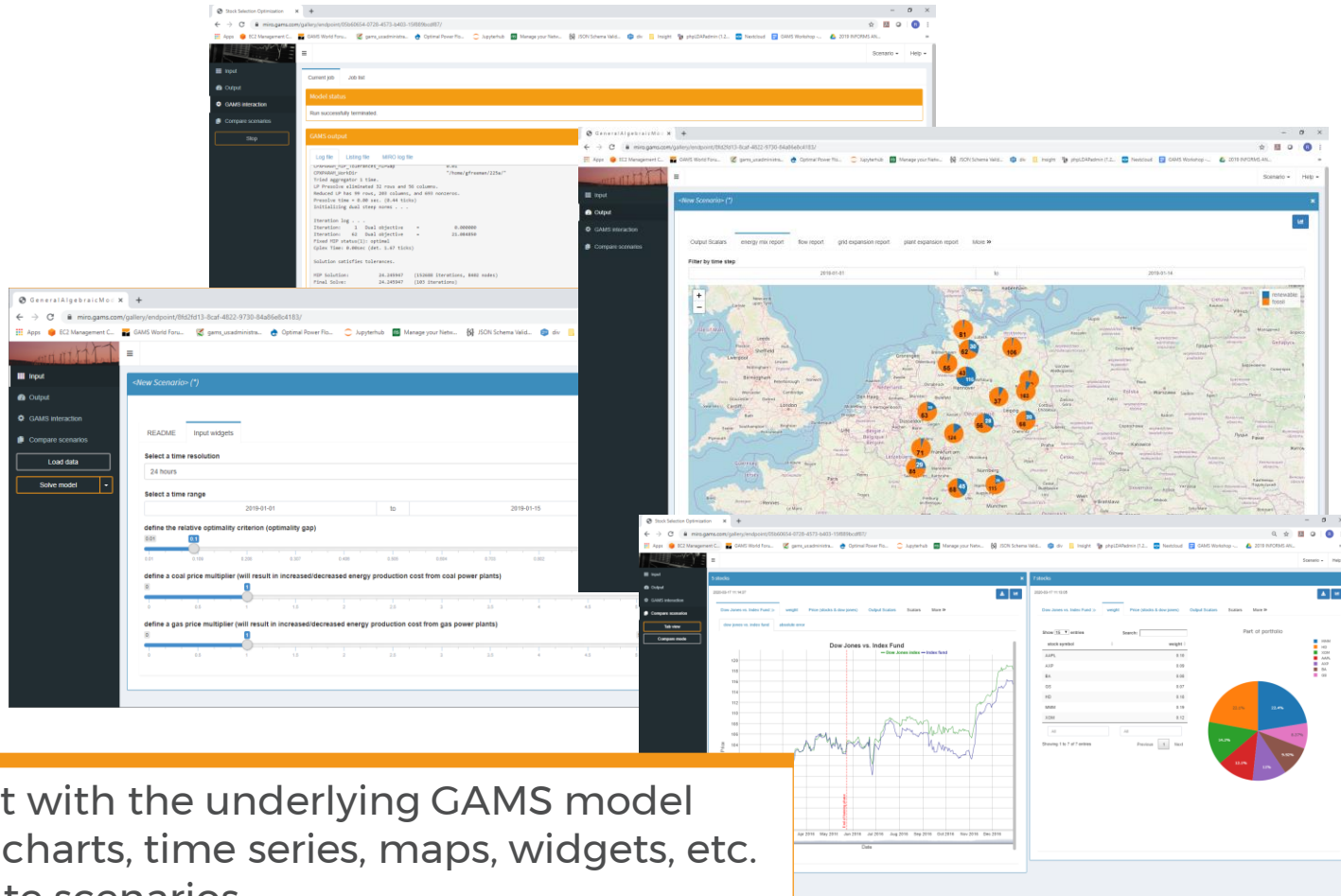


Observe supply limit at plant  $i$ :  $\sum_j x_{ij} \leq a_i$  for all  $i$  (cases)  
 Satisfy demand at market  $j$ :  $\sum_i x_{ij} \geq b_j$  for all  $j$  (cases)  
 Objective Function: Minimize  $\sum_i \sum_j c_{ij} x_{ij}$  (\$K)

- ✓ Interactive interface for GAMS models
- ✓ Usage via web browser
- ✓ GAMS as a black box
- ✓ Automatic deployment

# GAMS MIRO

## Model Interface with Rapid Orchestration



- ✓ Interact with the underlying GAMS model
- ✓ Create charts, time series, maps, widgets, etc.
- ✓ Generate scenarios
- ✓ Compare results

# From model to app

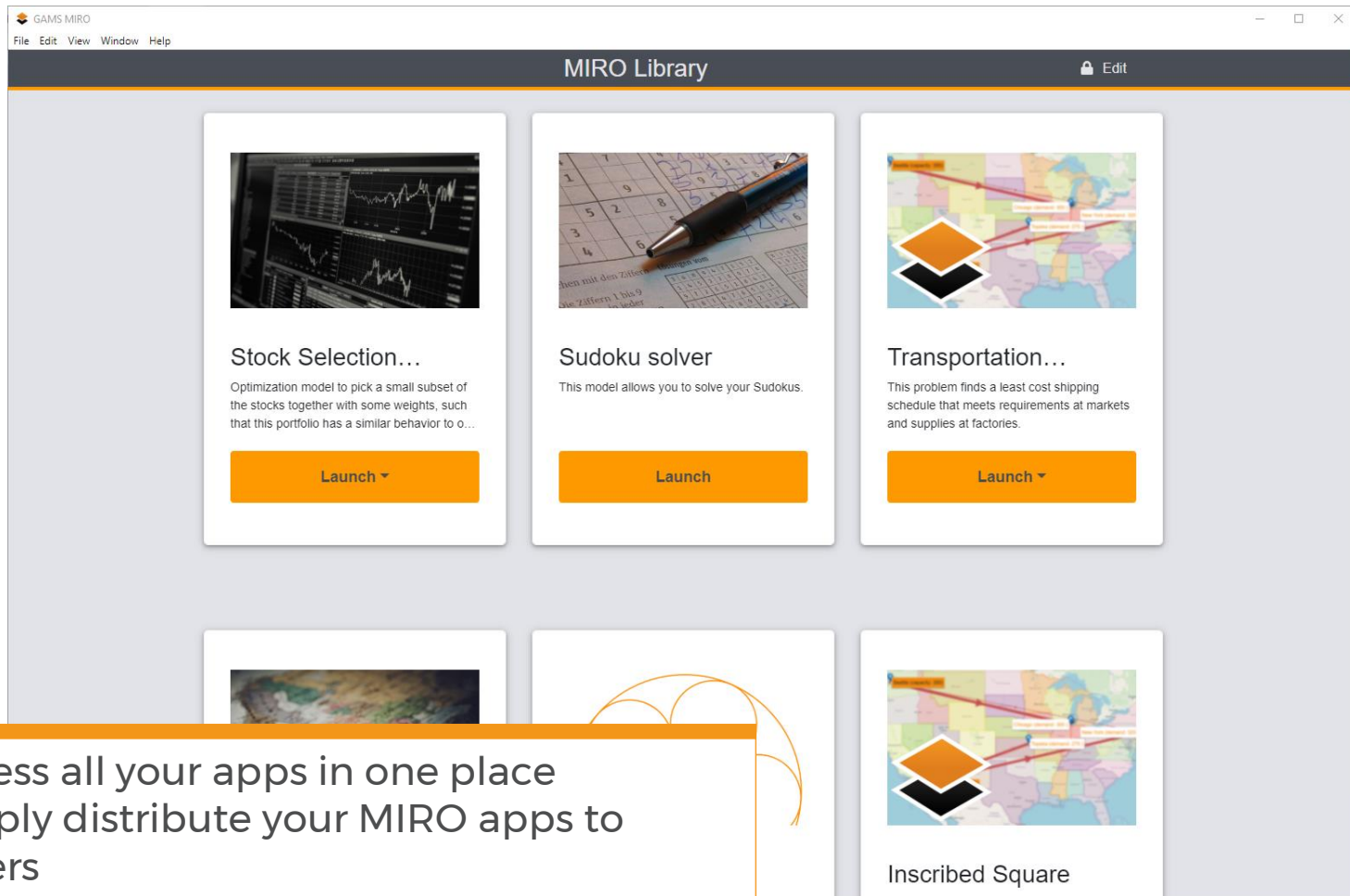
Hands-on

Software used in this talk:

GAMS: [www.gams.com/download/](http://www.gams.com/download/)

GAMS MIRO Desktop: [www.gams.com/miro/download.html](http://www.gams.com/miro/download.html)

# Deploying a MIRO app

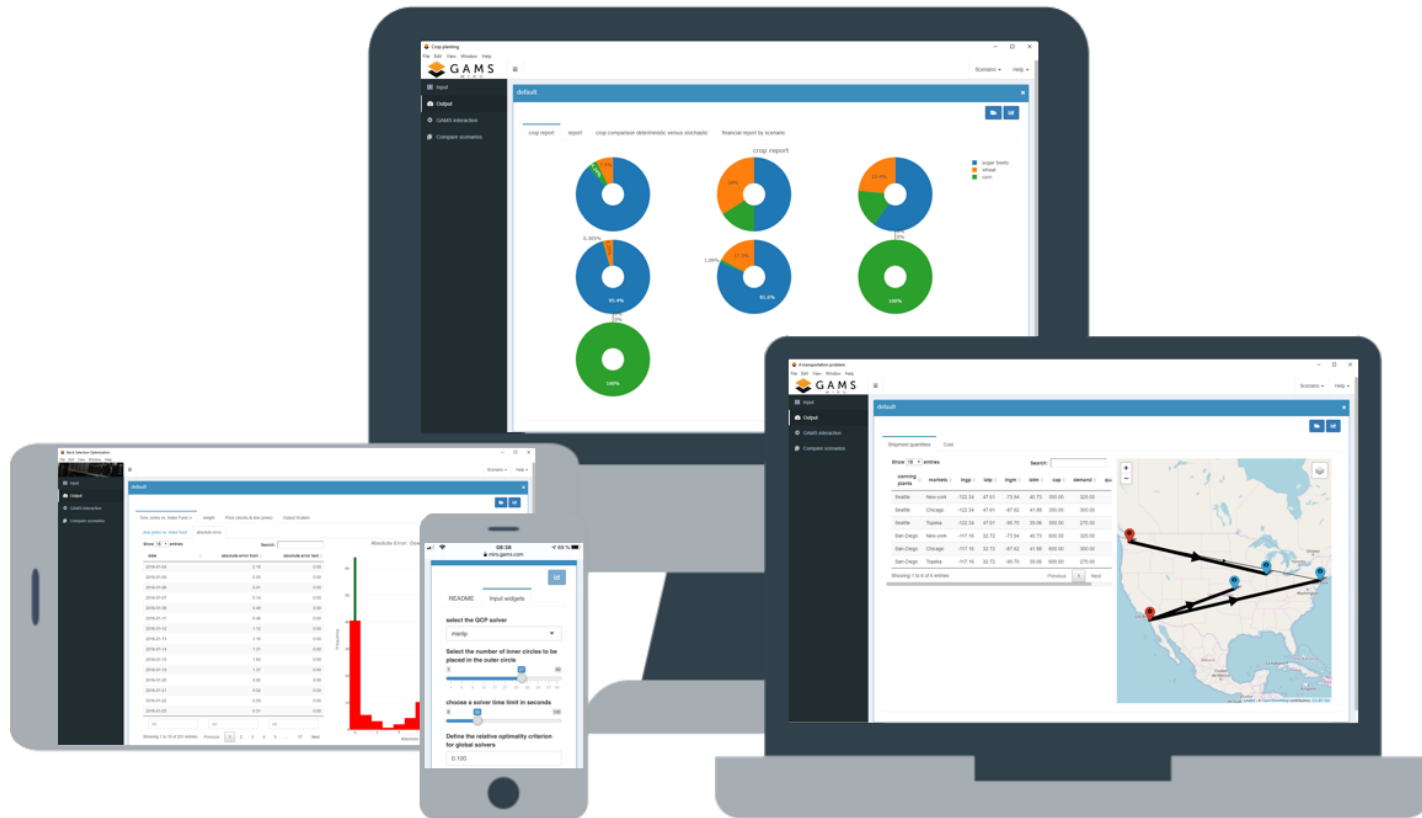


The screenshot shows the MIRO Library application window. The title bar reads "GAMS MIRO" and the menu bar includes "File", "Edit", "View", "Window", and "Help". The main header is "MIRO Library" with an "Edit" button. The interface displays a grid of application cards. Each card features a representative image, a title, a brief description, and a "Launch" button. The visible cards are:

- Stock Selection...**: Optimization model to pick a small subset of the stocks together with some weights, such that this portfolio has a similar behavior to o...
- Sudoku solver**: This model allows you to solve your Sudokus.
- Transportation...**: This problem finds a least cost shipping schedule that meets requirements at markets and supplies at factories.
- Inscribed Square**: (Partially visible at the bottom right)

- ✓ Access all your apps in one place
- ✓ Simply distribute your MIRO apps to others
- ✓ Add other people's apps to your library

# GAMS MIRO Engine: Optimizing in the Cloud



- ✓ Access via local apps or directly on a server
- ✓ High scalability
- ✓ Load balancing
- ✓ Rolling updates
- ✓ ...

[www.gams.com/miro/engine/](http://www.gams.com/miro/engine/)





For more information visit:

[www.gams.com](http://www.gams.com)

[www.gams.com/miro](http://www.gams.com/miro)

**Additional material**

# What the Solve Statement is doing

At each **Solve** statement, the following happens:

1. The model (= list of (indexed) equations) is compiled into a **model instance**, that is, a scalar (= no indices) list of constraints and those variables, that appear in at least one constraint. (The **Equation Listing** and **Column Listing** shows in the listing file shows parts of this model.)
2. Instance statistics are written to the log:

```
--- Generating NLP model m
--- alky1.gms(85) 5 Mb
---   8 rows 15 columns 32 non-zeroes
---   54 nl-code 19 nl-non-zeroes
```
3. Some error check is performed.  
The instance is passed on to a solver and processed there.

```
--- Executing IPOPT: elapsed 0:00:00.093
[solver log]
--- Restarting execution
```
4. Passed to a solver and processed there.
5. The result (model and solution status, solution, statistical information) is passed back to GAMS and reported in the listing file.

# GAMS Log

The GAMS log can be written to the console, to standard output and/or to a file. This is controlled by the command line parameter logOption (or lo). The following items (and more) are part of the log:

- **GAMS version**

```
*** ***** BETA release
*** GAMS Base Module 24.5.0 r53642 BETA Released 25Aug15 WEI x86 64bit/MS Windows
*** ***** BETA release
```

- **License**

```
Licensee: Max Mustermann                               G141124/0001AW-GEN
GAMS Software GmbH                                    DC8674
```

- **Problem statistics**

```
-- 2 rows 3 columns 5 non-zeroes
-- 2 discrete-columns
```

- **Solver log**

```
Cplex 12.8.0.0
```

```
Reading data...
Starting Cplex...
```

- **Results**

```
MIP Solution:      8.000000 (0 iterations, 0 nodes)
Final Solve:       8.000000 (0 iterations)
Best possible:     8.000000
Absolute gap:      0.000000
Relative gap:      0.000000
```

# Listing File

Running a GAMS model generates a listing file (.lst file).

## Compilation Errors:

- are indicated by **\*\*\*\***
- contain a '\$' **directly below** the point at which the compiler thinks the error occurred
- are **explained near the end** of the line-numbered listing part
- in the IDE, they are also indicated by **red lines in the process (log) window** (can be double-clicked)
- check carefully for the cause of the **first error**, fix it, and try again
- usual causes: undefined / undeclared symbols (parameters, variables, equations), unmatched brackets, missing semi-colons

# Listing File: Equation and Column Listing

## Equation Listing:

- listing of generated equations with **sets unrolled, parameters removed**, ...
- useful for **model debugging**: is the intended model generated?
- for nonlinear equations, a **linearization in the starting point** is shown

```
AcidDef.. AcidDilut*AcidErr =e= 35.82-22.2*F4Perf;
```

```
-> AcidDef.. (1)*AcidDilut + 22.2*F4Perf + (3.6)*aciderr  
           =E= 35.82 ; (LHS = 35.79, INFES = 0.03 ****)
```

- **activity and violation** of constraint in starting point also shown
- **Column Listing**:
- shows coefficients, bounds, starting values for generated variables

```
-- F4Perf F4 Performance Number
```

```
F4Perf
```

```
      (.LO, .L, .UP, .M = 1.45, 1.45, 1.62, 0)
```

```
22.2  AcidDef
```

```
(1)   F4Def
```

# Listing File: Solve Summary

- generated for each solve command
- reporting status and result of solve

```

                S O L V E          S U M M A R Y

MODEL      m                OBJECTIVE F
TYPE      NLP              DIRECTION MINIMIZE
SOLVER    CONOPT           FROM LINE 85

**** SOLVER STATUS      1 Normal Completion
**** MODEL STATUS      2 Locally Optimal
**** OBJECTIVE VALUE          -1.7650

RESOURCE USAGE, LIMIT      0.006          1000.000
ITERATION COUNT, LIMIT    16          2000000000
EVALUATION ERRORS         0              0
```

# Listing File: Solution Listing

- equation and variable **primal and dual values and bounds**
- **marking** of infeasibilities, “non-optimality”, and unboundedness
- ‘.’ = zero

	LOWER	LEVEL	UPPER	MARGINAL	
-- EQU Objective	.	.	.	1.0000	
-- EQU AlkylShrnk	.	.	.	-4.6116	
-- EQU AcidBal	.	-0.0020	.	11.8406	INFES
-- EQU IsobutBal	.	0.0952	.	0.0563	INFES
-- EQU AlkylDef	.	0.0127	.	-1.0763	INFES
-- EQU OctDef	0.5743	0.5747	0.5743	-25.9326	INFES
-- EQU AcidDef	35.8200	35.8533	35.8200	0.2131	INFES
-- EQU F4Def	-1.3300	-1.3300	-1.3300	-4.1992	
	LOWER	LEVEL	UPPER	MARGINAL	
-- VAR F	-INF	-1.4143	+INF	.	
-- VAR OlefinFeed	.	1.6198	2.0000	-0.1269	NOPT
-- VAR IsobutRec	.	1.3617	1.6000	-0.2133	NOPT
-- VAR AcidFeed	.	0.7185	1.2000	-0.0411	NOPT
-- VAR AlkylYld	.	2.8790	5.0000	-0.0076	NOPT
-- VAR IsobutMak	.	1.8926	2.0000	-0.4764	NOPT
-- VAR AcidStren	0.8500	0.8998	0.9300	0.5273	NOPT



# MIRO – Model Annotations

## Set

```
i 'canning plants' / seattle, san-diego /  
j 'markets' / new-york, chicago, topeka /;
```

\$onExternalInput

## Parameter

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

```
b(j) 'demand at market j in cases'  
/ new-york 325  
chicago 300  
topeka 275 /;
```

**Table** d(i,j) 'distance in thousands of miles'

	<b>new-york</b>	<b>chicago</b>	<b>topeka</b>
seattle	2.5	1.7	1.8
san-diego	2.5	1.8	1.4;

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

\$offExternalInput

**Parameter** c(i,j) 'transport cost in thousands of dollars per case';

```
c(i,j) = f*d(i,j)/1000;
```

\$onExternalOutput

## Variable

```
x(i,j) 'shipment quantities in cases'
```

```
z 'total transportation costs in thousands of dollars';
```

\$offExternalOutput