

:::dash optimization

Optimization Modeling and Programming in Xpress-Mosel

Alkis Vazacopoulos
CAPD Meeting
Carnegie Mellon University
Pittsburgh, PA
March 11-13, 2007

● ● ● Agenda

- Modeling Basics
 - Modeling and Programming Features
 - Optimization
 - Deployment
- Modeling Advanced
 - Algorithms
 - Debugging & Profiling
 - Mosel Modules
 - I/O Drivers
 - Model Separation

⌘dash optimization

● Developing an optimization model ● in Mosel

- Describe the business problem in the Mosel algebraic modeling language
- Ex: Capital Budgeting
 - Set of projects with net return
 - Each project has capital and personnel requirements
 - Limited capital and personnel resources
 - Select set of projects to maximize return

-
-
- Developing an optimization model

Building Block 1

- Model name and parameters

```
model "CapitalBudgeting"  
    uses "mmxprs"                ! Use Xpress optimizer  
  
    parameters                    ! Change at run-time  
        DATA_FILE = 'projects.dat'  
    end-parameters  
    .....  
end-model
```

-
-
- Developing an optimization model

Building Block 2

- Data declaration and reading

declarations

PROJECTS : set of string

CAPITAL_MAX : real

CAPITAL : array (PROJECTS) of real

end-declarations

initializations from DATA_FILE

CAPITAL_MAX

CAPITAL as 'CAPITAL_DATA'

end-initializations

⌘dash optimization

-
-
- Developing an optimization model

Building Block 3

- Variable and constraint declarations

declarations

do_project: array(PROJECTS) of mpvar

! Decision variable

MaxReturn, Capital, Personnel: lincr

! Constraints

end-declarations

forall (p in PROJECTS)

do_project (p) is_binary ! Binary variable

-
-
- Developing an optimization model

Building Block 4

- Build constraints and objective

Total return is sum of selected projects returns

MaxReturn :=

sum (p in PROJECTS) RETURN(p) * **do_project**(p)

-
-
- Developing an optimization model

Building Block 5

- Solve optimization model

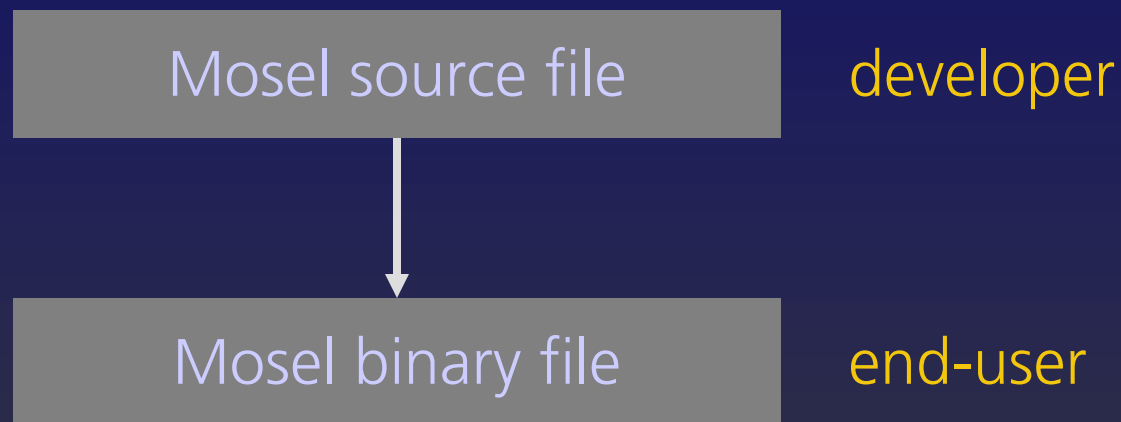
maximize (MaxReturn)

-
-
- Developing an optimization model

Building Block 6

- Solution
 - if **getprobat=XPRS_OPT** then
 - writeln("Solution:\n Objective: ", **getobjval**)
 - forall(i in PROJECTS)
 - write(" x(", i, "): ", **getsol(x(i))**)
 - end-if

Deploying Optimization Models



- protects intellectual property
- platform independent
- efficient

• • • More Mosel Features

- Modeling
 - Variable: free, integer, partial integer, semi-continuous, sos1, sos2
- Programming
 - Dynamic arrays
 - Sparse data
 - Selections: if-then-elif-then-else, case
 - Loops: forall, while, repeat-until

⚙️dash optimization

• • • Subroutines

Functions

```
forward function add ( a : integer, b : integer ) :  
    integer
```

```
.....
```

```
Temp := add(10,20)           ! this will return "30"
```

```
.....
```

```
function add ( a : integer, b : integer ) : integer
```

```
    returned := a + b
```

```
end-function
```

⌘dash optimization

● ● ● Algorithms

These functionalities allow

- Multiple optimization calls in one model
- Create and solve different problems in one model file
- Implement advanced algorithms, experiment and try more new ideas

Benefit: Spend more time in 'designing' rather than 'implementing'

-
-
- Algorithms: Modifying the Problem

After solving **first** problem, one can

- Create 'additional' new variables and constraints
- Delete existing constraints
- Add / delete variables to existing constraints
- Hide / Un-hide constraints

Solve **second** modified problem

-
-
- Algorithms: Modifying the Problem

Add / delete variables to existing constraints

Constraint := $5x + 2y \leq 20$

...

maximize (Objective_1)

...

Constraint += $7y$

...

maximize (Objective_2)

...

⌘dash optimization

the 'revised' constraint is

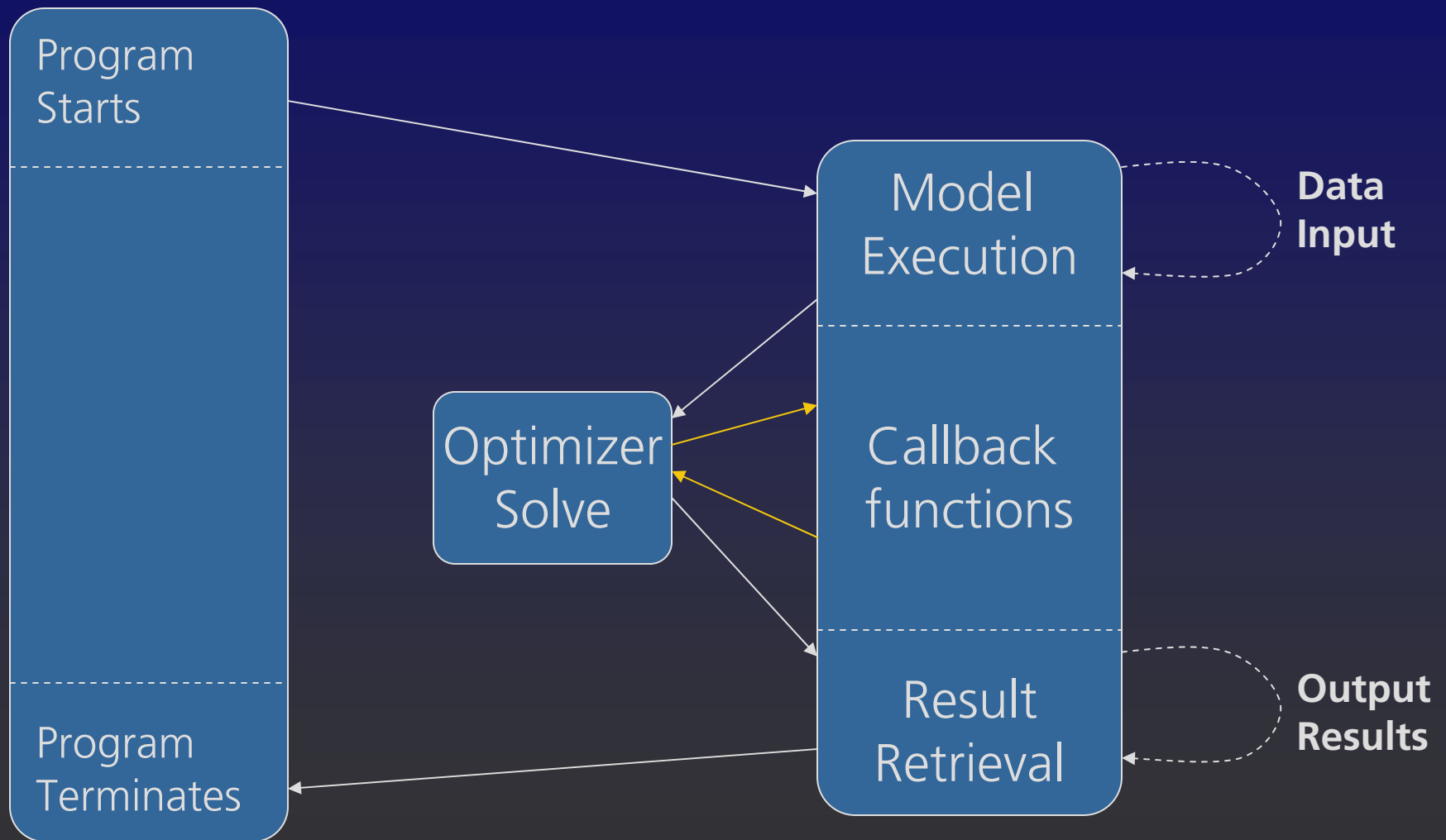
Constraint := $5x + 9y \leq 20$

-
-
- Algorithms: Modifying the Problem

Can implement algorithms / heuristics such as

- Chronological decomposition of planning period in scheduling problems
- Draw efficient frontier by changing R.H.S.
- Add constraints and monitor change in objective
- Column Generation (Master and Sub-problem)

-
-
- Algorithms: Interact with Optimizer



Algorithms

- Intermediate MIP solution
Set-up for integer solution

```
setcallback (XPRS_CB_INTSOL, "WriteReport" )
```

! Callback to function 'WriteReport'

.....

```
minimize ( Cost )
```

Algorithms: Interact with Optimizer

- Branch and Cut

Set-up for cut generation









```
setcallback ( XPRS_CB_CM, "add_cut" )
```









! Cut manager callback to function 'add_cut'

.....

```
minimize ( Cost )
```

• • • Debugging & Profiling

	Re/Set Breakpoint	F9
	Set/Remove breakpoint condition	
	Start / Continue	F5
	Step over	F10
	Step into	F11
	Run to cursor	F12
	Debug Options...	
	Profile	

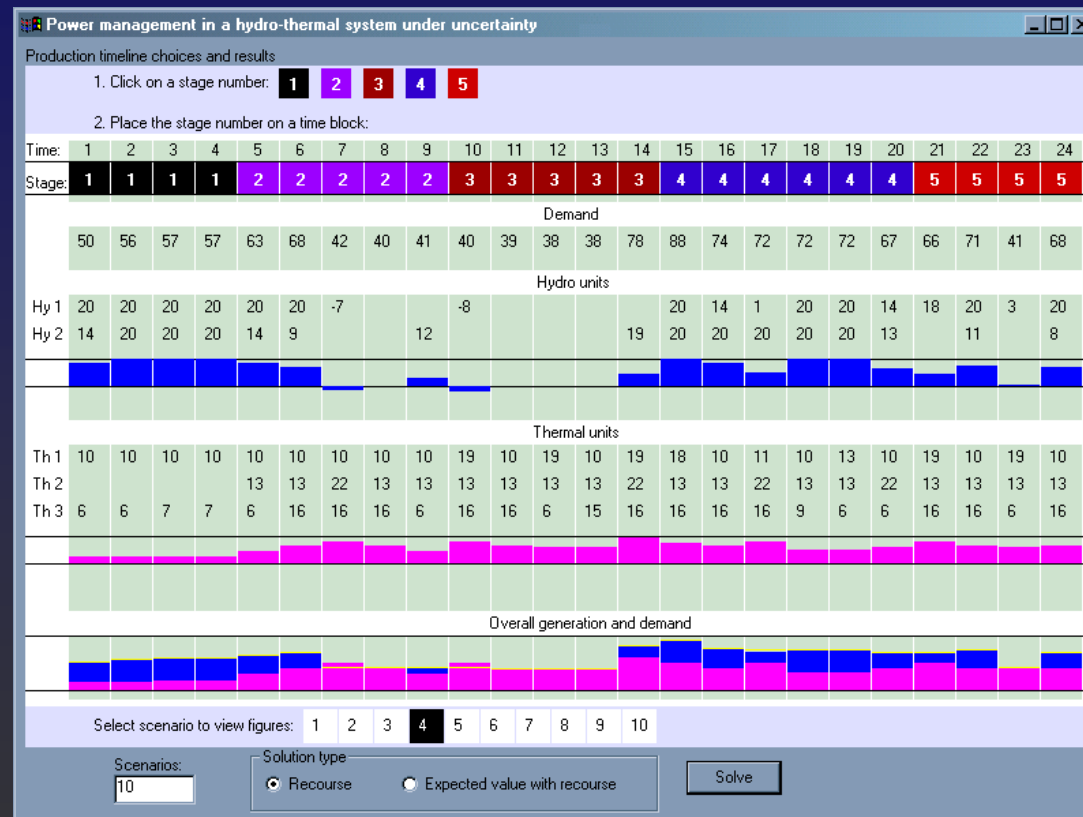
	Re/Set Breakpoint	F9
	Set/Remove breakpoint condition	
	Start / Continue	F5
	Step over	F10
	Step into	F11
	Run to cursor	F12
	Debug Options...	
	Profile	

• • • Modules

- mmodbc: ODBC connectivity
- mmquad: QP, MIQP
- mmxslp: NLP, MINLP
- mmsp: Stochastic LP, MILP
- mmive: Graphing
- mmxad: GUI builder
- kalis: CP

Extend Mosel: NI can be used to create user defined modules

Stochastic model



- mmrng: Simulation runs
- mmsp: Stochastic LP
- mmxad: Visual Interface

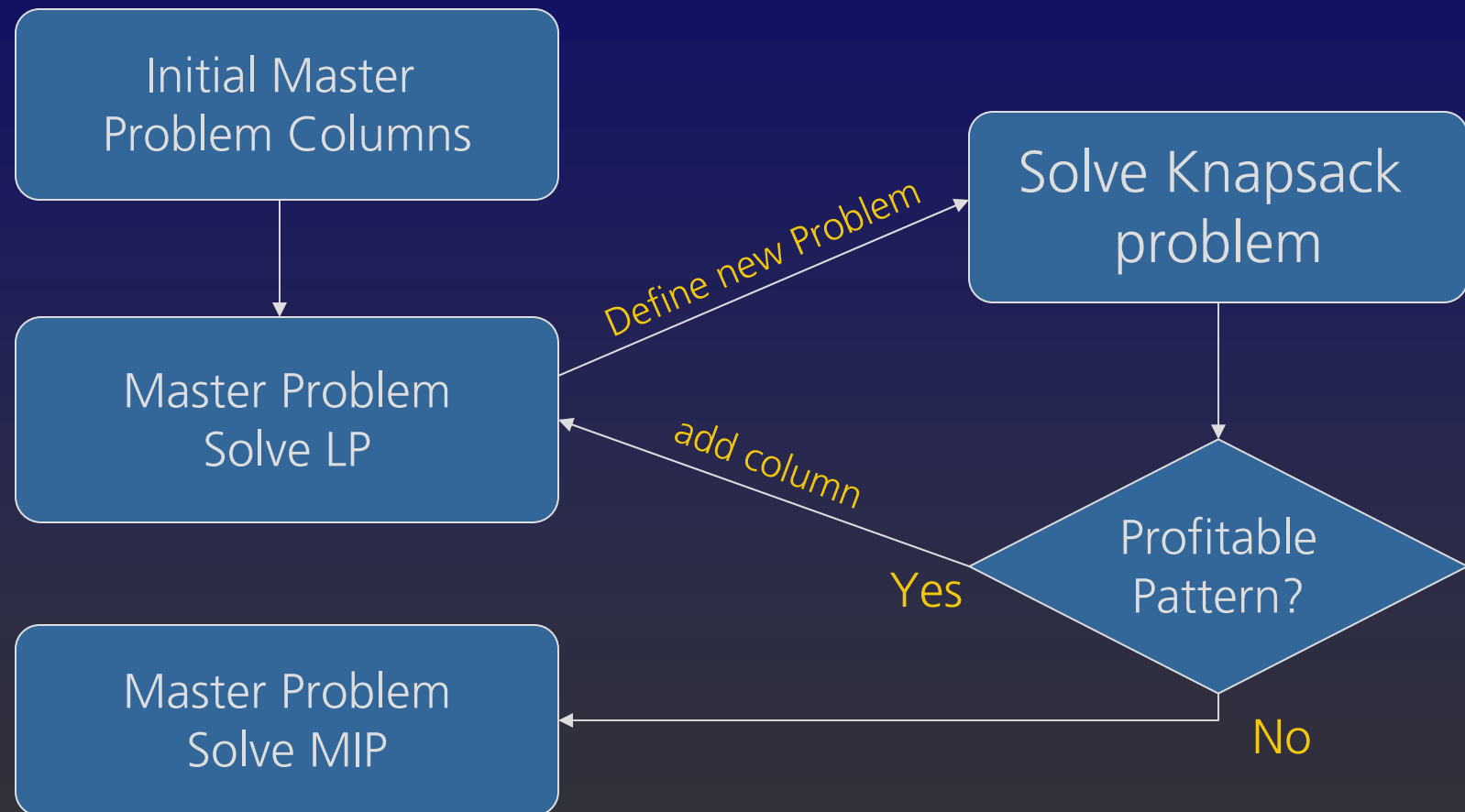
• • • I/O drivers

- Data exchange between concurrent models:
 - reading and writing data from/to memory
 - synchronization of data access
- 1. shmem
- 2. mempipe

-
-
- Model Separation

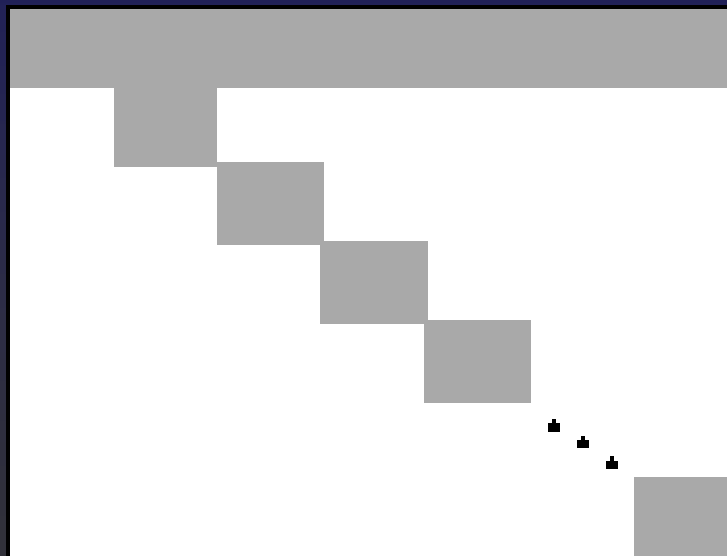
- module: mmjobs
- Sequential model
- Parallel model
- Exs:
 - Column Generation
 - Dantzig-Wolfe Decomposition

Column generation -Flowchart

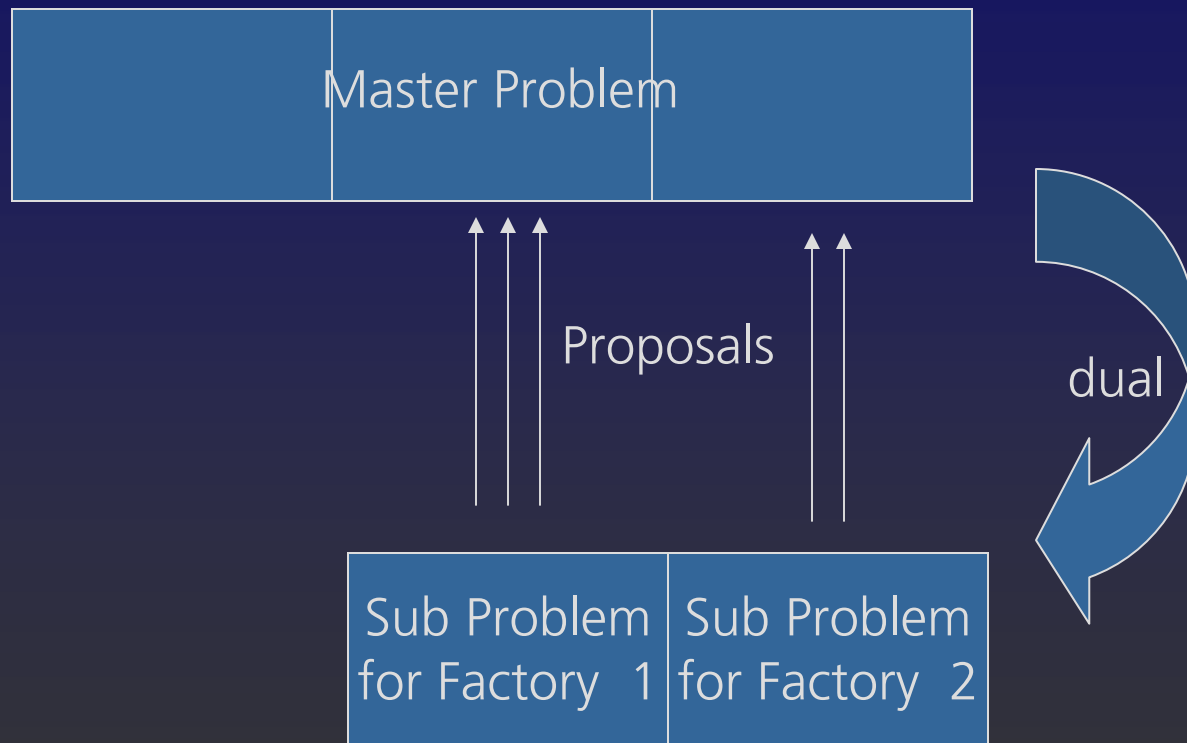


• • • Dantzig-Wolfe Decomposition

- Multi-item, multi-period production planning



-
-
- Dantzig-Wolfe decomposition



- Modeling statements with Xpress-Mosel and
- Xpress-IVE
-

The screenshot displays the Xpress-IVE application window titled "Xpress-IVE - [locdemo01x.mos]". The interface is divided into several sections:

- Menu Bar:** File, Edit, View, Build, Debug, Deploy, Modules, Wizards, Window, Help.
- Toolbar:** Contains icons for file operations (open, save, print), editing (undo, redo), and execution (run, stop, step through).
- Entity Manager (Left Pane):** Shows a tree view of entities. The "Entities" list includes Parameters, Constants, Primitives, Decision Variables, Constraints, and Subroutines. The "Most recent entities" list is also visible.
- Code Editor (Center):** Displays the Mosel model code for "locdemo01x.mos". The code defines sets, arrays, decision variables, constraints, and a minimization objective.
- Status Bar (Bottom):** Shows the current state as "Ready", the file path "(C:\location20complete\locdemo01.mos)", and the current line and column numbers (Line: 82/98, Col: 0).

The Mosel code in the editor is as follows:

```

declarations
  DemandCities: set of string
  SupplyCities: set of string

  TCOST : array(SupplyCities,DemandCities) of real
  DIST: array(SupplyCities,DemandCities) of real
  DEMAND: array(DemandCities) of integer

  !Decision variables
  x : array(SupplyCities,DemandCities) of mpvar !flow fraction
  y : array(SupplyCities) of mpvar !1 open, 0 closed
end-declarations

!constraints
forall(i in SupplyCities) y(i) is_binary

TotalCost := sum(i in SupplyCities,j in DemandCities) TCOST(i,j) * DEMAND(j) * x(i,j)

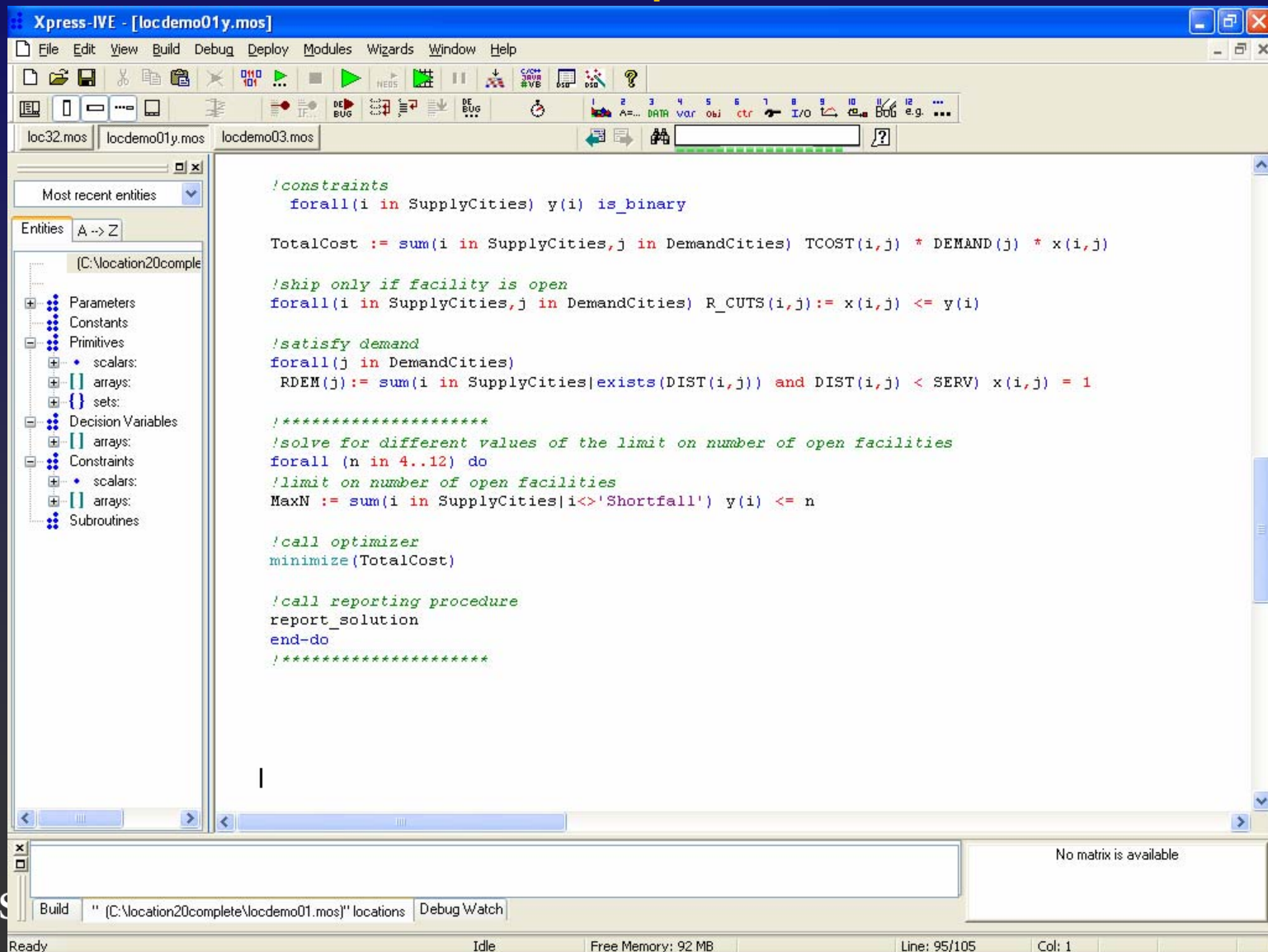
!limit on number of open facilities
MaxN := sum(i in SupplyCities|i<>'Shortfall') y(i) <= N

!ship only if facility is open
forall(i in SupplyCities,j in DemandCities) R_CUTS(i,j) := x(i,j) <= y(i)

!satisfy demand
forall(j in DemandCities)
  RDEM(j) := sum(i in SupplyCities|exists(DIST(i,j)) and DIST(i,j) < SERV) x(i,j) = 1

!call optimizer
minimize(TotalCost)
  
```

Modeling and programming statements with Xpress-Mosel



The screenshot displays the Xpress-IVE (Xpress Interactive Visual Editor) interface. The main window shows a Mosel model file named 'locdemo01y.mos'. The left sidebar contains a tree view of the model's entities, including Parameters, Constants, Primitives, Decision Variables, Constraints, and Subroutines. The main text area contains the following Mosel code:

```
!constraints
forall(i in SupplyCities) y(i) is_binary

TotalCost := sum(i in SupplyCities, j in DemandCities) TCOST(i,j) * DEMAND(j) * x(i,j)

!ship only if facility is open
forall(i in SupplyCities, j in DemandCities) R_CUTS(i,j) := x(i,j) <= y(i)

!satisfy demand
forall(j in DemandCities)
  RDEM(j) := sum(i in SupplyCities | exists(DIST(i,j)) and DIST(i,j) < SERV) x(i,j) = 1

!*****
!solve for different values of the limit on number of open facilities
forall (n in 4..12) do
  !limit on number of open facilities
  MaxN := sum(i in SupplyCities | i <> 'Shortfall') y(i) <= n

!call optimizer
minimize(TotalCost)

!call reporting procedure
report_solution
end-do
!*****
```

The status bar at the bottom indicates the current state: Ready, Idle, Free Memory: 92 MB, Line: 95/105, Col: 1. A message box at the bottom right states 'No matrix is available'.

- Browsing the solution and model
- entities in Xpress-IVE

The screenshot displays the Xpress-IVE interface for a file named 'locdemo01.mos'. The left sidebar shows the model's hierarchical structure, including parameters (DATAFILE, DBDATA3, DBDATA4, DBDATA5, N, SERV), constants, primitives (scalars, arrays), sets (DemandCities, SupplyCities), and decision variables (arrays: x, y). A tooltip for 'SupplyCities' indicates it is a set of string containing 145 elements.

The central 'Table view' displays the 'SupplyCities' data as follows:

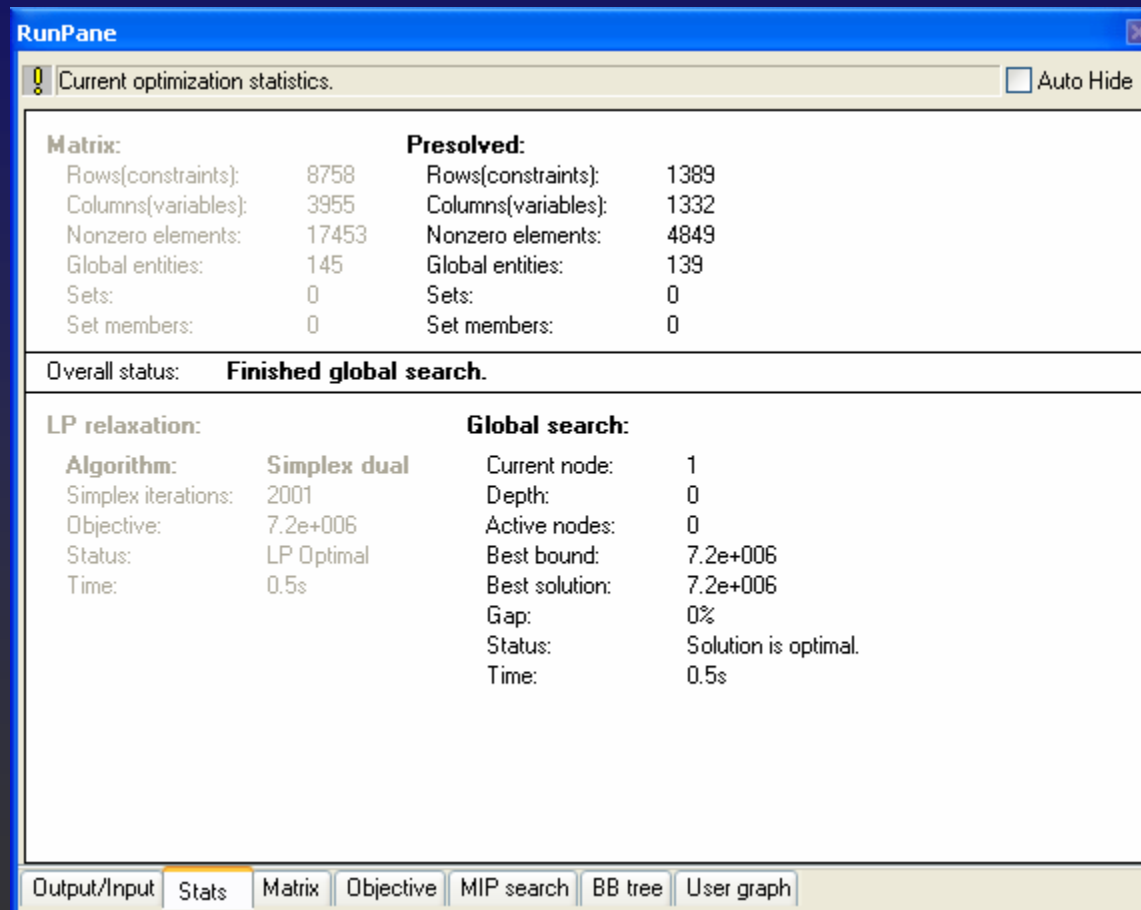
SupplyCities	y
Shortfall	1
Walnut, CA	1
Albuquerque, NM	1
Texarkana, AR	1
Gainesville, FL	1
South Bend, IN	1
Reading, PA	1
Walla Walla, WA	1
Cheyenne, WY	1
Jackson, MS	-0
College Park, GA	-0
Jacksonville, FL	-0
Modesto, CA	-0
West Jordan, UT	-0
Dallas, TX	-0
Harrisonburg, VA	-0
Westbrook, ME	-0
Harmony, PA	-0
Boonville, MO	-0
Concord, NC	-0
Phoenix, AZ	-0
Fremont, CA	-0
Harrisburg, PA	-0
Wilsonville, OR	-0

The right sidebar shows the solver output, indicating a successful solution with an objective value of 1962968.370. The output includes the number of integer feasible solutions, the best integer solution found, and the uncrunching matrix.

Output/Stats Matrix Objective MIP search
BB tree User graph

No matrix is available

Run statistics in Xpress-IVE



The screenshot shows a window titled "RunPane" with a close button in the top right corner. Below the title bar is a status bar with a yellow warning icon, the text "Current optimization statistics.", and an "Auto Hide" checkbox. The main content area is divided into several sections. The first section contains two columns of statistics: "Matrix" and "Presolved:". The "Matrix" column lists Rows(constraints): 8758, Columns(variables): 3955, Nonzero elements: 17453, Global entities: 145, Sets: 0, and Set members: 0. The "Presolved:" column lists Rows(constraints): 1389, Columns(variables): 1332, Nonzero elements: 4849, Global entities: 139, Sets: 0, and Set members: 0. Below this is a section for "Overall status:" which says "Finished global search.". The next section is divided into "LP relaxation:" and "Global search:". The "LP relaxation:" section lists Algorithm: Simplex dual, Simplex iterations: 2001, Objective: 7.2e+006, Status: LP Optimal, and Time: 0.5s. The "Global search:" section lists Current node: 1, Depth: 0, Active nodes: 0, Best bound: 7.2e+006, Best solution: 7.2e+006, Gap: 0%, Status: Solution is optimal., and Time: 0.5s. At the bottom of the window is a tabbed interface with tabs for "Output/Input", "Stats" (which is selected), "Matrix", "Objective", "MIP search", "BB tree", and "User graph".

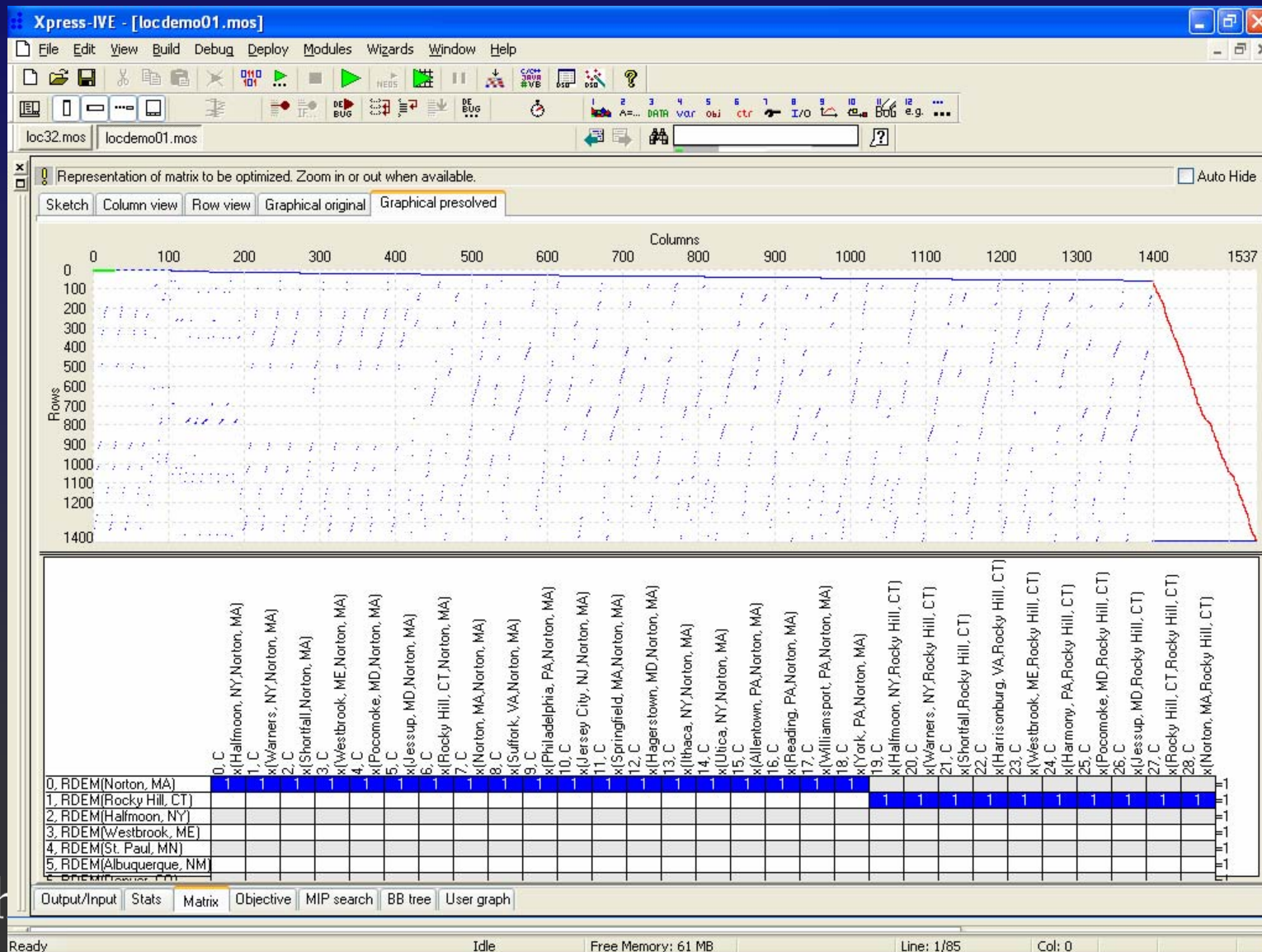
Matrix:		Presolved:	
Rows(constraints):	8758	Rows(constraints):	1389
Columns(variables):	3955	Columns(variables):	1332
Nonzero elements:	17453	Nonzero elements:	4849
Global entities:	145	Global entities:	139
Sets:	0	Sets:	0
Set members:	0	Set members:	0

Overall status: **Finished global search.**

LP relaxation:		Global search:	
Algorithm:	Simplex dual	Current node:	1
Simplex iterations:	2001	Depth:	0
Objective:	7.2e+006	Active nodes:	0
Status:	LP Optimal	Best bound:	7.2e+006
Time:	0.5s	Best solution:	7.2e+006
		Gap:	0%
		Status:	Solution is optimal.
		Time:	0.5s

Output/Input Stats Matrix Objective MIP search BB tree User graph

Problem matrix in Xpress-IVE



Problem matrix/solution in Xpress-IVE

Xpress-IVE - [locdemo01.mos]

File Edit View Build Debug Deploy Modules Wizards Window Help

loc32.mos locdemo01.mos

Representation of matrix to be optimized. Zoom in or out when available. ☐ Auto Hide

Sketch Column view Row view Graphical original Graphical resolved

	Column	Name	R...	LB	UB	Solution	Reduced cost	Type
~	1040	x(South Bend, IN,Ba...	2	0	1e+020	1	0	Continuous
~	1050	x(Gainesville, FL,Mia...	2	0	1e+020	1	0	Continuous
~	1072	x(South Bend, IN,Be...	2	0	1e+020	1	0	Continuous
~	1110	x(Texarkana, AR,Me...	2	0	1e+020	1	0	Continuous
~	1142	x(Texarkana, AR,Sa...	2	0	1e+020	1	0	Continuous
~	1181	x(South Bend, IN,Lo...	2	0	1e+020	1	-689.197	Continuous
~	1224	x(Reading, PA,Harm...	2	0	1e+020	1	0	Continuous
~	1256	x(Reading, PA,Harris...	2	0	1e+020	1	0	Continuous
~	1276	x(Texarkana, AR,Dal...	2	0	1e+020	1	0	Continuous
~	1307	x(Gainesville, FL,Jac...	2	0	1e+020	1	0	Continuous
~	1338	x(Gainesville, FL,Coll...	2	0	1e+020	1	0	Continuous
0/1	3954	y(Shortfall)	59	-0	1	1	0	Binary
0/1	1405	y(Cheyenne, WY)	60	-0	1	1	0	Binary
0/1	1486	y(Walla Walla, WA)	60	-0	1	1	0	Binary
0/1	1740	y(Reading, PA)	60	-0	1	1	0	Binary
0/1	2497	y(South Bend, IN)	60	-0	1	1	0	Binary
0/1	2797	y(Gainesville, FL)	60	-0	1	1	0	Binary
0/1	2881	y(Texarkana, AR)	60	-0	1	1	0	Binary
0/1	3178	y(Albuquerque, NM)	60	-0	1	1	0	Binary
0/1	3376	y(Walnut, CA)	60	-0	1	1	0	Binary
~	3886	x(Jacksonville, FL,St...	1	0	1e+020	0	20340.8	Continuous
~	3885	x(Jacksonville, FL,H...	1	0	1e+020	0	193490	Continuous
~	3884	x(Jacksonville, FL,P...	1	0	1e+020	0	101169	Continuous
~	3883	x(Jacksonville, FL,Je...	1	0	1e+020	0	162768	Continuous
~	3882	x(Jacksonville, FL,H...	1	0	1e+020	0	152420	Continuous
~	3881	x(Jacksonville, FL,S...	1	0	1e+020	0	87294.7	Continuous
~	3880	x(Jacksonville, FL,P...	1	0	1e+020	0	68299.3	Continuous
~	3879	x(Jacksonville, FL,Je...	1	0	1e+020	0	160101	Continuous
~	0	x(Halfmoon, NY,Nort...	2	0	1e+020	0	0	Continuous
~	3877	x(Modesto, CA,West...	1	0	1e+020	0	57473.9	Continuous
~	3876	x(Modesto, CA,Boise...	1	0	1e+020	0	67783.2	Continuous
~	3875	x(Modesto, CA,Phne...	1	0	1e+020	0	138507	Continuous

Output/Input Stats Matrix Objective MIP search BB tree User graph

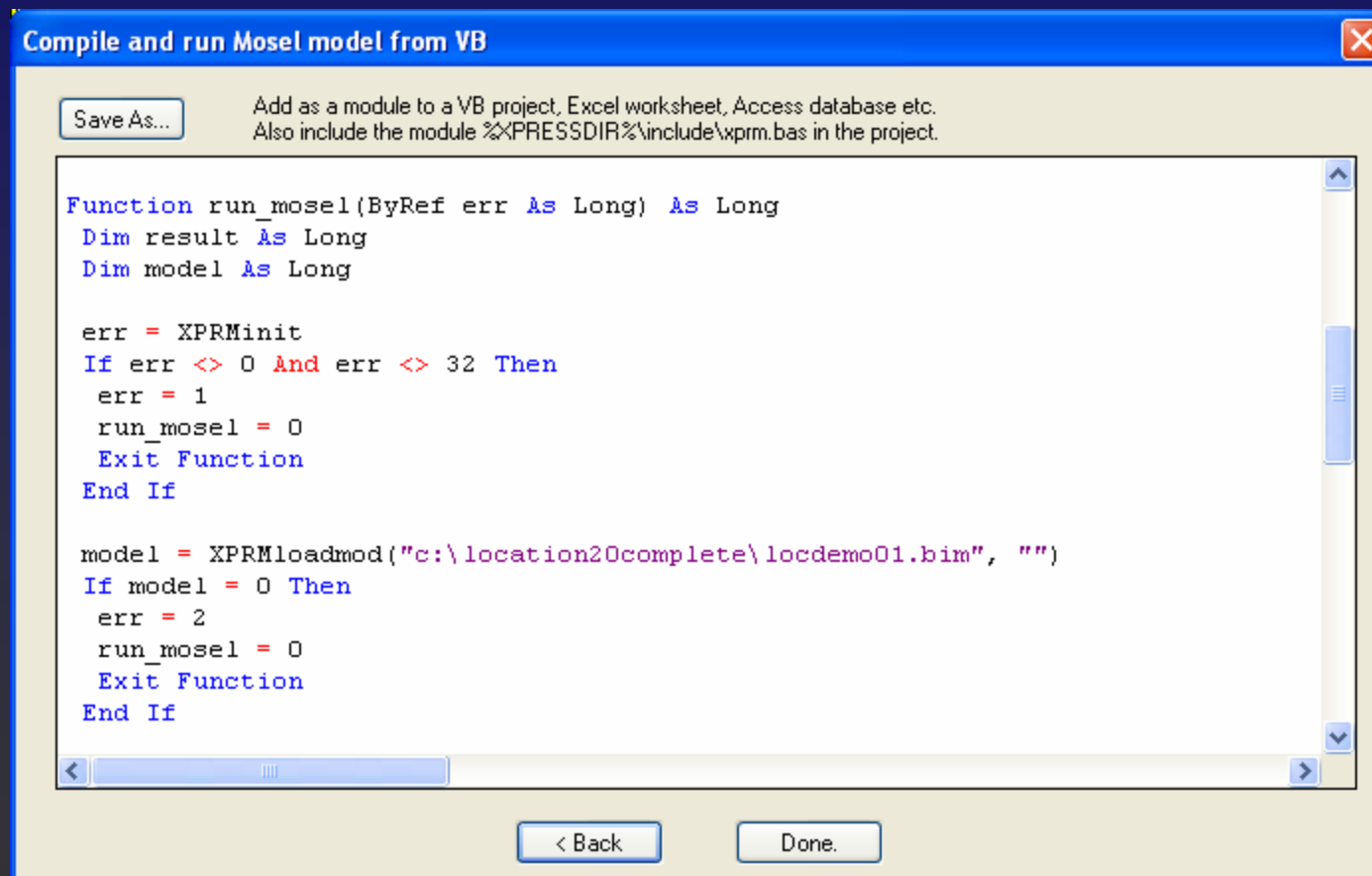
Ready Idle Free Memory: 76 MB Line: 1/85 Col: 0

Deployment wizard in Xpress-IVE

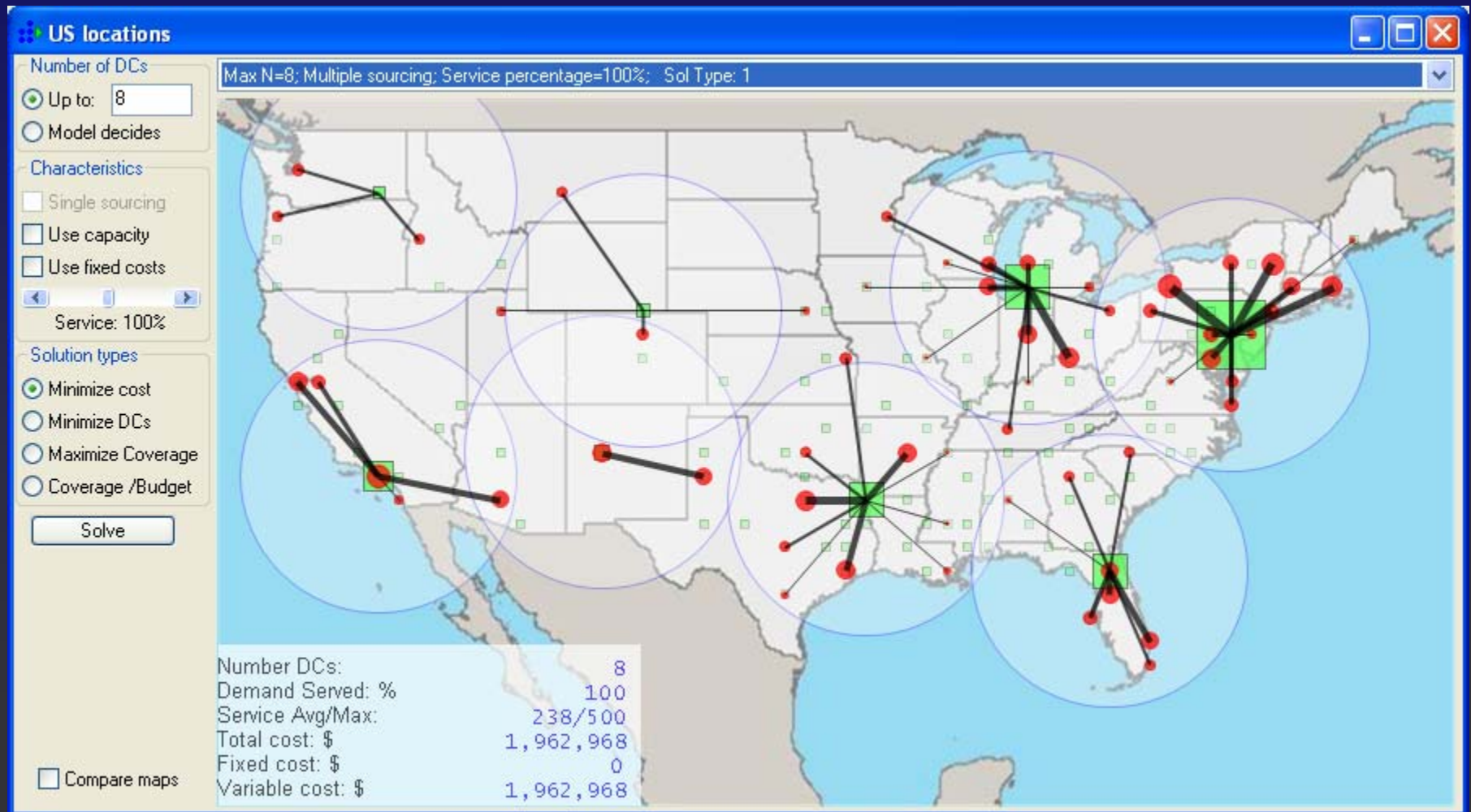
The screenshot shows a 'Deployment' dialog box with a blue title bar and a close button. The main area is light yellow. It contains the following elements:

- Title Bar:** 'Deployment' with a close button (X).
- Text:** 'The candidate file for deployment is:'
- Text Field:** 'C:\location20complete\locdemo01.mos'
- Text:** 'How would you like to use this Mosel model in your application?'
- Three Groups of Radio Buttons:**
 - Save .BIM file:**
 - ☐ With debug info
 - ☐ All names stripped
 -
 - Run Mosel model from:**
 - ☐ C
 - ☐ Java
 - ☒ Visual Basic
 - ☐ VB.NET
 - ☐ C#
 - Optimize matrix file from:**
 - ☐ C
 - ☐ Java
 - ☐ Visual Basic
 - ☐ VB.NET
 - ☐ C#
- Text:** 'To directly create a Windows executable that runs a .BIM file:'
- List:**
 1. Copy C:\XpressMP\bin\mrun.exe to the same folder as the .BIM file;
 2. Rename mrun.exe to match the name of the .BIM file, but with .EXE instead.
- Buttons:** 'Next >' and 'Cancel'.

Deployment wizard in Xpress-IVE



- Visualization, model controls, and
- mapping with Xpress-XAD



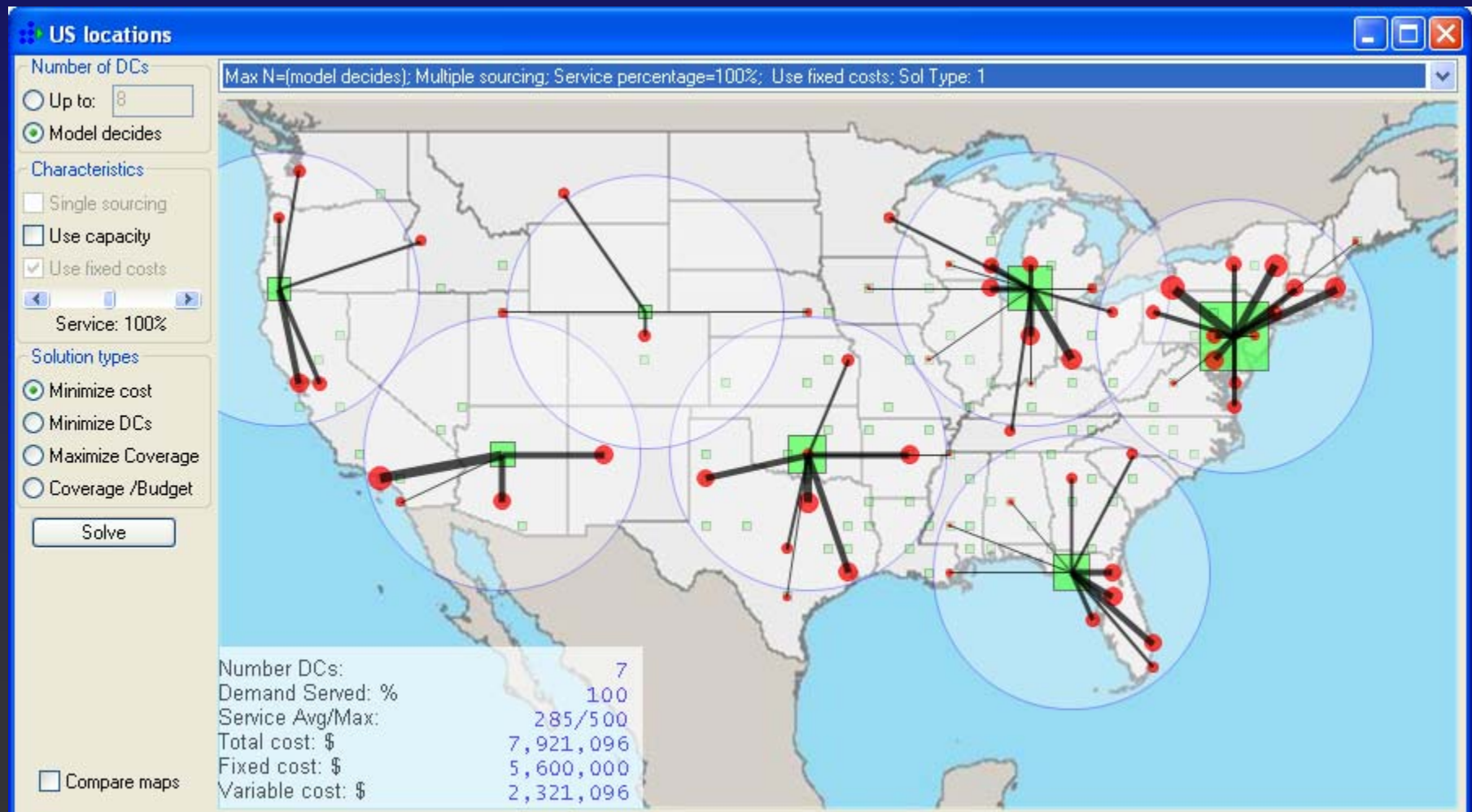
- One Program: Mosel model as
- procedure called from GUI

The screenshot displays the Xpress-IVE application window with the file 'loc32x.mos' open. The main editor contains a Mosel model procedure named 'run_model'. The procedure takes several parameters: 'n' (integer), 'switch_sing' (boolean), 'forceopen' and 'forceclosed' (arrays of strings to booleans), 'serv' (real), 'switch_nlim' (boolean), 'use_capacity' (boolean), 'use_fixedcost' (boolean), 'soltype' (integer), and 'budget' (real). The code defines 'FixedCost' based on whether 'use_fixedcost' is true or false, calculates 'VariableCost' as a sum over supply and demand cities, and then computes 'TotalCost', 'FixedCostR', 'VariableCostR', 'TotalCostR', 'TotalFacilities', 'TotalDemServed', and 'DistDemServed'. It also includes a constraint for 'MaxN' based on the 'switch_nlim' parameter.

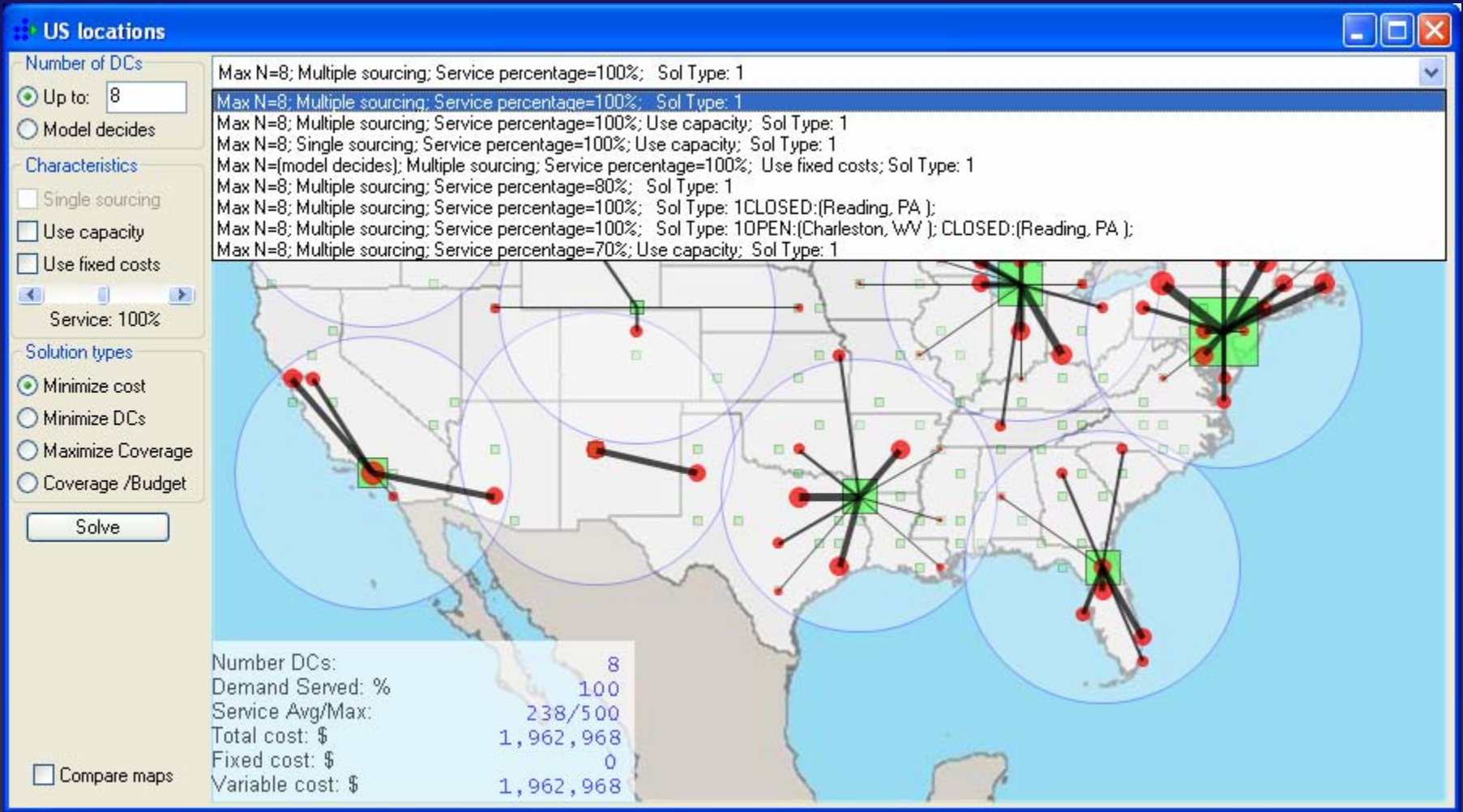
At the bottom, a status bar shows the following information:

- Module(s) in use: mmsystem version 1.6.1, mmxad version 1.0.0, mmxprs version 1.4.4, mmive version 1.17.2.
- Started running C:\location20complete\loc32
- Xpress-IVE: Model run complete
- Buttons: Build, "Subroutines" locations, Debug Watch, Copy to clipboard
- Status: Ready, Idle, Free Memory: 78 MB, Line: 977/1125, Col: 3, Modified

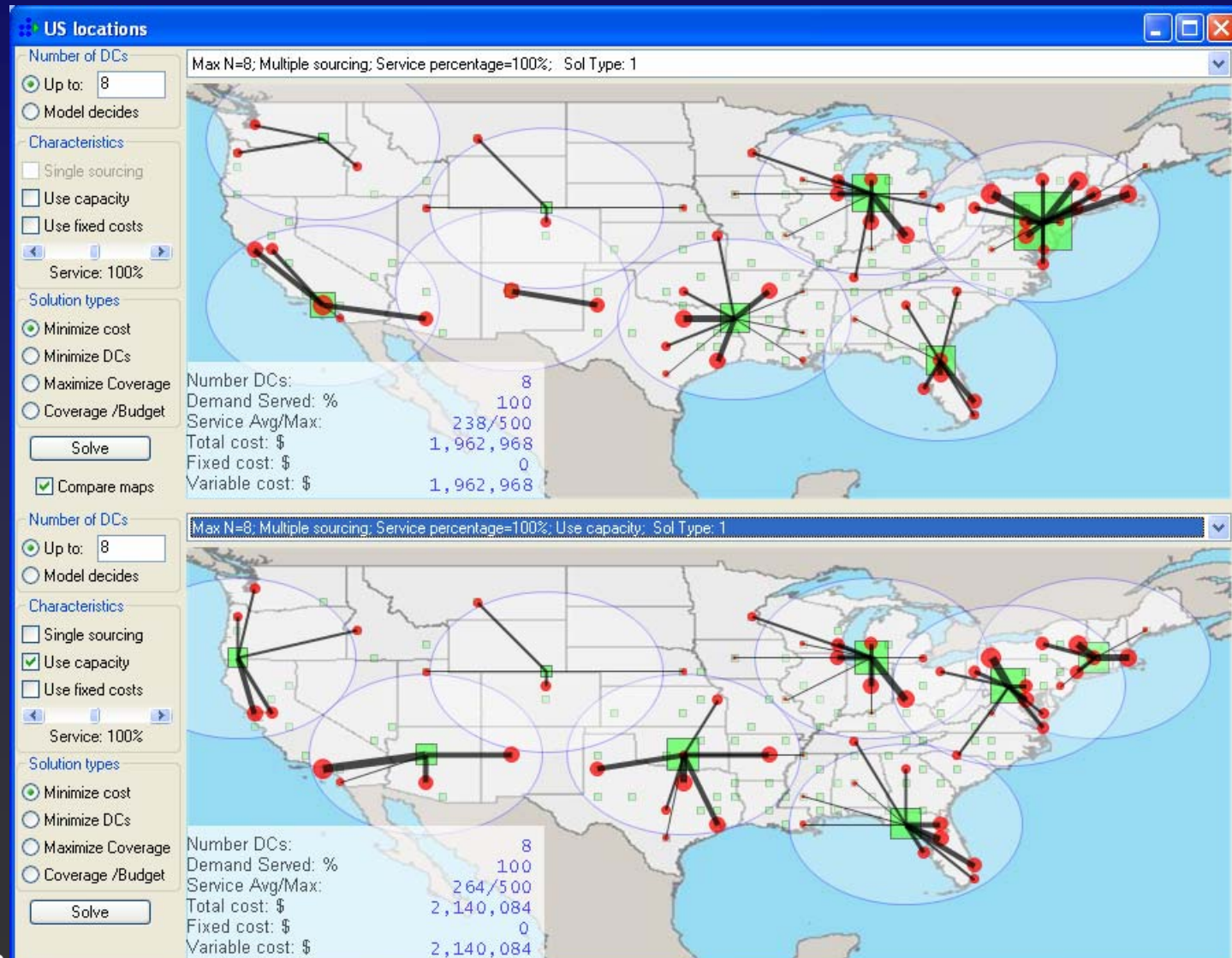
- Interactive solving through GUI
- with all data in computer memory



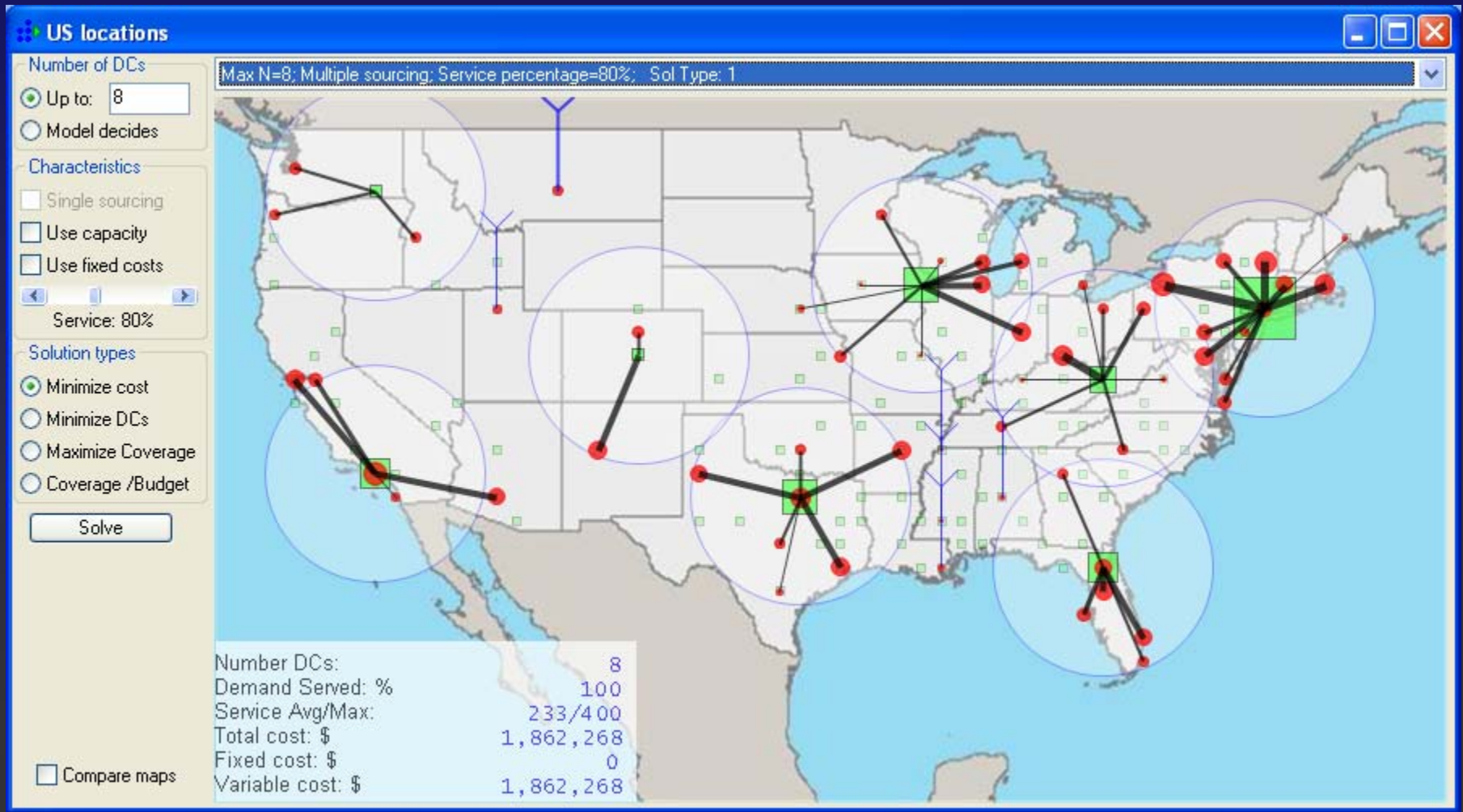
Scenario Management



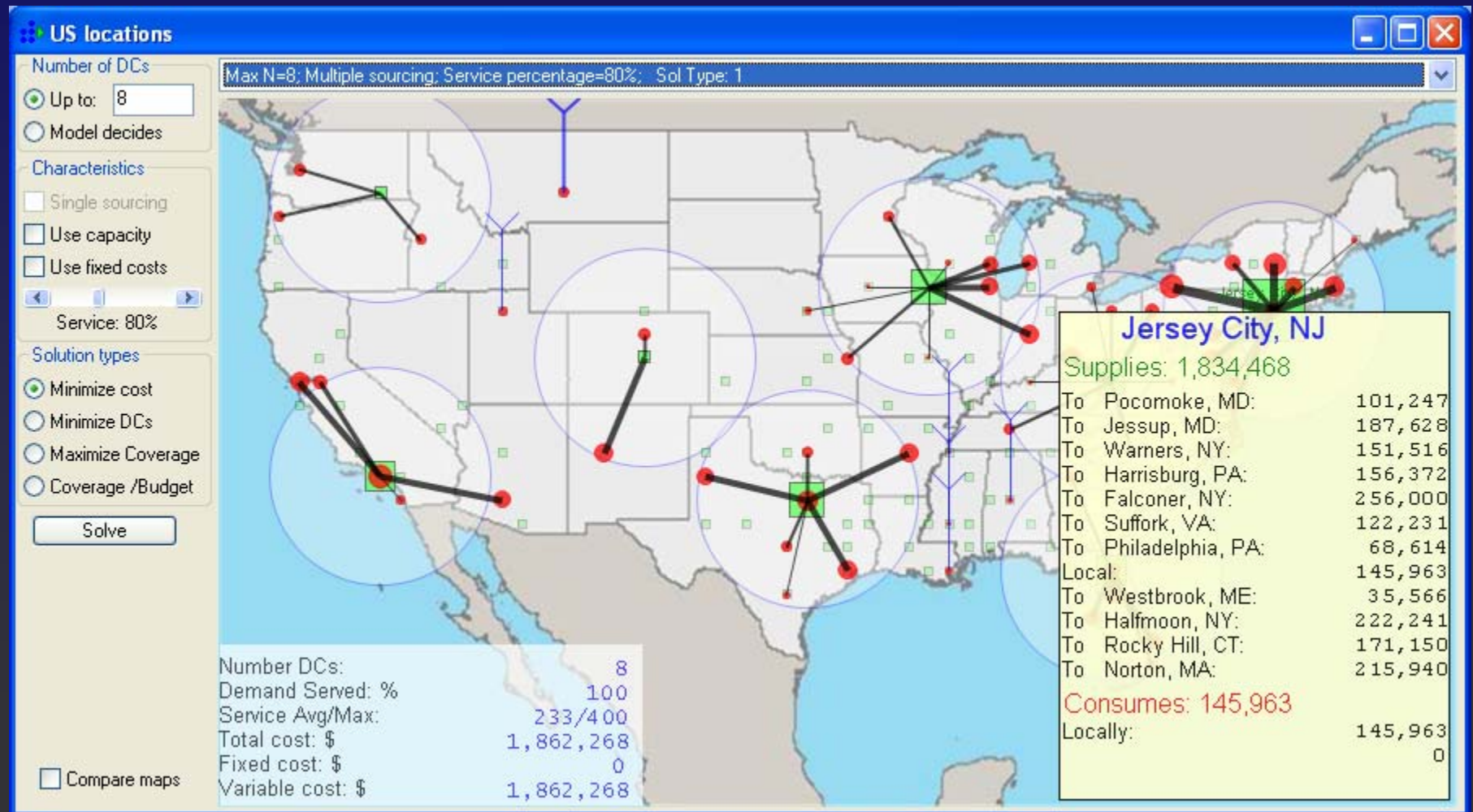
Scenario comparison/visualization



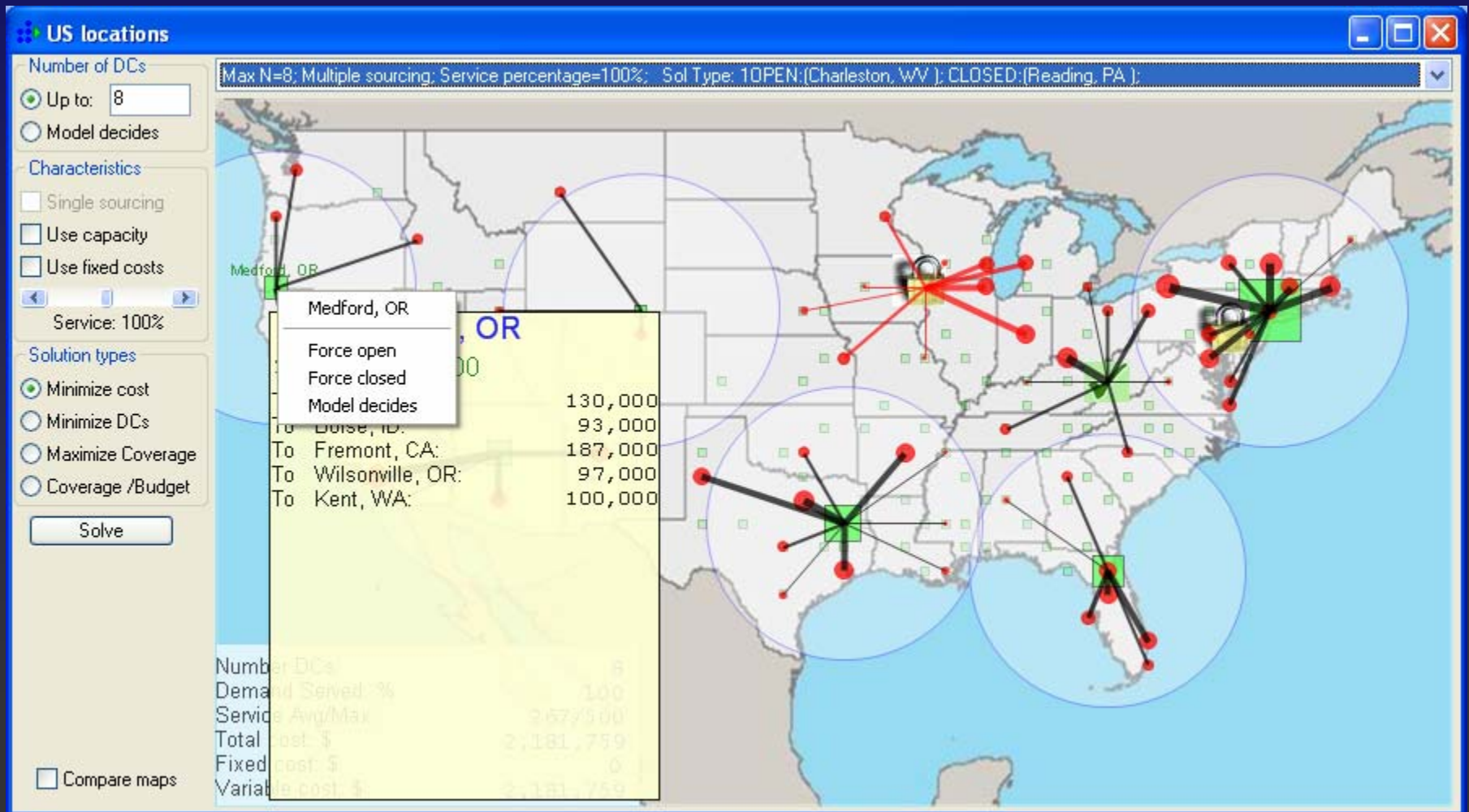
- Scenario with stricter service miles
- requirements and shortfalls



Visualization: Pop-up information



- Visualization: Forcing facilities open
- or closed



⋮dash optimization