
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

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
```

Developing an optimization model

Building Block 3

- Variable and constraint declarations

```
declarations
    do_project: array( PROJECTS ) of mpvar
                                        ! Decision variable
    MaxReturn, Capital, Personnel: linctr
                                        ! 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)

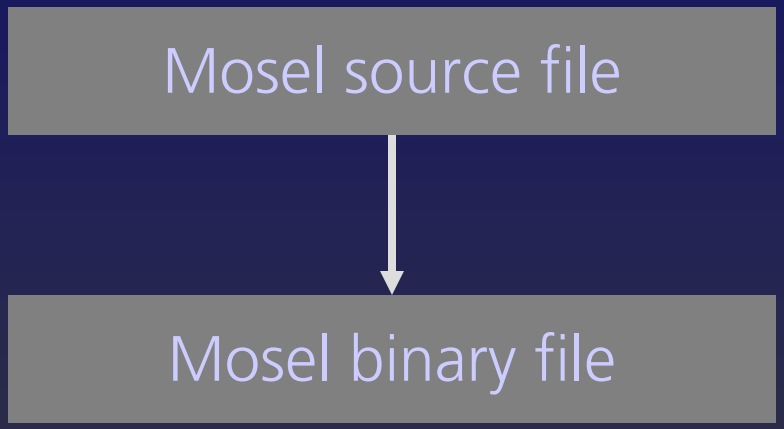
Developing an optimization model

Building Block 6

- Solution

```
if getprobstat=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

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 :=  5*x + 2*y <= 20
…
maximize ( Objective_1 )
…
Constraint += 7*y
…
maximize ( Objective_2)
…
```

the ‘revised’ constraint is

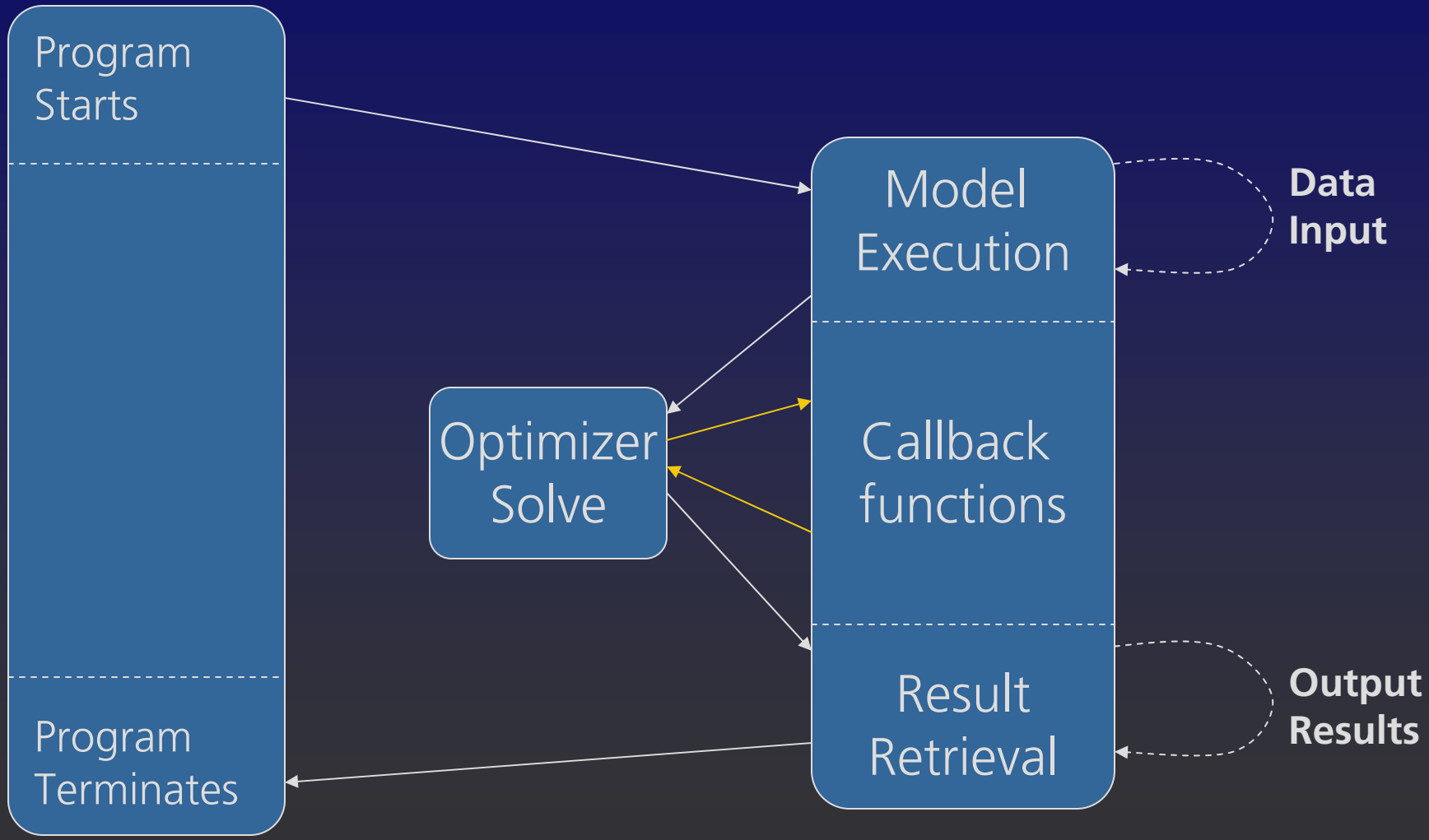
Constraint := 5*x + 9*y <= 20

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

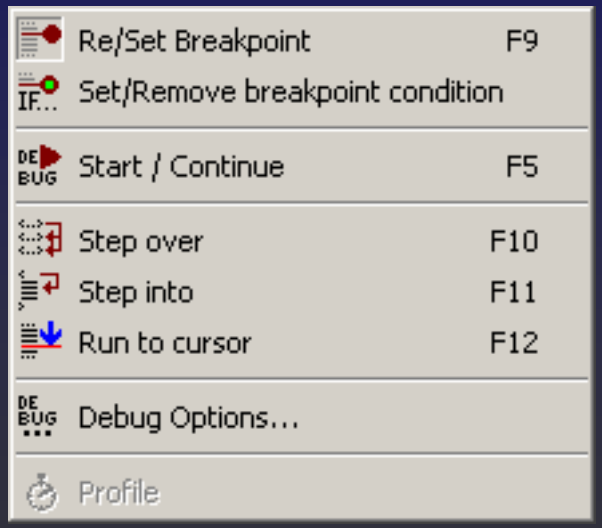
Set-up for cut generation

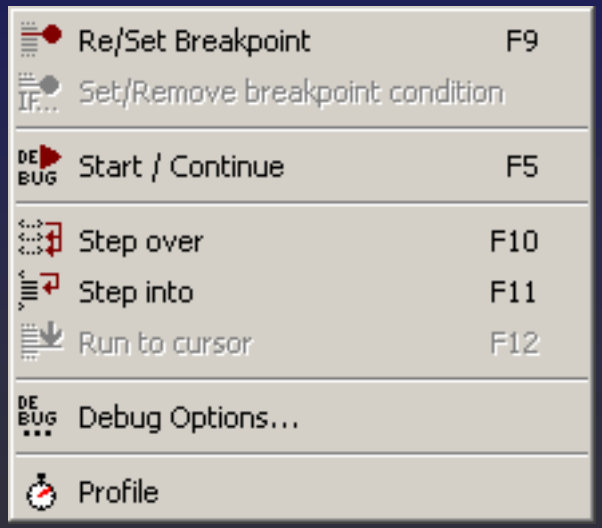
```
setcallback ( XPRS_CB_CM, "add_cut" )

        ! Cut manager callback to function 'add_cut'
……

minimize ( Cost )
```

Debugging & Profiling





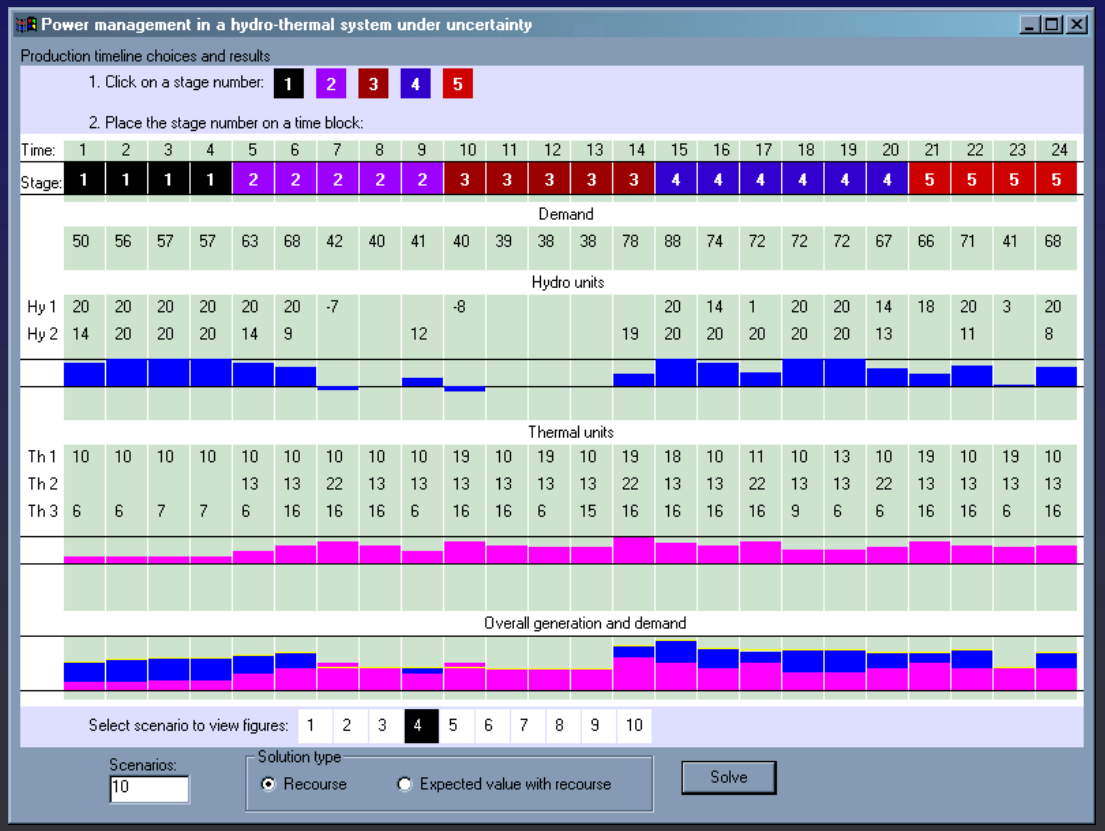
dash optimization

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

dash optimization

I/O drivers

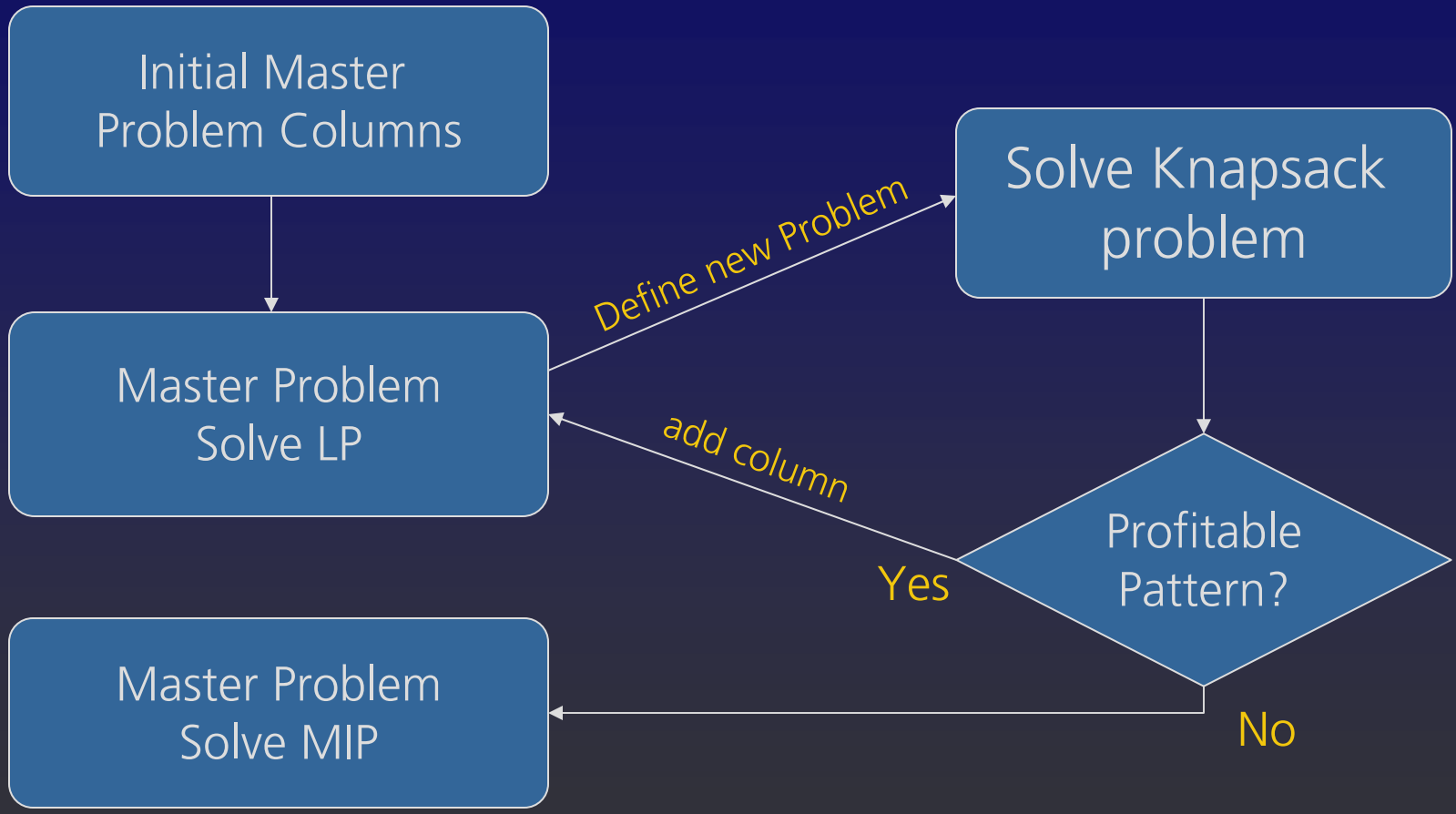
- Data exchange between concurrent models:
  - reading and writing data from/to memory
  - synchronization of data access

1. shmem
2. mempipe

Model Separation

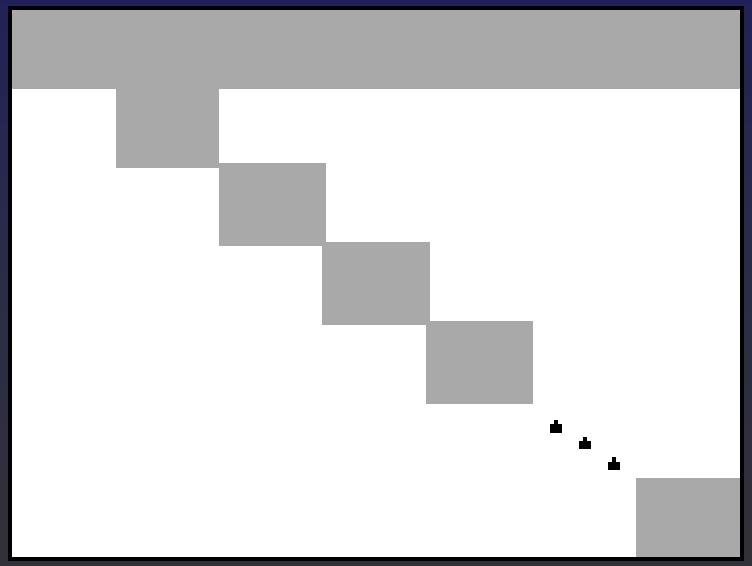
- module: mmjobs
- Sequential model
- Parallel model
- Exs:
  - Column Generation
  - Dantzig-Wolfe Decomposition

Column generation -Flowchart

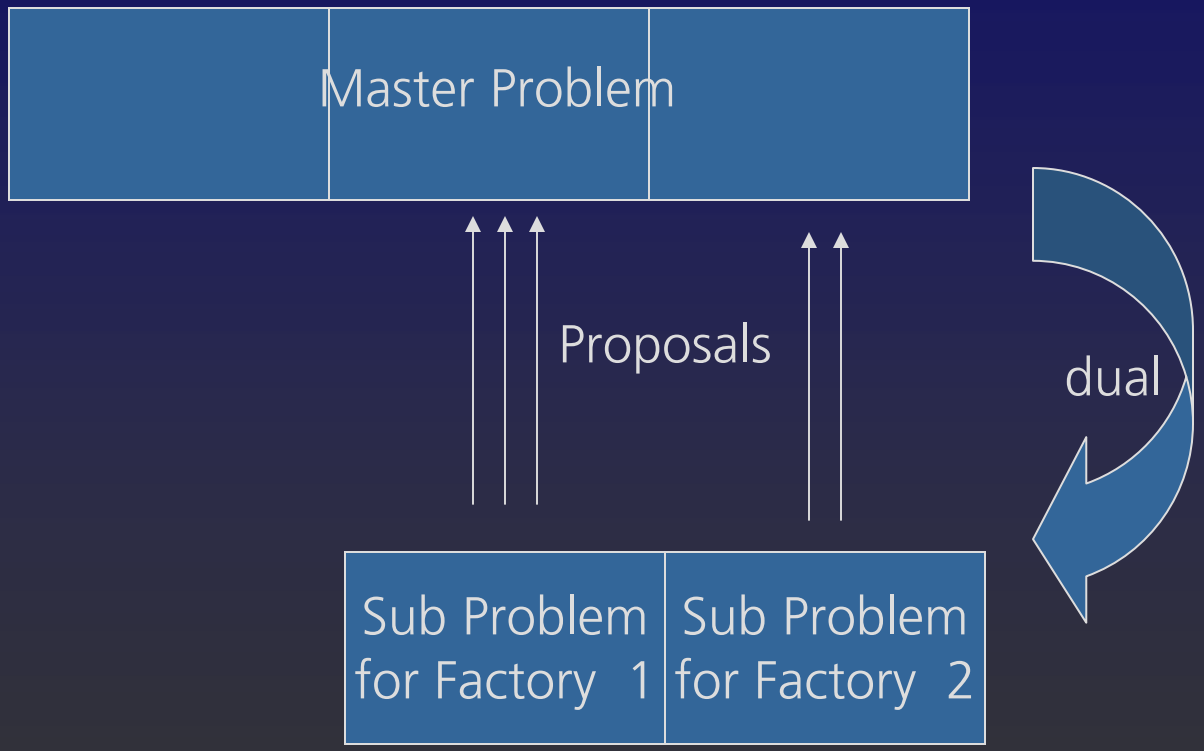


Dantzig-Wolfe Decomposition

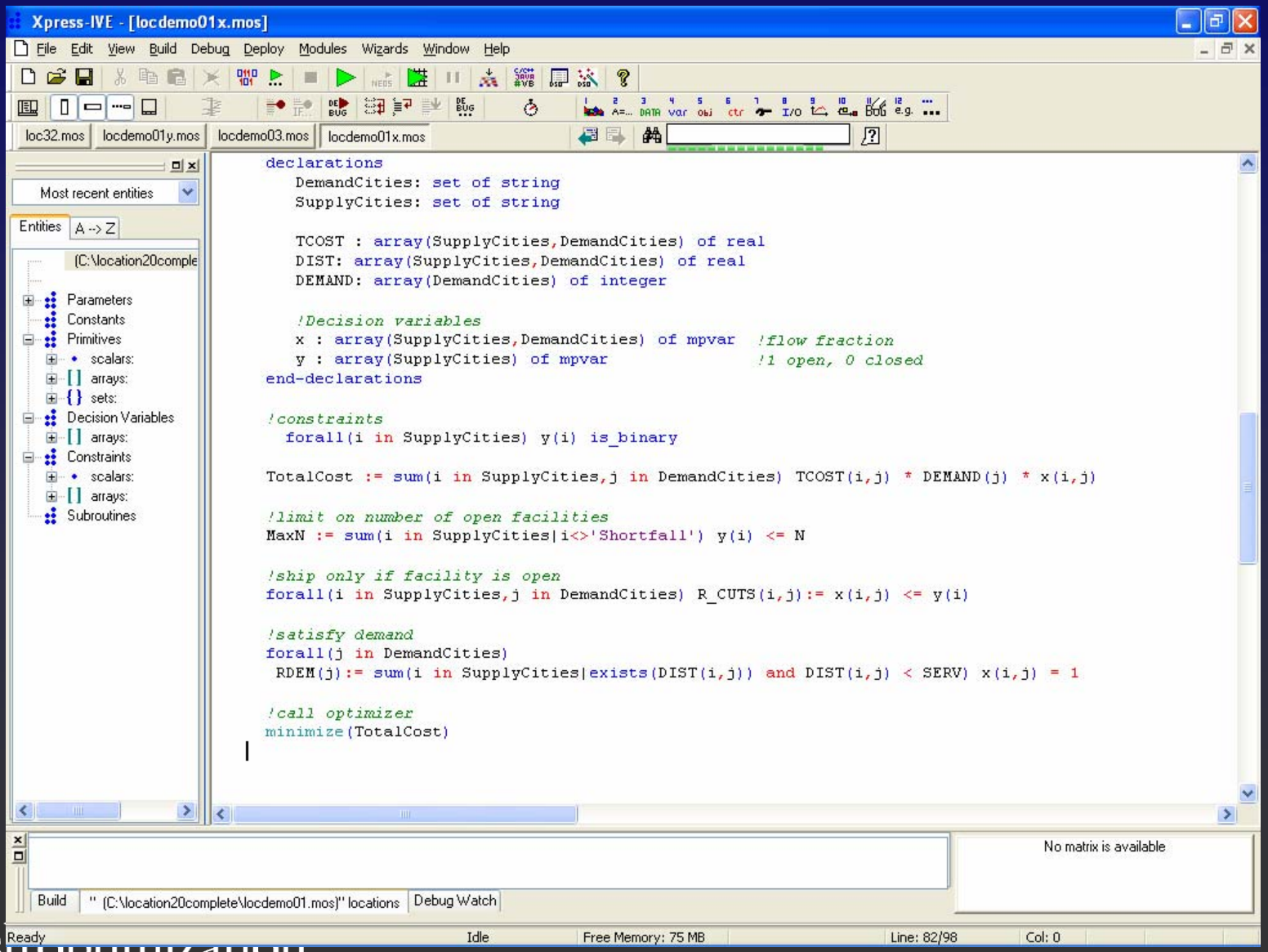
- Multi-item, multi-period production planning



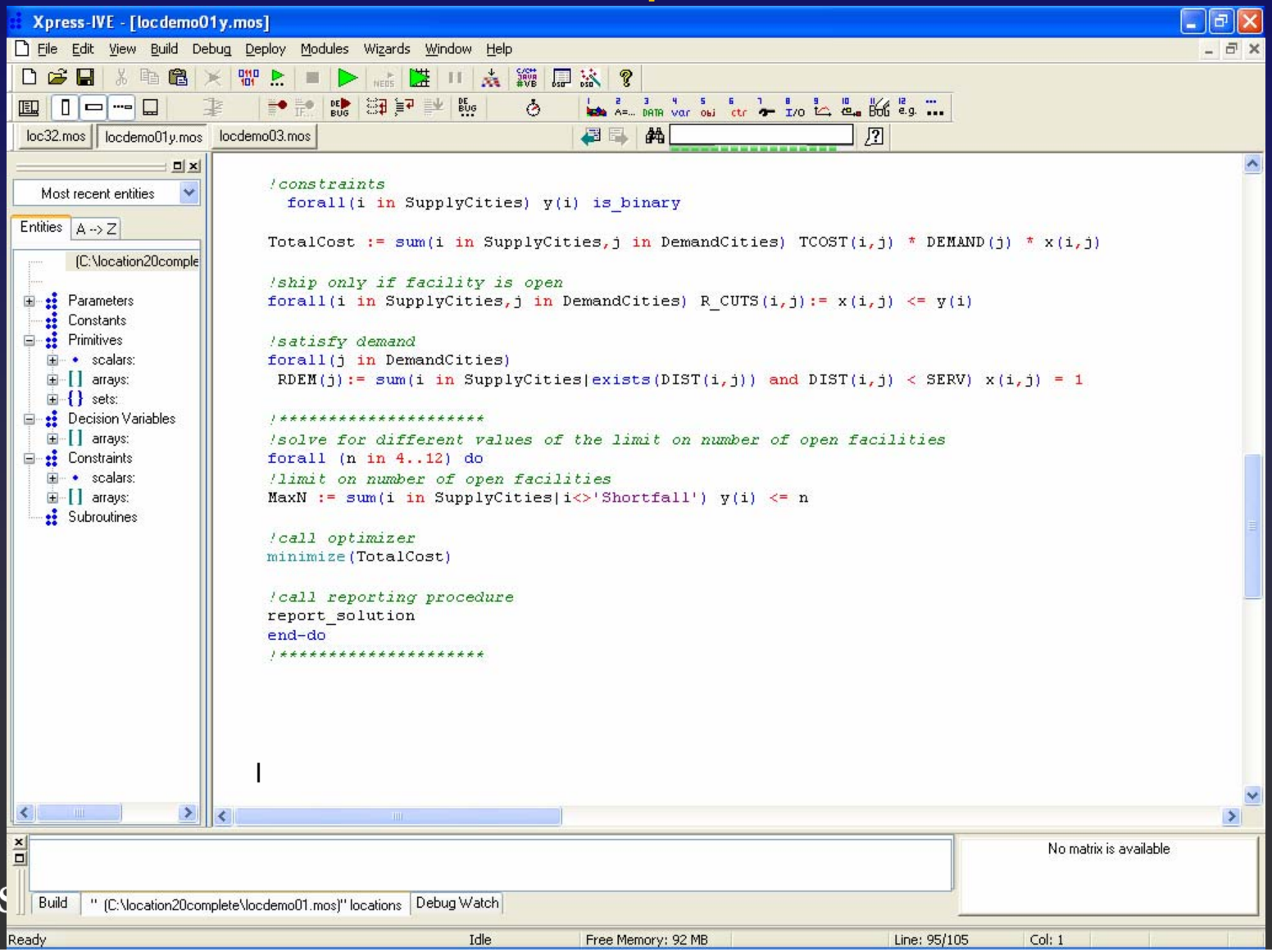
Dantzig-Wolfe decomposition



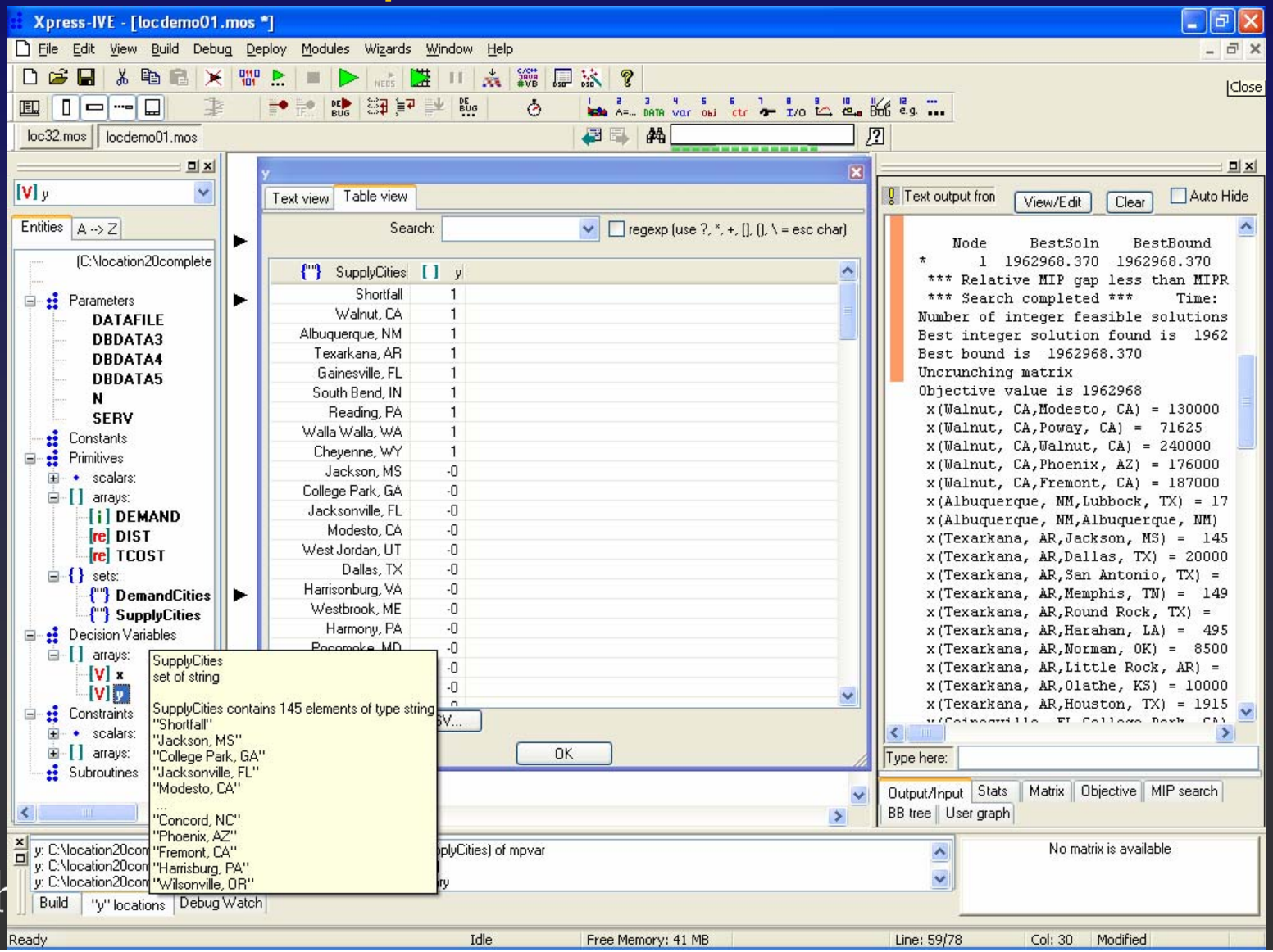
- Modeling statements with Xpress-Mosel and Xpress-IVE



Modeling and programming statements with Xpress-Mosel

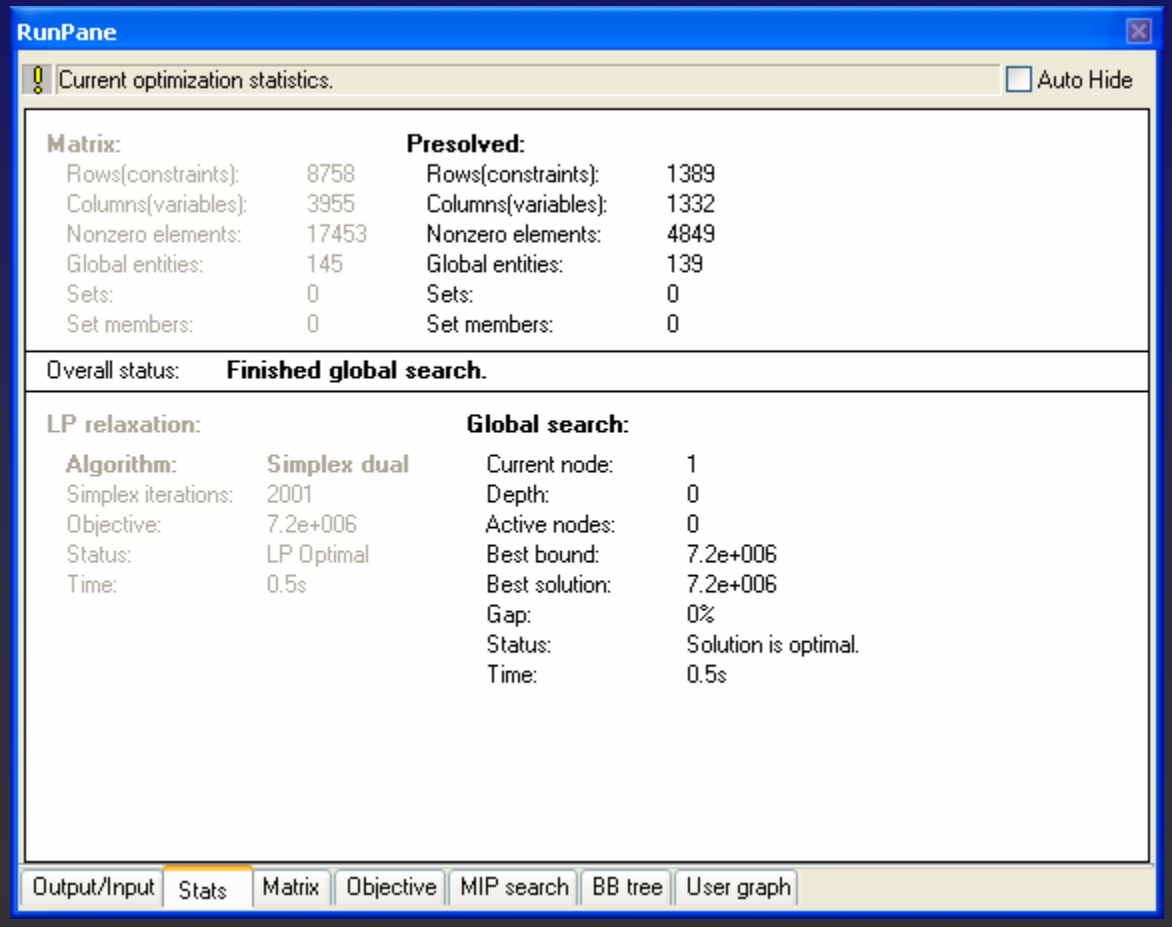


Browsing the solution and model entities in Xpress-IVE



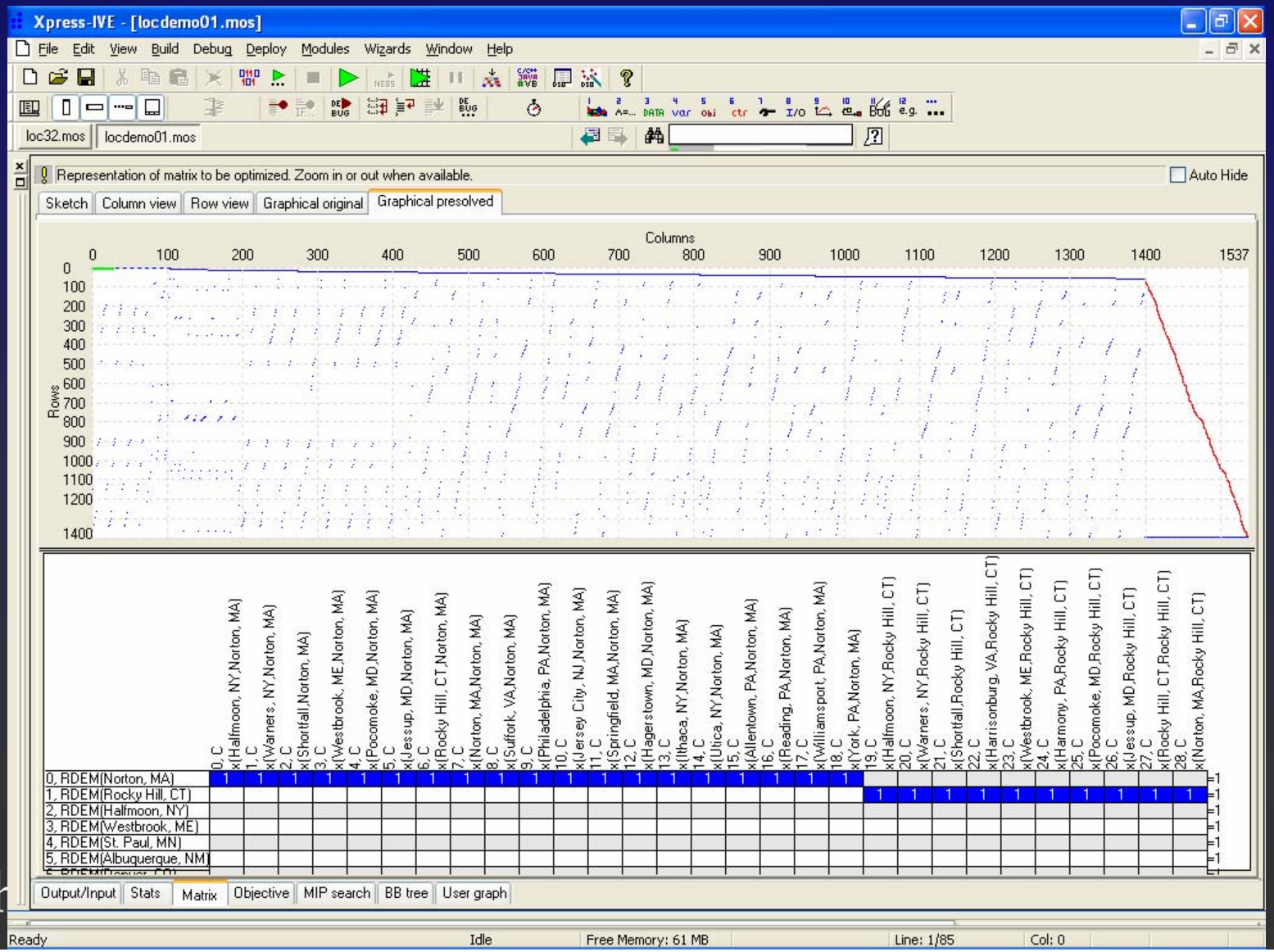
dash

Run statistics in Xpress-IVE

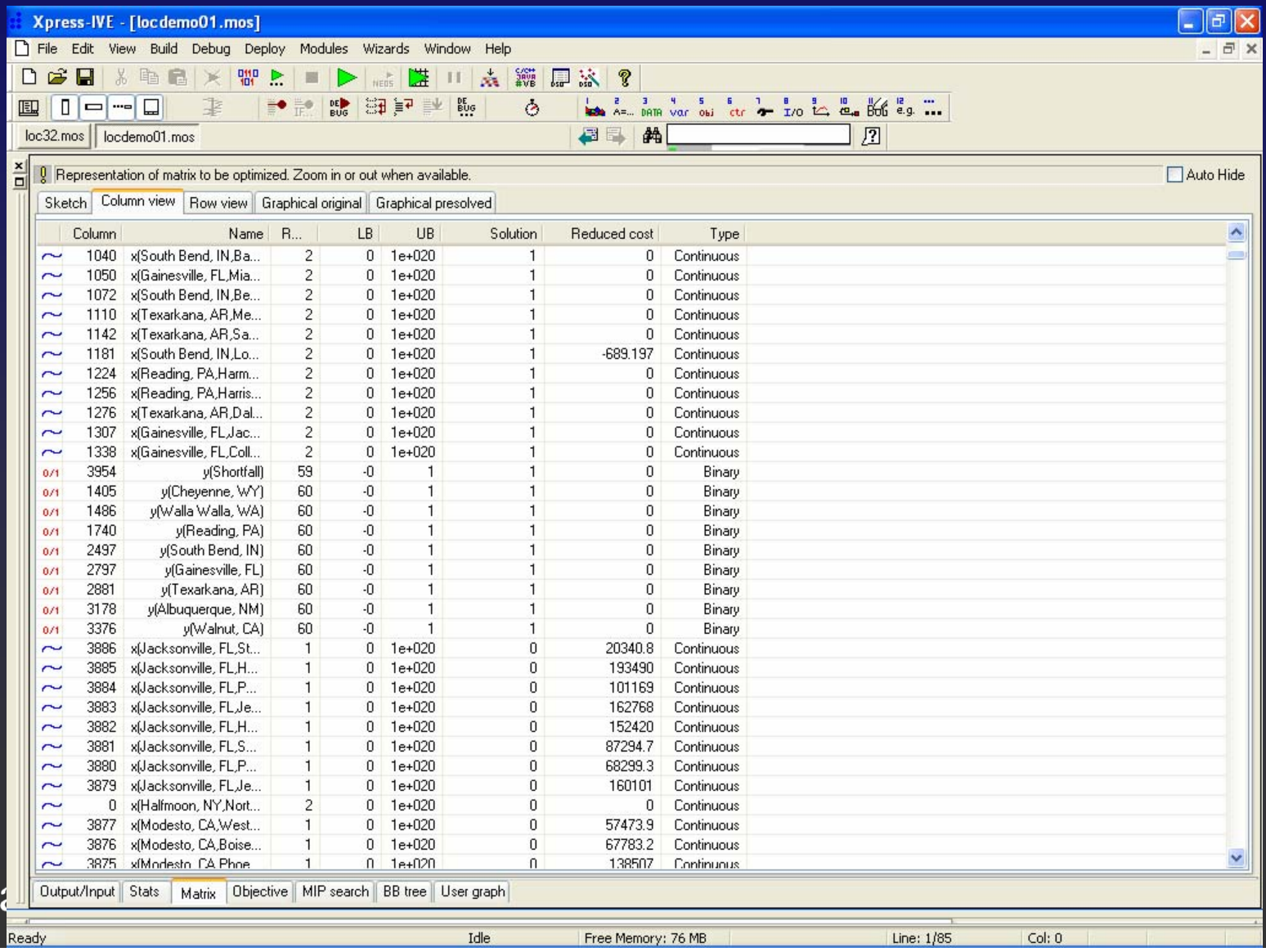


dash optimization

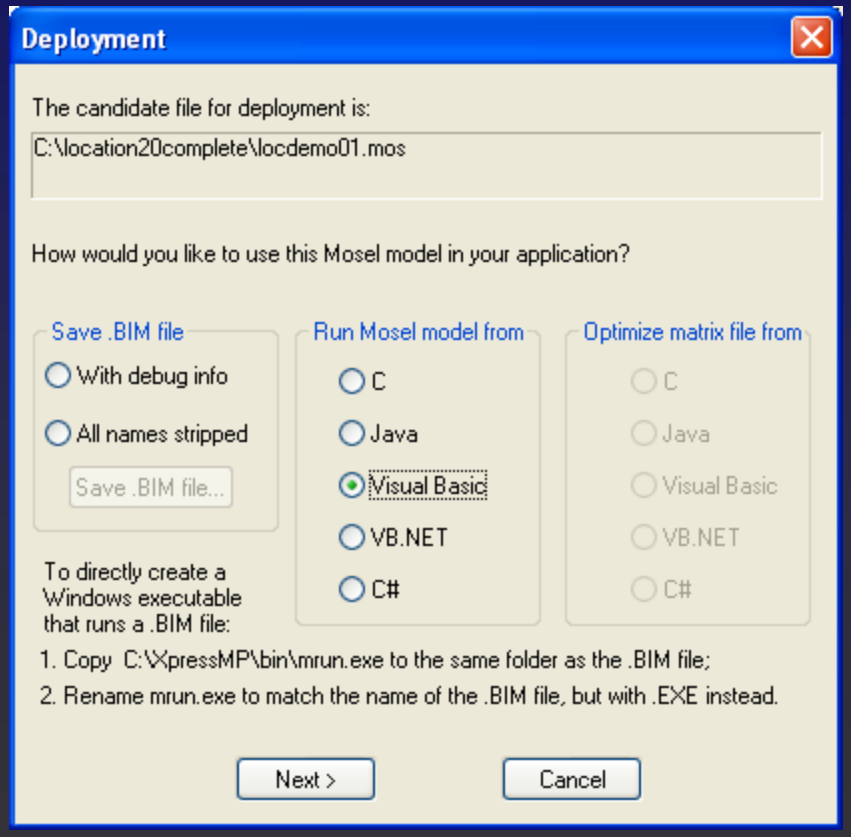
Problem matrix in Xpress-IVE



Problem matrix/solution in Xpress-IVE

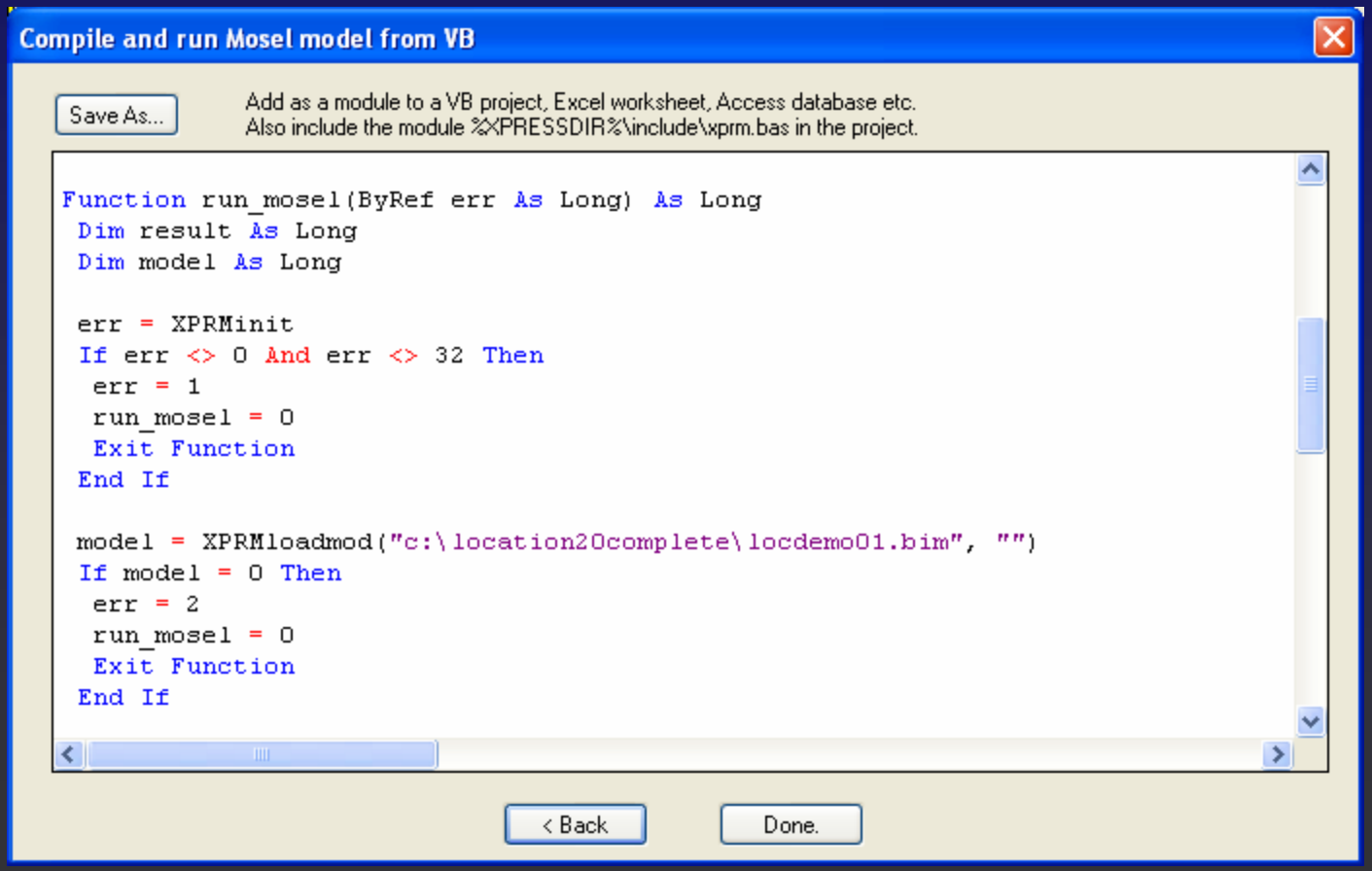


Deployment wizard in Xpress-IVE



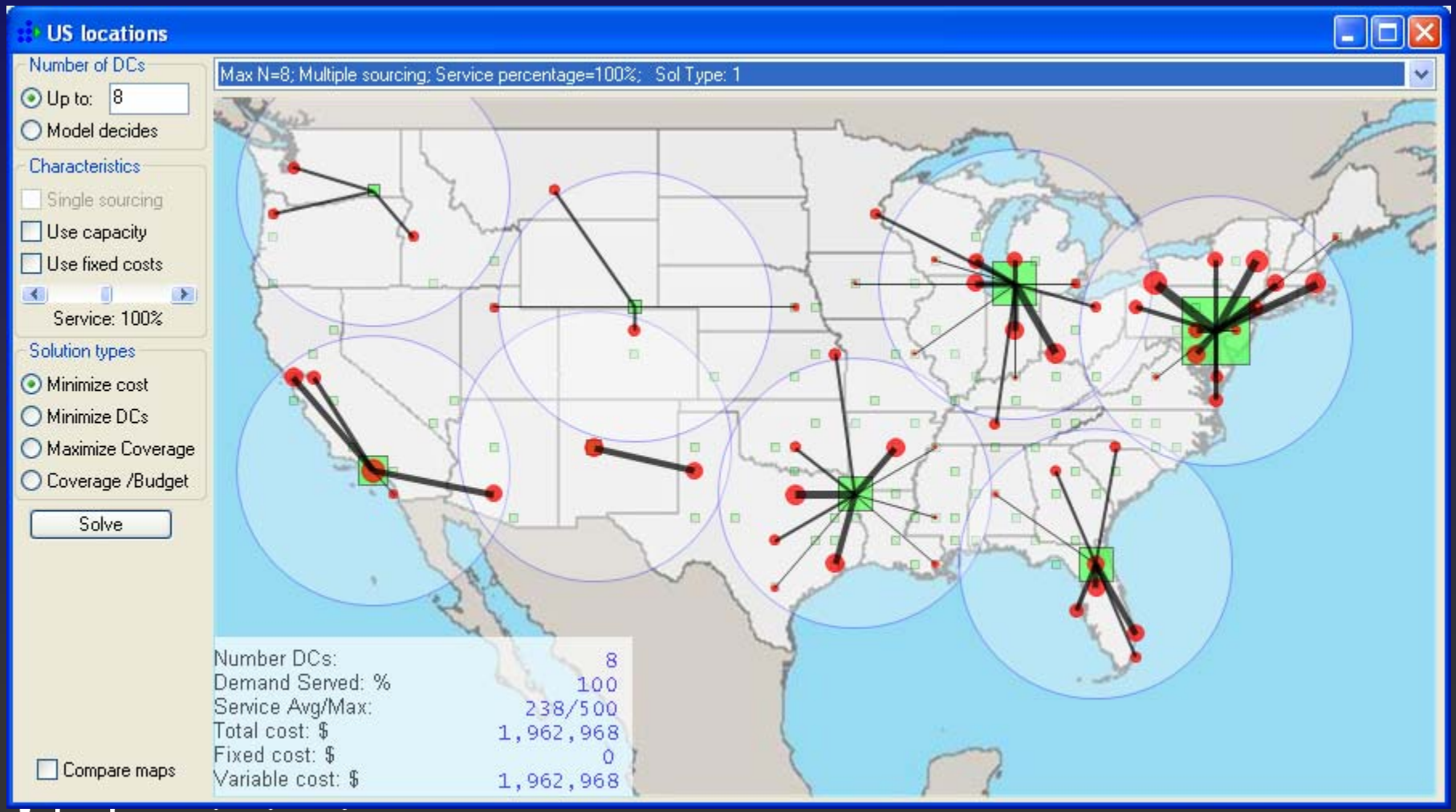
dash optimization

Deployment wizard in Xpress-IVE



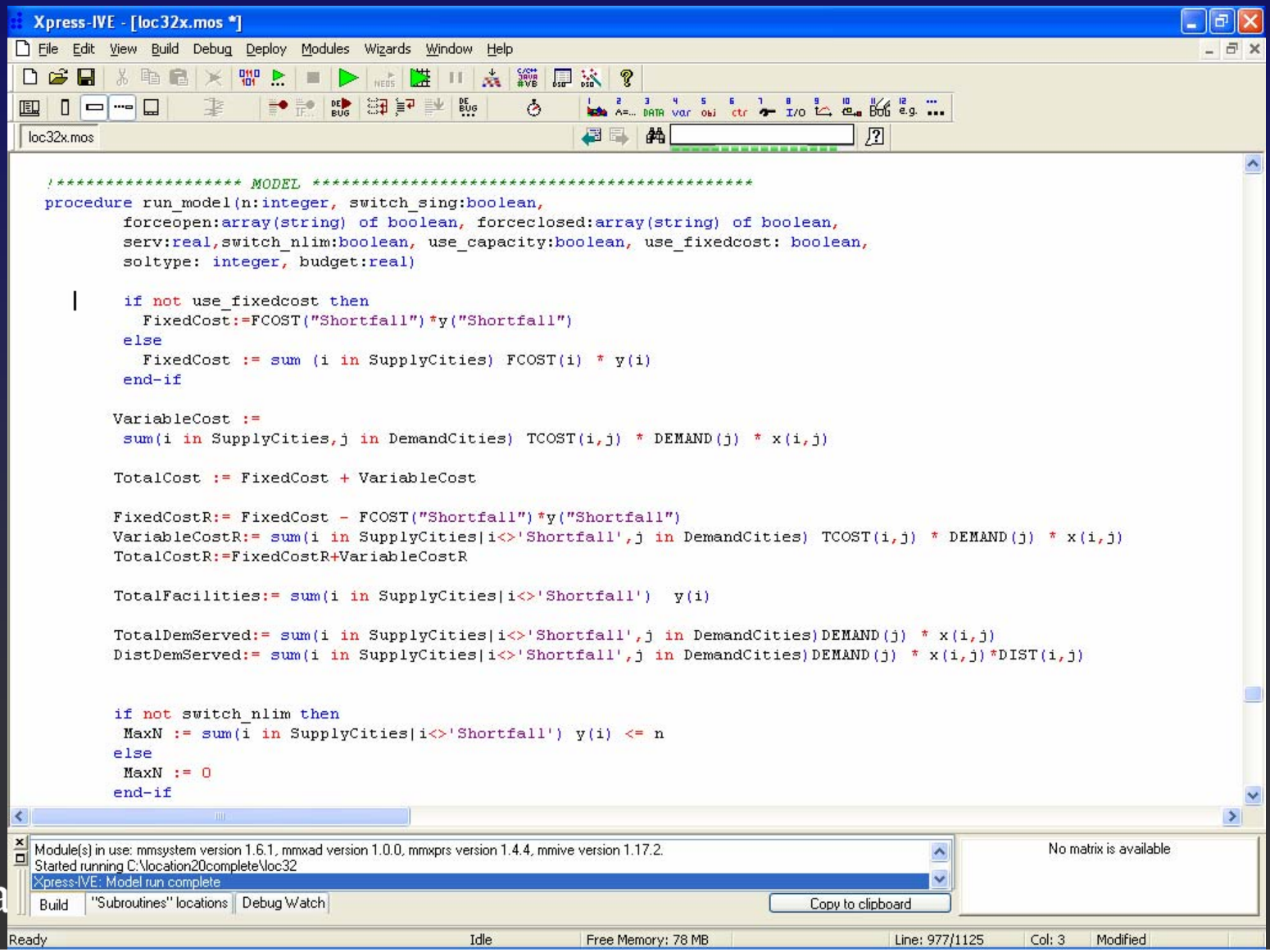
dash optimization

Visualization, model controls, and mapping with Xpress-XAD

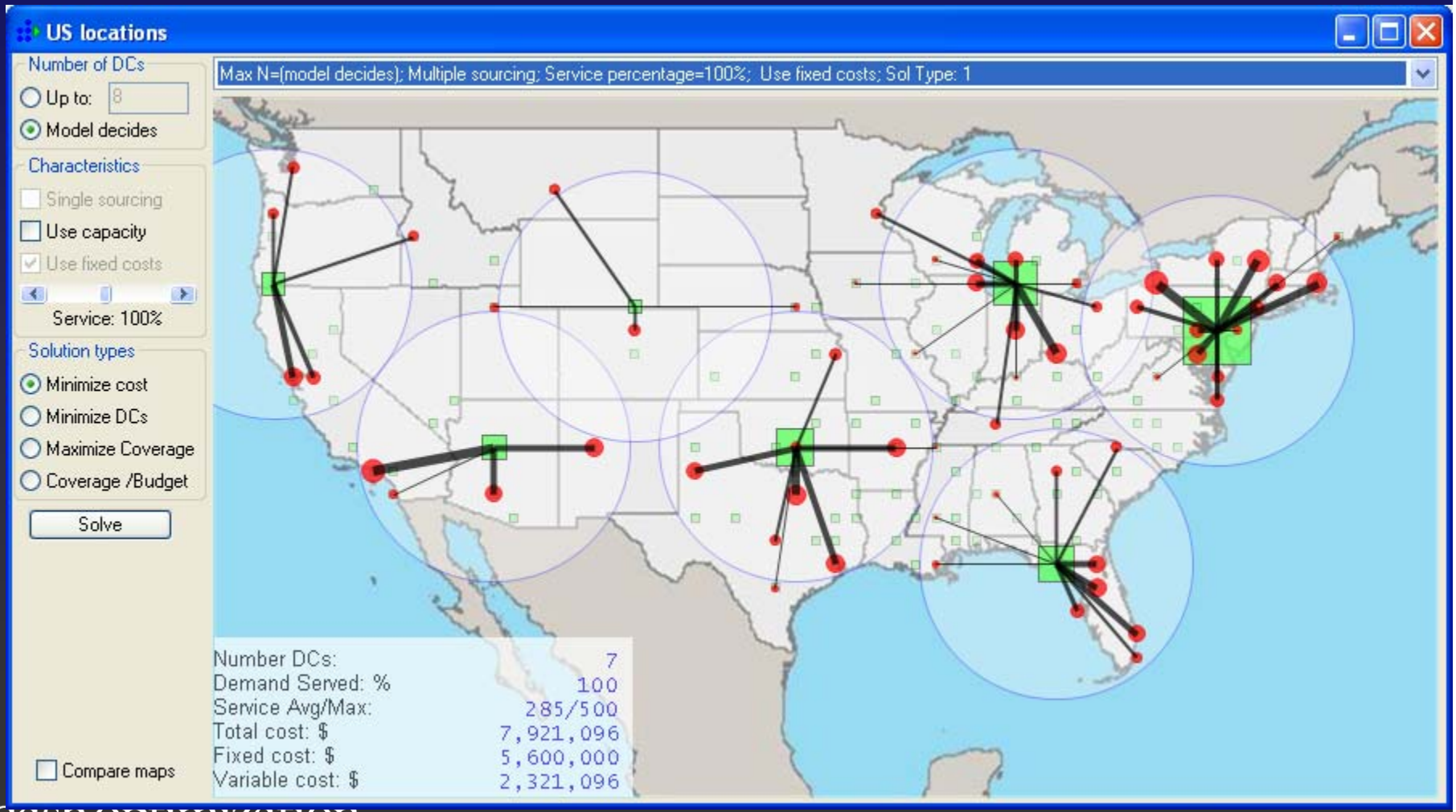


dash optimization

One Program: Mosel model as procedure called from GUI

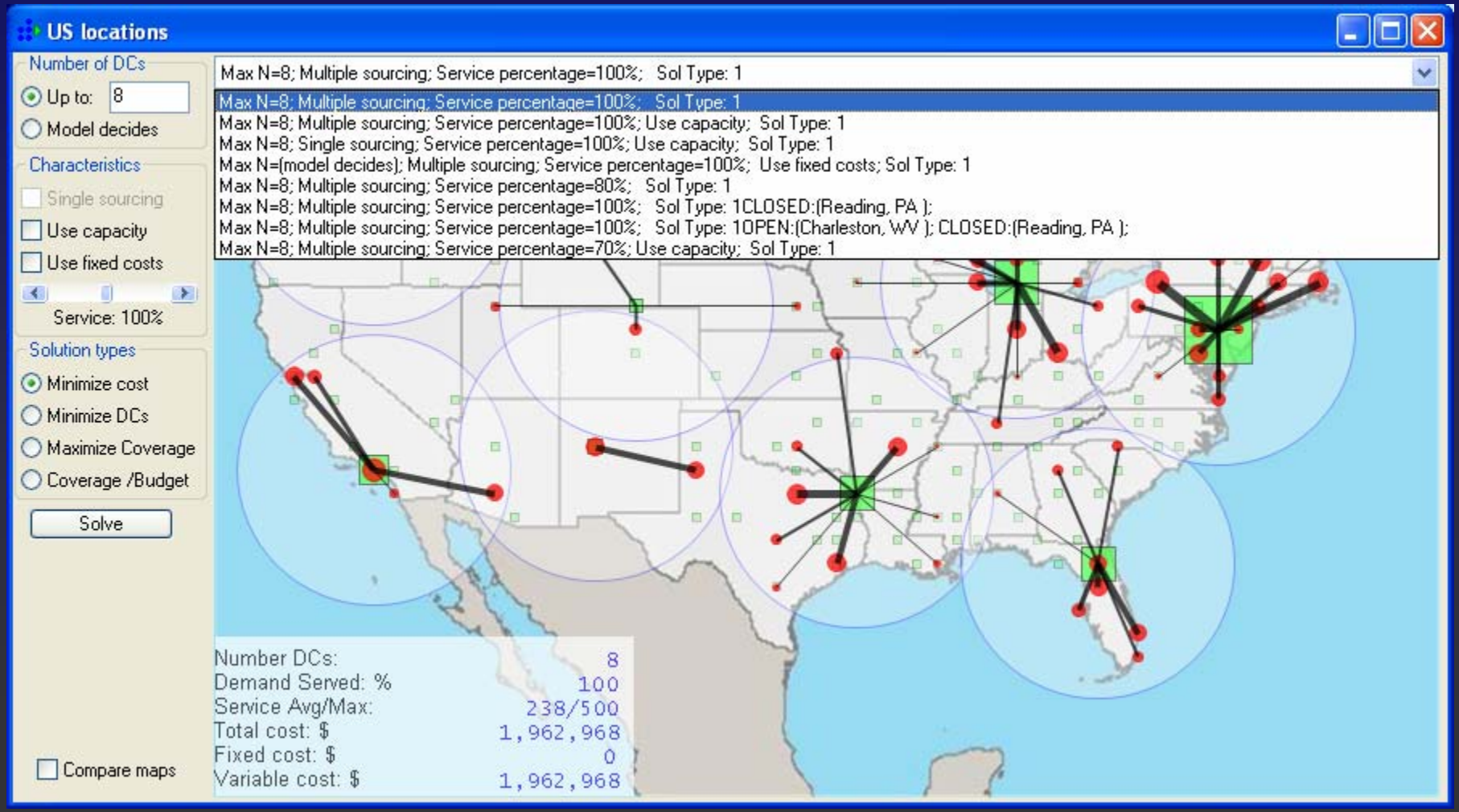


Interactive solving through GUI with all data in computer memory



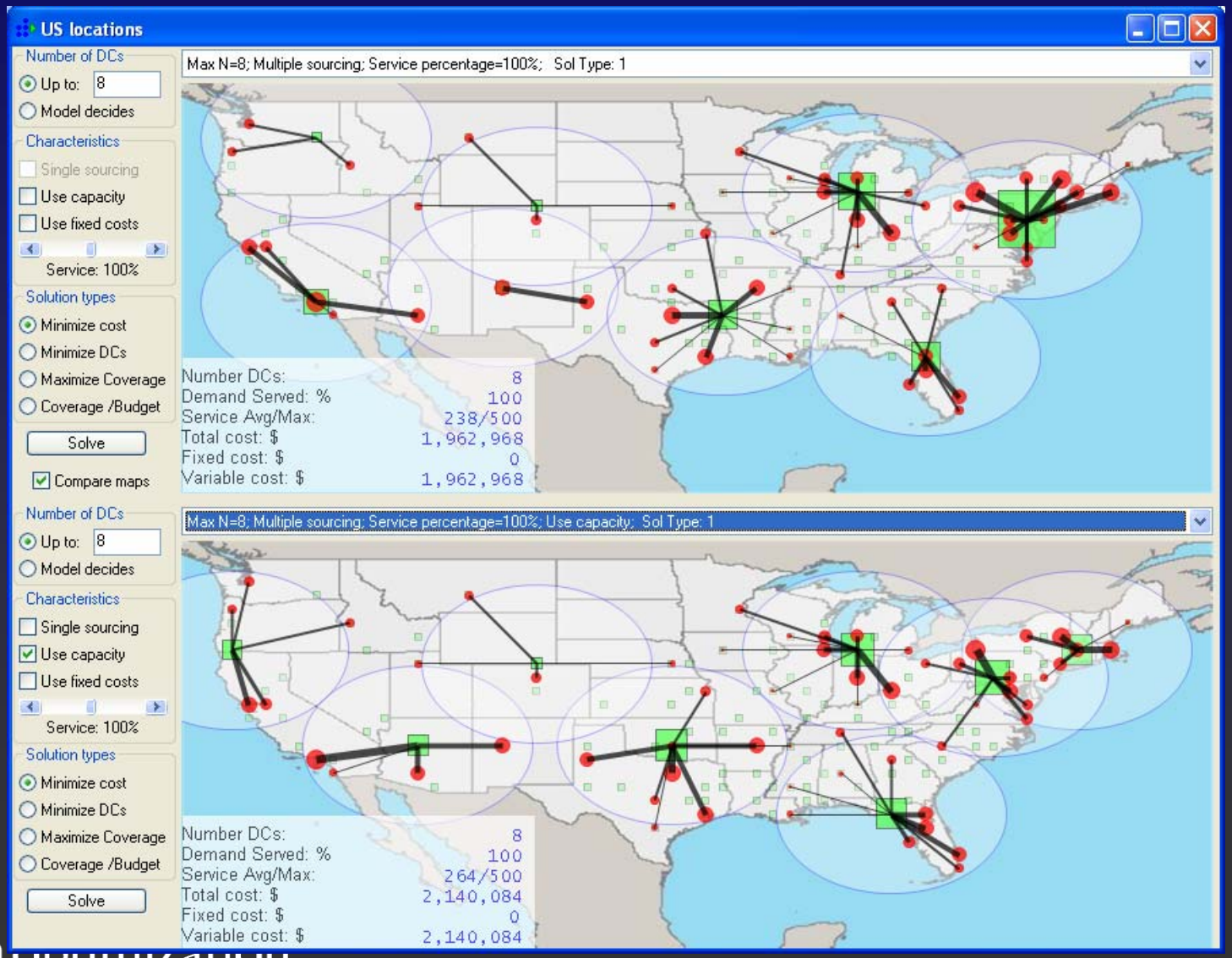
dashoptimization

Scenario Management



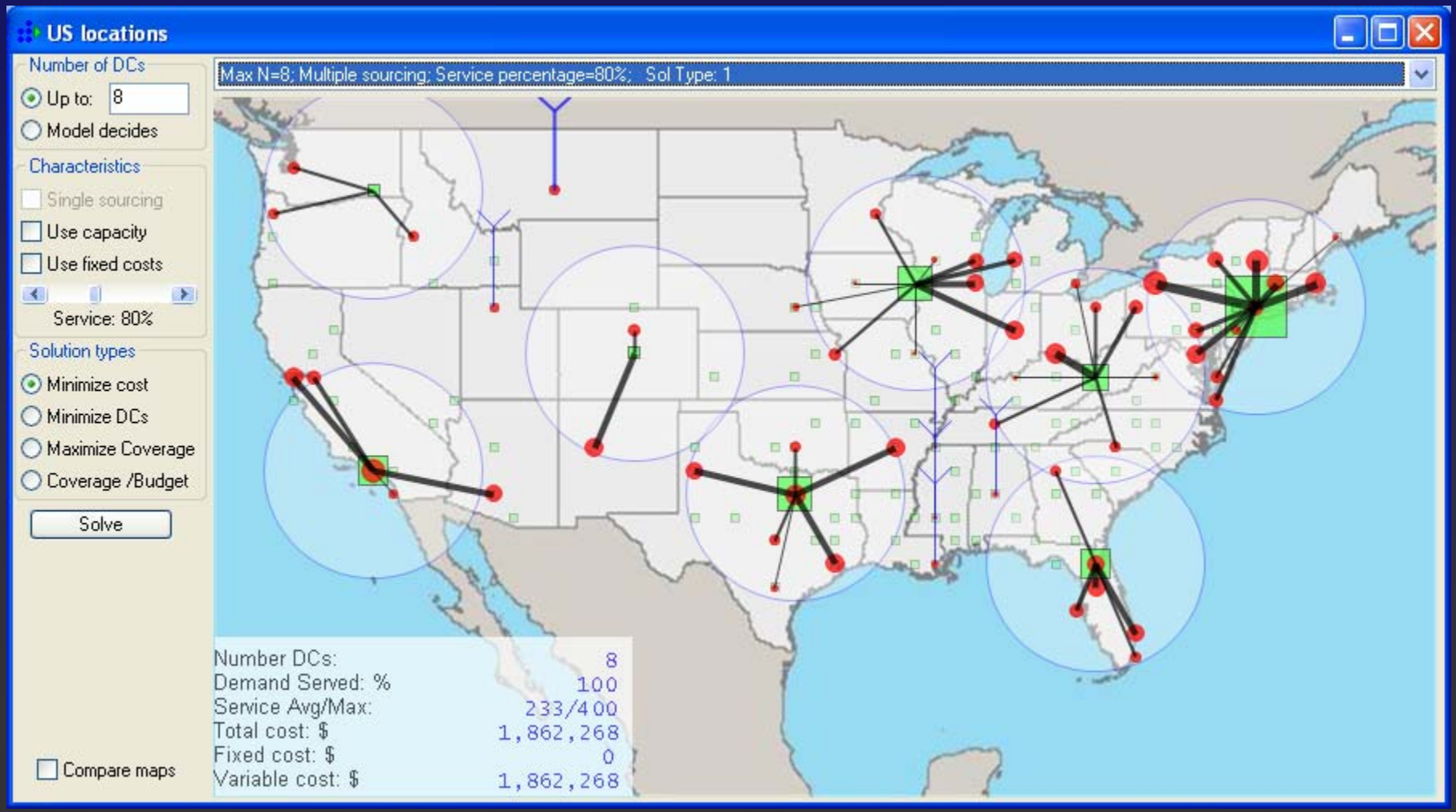
dash optimization

Scenario comparison/visualization



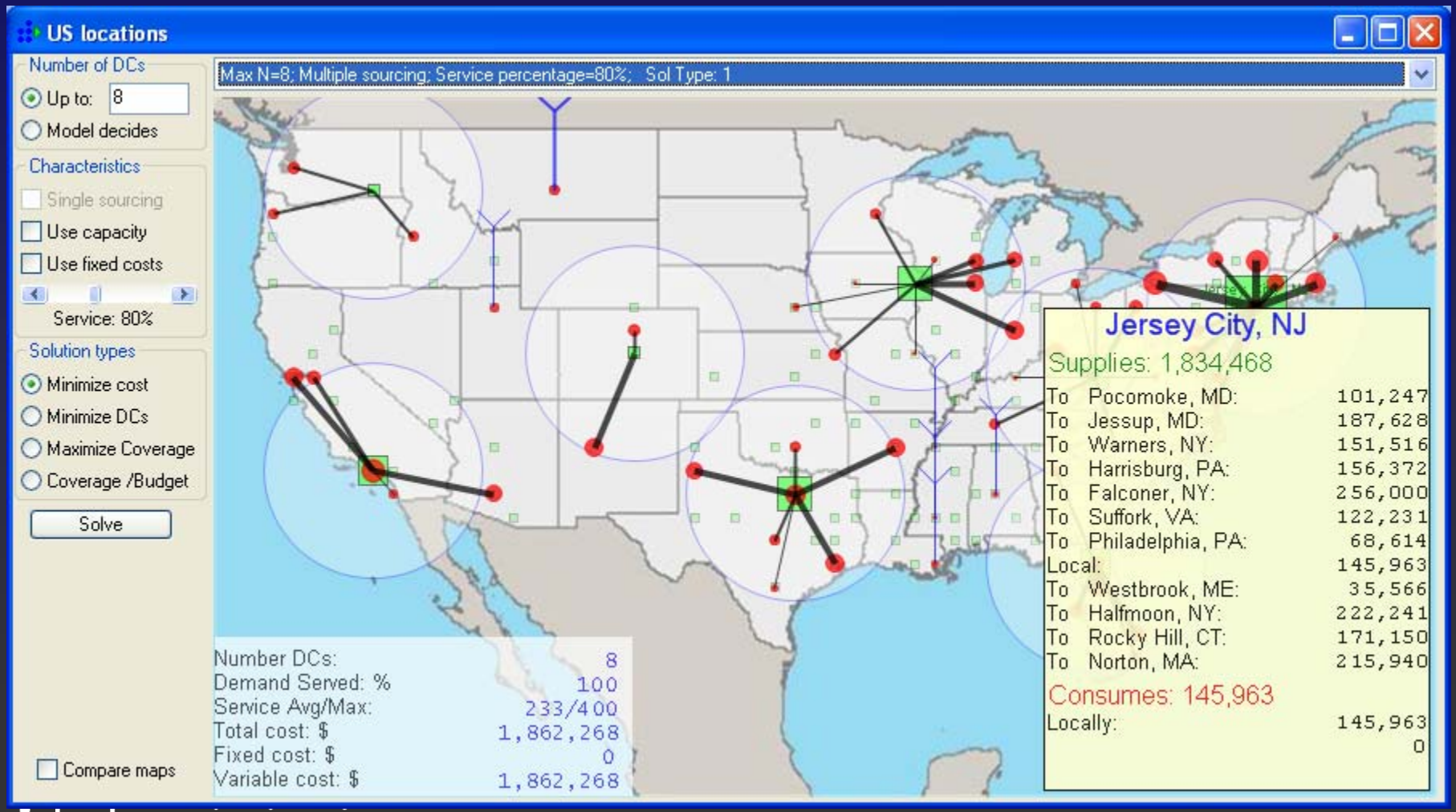
dashoptimization

Scenario with stricter service miles requirements and shortfalls



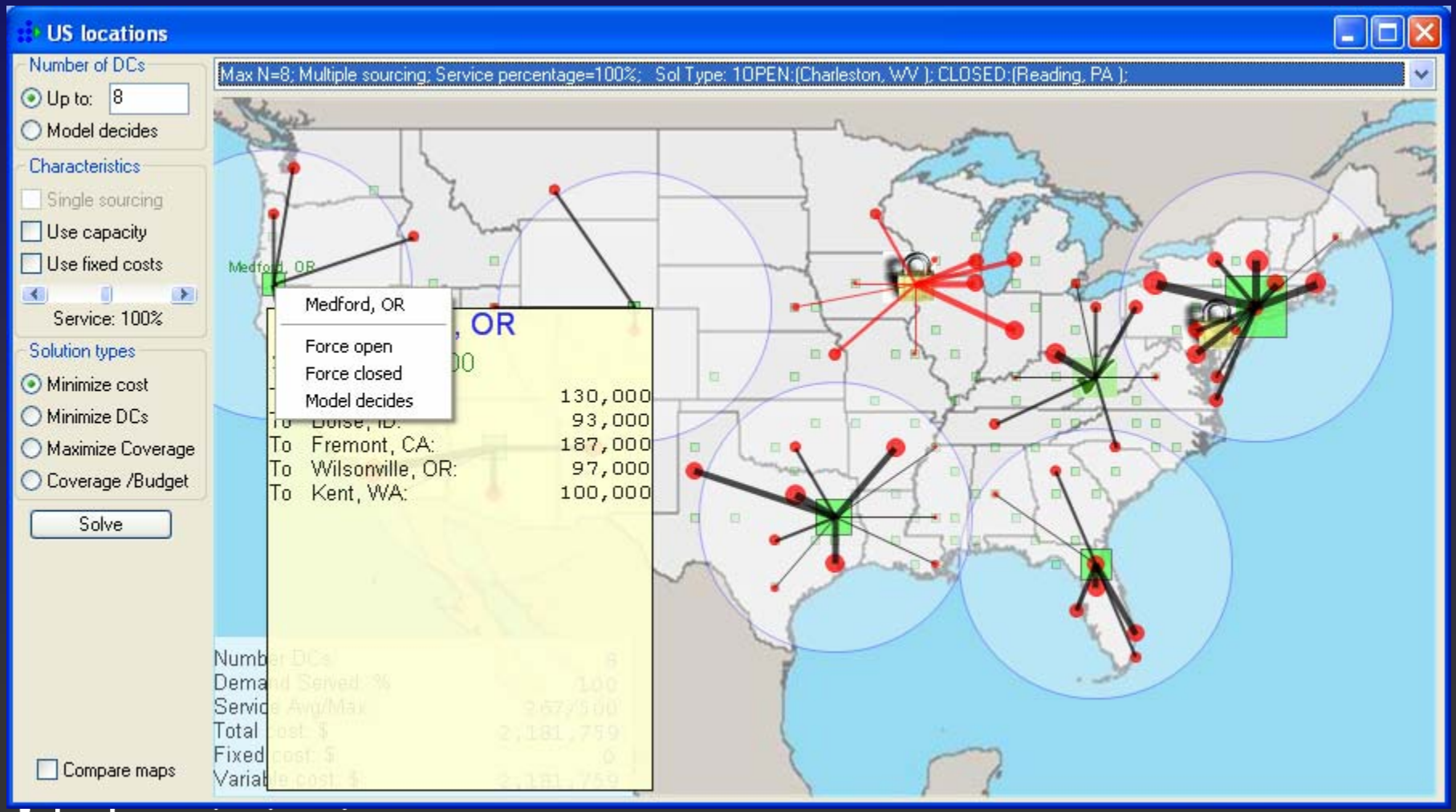
dash optimization

Visualization: Pop-up information



dash optimization

Visualization: Forcing facilities open or closed



dash optimization


dash optimization