Model Types Description

Constrained Nonlinear Systems

Mathematically, a Constrained Nonlinear System (CNS) model looks like:

$$\begin{array}{rl} \textbf{find} & x \\ & \\ & F(x) = 0 \\ \textbf{subject to} & L \le x \le U \\ & G(x) \le b \end{array}$$

where F and x are of equal dimension and the variables x are continuous. The (possibly empty) constraints $L \le x \le U$ are not intended to be binding at the solution, but instead are included to constrain the solution to a particular domain or to avoid regions where $F(x)$ is undefined. The (possibly empty) constraints $G(x) \le b$ are intended for the same purpose and are silently converted to equations with bounded slacks.

The CNS model is a generalization of the square system of equations $F(x) = 0$. There are a number of advantages to using the CNS model type (compared to solving as an NLP with a dummy objective, say), including:

- A check by GAMS that the model is really square,
- solution/model diagnostics by the solver (e.g. singular at solution, locally unique solution), and
- potential improvement in solution times, by taking better advantage of the model properties.

For information on CNS solvers that can be used through GAMS see the Solver/Model type Matrix.

Nonlinear Programming with Discontinuous Derivatives

Mathematically, the Nonlinear Programming with Discontinuous Derivatives (DNLP) Problem looks like:

$$\begin{array}{rl} \textbf{Minimize} & f(x) \\ \textbf{st} & g(x) \le 0 \\ & L \le x \le U \end{array}$$

where x is a vector of variables that are continuous real numbers. $f(x)$ is the objective function, and $g(x)$ represents the set of constraints. L and U are vectors of lower and upper bounds on the variables. This is the same as NLP, except that *non-smooth* functions (`abs, min, max`) can appear in $f(x)$ and $g(x)$.

For information on DNLP solvers that can be used through GAMS see the Solver/Model type Matrix.

Linear Programming

Mathematically, the Linear Programming (LP) Problem looks like:

```
Minimize or maximize   cx
          subject to   Ax ≥ b
                       L ≤ x ≤ U
```

where x is a vector of variables that are continuous real numbers. cx is the objective function, and $Ax \geq b$ represents the set of constraints. L and U are vectors of lower and upper bounds on the variables.

GAMS supports both free variables (unrestricted), positive variables, and negative variables. In addition user specified lower and upper bounds can be provided.

In a GAMS model equations are specified as a combination of less-than-or-equal-to or greater-than-or-equal-to inequalities and equalities (equal-to constraints).

For information on LP solvers that can be used through GAMS see the Solver/Model type Matrix.

MCP Models

Mathematically, the Mixed Complementarity Problem (MCP) looks like:

```
Find       z, w, v
such that  F(z) = w-v
           l ≤ z ≤ u, w ≥ 0, v ≥ 0
           w'(z-l) = 0, v'(u-z) = 0
```

MCP's are a class of mathematical programs which can be formulated in GAMS and solved by one of the MCP solvers that are hooked up to GAMS: see the Solver/Model type Matrix.

MCP's arise in many application areas including applied economics, game theory, structural engineering and chemical engineering.

Complementarity problems are easily formulated in the GAMS language. The only additional requirement beyond general NLP's is the definition of complementarity pairs.

MCP's constitute a fairly general problem class. It encompasses systems of nonlinear equations, non-linear complementarity problems and finite dimensional variational inequalities. Also inequality-constrained linear, quadratic and nonlinear programs are MCP's (although for these problems you may expect specialized solvers to do better). For instance, when we set the lower bounds *l* to minus infinity and *u* to plus infinity, both *w* and *v* have to be zero. This results in the problem

```
       Find        z
       such that   F(z) = 0
```

which is a system of non-linear equations.

Mixed Integer Nonlinear Programming

Mathematically, the Mixed Integer Nonlinear Programming (MINLP) Problem looks like:

```
      Minimize     f(x) + Dy
            st     g(x) + Hy ≤ 0
                   L ≤ x ≤ U
                   y = {0,1,2,..}
```

where *x* is a vector of variables that are continuous real numbers. *f(x) + Dy* is the objective function, and *g(x) + Hy* represents the set of constraints. *L* and *U* are vectors of lower and upper bounds on the variables.

For information on MINLP solvers that can be used through GAMS see the Solver/Model type Matrix.

Mixed Integer Programming

Mathematically, the Mixed Integer Linear Programming (MIP) Problem looks like:

```
      Minimize     cx + dy
            st     Ax + By ≥ b
                   L ≤ x ≤ U
                   y = {0,1,2,..}
```

where *x* is a vector of variables that are continuous real numbers, and *y* is a vector in variables that can only take integer values. *cx + dy* is the objective function, and *Ax + By ≥ b* represents the set of constraints. *L* and *U* are vectors of lower and upper bounds on the continuous variables, and *y={0,1,2,..}* is the integrality requirement on the integer variables *y*.

For information on MIP solvers that can be used through GAMS see the Solver/Model type Matrix.

Mathematical Program with Equilibrium Constraints

Mathematically, the Mathematical Program with Equilibrium Constraints (MPEC) Problem looks like:

```
Maximize or Minimize    f(x,y)
          subject to    g(x,y) ≤ 0
                        Lx ≤ x ≤ Ux
                        F(x,y) perp-to Ly ≤ y ≤ Uy
```

where *x* and *y* are vectors of continuous real variables. The variables *x* are often called the control or upper-level variables, while the variables *y* are called the state or lower-level variables. *f(x,y)* is the objective function. *g(x,y)* represents the set of traditional (i.e. NLP-type) contraints; in some cases, they can only involve the control variables *x*. The function *F(x,y)* and the bounds *Ly* and *Uy* define the equilibrium constraints. If *x* is fixed, then *F(x,y)* and the bounds *Ly* and *Uy* define an MCP; the **perp-to** indicates that such a complementary relationship holds. From this definition, we see that the MPEC model type contains NLP and MCP models as special cases of MPEC.

While the MPEC model formulation is very general, it also results in problems that are very difficult to solve. Work on MPEC algorithms is not nearly so advanced as that for the other model types. As a result, the MPEC solvers included in the GAMS distribution are experimental or beta versions.

For information on MPEC solvers that can be used through GAMS see the Solver/Model type Matrix.

Nonlinear Programming

Mathematically, the Nonlinear Programming (NLP) Problem looks like:

```
Minimize    f(x)
      st    g(x) ≤ 0
            L ≤ x ≤ U
```

where *x* is a vector of variables that are continuous real numbers. *f(x)* is the objective function, and *g(x)* represents the set of constraints. *L* and *U* are vectors of lower and upper bounds on the variables.

For information on NLP solvers that can be used through GAMS see the Solver/Model type Matrix.

Quadratically Constrained Programs

A Quadratically Constrained Program (QCP) is a special case of the NLP in which all the nonlinearities are required to be quadratic. As such, any QCP model can also be solved as an NLP. However, most "LP" vendors provide routines to solve LP models with a quadratic objective. Some allow quadratic constraints as well. Solving a model using the QCP model type allows these "LP" solvers to be used to solve quadratic models as well as linear ones. Some NLP solvers may also take advantage of the special (quadratic) form when solving QCP models.

In case a model with quadratic constraints is passed to a QCP solver that only allows a quadratic objective, a capability error (solver status 6 CAPABILITY PROBLEMS) will be returned.

For information on QCP solvers that can be used through GAMS see the Solver/Model type Matrix.

Mixed Integer Quadratically Constrained Programs

A Mixed Integer Quadratically Constrained Program (MIQCP) is a special case of the MINLP in which all the nonlinearities are required to be quadratic. For details see the description of the QCP, a special case of the NLP.

For information on MIQCP solvers that can be used through GAMS see the Solver/Model type Matrix.