The solver for nonlinear models

Outline

The commands for nonlinear models

Sapienza Università di Roma

Learning objectives (1/2)

At the end of this chapter you will be able to:

- write nonlinear functions with functional constraints $y = f(x)$ and ratios with bilinear constraints;
- choose tolerances and reformulations that give good scaling;

Learning objectives (2/2)

At the end of this chapter you will be able to:

- read the multipliers of a convex QP with `Pi`;
- estimate a shadow price by perturbation when the solver gives no duals.

Prerequisites

- From the module *LP Modelling*, Chapter 3 «The solver: building, solving, interpreting»: the general `gurobipy` commands: building, solving, reading the solution
- From this module, Chapter 2 «Background: convexity, quadratic programming and KKT conditions»: convex QPs and the KKT multipliers

The commands for nonlinear models

Introduction: The commands for nonlinear models

- A single solver for the whole laboratory
- Gurobi covers LP and QP natively (including non-convex ones, with `m.Params.NonConvex = 2`)
- From version 12 it also solves *globally* models with general nonlinear functions
- We see how to write these models, how to handle tolerances and scaling, and what to do when the duals are not there

Functional constraints and bilinear terms

- Nonlinear functions — logarithms, exponentials, powers — enter the model through **functional constraints**
- An auxiliary variable y is introduced and $y = f(x)$ is imposed
- Ratios such as $1/(\mu - \lambda)$ are instead reformulated with an auxiliary variable and a **bilinear constraint**:

$$w\,(\mu - \lambda) = 1$$

Functional constraints and bilinear terms: the code

```
# functions handled exactly (globally)
m.Params.FuncNonlinear = 1
z = m.addVar(lb=-GRB.INFINITY)
# z = log(g);  also Exp, Pow, Sin, ...
m.addGenConstrLog(g, z)
# bilinear terms: NonConvex = 2 is required
m.addQConstr(w * v == 1)
```

Tolerances and scaling

Two practical devices, both used in the scripts of these lecture notes:

- **Tolerances**: for accurate marginal analyses (differences between two nearby optima) tighten `MIPGap`, `FeasibilityTol` and `OptimalityTol` to 10^{-9}
- **Scaling**: reformulate to avoid tiny quantities — for instance

$$q \leq A\, p^{-\varepsilon} \quad \text{is written} \quad q \cdot r \leq A \text{ with } r = p^{\varepsilon}$$

  which keeps the numbers in a healthy range

When the duals are not there

- For models with functional or bilinear constraints the solver **does not provide the duals**
- Shadow prices and multipliers are estimated *by perturbation*:
  1. increase the right-hand side by ε
  2. re-optimize
  3. take the incremental ratio
- It is the same protocol as in the background chapter

Why a single solver

- Same syntax, same interpretation checklist
- Above all: a *certified global* optimum even in non-convex problems
- With a local solver every result would have to be accompanied by the question "is it only a local optimum?"
- Convex problems with a purely quadratic or conic structure (constraints such as $d_x^2 + d_y^2 \leq d^2$) require nothing special

Exercise: multiplier by perturbation

- Implement the QP min $x_1^2 + 2x_2^2$ with $x_1 + x_2 \geq 6$
- Print `Pi` and `Slack` of the constraint
- Verify the multiplier by re-optimizing with right-hand side 6.01

Solution

- The solver gives $\tilde{x} = (4, 2)$, value 24, $\texttt{Pi} = 8$, $\texttt{Slack} = 0$
- With right-hand side 6.01 the value is 24.080067: difference quotient 8.0067
- From $\tilde{f}(d) = \frac{2}{3}d^2$: the gap with 8 is the curvature, which vanishes at the margin

Exercise: a functional constraint

- Solve with Gurobi's functional constraints the one-variable problem

$$\max\ 5\log(1+x) - x, \qquad x \in [0, 10]$$

- Compare with the analytical solution:

$$\frac{5}{1+x} - 1 = 0 \quad \Rightarrow \quad x = 4$$

Solution

- Variables: $x \in [0, 10]$, $g = 1 + x$ and z free, with the functional constraint $z = \log(g)$ (`addGenConstrLog`)
- Objective linear in the new variables: $\max\ 5z - x$
- The solver returns $x = 4$ and optimal value $5 \log 5 - 4 \approx 4.047$, as the analytical solution

Summary

Nonlinear models:

- Nonlinear functions through functional constraints $y = f(x)$; ratios through bilinear constraints (`NonConvex = 2`)
- Tight tolerances for marginal analyses; reformulate for good scaling

Duals and solver:

- Without duals, shadow prices are estimated by perturbation
- A single solver: same syntax and certified global optimum

Final summary

- **General commands**: in LP Modelling
- **Nonlinear**: functional and bilinear constraints, tolerances and scaling
- **Multipliers**: `Pi` in convex QPs, exact at the margin; otherwise by perturbation