Nonlinear Optimization: A Comparison of Two Competing Approaches

Active-set SQP vs. IPM

Jan Fiala & Benjamin Marteau (Numerical Algorithms Group)

Introduction

Nonlinear programming is one of the main tools for mathematical modelling practitioners. The applications span many industries and academic fields such as:

- finance (portfolio optimization, model calibration)
- multiphysics modelling (oil and gas reservoir modelling, meteorology, climate simulations, engineering)
- statistics (machine learning, data fitting)

Such models can be formulated as

$$\begin{aligned} \min_{x\in\mathcal{R}^n} \quad & f(x) \\ \text{subject to} \quad & h(x) = 0 \\ & g(x) \geq 0 \end{aligned}$$

where the objective f and constraints h and g are sufficiently smooth nonlinear functions. In modern large scale applications, the number of variables n can be of an order of 10^4 or higher.

Solvers generally find a local solution (the best in a certain neighbourhood) satisfying the Karush–Kuhn–Tucker (KKT) optimality conditions. Two main competing approaches:

- active-set Sequential Quadratic Programming (SQP)
- Interior Point Method (IPM)

Both approaches play an important part in practice because of the fundamentally different ways of handling constraints.

In the rest of the poster, we compare two implementations of these methods in the NAG Library:

e04vh - nag_opt_sparse_nlp_solve (SQP) and **e04st** - nag_opt_handle_solve_ipopt (IPM)

Interior Point Method

Inequality constraints are tricky to handle due to their "combinatorial" nature: either the inequality constraint g_k is binding (is active) or it has no influence (its associated dual variable is 0). This is expressed by the KKT complementarity condition:

$$\mu_k g_k(x) = 0 \quad \text{for each} \quad k \qquad (1)$$

IPM does not tackle (1) directly, it works on its relaxation $\mu_k g_k(x) = \nu$ with $\nu > 0$.

Each iteration

- Performs one Newton iteration towards the solution of the relaxed KKT system
- Updates the current solution estimate and the relaxation parameter ν

IPM properties

- Perform few computationally expensive iterations
- Rely on efficient underlying linear algebra

IPM solvers scale very well to large NLP problems with a small number of constraints

Sequential Quadratic Programming

SQP methods try to guess which inequality constraints g_k are binding and iteratively refine that guess. Non-binding constraints can be discarded (have no influence) at the current iteration. The solver then works on the smaller space (null space) of the remaining constraints.

1. **Initialization**
   - Build a quadratic model of the problem
   - Take a first guess of the set of active constraints
2. **Each iteration**
   - Solves the quadratic model, warm start it by the active set estimation
   - Updates x_{k+1} and the guess of the active constraints
   - Builds a new quadratic model at x_{k+1}

SQP properties

- Perform lots of inexpensive iterations
- Work on the null space of the active constraints

⇒ The more active constraints there are, the cheaper the iterations become.

SQP solvers scale very well to large NLP problems with a high number of constraints

Demonstration on a few selected problems from the CUTEst test set

Highly constrained problems

Name	No. vars	No. constrs	SQP time (s)	IPM time (s)
MINC44	1113	1033	0.28	7.60
READING8	2002	1000	9.78	251.12
NCVXQP6	10000	7500	3.60	613.38
MADSSCHJ	201	398	0.34	5.51

Loosely constrained problems

Name	No. vars	No. constrs	SQP time (s)	IPM time (s)
JIMACK	3549	0	542.42	8.12
OSORIO	10201	202	303.00	0.78
TABLE8	1271	72	3.80	0.04
OBSTCLBL	10000	1	40.84	0.50

Recommendation of use

The number of the constraints is not the only factor affecting the convergence of the solvers:

IPM (e04st) advantages

- Efficient on unconstrained or loosely constrained problems
- Can exploit 2nd derivatives
- Efficient for quadratic problems
- Better use of multi-core architecture
- New and simpler interface

SQP (e04vh) advantages

- Efficient on highly constrained problems
- Can capitalize on a good initial point
- Stays feasible with respect to the linear constraints throughout the optimization
- Usually better results on pathological problems
- Usually requires less function evaluations
- Infeasibility detection
- Allows warm starting

NAG introduces at Mark 26 an interior point method (e04st) complementing the existing SQP solver (e04vh) for large NLP problems