

# Some Remarks on Constrained Optimization

José Mario Martínez

[www.ime.unicamp.br/~martinez](http://www.ime.unicamp.br/~martinez)

Department of Applied Mathematics, University of Campinas, Brazil

2011

# First Remarks

Every branch of **Mathematics is applicable**, directly or indirectly, to the “reality”.

**Optimization** is a mathematical problem with **many “immediate” applications** in the non-mathematical world.

**Optimization provides a model** for real-life problems. We use this model to take decisions, fit parameters, make previsions, understand and compress data, detect instability of models, recognize patterns, planning, finding equilibria, packing molecules, protein folding and alignment, etc.

We use **Optimization Software** to solve Optimization problems.

In Optimization one tries to find the **lowest possible values** of a real function  $f$  within some domain.

Roughly speaking, this is **Global Optimization**. Global Optimization is very hard. For approximating the global minimizer of a continuous function on a simple region of  $\mathbb{R}^n$  one needs to evaluate  $f$  on a **dense set**.

As a consequence one usually relies on **Affordable Algorithms** that do not guarantee global optimization properties but only local ones. (In general, convergence to stationary points.) Affordable algorithms run in **reasonable computer time**.

Even from the Global Optimization point of view, Affordable Algorithms are important, since we may use them many times, perhaps from **different initial approximations**, with the expectancy of finding lower and lower functional values in different runs.

## Dialogue between Algorithm A and Algorithm B

Algorithm A finds a stationary (KKT) feasible point with objective function value equal to 999.00 using 1 second of CPU time.

Algorithm B finds the (perhaps non-stationary) feasible point with objective function value equal to 17.00 using 15 minutes.

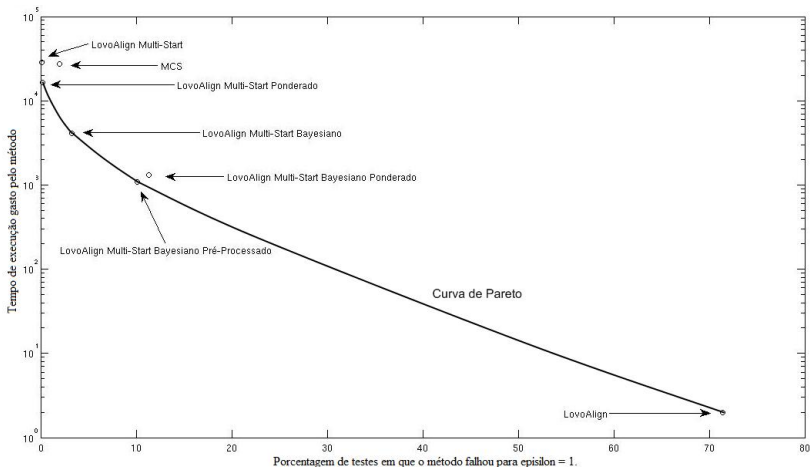
Algorithm B says: I am the best because my functional value is lower than yours.

Algorithm A says: If you give me 15 minutes I can run many times so that my functional value will be smaller than yours.

Algorithm B says: Well, Just do it!

# Time versus failures

1500 protein-alignment problems (from Thesis of P. Gouveia, 2011).



# Claim

**Affordable Algorithms** are usually compared on the basis of their behavior on the solution of a problem with a **given initial point**. This approach does not correspond to the necessities of most practical applications.

**Modern (Affordable) methods should incorporate the most effective heuristics and metaheuristics for choosing initial points, regardless the existence of elegant convergence theory.**

# Algencan

**Algencan** is an algorithm for constrained optimization based on traditional ideas (Penalty and Augmented Lagrangian) (PHR). At each (outer) iteration one finds an approximate minimizer of the objective function plus a shifted (quadratic) penalty function (**the Augmented Lagrangian**).

Subproblems, which involve minimization with simple constraints, are solved using **Gencan**.

**Gencan** is not a Global-Minimization method. However, it incorporates Global-Minimization **tricks**.

# Applying Augmented Lagrangian to Minimize $f(x)$ subject to $h(x) = 0, g(x) \leq 0, x \in \Omega$

- 1 Define, for  $x \in \mathbb{R}^n, \lambda \in \mathbb{R}^m, \mu \in \mathbb{R}_+^p, \rho > 0$ :

$$L_\rho(x, \lambda, \mu) = f(x) + \frac{\rho}{2} \left[ \left\| h(x) + \frac{\lambda}{\rho} \right\|^2 + \left\| \left( g(x) + \frac{\mu}{\rho} \right)_+ \right\|^2 \right].$$

- 2 At each iteration, minimize approximately  $L_\rho$  subject to  $x \in \Omega$ .
- 3 If **ENOUGH PROGRESS** was not obtained, **INCREASE**  $\rho$ .
- 4 Update and safeguard Lagrange multipliers  $\lambda \in \mathbb{R}^m, \mu \in \mathbb{R}_+^p$ .



## Why to safeguard

At the end of outer iteration  $k$  Algencan obtains Lagrange multipliers estimates

$$\lambda^{k+1} = \lambda^k + \rho_k h(x^k) \text{ and } \mu^{k+1} = (\mu^k + \rho_k g(x^k))_+.$$

$\lambda^k / \rho_k$  and  $\mu^k / \rho_k$  are the shifts employed at iteration  $k$ .

If (unfortunately)  $\rho_k$  goes to infinity, the only decision that makes sense is that the shifts must tend to zero. (It does not make sense infinite penalization with non-null shift.)

A simple way to guarantee that is to impose that the approximation Lagrange multipliers to be used at iteration  $k + 1$  must be bounded. We obtain that projecting them on a (large) box.

## When safeguarding is not necessary

If the sequence generated by Algencan converges to the feasible point  $x^*$ , which satisfies the Mangasarian-Fromovitz condition (and, hence, KKT) with only one vector of Lagrange multipliers, and, in addition, fulfills the second order sufficient optimality condition, then the penalty parameters remain bounded and the estimates of Lagrange multipliers converge to the true Lagrange multipliers.

## Feasibility Results

It is **impossible** to prove that a method always obtains feasible points because, ultimately, feasible points may not exist at all. All we can do is to guarantee that, in the limit, “stationary points of the infeasibility” are necessarily found.

Moreover, even if we know that feasible points exist, it is impossible to guarantee that an affordable method finds them.

“Proof”: Run your affordable method with an infeasible problem with only one stationary point of infeasibility. Your method converges to that point. Now, modify the constraints in a region that does not include the sequence generated by your method in such a way that the new problem is feasible. Obviously, your method generates the same sequence as before. Therefore, the affordable method does not find the feasible points.

# Optimality Results

Assume that Algencan generates a subsequence such that Infeasibility tends to zero.

Then, given  $\varepsilon > 0$ , for  $k$  large enough we have the following **AKKT** result:

① Lagrange:

$$\|\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \nabla g(x^k)\mu^{k+1}\| \leq \varepsilon;$$

② Feasibility:

$$\|h(x^k)\| \leq \varepsilon, \|g(x^k)_+\| \leq \varepsilon;$$

③ Complementarity:

$$\min\{\mu_i^{k+1}, -g_i(x^k)\} \leq \varepsilon \quad \text{for all } i.$$

# Stopping Criteria

The Infeasibility Results + the AKKT results suggest that the execution of Algencan should be stopped when one of the following criteria is satisfied:

- 1 The current point is infeasible and stationary for infeasibility with tolerance  $\varepsilon$ . (Infeasibility of the problem is suspected.)
- 2 The current point satisfies AKKT (Lagrange + Feasibility + Complementarity) with tolerance  $\varepsilon$ .

Theory: Algencan necessarily stops according to the criterion above, **independently of constraint qualifications**.

Algencon satisfies the stopping criterion and converges to feasible points that may not be KKT (and where the sequence of Lagrange multipliers approximation tends to infinity).

Algencon satisfies AKKT in the problem

$$\text{Minimize } x \quad \text{subject to } x^2 = 0.$$

Other methods (for example SQP) do not. SQP satisfies Feasibility and Complementarity but does not satisfy Lagrange in this problem.

# CAKKT

Algencan satisfies an even stronger stopping criterion. The Complementary Approximate KKT condition (CAKKT) says that, eventually:

- 1 Lagrange:

$$\|\nabla f(x^k) + \nabla h(x^k)\lambda^{k+1} + \nabla g(x^k)\mu^{k+1}\| \leq \varepsilon;$$

- 2 Feasibility:

$$\|h(x^k)\| \leq \varepsilon, \|g(x^k)_+\| \leq \varepsilon;$$

- 3 Strong Complementarity:

$$|\mu_i^{k+1} g_i(x^k)| \leq \varepsilon \quad \text{for all } i;$$

and

$$|\lambda_i^{k+1} h_i(x^k)| \leq \varepsilon \quad \text{for all } i.$$

However, CAKKT needs a slightly stronger assumption on the constraints:

The functions  $h_i$  and  $g_j$  should satisfy, locally, a “Generalized Lojasiewicz Inequality”, which means that the norm of the gradient grows faster than the functional increment.

This inequality is satisfied by every reasonable function. For example, analytic functions satisfy GLI.

The function  $h(x) = x^4 \sin(1/x)$  does not satisfy GLI. We have a counterexample showing that Algencan may fail to satisfy the CAKKT criterion when this function defines a constraint.

Should CAKKT be incorporated as standard stopping criterion of Algencan?



## Example concerning the Kissing Problem

The Kissing Problem consists of finding  $n_p$  points in the unitary sphere of  $\mathbb{R}^{n_d}$  such that the distance between any pair of them is not smaller than 1.

This problem may be modeled as Nonlinear Programming in many possible ways.

For  $n_d = 4$  and  $n_p = 24$  the problem has a solution. Using Algencan and random initial points uniformly distributed in the unitary sphere we find this solution in the Trial 147, using a few seconds of CPU time.

It is also known that, with  $n_d = 5$  and  $n_p = 40$  the problem has a solution. We used Algencan to find the global solution using random uniformly distributed initial points in the sphere, and we began this experiment on February 8, 2011, at 16.00 pm.

In February 9, at 10.52 am, Algencan had run 117296 times, and the best distance obtained was 0.99043038012718854.

The code is still running.

## Consequences:

- 1 Global Optimization is hard;
- 2 Stopping Criteria are not merely auxiliary tools on which we don't like to think about. Refined stopping criterion are crucial for saving computer time and, thus, having time to change the strategy. Few research is dedicated to this topic.
- 3 Multistart is a sensible strategy. Many other global strategies exist. The choice is difficult (and perhaps impossible) because no strong supporting theories exist.
- 4 Algencan has global-minimization properties when the subproblems are solved with global-minimization strategies. Global simple-constrained (perhaps unconstrained or box-) optimization is obviously easier than global general-constrained optimization.

# Algencon and Infeasibility (Thesis of L. Prudente 2011)

You are running Algencon and the sequence seems to be condemned to converge to an infeasible point. What should you do?

Alternatives:

- 1 You continue the execution until the maximum number of iterations is exhausted, because perhaps something better is going to happen.
- 2 You stop and try another initial point.

For deciding this question we need better theoretical knowledge about the behavior of Algencon in Infeasible cases.

## Algencan and Infeasibility

Assume that a subsequence generated by Algencan (solving the subproblems up to stationarity with  $\varepsilon_k \rightarrow 0$ ) converges to the infeasible point  $x^*$ .

Consider the Auxiliary Problem:

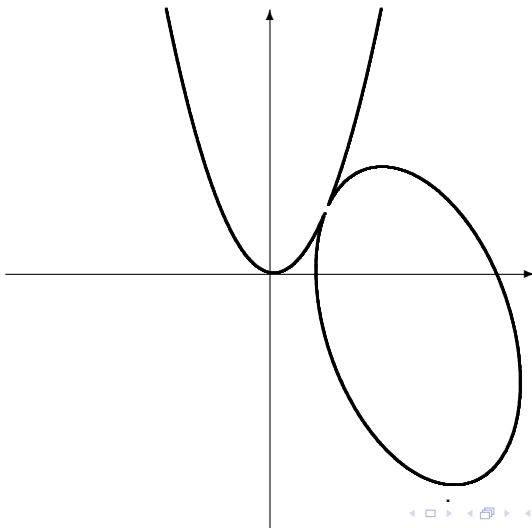
$$\text{Minimize } f(x) \text{ s. t. } h(x) = h(x^*), g(x) \leq g(x^*)_+$$

Then, for all  $\varepsilon > 0$ , there exists  $k$  such that the AKKT stopping criterion holds at  $x^k$  with respect to the Auxiliary Problem.

Algencan tends to find minimizers subject to the levels of feasibility of its limit points.

# Minimization with empty feasible region

Minimize  $4x_1^2 + 2x_1x_2 + 2x_2^2 - 22x_1 - 2x_2$  s. t.  $(x_2 - x_1^2)^2 + 1 = 0$ .



## More on Infeasibility

Suppose that one runs Algencan setting, for each subproblem, a convergence tolerance  $\varepsilon_k$  that **does not tend to zero**. For example,  $\varepsilon_k = 10$  for all  $k$ .

Then, the property that every limit point is a stationary point of infeasibility is preserved. (But the minimization property of  $f$  on infeasibility levels does not.)

**Practical consequence:** If your Algencan sequence is condemned to infeasibility, you can get the Infeasible-Conclusion spending a moderate amount of time on each subproblem.

## When the perfect solution is wrong

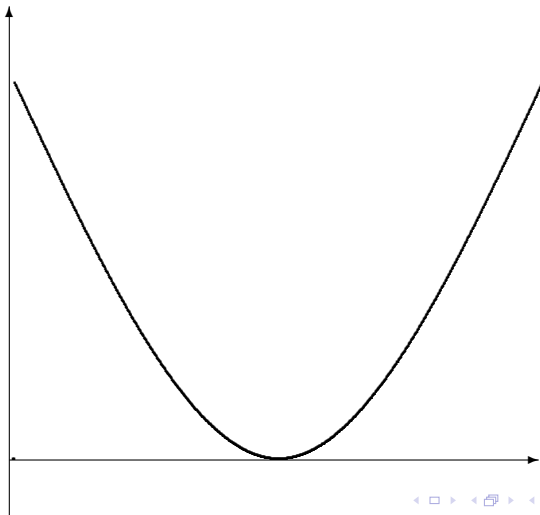
You run your Nonlinear Programming solver A with a strict tolerance for infeasibility, say,  $10^{-10}$ . Your solver converges smoothly to a AKKT feasible point up to that tolerance and an unexpectedly low value of the objective function.

You are very happy but your Engineer says that the solution is completely wrong. (This is the good case; in the bad case your Engineer believes that it is correct.)

Reason: Unexpected Ill-Conditioning of constraints. The tolerance  $10^{-10}$  is not enough to guarantee that the point is close to the constraint set. The “shape” of the solution is completely wrong and the rocket will fall over your head.

$$\text{Minimize } \sum_{i=1}^n x_i \text{ s. t. } x_i = \frac{x_{i-1} + x_{i+1}}{2}, i = 1, n, x_0 = x_{n+1} = 1.$$

Approximate solution ( $n = 2500$ ) with Norm of Infeasibility  $\approx 10^{-6}$ :





## Final Remarks: Beyond KKT

- 1 Global Optimization and KKT-like Optimization are parts of the same problem. Software developers should care with both problems as being only one.
- 2 Modelling is part of our problem. We are requested to find  $h(x) = 0$  (and not  $\|h(x)\| \leq 10^{-8}$ ) even when this is impossible.
- 3 From a theoretical point of view with potentially practical applications a challenging research program is the investigation of convergence to non-KKT points.