2008 International Symposium on Nonlinear Theory and its Applications
NOLTA'08, Budapest, Hungary, September 7-10, 2008

Can exact computation help optimization?

Daniel Lazard

UPMC Univ Paris 06, LIP6, F-75005, Paris, France
INRIA Paris-Rocquencourt, SALSA project team, F-78153 Le Chesnay, France
CNRS, LIP6, F-75005, Paris, France
Email: Daniel.Lazard@lip6.fr

Abstract—In this talk we present several problems of optimization coming from computational geometry, statistics and wireless networks which either have been solved by exact methods or whose numerical resolution have been validated by exact computation.

1. Introduction

It is almost immediate that many problems of optimization may theoretically solved through polynomial system solving. For example, the minimum of a multivariate polynomial function may be obtained as a common zero of the partial derivatives, if it is attained for finite values of the variables. In practice, such an approach was hopeless because of the lack of efficient tools for solving polynomial systems.

The availability of an efficient software for solving polynomial systems (Function `RootFinding [Isolate]` of Maple 11) changes the situation. It should be noted that the applications that are described below are anterior to Maple 11, but have used the same software as this function `Isolate`, namely FGB and RS[1].

The fact that polynomial system solving has a high complexity (at least exponential in the number of variables) has the consequence that it may certainly not be useful when the number of variables is high or when one knows that the cost function has only one local minimum. On the other hand we know of several optimization problems coming from applications which have been solved in an acceptable computational cost and which seem difficult for purely numerical methods.

The aim of this talk is to present these problems and to classify them by the kind of the optimization problem which they involve.

In this paper, an *optimization problem* consists in a cost function P to minimize and several sets of constraints, possibly empty: a set of equational constraints $\{f_1 = 0, \ldots, f_k = 0\}$, a set of strict inequalities $\{g_1 > 0, \ldots, g_l > 0\}$ and a set of large inequalities $\{h_1 \geq 0, \ldots, h_m \geq 0\}$. We suppose that the f_i, the g_i and the h_i are multivariate polynomials. Except in Section 2, we suppose also that the cost function is polynomial.

For each large inequality $h_i \geq 0$ the problem may be split in two sub problems where $h_i \geq 0$ is replaced either by the large inequality $h_i > 0$ or by the equation $h_i = 0$. Unless if the number m of large inequalities is very low, this introduces a number of cases which makes impossible the computation. Therefore, with the present state of the art, exact computation is of no help for problems with several large inequalities.

2. Reducing the equational constraints

As the f_i are polynomials, the equational constraints of an optimization problem define an algebraic variety, on which the minimum of the cost function is to be found. Without information on the structure of this variety (especially on its singularities) the classical numerical methods may fail to find the minimum of the cost function.

A classical theorem of algebraic asserts that an irreducible variety is birationally equivalent to a hypersurface. This means that the coordinates of almost all points of the variety are rational functions of the zeros of a single univariate polynomial. The condition that the variety is irreducible is not very strong, as in most applications the variety of the constraints is irreducible or easily decomposed in irreducible components.

Gröbner bases allow to compute such a rational representation of the points of a variety, and this reduces an optimization problem with equational constraints to another problem with only one constraint. Moreover, if algebraic functions are allowed for the cost function, one may "solve" this remaining constraint with respect to one of its variable and substitute the solution in the cost function to obtain an unconstrained optimization problem which may be solved by the usual numerical methods.

This approach has been used in [2] to design filter banks for 2D image compression. A witness of the efficiency of this approach is that some input systems of equations have a size higher than 60 Mb.

3. Minimum of a polynomial function

Finding the minimum of a multivariate function is

[1] Available at http://fgbrs.lip6.fr/salsa/Software/

the basic problem of global optimization. For a polynomial function, the problem is clearly very difficult, as the number of local extremums may be exponential in the number of variables. Another difficulty comes from the fact that the minimum is not necessary reached, if the function tends to its minimum when some variables tend to infinity.

In the most frequent case where the minimum is reached, it is reached at a critical point of the cost function, i.e. at a zero of its gradient. Thus finding the minimum may be done through polynomial system solving: it suffices to compute the common zeros of the partial derivatives of the cost function and to look at the values of the cost function at these zeros.

For being practicable, this approach needs an efficient polynomial system solver. Even with such a solver, it is not evident that optimizing through polynomial system solving may be competitive with the usual numerical methods. In fact, the complexity of polynomial system solving is at least polynomial in the number of complex roots of the system, which is itself exponential in the number of variables. On the other hand, the complexity of the numerical methods depend strongly on the number of local minimums of the cost functions, which may be of the same order as the number of complex zeros of the gradient of the cost function.

Thus a theoretical comparison seems very difficult between numerical and algebraic methods to find the minimum of a polynomial function. The best which can be done is to compare them on difficult problems coming from applications. In fact, there is difficult to guess the time needed to solve a particular polynomial system by extrapolation from the time needed by random examples. In particular, it seems that random examples are usually much more difficult than the explicit systems coming from applications. In this paper we try to show, by some challenge examples, that algebraic methods may effectively be competitive.

A problem which is strongly related to the global minimum of a function is the proof that a polynomial function is always positive (its minimum is positive) or never negative. As for every computational proof, if floating points are used, the computation has to be completed by a certification of the result.

I have encountered this kind of problems when studying the Voronoi diagram of lines in the 3D space ([1]): Given three lines in the 3D space, their *trisector* is the curve of the points which are at the same distance of the three lines. If the lines do not intersect nor are parallel to the same plane, the trisector is a quartic which is generically smooth and irreducible, but may be decomposed into a cubic and a line for some configurations. These configurations are characterized by the vanishing of a polynomial P of degree four in five variables, which has more than 300 monomials. To understand what are these special configurations, the first thing to decide is if they are a hypersurface (dimension 4) in the space of the triplets of lines or if they belong to a variety of lower dimension (in fact dimension 3). This means that all the real points of the hypersurface defined by P are singular and thus that P is either never negative or never positive.

This is effectively the case and we could prove it using the software RAGLIB of Mohab Safey-el-Din[2]. This software were primarily designed to find at least a point in every connected component of a semi-algebraic set. It has been recently extended to find the global minimum of any multivariate polynomial (even if this minimum is not reached for a finite value of the variables). It is not the place to describe here how it works.

4. Proving global convergence of a numerical optimization process

We have seen in the preceding section that the algebraic approach may be competitive when the cost function has many local minimums. On the other hand, it is clear that the classical numerical methods are much faster if there is no other local minimum than the global one, but they are usually unable to verify it.

Thus when the cost function depends on parameters, it useful to determine once for all the values of the parameters for which the global minimum is the only local minimum, i.e. the values of the parameters for which a numerical process converges globally.

This may be done by counting, as a function of the parameters, the number or zeros of the gradient where the principal minors of the Hessian matrix are all non negative. This may be done with the algorithm of [6].

Such a problem occurring in data transmission has been solved in [3], where it has been shown that the following cost function of four complex variables has only one local minimum if and only if the positive parameter α is different from 1 (the symbol * denotes complex conjugate).

$$\begin{aligned} J &= |g_{11}|^4 + |g_{12}|^4 + 4|g_{11}|^2|g_{12}|^2 + |g_{21}|^4 + |g_{22}|^4 \\ &\quad +4|g_{21}|^2|g_{22}|^2 - 2|g_{11}|^2 - 2|g_{12}|^2 - 2|g_{21}|^2 \\ &\quad -2|g_{22}|^2 + \alpha(|g_{11}|^2|g_{21}|^2 + |g_{12}|^2|g_{22}|^2 \\ &\quad +g_{11}g_{12}^*g_{21}^*g_{22} + g_{11}^*g_{12}g_{21}g_{22}^*) + 2. \end{aligned}$$

5. Identification of hidden state parameters

Another situation where an optimization problem may be solved advantageously by algebraic computation is the following. Consider a physical system with

[2] http://fgbrs.lip6.fr/salsa/Software/

k hidden state parameters from which $k+1$ measurable quantities depend rationally. Identifying the hidden parameters from the measurement amounts thus to solve an overdetermined system. Unfortunately it has usually no solution, because of the imprecision of the measures.

In fact, the dependence relations between the measurable quantities and the state parameters define a parametrization of some hypersurface in the measures space. Therefore the best way to identify the state parameters consists in finding the values of the parameters for the point of this hypersurface which is the closest to the measured point, i.e. to solve a problem of global optimization.

An example of such a situation, which has been solved in [4], is the following. Let us consider, in statistics, a probability law which is a mixture of two Gaussian laws. There are five unknown parameters which are the ratio of the mixture, the two mean values and the two square deviations. The measurable quantities are the six first moments, which are rational functions of the five parameters. After some reductions and changes of variables, these relations are equivalent to the system

$$\begin{aligned} W &= w \\ X &= -\frac{6u^3+2v^2-4vw-w^2}{3u} \\ Y &= -\frac{4vu^3+20u^3w+2v^3-5vw^2}{3u^2} \\ Z &= (144u^6+72u^3v^2-180u^3vw+4v^4 \\ &\quad -90u^3w^2-30v^3w+30v^2w^2+5vw^3)/9u^3 \\ u &> 0, \end{aligned}$$

where the state parameters are u, v, w and the measurable quantities are W, X, Y, Z. The relation between W, X, Y, Z, which is obtained by eliminating u, v, w, is a polynomial of degree 23 with 195 monomials and coefficients up to 14 digits.

To minimize the distance from a point (w, x, y, z) to the hypersurface defined by this parametrization, one has to compute the critical points of the distance function, i.e. the points (W, X, Y, Z) where the vector $(W-w, X-x, Y-y, Z-z)$ is orthogonal to the surface. This leads to a system with 176 complex solutions (see [5] for details), whose real solutions may be computed in a few minutes with a standard laptop. I do not know if the numerical methods for global optimization are able to solve a problem of such a size.

Structural stability of the solution. It may happens that there are several points of the hypersurface which are roughly at the same distance of the measured point. This may occurs if the measures are not precise enough. This is also the case if the measured point is close to a self crossing point of the hypersurface, i.e. a point corresponding to several values of (u, v, w). We say that such a point is a point of *structural instability* because increasing the precision does not allow to choose between the solutions.

The set of points of structural instability is clearly a subset of the singular locus of the hypersurface, but it may be much smaller: In the singular locus there may by real crossing points between two conjugate complex sheets or between sheets not satisfying the inequalities (here $u > 0$). In our example of the mixture of two Gaussian laws, it has been computed, using the algorithm of [6]: It consists in a single linear variety while the singular locus is the union of it with a variety of the same dimension and degree 150.

Thus the locus of structural instability is not only very small, but it has been fully characterized. This allows to know if the measured point is close to this locus before any computation. If it is the case, one may thus predict that there the computation will provide be several (usually two) admissible solutions.

6. Conclusion

All these examples show that exact computation may be helpful, in various situations, to solve optimization problems.

This help may consist in

- Simplifying the optimization problem.
- Certifying the numerical solution or detecting the regions where numerical methods are unstable.
- Solving the problem when the cost function is too irregular.

The exact computations are clearly not very fast, but they appear to be competitive in many case which are difficult for numerical methods, especially when global optimization or certification of the result are required.

References

[1] Hazel Everett, Daniel Lazard, Sylvain Lazard, and Mohab Safey El Din. The Voronoi diagram of three lines in $\mathbb{R}^3$. In *SoCG '07: Proceedings of the twenty-third annual symposium on Computational geometry*, pages 255–264, 6 2007.

[2] J.-C. Faugère, F. Moreau de Saint-Martin, and F. Rouillier. Design of regular nonseparable bidimensional wavelets using Gröbner basis techniques. *IEEE Transactions on Signal Processing*, 46(4):845–856, 1998.

[3] Nong Gu, Daniel Lazard, Fabrice Rouillier, and Yang Xiang. Using computer algebra to certify the global convergence of a numerical optimization

process. In *Mathematical Aspects of Computer and Information Sciences*, pages 100–112, 7 2006.

[4] Daniel Lazard. Injectivity of real rational mappings: the case of a mixture of two Gaussian laws. *Math. Comput. Simulation*, 67(1-2):67–84, 2004.

[5] Daniel Lazard. Computing with parameterized varieties. In *Algebraic geometry and geometric modeling*, Math. Vis., pages 53–69. Springer, Berlin, 2006.

[6] Daniel Lazard and Fabrice Rouillier. Solving parametric polynomial systems. *J. Symbolic Comput.*, 42(6):636–667, 2007.