Société de Calcul Mathématique SA
(Mathematical Modelling Company, Corp.)
Tools for decision help
since 1995

Coarse optimization in real-life problems

Though many sophisticated mathematical tools exist in order to solve academic optimization problems, they are generally of little help when it comes to solving real-life optimization problems. Moreover, the use of these tools may be misleading, and even dangerous, because it gives the false impression that a solution has been reached, whereas this solution is not the right one.

In real life, data are imprecise. There is no need for a tool which gives the result with 10 decimal places, after hours of computation, when the initial data have a precision of 20%.

Let us take an example. Often, an optimization problem is described the following way: Find the minimum (or the maximum) of some function $f(x_1,\ldots,x_n)$ under various constraints. The value of the minimum depends strongly on the aspect of the function f and on the constraints $g_1,\ldots,g_m$. Change a little bit any of the constraints, or modify the shape of the function f, and the value of the minimum may vary considerably.

The search for a minimum is always a delicate and lengthy operation, which may very well take hours on a computer, if the functions are numerous and complicated.

These two considerations explain very clearly why such tools are of very little value for real-life problems. Indeed, in a real-life problem:

– Many data are missing or are imprecise, so the constraints cannot be made precise;

– The shape of the function to be minimized is not clear. People do not know exactly what they want, or, more exactly, they want several things at the same time. For instance, do they want to increase short-term profits, or do they want to take into account long-term considerations ? The shape of the functions to be optimized will be different in both cases.

What people want for real life problems

In real life problems, people are aware that data are insufficient and imprecise and they are generally aware of the fact that their problem can be put in various forms. So what they want is in fact a **Quick Acceptable Solution** (QAS), not an optimum.

- Acceptable means that is satisfies their constraints or, more precisely, rough versions of their constraints. It does not make much sense to write the constraints precisely, because the data are imprecise;

- Quick means that the computer should give a solution within minutes, not hours.

When a Quick Acceptable Solution has been found, you can either:

- be satisfied with it;

- or use it as a starting point for a QAS of a refined problem.

Let's take an example, which comes from EDF (French Electricity). Everyday, they have to decide which plants will be used to produce electricity. There are about 200 plants, with various capacities, and extremely complex constraints: not all productions are possible in a given range by each plant, maintenance should be taken into account, and so on. The demand in electricity, depending on the weather conditions, should be satisfied, with minimal cost. So this is an extremely complicated optimization problem, which the existing software poorly solves within hours. A combinatorial approach is infeasible, since there are 2^{200} possibilities to examine.

What EDF wants is in fact a coarse solution, a Quick Acceptable Solution, of the following type: is there a configuration of the plants which gives a cost smaller than a given C (for instance last year's cost) ? If yes, find it quickly. We might later refine it, by diminishing C or adding more constraints. If not, we increase C, and we ask for a new QAS.

In the search of a QAS, there is no optimization any more: everything is treated as a constraint. We ask the computer to find the first solution satisfying our constraints, the first and not the best. So this is usually very quick.

Guidelines for the search of a QAS

The constraints must be simplified, in order to take into account the imprecision. For instance, it does not make sense to look for $x_1,\ldots,x_n$ satisfying $f(x_1,\ldots,x_n)>0$ with a very complicated and precise function f, which is not known in reality. Just take f to be piecewise linear, with coarse (simple) values. The number of constraints must be reduced to the essential ones, at least at the beginning.

The optimization request disappears, as we said earlier, and is itself treated as a coarse constraint.

Guidelines for the program

The attitude towards such a program is new in spirit, because mathematicians usually like precise answers to precise questions, whereas here we are trying to bring a coarse and quick answer to a coarse question. In all cases, a real-life problem, even under coarse form, is always difficult. If one wants a QAS, one cannot afford to be weak in any of the three aspects : mathematical modeling, numerical analysis, computer implementation.

In order to learn more, please see our research program "Robust Mathematical Modelling" : http://www.scmsa.eu/robust.htm