2005 ACM Symposium on Applied Computing

Editorial: Track Reliable Computations and their Applications

Martine Ceberio and Vladik Kreinovich
Computer Science Deparment
University of Texas at El Paso
500 West University Avenue
El Paso, Texas 79968-0518, USA
+1 (915) 747 69 {50,51}
{mceberio, vladik}@cs.utep.edu

Michel Rueher
ESSI
École Supérieure en Sciences Informatiques
Route des colles BP 145
06903 Sophia Antipolis, France
+33 (0)4 92 96 51 54
rueher@essi.fr

1 Objectives of the track

Many numerical computations, be they solutions to systems of differential equations or optimization problems coming from applied areas like protein folding, do not provide us with guaranteed computation results. In many situations, we have numerical solutions, we may even have a theorem guaranteeing that eventually, this numerical solution tends to the actual precise one, but the algorithm itself does not provide us with guaranteed bounds on the difference between the numerical approximate solution and the desired actual one.

Therefore, in some practical situations, numerical solutions are much farther from the actual (unknown) precise solutions than the users assume. As a result, we often end up with inefficient local maxima for practical optimization problems like chemical engineering - or even with a mission failure if we are planning, *e.g.*, a spaceship trajectory.

For some such algorithms, researchers have found guaranteed bounds, but producing a guaranteed bound for each algorithm requires a lot of work.

It is therefore desirable to develop a methodology that would provide algorithms with automatic result verification, i.e., with automatically generated upper bound on the difference between the actual and the numerical solution. In other words, we need computation techniques that produce reliable (guaranteed) results.

This problem was recognized already in the late 1950s when Lockheed wanted to develop algorithmic techniques guaranteeing trajectories of spaceflights. These techniques, largely developed by Ramon E. Moore, were later applied to other practical problems where deviations from the target are of critical importance. The main idea behind these techniques is that at any intermediate step of the computations, instead of the exact number, we keep an interval of possible values. For inputs (that usually come from measurements), we have an interval because measurements are never 100% accurate; if the manufacturer of the measuring instrument guarantees that the measurement error is Δ or smaller, then the measuring result $\widetilde{x}$ means that the actual value is in the interval $[\widetilde{x} - \Delta, \widetilde{x} + \Delta]$. At each elementary computational step, we apply interval arithmetic to the corresponding intervals and produce the interval for the result; *e.g.*, $[a, b] + [c, d]$ leads to $[a + c, b + d]$.

Of course, this "straightforward interval computation", that does not take dependence between intermediate results into consideration, does not always lead to efficient estimates; however, in the last 40+ years, efficient interval computations methods have been developed based on this original idea. There are a lot of interesting applications of interval computations, there is a lot of potential, but there are still numerous open problems, situations where new techniques are needed.

One such technique that has also been used to provide guaranteed bounds is the technique of constraint propagation. This technique originated in logical AI problems, and it has been lately successfully applied to numerical problems, often in conjunction with interval methods. For example, one of the latest textbooks on interval computations, by Jaulin et al., contains robot-related practical examples of combining these two techniques. This combination has started, it is the object of interest by many researchers, it has already led to interesting and efficient packages like Numerica, but there is still a lot of room for potential improvement.

Our track, with an emphasis on such a combination, brings together not only algorithm developers but also practitioners whose practical needs will help guide researchers in the proper directions.

2 Contributed papers

We received 12 article submissions, from the China, France, Germany, Italy, Singapore, and the USA. These papers were reviewed by Lucas Bordeaux (Universita Sapienza, Rome, Italy), David Daney (Institut National de Recherche en Informatique et en Automatique – INRIA, Sophia-Antipolis, France), Xavier Fischer (École Supérieure des Technologies Industrielles Avancées – ESTIA, Bidart, France), Laurent Granvilliers (Laboratoire d'Informatique de Nantes – LINA, France), Chenyi Hu (University of Central Arkansas – UCA, Conway, USA), Christophe Jermann (LINA, France), Randy Keller (University of Texas at El Paso – UTEP, USA), Yahia Lebbah (INRIA, Sophia-Antipolis, France), Olivier Lhomme (Ilog, Nice, France), François Modave (UTEP, USA), Brice Pajot (LINA, France), Gilles Trombettoni (INRIA Sophia-Antipolis, France), Martine Ceberio, Vladik Kreinovich, and Michel Rueher.

Over the 12 submitted papers, the following five papers (among which one was an invited paper – by Ramon E. Moore) were accepted for publications in the proceedings:

1. *Order Relations and Rigor in Computing*, by Ramon E. Moore.

2. *On Considering and Interval Constraint Solving Algorithm as a Free-Steering Nonlinear Gauss-Seidel Procedure*, by Frédéric Goualard.

3. *Box-Set Consistency for Interval-Based Constraint Problems*, by Gilles Chabert, Bertrand Neveu and Gilles Trombettoni.

4. *Enhancing Network Intrusion Detection Systems with Interval Methods*, by Qiang Duan, Chenyi Hu, and Han-Chieh Wei.

5. *Computing the Cube on an Interval Matrix is NP-Hard*, by Olga Kosheleva, Vladik Kreinovich, Günter Mayer, and Hung T. Nguyen.

3 Acknowledgments

We want to thank all authors, and all reviewers of our RCA track at SAC 2005. We are also grateful to the Symposium General Program Chairs, with the help and efforts of whom the organization of this track was made possible.