Optim Eng (2010) 11: 1–3
DOI 10.1007/s11081-010-9105-3

Editorial

Joaquim R.R.A. Martins · Ekaterina A. Kostina

Received: 2 February 2010 / Accepted: 3 February 2010 / Published online: 18 February 2010

For five days in November 2006, a group of applied mathematicians and engineers gathered at the Banff International Research Station (BIRS) to participate in the "Optimization and Engineering Applications" workshop. The goal of the workshop was to foster communication between optimizers who focus on algorithms and optimizers who focus on engineering applications.

There is a wide range of professionals in optimization: from pure mathematicians solving calculus of variations problems analytically, to applied mathematicians developing sophisticated algorithms that exploit the structure of specific problems, all the way to engineers that use commercial optimization software coupled to commercial PDE solvers. There are significant gaps within this range that are due to different technical languages, goals, and research interests. The aim of this workshop was to bridge these gaps by encouraging researchers to translate into each others' languages, discuss their different perspectives, and find common ground.

Consistently with this goal, one of the three panel discussions that took place at the workshop was entitled "What do engineers need? What do optimizers need to know about engineering?". One of the discussions focused on how engineers need to be more exposed to optimization from as early as the undergraduate level. Problem formulation by the engineer is particularly important because a solution to an optimization problem that was badly formulated in the first place is obviously far from optimal in practice. Problem formulation must rely on a combination of engineering

J.R.R.A. Martins (✉)
Department of Aerospace Engineering, University of Michigan, Ann Arbor, USA
e-mail: jrram@umich.edu

E.A. Kostina
Faculty of Mathematics and Computer Science, University of Marburg, Marburg, Germany
e-mail: kostina@mathematik.uni-marburg.de

intuition and knowledge of optimization theory. Optimizers, on the other hand, need to solve practical problems more often, to make sure that their algorithms are useful. Inevitably, their algorithms are used beyond their original intentions, and therefore, it is important to make the software versatile by taking those unintended uses into consideration. The overall conclusion was that rather than discussing theory versus practice, we should recognize theory *and* practice.

A second panel discussion asked two other questions: "Where do we need breakthroughs? What will be the next breakthrough?" The short answer to the second question was unanimous: "If I knew this, I would be working on it". The reality is that we do not know until the breakthrough happens, and history confirms this trend. There were, however, a few ideas on where breakthroughs are needed. One of the most discussed was the need to effectively utilize multiple core and parallel computing resources in optimization.

The final panel discussion was "Optimization Software", which focused on available software, as well as on identifying what new software is needed. Someone pointed out that many software users still use "lousy old optimization software", and that the burden is on the developers of optimization software to communicate to practitioners that better software is available. The reality is that sometimes, ease of use is more important to users than the quality of the algorithm. Maturity of the particular optimization field plays a role here. While "an idiot can take a linear programming package and get the right solution"—as someone said during this lively discussion—nonlinear optimization software cannot be used as a black box. Good documentation that includes simple examples is particularly important if research codes are to have an impact in the real world.

It would have been criminal to spend our whole time indoors with the beautiful Banff mountains beckoning. So yes, there was an afternoon of skiing or hiking, depending on the order of magnitude of one's preferred speed.

In this special issue, Rumpfkeil and Zingg present a method for the sensitivity analysis of unsteady flows using an adjoint method. The effectiveness of this method is demonstrated by solving two problems: the drag minimization of a rotating cylinder, and the remote inverse design of a multi-element airfoil.

Another adjoint-based method for sensitivity analysis is presented by Kennedy and Hansen, who propose a hybrid approach where both adjoint and direct sensitivity analysis are performed to reduce computational cost. This approach is motivated by the problem of determining optimal temperature profiles for composite cure cycles, for which results are presented.

Ding et al. cast a sensor network localization problem as a nearest-Euclidean-distance matrix model and formulate semidefinite programming relaxations of this model. Two methods for solving the problem are proposed, and numerical tests are performed to evaluate the new methods against each other and the current state of the art.

Another paper on aerodynamic shape optimization using adjoint sensitivity analysis is presented by Nadarajah and Tatossian. Their work focuses on using a nonlinear frequency domain method for the multiobjective optimization of a helicopter blade in three-dimensional viscous flow, as well as an aircraft wing.

Jarre et al. write about a new approach to correct the timestamps in computer network experiments that is based on solving a least squares problem. The authors compare this approach to a previously proposed one by running a simulation.

Zhang and Qian develop a new immune algorithm for solving time-varying, single-objective optimization problems. The algorithm is compared against five other evolutionary algorithms for three optimization problems, including a greenhouse control problem.

Luss and d'Aspermont study the application of sparse principal component analysis to clustering and feature selection problems. The resulting algorithm is used to solve classic clustering and feature selection problems that arise in biology.

This issue closes with a paper by Tedford and Martins, who aim to benchmark multidisciplinary design optimization (MDO) algorithms. MDO problems are prevalent in engineering, and a number of algorithms to solve such problems have been developed by engineering researchers. The authors present a framework that implements the various algorithms and use it to benchmark the algorithms on a variety of problems. They also devise a scalable problem to evaluate how each algorithm performs when different characteristics of the problem are varied.

The editors would like to thank the authors and referees for their contribution to this special issue.

Guest Editors