OPTIMA 99

Mathematical Optimization Society Newsletter

Note from the Editors

Dear MOS members,

Listening to what the leaders in our field have to say is not only very informative but also a most enjoyable experience. From the feedback after our interview with Jack Edmonds in *Optima* 97 in spring 2015, we concluded that many readers would love to see more such contributions from time to time. We are extremely happy to be able to follow up with another interview with another giant in optimization: Roger Fletcher was kind enough to answer questions posed by Sven Leyffer. Their conversation is the main article of this issue, along with a great list of links to material related to Fletcher's work carefully compiled by our interviewer. The discussion columns contributed by Philippe Toint and Frank E Curtis show how much Roger Fletcher is appreciated both as a scientist and as a person.

We furthermore commemorate two outstanding scholars our community lost in 2015. You will find obituaries for one of the founders of nonlinear optimization, Mike Powell, written by Coralia Cartis, Andreas Griewank, Philippe Toint, and Ya-xiang Yuan (that first appeared in *Optimization Methods and Software*, see remarks below) as well as for Che-Lin Su contributed by Jong-Shi Pang.

Finally, we hope that you have not yet finalized your travel plans for 2016. We have announcements of several very interesting events to come!

Merry Christmas and a happy new year!

Sam Burer, Co-Editor
Volker Kaibel, Editor
Jeff Linderoth, Co-Editor

Contents of Issue 99 / December 2015

It's to Solve Problems

An Interview with Roger Fletcher

[Sven Leyffer] *I'm sitting here with Roger Fletcher. Thank you very much for letting me interview you. Tell us a little bit about your career path – So, where did you go to university and where did you get your Ph.D.?*

I went to Cambridge University where I studied natural sciences, ending up doing theoretical physics in my final year. I was very fortunate at that time. In my final year we had the opportunity to do some computing and I was able to analyze some of my final year practical results with the least squares codes on the EDSAC computer [1], which I thought was just fantastic.

Which computer was that?

EDSAC – it's a famous early computer designed and built at Cambridge. So, that motivated me; I could see computing would be the thing, so I started looking at jobs and I got an interview with IBM, where I wasn't impressed with them and they weren't with me. But somebody from Leeds University called Sandy Douglas [2] who had just set up a computing lab, one of the first in the country, with a Pegasus computer, of which there were only three – a Ferranti Pegasus [3]. They were state of the art, and they were remarkable machines, really. He offered me a Ph.D. position, subject to getting a grant and everything, which I was pleased to accept. That was probably the happiest period of my life, doing the Ph.D. at Leeds – there were a lot of clever people there!

So who else was at Leeds?

My supervisor was Colin Reeves; he was a theoretical physicist doing molecular structure calculations, which was what I did. There was a guy (you wouldn't know the people's names) but there was one guy in bus scheduling; made a big thing out of bus scheduling. He's still one of the people who supplies these things to bus companies even now.

So was this mixed-integer programming?

No, no; this would be transportation problems, the sort of things you could do in those days. Nobody did mixed integers. Anyway, we had parameters in our models for molecular structure calculations. It was a very interesting Ph.D. problem. It had all sorts of things. It had linear transformations that preserved symmetry and things like that, done automatically to eliminate a lot of the calculations you needed to do. It also had automatic differentiation to derive integral formulae. Also generalized eigenvalue problems, for which I invented a method which happened to be known about. Using Jacobi methods, you get the eigenvalues for the B matrix and then you use that to transform the A matrix so that you get an ordinary eigenvalue problem, and you solve that with Jacobi methods. My idea, but it was obviously known about.

Oh, certainly.

Still it wasn't known about by many people, I think I'd have to say.

When was your Ph.D.?

It was '60 to '63.

OK, there weren't many books on linear algebra at the time.

No. In our model we had parameters; it was minimizing energy, so we had to choose the parameters to minimize the energy. And I read in a couple of books (Householder's book [4], I think it was); another book, Hildebrand [5] perhaps. They seemed to suggest that steepest descent was a very good method. So I tried it, and it generated reams and reams of paper, punched paper tape output as the iterations progressed, and didn't make a lot of progress. So after that I developed a great suspicion of what people were writing in books.

So then you built your own method?

Well, it was around that time that my supervisor got a copy of Bill Davidon's Argonne National Laboratory report [7], and he said, "Try that for your problem." And I realized that this was so many times superior to steepest descent that I was writing it up as a paper for the Computer Journal [16], and it turned out that somebody called Mike Powell [8] had been invited to give a seminar on his method for minimizing without derivatives. And a few days before the seminar he said he had just come across this paper by a guy called Bill Davidon [6] and could he talk about that instead? So when he got to Leeds, he found that we already knew about it, and had codes for it. Mike immediately also saw how important it was and also saw how it worked, so we pooled our resources and that was the basis for the famous DFP thing.

So when was that?

In 1963, which was my first publication. Also, Colin Reeves at this time was giving lectures on linear algebra and was lecturing on the conjugate gradient method. Again, not much was known about that at that time. Colin Reeves suggested to me that if you had some sort of search, you could use it to solve nonquadratic problems. So that became nonlinear CG, and I did the computations and wrote the paper [17] but his was the idea. So I had two huge good ideas given to me by people, so I got this undeserved reputation for being intelligent.

I'm sure it's not undeserved.

Well, I didn't realize how important it was at the time; you don't. For me, it's just okay, it works well, so that's good.

And, so, right from the start you were doing computation.

Absolutely.

And you think that's important to you in your research.

Absolutely. Like you were talking about today. I think.. that's where you should start if you are interested in problems. You should start from problems small or big, and do computations to understand how much is all behind it. So don't believe what you read in books; try something yourself, and be prepared to adopt a contrary view if you don't believe what people are telling you. But also be prepared to change your mind if somebody gives you a better argument. I'm always surprised that in politics it seems to be such a terrible thing for a politician to change his mind, it seems to damn him forever. But in our work I think you've got to, and you've got to be critical of what you've done, I think. With your own work; when you did it, at the time you thought was good. You've got to reexamine it in the light of what other inputs you get from people and from conferences and decide it wasn't such a good idea after all and pursue something else.

So what do you regard as your most sort of influential papers? Your career spans almost five decades, and I've sort of got a short list.

Well, BFGS, one has to say because it was my own idea. There were four papers published independently in 1970 [18, 19, 20, 21]. But I still like mine as the best way to describe BFGS [18].

So this is not one where you changed your mind.

I still think there is a better formula than BFGS which I have mentioned in a couple of papers, which I called the ultra-BFGS formula, but it never attracted any interest. It only gets you a 5 or 10% performance gain, so it's not worth making a big deal about it. One has to say that just because most people regard BFGS as the state of the art formula anyway, people think it's still important.

So I looked up your papers and saw that you actually did some early work on semi-definite optimization.

I did indeed.

And that's sort of interesting because it was a time when it wasn't fashionable to do semi-definite programming.

Indeed. In those days we used to have three-month M.Sc. projects, and I had two or three of them, I can't remember whether it was 2 or 3. And I got the student to solve these educational testing problems, which are SDPs, which have matrix constraints, matrix variables, and to solve them by BFGS or SQP or something, It always converged slowly, and I'd just give them to another guy to do. I just though they weren't very good students. But it was only after the third one that I realized that it was something to do with nondifferentiability of eigenvalue constraints, and then you suddenly start to realize what's going on. I wrote in my book somewhere that convex analysis was never very useful for computation. Then I changed my mind when I needed to use it for things like semidefinite matrices.

So when you solved the SDPs, you didn't use interior points methods because they were not yet around.

No, no, I think we factorized, we got the partial factors of the matrices, partial LDL^T factors. One thing you have to know is the rank of the matrix at the solution, because if all the eigenvalues are distinct there is no nonsmoothness. You only got nonsmoothness when the eigenvalues coalesce. You have to know the rank, something to do with the matrices being singular, you have to know the rank of these matrices. And we had to guess that, and then you could do things with partial factorizations rather than full factorizations. I think that's essentially how it went – which people are still doing, I believe.

I think people still do that now, sort of low-rank approximation. In fact, I was talking to Hongbo Dong – I've got one of Hongbo Dong's students (I think he was one of Sam Burer's [22] students) – so he was telling me that's a very good approach actually for solving large-scale problems. It's not an interior point approach, but it's still around.

Going through these things chronologically, the first thing after BFGS and DFP and all that was exact augmented Lagrangians, where you had augmented Lagrangians, but the multipliers were functions of x, obtained by a least squares solution of Ax equals the gradient.. and that was new at the time. It's not a popular approach now but it attracted a lot of interest at the time. A lot of the things that I've done that I sort of felt – that were – interesting but never ultimately in the scheme of things became popular. They keep getting used from time to time. Mike Powell took it up with a student at a much later date.

Roger Fletcher and Sven Leyffer at on Meall Glas Munro in Scotland (Photo: Sven Leyffer's camera)

QP – I, in the early days of QP I was really the first person to do indefinite QP, I think.

So was this with inequality constraints?

Yes, yes, inequality constraints. And active-set methods. Now the active-set methods came from Rosen, a paper of Rosen's [23, 24], sort of projected gradient stuff, a long time ago. He didn't do it for QP; he did it for nonquadratic objective functions.. He didn't pursue it, and I sort of took it over. The indefinite part was I think my main contribution. Gill [25] and a student, Elizabeth Wong [26]. They were referring to that in Berlin [27], you know, They're using that stuff.

So it's still getting used.

So that still gets used. Symmetric indefinite matrices and bi-CG [28] and all that came next. My original paper in 1976 [29] never got referenced very much. But people I talk to say, "Oh yes, we know about that." There was a reference to it at the recent Strathclyde meeting [31]; they made a reference to it.

It's certainly been used. It's a bit like that here - nobody quotes it...

Yeah, but they quote the bi-CGSTAB paper [30]. And again, bi-CG wasn't my idea – it came from Lanczos, but Lanczos only used it for eigenvalues [32]. And I pointed out you could use it for solving linear equations as well.

Then, there was composite nonsmooth optimization [33], which I did with Alistair Watson [13]. That sort of enables you to do L1, L-infinity, all sorts of polyhedral norms, all in a very nice structure, which before then it wasn't known or talked about at all; nowadays it is. You wouldn't say I invented it or anything, but I was there in the early early days of it.

I think that's one of the nicest parts of your book, actually. I enjoyed reading it as an undergraduate in nondifferentiable optimization; it's very structured. So, did that then lead to the penalty functions, exact penalty functions?

No. exact penalty functions, I think they came before that. L1 penalty functions go back yonks [34] to Andy Conn [35], his supervisor, Pietrzykowski, was a big L1 man, [36].

So then SQP came – I mean SQP dates back to Wilson in 1960. And Beale [37] had a paper; his is the best paper [38] – I mean, he's got everything in it; he's got trust regions and everything, or what was essentially trust regions, L-infinity trust regions, early infinity trust regions, shifting the bounds about and all that.

Is that the same Beale who wrote on integer programming?

Yes, Martin Beale.

OK, that's interesting.

It's a great paper. Nobody refers to it or reads it anymore, but that was really the best early paper on SQP: Wilson's was a Ph.D. thesis; he never took it anywhere. And I had a student working on it then, and the student never published anything on it, went in industry and never wrote it up, so I never got a paper out of that. And then about the same time was when Mike Powell popularized it; he did SQP with a quasi-Newton update (BFGS sort of update), so that was the beginnings of SQP.

Expected conditioning came next. I have always been suspicious of condition numbers because you can have triangular matrices and if you solve them one way around, like with L, and then you solve them the other way around in U, the conditioning is totally different; one way is much more error-prone than the other, and yet the condition number doesn't tell you that. So I wrote a paper called Expected Conditioning [38], which was a sort of statistical stochastic approach to getting condition numbers which predicts all these things, but it never caught on. I just liked that paper; I was really pleased with that paper [38].

We need to be make sure we get all the references into the optimal papers, so that people start picking these things up.

And then LU updates. I like LU updates of the Fletcher-Matthew's method [39]. It was commonly thought at the time that updating LU was potentially unstable.

Those are sparse. It's a sparse update or is it dense?

No, it's dense. And we discovered one that was stable. And I use that a lot. It has advantages for certain types of problems. So it was nice to do something that people said couldn't be done.

And there were L1 penalties, not really composite NDO, but using it as a penalty function, and that was an alternative to SQP, instead of SQP, you minimize and sort of do SQP but it is not really SQP but it's SL1QP.

You minimize the L1 in the function ...

you put the constraints in the function ...

... the linearized constraints ...

... the linearized constraints stay in the function. You don't take them out and linearize them. The subproblems involve L1 constraints ... So that was new.

Did you write a solver to do that because that requires a specialized solver for minimizing NDOs ...

That's right. Good question. Don't know what I did.

I remember that Steve Wright [40] told me he implemented it, but he implemented it by introducing slack variables and then rewriting, so he was solving the L1QP subproblem, by introducing slacks and then writing it as an ordinary QP. He had very good results with it. I don't think that work was published ...

I think that is something that would still be a competitor with filter methods. And you can do trust regions with it – that's the nice thing. You can prove stronger results with that than you can when you just do a trust region on ordinary SQP.

And that's because you're always feasible, or what gives you the stronger results?

Because you're always feasible. You get guaranteed convergence to a minimum of the L1 function, which may be feasible or may be not ... it sort of bundles it all together.

You don't need to separate the feasibility restoration from the optimality phase.

Degeneracy – I did a lot of work – There are a lot of different methods for doing degeneracy. But eventually I settled on Wolfe's method, and that never caught on very much … except Margaret Wright used to tell me she thought it was a great idea and I should publish it; and I published it just recently [41, 42]. I eventually wrote it up. It's in our BQPD solver. And people tell me – people such as you – when it's used in MINLP [43] codes, they tell me it's a way to deal with degeneracy because you don't do it with perturbations which messes up your accuracy. Wolfe's method creates degeneracy but I think this is the right thing to do, because usually the fact that it's near degeneracy means that it's really true degeneracy but rounding errors have made it not degenerate.

So the method takes a near-degenerate vertex and makes it degenerate.

Exactly. Makes it degenerate and then resolves it with Wolfe's method. You see Wolfe's method was recursive, and people were frightened of recursions, but it's not necessary; it's really very simple. Mike Powell introduced me to it a long time ago, I've never discovered any of these things myself.

So when you implemented BQPD, it's implemented in Fortran 77 so that's doesn't have recursive functions, so you had to do something else … you couldn't implement recursive functions …

No, no. You don't want to do it with recursive functions. It's only a few lines of Fortran. You only need to recur one scalar. You have to have a stack when you do recursion, and you only have to put one thing on the stack.

SLP-EQP: To avoid doing SQP you do SLP instead, and that gets you the the active constraints and the multipliers. That's a nice result, I believe my own. It relies on the fact that you can discover the active constraints without solving QPs; you only need to solve LPs near the solution. Obviously if you solve LPs, you're not going to get quadratic convergence.. but you do get near enough to the solution and you do get the right active set and then you use EQP techniques to find the solution. And that was Fletcher and Sainz de la Maza (he'was a PhD student) [44]. It fitted in with the composite NDO framework very nicely as well. And that has caught on a lot. Nocedal uses it, and other references.

Nick Gould recently did an SQP-EQP method where he solves a convex QP because he can do active set on the convex QP and then he does an EQP, which is potentially nonconvex and regularized.

Another interesting idea is L implicit U factors [46] which is what's used in BQPD; and I really like that because as opposed to ordinary LU factors. Michele Benzi gave me lots of references.

That's Michele Benzi [45]?

Michele Benzi, yes, I went to Cerfacs, and he gave me lots of references to it in the past, none of which has caught on. Mine has added to it as another one that has not caught on. But I still like it; it is more efficient for doing Fletcher-Matthews updates when using LIU factors.

And what does an implicit U mean?

It means, assume you have a copy of A available, as you usually do in LP or QP, and you have a copy of the Jacobian. All you need is L. You only need the space to store L. And U is implicit. I gave that paper at Mike Powell's birthday celebration, some time ago now. I like it [47]

Filters [48] that was next

I think I've heard of that.

It made much tingle when I thought of it. I was just thinking why, why penalty functions didn't work. You often look at the numbers and think, why can't I take SQP steps? Why am I having to throw this stuff away when it's obviously working well? A simple idea there – nobody thought of it, and now quite a number of people take it up.

Barzilai-Borwein [49] – I wrote a paper … I was the referee; Yuhong Dai [50] wrote a paper about Barzilai-Borwein and I was the referee, and I said this paper was of no interest – something along those lines. And then he wrote back and said, "Have you seen the paper of Marcos Raydan [51], where he is solving problems with 10^6 variables with this method." The original paper solves a two-variable problem, you see. So that's a case where I changed my mind.

And in fact you co-authored papers with him.

We've written various papers on it. And since then, I've had this Ritz limited memory steepest descent method where you use Ritz values to get steps. It speeds up Barzilai-Borwein quite a bit. I quite like this paper.

Now somebody at this meeting said you had a counterexample for some form of BB?

Oh, yes, we do … Dai and Fletcher.

So what does the example show?

It shows that it cycles, or potentially could cycle.

OK?

This is for box-constrained QP. If you take Barzilai–Borwein steps and project them. Nobody knew whether it cycled or not or whether you had to have a line search, And we answered in the affirmative, or I should say, Yuhong Dai answered it, I checked the results. And then nonnegative QP, which I am quite excited about.

This is what you talked about at the conference.

Yes. I am quite excited about that.

So this is applications in image processing?

Image processing, yes. It can also potentially solve dual QPs, and that could be big. No active set method in anything; you get the correct active set very quickly on huge problems, in my limited experience. I mean that could be a big thing,

And it uses a nice transformation of the augmented Lagrangian.

Yeah, a new transformation, that's right, just transformation of variables . It came about because I was trying to work out why I couldn't get MINRES to work on the Rockafellar's augmented Lagrangian, and I was messing about doing things to see why it wasn't working and just came across it. One thing I would say is to look at numbers: they're trying to tell you something.

This goes back to saying write software and the role of software.

Yeah, be a good programmer. Now if you give an idea to a Ph.D. student and it doesn't work, you have no idea why he thinks it doesn't work.

True

Whether it was a poor idea or maybe it was a good idea but his program has a bug in it. Maybe he's not a very good Fortran programmer.

So, I know when I was in Dundee, one of the things you always told us was to look at examples, so this is the same idea.

Part of it, yeah.

And I also remember that when you wrote BQPD, it was written in single precision. It wasn't that long ago, so it wasn't that it was an issue with *storage. We had 32-bit architectures, maybe 64-bit architectures, so why were you writing BQPD in single precision?*

Well, Mike Powell once told me that if an algorithm is good, it should work in single precision... except you were allowed to use double precision in inner products, and I sort of believed that until I started to solve LP problems. Then I had to change my mind on that. I must be the only person who implemented something in half-precision.

OK ...

I used software to cut down my single precision version to half precision, and it works even more badly. You can hardly solve anything with it at all in the LP case. But this thing I didn't appreciate: I think architectures are optimized to double precision, or were at the time. So thinking you would get savings because one ought to multiply two short numbers together more quickly than you multiply two big numbers may not be the case because the of how architectures are devised. It was a sort of misapprehension on my part that I thought it might be a good idea. I don't believe it anymore; I wouldn't write in single precision now. In fact I would rather go the other way. Mike Saunders was talking about quad precision. It would be nice to see. He says that quad precision is implemented in gFortran, if only by software. Most people will not go for that ... because it does become a limitation.

If I'm expected to give some advice ...

Oh, yes, I was going to say, What advice would you give to someone starting a Ph.D. or early in their career path?

I would say write clear slides and clear papers ... as well as some of the other things I've talked about. But don't fill your papers with guff [53]; don't write convergence proofs that are 30 pages long. Make it so that I can understand it. What works for me is to keep the notation as simple as possible. I don't know, maybe it doesn't work for other people, I like to get some really crisp notation. If I am working through something, I think there is better notation that would serve, I would spend time changing the notation to make the rest of the project easier. Notation is very important to get it right, and these people who like to have lots of Hilbert spaces and Banach spaces and ...

Sometimes you can't get away from them.

Yeah. but you don't have to stuff them down people's throats. It's because I'm a physicist and not a mathematician, I think, although physicists don't always write very clear papers either. But you go to lectures and a guy has 20 lines on a slide filled with 10 equations each with about 100 terms in it, and he flashes through that slide in 10 seconds.

... and you're sitting at the back of the room with very bad eyesight

It's common sense – but it doesn't seem to get through to people.

So what are kind of the open problems you see in optimization? What are the cool thinks people are doing?

Well, that MINLP thing with PDE with differential equation constraints.

Of course, that's only because I just gave a talk on them.

Whatever the users want. It should be driven ultimately by what problems people want to solve. you sort of alluded to that in your talk today a bit about when you were saying test problems. And test problems are often very badly selected because they don't really model any situation that is of any practical relevance. And you might get a model for something that has a 5 point discretization formula and it might make much more sense to model it as a radial basis function or something like that, or whatever. I'd much rather get my problems from the people who have problems to solve, rather than taking them from a library of test problems.

Right. Absolutely.

And I think we ... everybody publishes solves with 500 test problems from CUTE [54]. I'd rather see half a dozen real industrial problems ... problems of industrial interest being solved. That's where we should go – talk to industry. It's hard to do that; it's hard to get industry to talk to us.

So the UK has got more emphasis now that they're trying to place on industry by including it in the research assessment exercise ... their impact and ... there was a study that was made of the impact on the gross domestic product of applied mathematics.

I think for us in numerical analysis – in optimization in particular – because that's the purpose of it: it's to optimize; it's not to produce pure mathematics... in my view, differing from other people's view. But it's to solve problems. And we just keep solving the same old problems – getting another 5 % here and another 5 % there: it's a bit sterile.

So how do you go and find somebody to work with in industry?

That's a good question.

OK.

I think you perhaps look around and you see something that you think might benefit from optimization, and you go and bug somebody or go and read some papers in that area and see if there is anything you think you could contribute. And then if you've got the enthusiasm and the personality, you go and try and sell yourself to them. Things I am not very good at.

Thank you very much for the interview.

[We are grateful to Sven Leyffer for posing the questions, and to Gail Pieper for transcribing the recording.]

Bibliography

[1] https://en.wikipedia.org/wiki/Electronic_Delay_Storage_Automatic_Calculator
[2] https://en.wikipedia.org/wiki/Sandy_Douglas
[3] https://en.wikipedia.org/wiki/Ferranti_Pegasus
[4] https://en.wikipedia.org/wiki/Alston_Scott_Householder
[5] https://en.wikipedia.org/wiki/Francis_B._Hildebrand
[6] https://de.wikipedia.org/wiki/William_Davidon
[7] http://epubs.siam.org/doi/abs/10.1137/0801001 (reprint) originally published as Argonne National Laboratory Research and Development report 5990, May 1959
[8] https://en.wikipedia.org/wiki/Michael_J._D._Powell
[9] http://epubs.siam.org/doi/abs/10.1137/0902021
[10] http://www.jstor.org/stable/2098960?seq=1#page_scan_tab_contents
[11] http://link.springer.com/chapter/10.1007/BFb0080116
[12] http://nvlpubs.nist.gov/nistpubs/jres/045/4/V45.N04.A01.pdf
[13] http://www.maths.dundee.ac.uk/~gawatson/
[14] http://imajna.oxfordjournals.org/content/5/3/247.abstract
[15] http://link.springer.com/article/10.1007/BF02591933
[16] http://comjnl.oxfordjournals.org/
[17] http://comjnl.oxfordjournals.org/content/7/2/149.short
[18] http://comjnl.oxfordjournals.org/content/13/3/317.short
[19] http://www.ams.org/journals/mcom/1970-24-109/S0025-5718-1970-0258249-6/
[20] http://www.ams.org/journals/mcom/1970-24-111/S0025-5718-1970-0274029-X/home.html
[21] http://imamat.oxfordjournals.org/content/6/1/76.abstract
[22] http://tippie.uiowa.edu/sburer
[23] http://epubs.siam.org/doi/pdf/10.1137/0108011

[24] http://epubs.siam.org/doi/pdf/10.1137/0109044
[25] https://math.ucsd.edu/people/faculty/Philip-Gill/
[26] http://www.ccom.ucsd.edu/~elwong/
[27] http://ismp2012.mathopt.org/
[28] http://link.springer.com/chapter/10.1007/BFb0080116
[29] http://link.springer.com/chapter/10.1007/BFb0080116
[30] http://link.springer.com/article/10.1007/BF02141261
[31] http://numericalanalysisconference.org.uk/
[32] https://archive.org/details/jresv45n4p255
[33] http://link.springer.com/article/10.1007%2FBF01588325
[34] https://en.wiktionary.org/wiki/yonks
[35] http://researcher.watson.ibm.com/researcher/view.php?person=us-arconn
[36] http://epubs.siam.org/doi/abs/10.1137/0706028
[37] https://en.wikipedia.org/wiki/Martin_Beale
[38] http://imajna.oxfordjournals.org/content/5/3/247.abstract
[39] http://link.springer.com/article/10.1007%2FBF02591933
[40] http://pages.cs.wisc.edu/~swright/
[41] http://link.springer.com/article/10.1007%2FBF02023102
[42] http://link.springer.com/article/10.1007/s101070050113
[43] http://journals.cambridge.org/action/displayAbstract?fromPage=online&aid=8877390&fileId=S0962492913000032
[44] http://link.springer.com/article/10.1007%2FBF01582292
[45] http://www.mathcs.emory.edu/~benzi/
[46] http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.22.452
[47] http://www.cambridge.org/us/academic/subjects/mathematics/numerical-analysis/approximation-theory-and-optimization-tributes-m-j-d-powell
[48] http://link.springer.com/article/10.1007/s101070100244
[49] http://imajna.oxfordjournals.org/content/8/1/141.short
[50] http://www.cc.ac.cn/staff/dyh.html
[51] http://epubs.siam.org/doi/abs/10.1137/S1052623494266365
[52] http://www.maths.ed.ac.uk/hall/EUROPT15/
[53] guff – nonsense baloney, see https://en.wiktionary.org/wiki/guff
[54] http://www.cuter.rl.ac.uk/Problems/mastsif.shtml

Philippe Toint

Impressions of Roger's Interview

I have known Roger Fletcher since June 1975, when I attended my first Dundee Conference on Numerical Analysis as a young unexperienced PhD student. It was my first international meeting and the warm welcome in the optimizer's community of the time was a determining factor in my option to continue research in this area. This welcoming attitude was manifested by impressive luminaries in the domain, in particular by Olvi Mangasarian (who sat next to me in the coach from Edinburgh's airport to Dundee) and by Roger Fletcher, the "optimization host" of the conference.

Roger immediately struck me by his openness, honesty, and unassuming attitude towards young lads like me. I remember in particular watching the Wimbledon finals at his home where he and his wife Mary had very kindly invited me. As it turns out, we share the same interest for mountain trekking and I remember very well a climb of the Lochnagar peak along with him, Michael Powell and a Thai fellow student, as well as climbing another Munro in the mist with Nick Gould and him, where the only thing we saw for the whole day was our feet in puddles of black peaty water! Thus I came to appreciate Roger as a person before I would, in due time, fully appreciate his considerable mathematical contributions beyond the then already-famous BFGS formula. Reading the interview above was therefore a real pleasure as it reveals very nicely both the breadth and depth of his mathematical work and also his "hands on" approach, favouring experience and dialog with users. In reading the interview, I certainly learned a few things on his early career (when I was barely in high school). Although I had the pleasure to co-author a couple of papers with Roger on convergence theory for filters, I always felt that his real interest was in algorithm design, making sure a particular problem could be solved efficiently and reliably.

In this respect I would like to point out some of the contributions mentioned in the interview that I have always considered as major: BFGS (of course), but also SL1QP, SLP-EQP, exact augmented Lagrangians, and filter methods. These have all had a strong influence on my own work. I must admit I was really surprised that he downplays his contribution to trust-region methods (he called them "restricted step methods" in his wonderful book), giving a pointer to a paper by Beale (which I confess ignoring). This is far too modest in my view (and quite typical of him): as an excellent advocate, he definitely contributed significantly to making these methods the de-facto standard they are today.

One other thing that I learned in the interview is that Roger, like me and so many others, spent some time at the CERFACS in Toulouse. I am glad to hear that the CERFACS also played a role in Roger's career since it is there that he met Michele Benzi, resulting in noticeable work in numerical linear algebra. This private research institution has indeed be very welcoming to more than a generation (at least three, it seems) of applied mathematicians, and it is a great place to meet colleagues and learn about real industrial problems. Talking about industrial optimization problems naturally leads me to support Roger's expressed preference for such problems over standard test problems from the literature. While environments such as CUTE(st) remain absolutely crucial for tuning, comparison and standardization, the real purpose of optimizers is to provide tools for solving other people's practical problems. In the end, it is the solution and its quality that count, rather than the number of good/bad iterations or the CPU time used.

Reading on, I also had the pleasant surprise to read Roger showing an interest in high precision arithmetic (when he supports Mike Saunders's suggestion to use quadruple precision). Interestingly, Gerard Meurant and I raised the same issue again at the recent Sparse-Days in St-Girons (another CERFACS activity!) where the future and needs of high-performance computing were discussed by world experts. With the size of problems solved now getting closer to the inverse of machine epsilon in double precision, ignoring the issue and relying solely on random averaging of rounding errors seems a bit optimistic indeed.

I really spent a very good moment reading Roger's interview, which reminded me of so many things and exchanges, and also of the scope of his work (and of his smooth/dry humour). Well done Roger! It is a privilege to have you in the optimization community.

Philippe Toint, Department of Mathematics, The University of Namur (FUNDP), 61, rue de Bruxelles, B5000 – Namur, Belgium
philippe.toint@fundp.ac.be

Frank E. Curtis

Young Researchers Would Be Wise to Read this Interview!

Not only does the interview provide excellent advice solicited from a researcher who consistently has been a pioneer in his field, but it also illustrates that one can achieve such stature while being motivated by nothing more than a passion for science, and by possessing nothing but the utmost humility and respect for colleagues. Thanks to Optima for allowing us to have this glimpse into the mind of a true innovator and inspirator.

Fletcher's is a name found in all top textbooks on nonlinear optimization. And I am not just talking about their bibliographies, but about their tables of contents and indexes as well! Davidon-Fletcher-Powell (DFP), Fletcher-Reeves, Broyden-Fletcher-Goldfarb-Shanno (BFGS), Fletcher-Matthews, etc.—so many ideas that are used by and continue to inspire so many. He may claim that some of these accomplishments were about being in the right place with the right collaborators at the right time, but shame on us if we believe that such a thing can happen merely by chance so often within a single career. Let us all agree that Fletcher's reputation is truly well-deserved.

You would be hard pressed to find modern nonlinear optimization software that has not been influenced by Fletcher's work, and these influences promise to continue well into the future of the field. Take unconstrained optimization algorithms as an example. As such approaches will continue to be tasked to solve larger and more challenging problems, quasi-Newton methods will continue to serve as the go-to algorithms when Newton methods are intractable and steepest descent methods prove to be wholly inadequate. Fletcher and Powell's efforts to bring Davidon's idea to prominence truly was, as stated by Nocedal and Wright [1, pages 135–136], a "dramatic advance [that] transformed nonlinear optimization overnight." And this was only one of Fletcher's very early contributions! Essentially all leading software packages provide a quasi-Newton option when second-order derivatives are unavailable, and many offer it as their default approach.

But beyond the discussions of Fletcher's contributions of which many are already aware, what I find even more interesting about this interview are the mentions of his numerous contributions that may be lesser known. For example, as they form the foundations of much recent work on the subject (including my own), I am well aware of his fantastic contributions to constrained nonlinear optimization related to Sl1QP, SLP-EQP, and filters, but I have yet to look carefully at all of his work on expected conditioning, factorization methods, techniques for handling degeneracy, and more. (To be sure, these oversights on my part are being remedied as we speak!) A career built on such a comprehensive study of nonlinear optimization methods and software explains how prominent has been Fletcher's impact and how his own codes – e.g., filterSQP, filterSD, BQPD, etc. – represent some of the most reliable software available today.

Reading through the interview, I find numerous pieces of advice that are undoubtedly spot-on: (1) if you are interested in solving problems, then start by looking at the problems and focus on computation; (2) don't believe what you read in books, try it yourself; (3) talk to industry to see what's needed, even if selling yourself to them is not something you're good at; and (4) be critical of your own work, and reexamine it even after it's done. If you look over Fletcher's career, then you can see that this isn't just idle talk, but advice that he has followed himself. Take, for instance, his recent work on limited memory steepest descent methods, which he shows personally to be competitive with quasi-Newton methods for certain large-scale problems. While others take the superiority of BFGS-type methods almost as fact, Fletcher himself (the "F"!) is reexamining them and developing viable alternatives.

But it is the advice that I find between the lines that I may remember most about this interview. Did you notice how humbly he speaks about some of his most famous work, giving so much of the credit to others, while at the same time speaking so enthusiastically about ideas that have not received as much acclaim (e.g., ultra-BFGS)? I suppose the lesson is that you never know how your work will be received or be judged, or how important it will prove to be over time. These things are too difficult to predict. It is better, instead, to trust your instincts and follow opportunities to find passion in your work. This way, even if your name ends up being associated with an array of popular methods found in textbooks throughout your field—which is more than most of us could ever hope for—these won't represent the only contributions in which you will find fulfillment about your career.

To end on a personal note, I would like to add that I am immensely grateful to work in a field with leaders such as Roger Fletcher. For one of my first visits to a university overseas, Roger came down from Dundee to Edinburgh to attend my talk and, as it turned out, to chat with me on the side for a few minutes during the reception after the seminar. Especially after I spoke critically in my talk about the performance of one of his codes on a certain class of problems, one might think that the experience could have been quite intimidating for me! But with Roger the experience was nothing but a pleasure, and I can say without hyperbole that it will remain one of the most memorable conversations of my career.

Reference

[1] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006.

Frank E. Curtis, Department of Industrial and Systems Engineering, Lehigh University, 200 West Packer Ave., Bethlehem, PA 18015-1582, USA
frank.e.curtis@gmail.com

Coralia Cartis, Andreas Griewank, Philippe Toint and Ya-xiang Yuan

Obituary for Mike Powell

The optimization community has lost an outstanding member of the founding generation of nonlinear optimization, or programming as it used to be called. In the last century Mike Powell was probably the most influential optimizer in Europe, if we may lump England in with 'the continent', where Mike served as a young lad in the British forces shortly after the war. His politics were never very explicit but he certainly despised small minded nationalism of any stripe. Naturally, there were other very distinguished nonlinear optimizers from England, especially Roger Fletcher and Charles Broyden, who left us two years ago. Mike did not spread his influence by modern day *networking* or strategically placing the students of a *Powell school*, but by the shear excellence of his mathematical contributions and algorithm development.

Even in that respect he did not make things easy to the paper reader or program user. One finds no modularization of convergence theories into easily digestible and snappily named building blocks, and faces intricate Fortran 66 codes that were written for computers to read not for humans, as Mike reportedly said. His presentations with handwritten slides were always very well prepared and the symbols carefully color coded. Sometimes one could inspect the third Lagrange multiplier on the fifth iteration copied with at least four digits to highlight some peculiar behavior. Like most everything he did, Mike's algorithmic development was pains' taking, guarding against round off errors and other calamities. He personally checked all references in the proceedings of the 1981 Cambridge workshop on Nonlinear Programming.

Michael J. D. Powell was born on the 29th of July 1936 in Kensington as the eldest of three children and died in Cambridge on April 19th of this year. He obtained a diploma in numerical analysis

Michael J. D. Powell (Photo: Ya-xiang Yuan)

and computing from Peterhouse college, University of Cambridge in 1959. In the same year he married Caroline and they had two daughters and one son. He obtained neither Master nor PhD and worked 17 years at Harwell Laboratory belonging to the Atomic Energy Commission. Then at the age of 40 he returned to Cambridge as a Professor, and received the Dr. of Science for a collection of published papers on approximation theory. He was elected fellow of the Royal Society in 1983, member of the US National Academy of Science in 2001, and Fellow of the Australian Academy of Science in 2007. He was the first winner of the Dantzig prize in 1982 and in 1992 gave a 45-minute address at the International Congress of Mathematics in Warsaw. He was founding editor of the IMA journal on Numerical Analysis.

The nonlinear optimization literature is strewn with Mike's contributions. Some rather small, like the magic factor 0.8 in the definiteness test for BFGS updating Hessians of Lagrangians, others unsurpassed in their ingenuity, like the global convergence proof for BFGS on convex problems. In a paper in 1969 he had the original idea of augmented Lagrangian penalty functions for constrained optimization, which was instrumental for the success of SQP methods. Mike was also the pioneer of trust region algorithms. Indeed, in 1970 he proved the first convergence result for trust region methods in the unconstrained case, (though he did not call it *trust region* then). In terms of acronyms he is immortalized in the Davidon Fletcher Powell (DFP) method, which quickly became superceded by its even more attractive sister BFGS. There is also the Powell symmetric Broyden Update (PSB), which yields local and superlinear convergence even in its sparse variant, but due to its strong scaling dependence is much less successful than the Variable Metric Methods DFP and BFGS. Finally, there is the CPR (Curtis Powell Reid) scheme for grouping variables to reconstruct sparse Jacobians from just a few directional derivatives, usually approximated by divided differences. The scheme was later utilized in automatic differentiation, which Mike never considered to be part of optimization and thus his domain of interest. Rather, he later returned to much earlier work by himself and Richard Brent on derivative free methods that still implicitly assume enough smoothness to build up a local quadratic model. On the other hand, he is probably partly responsible for the skeptical view that the classical optimization community always has held of random search methods, be they inspired by evolutionary analogs or not. He believed that local optimality is all one can get in the nonconvex case.

It is curious that Mike edited quite a few proceedings, but never himself wrote a book on optimization. Possibly he thought that Roger Fletcher and others had done such a fantastic job. Probably motivated by his interest in radial basis functions, he did write a book on approximation theory, which was quite well received, and contained many original contributions concerning the univariate case. It was Mike's characteristic strength that, when faced with a mathematical problem, he would always first try to work it out by himself using his own notation and terminology. Only afterwards, often somewhat reluctantly, he would check out the literature and stoop to notational conventions. Of course, that gave him much deeper mathematical insights than the superficial understanding that one might gain nowadays from a quick electronic glance at the literature. Also, he had a much better chance of coming up with something truly original, rather than by group research into *hot topics*, which he disliked. In his long 2005 interview with Philip Davis for SIAM News he went as far as calling himself a *loner*. He certainly had no taste for university administration and committee work at the national or international level.

In the SIAM interview and his last work on derivative free algorithms he very much stressed the importance of good numerical performance, downplaying the relevance of convergence theory. By some accounts the emphasis used to be somewhat different, with him expecting water tight arguments from his students for whatever algorithmic idea they were proposing. Possibly, the decades long chase for the ultimate elucidation of the stellar BFGS performance has shifted his preferences. No doubt he always excelled at conducting systematic numerical experiments and drawing conclusions for the algorithmic design. In nonlinear constrained optimization that is no small feat, as various aspects of method and test problems tend to interact in seemingly erratic ways. Mike usually worked on small scale test case, which he sometimes ingeniously made up by himself to highlight certain difficulties. He had neither the inclination nor the software dexterity to generate comparative performance profiles on whole test sets, which a journal referee might ask for these days.

Mike combined a sharp mind, strong opinions and a competitive spirit with personal kindness, especially towards his students and associates and their families. He never got side tracked by outward appearances, and always focussed on the mathematical essentials, including seemingly minute details. For all this he will be missed and can serve as an example to young researchers in optimization and beyond. We, as members of the OMS editorial board, who did (post)doctoral research under Mikes supervision and inspiration bid him gratefully farewell.

Coralia Cartis, Mathematical Institute, University of Oxford, Woodstock Road, Oxford, OX2 6GG, Great Britain
coralia.cartis@maths.ox.ac.uk

Andreas Griewank, Institut für Mathematik, Humboldt Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany
griewank@mathematik.hu-berlin.de

Philippe Toint, Department of Mathematics, The University of Namur (FUNDP), 61, rue de Bruxelles, B5000 – Namur, Belgium
philippe.toint@fundp.ac.be

Ya-xiang Yuan, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, P.O. Box 2719, Beijing 100080, China
yyx@lsec.cc.ac.cn

This text has first appeared in: *Optimization Methods and Software* 30 (3), 2015, 669–671 (DOI:10.1080/10556788.2015.1051808). We are grateful to the authors and to the publisher Taylor & Francis for granting permission to reprint it here.

Jong-Shi Pang

Obituary for Che-Lin Su

It is with a very heavy heart and tremendous sadness that I write this article to honor Dr. Che-Lin Su of the Chicago Booth School of Business. Che-Lin passed away on Friday July 31, 2015 after a short battle with cancer. He was 41 years old. Che-Lin was my academic brother, an inspired collaborator, an accomplished researcher, a dedicated teacher, a dear friend, and a fun meal partner.

Che-Lin and I were Ph.D. students of Richard W. Cottle at Stanford University, with Che-Lin being Professor Cottle's last doctoral student. Che-Lin's major line of research was to apply the methodology, theory, and software of constrained optimization to an important area of econometrics known as structural estimation. He was the first person in the field of operations research and mathematical programming to make a dent in this area.

Che-Lin published two seminal papers in the highly demanding journal Econometrica on his approach to structural estimation. The first was co-authored with Kenneth Judd of the Hoover Institution at Stanford University and the second with Jean-Pierre Dubé of the University of Chicago and Jeremy Fox of the University of Michigan. Together with Yu-Ching Lee (now at National Tsing Hua University in Taiwan), Che-Lin and I co-authored a paper accepted for publication in Operations Research that illustrates Che-Lin's contributions for an operations research audience. In this paper, we offer a constructive approach to resolve a structural estimation problem in a pure characteristics demand model incorporating consumers' utility maximization in their product choice. This approach resolves a major computational difficulty of the model proposed originally by famed economists, which has hindered its wide-spread application in such areas as marketing and econometrics. In addition, our optimization approach enables us to extend the model to allow producers to be competitive Nash-Bertrand players in setting the prices of the products desired by the consumers.

Che-Lin's work on structural estimation attracted the attention of prominent economists at Harvard and Yale University where he was invited as a Visiting Professor and Visiting Associate Professor, respectively, of Economics. Most impressively, Che-Lin had no formal training in economics and yet was welcomed by two of the most prestigious economics departments in the world as a guest faculty, not a small feat indeed.

Our field has lost a distinguished young scholar, a wonderful colleague, a dedicated teacher, and most important of all, a kind, gracious, and beloved human being who will be missed by all his friends, colleagues, and acquaintances. Good bye, Che-Lin, and may peace be with you forever.

Che-Lin Su (Photo: Todd Munson)

Jong-Shi Pang, Daniel J. Epstein Department of Industrial and Systems Engineering, 3715 McClintock Ave, GER 240, Los Angeles, CA 90089-0193, USA
jongship@usc.edu

ICCOPT 2016 – Tokyo, Japan

August 6–11, 2016. The 5th International Conference on Continuous Optimization (ICCOPT 2016) will be held in Tokyo at the National Graduate Institute for Policy Studies (GRIPS), Roppongi.

ICCOPT, one of the three flagship conferences of the Mathematical Optimization Society, is dedicated to research on continuous optimization and related topics. This gathering consists of a conference and a Summer School.

Plenary Speakers

Twelve distinguished researchers will deliver plenary and semi-plenary lectures at the conference:

- Francis Bach (INRIA, France)
- Florian Jarre (Heinrich Heine Universität Düsseldorf, Germany)
- Jong-Shi Pang (University of Southern California, USA)
- Shuzhong Zhang (University of Minnesota, USA)

Semi-Plenary Speakers

- Yu-hong Dai (Chinese Academy of Sciences, China)
- Erick Delage (HEC Montréal, Canada)
- Mirjam Dür (Universität Trier, Germany)
- Katsuki Fujisawa (Kyushu University, Japan)
- Elad Hazan (Princeton University, USA)
- Jonathan Kelner (Massachusetts Institute of Technology, USA)
- Caroline Uhler (Institute of Science and Technology, Austria)
- Rachel Ward (University of Texas at Austin, USA)

There will also be a paper competition for young researchers in Continuous Optimization (see p. 10 of this issue; further information is available from the website below).

Roppongi is an attractive and vibrant area in the middle of downtown Tokyo, well-known as the center of Japanese contemporary culture and entertainment.

Summer School

August 6–7, National Olympics Memorial Youth Center (NYC), Yoyogi. The Summer School provides students and young researchers with an opportunity to become familiar with leading-edge developments in continuous optimization. The school will offer two courses taught by four distinguished lecturers.

Course 1: First-order, Splitting, and Related Methods for Big-Data Optimization

Lecturers: ○ Michael Friedlander (University of California Davis, USA); ○ Kim-Chuan Toh (National University of Singapore, Singapore)

Course 2: Links Between Continuous and Discrete Optimization
Lecturers: ○ Antoine Deza (McMaster University, Canada); ○ Kazuo Murota (Tokyo Metropolitan University, Japan)

Yoyogi is a very pleasant area situated on the edge of Yoyogi park, one of the largest parks in Tokyo. Located in between the large centers of Shinjuku and Shibuya, it is 3 km west of Roppongi and only minutes away from GRIPS by train or metro.

Call for Proposals for Contributed Talks and Poster Presentations

Please visit the conference website to submit your abstract. The deadline for submitting a proposal for a contributed talk is April 15; the deadline for submitting a proposal for a poster presentation is May 16.

Deadlines

March 15. Deadline for student applications for Summer School accommodation
April 15. Deadline for submission of abstracts for parallel sessions
Deadline for non-student applications for Summer School accommodation
May 16. Deadline for submission of abstracts for poster sessions
May 31. Deadline for presenter registration
Deadline for early-bird registration

We look forward to seeing you in Tokyo!

Call for Nomination/Submission
Best Paper Prize for Young Researchers in Continuous Optimization

Nominations/Submissions are invited for the Best Paper Prize by a Young Researcher in Continuous Optimization. The submitted papers should be in the area of continuous optimization and satisfy one of the following three criteria:

(a) Passed the first round of a normal refereeing process in a journal;
(b) Published during the year of 2013 or after (including forthcoming);
(c) Certified by a thesis adviser or postdoctoral mentor as a well-polished paper that is ready for submission to a journal.

Papers can be single-authored or multi-authored, subject to the following criterion:

(d) Each paper must have at least one qualifying author who was under age 30 on January 1, 2011 and has not earned a PhD before that date. In case of joint authorship involving senior researchers (i.e., those who fail both the age test and the Ph.D. test), one senior author must certify the pivotal role and the relevance of the contribution of the qualifying author in the work. The Selection Committee will decide on questions on eligibility in exceptional cases.

The selection criteria will be based solely on the quality of the paper, including originality of results and potential impact. The following items are required for submission:

A. The paper for consideration;
B. A brief description of the contribution (limited to 2 pages);
C. A statement about the status of the paper: not submitted, under review, accepted, or published (when) in a journal;
D. A certification of the qualifying author's eligibility in terms of age and Ph.D. (by the qualifying author's adviser or department chair);
E. In case of joint authorship involving a senior researcher, a certification by the latter individual about the qualifying author's pivotal role and relevance of the contribution.

The deadline for submission is March 15, 2016. Submission should be sent electronically in Adobe Acrobat pdf format, to the Chair of the Selection Committee, Professor Andrzej Ruszczynski, email address: rusz@business.rutgers.edu

Up to three papers will be selected as finalists of the competition.The finalists will be featured in a dedicated session at ICCOPT 2016, and the Prize Winner will be determined after the finalist session. The Young Researcher Prize in Continuous Optimization will be presented at the conference banquet.

The finalists will receive free registration to ICCOPT 2016 and to the conference banquet. Their university or department should cover the travel costs. All the three finalists will receive a diploma, and the winner will be presented a 1000 USD award.

ISCO 2016
Vietri sul Mare (Salerno), Italy

May 18–20, 2016. The 4th International Symposium on Combinatorial Optimization (ISCO 2016) will take place in Vietri sul Mare (Salerno), Italy. It will be preceded by a Spring School on "Extended Formulations in Combinatorial Optimization" on May 16–17, 2016.

ISCO is a biennial symposium whose aim is to bring together researchers from all the communities related to combinatorial optimization, including algorithms and complexity, mathematical programming and operations research. It is intended to be a forum for presenting original research in these areas and especially in their intersections. Quality papers on all aspects of combinatorial optimization, from mathematical foundations and theory of algorithms to computational studies and practical applications, are solicited.

Sessions

The conference is organized into plenary and parallel sessions. The conference languages is English. Each speaker can give only one talk. Proposals for invited sessions are welcome. Researchers who are interested in organizing an invited session should contact Raffaele Cerulli (raffaele@unisa.it).

Spring School

ISCO 2016 will be preceded by a spring school on "Extended Formulations for Combinatorial Optimization". Volker Kaibel and Samuel Fiorini will give 16 hours of lectures on May 16 and 17, 2016.

Important Dates

February 15, 2016. Submissions deadline
March 20, 2016. Notification of authors
April 4, 2016. Early registration deadline
(conference and spring school)
May 18-20, 2016. Conference
May 16–17, 2016. Spring School

Vietri sul Mare (Photo: M2m, Wikimedia Commons)

Keynote Speakers

- Volker Kaibel (Otto-von-Guericke-Universität Magdeburg)
- Adam Letchford (Lancaster University)
- Ramamoorthi Ravi (Carnegie Mellon University)

Submission – Publication

Papers presenting original unpublished results in all areas of combinatorial optimization and its applications are welcome. The submission deadline is Monday February 15, 2016. Simultaneous submissions to other conferences with published proceedings or journals are not allowed.

Two types of submissions: Regular papers (with up to 12 pages) and short papers (2 pages).

Accepted regular papers will be published by Springer-Verlag in the *Lecture Notes in Computer Science* (LNCS) series in a post-conference proceedings volume. The authors will have to prepare their camera-ready version two weeks after the end of ISCO 2016.

More information about the submission procedure is available on the website of the conference.

Special Issues

A special issue of *Networks* and other international journals will be associated to ISCO2016

Conference Chairs

- Raffaele Cerulli (University of Salerno, Italy)
- Satoru Fujishige (Kyoto University, Japan)
- A. Ridha Mahjoub (University Paris Dauphine, France)

Steering Committee

- M. Baïou (LIMOS, CNRS, University Blaise Pascal, Clermont-Ferrand, France)
- P. Fouilhoux (University Pierre and Marie Curie, Paris, France)
- L. Gouveia (University of Lisbon, Portugal)
- N. Maculan (Universidad Federal do Rio de Janeiro, Brasil)
- A. R. Mahjoub (University Paris-Dauphine, France)
- V. Paschos (University Paris-Dauphine, France)
- G. Rinaldi (IASI, Rome, Italy)

Organizing Committee

- Raffaele Cerulli (University of Salerno, Italy)
- Francesco Carrabs (University of Salerno, Italy)
- Monica Gentili (University of Salerno, Italy)
- Ciriaco D'Ambrosio (University of Salerno, Italy)
- Andrea Raiconi (University of Salerno, Italy)
- Carmine Cerrone (University of Salerno, Italy)
- Rosa Pentangelo (University of Salerno, Italy)
- Selene Silvestri (University of Salerno, Italy)

For more information on the Symposium and School, you can write to info@isco2016.it or consult the website of the conference: www.isco2016.it

First announcement

Mixed Integer Programming Workshop Miami, FL

May 23–26, 2016. We are pleased to announce that the 2016 workshop in Mixed Integer Programming (MIP 2016) will be held at the University of Miami, in Coral Gables, FL.

The 2016 Mixed Integer Programming workshop will be the thirteenth in a series of annual workshops held in North America designed to bring the integer programming community together to discuss very recent developments in the field. The workshop consists of a single track of invited talks and features a poster session that provides an additional opportunity to share and discuss recent research in MIP. Registration details, a list of confirmed speakers, a call for participation in the poster session, and information about student travel awards will be made in a subsequent announcement.

Program Committee

- Alberto Del Pia (chair), University of Wisconsin-Madison
- Sanjeeb Dash, IBM Research
- Fatma Kilinc-Karzan, Carnegie Mellon University
- Dan Steffy, Oakland University
- Kati Wolter, MOSEK ApS

Local Committee

- Tallys Yunes, University of Miami
- Hari Natarajan, University of Miami

Further information and updates:
https://sites.google.com/site/mipworkshop2016/

ICCOPT 2019 – Call for Site Proposals

The ICCOPT Steering Committee of the Mathematical Optimization Society (MOS) is requesting proposals for organizing ICCOPT VI, the Sixth International Conference on Continuous Optimization, which is scheduled to be held in August 2019 and follows up the forthcoming one in 2016 (see Page 9 in this newsletter). Being the flagship conference of the MOS in the area of continuous optimization, ICCOPT is held every three years at a site to be selected according to the criteria below. For information about the forthcoming one in 2016 and two prior ones, visit www.iccopt2016.tokyo, http://eventos.fct.unl.pt/iccopt2013 or www.iccopt2010.cmm.uchile.cl.

The proposal for organizing ICCOPT VI should include the candidate site and Organizing Committee. Selection criteria for the site are based on the following considerations:

- Existence of continuous-optimization researchers in the proposed geographic area who are interested in and can assist in the organization of ICCOPT VI.
- Attendance open to prospective participants from all nations.
- Availability of an attractive facility with a sufficient number of meeting rooms, standard lecture equipment, etc., preferably on a university campus.
- Availability of a sufficient supply of reasonably economical hotels and/or university dormitory rooms fairly near the meeting facility.
- Sites outside the next one (Tokyo, Japan) and the one prior (Lisbon, Portugal), such as Canada and the U.S. are particularly encouraged to apply, although the Committee is interested in the most compelling proposal regardless of the continent.

Some Characteristics of Previous ICCOPT Conferences

- Past such conferences had between 200 and 500 participants.
- Plenary, semiplenary lectures, invited and contributed sessions, poster session and poster competition.
- The lengths of the first (at Rensselaer Polytechnic Institute, Troy, New York) second (at McMaster University, Hamilton, Canada), third (in Santiago, Chile), and next ICCOPT were 3, 4, 4, and 4 days, respectively, excluding the tutorial workshop mentioned below.
- A 1 or 2-day long tutorial workshop for graduate students.
- All inclusive conference coffee and lunches.
- Young Researchers Prize in Continuous Optimization.
- Program and Prize committees formed in consultation with the Steering Committee.
- Social events, student's social, banquet.
- Reasonably low registration fee.
- No proceedings of papers, no competitive selection of talks, but only one presentation per paid participant.

Further information can be obtained from any member of the Steering Committee:

- Jong-Shi Pang, Chair (jongship@usc.edu)
- Coralia Cartis (coralia.cartis@maths.ox.ac.uk)
- Dominique Orban (dominique.orban@gerad.ca)
- Nick Sahinidis (sahinidis@cmu.edu)
- Fabio Schoen, (fabio.schoen@unifi.it)
- Mikhail Solodov (solodov@impa.br)
- Takashi Tsuchiya (tsuchiya@grips.ac.jp)
- Michael Ulbrich (mulbrich@ma.tum.de)

Submission Deadline

June 15, 2016 to the Chair of the ICCOPT Steering Committee: Jong-Shi Pang (jongship@usc.edu).

Hosts of the candidate sites should send an email to the Committee Chair by April 15, 2016 to indicate interest to submit a proposal.

Application for Membership

I wish to enroll as a member of the Society. My subscription is for my personal use and not for the benefit of any library or institution.

☐ I will pay my membership dues on receipt of your invoice.
☐ I wish to pay by credit card (Master/Euro or Visa).

Credit card no. *Expiration date*

Family name

Mailing address

Telephone no. *Telefax no.*

E-mail

Signature

Mail to:
Mathematical Optimization Society
3600 Market St, 6th Floor
Philadelphia, PA 19104-2688
USA

Cheques or money orders should be made payable to The Mathematical Optimization Society, Inc. Dues for 2015, including subscription to the journal *Mathematical Programming*, are US $90. Retired are $45. Student applications: Dues are $22.50. Have a faculty member verify your student status and send application with dues to above address.

Faculty verifying status

Institution

IMPRINT
Editor: Volker Kaibel, Institut für Mathematische Optimierung, Otto-von-Guericke Universität Magdeburg, Universitätsplatz 2, 39108 Magdeburg, Germany. kaibel@ovgu.de ■ **Co-Editors:** Samuel Burer, Department of Management Sciences, The University of Iowa, Iowa City, IA 52242-1994, USA. samuel-burer@uiowa.edu ■ Jeff Linderoth, Department of Industrial and Systems Engineering, University of Wisconsin-Madison, Madison, WI 53706-1572, USA. linderoth@wisc.edu ■ **Founding Editor:** Donald W. Hearn ■ Published by the Mathematical Optimization Society ■ Design and typesetting by Christoph Eyrich, Berlin, Germany. optima@0x45.de ■ Printed by Oktoberdruck AG, Berlin, Germany.