Research in Continuous Optimization:

Incremental,

Transformational,

or Recycled?

Margaret H. Wright
Computer Science Department
Courant Institute of Mathematical Sciences
New York University

Department of Energy
Applied Mathematics Principal Investigators Meeting

May 22, 2007

Official purpose of this talk: to give an overview that focuses on the *state of the art in optimization* and includes a *broad perspective on the field*, particularly as it relates to the DOE Office of Science mission area. . . in 40 minutes!

[Not possible.]

My greatly reduced goal: To stir things up a bit.

Preemptive apology to optimization experts: the talk is intended for a general audience, so please try to forgive me for saying what you already know.

The broad policy context: Growing concerns that U.S. science funding has become too conservative and too risk-averse.

What's wanted: innovative, high-risk/high-reward, bold research.

National Academies' Report *Rising Above the Gathering Storm* (2007): the United States needs creative, out-of-the-box transformational research that could lead to new ways of fueling the nation and its economy, as opposed to incremental research on ideas that have already been developed.

Testimony to Congress by Arden Bement, Director of the National Science Foundation, March 29, 2006:

"Creative disruption at the frontier and reduced lead-time between discovery and application are the principal drivers of global competition today. . . . Tinkering on the sidelines may be important, but it is not what drives cutting-edge innovation".

National Science Board report: "Transformative research is . . . driven by ideas with the potential to radically change our understanding of an important existing scientific or engineering concept. . . [it is] characterized by its challenge to current understanding or its pathway to new frontiers".

NASA and NIH have expressed similar wishes to fund transformational rather than incremental research.

And DOE wants transformational research also!

Raymond L. Orbach, Under Secretary for Science, Department of Energy, March 9, 2006:

"The Department of Energy will need to fund and perform science that is world-class, science that is at the far frontier of human knowledge, what I call transformational science. Transformational science is science that opens entirely new avenues and methods for solving problems, that gives us revolutionary new tools for mastering the challenges of our world. . . . Incremental changes . . . will not suffice; we need transformational discoveries and truly disruptive technologies".

What do "transformational" and "incremental" really mean?

Does it matter?

Are they just trendy buzzwords?

Should DOE-funded researchers care?

In the spirit of a thought exercise for this meeting: how does optimization research (or any applied mathematics research) stack up, using the ideas implicit in these statements?

- Are there genuinely new ideas with the potential to transform our ability to analyze and solve difficult problems?
- Is progress primarily incremental?
- To what extent are we recycling old ideas, shaping them into new forms, and applying them to new problems?
- How does the interplay between theory and implementation affect the prospects for transformational research?

A fundamental problem: distinguishing between transformational and incremental research is difficult at best, and may well be possible only with the perspective of extended hindsight.

Three examples from optimization for us to ponder:

1. Linear programming methods, 1947–today, including the interior-point revolution that began in 1984.

   Enough hindsight to make reasonable judgments.

2. Filter methods for constrained optimization, 1996–today.

   Some, but not yet enough, perspective.

3. Sparse recovery with the Dantzig selector, 2006–today.

   Too early to judge, but a case to watch.

Linear programming (LP) means optimizing (minimizing or maximizing) a linear function subject to linear constraints.

A variety of mathematically equivalent forms, for example:

$$\text{minimize } c^T x, \quad x \in \mathcal{R}^n$$

$$\text{subject to } Ax \geq b, \quad A \in \mathcal{R}^{m \times n}$$

How important is linear programming?

In the real world, including Department of Energy applications, its value is hundreds of billions of dollars per year!

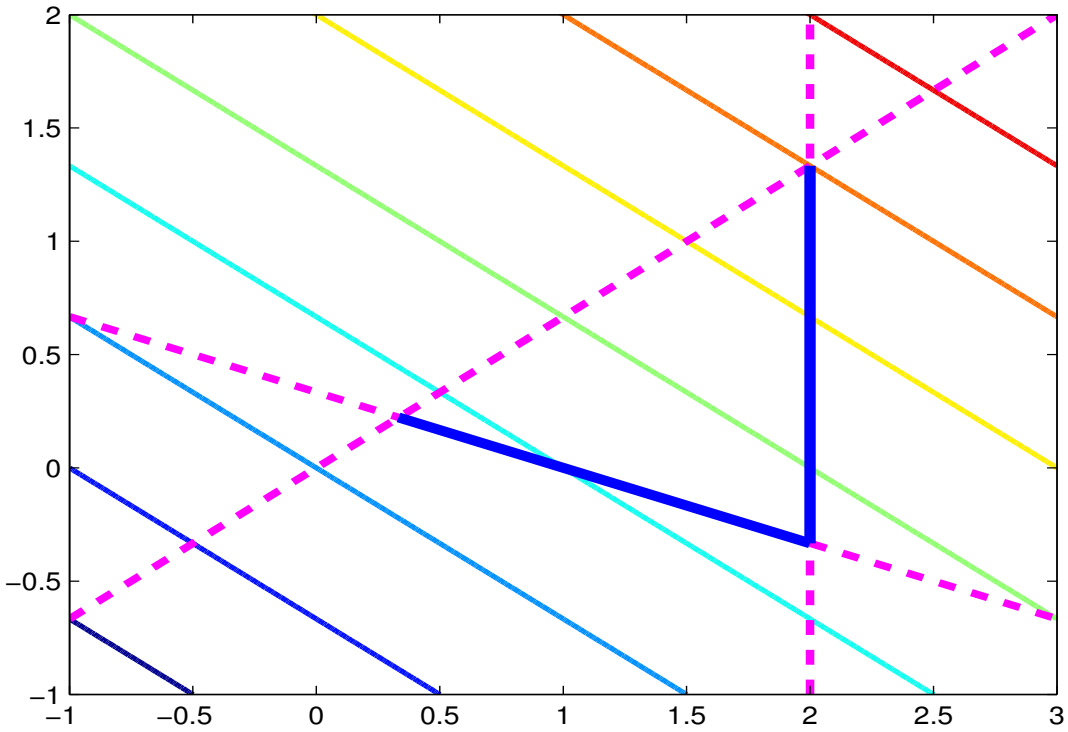
Some fields where linear programming is central:

- Analysis of energy alternatives
- Risk analysis
- Transportation
- Communication
- Finance and banking
- Manufacturing
- Health and agriculture.

Linear programming is closely tied to mathematical research in

- convex geometry
- discrete mathematics; integer programming
- complexity theory
- game theory
- linear algebra and matrix theory.

George Dantzig's great (transformational) invention in 1947: the *simplex method*, which solves linear programs by starting at a vertex and moving from vertex to vertex, reducing the objective function (if we're minimizing) as it goes.



Until the mid-1980s, practitioners (very!) happily used the simplex method to solve increasingly large linear programs.

Why were they so happy?

Because the simplex method *essentially always* took $2n$–$3n$ steps to solve the problem, where n is the number of variables, and each step took $O(n^2)$ or fewer operations.

Thus, *in practice*, the simplex method behaved like a fast polynomial-time algorithm.

However, starting in the 1960s when the study of computational complexity took off, the simplex method's success in practice did not impress the theoretical computer science community, who were concerned about its *worst-case* complexity.

Their unease was justified: in 1972, Klee and Minty produced an n-variable linear program with 2^n vertices for which the textbook simplex method, if started at the "wrong" vertex, visits every one of those 2^n vertices.

The Klee–Minty and related examples show that the worst-case complexity of the simplex method with all known pivot rules is *exponential*—which should mean that the simplex method is a VERY bad algorithm.

As a result, there was an intense search for a polynomial-time linear programming method that, in theory, would be inherently faster than the simplex method.

1979: Leonid Khachian, a 28-year-old mathematician in the Soviet Union, defined the *ellipsoid method*, the first algorithm to solve linear programs in polynomial time.

Khachian's method is based on approaches for nonlinear problems developed earlier by other Soviet mathematicians, notably Shor, Yudin and Nemirovskii. It does not rely, as the simplex method does, on highly specialized features of the linear programming problem.

Wild excitement around the world—but numerical tests quickly revealed that in practice Khachian's algorithm is *much*, *much* slower than the simplex method.

So how do we classify Khachian's work?

Transformational? YES, definitely, even though the ellipsoid method was not faster than simplex.

Incremental?? No.

Recycling?? To a limited extent.

Khachian's contribution was *highly disruptive*, challenging accepted ways of thinking about linear programming. And although he used ideas developed by others, he applied them in a new, unexpected, and (at the time) startling way.

1984: Narendra Karmarkar, a 28-year-old researcher at Bell Labs, announced a polynomial-time linear programming method stated to be consistently 50 times faster than the simplex method.

Details of his method and software were secret (AT&T-confidential).

1985: Karmarkar's algorithm was proved to be formally equivalent to barrier methods for nonlinear optimization, popular (but never used for LP) in the late 1960s. A Newton barrier method was also shown to be competitive with the simplex method on a range of problems.

How do we assess the interior-point revolution, begun with Karmarkar's work?

Transformational?? YES, definitely; continuous optimization has changed tremendously since 1984.

Incremental?? Not Karmarkar's work, but most other work since 1984.

Recycling? Not Karmarkar's seminal paper, but much subsequent work.

The transformation wrought by Karmarkar was two-fold:

1. Since 1984, researchers have developed interior-point methods for linear programming, quadratic programming, nonlinear programming, etc., etc., and have devised new kinds of optimization problems such as semidefinite programming. Results from the 1960s have been re-analyzed, developing new insights that had been overlooked.

2. Devotees of the simplex method, stunned by the challenge to its previous unquestioned dominance, were highly motivated to improve it, with remarkable results.

The cumulative power of incremental research is clearly shown by the magnitude and extent of performance improvements in the simplex method, whose basic concept has not changed since 1947.

Bob Bixby, the original developer of CPLEX (commercial software for linear and integer programming considered a benchmark standard) summarized improvements in the simplex method since 1984 in "Solving real-world linear programs: a decade and more of progress" (2002).

Improvement: Steepest-edge pivot selection

The classic "textbook" simplex pivot selection strategy depends on the scaling of the constraints.

The *steepest-edge strategy* (Goldfarb and Reid, 1977) produces the greatest rate of reduction in the objective function along any feasible edge. But it requires more calculation and was widely viewed as impractical because the extra work did not produce sufficiently better results.

It was revisited (recycled?) in the early 1990s, and today, for very large linear programs, the steepest-edge strategy typically reduces (significantly) the number of simplex iterations—but not always!

Improvements in linear algebra, all "incremental" in some sense:

- Dynamic LU factorization with threshold pivoting;
- Improved stable updates of the LU factorization;
- Taking advantage of "hyper-sparsity" so that the work is linear in the number of elements "touched" during the solves (hence approximately constant as problem size grows);
- And many more.

Bixby's numerical tests confirm the very significant cumulative benefits of these and other improvements to the simplex method.

680 linear programming models were tested, with m (the number of equality constraints) up to 7 million, imposing a time limit of 4 days per LP, using version 8.0 of CPLEX, and machines and algorithms dating back to 1990.

m	Number of LPs tested
> 0	680
> 10000	248
> 100000	73

Testing for algorithm only: Run old and new simplex algorithms on new machine.

Algorithm only	Speedup
Best simplex	960

Testing for hardware only: Run new simplex algorithms on old and new machines.

Hardware only	Speedup
Best simplex	800

Algorithmic speedup exceeds hardware speedup!

Further gains in simplex speed since 2002.

Overall, even for the largest problems, the best simplex is approximately comparable with the barrier interior-point method—sometimes much better, sometimes much worse, sometimes similar.

Many puzzles remain.

Example: Solution times in seconds for two linear programs, believed to be very similar in structure.

Version 1: approximately 5 million variables and 7 million constraints

Version 2, with a similar structure: 750,000 constraints

	Primal simplex	Dual simplex	Barrier
Version 1	1880	6413	5642
Version 2	1.8×10^6	48.3	28,161

Conclusions:

Linear programming methods are *much*, *much* better than in 1984, and progress continues. Huge linear programs formerly considered essentially impossible can be solved rapidly and routinely today.

There is no uniformly best LP algorithm.

We can't predict accurately which methods will do well on which problems.

Note especially that the transformational research did not sweep away and supersede all previous work!

A second example of transformational research:

Filter methods for constrained optimization

Problem: minimize $f(x)$ subject to $c(x) \geq 0$ and $h(x) = 0$.

Since the 1970s, sequential quadratic programming (SQP) methods have been a popular and effective solution technique, using a Newton-based formulation that minimizes a quadratic approximation to the Lagrangian function subject to linearizations of the constraints, sometimes with trust-region constraints.

A longstanding strategy for ensuring progress toward a solution from an arbitrary starting point: each new SQP iterate is required to reduce a *merit function*, a combination of the objective function and the constraint violations, e.g.

$$M(x, \rho) = f(x) + \rho v(x),$$

where $v(x)$ is a measure of the constraint violations and ρ is a penalty parameter.

Research on merit functions goes back to the 1970s, and there are numerous variations.

Choosing the penalty parameter is often problematic.

In theory, SQP methods will converge if ρ is large enough, but this does not provide guidance for selecting a concrete value for ρ in an implementation.

If ρ is too small, the method may diverge; if ρ is too large, the method may inefficiently creep along the constraint boundary.

The classic "Let the user decide" strategy is not effective since most users have little or no intuition about a good value of the penalty parameter for their problem. Default values may not work well on badly scaled problems.

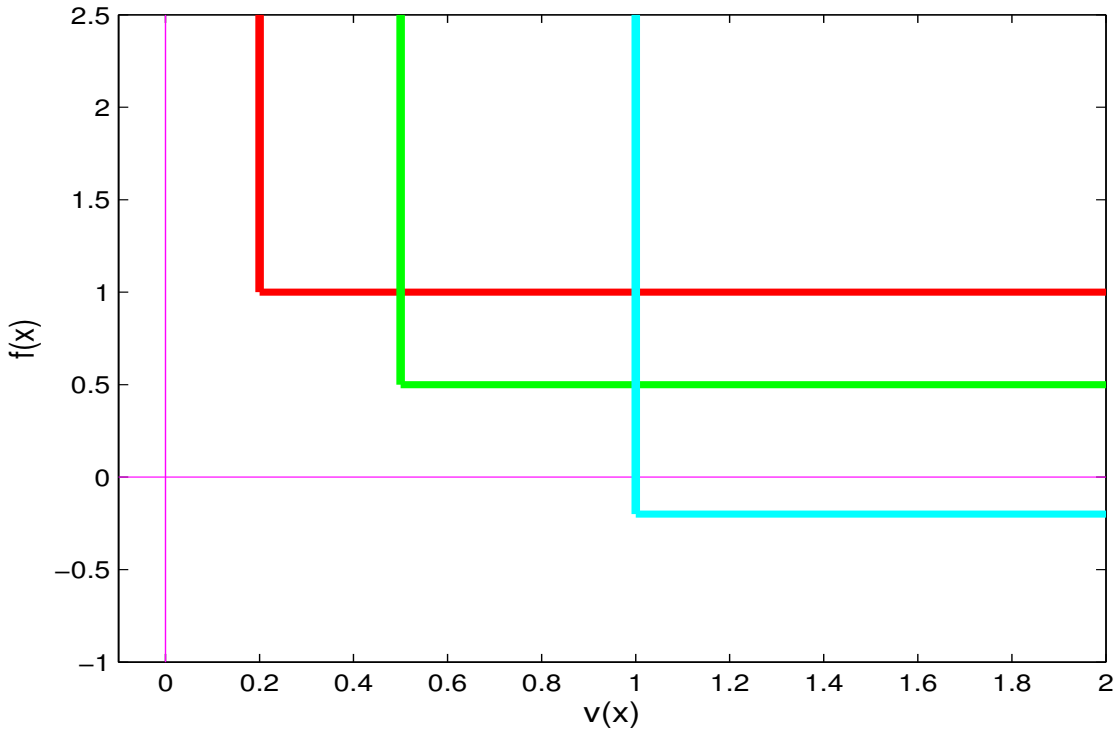
A turning point: a plenary lecture by Roger Fletcher at the 1996 SIAM Conference on Optimization describing work with Sven Leyffer on using a *filter* to measure progress.

The appealing idea of a filter: view the question of accepting a new iterate as a two-objective problem in which the goals are to minimize *both* $f(x)$ and $v(x)$.

Using the concept of "domination" from multiobjective optimization, the iterate x_j *dominates* the iterate x_i if

$$f(x_j) \leq f(x_i) \quad \text{and} \quad v(x_j) \leq v(x_i).$$

A *filter* $\mathcal{F}$ is a set of pairs $[v(x_i), f(x_i)]$ in which no point dominates any other point. A new iterate x_{k+1} is accepted only if it is *not dominated* by any point in the current filter.



Initial intuition presented by Fletcher: All points $(v(x), f(x))$ below and to the left of the filter are acceptable.

Fletcher's talk was a transformational moment: a challenge to the accepted paradigm.

Since then, it has turned out that refinements to the original idea are needed to provide an efficient algorithmic tool.

1. A small "envelope" is added around the border of the filter, and a new iterate x_{k+1} is acceptable if

$$v_{k+1} \leq \beta v_j \quad \text{or} \quad f_{k+1} \leq f_j - \gamma v_{k+1},$$

   for $(v_j, f_j) \in \mathcal{F}$ and for $j = k$, where $0 < \beta, \gamma < 1$.

2. If the constraint violations become small, a sufficient reduction condition on the objective function is imposed similar to that in unconstrained optimization.

3. If the current point is too far from feasibility, an SQP-like feasibility restoration phase is invoked.

Still a lively field of research!

Filter methods are the subject of active research, but after 11 years it's fair to ask ...

Was the idea of filter methods transformational?

Yes. Fletcher, Leyffer, and Toint received the 2006 Lagrange prize in continuous optimization from the Mathematical Programming Society and SIAM for work on filter methods.

Recent incremental developments have applied filter techniques to interior-point methods, nonlinear equations, nonsmooth optimization, and non-derivative optimization. See *A brief history of filter methods*, R. Fletcher, S. Leyffer, and Ph. Toint (2006), preprint ANL MCS-P1372-0906.

Have filter methods made merit functions obsolete?

No. Researchers are continuing to examine several "old" merit functions, with good results. (Note recycling.)

The third and final example:

Sparse recovery, a statistics problem closely related to optimization.

Consider estimating a p-dimensional parameter β in the linear model

$$y = X\beta + z,$$

where y is an n-vector of observations, X is an $n \times p$ predictor matrix, and z contains stochastic measurement errors.

Motivation: In many statistical applications, the number p of variables or parameters is *much larger* than n, the number of observations.

Examples: imaging, tomography, genomics, signal processing.

This seems initially to be hopeless—how can β be estimated reliably if $p \gg n$??

Recent work has shown, surprisingly, that this may be possible with high probability when β is known to be structured in the sense of being sparse or compressible.

A widely publicized suggestion of Candès and Tao: the *Dantzig selector* (named after George Dantzig), a new estimator that solves a convex ℓ_1 optimization problem that can be recast in various ways as a linear program.

One Dantzig selector formulation:

$$\underset{\beta, r}{\text{minimize}} \, \|\beta\|_1 \;\; \text{subject to} \;\; \|X^T r\|_\infty \leq \lambda, \;\; r = y - X\beta.$$

No time to discuss details!

See, for example:

E. Candès and T. Tao (2004), "Near optimal signal recovery from random projections: universal encoding strategies", *IEEE Transactions on Information Theory* 52, 5406–5425.

E. Candès and T. Tao, "The Dantzig selector: statistical estimation when p is much larger than n", *Annals of Statistics*, to appear.

D. Donoho (2006), "Compressed sensing", *IEEE Transactions on Information Theory* 52, 1289–1306.

Great excitement about this research...

"This work is nothing short of revolutionary; it promises to take the field to a whole new level"
John Cozzens, National Science Foundation

The Candès–Tao paper (which has not even appeared in a journal) has widely been called a "breakthrough", "innovative", "seminal", and "remarkable".

Transformational ideas?

So it would appear!

Clearly superior in efficiency to other approaches, such as basis pursuit denoising?

Not yet clear.

Friedlander and Saunders (2007) show that computation of the Dantzig selector using general-purpose linear programming software can be very expensive.

More time is needed for careful judgment.

The conclusions of this talk are mostly for you to draw, but here are a few thoughts:

- Radically new ideas are crucial, but remember that transformational research can involve recycling in the sense that it applies "old" ideas in an unexpected setting;
- Those who support science should not dismiss or underestimate the power of incremental research (e.g., simplex improvements), especially when done by very smart researchers;
- Transformational research does not necessarily sweep away everything from the past.