Old Stories

Eugene L. Lawler

What do you say when you are asked to write recollections of the 'old days'? How do you respond, when your own career is quite incomplete, and – you devoutly hope – your best work is yet to be done? How do you respond, when – as your wife reminds you – you can't even remember what you had for lunch? You say, yes, of course. Following are a few stories of the 1950's, . . . , 1980's, with a few speculations about the 1990's.

The Fifties

When I began graduate study at Harvard in 1954, I had little purpose and much innocence. I had completed a three-year baccalaureate at a southern university, in the course of which I had taken every undergraduate and post-graduate offering in mathematics in the catalog. But somehow, I had never heard there was such a thing as 'applied' mathematics. And, certainly, I was unprepared for the higher standards of academic competition I was to encounter in graduate school.

I soon decided that abstract algebra and real and complex analysis were not for me, particularly not when Harvard undergraduates were running circles around me in my classes. Sometimes I thumbed through library copies of journals, looking for something more meaningful. I recall once running across a quaint little journal with the title *Operations Research.* I wish I could say that this was a moment of awakening, but my reaction was only one of bafflement. Awakening came later.

For variety, I elected two courses that were polar opposites. The first of these was W. V. O. Quine's course on mathematical logic. Quine had been away on sabbatical at Oxford. His fame, and the pent-up demand for his course, attracted an initial attendance of a hundred or more. Curiously, he seemed unperturbed by the small size of the classroom. But I soon understood why. At the end, only a half-dozen of us were left in the class. I passed, but with little comprehension.

About twenty five years after Quine's course, I happened to meet Garrett Birkhoff at a conference. After I had introduced myself as one of his former students, Birkhoff delighted me with the following story. (Quotation is from memory.)

> "You know, many years ago — back in the 1930's — I thought I was interested in the foundations of mathematics. I even wrote a paper or two in the area. We had a brilliant young logician on our faculty who was making a name for himself. One term I happened to lecture in the same classroom, the hour immediately after he did. Each day, before erasing the blackboard, I would take a look to see what he was up to. Well, in September he started out with Theorem 1. Shortly before Christmas he was up to Theorem 747. You know what it was? 'If $x \leqslant y$ and $y \leqslant z$, then $x \leqslant z$'! At that point, something within me snapped. I said to myself, 'There are some things in mathematics that you just have to take for granted!' And I never again had anything to do with the foundations of mathematics."

My other elective was the maiden offering of a course in 'Automatic Data Processing', taught by Kenneth Iverson with Fred Brooks as one of the TA's. Like most of those taking the course, I hoped to learn something about 'giant brains' that could 'think'. Dick Karp, then a Harvard senior, was one of my classmates. We share the same recollection.

One day Iverson announced that he would lecture on 'tape sorting'. My reaction: "Fine, I know what tapes are. They come on reels. There are lots and lots of reels of tape in the Computation Laboratory. It must be a common event for those reels to become disordered. We're going to learn how to sort those reels of tape and put them in order again. Not a terribly interesting issue perhaps, but a necessary bit of housekeeping if one is going to run a computer installation . . . What's this? He's going to use the computer to sort those reels of tape? Strange! Computers aren't robots! How can they sort reels of tape? This doesn't make any sense at all!" About twenty minutes into the lecture, the light dawned. But by then my mental processes had been completely derailed.

I left graduate school for a few years, trying out law school, the army, and employment at a grinding wheel company — a real, honest-to-gosh job shop. I also acquired a family. When I returned to Harvard in 1958, I knew it was the program in *applied* mathematics at the Computation Laboratory that I wanted to enroll in.

'Commander' Howard Aiken was still in charge of the Comp Lab, though he would shortly take early retirement. The Lab was still turning out volume after volume of tables of mathematical functions (hard as that is to believe in this age of pocket calculators). Aiken took pride in the bound volumes that issued from the Lab. He insisted that technical reports be assembled into leather-bound volumes for presentation to the research sponsor. And he tolerated no errors. Once, when he found Dick Karp had included an errata sheet, Aiken directed his secretary to razor out the offending page from the already bound volumes.

Needless to say, under Aiken's direction the Lab was orderly. And it did have a nice ambiance. The Mark I and Mark IV computers, though quite obsolete, ran 24 hours a day. Each machine was as big as a small house. I enjoyed walking around inside the computers, in order to admire the machinery. The Mark I was much the prettier of the two machines. Great masses of colored wires were neatly bundled into giant cables. And the hundreds of relays made a soothing clickety-clack sound. (The machine had been retrofitted with a relay unit capable of multiplying two 23-decimal digits in seven seconds. I was told that, at its dedication, Aiken declared that the world would never need a faster multiplier.)

My fellow graduate students at the Lab were a capable and interesting lot. I recall that one of them, Fisher Black, was an intrepid participant in experiments with mind-altering drugs, then being conducted in the psychology department under the direction of Professor Timothy Leary. Fisher is today a Wall Street guru, famed for the Black-Scholes model for futures pricing. Another student eventually joined the FBI's Ten Most Wanted List for a brief period. But most of us went on to make our marks in more prosaic ways.

Once again I took a course from Ken Iverson, this time in 'Operations Research'. While I had been away from school, Ken had become obsessed with the invention of a new algorithmic notation. With some grumbling, he led the class through the simplex method, using conventional notation. Then one day he arrived in the lecture hall, with a particularly bright face. "Today, I will show you the *right* way to describe the simplex method!" He then drew a small box on the board, wrote a half-dozen lines of his idiosyncratic notation in the box, and added a single backward arrow to indicate iteration. He stepped back from the board. A few sentences of explanation and he was done. "Questions?" There were none. Only twenty minutes had elapsed, and he was unprepared to any more. "Well," he concluded, "This shows what happens when you have right notation. It takes much less time to explain things than you expect." He then wheeled about and exited the lecture hall a half-hour early. Only years later did I realize that I had been present at the birth of APL.

Iverson's course was my awakening. Here, at last, I found mathematics that was both interesting and useful. I had never worked on combinatorial problems before, but I found they were something I could really *do*. I read some graph theory, at that time a rather obscure specialty, and I liked it. Ralph Gomory gave a lecture at Harvard on his revolutionary integer linear programming algorithm, and I was intrigued. Willard Eastman wrote a PhD thesis on the solution of the traveling salesman problem by a method that later came to be known as branch-and-bound. I applied the same method to the quadratic assignment problem, and that became a major part of my own PhD thesis.

I recall my years at Harvard with great fondness, and will always remember the kindness and encouragement of the young faculty, particularly Willard Eastman, Robin Esch, Ken Iverson and Gerry Salton.

The Sixties

In 1962 I became assistant professor of electrical engineering at the University of Michigan, where I started off by doing research and teaching in switching theory. At the time, Bob Thrall and George Minty were in the Mathematics department, and Art Burks, Bernie Galler, John Holland, and Anatol Rapaport in the Communication Sciences department. Together, they provided a wealth of stimulation. Thrall took me under his wing and generously made me co-director of the annual Michigan summer conference on operations research. We invited an impressive group of people to lecture at these short courses, including such luminaries as Michel Balinski, Richard Bellman, George Dantzig, Jack Edmonds, and Ray Fulkerson.

One evening I was honored to have George Dantzig as a guest at my house. At the time George was in the midst of developing his concept of a 'compact city', later published as a book with Thomas Saaty. Earnestly, he described his ideas for housing millions of people in what sounded like a gigantic parking structure with forty-foot-high ceilings. People would work in shifts, he explained, and all public facilities would be open around the clock. Combinations of moving walkways and elevators would enable anyone to get from any point in the city to any other point in minutes. For anyone who wanted to experience sunlight, there would be an elevator to a grassy park on the roof of the city. During this discourse, my wife became visibly agitated. A sputtered series of objections ensued, and George responded calmly to all of them. Finally, in exasperation my wife demanded, "And what does *your* wife think of this?" With great sweetness, he smiled and replied. "Compared to her," he said, "you are a kind soul!"

George Dantzig and T. C. Hu organized the most rewarding technical meeting I have ever attended. This was a workshop on 'Integer Programming and Network Flows' held at Lake Tahoe in 1965. Here I learned about matroids, nonbipartite matching, the Chinese Postman's problem, polyhedral characterizations of combinatorial problems, and the importance of estimating the computational efficiency of algorithms. I heard of more applications of network flow theory than I had dreamed existed. Jack Edmonds was involved in everything. What he had not created, he had improved. I was dazzled by his brilliance.

Jack's papers were an endless source of inspiration and exasperation. The theorems were wondrous, but many proofs were inscrutably succinct or seemed entirely lacking. Algorithms were implied, but were far from explicit. Gradually I came to appreciate that bipartite matching was simply a very special case of matroid intersection, and to see that matroid intersection algorithms could be modeled after classic matching and assignment algorithms. When I described such a matroid intersection algorithm in a conference paper, Jack was not amused. "It's all there in my papers!" he insisted. I am sure he was right.

The summer of 1966 I made my first trip to Europe to give a paper at a graph theory meeting in Rome. When I read the program, I was much surprised to find that Dantzig and I were to give papers in the same session on exactly the same problem — the computation of optimal ratio cycles in graphs. But I was also much relieved to find that our solutions to the problem were entirely different: Dantzig employed a variant of the simplex method, whereas I used shortest path computations as the basis for bisection search.

I felt that my contribution to ratio-cycle problem was quite modest. (Years later, Nimrod Meggido provided a vastly more sophisticated and elegant solution, also based on bisection search.) But apparently my paper attracted some attention. A couple of years after the Rome meeting, I was invited to participate in an Italian conference on periodic control. I was dumbfounded. "I don't even know what 'periodic control' is," I demurred. "We want you to talk about the ratio cycle problem," I was told. "Your Rome paper is *the* fundamental result for the discrete case." Naturally, I accepted the invitation to speak, happy that my modest contribution had earned me not one, but two, European junkets.

The academic year 1968-69 was my first sabbatical. I spent it in Berkeley, where I began serious work on my book on combinatorial optimization. Little did I know that this was the beginning of a seven-year ordeal; *Combinatorial Optimization: Networks and Matroids* would not be published until 1976.

Two particularly notable events occurred in the spring semester of 1969, one political and the other technical. Governor Ronald Reagan ordered a military occupation of Berkeley by 2,000 National Guard troops. I became a member of the Berkeley 483, a group of folks arrested *en masse* for the heinous offenses of illegal assembly, blocking a public sidewalk, and creating a public nuisance. I have no documentary evidence of my arrest, because all charges against the 483 were eventually dropped and arrest records

Edmonds, Lawler (Banff, 1977)

expunged. But Dick Karp can bear witness to my brief career as a political criminal; he accompanied my wife when she bailed me out of jail in the morning.

The technical event of the spring of 1969 was Dick Karp's discovery of the theory of *NP*-completeness. Dick commented at the time that the hard part had been done by Steve Cook. Yet it is truly awesome that Dick was able to perceive the significance of Cook's work, and to apply Cook's theorem as he did. I became a handmaiden, contributing some *NP*-completeness proofs for Dick's landmark 1971 paper.

The Seventies

The new decade began with mixed signals. I joined the Berkeley faculty in Electrical Engineering and Computer Sciences. I no longer had to contend with the malorganization of Computer Science at Michigan, and that was good. Berkeley had its own CS turf war, but it was soon resolved, and that was good. I had a war in my own household, and that was very bad.

The new theory of *NP*-completeness had profound implications for both the CS and OR communities. It spawned much new research into complexity theory, including the study of the polynomial hierarchy, *P*space-completeness, and provably exponential problems. Even the purest of mathematicians came to appreciate that there were issues of intellectual substance to be found in theoretical computer science. As a practical matter, we now had a useful new tool to guide us in the development of algorithms for combinatorial problems. However, the Operations Research community was slow to grasp the implications of the new theory. For a period of two or three years, I made a mini-career of visiting ORSA-TIMS meetings, spreading the gospel. I even organized a rump session at Bernard Roy's 1974 Versailles symposium on Combinatorial Programming. Unfortunately, I gave a rather lame presentation and probably won no converts.

However, two Dutchmen at the Versailles meeting were already converts. Jan Karel Lenstra and Alexander Rinnooy Kan knew *NP*-completeness. Even more impressive, they knew my own work on scheduling theory. I sensed immediately that these eager young graduate students were winners! They easily coaxed me into being a thesis examiner for each of them. This resulted in many visits to Amsterdam, and the beginning of a long and fruitful collaboration. Best of all, Jan Karel and Alex introduced me to their circle of Amsterdam friends, and within this circle I found a new wife.

Jan Karel occupied a dingy office in the old Mathematisch Centrum, adjacent to the Amstel brewery. Amid pungent odors of malt and hops, we set about revising the classification system for scheduling problems devised by Conway, Maxwell and Miller. Soon our technical conversations were laced with shorthand. "We know the status of one-deejay-sum-ceejay and one-arejay-sum-ceejay," I would say, "but what about one-preemption-arejay-deejay-sum-ceejay?" I couldn't speak Dutch, but I *could* speak a new argot!

Jan Karel decided to make Alex an unusual gift on the occasion of his thesis defense: a computer-tabulated listing of the thousands of types scheduling problems we had classified, with each problem annotated with its complexity status. Ben Lageweg, a master programmer, provided a listing. We then observed that it would be amusing to generate a sublisting of the maximal problems solvable in polynomial-time, the minimal *NP*-hard problems, and the minimal and maximal open problems. When Ben obliged with this new listing, we were enthralled with the results. There were numerous open problems that we had previously overlooked. In the next few days, a group of us — Jan Karel, Ben, Dick Karp and I — had the pleasure of knocking off one obscure problem type after another in rapid succession. Alex, of course, was denied the satisfaction of licking this cream. After all, we were preparing a surprise for him!

About this time I made an innocent suggestion to Alex and Jan Karel. "Why don't we write a book on scheduling theory?" I asked. "We have your two theses and my own papers. We can just get out the scissors and pastepot. It shouldn't take much time!" Book-writing, like war-making, often entails grave miscalculations. Fifteen years later, *Scheduling* is still unfinished. (But, with David Shmoys as an added coauthor, our blockbuster *will* be published in 1992.)

The decade of Seventies ended with my once again playing the role of handmaiden to a significant discovery. In January 1979, Rainer Burkard brought the reprint of a *Doklady* paper to Oberwolfach. The author, somebody named Khachian, purported to have a polynomial-time algorithm for solving linear programming problems. The next week in Amsterdam, sitting at the elbow of Milan Vlach, I scribbled out a translation. I sent this to a long list of people, and before too long, Peter Gács and László Lovász had supplied Khachian's missing proofs and verified his results. Then the uproar began. One thing led to another, and before long newspapers the world over were acclaiming the cataclysmic result: the traveling salesman problem had been solved! I gained a brief moment of notoriety when the *New York Times* cited me on its front page as the Discoverer of the Discoverer. (For those interested in more details of this bizarre episode, I will be happy to send a reprint of my article, 'The Great Mathematical Sputnik of 1979'.)

The Eighties

The Eighties began with a spectacular enrollment boom in computer science. I found myself presiding over an undergraduate program that was mushrooming all out of proportion; it seemed that every Berkeley undergraduate desperately wanted into the CS major. Almost daily, for three years, students left my office in tears. I worried that someday a rejected student would head straight for the Campanile, and following a quick elevator ride, end his life – splat – beneath my window. I even had a call from a US Senator who tried to intervene on behalf of his daughter-in-law. (I listened attentively to the senator, assured him I would 'give serious consideration' to his plea, and rejected her.)

Two or three times, I taught a special topics course in Sequencing and Scheduling. I've always found it hard to force myself to prepare lectures on my own work, so I often got stuck on details. But the students seemed to get the idea anyway. At least two, Barbara Simons and Chip Martel, wrote doctoral theses on open problems I posed in lecture.

A student's first significant research result is always a treasure. A student's first explanation of the same result is something else again. When Chip Martel first dirtied my blackboard with his algorithm for cue-preemption-arejay-deejay-ellmax, I could make neither head nor tails of his ideas. But after much yammering and yakking, we massaged his procedure into a 'polymatroidal' network flow algorithm, thereby transforming a small treasure into a much greater one. The polymatroidal network flow model we developed provides the true *right* way to formulate and solve matroid and polymatroid optimization problems – or so I maintain. (Later, we learned that Chip's scheduling problem can be solved by ordinary network flow techniques. But this is an irrelevant irony. More humbling is the fact that, in the real world, there is about as much of a market for polymatroid optimization algorithms as there is for theorems in mathematical logic.)

During the 1980's, I was blessed with a number of other outstanding doctoral students. David Shmoys co-edited, with Alex and Jan Karel and myself, the highly successful *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Marshal Bern collaborated with me in the development of a theory of linear-time algorithms for decomposable graphs. Arvind Raghunathan and Huzur Saran, together with Rajeev Motwani, obtained brilliant results on the linkless and knotless embedding of graphs in 3-space.

Other Berkeley students were responsible for two of the most significant developments in mathematical programming in the 1980's. Narendra Karmarkar began work on interior point methods while completing a thesis on another topic under Dick Karp's supervision. Andrew Goldberg, after enrolling at Berkeley for a period, returned to MIT to invent his marvelous new network flow algorithms.

I think there is a moral to be drawn from the work of Karmarkar and Goldberg: No matter how worked-over an area may be, there may still be gold to be mined. Interior point algorithms are not new, and penalty-function computations date back at least to the 1960's. But Karmarkar's work has demonstrated that there is much to be gained by resuscitating these old and discarded methodologies. Numerous very capable investigators have worked on network flow algorithms over the years. Yet Goldberg was able to achieve a breakthrough.

The Nineties

As the 1990's begin to unfold, I find myself turning to computational problems in biology, specifically to the analysis of DNA and protein sequences and the reconstruction of evolutionary histories from molecular data. I believe that in the 1990's, biology will provide a fruitful area of application for combinatorial optimization, and for mathematical programming as a whole. (Perhaps someone in the mathematical programming community will unlock the secret of protein folding?)

We are now able to solve combinatorial problems of a size that would have seemed almost inconceivable even ten years ago. For example, it is now possible to obtain exact or near-optimal solutions to traveling salesman instances with thousands of cities. As the ratio of computing power per dollar increases (at the rate of about 30% a year), the size of solvable problem instances will, of course, continue to increase. However, as parallel architectures become commonplace, it will become increasingly apparent that memory space, not processor cycles, is the bottleneck. It will become increasingly important for algorithm designers to understand the problems of coordination and synchronization of parallel processes.

Software systems for mathematical programming will become more adaptive, i.e., more 'intelligent', in that these systems will be enabled to make selections from a suite of algorithmic possibilities. We will learn how to design systems that are 'meta-algorithmic', in the sense that they will employ algorithms to design algorithms. Very large collections of data will become readily accessible, either over high speed networks or in the form of CD ROM's, giving mathematical programmers a greatly enhanced opportunity to experiment with realistic data. Visualization of data will become an increasingly important issue, particularly with the advent of a new generation of flat color displays, which is likely to occur about five to seven years from now.

I claim these speculations are neither particularly original nor insightful. To the extent the reader finds them mundane, the reader must accept that certain conclusions are inevitable. Algorithm developers will need to be increasingly computer-sophisticated. Academic departments of Operations Research will have to find ways to dove-tail their programs with Computer Science, else they must remain a step behind, and resign themselves to teaching the applications of software that is developed elsewhere.

A final note, on the $P = ?NP$ question. In 1977, I bet Alex a case of fine champagne that the issue would not be resolved by 1989. I predict that the Decade and the Century will end without resolution of the issue. Sad to say, but it will be many more years, if ever, before we really understand the Mystical Power of Twoness (or what Jan Karel calls the Magical Power of Threeness): 2-SAT is easy, 3-SAT is hard, 2-dimensional matching is easy, 3-dimensional matching is hard... Why? Oh, why?