International Series in
Operations Research & Management Science

Allen Holder *Editor*

Harvey J. Greenberg

A Legacy Bridging Operations Research and Computing

Springer

International Series in Operations Research
& Management Science

Volume 295

More information about this series at http://www.springer.com/series/6161

Allen Holder
Editor

Harvey J. Greenberg

A Legacy Bridging Operations Research and Computing

Springer

Editor
Allen Holder
Department of Mathematics
Rose-Hulman Institute of Technology
Terre Haute, IN, USA

ISSN 0884-8289 ISSN 2214-7934 (electronic)
International Series in Operations Research & Management Science
ISBN 978-3-030-56428-5 ISBN 978-3-030-56429-2 (eBook)
https://doi.org/10.1007/978-3-030-56429-2

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Contents

Harvey J. Greenberg

Chapter 1
A Commemorative Review of Harvey Greenberg's Career

Allen Holder

Harvey J. Greenberg's career in Operations Research (OR) and Computer Science (CS) spanned the half-century from his Ph.D. dissertation in 1968 to his death in 2018. The magnitude of his accomplishments in those 50 years is profound, enough so that it is difficult to communicate his legacy's entire imprint. Harvey had outstanding academic success, but he also played critical roles in guiding public policy, in advancing OR's industrial employ, and in building community. Most OR professionals could pridefully review their careers with noteworthy success in just one of these categories, but Harvey's substantive influence in each has distinguished him as one of his era's definitive bellwethers. Harvey's era was of particular note because it witnessed OR blossom from its military and industrial origins to its expansive embrace of modern algorithms and computing. These computational advances now solve problems across a broad taxonomy of OR, a taxonomy that is mathematically and computationally diverse and that is replete with application. Harvey adored OR's increase, and he never tired of learning new applications, new methods, and new computing aspects. Harvey's career began in a time when erudite students could canvass OR, but it ended in a time when scholars had to be much more selective. Harvey's generation held the last of OR's renaissance experts, a group of which Harvey was an exemplary representative.

The contributed chapters of this volume intersect several facets of Harvey's career, and they pivot from collegial commentary and review, to concluding research, and to inspired new projects. Topics range from the method of Generalized Lagrange Multipliers, which was one of Harvey's most cherished and sustained research topics, to a new effort in Computational Biology, which was an emerging area of research that Harvey championed toward the end of his career. Another chapter reflects on Harvey's important work at the Federal Energy Administration

A. Holder (✉)
Department of Mathematics, Rose–Hulman Institute of Technology, Terre Haute, IN, USA
e-mail: holder@rose-hulman.edu

A. Holder (ed.), *Harvey J. Greenberg*, International Series in Operations Research & Management Science 295, https://doi.org/10.1007/978-3-030-56429-2_1

(FEA), and it thankfully archives one of OR's most unsung success stories. Harvey's disposition regularly became impassioned as he discussed his time at the FEA, and his work there was very clearly the perfect blend of research, application, social charge, and camaraderie. I envied Harvey's zeal as he narrated events at the FEA, and I have had more than twinges of jealously as I have realized that my career will surely lack such experiences.

This collected volume illustrates the mosaic of Harvey's career as it skips across time and topic, but we should also comment on the very many accomplishments not discussed. The remaining goal of this introduction is to do just that. The majority of what follows is factual and can be readily verified, but I also comment occasionally on my memories. I do not intend to spin tall tales, but folklore has, nonetheless, a way of amplifying itself into something grander than it really was. I ask for indulgence as I remember my dear friend, someone who could mythically fathom the infeasibilities of vexing problems as he rallied a posse of OR and CS experts to efface the world of slipshod analysis. Harvey did not just study and advance OR and CS because he enjoyed it, which he did immensely, but rather because he also believed in the utility of OR and CS and how they could improve peoples' lives. His passion was infectious and motivating, and it will be missed.

1.1 Research Prowess

Harvey earned his Ph.D. from the Department of Operations Research and Industrial Engineering at Johns Hopkins University in 1968. His dissertation title was *Optimal Attack of a Command and Control Communications Network*, although this was not his first Ph.D. dissertation. Harvey's original thesis topic had instead stemmed from his penchant for Lagrange multipliers, and in his first dissertation he independently established the Fritz John optimality conditions. These conditions appeared in 1948 in *Studies and Essays, Courant Anniversary Volume* [3], but they were not widely distributed or read in the emerging operations research community. However, several researchers at the time were investigating constraint qualifications to advance the theory of Lagrange multipliers, and the Fritz John conditions were natural and important in this line of research. Indeed, Mangasarian and Fromovitz published a new constraint qualification in their 1967 article titled *The Fritz John Necessary Optimality Conditions in the Presence of Equality and Inequality Constraints* [6], and they acknowledged in their introduction that the Kuhn–Tucker criteria were "best-known" even though the Fritz John criteria were more general and previously published. Harvey found himself in the awkward situation of having a significant independent result, one that would have cemented him as a rising academic star, but that was insufficient as original research. Harvey had already accepted a new job and had released his graduate stipend when he learned the news, and he found it difficult to return to student life and to write a second dissertation. Each Ph.D. student is acquainted with the fear of finding her or his expectant thesis in the latest literature or in an obscure or overlooked publication, but Harvey lived that fear. This difficult

experience gave him a unique empathy with his students. Indeed, he advocated that such occurrences proved that the research was substantial and of publishable quality, and if truly independent, then it was also sufficiently novel.

Harvey's academic credentials burgeoned after he earned his Ph.D., and his early résumé was a young academic's paragon. His first 23 publications included 10 articles in *Operations Research* (5 sole authored), 2 in *Management Science*, 2 in *Journal of Optimization Theory and Applications*, 1 in *Mathematical Programming*, and 1 (sole authored) in *Technometrics*. These early publications have over 1000 citations and an average of more than 45 citations per paper. Twelve of Harvey's articles have at least 100 citations, averaging 187 citations per article in this group, and 27 papers have at least 50 citations, averaging 118 citations in this group. These highly cited articles stretch from his earliest research, for example, *Surrogate Mathematical Programming* with W. Pierskalla in 1970 has 275 citations [2], to the end of his career, for which *Reconstruction and Functional Characterization of the Human Mitochondrial Metabolic Network Based on Proteomic and Biochemical Data* with T. Vo and B. Palsson in 2004 has 176 citations [8].

Harvey's publication record was as diverse as it was significant, and his expertise went deep into any taxonomy of OR and CS. His publications intersect both sides of our standard divisions: continuous versus discrete, deterministic versus stochastic, convex versus nonconvex, applied versus theoretical, etc. I sat with Harvey through dozens of talks and seminars, and I was both amazed and intimidated by his unremitting and penetrating questions, which came independent of topic. Speakers, from nervous fledgling students making their initial research overtures all the way to seasoned and steadfast professionals delivering well-tested presentations, often found themselves learning more about their research than what they had been offering during their talks. Harvey's research latitude was motivated, at least in part, by his ability to quickly perceive how an emerging idea could advance another domain. For instance, he had longstanding interests in parametric programming and sensitivity analysis, which until the 1990s had largely been reliant on simplex algorithms. Harvey immediately altered his perspective once he saw that the then emerging interior-point algorithms provided qualitatively different solutions. Another example was his early awareness of how OR and CS could further problems in the life sciences. Harvey had to learn significant amounts of new material to leverage these connections toward new research, but he never tired of the effort—indeed, I do not think he thought of it as an effort at all. He constantly had piles and piles of papers nearby, and while his filing system was sketchy, his memory was lucidly lexicographic.

Harvey's research accomplishments earned him several accolades. He was awarded the 1999 Harold Larnder Prize by the Canadian Operations Research Society for having "achieved international distinction in Operations Research," and he won the Operations Research Society of America's research prize presented by the Computer Science Special Interest Group in 1986. He further received the University of Colorado at Denver's Chancellor's Lectureship Award for Outstanding Scholarship in 1993 and the College of Liberal Arts and Sciences Award for

Outstanding Achievement in Research in 1988. He became an INFORMS Fellow in 2011.

Harvey's academic credentials are without doubt impressive, especially for his era, but they are even more impressive against the reality that Harvey spent many of his prime academic years in public service. Moreover, Harvey only had a handful of Ph.D. students, four in mathematics and one in computer science. So his academic output was not the natural, if not perfunctory, expectation of a modern academic's research program, but it was instead the derivative of a pursuant intellect that relentlessly sought challenging problems. Those of us fortunate to have worked with Harvey experienced his uncanny industry, adroitness, and alacrity, all of which have become legendary and, at times, even humorous. He was a joy to work with, but you had to be ready for a late night call if he discovered a new result, or an impromptu polemic if an outcome was in question. Conversations could be heated, but only because the truth would be better honed by the fire of debate. Harvey loved banter, especially when it ended in the celebration of a mathematical novelty, a fresh computational perspective, or an original analysis. Some might say that Harvey enjoyed argument, but I do not think that that hits the nail on the head. Harvey certainly relished the back and forth, which from the receiving end always seemed more back than forth, but what he really fancied was the combined effort of identifying what was being sought. It was as if each person's parlances were their individual pickaxes, and the argument was a way for everyone to swing at the rock that held the gem. Harvey could swing harder than most, and his rapid and accurate blows were something to behold.

1.2 Pedagogical Imprint

The educational standing of a research academic is regularly assessed within the realm of graduate education and Ph.D. advising, but Harvey's pedagogical imprint differs from this standard and is, in the author's opinion, more lasting and altruistic. Harvey taught at three universities and made substantive programmatic changes at each. He also worked tirelessly to initiate online and free educational materials that have continued to aid students of OR and CS worldwide.

Harvey's first academic position was in the Department of Computer Science and Operations Research at Southern Methodist University, where he helped launch a new Ph.D. degree in Computer Science and Operations Research, a new M.S. degree in Engineering Administration, and a new program that introduced undergraduates to research. The latter of these was way ahead of its time and came when the term "undergraduate research" would have been an oxymoron to most academics. His second academic position was at the Virginia Polytechnic Institute and State University, where he directed the off-campus graduate program in Computer Science.

Harvey left Virginia Tech. for the Federal Energy Administration in 1976, a time upon which we reflect in the next section, and he then joined the mathematics

department at the University of Colorado at Denver from 1983 to 2008. Part of the draw to CU-Denver was its goal to initiate a doctoral degree in applied mathematics. Harvey chaired the proposal committee, which successfully installed the new degree. The field of mathematics was then at the height of its division between its pure and applied factions, and the applied side was further dominated by the field of partial differential equations (PDEs). Harvey's OR and CS perspective on applied mathematics was much broader than PDEs, and the new degree included discrete mathematics, OR, Probability, Statistics, and PDEs. The research and educational purview of the new Ph.D. also welcomed computational studies, which were largely shunned by pure mathematicians of the day. The CU-Denver math department grew around these broad guidelines, and its liberal scope of applied mathematics helped distinguish it from other graduate programs. The first Ph.D. was awarded in 1988, and the department surpassed its 100th doctoral degree in 2019.

Harvey also helped start the CU-Denver Math Clinic, the Center for Computational Mathematics, and the Center for Computational Biology (CCB). The Math Clinic provided practical and industrial research opportunities to graduate and undergraduate students, and it was an uncommon educational experience within mathematics—Harvey Mudd's math clinic was the only other example known by the author at that time. The CCB deserves special comment since it stemmed from Harvey's early awareness that the life sciences were becoming increasingly dependent on mathematics and computing. He founded the CCB in 2001, a few years before the National Academy of Sciences' BIO2010 report definitively recognized the importance of mathematics and computing within the biological curricula. The CCB dissolved in 2009, but it anticipated the sweeping changes in the life sciences that were then happening, and it promoted education and research in the emerging disciplines of computational biology, bioinformatics, and systems biology.

The educational programs Harvey helped initiate demonstrate his particular gift to start something new. Initiating such projects requires an aspirational energy, an energy that Harvey brought regularly to his other professional responsibilities like classroom education. Being a student of Harvey's was not for the faint of heart, and he could overwhelm with content, expectation, and pace. The author's first day as a pupil in one of Harvey's courses started with Harvey wheeling a cart around the classroom to deposit foot high stacks of papers in front of his students. He pulled a thick, originally authored manuscript from the top and announced that we would complete it by the next class, and he then told each of us to go to the board and state and prove a theorem of our choice from the prerequisite course. It was a long weekend of studying motivated by our ineptness at the board. Harvey often used dual projectors to cover all that he wanted within a class period—he just moved so very quickly. A classmate found a flaw in the course materials mid-term, and in our barbed exhaustion we so anticipated pointing this out when we reached it in class. I can still hear the exchange as my classmate interjected, "Professor Greenberg, I think there is an error." There was a halting silence, and then Harvey started to attack the concern, debating with himself for a few moments. He soon concluded that my classmate was correct and that he had found a flaw in a long-standing published

result. Harvey giggled and just kept going. That moment somehow gave us hope, hope that we could survive and gain what he expected.

Harvey was quick to notice educational voids that lent themselves to his expertise, and he persistently toiled to fill them so that others might benefit. Three such projects deserve special comment, those being his *LP Short Course*, *The Mathematical Programming Glossary* [7], and *Myths and Counterexamples in Mathematical Programming* [1]. These were considerable efforts that required protracted dedication, efforts that most academics would have avoided because they would have lacked professional recognition. Harvey completed these projects because they were laudable in and of themselves, and he completed them knowing that his career might not benefit. That is not to say that he did not hope for, or indeed maybe covet, recognition for his altruistic exertions, but conceit would never forestall Harvey from doing what he knew to be right. He held, and lived by, deep-seated convictions, and it was imperative for him to be conscientious. Indeed, these educational enterprises only illustrate his general willingness and aptitude to work where others would not and to work toward the betterment of the greater good.

Harvey authored a curriculum for linear programming that was designed as an online educational resource. The course was called an *LP Short Course*, and he sponsored it on his web page for anyone interested in learning linear programming. This was decades before the massive open online course (MOOC) concept, the Khan Academy, or any of the other free educational outlets available today. The course had nine lessons titled: (1) What is LP? (2) What is a solution? (3) How do we solve linear programs? (4) What do the solutions mean? (5) More formulation and analysis, (6) Mathematics of LP, (7) Computer Science of LP, (8) Economics of LP, and (9) Debugging. The lessons included definitions, examples, and exercises, all of which worked interactively through a web browser. The course also incorporated the use of software as it motivated several practical problems. The course was, in hindsight, a foretelling archetype of what online education would become.

The *Mathematical Programming Glossary* (MPG) grew out of Harvey's desire to help students learn the language of mathematical programming. His first version was a simple list of terms covering linear and nonlinear programming, a version which the author used as a student in Harvey's courses. The educational advantage of hyperlinking terms sparked Harvey's interest and motivated him to extend the glossary well beyond its humble start. The MPG now contains over 800 terms and covers all areas of mathematical programming and their connections to computer science. The INFORMS Computing Society has sponsored the MPG since 2006, and it has become a highly visited resource within the INFORMS online presence. The MPG underwent a major overhaul in 2009–2010, and the glossary now permits customized word lists and supports standard mathematical notation. The MPG is currently on its third editor and is well positioned to maintain its original educational intent for the OR and CS community.

Harvey began cataloging folkloric concepts and esoteric examples in mathematical programming in 1996, and his collection grew over the next 14 years into a substantial volume titled *Myths and Counterexamples in Mathematical Programming* (*Myths* for short). Harvey had an apt propensity to discern what really

was versus what was really close, a skill that galvanized his authoring of *Myths*. He was also keenly aware that disciplines such as real analysis had profited from similar collections, and he wanted OR and CS to benefit similarly. The educational merit of a collection like *Myths* is that students best understand the theoretical and practical confines of a theorem, a calculation technique, or any other similar entity by learning how it loses its validity once it escapes its precise boundary. For example, is it true that the simplex method terminates once it reaches an optimal vertex? The author is confident that OR experts would overwhelmingly answer this question in the affirmative even though it is false as demonstrated by the counterexample to Myth 17. *Myths* is a remarkable and valuable compilation containing 47 myths in linear programming, 45 in integer programming, 32 in dynamic programming, 49 in nonlinear programming, 21 in multiple objective programming, and 19 in other problem classes.

I end this section with a noneducational tidbit that further explicates Harvey's gift to find what was amiss. It is difficult to exaggerate the acuteness of his talent in this regard, and while it may have favorably prompted efforts like *Myths*, it was also frustrating. For instance, Harvey regularly and immediately found flaws in software. He could sit in front of a new system and try the one feature that would fail from among the very many that would not. Indeed, his first attempt would frequently pinpoint the odd example upon which that exact feature would fail. Software regularly seemed mercurial, working on occasion but commonly lapsing—although who could really tell when, or under what circumstances, outcomes could be trusted. I was Harvey's system administrator for several years, a job that kept me busier than I would have ever guessed. I remember him asking why a certain command did not work, and he showed me that it did not as I looked over his shoulder. We swapped positions, and I typed the same command. Voilà, it worked! It was as if the computer had learned to read his fingerprints so that it could play tricks on him. I have never been able to reconcile the paradox of Harvey's yin and yang experiences with computing. He produced world class software, but he could barely use a computer at times.

1.3 Governmental Success

Harvey joined the Federal Energy Administration (FEA) in 1976, a time when the country was gripped by the aftermath of the OPEC oil embargo. The next chapter, which is written by his longtime colleague and friend, Fred Murphy, whom he met at the FEA, archives how Harvey and Fred, as well as many others, succeeded in advising the White House to advance public policy. I strongly recommend this chapter to readers interested in the historical importance of OR and CS. We should all feel proud of OR's well-chronicled triumphs, but the fact that OR helped guide the United States out of an energy crises so perilous that it was called the "moral equivalent of war" has been less documented, and subsequently, less heralded. Many young and publicly spirited virtuosos were attracted to the government at that time in

the hope that they could help society. Harvey and Fred brought particular expertise in OR, and they used it to influence policies that have since helped steady our economy.

My goal in this section is to add my perspective on Harvey's time at the FEA. Harvey was never far from his FEA experience in my relationship with him, and conversations of any significant length always meandered through his FEA memories. His years at the FEA were indeed special. It was a moment when Harvey had a young family and when he was answering John F. Kennedy's call to "ask not what your country can do for you—ask what you can do for your country." Harvey so admired President Kennedy, and others like Abraham Lincoln and Albert Einstein, and he felt responsible to help as he could. One of the ways he helped was to support women in the workforce, with Susan Holte, one of his colleagues at the FEA, stating soon after Harvey's death, "this was an era when my salary did not count when my husband and I were applying for a mortgage, and some men in the workplace did not take the women, particularly younger women, seriously. I was fortunate to work with several men who ignored those outmoded ideas and really boosted me and my career. Harvey was definitely one of my main boosters."

Harvey's work at the FEA combined mathematics, computing, and analysis, and there was a necessity for brisk advancement. Harvey relished the excitement of the challenge, and he acknowledged the joy of having Potomac fever. He would smile and reminisce about how he could apprise officials in the morning and then later in the day hear about their decisions on the national news. The bustling clip of work demanded solutions to what were then difficult to solve economic models, but time on a computer was at a premium. I once sat with Harvey and Johannes Bisschop, who was at the World Bank when Harvey was at the FEA, as they recalled an earlier tussle between the FEA and the World Bank about who had precedence on the big mainframes. Calculations were so important at that time that Mayor Daley in Chicago had been asked to stop road work that would have otherwise caused a machine to go down. The FEA thankfully had its own mainframe during Harvey's tenure.

Fulfilling the charge to solve and analyze substantial energy models with limited computing resources gave Harvey a unique research acuity. One nugget he passed to me was that the simplex algorithm could terminate with different solutions depending on when it was started. The reason was that the number of reduced costs computed at each iteration depended on the state of the computing system. So while the algorithm was deterministic, its employ was stochastic, and solutions would vary from run to run. Successful runs would typically follow after the completion of 40–100 learning runs, of which each could take several hours itself. The difficultly to obtain success prompted the need to squeeze all possible information from the various runs, and Harvey began to pursue software to aid such analysis, a pursuit that ultimately led him to author the software packages ANALYZE, MODLER, and RANDMOD. These packages foreshadowed many of our modern software resources, although ANALYZE remains unique as a system designed to answer a practitioner's queries. The software ensemble is Harvey's practical response to the fact that "we can solve far larger problems than we can understand," an oft

repeated phrase that he wrote in 1988 in support of developing an intelligent mathematical programming system. Harvey's software packages are reviewed by Matthew Saltzman in Chap. 3.

Harvey's career largely splits into pre- and post-FEA, and the division highlights the effect that his work at the FEA had on his career. His research pre-FEA was mostly mathematical and algorithmic, focusing on Lagrange multipliers, dynamic programming, and duality theory. His research at the FEA became more practical and more computational, a trend that he maintained after the FEA. Harvey enjoyed theory in my experience, but he always looked for it to translate back into practice. One of his favorite adages was, "there is nothing more practical than a good theorem." He saw theory and abstraction as ways to address particular problems but not generally as means in themselves. My relationship with Harvey was post-FEA, and I believe his focus on solving real problems stemmed from his work at the FEA. In any event, working at the FEA was to him a high point of his career, and the work he accomplished there impressed itself upon the rest of his professional activity.

1.4 Service

Harvey's dedication to advancing OR and CS has had, and will continue to have, profound effects. I have already commented on his willingness to undertake projects like the MPG and *Myths* and on his enthusiasm for starting new educational programs. Here I want to comment on his service to the OR and CS profession beyond these educational elements, and in particular, I want to distinguish his remarkable ability to create community and opportunity.

Harvey was one of the founders of the Computer Science Special Interest Group within the Operations Research Society of America (ORSA), which later became the INFORMS Computing Society (ICS). He was the group's second chairman, and he (co)chaired several conferences and symposia. He also provided a welcoming atmosphere and befriended young talent, encouraging them to consider the intriguing problems in OR and CS. The ICS is one of INFORMS' original and most successful societies, and it has served thousands of OR and CS professionals through its biennial conferences, its journal, its awards, and its research voice, all of which Harvey played foremost roles in launching, managing, and sustaining.

The opportunity to start a new society in OR and CS was born out of what was then an entanglement of the two disciplines. OR was pioneered before the concept of computing became widespread, but the emerging discipline of OR naturally agreed with CS's utility and theoretical study. Harvey was quick to remind us that many of the original computers were debugged by solving optimization problems and that numerous problems in CS were most naturally within our OR purview. He did not really see OR and CS as distinct disciplines but instead as a sort of Janus through which ever greater problems could be solved. Computing could be theoretical, methodological, or operational, and OR could be algorithmic, numeric,

or computational. Time has altered our broad community's perspective on OR and CS, and the interface between the two is often interpreted today as the computational side of the more methodological/theoretical discipline of OR. Harvey held disdain for this restricted ambit, and he was quick to espouse the ICS community as being so much more and as having a potential that was both momentous and noble.

Harvey championed two publication outlets, creating both the *ORSA Journal on Computing*, which later became the *INFORMS Journal on Computing* (IJOC), and *TutORials*. Cole Smith reviews these series in Chap. 5. Many academics serve journals, and Harvey certainly did his share of this type of service, but starting new publication venues and nurturing them from their inaugurations to their mature postures is assiduous. I was fortunate to have served under Harvey as an associate editor of the IJOC, and he taught me much about the earnest responsibility of being an editor. He was not the type of editor who would merely tally referees' comments; no, he would instead actively counsel the review process to help ensure the publication of meaningful novelty.

Harvey further created an industrial consortium in the 1980s in support of developing an Intelligent Mathematical Programming System (IMPS). The goal was to unify and to scientifically study the progression experienced by practitioners of mathematical programming. The IMPS provided software, along with its theoretical underpinnings, that encompassed modeling, solving, and analyzing mathematical (linear) programs. The consortium sponsored regular symposia and supported many academics. Those who participated in the consortium speak of the specialness of the research environment and how it provided career-long research threads. Chapters 2, 3, and 4 review some of the impacts of the IMPS consortium.

Harvey won numerous service awards. He was recognized by the Association of Computing Machinery in 1985 and by ORSA in 1993. He also won an Outstanding Service Award from the College of Liberal Arts and Sciences at CU-Denver in 2001. The ICS created the Harvey J. Greenberg Award for Service in 2007 in recognition of his very many contributions to the society.

Harvey continued to serve the entities that he helped start, e.g., by serving on their advisory boards, as one of their reviewers, or as an editor. He was quick to promote new efforts, and he was willing to spearhead them if allowed. He could pester an editor-in-chief or a society's chair with new ideas, and he had the remarkable knack of sending terse, one line emails that would require paragraphs in return. He was a fountain of new ideas and new opportunities, enough so that it could exhaust those around him. Harvey always seemed to have time to ballyhoo a new option, even if everyone else was sated with those already in place.

1.5 A Constant Challenge to Norms

I hope the previous sections have provided an impression of who Harvey was and how his accomplishments have remained valuable. Harvey was difficult to pin down, and he was constantly thinking outside the box. He had superb vision and was able

to advocate for his causes in ways that made others want to join. Many of his original OR and CS interests were in artificial intelligence and machine learning, topics that have reemerged today as important subjects within OR [4, 5]. These studies were in their infancy when Harvey first espoused their value, and they have flourished as computing has advanced. Harvey was also an original proponent of OR and computational biology, and he dreamed of a time when OR and CS would provide the backbone of personalized medicine, a time when medical treatments would be optimized to an individual's unique biology.

Harvey's constant prodding toward the novel challenged the status quo, and while we have already noted a couple of examples illustrating this fact, I want to add another that he prompted even after his death in 2018. The ICS had already created the Harvey J. Greenberg Award for Service in 2007, and his decisive roles in the ICS's formation and continuation were assuredly meritorious of his name being associated with this service award. However, his research standing and accomplishments were no less important, and several wanted to add a research award in Harvey's name following his death. Doing so would have led, for the first time, to two INFORMS awards being in one person's name, a potential precedent that raised understandable concern. The fact that Harvey, and not one of OR's great early luminaries, was the one who provoked the concern is telling, and the situation is emblematic of how Harvey regularly found himself pushing, or at times rubbing up against, regulation. Indeed, Harvey could occasionally simply work through bureaucracy in creative ways. For instance, he once caused a bodacious stir by expropriating five nice chairs for his staff at the FEA from the room in which the head of the FEA had scheduled a press conference. The chairs were quickly returned to the conference room because Harvey had used his actual name to sign for the dolly.

The concern of having two awards in Harvey's name also reminds me of how Harvey doubted his own legacy. He was confident in his own credit worthiness, but he questioned how others might acknowledge his impact. He was not alone in this regard, and as Richard O'Neill, one of Harvey's colleagues and co-authors, wrote in a commemorative email following Harvey's death, "I do not think Harvey got the recognition or appreciation he deserved." The ICS has always felt to me like a group of crack, cutting edge renegades who were off solving the grand avant-garde problems of the day. This was the group who was designing new algorithms, new mathematics, and new models to advance OR and CS toward the applied benefit of humankind. This was Harvey's renegade clan, and he would have found such heartfelt solace in the knowledge that they wanted to honor him in multiple ways. The brouhaha following the suggestion of a second award was amicably settled with a policy that one person could have at most one award, what I refer to as the Harvey rule. Harvey would have found this new policy to be silly, but I also think he would have giggled knowing that his legacy had caused such a fuss. Harvey would have also been practical and would have supported the ICS's decision to rename the service award so that it could gain the newly formed, and endowed, *ICS Harvey J. Greenberg Research Award*. As John Chinneck stated just before the vote to establish the new award, Harvey would have liked the research award better

himself. The service award retains an inscription that clearly honors Harvey as the motivating influence, so in some way, Harvey is the first to have two awards.

Harvey Greenberg had a wonderful career full of wide ranging accomplishments, but his truly exemplary characteristic was his pursuit of a magnanimous vision for the future, one that held promise and opportunity for everyone and that was motivated by the spirit of good intention. He was a high-energy idealist at heart, even though he could be pragmatic if needed. He was an outstanding academic, a gifted educator, a motivating will, a fatherly adviser, and a tireless voice for OR and CS.

Acknowledgments The author thanks John Chinneck, Leanne Holder, Fred Murphy, Matthew Saltzman, and Cole Smith for their comments on earlier drafts of this memoir.

References

1. H.J. Greenberg, *Myths and Counterexamples in Mathematical Programming*. INFORMS Computing Society (2010). http://glossary.computing.society.informs.org
2. H.J. Greenberg, W.P. Pierskalla, Surrogate mathematical programming. Oper. Res. **18**(5), 924–939 (1970)
3. F. John, Extremum problems with inequalities as side conditions, in *Studies and Essays, Courant Anniversary Volume*, ed. by K. Friedrichs, E. Neugebauer, J. Stoker (Wiley Interscience, New York, 1948), pp. 187–204
4. R. Krishnan, O.R., AI activities expand INFORMS' outreach, impact. ORMS Today **46**(3), 8–9 (2019)
5. R. Krishnan, The Intersection of O.R. and AI. ORMS Today **46**(5), 8–9 (2019)
6. O. Mangasarian, S. Fromovitz, The Fritz John necessary optimality conditions in the presence of equality and inequality constraints. J. Math. Anal. Appl. **17**, 37–47 (1967)
7. J. Sauppe, (ed.), *Mathematical Programming Glossary*. INFORMS Comput. Soc. https://glossary.computing.society.informs.org, 2006–2018. Originally authored by Harvey J. Greenberg, 1999–2006
8. T. Vo, H.J. Greenberg, B.O. Palsson, Reconstruction and functional characterization of the human mitochondrial metabolic network based on proteomic and biochemical data. J. Biol. Chem. **279**(38), 39532–39540 (2004)

Chapter 2
How the Work That Harvey and I Did at the Federal Energy Administration (Later Department of Energy) Shaped Our Research Careers and Led to Our Decades Long Collaboration and Friendship

Frederic H. Murphy

Abstract Harvey and I began working together at the Federal Energy Administration, the predecessor to the Energy Information Administration. Our most intense period was when we were doing research on the fly to model the impacts of policies that were under consideration when the White House Energy Policy Office was developing Carter's National Energy Plan. I describe here the equilibrium modeling and analysis we did to estimate the impacts of the National Energy Plan and other policy proposals. Because we were in untrodden territory, we wound up with decades worth of research questions from that short, intense period. I cover some of these areas and describe some of our work together after leaving Washington, and I mention the current state of the art in these areas and the areas where our research took different paths. I also point out some of the research questions that still need answers.

2.1 Background

I joined the Federal Energy Administration, which became part of the Department of Energy, in 1975, and Harvey came one year later. To put that period in context, after World War II the country was proud of accomplishments that included winning World War II, rebuilding Europe and Japan, and creating a domestic economy with a large middle class. Yet, the country seemed to be descending into chaos. Cities had been burning due to the race and antiwar riots. Lyndon Johnson betrayed our trust in government leaders through his lies and bad decisions about the Vietnam War.

F. H. Murphy (✉)
Professor Emeritus, Fox School of Business, Temple University, Philadelphia, PA, USA
e-mail: fmurphy@temple.edu

A. Holder (ed.), *Harvey J. Greenberg*, International Series in Operations Research & Management Science 295, https://doi.org/10.1007/978-3-030-56429-2_2

Richard Nixon had ruined what remained of our trust through the Watergate break-in, the escalation of the Vietnam War, and his other cynical actions. It seemed at the time that the USA had lost control of its destiny. Then the 1973–1974 oil embargo hit.

Harvey and I joined the federal government because in that era we, and most people, still believed that government could solve problems. We wanted to participate in the solution.

Commentators tend to use political events and social upheavals to demarcate historical and cultural turning points. However, ongoing intellectual and technological progress have an equally important role in shaping eras. Developments in three intellectual fields, along with the political and economic problems of the era, shaped Harvey's and my choices and actions: developments in operations research, economics, and computing. He and I completed our Ph.D.'s just as operations research was becoming an established subject in universities. The field had played an important role in WWII, and soon after it started having major impacts in business. Although the field of economics has been developing for centuries, Samuelson and others set microeconomics into a new direction with his [50] book, *Foundations of Economic Analysis*, by formalizing microeconomics using mathematics. Samuelson's work had an immediate impact on economics research because a core of other economists, such as Kenneth Arrow, Robert Dorfman, Wassily Leontief, Tjalling Koopmans, and Gerard Debreu, were infusing economics with mathematics. Some of these economists were early contributors to operations research. For an accessible summary of the economics literature on how markets do or do not work, see Cassidy [5]. George Dantzig's [10] work on linear programming and the simplex algorithm led to successful applications in many industries [6]. Because of the value of linear programming in optimizing refineries, oil companies underwrote the development of faster LP solvers. Rapid advances in hardware and software meant the largest solvable linear program, Manne [35], went from 50 rows in the early 1950s to thousands by the mid-1970s, turning the early promise of linear programming into a reality.

These three research streams came together with a new reframing of old problems: a systems perspective. Jay Forrester was the first to articulate the systems viewpoint in an industrial setting with his book *Industrial Dynamics* [17], examining how different portions of firms or supply chains of multiple firms can interact well or badly. Systems thinking became a way of framing problems in science, economics, and public policy, e.g. Churchman [8] and Ackoff [1]. The ability to combine systems thinking, OR, and economics in the 1970s was a direct result of research that linked the theory of economic equilibria and linear programming, showing that the solution to a linear program is also an economic equilibrium, Samuelson [51] and Enke [13].

2.2 The Energy Crisis Years

In 1972 the Department of Interior published a study of energy that concluded there would be a "gap" between the declining supply of crude oil in the USA and the growing demand for oil products, and that gap would be filled with coal, see Dupree and West [12]. Despite the gap, oil prices would remain stable. This was a then-standard study that used "judgment" and accounting tables but failed to include the new methodologies of equilibrium economics and computing. As a consequence, the USA and the rest of the world were unprepared when Arab nations were able to raise the oil price successfully through an oil embargo they imposed in response to the 1973 Arab–Israeli war. The embargo demarcated an evolving shift in market power from the oil companies and the Texas Railroad Commission, which controlled oil production rates in Texas, to OPEC producers.

The worst crises a country faces usually are a confluence of multiple events and badly conceived policies. The USA was already experiencing serious price inflation, due to the unbalanced federal budget that resulted from overspending on the Vietnam War without a corresponding tax increase. To lower the inflation rate, Richard Nixon, who described himself as "running the country," imposed a freeze on wages and prices. This freeze meant that the economy had no flexibility to adjust to the subsequent oil-price shock through normal market mechanisms. The other contribution Nixon made to aggravating the energy problem was that he defined it as an engineering problem when he said in his "Address to the Nation About Policies To Deal With the Energy Shortages," Nov. 7, 1973, (http://www.presidency.ucsb.edu/ws/?pid=4034).

> Let us unite in committing the resources of this Nation to a major new endeavor, an endeavor that in this Bicentennial Era we can appropriately call "Project Independence." Let us set as our national goal, in the spirit of Apollo, with the determination of the Manhattan Project, that by the end of this decade we will have developed the potential to meet our own energy needs without depending on any foreign energy sources.

With that statement Nixon mischaracterized a fundamentally microeconomic problem, due in good part to misguided energy policies then in place, as a technology problem. The rhetoric, however, suited a Washington where the leadership in the administration and Congress had lived through the Depression, a massive failure of markets, and WWII, a huge success of the government in marshaling resources to win the largest war the world had ever seen.

As is common when there are no existing institutions charged with solving a problem and a new agency is formed, a more flexible younger generation of talent takes on the challenge. The group at the Federal Energy Office, later the Federal Energy Administration (FEA), followed that pattern.

Bill Hogan, then a young Air Force officer, became the force behind the government's energy modeling and policy analysis. He conceived of and led the first two rounds of development of the Project Independence Evaluation System (PIES), which was a large-for-the-times economic equilibrium model that incorporated all key sectors of the energy system. He dealt with all of the issues of data creation, as

the coherent data on energy was minimal; model development; and the production of studies of energy policies under consideration. For partial equilibrium models that could not be represented as linear programs, he developed an iterative algorithm that involved solving sequences of linear programs for finding economic equilibria. That algorithm remained the state of the art for many years. The Project Independence Report was published in [15]. The next iteration of PIES was used for the 1976 National Energy Outlook [16]. PIES started to make the transition from a scoping model that estimated future energy balances into one that evaluated government policies, including the impacts of the myriad regulations that affected the energy economy. This second version of the model could handle both the pricing rules under electricity regulation, basically average-cost pricing, and the effects of the oil price controls imposed by Nixon. It did not, however, capture the distortions due to government regulations of natural gas markets.

I cannot overstate the impact of Bill on how policies are assessed today. PIES was the first multi-sector microeconomic policy model used in major policy analyses that helped shape legislative choices. Its impact was so powerful that Congress saw that it lost power to the administration when it came to assessing policies. Congress then passed a law requiring that PIES and its successors had to be audited. The law required that Congress has access to the models and could request its own policy analyses.

Under Jimmy Carter the Federal Energy Administration became the Energy Information Administration (EIA) of the Department of Energy. Congress and EIA worked out a process for EIA to do studies for Congress with Congress's own assumptions that continues to this day.

As with all new organizations, turnover was high. Hogan and most of the original team either left FEA or moved to less demanding areas of the agency. I joined in 1975 after the Project Independence Report and did my teething on the 1976 National Energy Outlook. Harvey joined as head of the core PIES equilibrium model after the National Energy Outlook, and I worked for Harvey.

In 1976 the country elected an outsider, Jimmy Carter, as president. Nixon was the last New Deal president, where the president presumed to "run the country," including the economy, and Carter was the first post New Deal president who saw that large segments of the economy were hindered by government regulation and that the latest developments in microeconomics should inform the necessary policymaking for revising detrimental regulations, something for which he is given little credit.

Carter brought in Alfred Kahn, the most notable regulatory economist of the time, to restructure the regulations governing many major industries. Carter deregulated energy markets, trucking, airlines, and railroads. He was the deregulation president, not Ronald Reagan. A consequence of Carter's elimination of burdensome regulations was that railroads stopped going bankrupt and current business-class airfares match, in real dollar terms, the economy fares of the 1970s. Even more importantly, when oil and natural gas prices soared in the 2000s, the economy did not experience the inflation of the 1970s. Prices in the general

economy remained stable because he gave the economy the necessary flexibility to adjust.

Carter's White House Energy Policy Office was staffed by EPA regulators who preferred rulemaking to markets. Because this Office was part of a Democratic administration, its staff did not trust FEA, an agency that was set up by Republicans and was market focused. Consequently, they initially contracted with a firm called ICF to analyze the impacts of their policies. The President of ICF at the time, William Stitt, had a close and supportive working relationship with the PIES modelers, starting with the work on Project Independence and insisted that the White House Office uses the PIES model to ensure the numbers added up when estimating the impacts of policies. The recently appointed head of data and analysis at FEA did not want to participate because he saw working on the project as a lose–lose proposition. David Nissen, the head of the analysis activities, fought to participate, and the new head gave his permission for us to participate but distanced himself to avoid the consequences of any fallout. A combination of Bill, the PIES modelers, and Kahn's activities moved the White House group from taking classic regulatory stances to recognizing that markets matter. Carter's National Energy Plan was the result of the extended interactions among the PIES modelers, ICF, and the White House Energy Policy Office.

After wrenching debates and extensive negotiations between the Administration and Congress, several laws were passed that shifted the sectors away from regulatory mandates to market-oriented policies.

A book is necessary to describe how Carter could change the structure of so much of the economy while dealing with a Congress filled with people who came of age in the Depression, the biggest market failure in history. The key reasons for his administration's legislative success in energy were as follows:

- The country was in pain, coping with shortages of natural gas, because the Supreme Court forced an untenable regulatory structure on gas markets out of ignorance,
- Citizens lived in long lines at gasoline stations due to the price controls on oil and gas-station margins imposed by Nixon.

Outside of energy, Carter was able to deregulate railroads because they were constantly going bankrupt and airlines because some of the carriers were despised by their customers and airfares were quite high. There was an organization called WHEALS, which stood for We Hate Eastern Airlines because of its monopoly position and associated bad service in certain markets on the Eastern Seaboard. Few people remember how abusive regulated monopolies could be. See Sanders [52] for an insightful discussion of the politics of deregulating natural gas.

Energy and environmental issues are now tribal, and staying within a tribe is more important than addressing the facts, which means that sensible strategies on global warming will be difficult to achieve. Sadly, sometimes memories are short. President Obama periodically talked about energy independence using the false rhetoric of Nixon. It may seem odd for Obama to quote Nixon. However, Nixon was a New Deal president, continuing in the tradition of Franklin Roosevelt. Carter was not a

New Dealer and brought in the latest economic thinking, probably because he ran a small business as a peanut farmer and had firsthand knowledge of how markets worked. Obama was a product of the urban politics of Chicago, never worked in the private sector, and brought from his urban experience a New Deal outlook.

2.3 The Work the PIES Team Was Doing

Harvey and his boss, David Nissen, spent a lot of time working with the White House Energy Policy Office to make sure their policies made economic sense and translating their policies into something that could be implemented in PIES. I used to say, "If we can't model it, the policy will not work in practice." Dave and Harvey latched onto this statement. It is actually a serious statement and not arrogance, because if a policy cannot be modeled in the abstract, then the implementation rules would be too complex for an effective implementation of the policy.

Harvey was making major contributions to the computational aspects of PIES and policy representations in the electricity sector on top of his managerial activities. My contribution to the analysis of Carter's National Energy Plan included developing the representations and algorithms to model regulated natural gas markets and policies impacting this sector, see Murphy et al. [43]. Whenever the model did not solve, I piled through row and column listings and the solution file to figure out why the model was either infeasible or unbounded. When it did solve, I would figure out the underlying economic rationale for why the results came out the way they did. I also dealt with some of the people in the White House Energy Policy Office.

I would periodically construct what we called "walk backs." These were estimates of policy impacts using predetermined sequences of policies. These were necessary because the impact of a policy depended entirely on the set of policies to which it was added. For example, gasoline taxes lead to improved efficiencies in automobiles and lessen the impacts of fuel efficiency standards. By the same token, the improvement in miles per gallon (better measured as gallons per mile) from adding a gasoline tax after fuel efficiency standards is far less than adding the tax before standards. The ad hoc nature of the walk back always bothered me. To illustrate how these experiences shape one's research, it was not until 2005, almost 30 years later, that I and Ed Rosenthal figured out a better approach using the Shapley value, see Murphy and Rosenthal [43]. The other insight here is never forget any real problem you faced, as eventually you will likely find a solution.

Our efforts may seem like such a short list. However, policy proposals were flying all over the place. Model structures and solution algorithms had to be invented and reinvented in the face of hard deadlines. Data needed to be developed in areas where it was nonexistent. We had to figure out what the results meant and if they were meaningful. Consequently, we worked 7-day weeks, 10–12 hours a day, for months. Harvey coined the term "reference point," which meant a day off so that we could remember what the day of the week was.

Many people worked exceptionally hard. I mention two. Susan Holte (Shaw in the references) managed the computer runs, coordinating with the various groups producing input files during the day and nursing the runs through the system throughout the evening, all while caring for her infant daughter. She knew every number and file used for every run and she could reconstruct an old run from memory months after it was done without documentation. David Knapp developed the demand models and spent hours with the White House staff going over the numbers and doing walk backs.

Harvey and I described how we represented the National Energy Plan with PIES in Greenberg and Murphy [26].

During intense periods of work, you develop deep friendships. Harvey and I worked especially closely on the National Energy Plan and we became fast friends. Harvey and I would fill blackboards with scribbles, trying to figure out what model structures would work. One day he and I were working on a particularly difficult problem and were really into it, raising our voices out of excitement. Dave Nissen's secretary came running in thinking she had to break up a fight between Harvey and me and she was surprised to find we were having fun. The intensity and style of our working relationship set a pattern for many years. We enjoyed raising our voices at each other when working through ideas.

Those who knew Harvey knew that the workplace was never all work. Once the weather turned warm and our results were delivered, the group would periodically go to the Watergate complex, which had an excellent French pastry store. They sold a $12'' \times 12''$ box of pasty "ends" for $2.50. We would buy two and a couple of bottles of champagne. Sitting on the banks of the Potomac, we would gorge on sugar, butterfat, and good drink, three of Harvey's favorite food groups.

PIES was used for assessing the impacts of many different policies. For example, after the work on the National Energy Plan, Harvey ran a study on a proposed multi-billion-dollar plant to convert coal into methane. The thought was that natural gas was scarce and that the world needed coal gasification technology to meet expected shortfalls. Harvey showed that the multi-billion-dollar plant would be unprofitable. He was right because that plant made money only on an operating basis and later in its life only because the waste CO_2 was piped to Canada to improve oil recovery in their oil fields. It turned out that gas supply was low simply because no one explored for gas due to historically low prices while oil was more profitable. The gas reserves at the time were either associated with oil or the drillers found gas by mistake when searching for oil.

This plant illustrates how, despite best analytical efforts, the policy process can lead to poor choices. The meme in the White House was that natural gas was in short supply and we could not dispel that belief no matter how much we tried. This is a constant problem in Washington, as facts become less important than exercising power and influence. Furthermore, no one thought about climate change back then and this technology is a huge emitter of CO_2. If this technology had been deployed around the world, global warming would be even more severe than it currently is.

Those who think Washington was a nicer place then than now have it wrong: Dave Nissen was fired with the formation of the Department of Energy (DoE)

because those making the staffing decisions chose to remove everyone who did not have Civil Service protection, as FEA was set up by the Republicans. For naught the White House Energy Policy Office protested his firing because of his enormous effort and contributions to the development of the National Energy Plan. With Dave's departure, Harvey had enough and stepped into a research role. He could never accept injustice, starting with his participation in civil rights marches in the 1960s.

None of us was pleased with how Dave and our efforts were treated. I took over the PIES integrating model and did the analyses of the impacts of bills as they went through Congress and were signed into law. I then moved into the research arm of the Energy Information Administration, joining Harvey as a colleague.

A key takeaway from these events is that government employees are not always the collection of deadwood that anti-government types like to describe. There is a strong core of professionals who respond during crises. Harvey regularly said the intense level of effort was "for the good of the country." Carter described dealing with energy problems as the moral equivalent of war.

We were veterans of that effort. Bill Stitt in a private communication recently described this period as "a unique time, a uniquely talented group of people pushing the state of the art, and a high degree of policy urgency and relevance." All of us feel privileged to have had this experience, just like combat veterans. Dave Nissen recently expressed this view, despite his ultimate mistreatment by the people he served.

2.4 Modeling Regulated Market Equilibria in PIES

One of the features of inventing new approaches to modeling when faced with serious deadlines is that you do things because they work. You often do not know why they work, but you do not have the time to address the why. The best example with PIES was that during the original development, Bill Hogan invented an iterative algorithm to find the equilibrium because the structure of the demand equations meant that PIES was not an optimization model. Yet the supply structure was an optimization problem, consistent with the work of Samuelson [52]. Bill's idea was to approximate the demand curve with a simple function that made the whole model an optimization, solve a linear program, adjust the approximation based on the trial solution, and re-solve until two successive trial equilibria were within tolerance. He and Susan Holte tried this out with a small problem. It worked and he went ahead with PIES. Only much later did he and a student figure out why the algorithm found an equilibrium, see Ahn and Hogan [2].

After Harvey and I moved to the research side, we formalized the modeling and algorithms for computing regulated market equilibria, see Greenberg and Murphy [27]. I am covering this work in detail because it illustrates how you can use and manipulate duals and because it is an example of how aspects of a field develop.

The techniques remained the cutting edge for roughly 40 years but have now been superseded, see Murphy et al. [42].

The three representations of regulations we cover here all use successive over-relaxation algorithms, basically, Jacoby iterations. Whenever there was an adjustment to the demand curves in the LP there were Jacoby iterations to convert marginal costs (LP duals) to regulated prices.

2.4.1 Electricity Pricing

Electricity prices were always regulated until recently, and consumers were charged the average cost of generating electricity. Some regions in the USA, as well as other countries, have since restructured the electricity sector and now use daily auctions to acquire kilowatt hours and periodic auctions to acquire capacity. Under regulation, the price the demand model needed to see was the average cost, not the marginal cost that comes from a linear program.

The electricity sector of PIES had a regional structure for production that coincided with the demand regions. Thus, a single activity moved electricity from each electricity region to the corresponding demand region. Without any adjustment, the dual on the electricity balance in the demand region is the marginal cost. The price had to be converted to an average cost through a series of calculations. To tally the costs of electricity generation without massive calculations, a constraint for each region was added, consisting of the cost terms from the objective function for electricity plus another activity that had a -1 in the constraint and 0 for the objective coefficient. The level of that activity equaled the total non-fuel costs, including the return on capital. This constraint had a dual of 0 and did not impact the solution.

Adding the expenditures on fuels (the inflows of fuels times their duals) to the internal costs gives the total cost of generation. The flow on the transportation activity transmitting electricity from the utility region measures the total demand. Dividing total demand into total costs gives the average cost per kWh of electricity.

When electricity demand is growing, the marginal cost generally exceeds the average cost, which was our situation. Thus, we had to lower the marginal cost to the average cost. The transportation activity had an objective coefficient consisting of the transmission and distribution costs of electricity. In going from iteration t to $t + 1$, while adjusting the demand approximations, we subtracted the difference between the marginal and average costs from the transportation cost coefficient, making the dual in the demand region the average cost. When this difference oscillated, we smoothed the $t - 1$ and t values.

Because marginal cost is above average cost when demand is growing, the average cost increases with increased demand and the model is convex in the region that contains the equilibrium. This procedure converged only because demand was growing: the costs of added capacity set the marginal costs, and the capacity costs were stable. Otherwise, the adjustment would have induced an oscillation. We were lucky but we did not fully understand why. Like the problem of placing a value on

a policy in the context of other policies, this was a research question. In Mudrageda and Murphy [38] we are able to explain the convergence issues caused by step-function representations of supply and demand curves in linear programs.

There was a collection of regulations that affected what fuels could be burned in power plants. Harvey worked out the details of how to implement them. The policies included prohibiting the use of natural gas outside of meeting peak demand because of the natural gas shortages. This involved a set of restrictions on the activities that represented the operation of natural gas plants.

2.4.2 Pricing Crude Oil

The Nixon wage/price controls fixed the prices and wages of everything at their August 15, 1971 values, the date the controls were imposed, which was prior to the oil-price shock in 1973. The controls made him popular and were supposed to be in place until just after the 1972 election, at which time he no longer faced re-election. However, because of built-up price pressures, they lasted for most goods until April 1974, when they were deregulated. The price of crude oil was not deregulated because it was politically difficult to allow producers to reap "windfall profits." The 1972, pre-embargo OPEC price in nominal 1972 dollars was $1.82 per barrel and in 1974 it averaged $11.00. The US price in 1972 was $3.60. Straight deregulation would have tripled the US price to the world price, giving the oil companies quite a profit bump at a time when they were very politically unpopular.

In unregulated markets, with domestic crude oil selling at $3.60 and world crude oil selling at $11.00, anyone receiving domestic oil would buy it at $3.60 and sell it at $11.00, and they would reap a profit of $7.40 a barrel. Consequently, the consumer would get no benefit from the price controls on domestic crude oil without further rules. To solve this problem, the Nixon Administration calculated a quantity-weighted average of the domestic controlled price and the world price, and set that average as the domestic price. They placed a tax on domestic oil that raised it to this average and used the tax revenues to subsidize the per-barrel price of imports so that the post-subsidy marginal cost of imports became the weighted average price. The tax and subsidy revenues balanced and neither added nor decreased government revenues. Essentially, the supply and demand curves had the shapes in Fig. 2.1.

From a modeling perspective, the crude oil supply was taken from the supply curve at the controlled price and was a fixed quantity in the model. That is, all of the supply curve below the price ceiling was domestic supply, as shown in Fig. 2.1. The upper asymptote in the figure was the imported price. The monotonically increasing supply function was priced at the weighted average for increasing levels of imports and fixed domestic supply.

Given the number of LP activities, due to the multiplicity of crudes and regions, extracting each quantity and price to calculate the average cost and then imposing the tax and subsidy would have been time-consuming and difficult. Michael Wagner, Harvey's predecessor, came up with a simple way to do this using duality theory. He

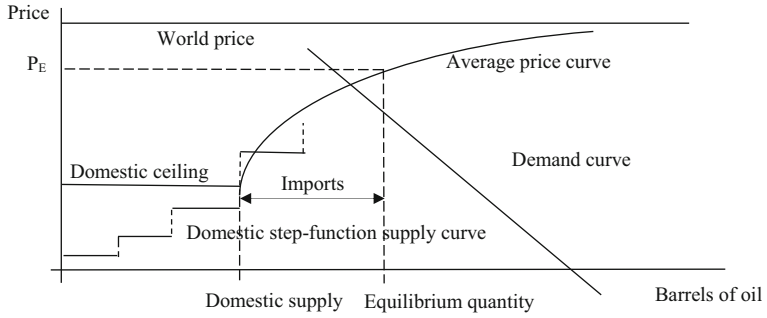


Fig. 2.1 The supply function with averaging of the domestic and world prices

added two equations, one for imports and one for domestically produced oil. Each constraint had an extra activity. One activity totaled the amount of imports and the other totaled the domestic production. The activity in the domestic constraint for PIES iteration $n + 1$ had a cost, c_D, equal to the tax on domestic crude that was calculated using the flows from iteration n. At the same time, the activity on the imports constraint had an objective coefficient for the subsidy of $-s_I$, determined in the same set of calculations:

Let

- x_i be the level of imports of crude oil i,
- y_j be the level of domestic production of crude oil j,
- v be the added activity that totals domestic crude production, and
- w be the added activity that totals imported crude production.

The submodel containing these constraints is

$$\min \cdots + c_D v - s_I w \ldots .$$

subject to

. . . . . .

$$\sum\nolimits_j y_j - v = 0$$

$$\sum\nolimits_i x_i - w = 0. \tag{2.1}$$

Because domestic production and imports were both positive, v and w had to be positive. Thus, the dual on the domestic constraint equaled c_D and raised the cost of domestic crude to the average, while $-s_I$ lowered the cost of imports to the average. This duality trick meant that only two numbers needed to be changed in PIES rather than all of the prices for the supplies and demands.

When the Carter administration and Congress deregulated crude oil prices, they imposed a tax on the oil fields that were producing at that time, termed "old oil," and deregulated oil from new fields, termed "new oil." This meant the model could represent oil markets as fully deregulated with a supply curve adjusted for the different prices for different portions of the domestic oil.

The breakdown between "old" and "new" was politically necessary. The policy change, however, had unanticipated consequences, which is quite common when the new rules alter the incentives of participants in a market. The new policy led to unnecessary drilling as companies moved oil to higher-priced categories by producing from newly drilled wells. Lawsuits overpricing involving billions of dollars were another outcome of the legislation. Nevertheless, the country got out from under the last vestiges of the Nixon price controls.

2.4.3 Natural Gas Regulation

With both electricity and oil there was always enough supply to meet demand. Utilities had guaranteed profits on capacity additions and any reductions in domestic production of oil were met by imports. There were brownouts in the early 1970s because demand grew faster than expected and some regions of the USA were short of capacity. However, that was a temporary problem and not due to faulty regulation as much as underestimates of demand growth. The long lines at gas stations were due to the Nixon price controls capping the margins of gas station owners and the owners deciding it was more profitable to close their stations earlier than usual. The gasoline lines, in reality, were a queuing problem with not enough server capacity. Oil inventories were actually higher at the end of the embargo than at the beginning.

Natural gas markets were different. The country was experiencing shortages that caused factories and schools to close. The shortages were due to a 1954 legal case where the Supreme Court ruled that the Federal Power Commission (FPC, now the Federal Energy Regulatory Commission) was required to regulate the wellhead price of natural gas sold across state borders. Natural gas sold and consumed within a state could not be regulated by the federal government because the 1938 Natural Gas Act, the basis for the decision, regulated only those pipelines that crossed state boundaries.

Regulating natural gas prices is very different from regulating a pipeline or an electric utility, something the Supreme Court failed to consider. Returns are allowed only on used and useful capital investments. Capital expenditures on pipelines, power plants, and transmission facilities rarely produce unusable capacity and the investment is easily measured for determining allowable returns. Oil and gas drilling is a risky business. In newly explored areas the odds of success can be as low as 5–10% and well below 100% for some wells in developed areas. If a regulator sets the price to recover the cost of the well that is drilled successfully, the price would provide too low a return on investment because only a fraction of wells are successful. If the regulator allows a higher price to compensate for dry holes, then

incompetent drillers could be rewarded and not put out of business. Plus, the geology is heterogeneous, eliminating any chance of cost uniformity. A third confounding factor is that a significant portion of natural gas reserves are in oil fields and there is no meaningful way to allocate the capital costs of a well between the two products. The FPC never figured out how to regulate the price based on cost because there was no solution within a standard regulatory framework that used costs and a uniform price. They essentially threw up their hands and set an arbitrary price ceiling for gas sold across state lines.

The net result of the Supreme Court decision was that two markets formed, a national interstate market and the intrastate market, where gas did not cross state lines and was not subject to federal regulation. During the build-out of the pipeline system, there were no shortages, as there were surpluses of gas associated with existing oil reserves that were flared during oil production. Consequently, the price of gas was so low that no one was searching for gas. However, as demand grew with the expansion of the pipeline network, the reserves of gas associated with oil proved insufficient to meet demand. Once the price of intrastate gas exceeded the price of interstate gas, no one wrote new contracts to sell gas into the interstate market and shortages cropped up in that market. The interstate price was too low for companies to explore for gas fields. It was an unlucky outcome that this occurred around the time of the jump in oil prices.

Given the structure of the market, shortages had to be modeled explicitly, making the modeling an order of magnitude more difficult than with oil or electricity. Furthermore, we had tight deadlines for delivering a base case for evaluating Carter's National Energy Plan.

First, note that in an economic equilibrium model, by definition, supply equals demand. In a model where this is not possible, given the regulations, what you have to do is measure the extent of the shortages and their ramifications for the economy, while still using the equilibrium framework. I was tasked with figuring out how to model this irrational policy.

I started by trying to model structures that mimicked the way duals were used to adjust oil prices. These kept failing because I did not fully understand the impact of adjusting duals through added constraints and variables. After a week of failures, I had gone home in the middle of the evening on a Saturday after submitting a run with one more try at a representation. The base case was due the following week. Dave and Harvey stayed at the office to check on the results. The run failed.

I got a call around 1:00 AM on Sunday from Dave–Harvey thought I would be too mad at him if he placed the call. Dave asked if I had any other ideas as I was waking up. After we talked, he said they would try a set of runs where they fixed the shortage at a range of levels to see what the solutions looked like. The runs produced a rough solution. After understanding why this run came close, I formalized the idea of inserting a shortage level at a proxy price for shortages. The details of the approach are in Murphy et al. [44]. The basic idea was that trying to find both a price and shortage level, as I was trying to do, was impossible because there are an infinite number of solutions. However, if you fix the price at which a

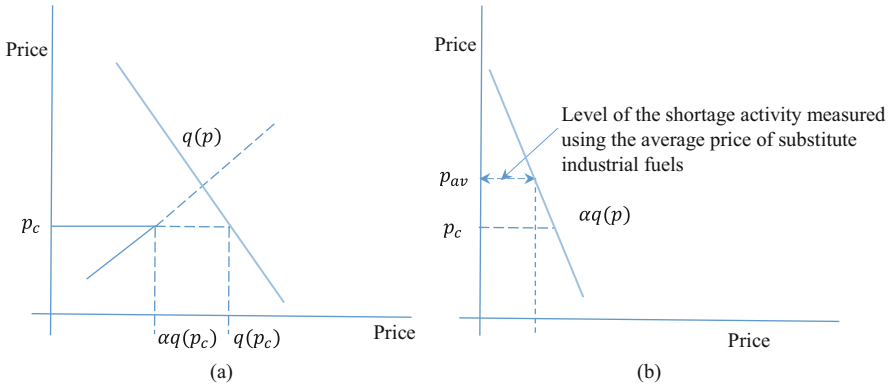


Fig. 2.2 The representation of natural gas shortages. (**a**) Original demand curve with capped price p_c. (**b**) Residual demand curve

shortage is measured, you then get a unique shortage level. This means we had to have a pricing rule for the shortages.

We fixed the price we used to measure the shortage based on the costs of the other fuels in the industrial sector. We also allocated gas to the interstate market based on historical demands before the shortages developed. To determine the level of a shortage in a region, we used the demand curve $q(p)$ and the interstate price, p_c to measure the gas demand in each region served by interstate gas, $q(p_c)$ in Fig. 2.2a, and subtracted the allocation. This gave us the share $\alpha < 1$ of customers without gas at the regulated price. We constructed a new demand curve, $\alpha q(p)$ from $q(p)$ in Fig. 2.2b to measure the level of unmet demand at every possible price. We used the gas allocation plus the demand curve for unmet demand as the gas demand curve in PIES. In the LP we inserted an activity that fed supplies of other fuels to meet the gas shortfalls in proportion to their consumption in the industrial sector. This set the dual for meeting the unmet demand at the weighted average industrial price for the substitute fuels at this price. Thus, the demand curves measured the shortfall at the average price per million BTUs of the substitute fuels. The implementation was more complex because the demand curves had n-dimensional domains and modeling them for the shortfall was not simple. Note that the shortfall measured by using the average price of the substitute fuels is less than the shortfall at the lower controlled price.

Early the next week after the failed runs, Dave had to present the base case. As Harvey, Susan, I, and others were going over the latest run on the morning of his presentation, Dave walked in wearing a new suit. We asked him why the suit? He said that since he was going to have to present the base case that day with no results, he needed to look his best. He actually said something that today is politically incorrect, comparing his lack of results to the lack of typing skills of Elizabeth Ray, the secretary and mistress of a then-powerful congressman Wayne Hayes, the latest

sex scandal of the day. Imagine his relief when we told him we finally had a good solution.

After the base case was done, implementing the White House proposals on natural gas was simpler because of their structure, and because we had a method for dealing with shortages. The proposal applied a rising cap to the natural gas price for all gas with deregulation in 1985. Furthermore, the legislation included a price ceiling for 1985 that meant no shortages in the model. That is what actually happened in the market, more by luck than wisdom.

This incident illustrates where Harvey's and Dave's leadership skills stood out. Everything with them was teamwork and problem-solving. Rank did not matter when it came to ideas and contributions. It also illustrated their trust in key staff despite inexperience. I was a year and a half into my first job outside of academia. Susan had a freshly minted master's degree in mathematics and a master's thesis on topology with no work experience when she joined the government. Yet she was responsible for the integrity of the model and its results 3 years after she started.

2.5 Replacing PIES with IFFS

The problem with PIES was that the difficulty of getting runs through even with priority access to the mainframe led to too many late hours. Runs were set up during the day, and after some time at a local bar, we would return to see if they worked. During the development of the National Energy Plan, Susan nursed the runs from home while tending to her infant daughter, and the rest of us were at the office. There was too much hardship and burnout. The hardest working team members turned over too quickly.

Harvey and I were doing different things during our research time at EIA. Harvey focused on model analysis, while I focused on model structures and the trade-offs between alternative model structures and solution times. Harvey started to develop ANALYZE, his tool for probing linear programs. I devoted my time to building a deeper understanding of the economics of different model structures to improve the representations of the different energy sectors, rethinking how one should design and organize a large-scale systems model so that it did not burn through people. Our experiences with the real problems of building and analyzing large-scale energy models defined the paths of our research.

One of the issues I was concerned about was making the tradeoff between detail/size and the difficulty in running the model. PIES was a static equilibrium model in that the market equilibrium was found for one year with the underlying dynamics embedded within the models that generated the single-year supply and demand curves. This meant the model had activities for building capacity that would last for years and would add capacity based on single-year prices, not the net present value of costs over the lives of the plants. Furthermore, demand is uncertain. Al Soyster and I were able to explain when single-year prices in a static model would find the optimal capacity mix of a multi-period model, see Murphy et al. [45]. We

needed to do this because multi-period models were just too big unless we removed detail in other parts of the model.

When one represents economic agents using optimization tools, one needs to make sure that the model has the objective function that the players actually use. Utilities were considered highly inefficient because Public Utility Commissions set electricity prices to average costs and guaranteed the profits of the utilities. This led to the Averch–Johnson [4] theory of utility behavior, where Averch and Johnson modeled utilities as maximizing profits subject to a rate of return constraint. Their conclusion was that because the allowed rate of return was higher than the cost of capital, utilities would overbuild capacity and run themselves inefficiently. Al and I took the PIES electric utilities sector and implemented the Averch–Johnson objective function to see what would happen. We found that because any reasonable demand elasticity was less than one, revenues would increase with higher prices, and any decrease in capacity would lead to higher profits, resulting in an unbounded solution. That is, utilities could not follow the Averch–Johnson theory, and a standard cost-minimizing objective function was better, even if imperfect, see Murphy and Soyster [47].

PIES burned through staff because of the huge amounts of time to produce a run, not just computer time but also the problems of debugging sectors to produce a usable run. There were two reasons why PIES was suitable only for crisis-mode situations and not for the long haul. First, given the capabilities of mainframe computers of the era, it took several hours from start to finish for a run. What makes this untenable is that there are dozens of trial runs behind every final result. Consequently, it was a slog to get through the debugging process. Second, teams were developing the supply curves and representations of energy conversion sectors, refining, and electric utilities. They had no way to test their sectors alone in a quick run. Harvey would regularly call for priority time slices on the mainframe, irritating other users and the managers of the computer systems. Given a ratio of 50 learning/bad runs to a good scenario, a lot of resources were wasted waiting for results from debugging only one sector of the model.

To try to solve the problem, Harvey hired a consulting firm with modeling expertise to see if there was a way to partition the model for debugging purposes and then put the whole model together for final runs. They came to the conclusion that this could not be done. Because of the failure of that project, I started looking at alternative formulations where decomposition was more natural. I used my one week of learning workflow diagrams in an undergraduate industrial engineering course and added the symbol for an if statement to represent the workflow with PIES and alternative designs for breaking apart the large linear program, see Murphy [39]. This was well before business process reengineering became a fad and illustrates how seemingly irrelevant, non-technical disciplines can contribute to deeply technical problems.

The result was the next generation energy model, called the Intermediate Future Forecasting System (IFFS), Murphy et al. [40], a collection of submodels linked through Jacobi iterations. Murthy Mudrageda and I (1998) formalized the properties of the underlying solution method much later. The decomposition approach I had

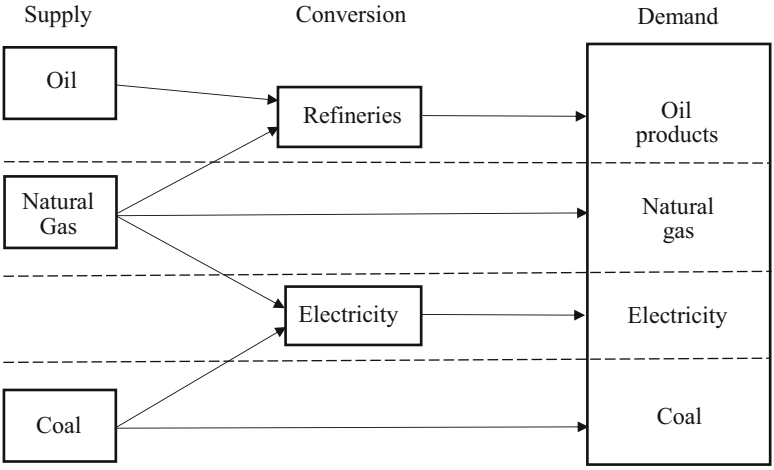


Fig. 2.3 A systems diagram of energy markets

designed for IFFS remains in the current EIA model, the National Energy Modeling System (NEMS), see Murphy and Shaw [46].

When I designed the model, all proposed energy policies were focused within individual sectors, not across sectors, improving the convergence properties. However, with global warming and policies such as carbon constraints, this was no longer true. As long as there was only one major crosscutting constraint, such as one on CO_2, one could use Everett's work on generalized Lagrange multipliers [14]. This was one of Harvey's original research areas with Greenberg and Pierskalla [31]. I was fully aware of the utility of Everett's approach from discussions with Harvey while waiting for PIES runs to process in the mainframe, and it became part of IFFS and NEMS, illustrating that even when we were not publishing together, we were helping each other and constantly talking about modeling and analysis questions.

I had the opportunity to revisit these issues on a couple of occasions. IFFS was designed around the computing capabilities of the late 1970s, a period when computers were far more expensive than people. With plummeting computing costs and vastly improved algorithms, I recommended that NEMS be rebuilt around a single linear program to deal with the convergence issues that resulted from multiple crosscutting constraints and the convergence problems associated with step functions in SOR algorithms. I at least got EIA to combine the coal and electricity sectors. Outside of that, they did not take my advice.

More recently, because of revisiting many of the modeling issues through my energy modeling work in Saudi Arabia, I realized that the consulting firm that could not decompose PIES was not qualified to do the job, and one can decompose the sector development of a linear program. To see how this can be done, begin with Fig. 2.3, a schematic of a typical multi-sector energy model.

The model consists of non-network structures within the individual sectors and basically a network that can have losses connecting the sectors. I have drawn the diagram as if the flows are acyclic, although there are small non-modeled backflows. To decompose a model for debugging sector i, one need only modify the formulation by fixing the outflows from i to sectors $j \neq i$ at the levels in some trial run and set the prices of the inputs to i from $j \neq i$ at the duals from the same trial run. If the trial run has a bounded feasible solution, then the isolated sector has a bounded feasible solution and a modification within a sector can be solved extremely quickly on its own. With individual sectors debugged on their own, the solution time for the entire model is much less of a concern, since far fewer debugging runs of the whole model are needed.

2.6 What We Learned About the Regulatory Polices

Through the modeling of the National Energy Plan, PIES became a full-fledged policy model that could represent the essential aspects of energy regulations then extant, along with all of the policy alternatives under consideration. At the same time, the various pieces of energy legislation completely deregulated fuel prices over time, which meant those representations of regulations were no longer needed, except for average-cost pricing of electricity in some parts of the USA.

The success of the deregulation of natural gas and oil prices increased policy-makers' interests in using markets to achieve policy goals. For example, the Clean Air Act Amendments of 1990 led to a market for sulfur oxides, the emissions from electricity generation, especially from coal plants. This market has greatly reduced the ecological damage from high-sulfur coal, especially in the forests in the Northeast and the rest of the country. These amendments contain the core ideas around establishing a carbon market to reduce the growth of greenhouse gases, the most efficient way to address climate change.

One outcome of the restructuring of markets under Carter was the ability of independent generators to sell their electricity into the grid at reasonable prices. This seeded the ideas around current auction markets for electricity.

In electricity restructuring, no meaningful base cases were ever developed that captured utility behavior under regulation. The only representation of utility incentives was the Averch–Johnson theory that showed firms operating under rate-of-return regulation gold plated capacity, which, as discussed above, could not apply to electric utilities.

After Carter's deregulation program and the restructuring of the electricity sector, it was possible to see where the inefficiencies existed in many industries. What the regulated firms were doing involved some Averch–Johnson gold plating because of over-engineering for increased reliability. Think of those ancient Bakelite telephones from the original AT&T that still work, and the then lower capacity utilization of airliners where passengers were not packed in like sardines. Furthermore, old capacity was kept as long as it was not fully depreciated, even

if it was obsolete, as with electromechanical telephone switches after electronic switches were developed and some old power plants when the burning of certain fuels was restricted. Note that the gold plating was not entirely wasted in that seating areas were larger, airlines had the capacity to recover quickly from disruptions and flying was much more pleasant than now. Furthermore, none of the land-line phone companies outside the USA matched the original AT&T's reliability and service, especially the nationalized companies in Europe. (Ironically, the bad service and high cost of landlines is why Europeans jumped ahead with cellular phones and there are no major manufacturers of cellular equipment in the USA.)

With deregulation the biggest cuts were in union wage rates and substantially decreased employment per unit of output. Because there was so little demand for new graduates in power engineering due to staff reductions, major university programs shut down. Under regulation labor actually benefited more than capital.

Essentially, with an eye on the regulators, the executives were buying labor peace with good wages and working conditions so that there would be no strikes that irritated regulators, giving customers better experiences than they have now. In the airline industry the residual padding was eliminated as the legacy carriers went bankrupt.

Note that I am not saying markets are always better than regulations, as unmanaged markets can lead to terrible outcomes. I am saying that market mechanisms, when they work, are lower in cost with better outcomes than rules that tend to proliferate as weaknesses in existing rules show themselves, as with air quality standards. Rules are necessary. However, they should constrain decisions, rather than specify them, or set standards for outcomes. Examples of essential rules that constrain decisions are safety rules for the air passenger industry, workplace safety rules in general and regulations on lost baggage. Food safety should be regulated by prohibiting unsafe methods and setting outcome standards for contamination (e.g. salmonella).

An example of a regulatory policy failure through specifying utility decisions is in the Clean Air Act Amendments of 1977. That law required that all new coal plants include what is known as the "best available control technology" to remove sulfur from coal, no matter how little sulfur was in the coal. Consequently, utilities chose to burn the cheapest high-sulfur coal even though burning low-sulfur coal without scrubbing the combustion gasses emits lower levels of sulfur oxides at a lower cost. The 1990 Clean Air Act Amendments corrected this costly error.

What is almost universally true is that almost all monopolies, including government agencies such as the Department of Motor Vehicles, give terrible service.

2.7 Modeling Regulations Since the Representations in PIES, IFFS, and NEMS

What we did in the 1970s and early 1980s remained the state of the art until recently, mainly because no one in developed countries wanted to reregulate deregulated sectors.

One of the big issues with restructured electricity markets is that as demand reaches its daily peak, the number of firms with available capacity shrinks, and these firms have the potential to exercise market power. Think of the debacle in California when that state "deregulated" in 2000 and peak-period prices shot through the roof. Consequently, the interesting modeling was in oligopolies, markets with a few large players. Since oligopolies cannot be represented directly in optimization models, these markets have to be modeled as mixed complementarity problems (MCPs).

Dirkse and Ferris developed a good solver for MCPs called PATH in [11]. This meant that one no longer had to iterate LPs to find an equilibrium for not only oligopoly models but also models like PIES, see Gabriel et al. [18]. Prior to PATH one had to iterate over linear programs to find an oligopolistic equilibrium, see Murphy et al. [48].

Although the OECD countries had restructured their economies to eliminate the kinds of regulations that the USA had removed under Carter (think Margaret Thatcher in the UK), nations in the developing world have retained cumbersome and costly regulations on prices and quantities. The only organizations addressing these issues have been the World Bank and International Monetary Fund. These institutions have been dominated by economists and the economics profession has not been willing to invest the effort in the large-scale models necessary to represent those regulations. Consequently, with the regulatory issues mainly addressed in OECD nations and other nations not having the capacity or willingness to examine these issues, modeling regulations in large-scale equilibrium models was moribund.

Large-scale models have been built, e.g. Loulou [33] and Zhang et al. [54]. However, these models are linear programs that ignore the regulations that shape the outcomes in industries and are not useful for analyzing policies. The state of the art in modeling regulations had not changed since PIES and IFFS until recently. With the cost of modeling tools dropping, countries that are modernizing their economies, such as India and China, now have the capacity for more sophisticated policy analysis.

I started visiting Saudi Arabia after my retirement from academia at the end of 2012. There I worked with Axel Pierru and a small group he headed in providing a tool for analyzing Saudi Arabia's energy policies. We invented new approaches for evaluating regulations using MCPs. The MCP framework simplifies the process of representing regulations greatly.

This work is described in Murphy et al. [41]. An analysis of the Saudi economy with a representation of the regulations is in Matar et al. [36].

Because of the interest in energy beyond Saudi Arabia at the King Abdullah Petroleum Studies and Research Center, we built an energy model of China with

representations of regulations that are more convoluted than the old US regulations, see Rioux et al. [49, 50].

The basic idea for modeling regulations is simple. The MCP that corresponds to a linear program

$$\begin{aligned} &\max c^T x \\ &\text{subject to} \\ &Ax \leq b \quad (u). \\ &x, u \geq 0 \end{aligned} \tag{2.2}$$

is

$$\begin{aligned} Ax + s &= b \\ A^T u - v &= c \\ x, s, u, v &\geq 0 \\ x \perp v, s &\perp u. \end{aligned} \tag{2.3}$$

So, the MCP consists of the primal and dual constraints and the complementary slackness condition. To model price regulations that do not lead to shortages, one just inserts the regulated price, u_i', in place of u_i. That is, for average-cost pricing, you add an equation $f(x, u) = u_i'$ to (2.3) that calculates the average cost, and the new u_i' is used in (2.3) instead of u_i. Note that since the added equation is an equality, it does not have to be complemented.

Other regulations are more complicated. For example, China imposes ceilings on what electricity generators can be paid. To find a feasible generation plan, I had to figure out how generators and utilities could work together to beat the rules. By offering bundles of different kinds of generation plants as a single package with all units priced at the ceiling, the overpayments on some plants cover the underpayments on others, and utilities can meet peak demand. The implementation is described in Rioux et al. [50].

Despite the long hiatus in modeling regulations, this area should increase in importance. The low cost of computing, the greater availability of data, and the increasing incomes and education levels of many developing nations means they have, or will have, the skill set and can afford to look at how their current regulations distort major sectors of their economies.

Most importantly, climate change is the central energy/environment issue of our time. Currently, many governments and large segments of the population see only near-term costs for only long-term gains. One of the big policy issues with global warming is designing transition policies that make decarbonization palatable to

governments and the populace. This means transfer payments, subsidies, prohibitions, and other forms of regulations and controls. Those who are involved should want to know the costs and benefits of the different transition rules that could be employed and the effect of price regulations on countries' abilities to meet carbon goals. However, the most notable model addressing this at the International Energy Agency, Loulou [33] is a linear program without any representation of regulations.

2.8 Model Representation and Analysis

Although PIES would be considered small by today's standards, it would still be considered complex because its heterogeneous sectors have very different structural features. Much of what is in this section reflects how we conceptualized the model and figured out what was going on inside. The issues discussed here defined Harvey's and my research careers for decades.

There are two reasons for having a strong mental representation of a model. First, it helps in understanding why a run failed. Second, you need it to explain why the model produces the results it does or why the run failed to solve. An unacceptable explanation is "that is what the model said . . . ," which is heard too often. The explanation has to translate the details of the analytical outcome into a description of what the outcome means for the physical world.

Figure 2.3 is the standard diagram we used for the PIES model. It is basically a systems diagram with non-network activities in boxes and arcs representing transportation networks. It is a useful representation for providing an overview to an outsider and it helps in develop some debugging heuristics when a model is infeasible or unbounded.

The standard representation (2.1) is useful for proving theorems about the properties of linear programs but useless for getting into the nitty-gritty of building and understanding a model. The current standard representation for modeling is computer readable algebra, the representation in AIMMS, AMPL, GAMS, MPL, and Harvey's language, MODLER [19]. I now present a simplification of PIES as algebra that translates directly into one of these matrix generators. Although a more compact representation would use an index to define the different energy sectors, I define each sector as its own block of equations to emphasize the structural features of the model. In practice they are distinct as shown here because the set of equations that define the details of these sectors are different. Label the sectors as follows, O for oil, G for natural gas, C for coal, P for oil products, E for electricity, R for refineries, U for utilities, and D for demand. I use and misuse i to index activities in a sector and r and r' to index regions. The equations are as follows.

Oil material balance with supply curve steps, x^O_{ir}, for a step-function approximation to a supply curve and transportation $t^{OR}_{rr'}$ from oil regions to refineries:

$$\sum_i x^O_{ir} - \sum_{r'} t^{OR}_{rr'} = 0 \quad \forall r$$

Gas material balance with supply curve steps and transportation from gas regions to gas consuming sectors:

$$\sum_i x_{ir}^G - \sum_{r'} t_{rr'}^{GR} - \sum_{r'} t_{rr'}^{GU} - \sum_{r'} t_{rr'}^{GD} = 0 \quad \forall r$$

Coal material balance with supply curve steps and transportation from coal regions to utilities:

$$\sum_i x_{ir}^C - \sum_{r'} t_{rr'}^{CU} = 0 \quad \forall r$$

Refineries take in transported crude oil that is used by production activities:

$$\sum_r t_{rr'}^{OR} - \sum_i x_{ir'}^R = 0 \quad \forall r'$$

Refinery activities consume natural gas, transported in, accounting for volume reductions, $\alpha_{rr'} \leq 1$, due to losses during transmission:

$$\sum_r \alpha_{rr'} t_{rr'}^{OR} - \sum_i x_{ir'}^R = 0 \quad \forall r'$$

Refineries have internal processes, k, with capacity constraints:

$$\sum_i a_{ki}^R x_{ir}^R \leq b_{kr}^R \quad \forall k, r$$

With d_i^R being the product yield from process i, refineries ship products coming from production activities to customers in utility and demand regions:

$$\sum_i d_i^R x_{ir}^R - \sum_{r'} t_{rr'}^{PU} - \sum_{r'} t_{rr'}^{PD} = 0 \quad \forall r$$

Utilities receive oil products from transportation activities and use them to generate electricity:

$$\sum_r t_{rr'}^{PU} - \sum_i x_{ir'}^U = 0 \quad \forall r'$$

Utilities receive natural gas and use it to generate electricity:

$$\sum_r t_{rr'}^{GU} - \sum_i x_{ir'}^U = 0 \quad \forall r'$$

Utilities receive coal and use it to generate electricity:

$$\sum_r t_{rr'}^{CU} - \sum_i x_{ir'}^{U} = 0 \ \ \forall r'$$

Utilities have internal resources necessary for producing electricity, including plant capacities:

$$\sum\nolimits_i a_{ki}^{U} x_{ir}^{U} \leq b_{kr}^{U} \ \ \forall k, r$$

Utilities produce electricity and transmit it to customers. Note that the transportation activity has only one index because a utility does not ship outside of its associated demand region (which was mainly the case in the 1970s):

$$\sum\nolimits_i b_{ki}^{U} x_{ir}^{U} - t_r^{ED} = 0 \ \ \forall r$$

Electricity is delivered to customers and demand is represented by a step-function approximation of the underlying demand curve, y_{mr}^{E}:

$$-t_r^{ED} + \sum\nolimits_m y_{mr}^{E} = 0 \ \ \forall r$$

Oil products are delivered to end users with step function demand curves

$$\sum_r t_{rr'}^{PD} - \sum\nolimits_m y_{mr'}^{P} = 0 \ \ \forall r'$$

Natural gas is delivered to end users

$$-t_{rr'}^{GD} + \sum\nolimits_m y_{mr'}^{G} = 0 \ \ \forall r'.$$

We place bounds on all of the steps in the step-function approximations to supply and demand curves. The objective function consists of maximizing what is known as consumer surplus, the area under the demand curve, defined by steps with prices P_{ir} for each product, less all supply costs c. For supply curves the c's are the production costs on a step function.

$$\max \sum\nolimits_{mr} P_{mr}^{G} y_{mr}^{G} + \sum\nolimits_{mr} P_{mr}^{P} y_{mr}^{P} + \sum\nolimits_{mr} P_{mr}^{E} y_{mr}^{E} - \sum\nolimits_{ir} c_{ir}^{O} x_{ir}^{O}$$
$$- \sum\nolimits_{ir} c_{ir}^{G} x_{ir}^{G} - \sum\nolimits_{ir} c_{ir}^{C} x_{ir}^{C} - \sum\nolimits_{ir} c_{ir}^{R} x_{ir}^{R} - \sum\nolimits_{ir} c_{ir}^{U} x_{ir}^{U}$$

$$- \sum\nolimits_{rr'} c_{rr'}^{OR} t_{rr'}^{OR} \sum\nolimits_{rr'} c_{rr'}^{GR} t_{rr'}^{GR} - \sum\nolimits_{rr'} c_{rr'}^{GR} t_{rr'}^{GU} - \sum\nolimits_{rr'} c_{rr'}^{GR} t_{rr'}^{GD} - \sum_{rr'} c_{rr'}^{PU} t_{rr'}^{PU}$$
$$- \sum_{rr'} c_{rr'}^{PD} t_{rr'}^{PD} - \sum_r c_r^{ED} \ t_r^{ED}.$$

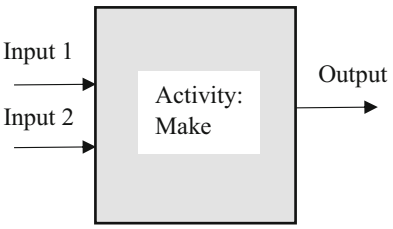


Fig. 2.4 A black box view of an activity as described by Dantzig [9]

Although I cheated on the indexing and variable names, the above equations translate directly into the model statement when using an algebraic modeling language. Note also, any real model should not have variables named *x*, *y*, or *z*. They should instead have names with physical meaning.

An algebraic statement is useful as an intermediate level of detail. However, you have to read several equations to understand why an activity is in the solution at the reported level or not in the solution at all.

Contrast this algebraic view with earlier matrix generators where you formulate a model by defining the activities followed by the rows they intersect [32]. That is, the generator loops through multiple groups of activities with each group having a common structure and related data, associating coefficients with row names. The matrix generator then combines the activities together into a matrix. I suspect the original reasons for a focus on activities rather than constraints was that each column has fewer coefficients than the typical row and facilitates solving the model when matrices had to be stored on cards.

One of the important readings on model formulation is in Dantzig's original book, Dantzig [9]. I reread it recently, and one of the most remarkable features is how few equations he used. He focused on activities as black boxes with inputs and outputs that came together through linking equations into matrices. This was partly a result of Leontief input–output models inspiring his original work. Here is the kind of visual for the black box he used (Fig. 2.4).

To me, the activity view is the most important one for understanding the meaning of a solution, which is why I focus on the sample activities that GAMS outputs when debugging a model. An activity is an agent. It sees the prices for its inputs and outputs and decides to engage in positive economic activity only if it is profitable. The function of the duals is to extract the profit and leave no economic rents (zero reduced costs) outside of the duals on the bounds of the activities. This is a property of the competitive economy. I ask what are the chains of basic activities that feed into the activity of interest? In a model with an acyclic structure, that is, the sign patterns on the activities do not create cycles, a simple tree can be used to construct the duals using the costs on the basic activities in the tree and the coefficients in the constraint matrix. One or two of the branches in the tree typically explain the interesting features of the results. When I discuss infeasibility analysis, I show how one has to look at rows and columns together to trace out the source of an infeasibility.

The previous paragraph leads to the next visualization, an abstraction of the entire matrix known as a block schematic. Only one modeling system uses this visualization to generate a linear program, MathPro [37]. Figure 2.5 is one variant of a block schematic representation. It was used for visualizing PIES. Essentially, it is the systems diagram above rendered in matrix-like form. Like the systems diagram, this visualization isolates complex production activities and clarifies the link among sectors. This view is important for understanding the flow patterns in a solution and is useful in diagnosing infeasibilities. I keep it as a mental model whenever I want to understand complex interactions in model components.

Our thinking about how to visualize models for model analysis led Harvey and I to write on representations of mathematical programs, Greenberg and Murphy [30], and compare mathematical programming systems, Greenberg and Murphy [29]. I also worked on this with other colleagues, Ma et al. [34] and Asthana et al. [3]. Harvey went well beyond what I was doing with the development of MODLER [19].

2.9 Diagnosing Infeasible Linear Programs

Understanding the sources of infeasibilities and unbounded solutions in general is a difficult problem in that no algorithm can find the specific coefficient or model component that causes a model to go awry. Developing tools for diagnosing these problems became one of Harvey's passions because finding sources of the infeasibilities or unboundedness is a constant challenge in large models. While there are algorithms for isolating the region in the model that contains the infeasibility, no algorithm can identify which coefficient is out of scale or missing, because the cause is context dependent. Working without video monitors or software for searching through a matrix or solution file, I had to have printouts that were several inches thick of the row and column listings of the matrix and the solution file. I used paper clips to mark rows and columns of interest when tracing the paths leading to the sources of infeasibilities or unbounded solutions. Harvey saw how difficult the searches were, leading him to develop ANALYZE.

The first fact that you learn when doing the search is that the output at an infeasible termination of the Phase-1 solution lists only the variable chosen for that pivot not the cause of the infeasibility. Nevertheless, the Phase-1 solution bounds where you have to look. Any row or upper bound with a non-zero dual or a reduced cost with the wrong sign is a potential location for the problem. Another feature of the typical infeasibility is that once you isolate a relevant submatrix, the problematical number or numbers are off by a large amount or a coefficient is missing. I developed a set of heuristics to guide my search. Here are some of them.

In economics inputs are categorized as substitutes or complements. For example, for most people sugar and milk are added in fixed proportions to their coffee and are complements. Artificial sweetener and sugar are substitutes. In linear

		Primary Production			Primary Transportation			Refining	Electrical Generation		Product Transportation		Capacity expansion		Imports			Exports	Demand	Constraint	
		Oil	Gas	Coal	Oil	Gas	Coal	Activity	Fossil	Non-fossil	Oil Products	Electricity	Refining	Electricity	Oil	Oil Prod.	Gas	Coal	Approx.	Type	Value
Objective: Minimize		Variable costs of activities											Annualized Capital Cost		Import Prices				Delivered Prices		
Process limits	Refining							+1					-1							≤	Existing Capacity
	Electricity								+1					-1						≤	Existing Capacity
Material limits		+A						+A	+A											≤	Material Supply
Production Regions	Oil	+A	+A		-1															≥	0
	Gas	+A	+A			-1	-1													≥	0
	Coal			+A			-1	-1										-1		≥	0
Refinery Regions	Oil				+1			-A							+1					≥	0
	Oil Products							+A			-1	-1								≥	0
Utility Regions	Gas					+1			-A								+1			≥	0
	Coal						-1		-A											≥	0
	Oil Products								-A		+1					+1				≥	0
	Electricity								+1	+1		-1								≥	0
Demand Regions	Gas					+1											+1		±	=	NG Demand
	Coal						-1												±	=	Coal Demand
	Oil Products										+1					+1			±	=	Petro. Demand
	Electricity											+L							±	=	Elect. Demand
Activity Bounds		+	+	+	+,∞	+,∞	+,∞	+,∞	+,∞	+,∞	+,∞	+,∞	+,∞	+,∞	+,∞	+,∞	±	F	+		

+1 denotes a matrix with just positive 1's if non-zero
-1 denotes a matrix with just negative 1's if non-zero
+A denotes a matrix with positive coefficients
-A denotes a matrix with negative coefficients
± denotes a matrix or vector with both positive and negative coefficients
+L denotes line losses in electricity transmission
F denotes activities with levels fixed by the bounds

Fig. 2.5 A block schematic of the PIES model

programming, when there are multiple inputs to each activity, they are complements within the activity, and other activities that provide some of the same outputs while using different inputs or the same inputs in different proportions are substitutes. Transportation activities in the standard transportation model are pure substitutes: any supply region can serve any demand region to which it is connected in a transportation network, substituting for any other supply region connected to that demand region. Furthermore, there is only one input and output to a transportation activity, making it a model of pure substitution.

The consequence of transportation activities being pure substitutes is that if one activity intersects a constraint involved in an infeasibility, then all constraints connected to the infeasible constraint through activities are infeasible in the same transportation submodel. Consequently, aggregate supply is less than aggregate demand. If some subset of regions is not involved, then transportation activities are likely to be missing. If all supply and demand regions are involved, then the cause is typically a scaling error in a set of supply or demand coefficients.

Inventories are perfect substitutes for future production. When multi-period models are infeasible in some periods and not others, there is insufficient capacity to build and/or store inventories starting in the first infeasible period.

Say an infeasibility involves only one set of production submodels such as the utility sector with activities having a complement structure. I first check if all regions are involved or just a few. If all are involved, I then look for structural features that are missing or a data error for something that is present in all regions. If it is in one region, I then look at the data unique to that region.

When examining unbounded solutions, I look for a "money pump." This typically involves a set of activities that form a cycle with + and − coefficients in some row for each adjacent pair in the cycle, a pattern that appears in a different row for each pair. You can think of this as a cycle where it is possible to pump money, or something else, that is profitable around the cycle with gains.

Clearly, searching for the source of an infeasibility involves a lot of judgment. Harvey was fascinated by these explorations. I mostly thought it was just part of the job and did not recognize the extent to which it was possible to formalize the search. We published one paper together on this topic, see Greenberg and Murphy [28]. He then delved into the subject far more deeply than I had thought possible. I thought what he achieved in this area was so useful that I asked him to write a set of tutorials for *Interfaces* in infeasibility analysis, see Greenberg [20–23]. These tutorials are extremely useful for developing the skills necessary to understand what is going on in a model.

Critical to his success in this area was the development of ANALYZE, Greenberg [23, 24, 25]. ANALYZE is a software tool for searching through the body of the matrix and the rim of a linear program, tracing paths to the source of the infeasibility. That software is now ancient and desperate for a new interface. Nevertheless, people at EIA still use it when getting into the innards of the NEMS submodels. Updating and enhancing this software would be a significant contribution to the practice of mathematical programming.

For a deeper understanding of this subject see John Chinneck [7]. John and Harvey had collaborated over the years after Harvey had brought John to one of his consortium meetings. Their collaboration was remarkably productive.

As a footnote, I want to point out that in mixed complementary models, unboundedness is the same as infeasibility. This slight change means finding the source of an infeasible solution is a monster and an untapped research area.

2.10 Reflections on Our Experiences in Government

Our time in government was a remarkable experience that we have always cherished. We were participants in the redrawing of energy policy for the nation. The country was at one of those transitions where economic theory, a systems view, and the estimates of economic impacts mattered in making major changes to the laws that affected large sectors of the economy. The concentration of talent and the development of new ideas and methodologies created an excitement that was akin to being at a startup during the early days of the Internet.

We could also observe firsthand what was good and what was not so good about government. That so much talent dove in to work on energy issues shows how government can bring the right resources to bear in crises. However, we were essentially undoing policy mistakes from previous administrations that did not understand the contribution of markets in providing opportunities for people.

Governments work best when the senior officials care about the facts. At the same time, successful policy frameworks are like waves. They wash ashore with momentum, taking them above the waterline, and then they retreat. This is happening with deregulation. The deregulation of industries under Carter meant that the economy could start growing again. These policies made the economy more adaptable so that the high oil prices of the 2000s did not trigger massive inflation, unlike the 1970s. As a result of the Clean Air Act Amendments of 1990 that were passed under G. H. W. Bush, using markets for sulfur oxides substantially reduced acid rain to the point where it rarely enters the public discourse. The furthest point of the wave running up the shore was the liberalization of the banking laws under Clinton. If banking had not been partially deregulated, which shifted so much risk from companies to the country, the financial crisis would not have been so disastrous. Still, the government response in managing the banking crisis and unemployment was exceptional.

Currently, the use of markets as a policy tool is a receding wave. The "Great Recession" and the rise of the Internet giants have many looking nostalgically back towards the "good old days" of socialism. Yet what the political economy needs is balance: the government has to set the rules and control excessive market power but should avoid counterproductive meddling. The only way to do this is to rely on the facts and to do formal analyses that include examining the potential abuse of power by firms, politicians, and bureaucrats.

I left the government soon after Reagan appointed a dentist to run the Department of Energy. It was clear the facts no longer mattered. A contributing factor in my departure was that in the formation of the Department of Energy, the old Atomic Energy Commission bureaucracy took over the internal business processes of the organization. The incredibly slow business processes matched the main business of DoE, which is building and maintaining nuclear warheads, cleaning up the environmental messes from the nuclear programs, and building and managing expensive equipment such as accelerators. Initiating anything interesting in modeling and analysis felt like fording a river of molasses. A dynamic young agency had ossified overnight.

The central problems of managing in government are regenerating talent and building an adaptable organization that can fully use its talent, while having sufficient structure to keep staff focused on the mission. Furthermore, the business processes should not be the same across all agencies. They should match the different kinds of work the different agencies performs.

2.11 Harvey's Legacy and Contribution to My Career

Harvey had a very successful research career. Moreover, the fact that I have written this tutorial as part of a collection of articles on Harvey and his work reflects one of his other accomplishments: his ability to bring people together and create research networks.

I am one example of Harvey's networking talents. I went to FEA because I had gone to graduate school immediately after my undergraduate degree with no work experience. I went directly to teaching after spending 3 years getting my Ph.D. I was developing some research momentum. However, I was not satisfied because I felt I was doing n+1 research rather than something truly novel. I went into OR because of the promise of using mathematics to address real-world issues but was not able to do that in a university. As a junior faculty member you rarely get the fresh and interesting research questions relating to real problems that can be used immediately unless you have a working relationship with a senior faculty member who is connected to the practice world or you have a relationship with a part time student who has a problem at work. I also did not know anything beyond mathematics and the mathematics of OR. I wanted to see what was real about my chosen field. I left for Washington to do that and I succeeded beyond my expectations.

The move to government led to a set of research questions that kept me busy for my subsequent academic career. Without Harvey a critical piece of a successful career would have been missing. Knowledge generation requires participating in social networks of like-minded researchers. Ideas do not arise in a vacuum. They come from meeting and learning about other people's research. This means you have to belong to a community where the members have similar research interests. Harvey was the quintessential networker and community builder. What Harvey gave

me and many others was his social network. His gregariousness meant that he was always connecting people. For example, Steve Kimbrough at Wharton and I have developed a research relationship and friendship because we both participated in Harvey's Intelligent Mathematical Programming System Consortium in Denver, which also brought John Chinneck into a research relationship and friendship with Harvey and me. Steve and I liked each other's ideas and I live a little more than a mile from where Steve works. Having a social network plus research questions meant that I could enjoy one of the best aspects of academic life, the opportunity to do research on meaningful topics with friends.

Harvey's love of bringing people together around ideas through consortia meetings, journals, and the INFORMS Computing Society (ICS) has created long-lasting social networks before we knew the phrase. That is why so many people wanted to celebrate Harvey at the Nashville ICS meeting, especially those of us who have benefited directly from his energy, ebullience, and joy in ideas and people.

References

1. R.L. Ackoff, Toward a system of systems concepts. Manage. Sci. **17**(11), 661–671 (1971)
2. B.H. Ahn, W.W. Hogan, On convergence of the PIES algorithm for computing equilibria. Oper. Res. **30**, 281–300 (1982)
3. A. Asthana, F.H. Murphy, E.A. Stohr, Representation schemes for mathematical programming models. Manage. Sci. **38**(7), 964–991 (1992)
4. H. Averch, L.L. Johnson, Behavior of the firm under regulatory constraint. Am. Econ. Rev. **52**(5), 1052–1069 (1962)
5. J. Cassidy, *How Markets Fail* (Penguin, New York, 2009)
6. A. Charnes, W.W. Cooper, *Management Models and Industrial Applications of Linear Programming*, vol I (Wiley, New York, 1961)
7. J. Chinneck, *Feasibility and Infeasibility in Optimization:: Algorithms and Computational Methods* (Springer, NYC, 2008)
8. C.W. Churchman, *The Design of Inquiring Systems Basic Concepts of Systems and Organization* (Basic Books, 1971)
9. G. Dantzig, *Linear Programming and Extensions* (Princeton University Press, 1963)
10. G.B. Dantzig, Programming of interdependent activities: II mathematical model. Econometrica **17**(3), 200–211 (1949)
11. S.P. Dirkse, M.C. Ferris, The path solver: a non-monotone stabilization scheme for mixed complementarity problems. Optim. Methods Softw. **5**(2), 123–156 (1995)
12. W. Dupree, J.A. West, *United States Energy Through the Year 2000* (U.S. Department of Interior, Washington, D.C., 1972)
13. S. Enke, Equilibrium among spatially separated markets: solution by electric analogue. Econometrica **19**(1), 40–47 (1951)
14. H. Everett III, Generalized Lagrange multiplier method for solving problems of optimum allocation of resources. Oper. Res. **11**(3), 303–473 (1993)
15. Federal Energy Administration. 1974. Project Independence Report, GPO No. 4118-00029
16. Federal Energy Administration. 1976. The National Energy Outlook
17. J.W. Forrester, *Industrial Dynamics* (Pegasus Communications, Waltham, MA, 1961)
18. S.A. Gabriel, A.J. Conejo, J.D. Fuller, B.F. Hobbs, C. Ruiz, *Complementarity Modeling in Energy Markets* (Springer, NYC, 2012)

19. H.J. Greenberg, MODLER: Modeling by Object-Driven Linear Elemental Relations. Ann. Oper. Res. **38**, 239–280 (1992)
20. H.J. Greenberg, How to analyze results of linear programs, Part 1: preliminaries. Interfaces **23**(4), 56–67 (1993)
21. H.J. Greenberg, How to analyze results of linear programs, Part 2: price interpretation. Interfaces **23**(5), 97–114 (1993)
22. H.J. Greenberg, How to analyze results of linear programs, Part 3: infeasibility diagnosis. Interfaces **23**(6), 120–139 (1993)
23. H.J. Greenberg, How to analyze results of linear programs, Part 4: forcing substructures. Interfaces **24**(1), 121–130 (1994)
24. H.J. Greenberg, *A Computer-Assisted Analysis System for Mathematical Programming Models and Solutions: A User's Guide for ANALYZE* (Now distributed by Springer, Kluwer, 1993)
25. H.J. Greenberg, The ANALYZE rulebase for supporting LP analysis. Ann. Oper. Res. **65**, 91–126 (1996)
26. H.J. Greenberg, F.H. Murphy, Modeling the national energy plan. Oper. Res. Q. **31**, 965–973 (1980)
27. H.J. Greenberg, F.H. Murphy, Computing regulated market equilibria with mathematical programming. Oper. Res. **33**(5), 935–955 (1985)
28. H.J. Greenberg, F.H. Murphy, Approaches to diagnosing infeasible linear programs. ORSA J. Comput. **3**(3), 253–261 (1991)
29. H.J. Greenberg, F.H. Murphy, A comparison of mathematical programming modeling systems. Ann. Oper. Res. **38**, 177–238 (1992)
30. H.J. Greenberg, F.H. Murphy, Views of mathematical programming models and their instances. Decis. Support Syst. **13**(1), 3–34 (1995)
31. H.J. Greenberg, W. Pierskalla, Surrogate mathematical programming. Oper. Res. **18**(5), 924–939 (1970)
32. Haverly Systems, Inc. 1974. PDS/MAGEN: A General Purpose Problem Descriptor System
33. R. Loulou, ETSAP-TIAM: the TIMES integrated assessment model Part 2: model structure. Comput. Manage. Sci. **5**(1–2), 41–66 (2008)
34. P. Ma, F.H. Murphy, E.A. Stohr, A graphics interface for linear programming. Commun. ACM **32**(8), 996–1012 (1989)
35. A.S. Manne, A linear programming model of the U. S. Petroleum Refining Industry. Econometrica **26**(1), 67–106 (1958)
36. W. Matar, F.H. Murphy, A. Pierru, B. Rioux, Lowering Saudi Arabia's fuel consumption and energy system costs without increasing end consumer prices. Energy Econ. **49**, 558–569 (2015)
37. MathPro, Inc., *MathPro Usage Guide: Introduction and Reference* (Washington, D.C., 1989)
38. M. Mudrageda, F.H. Murphy, A decomposition approach for computing large-scale economic equilibria. Oper. Res. **46**(3), 368–377 (1998)
39. F. Murphy, Large-scale modeling from an operations management perspective. Oper. Res. **41**(2), 241–252 (1993)
40. F.H. Murphy, J.J. Conti, R. Sanders, S.H. Shaw, Modeling and forecasting energy markets with the intermediate future forecasting system. Oper. Res. **36**(3), 406–420 (1988)
41. F. Murphy, A. Pierru, Y. Smeers, A tutorial on building policy models as mixed-complementarity problems. Interfaces **46**(6), 465–481 (2016)
42. F. Murphy, A. Pierru, and Y. Smeers, Measuring the Effects of Price Controls using Mixed Complementarity Models. EJOR **275**(2), 666–676 (2019)
43. F.H. Murphy, E. Rosenthal, Energy policies and the allocation of their value added. Energy J. **27**(2), 143–156 (2006)
44. F. Murphy, R. Sanders, S. Shaw, R. Thrasher, Modeling natural gas regulatory proposals using the project independence evaluation system. Oper. Res. **29**(5), 876–902 (1981)
45. F. Murphy, S. Saraf, A. Soyster, The replication of multi-year solutions using single period models of electric utility capacity expansion planning. IIE Trans. **17**(4), 396–399 (1985)

46. F.H. Murphy, S.H. Shaw, The evolution of energy modeling at the federal energy administration and the energy information administration. Interfaces **25**(5), 173–193 (1995)
47. F.H. Murphy, A. Soyster, The Averch Johnson model with Leontief production functions. Energy Econ., 169–179 (1982)
48. F.H. Murphy, H. Sherali, A. Soyster, A mathematical programming approach for determining oligopolistic equilibrium. Math. Program. **24**(1), 92–106 (1982)
49. B. Rioux, P. Galkin, F. Murphy, A. Pierru, How do price caps in China's electricity sector impact the economics of coal, power and wind? Potential gains from reforms. Energy J **38**, 63–75 (2016)
50. B. Rioux, P. Galkin, F. Murphy, A. Pierru, Economic impacts of debottlenecking congestion in the Chinese coal supply chain. Energy Econ. **60**, 387–399 (2016)
51. P. Samuelson, Enlarged ed., 1983, in *Foundations of Economic Analysis*, (Harvard University Press, Cambridge, 1947)
52. P.A. Samuelson, Spatial price equilibrium and linear programming. Am. Econ. Rev. **42**, 232–260 (1952)
53. M.E. Sanders, *Regulation of Natural Gas: Policy and Politics, 1938–1978* (Temple University Press, Philadelphia, 1981)
54. Q. Zhang, Z. Li, G. Wang, H. Li, Study on the impacts of natural gas supply cost on gas flow and infrastructure deployment in China. Appl. Energy **162**, 1385–1398 (2015)

Chapter 3
Software for an Intelligent Mathematical Programming System

Matthew J. Saltzman

Abstract Creating and understanding optimization models, instances, and solutions of any significant size present a serious challenge, even to experts in the field. Greenberg pursued an initiative in the 1980s and 1990s to support research and development of computer-assisted technologies to aid decision makers in developing models and investigating model, instance, and solution structures and implications, which he dubbed the Intelligent Mathematical Programming System (IMPS). Among Greenberg's contributions is a suite of software tools that demonstrated the potential for the initiative, including MODLER (a structured model and instance builder), RANDMOD (a structured randomization tool), and ANALYZE (a system for analyzing the structure of model instances and solutions). This paper surveys the capabilities of these tools and their underlying technologies.

3.1 Introduction

It is folk history that in the decades after its early accomplishments in military applications in World War II, operations research (OR) met with mixed success as we discovered both the breadth of applications amenable to OR approaches and the computational hurdles that needed to be overcome. Greenberg wrote in the preface to an unpublished monograph [5], "Due to the explosive growth of inexpensive computer power and to the highly successful applications during the 1960s, *we can solve far larger problems than we can understand.*[1]" However, our aspirations outstripped even those developments. As we can see in retrospect, the "explosive growth" of computer power and algorithm technology of the 1960s was merely the prelude to the dramatic progress that has occurred since.

[1]Emphasis in the original.

M. J. Saltzman (✉)
School of Mathematical and Statistical Sciences, Clemson University, Clemson, SC, USA
e-mail: mjs@clemson.edu

A. Holder (ed.), *Harvey J. Greenberg*, International Series in Operations Research & Management Science 295, https://doi.org/10.1007/978-3-030-56429-2_3

Even for optimization professionals, understanding the intimate relationships between parts of models and the extended impact on solutions of model and instance modifications is utterly impractical without technological assistance. For experts in application areas who have less expertise in optimization, the challenge is even greater. As solver engine power has advanced dramatically over the last few decades, the challenge of understanding the structure of larger and larger models has grown as well.

Computer-assisted analysis tools can be created for individual problems through a collaboration between optimization specialists and problem domain experts, but such tools are expensive: they require repeated, substantial investment of skilled labor. In the late 1970s and early 1980s, Greenberg engaged in a research program "to move some of the art of modeling and analysis to the realm of science [6]."

3.1.1 The Drive for an Intelligent Mathematical Programming System

3.1.2 Technology Context

It is worth recalling the state of computing technology during the period when Greenberg's tools were being developed. IBM announced the introduction of the personal computer (PC) in 1981. Previously, such small computers had been almost exclusively the province of hobbyists, but the PC rapidly began to penetrate the business and academic markets. Those machines offered 16-bit words (32-bit longs), less than a megabyte of RAM, and, typically, 24 × 80-character monochrome screens. Floating-point calculations were either emulated or performed by an extra-cost auxiliary processor. Early hard disk drives offered 5–10 megabytes of space. Color graphics were expensive, and for PCs, offered limited resolution and color depth. The "3M" workstation that was a target of R&D efforts at Carnegie Mellon when the author was in graduate school there included "a megaFLOPS,[2] a megaword,[3] and a megapixel.[4]" Time-sharing mainframes and minicomputers were the main interactive technologies available, but even these had on the order of a few megabytes of memory and tens of megaFLOPS. FORTRAN, COBOL, and to some extent, PL/I were mainframe languages, with FORTRAN 77 the choice for scientific computing. Pascal was the most common structured language on PCs, but it was not standardized, so different compilers supported different features and syntaxes. Unix, C, and C++ were starting to penetrate the small time-sharing system market but were not widely deployed on PCs.

[2]Floating-point operations per second.

[3]RAM.

[4]Monochrome display resolution.

By the 1990s, 32-bit CPUs, a fraction of a gigabyte of RAM, and basic color graphics were the norm for workstations. C and C++ were becoming common languages on PCs and were penetrating the scientific space. Hard disk drives were still tens of megabytes. Linux was just getting its start. Java was gaining a foothold. The World Wide Web, with support for graphics and media, was rapidly becoming the standard for disseminating information on the Internet, which was still mainly the province of government and academia. Remote access to networks was provided over voice lines with modems that could transmit and receive about 10 kb per second.

With regard to mathematical programming, commercial algebraic modeling languages such as GAMS and AMPL existed, but mathematical programming instances were still often created with custom matrix generators. The nearly universal instance interchange format for instances of linear programs (LPs) was IBM's MPS format. MPS was never standardized, so even to this day, different solvers expect slightly different variations. In addition, MPS format provides few mechanisms for expressing special structures. While algebraic modeling packages often used their own file formats for interacting with solvers, these were generally restricted to use with the corresponding packages.

3.1.3 Industrial Sponsorship

Another bit of folk wisdom is that obtaining federal funding for development of software tools to support research in multiple disciplines has historically been more difficult than obtaining funding to carry out traditional "knowledge creation" research. Greenberg also encountered that challenge in the 1980s. In response, he created an industrial consortium to support his IMPS program [6].

Greenberg's consortium proposal was developed in 1984–1985. The first company brought on board was Amoco Oil Co. Later additions included General Research Corporation (apparently now defunct), Shell Research, Ketron Management Science, US West (one of the Baby Bells), and MathPro, Inc. Phase 1 lasted until about 1989. It included development of ANALYZE, MODLER, and RANDMOD and produced over 35 documents, including software manuals and refereed journal articles. Phase 2 was under way when Greenberg's report appeared in 1990, with Phase 3 planned. The author has not located documentation on later phases.

According to Greenberg's self-assessment, there were several features of the consortium model that contributed to its success. To attract consortium members, Greenberg offered clear objectives with associated deliverables, early access to results for consortium members, and inclusion of consortium members in the priority setting process. Several workshops were held in support of the initiative, with presentations of research results and opportunities for collaboration.

3.2 Anatomy and Views of a Model

While mathematicians are comfortable with the algebraic or netform description of an optimization model, subject matter experts may not be comfortable working with those expressions.[5] Greenberg and Murphy [13] provide a taxonomy of LP views, several of which are intended to be more accessible to non-mathematical subject matter experts.

Greenberg and Murphy [13] (and Greenberg in several other publications) describe a mathematical programming model as having the partial structure depicted in Fig. 3.1. They proceed to investigate several different *views* of a mathematical programming model or instance and its solution, each of which may be appropriate for different constituents or may provide a different form of insight into model structure. MODLER and ANALYZE together provide a subset of the views presented in [13]. Commercial algebraic modeling systems generally present only one or two of these collected views.

We denote a (linear) *model* as an abstract representation of a class of *instances*. The instance class is typically infinite and is parameterized by sets of index names or values and numerical parameter and data values. The parameters defining a particular instance class share a common structure, which may be specified with more or less detail to define the structure of the abstract model.

Data objects map to the index sets and the coefficients of the objective, right-hand side, constraint matrix, and bounds. These are the components of a model that change from instance to instance. Sets are considered *symbolic* data, although they can also be described by discrete numerical values. Sets must be discrete because they provide values for indices of discrete objects of other types. For example, sets can index the terms in a summation. (If a set consisted of an interval in $\mathbb{R}$, the summation would decay to an integral, which is outside the scope of these tools.) Data objects can be *explicit* (expressed as a list or table) or *implicit* (expressed as transformations of other sets or tables).

Relations among objects also determine how an instance is generated. *Generation conditions* determine whether decision or data objects appear in a particular instance. *Admissibility conditions* express requirements on data objects such as a numeric range for a table entry or a parameter.

Greenberg and Murphy note that there is some ambiguity regarding what features are considered part of the model and what are part of an instance. In algebraic modeling languages—including MODLER—all objects and relations that are not explicit data must be declared as part of the model definition. Only the specification of explicit set, table, and parameter values distinguish instances.

[5]The term *mathematician* is used somewhat loosely here to refer to someone familiar with the mathematical aspects of optimization models, including their algebraic description, algorithmic solution, and theoretical properties, such as duality relations. *Subject matter experts* are, by contrast, familiar with the terminology related to the application area of a model, but not necessarily with its mathematical properties.

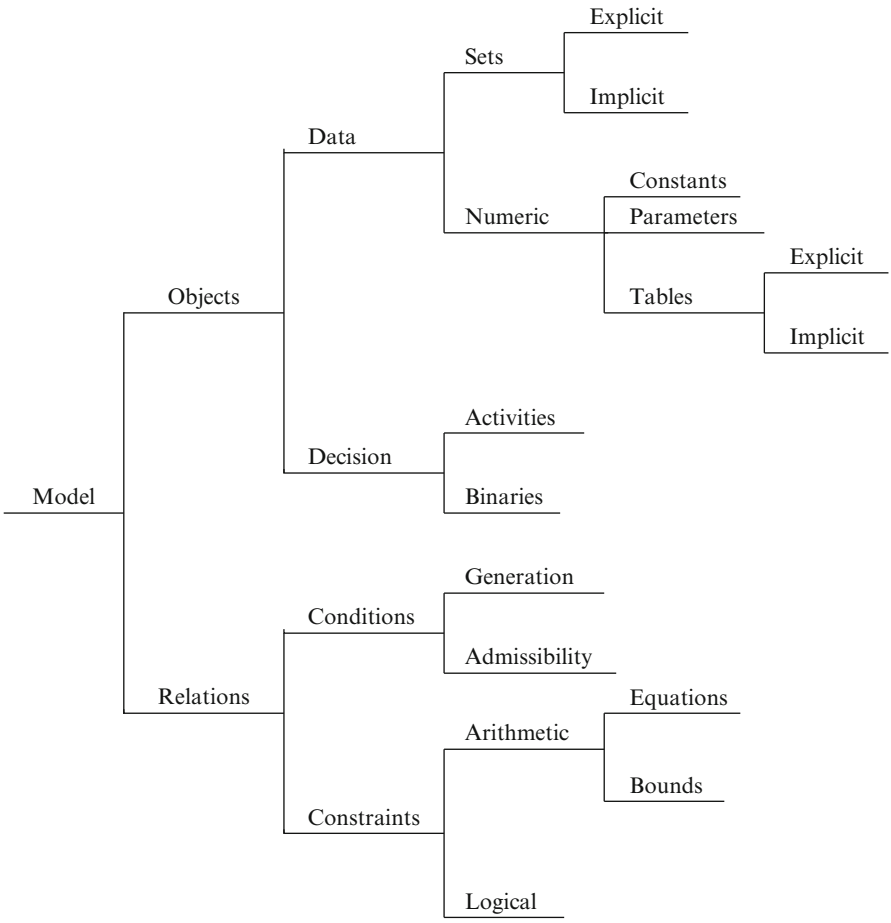


Fig. 3.1 Anatomy of a mathematical programming model

A *view* of a model is a representation of model components in a form that is comprehensible by people involved in the modeling process. These people may have different cognitive expectations for model presentation, and different views can be designed to conform to the expectations of different constituents.

As an example, we map the objects and relations associated with the *capacitated transportation model* to the entities displayed in Fig. 3.1. Model properties include identifying supply and demand points, units of goods available at supply points and required at demand points, and costs and capacities of routes connecting supply and demand points. The explicit sets in an instance definition consist of lists of identifiers for the supply and demand points. Explicit tables provide the number of available units of a good at the supply points, the number of units required at each demand point, and the shipping cost per unit and capacity in units for each

connecting route. Note that the costs and capacities are indexed by an implicit set, namely the Cartesian product of the supply and demand sets. The actual set elements and table values are not specified as part of the abstract model; they must be specified to construct an instance of the model. The decision objects here are activity levels corresponding to the number of units of goods moved from each supply point to each demand point.

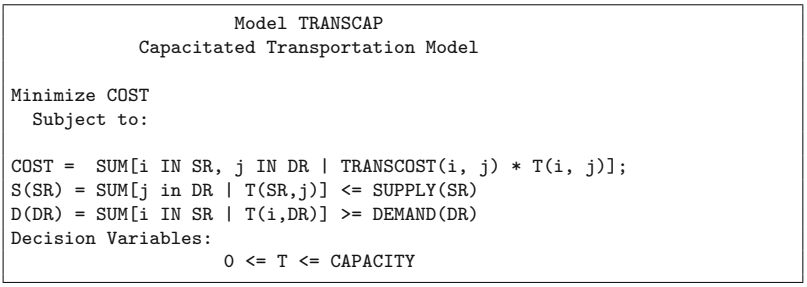
The anatomy diagram does not specify an objective object, but we can define a constraint that specifies the computation of the total shipping cost as the sum over all routes of the unit cost to ship on a route times the amount shipped on that route; then we can specify that quantity is to be minimized. The remaining constraints specify that the amount shipped out of each supply point must not exceed the supply available, that the demand at each demand point must be met, and that the amount shipped on each route must be nonnegative and must not exceed the route capacity.

Greenberg and Murphy provide examples of different views of a capacitated transportation model, and Greenberg [8, 9] provides a collection of models with the MODLER and ANALYZE software with which a user can experiment. Greenberg and Murphy illustrate several views that could be useful to various participants in the modeling process. The views presented in Fig. 3.2 are based on the abstract model of the capacitated transportation example, while those presented in Fig. 3.3 are based on a completely specified instance.

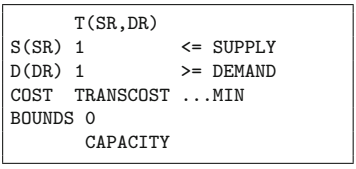
Three of the views in Fig. 3.2 are generated by MODLER. Figure 3.2a presents an algebraic view, Fig. 3.2b is a block schematic view, and Fig. 3.2d is an activity input-output view. A transportation activity in Fig. 3.2d consists of one input (the coefficient from the corresponding supply equation) and one output (the coefficient from the corresponding demand equation). MODLER's views are described in detail in Sect. 3.3. The remaining views in Fig. 3.2 were generated by other tools. Figure 3.2e shows a *netform*, or a network-based model view. The underlying model here is a classical network flow model, so each activity is represented by an arc connecting a supply node at the tail to a demand node at the head. Figure 3.2f shows a condensed version of an *activity-constraint digraph*. Figure 3.2c shows a graphical representation of activity input and output produced by the LPFORM tool [16] (more detail would be available in subordinate screens).

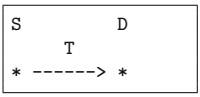
Once a model instance is *instantiated* by assigning values to all data objects, additional views are possible. Figure 3.3 shows some of these views, created by ANALYZE. Figure 3.3a is an algebraic view with coefficients displayed. Figure 3.3b is a block schematic view with coefficient values or ranges included. Figure 3.3d shows a syntax view, where descriptions of objects are expressed in text form using data provided by MODLER. Figure 3.3c displays the sign pattern of entries in the coefficient matrix and rim vectors. Figure 3.3e is an instantiated version of the activity-constraint input/output view in Fig. 3.2d. Figure 3.3f illustrates flows from supply centers to demand centers. ANALYZE's views are described in Sect. 3.5.

Commercial algebraic modeling systems are primarily designed to support an algebraic view, which is familiar to mathematicians but possibly not to other constituents.



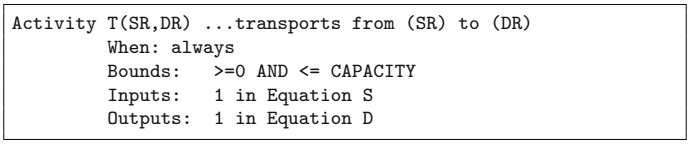
(a)



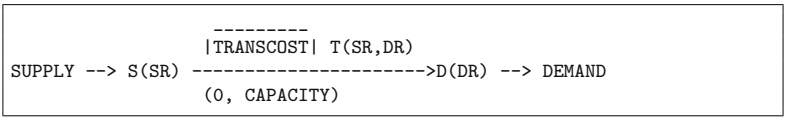


(c)

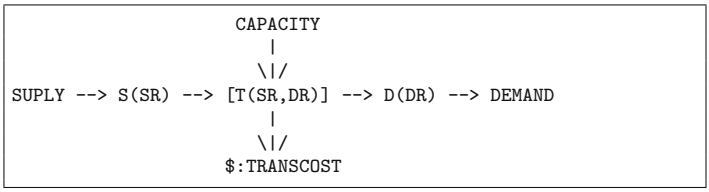
(b)



(d)



(e)



(f)

Fig. 3.2 Some model views of the capacitated transportation problem [13]. (**a**) An algebraic model view. (**b**) A block schematic view. (**c**) A block/link view. (**d**) An activity input/output view. (**e**) A netform view. (**f**) An activity-constraint view (condensed)

```
   MIN COST = TNTNT + TSWSW + 10 TNESW + 10 TSWNE
 50  <= DNE = TNENE + TSWNE
100  <= DSW = TNESE + TSWSW
100  >= SNE = TNENE + TNESE
 50  >= SSW = TSWNE + TSWSW

COL    LO_BOUND   UP_BOUND
----------------------------
TNENE       0          *
TNESW       0         50.000
TSWNE       0         50.000
TSWSW       0          *
```

(a)

```
       T(SR,DR) RHSMMODL
S(SR)  1        <= 50/100
D(DR)  1        >= 50/100
COST   1/10     ...MIN
:LO    0
:UP    50/*
```

(b)

```
       T T T T
       N N S S
       E E W W
       N S N S
       E W E W
COST   + + + + - MIN
DNE    +   +   >  +
DSW      +   + >  +
SNE    + +     <  +
SSW        + + <  +
```

(c)

```
Row syntax has 2 classes
 A row that begins with S limits supply at some supply region.
 A row that begins with D requires demand at some demand region.

Column syntax has 1 class
 A column that begins with T transports from some supply region to
    some demand region.
```

(d)

```
100 --> (SNE) ---> [TNENE] ---> (DNE) --> 50
                     $1
          50 ---> [TNESW] ---> (DWS) --> 100
                     $10
 50 --> (SSW) ---> [TSWNE]  ---> * (DNE)
                     $10
          50 ---> * [TSWNE]
                  [TSWSW] ---> * (DSW)
                     $1
```

(e)

```
      ROW DIGRAPH

Northeast ----> Northwest
          ----> Southwest
Southwest ----> Northeast
          ----> Southwest
```

(f)

Fig. 3.3 Some instance views of the capacitated transportation problem [13]. (**a**) An algebraic instance view. (**b**) A schematic view. (**c**) A sign-pattern view. (**d**) A syntax view. (**e**) An activity-constraint I/O view. (**f**) A flow view (with English translation)

3.3 MODLER: Modeling by Object-Driven Linear Elemental Relations

Custom matrix generators were the primary method of constructing nontrivial instances of linear programming models in the early days of computational optimization. Matrix generators pulled data from whatever sources were necessary and formatted them into MPS-format files for input to solvers. Separate custom programs took solution files in whatever form the solver provided them and produced reports formatted for the decision maker's convenience. As the sizes of instances solvable by computers increased and the reach of personal computers expanded through the 1980s, it became clear that these tools were not adequate to the needs of decision makers who were becoming interested in using optimization in their work. Later in the 1980s, a number of commercial products brought to market the idea of "algebraic modeling languages." AMPL, GAMS, AIMMS, MPL, and LINGO all date from that period.

Algebraic modeling languages share two key capabilities:[6]

- They support the abstract description of a model using an analog of the mathematical notation common in academic writing, with sigma notation for sums and other arithmetic and logical operators. Capabilities include the construction of flexible, abstract indexing sets and multi-subscript parameters, variables, and constraints.
- They separate the specification of the abstract model from the provision of actual values of the indexing set members and the coefficients. Thus, a single abstract description can be reused to specify multiple instances of a problem, simply by providing different data sets to accompany it.

Note that algebraic modeling languages mainly map to a mathematician's view of a problem. The level of abstraction is just what a mathematician thinks about: index sets, coefficients, variables, objective functions, and constraints.

MODLER has a more ambitious agenda [10]. As an interactive system for creating models and instances, MODLER implements an algebraic modeling language. MODLER eschews some of the more esoteric features of commercial algebraic modeling languages and is restricted to formulating linear models; however, it attempts to provide a bridge between the entities and actions that a subject matter expert might consider and the modeling objects (variables, coefficients, constraints, blocks, objectives) that form the mathematician's view. It also supports expression of logical constraints with Boolean variables and automatically converts them to linear inequalities. MODLER's language supports implied indexing and implied summations for expressions with unbound indexing variables.

[6]This definition excludes the simple, row-oriented, written-out expression languages such as LINDO or CPLEX's LP format as well as spreadsheets.

MODLER implements a strict separation of an abstract model from the data associated with an instance. It also supports randomization features that are closely tied to model structure for rapid prototyping of models.

One of MODLER's key features is the ability to generate syntactic data structures for use with ANALYZE. This feature supports expressing results of analyses in natural-language terms that would be familiar to the subject matter expert, as opposed to the language of model formulations that would require a mathematician to interpret. The instance views supported by ANALYZE are described in Sect. 3.5.

MODLER's extensive library of views and queries provides perspective primarily at the level of abstract models.

- The *algebraic view* will be largely familiar to the mathematician. It includes the usual representation of indexed constraints and summations describing a linear program.
- The *block schematic* view is an abstraction of the blocks of variables and constraints that share common names and index sets. The result is a grid with columns corresponding to variable blocks and rows corresponding to constraint blocks. The cells in each row/column indicate where the coefficients are defined. This could be a table, a range of explicit values, etc. Blocks can also appear for logical constraints and bounds.
- The *activity input/output* view shows the model as a collection of transformations. As formalized by Ma et al. [16], transformations represent conversions of form (transforming raw material into product), place (transporting from origin to destination locations), or time (carrying inventory or investments). In a canonical-form LP (minimizing subject to greater-or-equal constraints and nonnegative or bounded variables), an input to an activity is represented by a constraint with a negative coefficient and an output is represented by a constraint with a positive coefficient. MODLER also supports assigning these and other user-defined attributes to sets for display in MODLER and ANALYZE views. MODLER's activity I/O view displays for each activity class a list of constraints where the activity takes an input and where it produces an output.
- *Dependency relations* can be displayed, showing which objects are defined in terms of the sets, parameters, and tables that provide the data for instantiation of an instance of the model. Implicit sets and tables are dependent on the explicit objects that define them, and variables and constraints are dependent on the sets that index them and the parameters and tables that provide their coefficients.

MODLER includes a randomization function that is designed to rapidly prototype instances of a model for testing. Limited randomization can be accomplished interactively from MODLER's console or, more flexibly, from input files that provide explicit set, parameter, and table values. The randomizer can set probabilities for selection among a specified list of ranges or a default range; then random numbers of specified precision are generated with a specified distributions.

3.3.1 Capturing Structure in Instance Representations

MODLER's output is intended to provide input to a solver engine and to the companion tools, RANDMOD (a tool to construct random instances from a template instance) and ANALYZE (MODLER's companion tool for analyzing instances and solutions). The *matrix file* is a standard MPS-format description of the instance, which is input to the solver and to RANDMOD and ANALYZE. The *syntax file* provides a collection of verbal descriptions of objects that can be used with MODLER's description of the model, the matrix file, and the solution report from the solver to display properties of an instance and its solution in natural language.

For generating views and responding to queries regarding instances, Greenberg describes a mapping from object identifiers (variable and constraint group names, index set members, etc.) to instance row and column names in the matrix file. In MPS format, row names, column names, and bound and right-hand side block names are all simple strings of eight characters. (In some MPS extensions, longer names are permitted, but the forms and restrictions are far from universal. These tools generally kept to the most widely supported formats.) ANALYZE and RANDMOD identify substructures and generate views and query responses by matching substrings to patterns. For example, in Greenberg's WOODNET sample model describing production and distribution of lumber, the activity name `TMOSFSE` represents transportation (`T`) of mahogany (`MO`) from a supply point in San Francisco (`SF`) to a demand point in Seattle (`SE`). A syntax for masks supports substring matching to select groups of objects.

3.4 RANDMOD: Controlled Randomization of Linear Programs

RANDMOD [7] is a tool for constructing random instances of linear programs for algorithm testing purposes. Given an input instance specified in an MPS-format matrix file, RANDMOD can produce transformed instances using any of several transformations and generate random values according to any of several distribution classes. The transformations include:

- Augmentation—adding rows to a problem instance constructed from conic combinations of existing inequality rows. The additional rows can be shifted to be strictly redundant or to create degeneracies or infeasibilities.
- Perturbation and scaling—changing row or column bounds or coefficient values.
- Removing bounds.

Row augmentation and perturbation are mutually exclusive operations.

The weights used to construct combinations of rows or to modify coefficient values are randomly generated. The user can specify a range and distribution for a base value, scale factor, offset, and number of modifications for each operation. The

supported distributions are uniform, triangular, normal, and exponential. Transformations can be restricted to submatrices based on name patterns. Each collection of transformations produces a new instance that can be saved in a matrix file with the same naming patterns as the template (except for added rows).

3.5 ANALYZE: A Computer-Assisted Analysis System for Mathematical Programming Models and Solutions

Once an instance of a mathematical programming model is instantiated, a number of views can be produced that present the detailed data provided in context. In addition, if a solution is available, more insights can be provided into the relationships between activities and constraints at that solution. Even if it is determined that no feasible solution exists, it is possible to determine what parts of the model or instance might be responsible for that outcome. ANALYZE can provide all these perspectives and more, and can present them in natural-language form if provided with an appropriate syntax file. In addition, ANALYZE provides a customizable, rule-driven interface for adding new knowledge generation tools for problems with special structure.

A summary of the inputs to ANALYZE and the general classes of outputs are shown in Fig. 3.4. ANALYZE requires at least a *matrix file* describing an instantiated instance, and with only that input, ANALYZE supports a limited set of queries that do not rely on the model's structure. *Dictionaries* and *documents* define the interaction between program and user, mediated by the FLIP subsystem (the FORTRAN Language Interactive Processor), the dialog engine for ANALYZE as well as MODLER and RANDMOD. The *solution file* is the output of any of a handful of solver engines that ANALYZE is able to parse, as there is no widely used format for expressing solutions.

The key to ANALYZE's power as an investigative tool is the *syntax file* provided by MODLER. This file includes the maps from the row and column names in the matrix file to the block structure object names and indices of the original model. It also contains the natural-language descriptions of objects used in ANALYZE's natural-language interface. ANALYZE's reasoning capabilities are driven by rule-based logic. Standard and custom rules are provided via *rule files*. Finally, ANALYZE is capable of interacting with external tools such as Chinneck's IIS (irreducible infeasible subsystem) analyzer [2].

Provided with appropriate inputs, ANALYZE supports sensitivity analysis, various views and queries, model simplification, and interpretation of model and solution structure as well as debugging inquiries such as identifying infeasibilities.

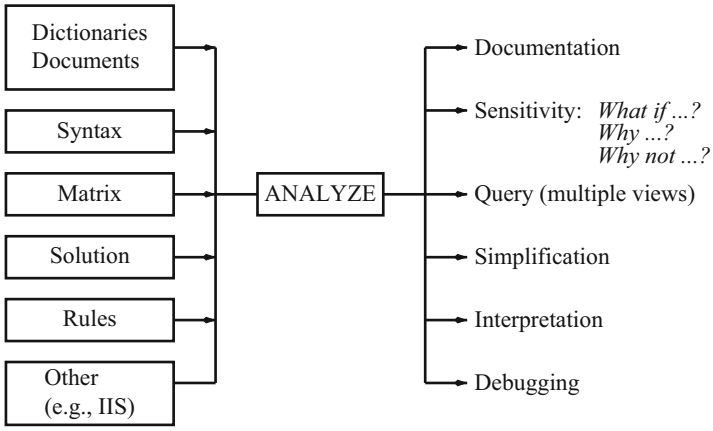


Fig. 3.4 ANALYZE input/output

3.5.1 Views and Analyses

Some available views of an instantiated instance of an LP refine similar views of an abstract model. An algebraic view of a model might involve summations over named index sets of named coefficients, but the instance view can show the actual index and coefficient values. The block schema for a model shows groupings of rows and columns by name, but the instance view can show ranges of coefficients. See related views in Figs. 3.2 and 3.3 for comparison. ANALYZE can also display constraint matrix sign patterns. While the lack of graphics capabilities limited the size of such displays, they were still useful for selected submatrices.

In addition to refinements of model views, there are many ways to explore relationships among components of instances and solution values. By tracing through submatrices associated with active resource constraints and basic activities, ANALYZE can provide information about the makeup of shadow prices and reduced costs, marginal substitution rates, sensitivity of the solution to changes in coefficient values, and other properties of the instance and solution. Those insights can be presented through displays of objects and their properties in diagrams or tables. By using the natural-language descriptions of objects in the syntax file, ANALYZE can also represent its findings in verbal summaries.

The *fundamental digraph* of an instance [4] is a directed bipartite graph with a node for each row and column and an arc connecting row i to column j if matrix coefficient a_{ij} is negative and an arc connecting column j to row i if a_{ij} is positive. For an LP in the canonical form (minimize subject to greater-or-equal constraints and nonnegative variables), one can interpret a negative coefficient as indicating that the row resource is an input to the column activity. A positive coefficient indicates that the row resource is an output of the column activity.

The fundamental digraph can be projected onto the row or column node sets, with an arc between rows in the former or columns in the latter corresponding to directed paths of length 2 in the fundamental digraph. The row digraph captures transformations between resources connected by an activity that takes one resource (the tail) as input and produces another (the head) as output. The column digraph captures precedence, in which one column (the tail) produces a resource that another activity (the head) consumes. ANALYZE can display subgraphs of these graphs to visualize these relations.

While the general question of whether a constraint is redundant has the same complexity as solving the original LP, some redundancies can be verified through the same sorts of analyses as those listed above. ANALYZE can also diagnose infeasibilities using a successive bounding procedure or by hooking to an external engine that implements Chinneck's IIS detector.

3.5.2 Algorithmic Analysis

ANALYZE includes several algorithms and heuristics that support a deeper understanding of the interactions between model instances and solutions than is afforded by simply looking at activity levels and dual prices. The key algorithms in ANALYZE's repertoire include:

- *Path tracing* builds a submatrix that includes rows corresponding to the resources associated with an activity or subset of activities and all the activities that interact with the activity of interest. From that submatrix, ANALYZE can determine the impact of marginal changes in the activity of interest.
- *Basis rearrangement* permutes basis rows and columns to bring the basis matrix to a triangular or near-triangular form.
- *Rates of substitution* can be computed by completing the product-form factorization of the triangularized basis and invoking the FTRAN and BTRAN procedures from the simplex method (to solve $B\boldsymbol{x} = \boldsymbol{a}$ and $B^T\boldsymbol{\pi} = \boldsymbol{c}$, respectively, where $\boldsymbol{a}$ is a column of the constraint matrix and $\boldsymbol{c}$ is a subvector of the objective).
- Some cases of *redundancy* can be detected by computing ranges on basic variables that maintain feasibility as nonbasic variables are set to their most permissive bounds. If the upper or lower bounds on the left-hand sides are tighter than the upper or lower bounds on the right-hand side, the corresponding constraint is redundant.
- *Primal and dual bounds* can be reduced sequentially until infeasibility is detected or the bound reduction process stabilizes.
- *Logical implications* for binary variables can be imputed based on constraint left-hand side bounds.

3.5.3 *The Rule Base*

Rule-based reasoning is one research thrust of artificial intelligence. The idea is to capture the thought process of an expert analyst in the field of interest in a form that can be carried out automatically by a computer. ANALYZE contains a rule-based reasoning engine that includes a number of standard analytical procedures such as interpreting a shadow price or identifying an embedded network. The rulebase is extensible and customizable so that new analyses for special problem structures can be implemented.

Rules can be invoked by the user and can in turn implement algorithms automating the steps of an analysis, such as identifying the contributions of activities to a shadow price or the contribution of resources to a reduced cost in a problem instance. Rules can invoke the core algorithms described in Sect. 3.5.2, where the components that contribute to an interpretation may depend on special structure of the problem.

3.6 WRIP: A Workbench for Research in (Linear) Programming

In 1991, Greenberg and Marsten released a package [12] containing the three analysis tools described here together with an LP solver: Marsten et al.'s OB1 [1, 15, 17, 18]. OB1 is a FORTRAN code that includes Marsten's XMP simplex solver and several different interior-point solvers, plus a crossover code to recover a basic optimal solution from an optimal interior-point solution. The package also includes test instances from Netlib and elsewhere [3, 14] as well as tools for visualization.

Greenberg and Marsten's view of a workflow for experimenting is pictured in Fig. 3.5. A matrix file for a base LP instance could be selected from a library or created using MODLER. The LP could be solved with OB1 or processed through RANDMOD to create additional, similar instances. Solutions from OB1 could be analyzed with ANALYZE, and the output of RANDMOD, OB1, and ANALYZE could be fed to a statistical analysis of, for example, solver performance. The results of the analysis could be reported and could be fed to RANDMOD to produce additional instances for further testing.

3.7 Conclusion

The Intelligent Mathematical Programming System initiative spearheaded by Greenberg in the 1980s and 1990s was an ambitious program to harness emerging computing power to enhance the analyst's ability to formulate, analyze, and reason about optimization problems in the context of decision support systems. Greenberg's

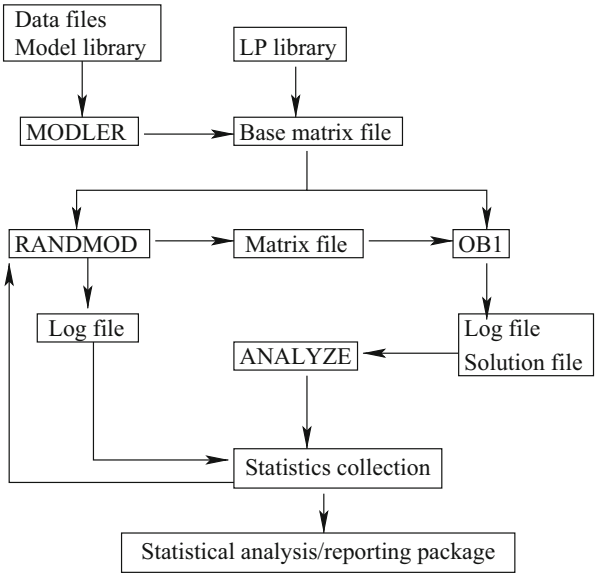


Fig. 3.5 Job flow for experimental analysis

1996 bibliography [11] lists over 500 references, which are classified as relevant to background, analysis, discourse, formulation, model management, and software engineering and implementations, plus relevant general knowledge. Greenberg himself is listed as author or coauthor on nearly 50 of the publications. But among his most influential contributions to the initiative is the fact that he put into practice the principles that he developed and assembled by publishing the software packages MODLER, RANDMOD, and ANALYZE.

While some of the knowledge developed through the initiative and related efforts has been integrated into widely used tools, many of the capabilities of Greenberg's codes have not been so widely deployed. Compiled versions of MODLER and ANALYZE for Microsoft Windows are distributed with user's guides currently available from Springer [8, 9]. Windows and Linux executables for MODLER and ANALYZE and Windows executables for RANDMOD were available for download from Greenberg's University of Colorado at Denver Web pages. These can still be run on systems available as of this writing. Source code for ANALYZE exists and should eventually be available as open source, once proper permissions can be secured. Sadly, source for MODLER and RANDMOD appears to be lost to history, unless some kind reader has an archive that they can share with this author.

References

1. I. Adler, N. Karmarkar, M.G.C. Resende, G. Viega, Data structures and programming techniques for the implementation of Karmarkar's algorithm. ORSA J. Comput. **1**(2), 84–106 (1989)
2. J.W. Chinneck, E.W. Dravnieks, Locating minimal infeasible constraint sets in linear programs. ORSA J. Comput. **3**, 157–168 (1991)
3. D.M. Gay, Electronic mail distribution of linear programming test problems. Math. Program. Soc. Committee Algorithms (COAL) Newslett. (1985)
4. H.J. Greenberg, A new approach to analyze information contained in a model, in *Energy Models Validation and Assessment*, ed. by S.I. Gass, vol. 569 (National Bureau of Standards, Gaithersburg, 1978), pp. 517–524
5. H.J. Greenberg, Foundations for an intelligent mathematical programming system. Draft monograph (1988)
6. H.J. Greenberg, An industrial consortium to sponsor the development of an intelligent mathematical programming system. Interfaces **20**(6), 88–93 (1990)
7. H.J. Greenberg, RANDMOD: a system for randomizing modifications to an instance of a linear program. ORSA J. Comput. **3**(2), 173–175 (1991)
8. H.J. Greenberg, *A Computer-Assisted Analysis System for Mathematical Programming Models and Solutions: A User's Guide for* ANALYZE. Operations Research/Computer Science Interface Series, vol. 1 (Springer, Berlin, 1992)
9. H.J. Greenberg, *Modeling by Object-Driven Linear Elemental Relations: A User's Guide for* MODLER. Operations Research/Computer Science Interface Series, vol. 2 (Springer, Berlin, 1992)
10. H.J. Greenberg, MODLER: modeling by object-driven linear elemental relations. Ann. Oper. Res. **38**, 239–280 (1992)
11. H.J. Greenberg, A bibliography for the development of an intelligent mathematical programming system. Ann. Oper. Res. **65**, 55–90 (1996)
12. H.J. Greenberg, R.E. Marsten, WRIP: a workbench for research in (linear) programming. Software Manual (1991)
13. H.J. Greenberg, F.H. Murphy, Views of mathematical programming models and their instances. Decis. Support Syst. **13**, 3–34 (1995)
14. I.J. Lustig, An analysis of an available set of linear programming test problems. Comput. Oper. Res. **16**(2), 173–184 (1989)
15. I.J. Lustig, R.E. Marsten, D.F. Shanno, On implementing Mehrotra's predictor-corrector interior-point method for linear programming. SIAM J. Optim. **2**(4), 435–449 (1992)
16. P.-C. Ma, F.H. Murphy, E.A. Stohr, A graphics interface for linear programming. Commun. ACM **32**(8), 996–1012 (1989)
17. R.E. Marsten, M.J. Saltzman, D.F. Shanno, G.S. Pierce, J.F. Ballintijn, Implementation of a dual affine interior point algorithm for linear programming. ORSA J. Comput. **1**(4), 287–297 (1989)
18. R.E. Marsten, R. Subramanian, M. Saltzman, I. Lustig, D. Shanno, Interior point methods for linear programming: just call Newton, Lagrange, and Fiacco and McCormick! Interfaces **20**(4), 105–116 (1990)

Chapter 4
Harvey Greenberg: Analyzing Infeasible Mathematical Programs

John W. Chinneck

Abstract As part of his *Intelligent Mathematical Programming System* project, Harvey Greenberg investigated theory and developed methods for diagnosing the cause of infeasibility. The emphasis was on developing useful and practical tools for isolating the problem to a small part of a large model and arriving at an understandable explanation, or diagnosis, of the infeasibility. He leveraged known mathematical theorems—and developed new ones—to create the requisite tools for incorporation into his ANALYZE software. This chapter summarizes his contributions to practical methods for analyzing infeasible mathematical programs.

4.1 Introduction

As described elsewhere in this book, Harvey Greenberg initiated the *Intelligent Mathematical Programming System* (IMPS) project [12]. A main goal of the IMPS was to provide tools to deal with the complexities of large-scale mathematical programming models and solutions, e.g. explaining their behavior, understanding causal relationships, and providing useful insights. Post-solution analysis was a major part of this effort, including providing explanations for pathological outcomes such as infeasibility and unboundedness.

This chapter deals with Harvey Greenberg's contributions to the analysis of infeasible mathematical programs, mainly linear programs. Greenberg assembled and extended the known theory on infeasibility with an eye to making it useful in practice by incorporating it into his ANALYZE software [8, 15, 17, 18, 22] for the analysis of linear programming (LP) models. The value of this difficult task should not be underestimated: there are subtle pitfalls in converting theory to practice in such a way that it is indeed useful.

J. W. Chinneck (✉)
Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada
e-mail: chinneck@sce.carleton.ca

A. Holder (ed.), *Harvey J. Greenberg*, International Series in Operations Research & Management Science 295, https://doi.org/10.1007/978-3-030-56429-2_4

4.1.1 Defining Infeasibility

A fundamental point is the definition of infeasibility in a mathematical program. Mathematically, this means that the solution point $\boldsymbol{x}$ violates one or more constraints. In practice, however, solutions are calculated on machines having a limited number of bits to represent real-valued numbers, and hence a certain tolerance on precision must be allowed. The most common resolution of this difficulty is to consider a constraint to be satisfied even if it is violated by up to a small *absolute feasibility tolerance* τ_a, often on the order of 1×10^{-6}. This is called a *function tolerance test*. For example, where the constraint takes the form $g(\boldsymbol{x}) \leq b$, the function tolerance test allows all solutions that satisfy $g(\boldsymbol{x}) \leq b + \tau_a$.

But even this is not a complete resolution of the difficulty because models are normally *scaled* prior to solution by applying multipliers to the constraints and variables to help prevent calculation errors. Scaling can impact whether or not a constraint is considered to be satisfied or violated using a function tolerance test. For example, where τ_a is 1×10^{-6} the point $x = 1.000001$ just satisfies the constraint $x \leq 1$, but violates the equivalent constraint $10x \leq 10$. For reasons such as this, it is not uncommon to see one solver consider a solution point to be optimal while a different solver considers the model itself to be infeasible.

However there are a number of other ways to deal with the imprecision of finite-length digital representations of numbers and their impacts on feasibility assessment. Greenberg [23] provides an in-depth analysis of this topic. He points out that solvers may also use a relative feasibility tolerance τ_r, which is applied to a constraint of the form $g(\boldsymbol{x}) \leq b$ as follows: $g(\boldsymbol{x}) - b \leq \tau_r|b| + \tau_a$. Solvers may provide different tests for primal and dual feasibility as well as the duality gap. In mixed-integer solvers a tolerance is used to determine whether an integer variable value, treated as real-valued, is close enough to an integer value to be rounded to that integer value. For integer rounding decisions, it is common to consider v close enough to its integer rounding if $|v - \lfloor v + 0.5 \rfloor| \leq \tau_r|v|$.

4.1.2 Isolating and Diagnosing Infeasibility

With infeasibility precisely defined as above, Greenberg addressed the issue of explaining infeasibility with the goal of providing a useful diagnosis of the cause. Such a diagnosis is vital when dealing with very large models, e.g. LPs having millions of constraints and bounds. When the solver reports that the model is infeasible, where does the analyst look to effect a repair? Greenberg's approach was to try to isolate the portion of the model that generates the infeasibility [13], which eases the generation of the diagnosis in large models.

The main terms are defined by Greenberg and Murphy [24]: an *isolation* is a small portion of the model that contains the infeasibility, preferably as small as

possible. A *diagnosis* is a meaningful isolation that is interpretable by the human modeler. A good diagnosis leads to the quick resolution of the modeling error.

Greenberg developed a number of practical approaches to narrowing the focus of the search for the cause of infeasibilities:

- Analyzing the logic trail of bound tightening sequences that lead to infeasibility,
- Path and cycle tracing,
- Using the properties of interior point LP solutions,
- Specific techniques for infeasible network models based on flow balancing theorems,
- Finding maximum cardinality feasible subsets of constraints in an infeasible LP.

Greenberg developed these methods before practical tools for isolating an *Irreducible Infeasible Subset* of constraints (IIS) in an infeasible model became available. An IIS is an infeasible set of constraints that is rendered feasible if any single constraint is removed, thus all of the members contribute directly to the infeasibility. Techniques for isolating IISs in a large set of constraints (see, e.g., [3]) eventually became the method of choice in commercial LP solvers, though the methods listed above have their uses in particular situations. Greenberg [14, 16] compared several methods for infeasibility analysis and concluded that isolating an IIS is generally the best approach. Greenberg [18] then incorporated IIS isolation software into his ANALYZE software.

Greenberg's numerous contributions to infeasibility analysis in mathematical programs are described below.

4.2 Reasoning About Bounds

Greenberg used reasoning about bounds to help isolate infeasibility in LPs in two ways: (i) calculating aggregate constraint violations using variable bounds and dual prices, and (ii) successive bound reduction, including extensions such as reasoning about specific LP structures such as block-and-link constraint matrices. The goal in all cases is to isolate the infeasibility to as small a portion of the model as possible.

4.2.1 Phase 1 Dual Prices and Aggregate Constraints

This approach makes use of the following theorem [13]: Let $S = \{\boldsymbol{x}: \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}, \boldsymbol{a} \leq \boldsymbol{Ax} \leq \boldsymbol{b}\} = \varnothing$, and let $\boldsymbol{\pi}$ be a phase 1 dual solution price vector associated with the range constraint on $\boldsymbol{Ax}$. With notation $\pi_i^+ = \max(0, \pi_i)$ and $\pi_i^- = \min(0, \pi_i)$, it then follows that $\{\boldsymbol{x}: \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}, \alpha \leq \boldsymbol{\pi Ax} \leq \beta\} = \varnothing$, where $\propto = \boldsymbol{\pi}^+\boldsymbol{a} + \boldsymbol{\pi}^-\boldsymbol{b}$, and $\beta = \boldsymbol{\pi}^+\boldsymbol{b} + \boldsymbol{\pi}^-\boldsymbol{a}$. Defining $\lambda = \min(\boldsymbol{\pi Ax}: \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u})$ and $\mu = \max(\boldsymbol{\pi Ax}: \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u})$, it follows that either $\mu < \alpha$ or $\lambda > \beta$ in an infeasible system. If $\mu < \alpha$, then the greatest value of $\boldsymbol{\pi Ax}$ is not enough to satisfy its lower bound, that is

activities having positive coefficients have upper bounds that are too low or activities having negative coefficients have lower bounds that are too large. If $\lambda > \beta$, then the smallest value of $\boldsymbol{\pi A x}$ is too large to satisfy its upper bound, and hence activities having positive coefficients have lower bounds that are too large, or activities having negative coefficients have upper bounds that are too small. In some cases, e.g. flow models, this information provides sufficient clues to narrow the diagnostic effort to a small part of the model.

This approach allows the partitioning of the constraints into two distinct sets: those with $\pi_i \neq 0$ (associated with active constraints) and those with $\pi_i = 0$ (associated with inactive constraints). The cause of the infeasibility must necessarily include some or all of the constraints from the first set. This is helpful in isolating the cause of the infeasibility, but it is not definitive.

4.2.2 Successive Bound Reduction

Successive bound reduction is commonly seen in the bound tightening that is carried out in any standard LP presolver. Given a constraint and its bounds, tighter bounds can often be deduced. For example, given the constraint $5x + 10y \leq 12$ with $0 \leq x \leq 5$ and $0 \leq y \leq 8$, we can tighten the upper bounds on both variables as follows:

- x achieves its largest feasible value when $10y$ is as small as possible, which is at $y = 0$. Thus $5x \leq 12 \rightarrow x \leq 2.4$.
- y achieves its largest feasible value when $5x$ is as small as possible, which is at $x = 0$. Thus $10y \leq 12 \rightarrow y \leq 1.2$.

A standard presolver carries out a chain of such bound reductions, with each tightened bound potentially initiating a cascade of other bound reductions on both variable bounds and constraints.

To continue the example above, suppose we have another constraint $x + y \leq 10$. With the tightened bounds deduced above, we can now tighten this constraint to $x + y \leq 3.6$. Such bound tightening sequences can lead to the detection of infeasibility. For example, if the model also contained the constraint $x + 2y \geq 5$, the tightened bounds on x and y would now reveal the infeasibility that the maximum possible value of $x + 2y$ is 4.8.

While a standard LP presolver can detect infeasibility when it finds a conflict of this sort, Greenberg tried to use such an outcome to reach an isolation of the infeasibility by analyzing the chain of reductions. In practice, analyzing infeasibility via successive bound reduction is fraught. The chain of reductions can be very long and include numerous reductions that are irrelevant to the infeasibility diagnosis. Greenberg [13] provides a small example of an infeasibility diagnosis via successive bound reduction in which the reduction chain has 10 irrelevant reductions, plus 5 relevant equations. Though it can be very helpful, Greenberg notes that "Successive bounding can fail, and it is very unpredictable."

Successive bound reduction can also lead to the detection of a forced value for a variable. This happens when the bounds and constraints act in a way that forces a variable to a single value. If the forced value is zero, which is one of the more common forced values, then the variable is said to be *nonviable* [2]. Greenberg [19] explores techniques for finding forcing substructures of a model (bounds and constraints) and shows that the techniques for doing this are the same as those used for analyzing infeasibility.

4.2.3 Block-and-Link Structures

The idea of successive bounds reduction can be extended to collections of constraints if the LP has a block-and-link structure, which is not uncommon. In a block-and-link LP, the constraint matrix can be decomposed into non-overlapping blocks of coefficients, with a generally smaller set of variables that links all of the blocks. Greenberg and Murphy [24] show how to leverage this structure to isolate an infeasibility to a particular block. The blocks can be tested individually by considering each block as an individual linear program, with the link variables used as slack or surplus variables, and any one of the link variables used as the objective function. If this LP is infeasible, then the infeasibility has been isolated to the block. If this block LP is feasible, then tighten the bounds on the link variables by using the block LP with each link variable solved for its maximum and minimum values. These bounds can then propagate through the model, potentially leading to the isolation of an infeasibility elsewhere.

4.3 Path and Cycle Tracing

A fundamental issue in analyzing infeasibility is understanding how the constraints and variables in a model influence each other. For example, when the infeasible solution point is shown to violate a particular constraint, a diagnosis can be sought by tracing the set of other constraints and variables that influence the violated constraint. This is the goal of path and cycle tracing.

Greenberg [8] represents constraint influences via a bipartite directed graph based on the signs of the coefficients in a constraint. The two vertex sets are the rows (R) and columns (C). The *fundamental digraph* is constructed as follows. For i in R and j in C, there is one arc for each nonzero coefficient in the constraint matrix A: (i,j) is an arc if $A_{ij} < 0$ and (j,i) is an arc if $A_{ij} > 0$. Tracing paths through the fundamental digraph provides information on variable and constraint influences. Cycles in the fundamental digraph are of particular importance because they can help explain infeasibility, particularly dual infeasibility; the ANALYZE software has a command to find such cycles. See the examples in Greenberg [8].

4.4 Interior Point Solutions and Infeasibility

Greenberg [20, 21] noticed that an interior point method solution of an infeasible LP separates inequality constraints into two sets: (i) those that might be part of some IIS and (ii) those that cannot be part of any IIS. This is an improvement over other methods for isolating IISs that cannot consistently identify all of the constraints that are part of some IIS. It provides a way to immediately discard all of the constraints that are irrelevant to the infeasibility.

Interior point solutions provide a strictly complementary partitioning of the constraints. If $S = \{\boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}\}$ is a finite collection of inequalities, $X(S) = \{\boldsymbol{x}$: $\boldsymbol{x}$ is feasible in $S\}$, and the dual system is $S^d = \{\boldsymbol{\pi} \geq \boldsymbol{0}, \boldsymbol{\pi}\boldsymbol{A} = \boldsymbol{0}, \boldsymbol{\pi}\boldsymbol{b} \geq 0\}$, then the strictly complementary partitions theorem is stated by Greenberg [21] as follows: If S is consistent, then there exists a strictly complementary solution, $(\boldsymbol{x}, \boldsymbol{\pi}) \in X(S) \times X(S^d)$. Further, the support partition is the same for all strictly complementary solutions.

Greenberg goes on to apply this property to infeasible systems as follows. Define the feasible *LP*: max $\boldsymbol{\pi}\boldsymbol{b}$ subject to $\boldsymbol{\pi}\boldsymbol{A} = \boldsymbol{0}$, $\boldsymbol{\pi} \geq \boldsymbol{0}$, $\boldsymbol{\pi}\boldsymbol{b} \leq 1$. Define the support set $\sigma(v)$ of a nonnegative vector v as the set of indices for which the coordinate is positive. A solution in $X(S^d)$ has the support set $\sigma(\boldsymbol{\pi}) = \{i|\pi_i > 0\}$. If $\boldsymbol{x} \in X(S)$ and $\boldsymbol{\pi} \in S^d$, then we have complementary slackness, i.e. $\boldsymbol{A}_i\boldsymbol{x} = b_i$ for all $i \in \sigma(\boldsymbol{\pi})$. The solutions are strictly complementary if $\boldsymbol{A}_i\boldsymbol{x} > b_i$ for all $i \notin \sigma(\boldsymbol{\pi})$. A strictly complementary solution induces a support partition, $\sigma(\boldsymbol{\pi}) \cup \sigma(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})$ on the indices of the inequalities.

If the optimal solution to *LP* is obtained by an interior point method, then the optimal partition, say $\boldsymbol{\pi}^0$, is strictly complementary. Now $\sigma(\boldsymbol{\pi}^0) = \{i|\boldsymbol{A}_i\boldsymbol{x} \geq b_i$ is in some IIS of $S\}$. S is separated into two parts by the strictly complementary solution: those that might be part of some IIS and those that are not part of any IIS. This partition can be used to eliminate the inequalities that are not part of any IIS, immediately improving the focus of the search.

4.5 Analysis of Infeasible Networks

Greenberg [10, 11] considered how to diagnose infeasible minimum cost network flow programs in a pair of papers. His approaches mainly relied on the logical application of existing supply and demand balancing theorems.

As an example, the main theorem by Gale [7] states that the total demand over a network is feasible if and only if for every subset S of nodes, the total demand over the complement of S is less than or equal to the total capacity of the arcs that cross from S to its complement. The proof depends mostly on the minimum cut theorem. This applies to individual nodes as well as any larger collection of nodes. This means that there must always be enough supply to meet demand in any partitioning of the arcs in the network. Greenberg used these balancing rules

to construct more sophisticated analysis procedures in an effort to better isolate an infeasibility.

Other relevant flow balancing algorithms are due to Fulkerson [6], Hoffman [26], and Ford and Fulkerson [5]. Greenberg and Murphy [24] refer to the following theorem as the *Gale–Fulkerson–Hoffman theorem:* given a network flow model consisting of vertices V subdivided into the set S of supply nodes, the set D of demand nodes, and the set T of transit (flow balancing) nodes, there exists a feasible flow for $[V,D,S,T]$ if, and only if, the value of the min cut from S to D is in the interval of the lower bound on D to the upper bound on S.

The Gale–Fulkerson–Hoffman theorem can be used to help identify the bottleneck between supply and demand that is causing an infeasibility. But Greenberg and Murphy [24] point out that its guidance is frequently insufficient to clearly identify the cause of an infeasibility. More exact localization is needed. Greenberg [9, 11] combines the flow balancing results with logic about network behavior to yield heuristics that give better localization of an infeasibility. New specific tests such as path and cycle generation are combined with methods akin to bound reduction. These heuristics improve the usefulness of the base flow balancing techniques, but there is no guarantee that an irreducible infeasibility will be isolated, or that the resulting reductions will be helpful in understanding the infeasibility, as for all logical reduction/presolving methods. These techniques are available in the ANALYZE software [8, 15, 17, 18].

4.6 Comparison of Infeasibility Analysis Techniques

Greenberg compared the main methods for analyzing infeasible linear programs described above in a pair of papers. Greenberg and Murphy [24] describe these techniques:

- Phase 1 dual methods (see Sect. 4.2.1).
- Elastic programming, in which elastic variables are added to allow all constraints to be satisfied. Elastic variables that have nonzero values must be part of the infeasibility.
- Bound reduction (see Sect. 4.2.2) and propagation of tightened bounds.
- Gale–Fulkerson–Hoffman flow balancing theorems (see Sect. 4.5).
- Bound reduction in block-and-link LP structures (see Sect. 4.2.3).
- Parametric programming to find the closest possible feasible solution for an infeasible LP (see Sect. 4.10).
- Partitioning, or finding maximal feasible subsets of constraints (see Sect. 4.8).
- Minimal dependency sets.

The overall conclusion is that there is no one method that best analyzes all types of infeasibility, so Greenberg concentrates on assembling the available methods into a toolkit that can be applied by people or artificially intelligent assistants.

In the second comparison, Greenberg [14] considered three methods for the analysis of infeasible blending models (common in the petrochemical industry): (i) phase 1 price aggregation (Sect. 4.2.1), (ii) irreducible infeasible systems, and (iii) bounds reduction (Sect. 4.2.2). The criteria for the comparison were how much effort was needed to arrive at a diagnosis and the quality of the final diagnosis. The ANALYZE software [18] is used to manage the diagnostic process in all three cases (with IISs supplied by Chinneck's MINOS(IIS) code [4]).

The paper concludes that the isolation of IISs "performed consistently above midrange, and it never failed to provide useful information. It frequently gave an immediate diagnostic." See also Greenberg [16] for further study of the value of isolating IISs during the diagnostic process. Phase 1 price aggregation proved useful in simple blending models, while bound reduction failed completely in some cases but gave insightful diagnostics in others.

Greenberg [9] had earlier addressed the idea of searching for IISs substructures in infeasible LPs, but noted that how to find them was unclear. He considered Van Loon's [27] search for tableauxs that meet certain conditions to identify an IIS, but noted that Van Loon's search is undirected and will in general enumerate many bases that do not provide any information about the cause of the infeasibility. Greenberg and Murphy [24] point out that his method could be extended to find IISs more efficiently by pivoting through alternative bases.

4.7 Infeasibility Analysis in ANALYZE

The *ANALYZE* software [8, 15, 16, 18, 22] is a general purpose tool for manipulating and analyzing linear programs. It includes routines for the infeasibility analysis methods described above, including bound tightening, path and cycle tracing for infeasible networks, row aggregation, and tools for syntax-based explanation. While it is not able to isolate IISs directly, it can read IIS output files produced by MINOS(IIS) and apply the tools mentioned above to provide a deeper analysis of the infeasibility.

4.8 Maximum Feasible Subsets of Constraints

Infeasible sets of constraints can also be analyzed by attempting to find a *Maximum Feasible Subset* (*maxFS*) of constraints, i.e. a largest cardinality feasible subset. It is NP-hard to find such a subset [1], so heuristics are generally used. The complementary subset of constraints, called the *IIS set cover* among other names, consists of constraints that are involved in one or more IISs, and hence this set is more important to the infeasibility. Greenberg and Murphy [24] refer to this division of the constraints as *partitioning*.

An exact *maxFS* solution via mixed-integer linear programming has been suggested several times. Greenberg and Murphy [24] formulate it as a mixed-integer bilinear problem as follows. Define the binary variables $u_i = 1$ if the maximum feasible subset includes constraint i, and $u_i = 0$ if it does not, and define $U = \text{diag}(u_i)$. The LP $\boldsymbol{Ax} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}$ is thus equivalently represented as $\boldsymbol{UAx} = \boldsymbol{Ub}$, $\boldsymbol{x} \geq \boldsymbol{0}$. In a feasible system all $u_i = 1$, but this is not possible in an infeasible LP, hence we seek to maximize $\Sigma_i u_i$ s.t. $\boldsymbol{UAx} = \boldsymbol{Ub}, \boldsymbol{x} \geq \boldsymbol{0}$.

The MIP formulation is difficult to solve, especially for very large infeasible LPs, so heuristic solutions have become dominant in practice (see [3], chapter 7). These heuristics cannot guarantee to find a maximum feasible subset, but they will always find a maximal feasible subset. In a maximal subset, moving any constraint from the complement into the maximal set renders it infeasible, but the set is not of maximum cardinality. For example, suppose we have two IISs {A,B,C} and {C,D,E}. The maximum feasible subset is {A,B,D,E}, but there are various maximal subsets that are not of maximum cardinality, such as {B,C,D}. However there is diagnostic value in any of these sets, since it focuses attention on the constraints in the complement of the maximum/maximal feasible subset. In the preceding example, finding the maximum feasible subset focuses attention on constraint C, the only constraint that appears on both IISs.

4.9 Minimum Feasible Partitions

Finding a maximum feasible subset of constraints as in Sect. 4.8 divides the constraints into two sets: the feasible subset and its complement. There is no guarantee that the complement is itself a feasible set. Thus arises the *Minimum Number of Feasible Partitions problem* (min PFS): partition the original infeasible set of constraints into the smallest number of partitions such that every partition is feasible.

Any set of linear inequalities $\boldsymbol{Ax} \geq \boldsymbol{b}$ can be partitioned into two sets that are both feasible. The proof is provided by Greenberg [21] in the following theorem: Suppose a set S of linear inequalities is inconsistent. There exists a partition of S, say $S' \cup S''$ such that S' and S'' are each consistent and S' is a maximal consistent subsystem (in which case $X(S') \cap X(S'') = \varnothing$).

Proof Construct a line that intersects each hyperplane, $H_i = \{\boldsymbol{x} | \boldsymbol{a}_i\boldsymbol{x} = b_i\}$ where $\boldsymbol{a}_i \neq \boldsymbol{0}$ for each i. Totally order the points along the line; rename and reorder so that $\boldsymbol{x}^i$ is the point on H_i. Now initialize $S' = \{\boldsymbol{a}_1\boldsymbol{x} \geq b_1\}$ and continue to add $\boldsymbol{a}_i\boldsymbol{x} \geq b_i$ to S' as long as $\boldsymbol{a}_i\boldsymbol{x}^k \geq b_i$ for all $k < i$. The first time this fails, initialize $S'' = \{\boldsymbol{a}_i\boldsymbol{x} \geq b_i\}$. For each $i > k$, the half-space $X(\{\boldsymbol{a}_i\boldsymbol{x} \geq b_i\})$ intersects either $X(S')$ or $X(S'')$, so the inequality can be added to S' or S'', respectively. Test first if $S' \cup \{\boldsymbol{a}_i\boldsymbol{x} \geq b_i\}$ is consistent and if so add this inequality to S'. It then follows that all inequalities not in S' are precisely those whose augmentation renders inconsistency. This means that S' is a maximal consistent subsystem (and that $X(S') \cap X(S'') = \varnothing$).

This theorem does not apply when linear equalities are included in the set. For example, a set of three or more parallel but separated linear equality hyperplanes necessarily decomposes into the same number of feasible partitions, each including a single hyperplane.

4.10 Finding the Closest Feasible Solution

If a model is infeasible, then some understanding of the cause can be obtained by finding the closest feasible solution. Greenberg and Murphy [24] suggest that this might be done via parametric programming. If $\boldsymbol{Ax} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}$ is infeasible, then solving the LP *maximize* θ *s.t.* $\boldsymbol{Ax} = \theta\boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}$ provides guidance concerning which constraints are preventing a feasible solution (note that θ must be strictly less than 1 because there is no feasible solution when $\theta = 1$).

The parametric programming multiplier can be applied to only the subset of constraints that are under suspicion. For example, if the supply limits are thought to be the cause of the infeasibility, then applying the parametric parameter to just those constraints (while retaining the rest of the model) supplies an answer: if there is no feasible solution, then the source of the infeasibility lies elsewhere in the model. On the other hand, if there is a feasible solution, then there is an indication of how much the supply must increase so that a feasible solution can be found.

4.11 Analyzing Infeasible Mixed-Integer Linear Programs

Analyzing infeasible mixed-integer linear programs is more difficult than analyzing infeasible LPs. Guieu and Chinneck [25] apply IIS isolation techniques to MIP problems, but this requires the solution of a large number of MIPs and is inherently slow (and subject to other numerical issues). Greenberg applies bound reduction methods for dealing with binary variables in the reduce command of his ANALYZE software [17] (see Sect. 4.2.2), which can be helpful in some cases.

4.12 Conclusions

Harvey Greenberg published a series of influential papers on analyzing infeasible mathematical programs during the years 1983 to 1996. This work was conducted as part of his *Intelligent Mathematical Programming System* project, with the ultimate goal of providing practical tools for diagnosing infeasibility in large and complex mathematical programs, which may consist of millions of constraints and bounds. Practical tools are needed, and developing these requires skills at the interface of mathematics, operations research, and computer science. Greenberg incorporated

many of these tools into his ANALYZE software, the first practical demonstration of a mathematical modeling analysis tool, and a forerunner of an entire class of later software.

References

1. E. Amaldi, V. Kann, The complexity and approximability of finding maximum feasible subsystems of linear relations. Theor. Comput. Sci. **147**, 181–210 (1995)
2. J.W. Chinneck, Formulating processing network models: viability theory. Naval Res. Logist. **37**, 245–261 (1990)
3. J.W. Chinneck, *Feasibility and Infeasibility in Optimization: Algorithms and Computational Methods, International Series in Operations Research and Management Sciences* (Springer Science+Business Media, LLC, New York, 2008)
4. J.W. Chinneck, E.W. Dravnieks, Locating minimal infeasible constraint sets in linear programs. ORSA J. Comput. **3**, 157–168 (1991)
5. L.R. Ford, D.R. Fulkerson, *Flows in Networks* (Princeton University Press, Princeton, 1962)
6. D.R. Fulkerson, A network flow feasibility theorem and combinatorial applications. Can. J. Math. **11**, 440–451 (1959)
7. D. Gale, A theorem in networks. Pac. J. Math. **7**, 1073–1082 (1957)
8. H.J. Greenberg, A functional description of ANALYZE: a computer-assisted analysis system for linear programming models. ACM Trans. Math. Softw. **9**, 18–56 (1983)
9. H.J. Greenberg, Computer-assisted analysis for diagnosing infeasible or unbounded linear programs. Math. Program. Stud. **31**, 79–97 (1987)
10. H.J. Greenberg, Diagnosing infeasibility in min-cost network flow problems; Part I: dual infeasibility. IMA J. Math. Manage. **1**, 99–109 (1987)
11. H.J. Greenberg, Diagnosing infeasibility in min-cost network flow problems; Part II: primal infeasibility. IMA J. Math. Manage. **2**, 39–50 (1988)
12. H.J. Greenberg, An industrial consortium to sponsor the development of an intelligent mathematical programming system. Interfaces **20**, 88–93 (1990)
13. H.J. Greenberg, Rule-based intelligence to support linear programming analysis. Decis. Support Syst. **8**, 1–23 (1992)
14. H.J. Greenberg, An empirical analysis of infeasibility diagnosis for instances of linear programming blending models. IMA J. Math. Bus. Ind. **4**, 163–210 (1992)
15. H.J. Greenberg, Intelligent analysis support for linear programs. Comput. Chem. Eng. **16**, 659–673 (1992)
16. H.J. Greenberg, How to analyze the results of linear programs - Part 3: infeasibility diagnosis. Interfaces **23**, 120–139 (1993)
17. H.J. Greenberg, *A Computer-Assisted Analysis System for Mathematical Programming Models and Solutions: A User's Guide for ANALYZE* (Kluwer Academic Publishers, Boston, 1993)
18. H.J. Greenberg, Enhancements of ANALYZE: a computer-assisted analysis system for linear programming. ACM Trans. Math. Softw. (TOMS) **19**(2), 233–256 (1993)
19. H.J. Greenberg, How to analyze the results of linear programs – Part 4: forcing substructures. Interfaces **24**, 121–130 (1994)
20. H.J. Greenberg, The use of the optimal partition in a linear programming solution for postoptimal analysis. Oper. Res. Lett. **15**, 179–185 (1994)
21. H.J. Greenberg, Consistency, redundancy, and implied equalities in linear systems. Ann. Math. Artif. Intell. **17**, 37–83 (1996)
22. H.J. Greenberg, The ANALYZE rulebase for supporting LP analysis. Ann. Oper. Res. **65**, 91–126 (1996)

23. H.J. Greenberg, Mathematical Programming Glossary Supplement: Tolerances, World Wide Web (2003), http://glossary.computing.society.informs.org/notes/tolerances.pdf
24. H.J. Greenberg, F.H. Murphy, Approaches to diagnosing infeasibility for linear programs. ORSA J. Comput. **3**, 253–261 (1991)
25. O. Guieu, J.W. Chinneck, Analyzing infeasible mixed-integer and integer linear programs. INFORMS J. Comput. **11**, 63–77 (1999)
26. A.J. Hoffman, Some recent applications of the theory of linear inequalities to extremal combinatorial analysis. Proc. Symp. Appl. Math. **10** (1960)
27. J. van Loon, Irreducibly inconsistent systems of linear inequalities. Eur. J. Oper. Res. **8**, 283–288 (1981)

Chapter 5
Development of Publications and Community at the Interface Between Operations Research and Computing

J. Cole Smith

Abstract Harvey J. Greenberg's energy and dedication to the field of operations research yielded an impressive array of contributions in undergraduate and graduate education, research, and professional service. This chapter focuses on his instrumental role in creating a journal, book series, and professional society, all of which remain strong and influential today. The journal is now the *INFORMS Journal on Computing* and is currently publishing its 31st volume. The *TutORials in Operations Research* book series is currently publishing its 15th annual volume, and the ever-growing INFORMS Computing Society traces its roots back over 40 years from the time it was a special interest group within ORSA.

5.1 Introduction

This chapter explores part of the history of computing publications and its associated community within the operations research and management science world, focusing on the leadership of Dr. Harvey J. Greenberg. Especially throughout the 1970s and 1980s, the call for infusing computing technology within operations research became increasingly loud from top researchers in the field. There are a great many people—several still very active today—who share responsibility and credit for cultivating the links between these areas.

Notable for his leadership in this area was Greenberg, whose efforts led him to take on three particular projects that have substantially influenced many careers, my own included. Those projects include a journal, a book series, and a community, all of which are more prominent currently than ever before.

– The journal began as the *ORSA Journal on Computing*. ORSA (Operations Research Society of America) merged with TIMS (The Institute for Management Sciences) and became INFORMS (Institute for Operations Research and the

J. C. Smith (✉)
Clemson University, Clemson, SC, USA
e-mail: jcsmith@clemson.edu

A. Holder (ed.), *Harvey J. Greenberg*, International Series in Operations Research & Management Science 295, https://doi.org/10.1007/978-3-030-56429-2_5

Management Sciences) effective at the beginning of 1995 [15]. The journal then changed its name as of the first issue (vol. 8, issue 1) of 1996 to the *INFORMS Journal on Computing*. Where needed to avoid confusion in this chapter, we refer to the journal simply as the *Journal on Computing*, or just JOC.
– The book series is *TutORials in Operations Research*, published by INFORMS. The concept was to collect book chapters based on the very successful tutorials track offered annually at the INFORMS Annual Meeting. The idea for this series was attempted, abandoned, and successfully relaunched in the mid-2000s.
– The society is the INFORMS Computing Society (ICS), whose current membership exceeds 1600 people and is affiliated with the *INFORMS Journal on Computing*.

The goal of this chapter is not to tell a complete history of the aforementioned journal, book series, or society but is to instead acknowledge that these three "projects" have had lasting influence on the operations research community. This chapter draws in part from my own experience, from published literature, and perhaps most significantly from conversations with many of Greenberg's contemporaries. Section 5.2 discusses the evolution and impact of the *Journal on Computing*. Section 5.3 examines the influence of the ICS and the *TutORials* book series. Section 5.4 concludes with some summary thoughts on Greenberg's contributions.

5.2 The Journal on Computing

> I believe it is, and always has been, the ICS mission to articulate and lead the development of interfaces between operations research and computer science. It is not just a fact of history, but a matter of necessity, that these communities interact.—Harvey Greenberg [12].

This section examines the development and evolution of the *Journal on Computing*, starting from its origins in the 1980s and continuing through the time of this writing (summer 2019). Section 5.2.1 examines the origins and leadership of this journal from the time of its inception. Section 5.2.2 provides a timeline of the journal's areas and the editors for those areas.

5.2.1 Origins and Leadership

One of Greenberg's original research areas was the infusion of the science of computing with operations research. Today, the two seem inextricable, but at a time when the term "computer science" was still relatively new, there were many open questions. In fact, Greenberg himself reflects in [12] that one such open question was the nature of the interface between operations research and artificial intelligence—an interface that remains of great interest today. It was not until 1976 that a team consisting of Gordon Bradley, Gerald Brown, Milt Gutterman,

Table 5.1 Editors-in-chief of the *Journal on Computing*

Years	Editor-in-chief
1987–1992 (issue 2)	Harvey J. Greenberg
1992 (issue 3)–1999	Bruce Golden
2000–2006	W. David Kelton
2007 (issue 1)	Prakash Mirchandani
2007 (issue 2)	W. David Kelton (interim)
2007 (issue 3)–2012	John W. Chinneck
2013–2018	David L. Woodruff
2019–present	Alice E. Smith

and Greenberg created an ORSA Computer Science Special Interest Group. The leadership of that group extends well beyond these four, including also early leaders like Karla Hoffman, Ric Jackson, Dick Nance, and Dick O'Neill, who collectively transformed the Special Interest Group into the Computer Science Technical Section around 1980.

With that foundation, and with the growing interest in the interface between operations research and computing, Greenberg led a team with Karla Hoffman, Bob Jeroslow, Don Kraft, and Bill Pierskalla to explore the creation of the JOC. A truly significant decision was made in 1987 by the Computer Science Technical Section membership to have ORSA publish the JOC, as opposed to using an outside publisher. Greenberg served as the founding editor-in-chief as a result of his efforts, serving in this role for three-and-a-half years of its publication, in addition to the time he spent laying the groundwork to launch the journal between 1987 and 1989. Table 5.1 displays the timeline of the journal's editors-in-chief.

The JOC was established as a quarterly publication, and it has remained so through 2019. The very first issue (published in winter 1989) had articles by (a) Robert Jeroslow and Jinchang Wang on a topic intersecting integer programming and computational logic, (b) Stavros Zenios on parallel optimization, (c) Jaya Singhal, Roy Marsten, and Thomas Morin on a software system for binary optimization, and (d) Daniel Heyman and Alyson Reeves on solving linear equations in Markov chain analyses. In fact, Greenberg's stewardship of this journal led to the JOC quickly garnering a reputation as a high-quality journal with an exceptional editorial board. In the third issue of volume 1, the journal published perhaps its most cited and well-known paper to date: Part one of a two-part paper on tabu search by Fred Glover [10]. It is still instructive to read some of the early issues and uncover advice on several practical research challenges that exist today, e.g., how authors can help expedite the review process, and how to set up meaningful computational experiments.

Starting midway through the fourth volume, Bruce Golden took over as editor-in-chief, and he immediately instituted a suite of new areas (see Sect. 5.2.2) along with attractive art for the cover that changed with the season corresponding to the issue. Golden's leadership came at a formative time for the journal, during which the journal was challenged to become more profitable, change from the *ORSA*

Journal on Computing to the *INFORMS Journal on Computing*, and perhaps most importantly, compete for inclusion in the list of journals having an International Scientific Indexing (ISI) impact factor. The former mission was accomplished by a multipronged approach, including obtaining sponsorship of the journal. As for the latter, Golden was able to continue Greeenberg's momentum in establishing the JOC as a prestigious journal. His focus was not only on maintaining high standards but also in seeking opportunities for themed issues and feature articles.

The feature article concept stemmed from Golden's vision of an article that would be, "informative, accessible, provocative, and exciting," to use Golden's words. The concept was to invite a researcher to cover an important and emerging topic and then solicit a few commentaries on that article from other experts in the field. The authors of the feature article would then respond to the commentaries in a rejoinder article. This ambitious but informative and engaging idea started in the first issue of volume 5 with the work of Richard Barr and Betty Hickman on parallel algorithms. Assembling the feature article–commentaries–rejoinder triads was a challenge, but the output was generally well worth the effort. These feature articles would remain a part of the JOC for over 21 years.

The next editor-in-chief, David Kelton, steered the journal through a number of critical initiatives, especially regarding web presence and on-line supplements. Although these are accepted facets of archiving research now, undergoing the paper-to-electronic conversion was rarely an easy challenge for any journal at the time. One also notices a few special issues appearing shortly after Kelton took over the journal: These issues served not only to explore new avenues for computing research but also to help build a pipeline of accepted articles for the journal. That backlog was healthy enough to assure JOC's long-term stability and help with the impact factor ratings. And indeed, with the groundwork that Golden had supplied, impact factors were established for JOC and quickly established the journal as one of the best in the field. In 2001, JOC's impact factor was 0.729, good for 8th out of 53 journals in the OR/MS category. By 2005, the impact factor grew to 1.762, ranking 1st among the 56 journals that were in this category.

Prakash Mirchandani was eminently well qualified to take leadership of JOC but had to step down after a few months as editor-in-chief to tend to matters more important than academics. After Kelton graciously served as an interim editor for the next issue, John Chinneck took over for the next (almost) 6 years starting in the summer of 2007. It was during this time that electronic submission was being introduced, an initiative that was fully implemented during Chinneck's term. Among the many important developments during Chinneck's term was the creation of the area Computational Biology and Medical Applications, now named Applications in Biology, Medicine, and Health Care. Establishing this area was a prescient move for the journal, as it has now grown into a major research focus for the community.

After 14 years of service with the journal, David Woodruff took over as editor-in-chief in 2013. During this time, JOC became entrenched in several lists of "A journals" (prestigious publications that would earn academic authors extra credibility for promotion and recognition), and both the volume of submissions and the number of published papers climbed. Under his leadership, the journal began

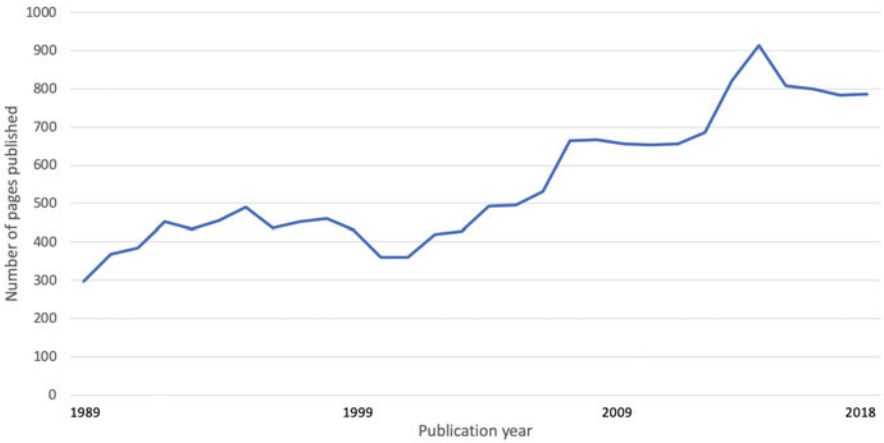


Fig. 5.1 Number of JOC pages published per year, 1989–2018

to publish data and code, occasionally requiring this material. This was perhaps a nonstandard requirement at the time but was forward-thinking in terms of ensuring reproducibility and validity of results. By the time his six full years as editor had expired, Woodruff turned over a prestigious and exceptionally healthy journal to Alice Smith, who started her term as editor-in-chief just months before the writing of this chapter.

Figure 5.1 shows the total number of pages published annually for the JOC from its inception in 1989 through 2018. JOC publishes roughly twice as many pages now as it did over its first several years of existence, hitting a peak of 914 pages in 2014.

5.2.2 JOC Areas and Their Editors

One especially interesting way to see how the focus and scope of the JOC have evolved is to study the evolution of the journal's areas over time. From the outset, the JOC has employed a three-tiered editorial structure consisting of the editor-in-chief, area editors, and associate editors. The initial areas envisioned for the journal consisted of:

- Cognitive Modeling and Analysis
- Computational Probability and Analysis
- Database Theory, Optimization and Integration
- Decision Support Systems
- Design and Analysis of Algorithms
- Fuzzy Systems
- Heuristic Search and Learning

– Information Storage and Retrieval
– Parallel Computation
– Representability and Computational Logic
– Simulation
– Telecommunications

(In looking at volume 1, issue 1 of the JOC, one also notes the method of contacting area editors: Only one, Jan Karel Lenstra, listed an email address, but five listed BITNET addresses, two listed CSNET addresses, and one listed an ARPANET address. Bearing in mind that this issue appeared in 1989, the area editors were ahead of their time.)

Some areas, though, underwent a set of name revisions and still exist in their new form today (through volume 31, issue 2). A few others ended at various points throughout the journal's history, either because of a shifting focus in the journal, emerging research areas intersecting the JOC's mission, or because contributions to those areas were simply spread among other areas of the journal.

Figures 5.2, 5.3, 5.4, 5.5, and 5.6 depict timelines associated with the JOC areas, including the times at which the areas began, changed names, or ended. The area editors are displayed above and below each area bar corresponding to the time at which these individuals served in their role. Figure 5.2 covers the four original areas that still exist through the most recent issue at the time of this writing. In particular, Computational Probability and Analysis kept its name for the entirety of the journal's history before being updated to Stochastic Models at the beginning of 2019. Heuristic Search and Learning is another area with considerable stability, dropping the "Learning" part of the title in mid-2013. Only Simulation exists with the same title today as it had in the first issue of JOC. The Telecommunications area added "Electronic Commerce" to its name in 2000, bringing along Ramayya Krishnan (who become President of INFORMS in 2019) to help nurture this side of the area.

Three other original areas persisted in the journal for several years before terminating or splitting, as shown in Fig. 5.3. The Design and Analysis of Algorithms area persisted in its original form for 30 years. The number of papers submitted to this area was impressively large, and in 2019, the area split into two areas: One each for continuous and discrete problems. Parallel Computation changed names to High-Performance Computation in the middle of 1998 (volume 10, issue 3), coinciding with an area editor change from Robert Meyer to Richard Barr. This area ended in 2003, diverting papers that might be sent to that area into alternative areas. Finally, Fig. 5.3 shows the path that Representability and Computational Logic took through its first three decades, changing names four times to stay current with the field (in sharp contrast to its stability with respect to area editors). The final iteration of this area, Constraint Programming and Hybrid Optimization, terminated at the beginning of 2019.

To maintain relevance in the areas covered by JOC, several new areas have been launched over the past three decades. Figure 5.4 depicts some of the areas that are still included in the journal now. Notably, two of these areas, Knowledge-

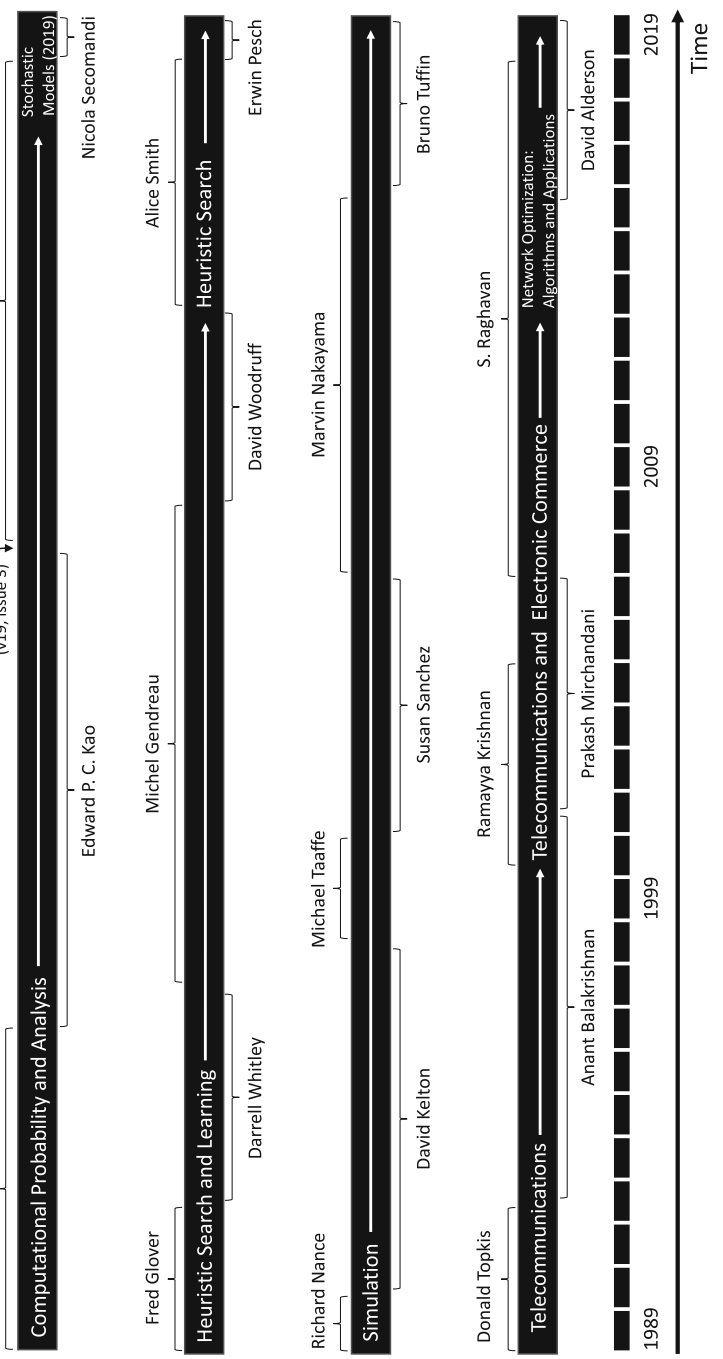


Fig. 5.2 Original JOC areas that remain in the journal today (volume 31, issue 2)

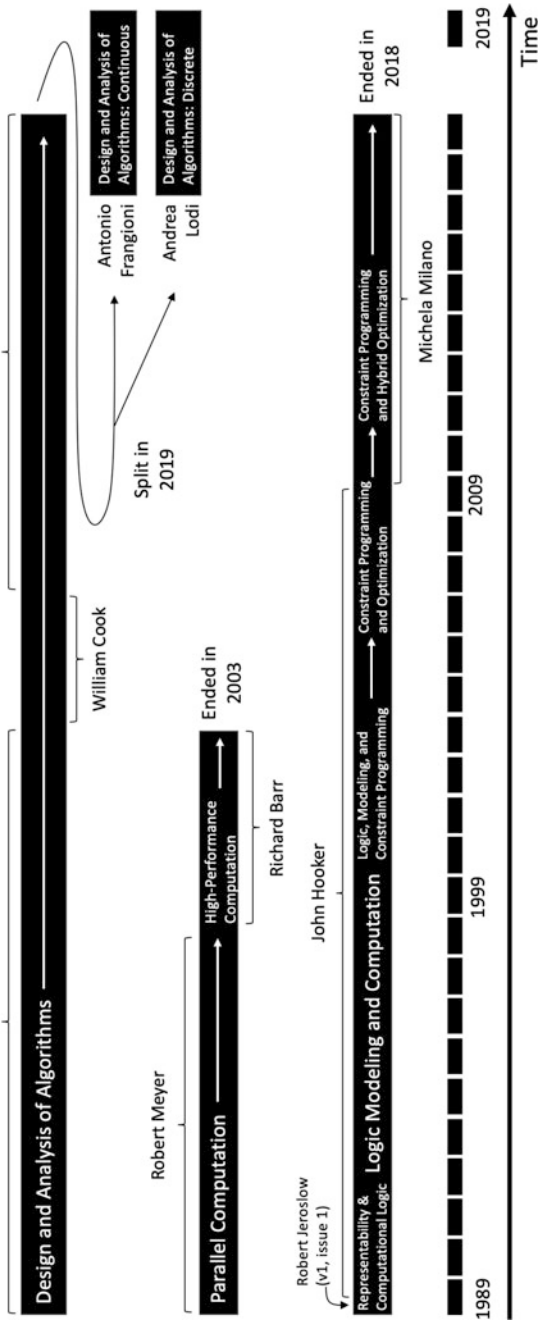


Fig. 5.3 Major areas of JOC that have been split or terminated

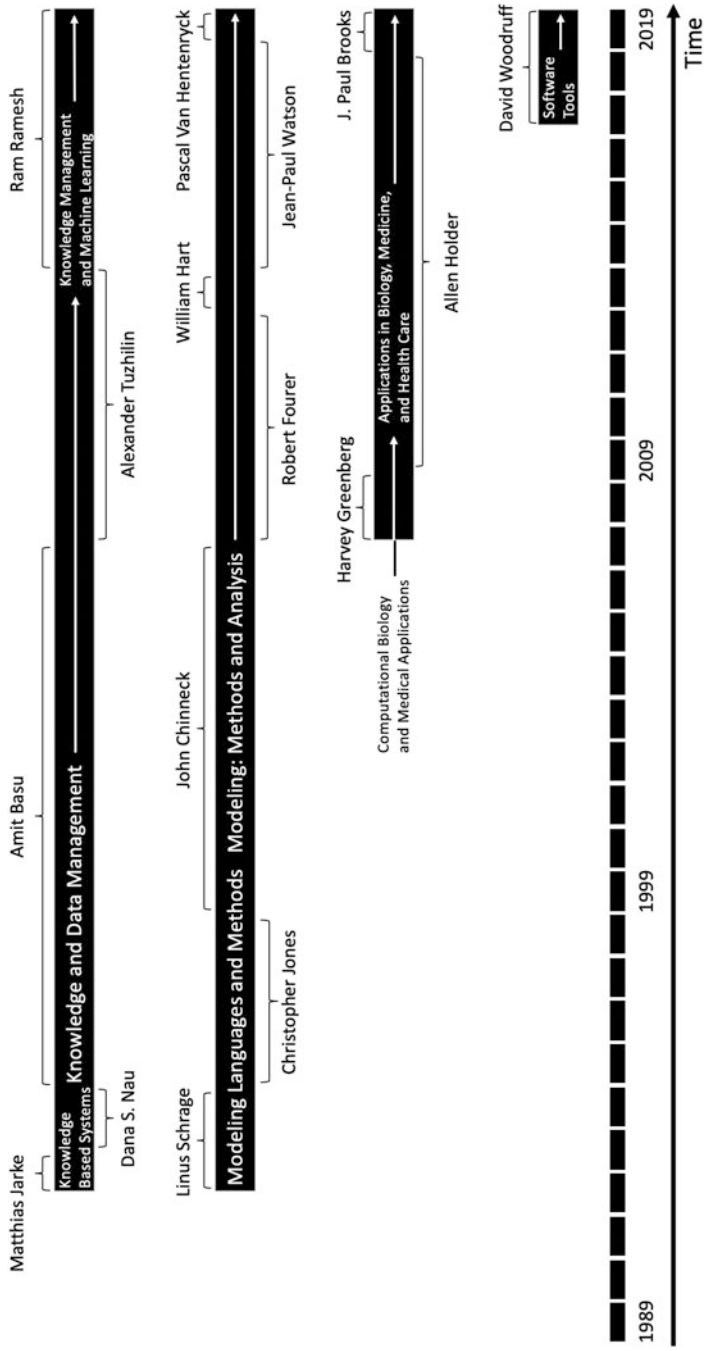


Fig. 5.4 Areas introduced after the original issue that currently remain in JOC

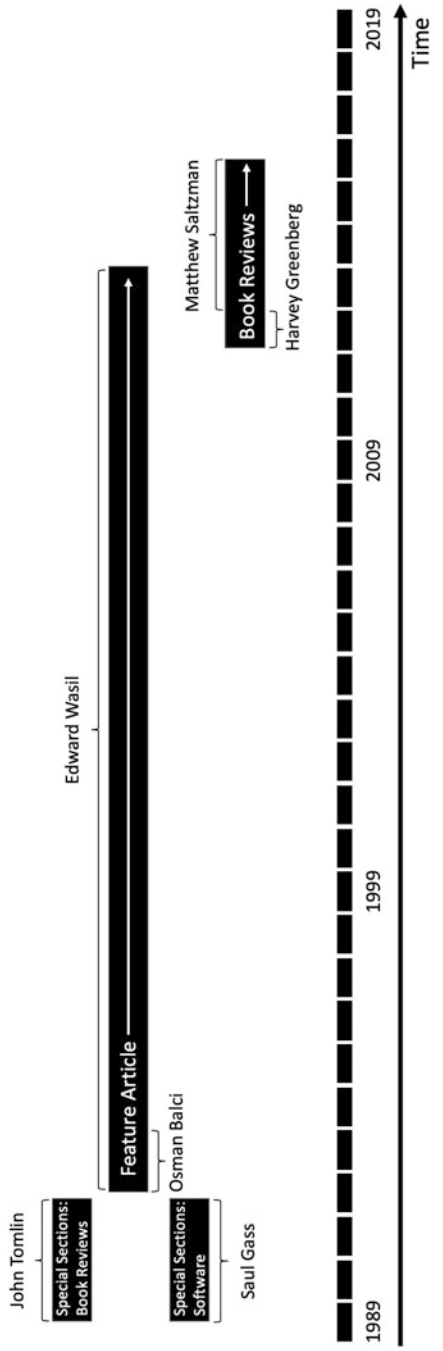


Fig. 5.5 Timeline of special sections and features in JOC

Based Systems (which is now Knowledge Management and Machine Learning) and Modeling Languages and Methods (now Modeling: Methods and Analysis) were started by Bruce Golden upon his arrival as editor-in-chief in 1992. Harvey Greenberg stayed very involved with JOC long after his departure as editor, and based on his vast research expertise in the area, established the Computational Biology and Medical Applications area in the last issue of 2007. He remained as area editor for six issues before passing this responsibility to Allen Holder, who held the position for a decade and renamed it as Applications in Biology, Medicine, and Health Care (with J. Paul Brooks serving as the current area editor). Finally, David Woodruff helped to realize his vision of the journal as editor-in-chief by originating an area for Software Tools starting in 2017, for which he remains area editor now.

A distinguishing feature of JOC is its capacity to explore and review topics in depth. This goal has arisen in various forms throughout the history of the journal, as shown in Fig. 5.5. Two special section areas, one on book reviews and the other on software, were present for all but the first two of JOC issues with Greenberg as editor-in-chief. The book reviews section was revived under Greenberg's leadership in 2012, with Matthew Saltzman serving three-and-a-half years as area editor following Greenberg until the area was closed. Edward Wasil (jointly with Osman Balci for six issues) served from 1992 to the end of 2013 as the Feature Article area editor.

Finally, Fig. 5.6 shows the short-lived areas that began with the journal's launch and were quickly reoriented and adjusted as the JOC found its footing. Interestingly,

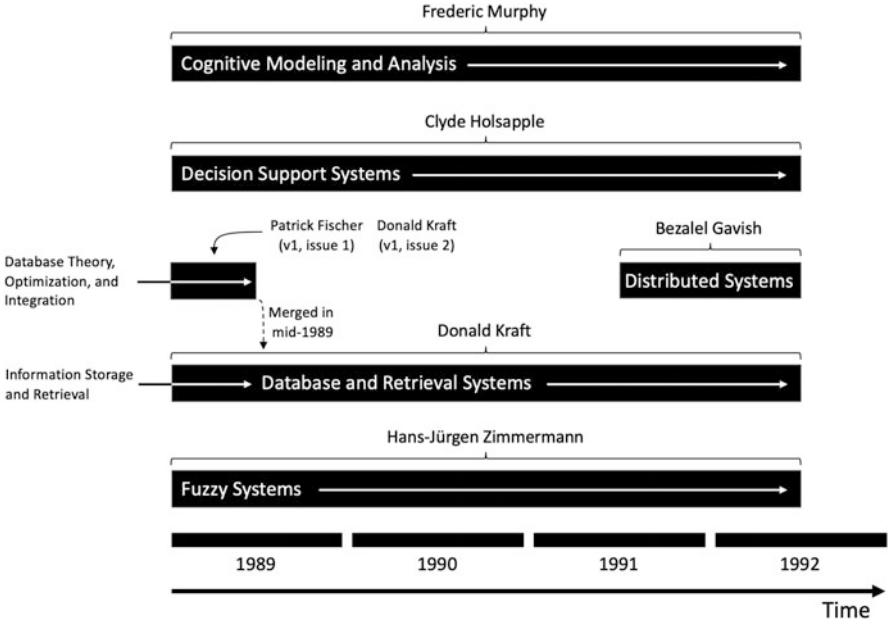


Fig. 5.6 Original journal areas terminated by 1992

many of these areas, such as Information Storage and Retrieval (later Database and Retrieval Systems), reflect the true *computing* origins of this journal.

5.3 Influence in Professional Societies

Two additional professional contributions due either partially or wholly from Greenberg's notable work in professional service are evident today. One is the INFORMS Computing Society and its associated activities, and the other is the *TutORials in Operations Research* book series. Sections 5.3.1 and 5.3.2 discuss these below.

5.3.1 The INFORMS Computing Society

ORSA contained a society known as the Computer Science Special Interest Group, which would grow into the Computer Science Technical Section, and then later the INFORMS Computing Society (ICS). The mission of this community is to be the INFORMS group responsible for research in the integration of computation and technology with operations research, management science, and analytics. Its mission statement includes taking a leading role in "computing and how it affects OR (e.g. XML modeling standards, OR services offered over the web, open source software, constraint programming, massively parallel computing, high performance computing)." Today, the ICS numbers 1653 members, hailing from (at least) 33 countries, including five Canadian provinces and 44 US states, in addition to Washington, DC.

But, it was a far smaller group several decades ago that launched not only the *Journal on Computing* but also a biennial conference and a set of awards recognizing research and service in the computing community.

The INFORMS Computing Society Conference With major annual conferences continuing to grow to once unimaginable numbers of participants, the role of subdivision and society conferences becomes ever greater. The INFORMS Computing Society conference typically numbers around 100–200 speakers. Because of the participants' common background in computing and OR/MS, the ICS conference enables presentations to be delivered in greater depth, while more effectively facilitating collaboration and networking activities. In the last two decades, this conference has been held in Knoxville, TN; Austin, TX; Richmond, VA; Santa Fe, NM; Monterey, CA; Charleston, SC; Coral Gables, FL; Annapolis, MD; Phoenix, AZ; and Cancún, MX. A proceedings is published that captures representative works presented at the conference periodically. The most recent of these was published in 2015 from the Richmond conference [6], containing an impressive 19 chapters.

Awards The ICS currently manages three awards for its membership. The first is the *ICS Student Paper Award*. From the description, that award "is given annually to the best paper at the interface of computing and operations research by a student author, as judged by the award selection committee." Because Greenberg loved supporting younger people in our community, it is inspiring to see the number and quality of papers submitted for this competition. The last six winners of this award were:

- 2019: Ryan Cory-Wright and Jean Pauphilet of the Massachusetts Institute of Technology for the paper, "A Unified Approach to Mixed-Integer Optimization: Nonlinear Formulations and Scalable Algorithms," advised by Dr. Dimitris Bertsimas.
- 2018: Aleksandr M. Kazachkov of Carnegie Mellon University for the paper, "V-Polyhedral Disjunctive Cuts," advised by Dr. Egon Balas.
- 2017: Berk Ustun of the Massachusetts Institute of Technology for the paper, "Learning Optimized Risk Scores from Large-Scale Datasets," advised by Dr. Cynthia Rudin.
- 2016: Georgina Hall of the Georgia Institute of Technology for the paper, "DC Decomposition of Nonconvex Polynomials with Algebraic Techniques," advised by Dr. Amir Ali Ahmadi.
- 2015: Young Woong Park of Northwestern University for the paper, "An Aggregate and Iterative Disaggregate Algorithm with Proven Optimality in Machine Learning," advised by Dr. Diego Klabjan.
- 2014: Kalyani Nagaraj of Virginia Tech for the paper, "Stochastically Constrained Simulation Optimization on Integer-Ordered Spaces: The cgR-SPLINE Algorithm," advised by Dr. Raghu Pasupathy.

The second of these awards is known simply as the *ICS Prize*, which is a best paper (or best group of papers) award at the interface of operations research and computer science. The most recent six winners of that prize are given below as well. Note the clear overlap between programming languages, computing, and optimization evident in these prizes: This perfectly reflects the original vision of the Computing Society.

- 2019: William E. Hart, Carl D. Laird, Jean-Paul Watson, David L. Woodruff, Gabriel A. Hackebeil, Bethany L. Nicholson, and John Siirola for spearheading the creation and advancement of Pyomo, an open-source software package for modeling and solving mathematical programs in Python.
- 2018: James V. Burke, Frank E. Curtis, Adrian S. Lewis, and Michael L. Overton for their pioneering work on gradient sampling methods for nonsmooth optimization.
- 2017: Shabbir Ahmed, George Nemhauser, and Juan Pablo Vielma for their pioneering work on mixed integer linear programming formulations for piecewise linear functions.
- 2016: Iain Dunning, Joey Huchette, and Miles Lubin for their development of the JuMP optimization package.

– 2015: Suvrajeet Sen, Dinakar Gade, Julia Higle, Simge Küçükyavuz, Lewis Ntaimo, and Hanif Sherali for their seminal work on stochastic mixed integer programming.
– 2014: Jim Ostrowski, Jeff Linderoth, Fabrizio Rossi, and Stefano Smriglio for their work on handling symmetry in combinatorial optimization problems.

The third of these awards is given out every other year, and it is a lifetime achievement award for service to the Computing Society. Appropriately, it is named the *Harvey J. Greenberg Award for Service*. The award began in 2009 and has honored six people so far whose service and dedication have shaped the direction of the ICS over their careers. These winners and a brief outline of their achievements are given below:

– **Dr. Karla Hoffman** (the 2009 winner) was a key figure in transforming the computing group at ORSA from a special interest group to a technical section within ORSA in 1980. She also helped in the organization of the first "computing society conference" in 1985 in Denver, CO, and served as a co-editor for the proceedings from that conference. Hoffman was also one of the founding influences in leading the launch of the *ORSA Journal on Computing* and a strong proponent of the computing community through its formative phase.
– **Dr. Bruce Golden** won the 2011 award in recognition of his efforts as editor-in-chief of the *Journal on Computing*, which he helped to grow during its nascent (and still uncertain) phase. Golden's efforts were vital in stabilizing the financial footing of the journal, gaining visibility and earning an impact factor, and creating a memorable redesign of the journal's cover (an accomplishment about which he remains proud). Golden also served as program co-chair for two of the computing society conferences (1989 and 2005).
– **Dr. Ramesh Sharda** won the 2013 award, largely in honor of his success in establishing the Computing Society conference. The first "official" Computer Science Technical Section conference was held in 1988 with Sharda as General Chair. His influence led to strong conference turnout, along with elite plenary speakers given by the likes of George Dantzig. His leadership during the initial meetings entrenched this conference as the regular high-profile meeting it remains today.
– **Richard S. Barr** won the 2015 award in honor of his consistent service to the Computing Society in virtually every phase of its operations. For the *Journal on Computing*, Barr was the Area Editor for High-Performance Computation. Barr served as Chair of the 1996 Computing Society conference in Dallas, the first held under the INFORMS banner. In terms of leadership specifically for ICS, Barr was elected as Chair for 1997 and 1998 and has since served as a member of ICS Prize Committee and of the ICS Board of Directors.
– **John W. Chinneck** won the 2017 award after serving two full terms as editor-in-chief of the *Journal on Computing* from 2007 to 2012, co-chairing the 2009 ICS Conference in Charleston, SC, and serving as the ICS chair from 2006 to 2007. It was under Chinneck's leadership that the Greenberg Award and the Student Paper Award were founded. Chinneck was also responsible for leading the journal into

new areas relevant to computing while also guiding the journal's transition to an on-line manuscript review and publication system.
– **Allen Holder** won the 2019 award for epitomizing what the award was created to recognize: the spirit of selfless volunteerism displayed in the taking on of largely thankless jobs to the benefit of the entire ICS community. One such contribution was in taking over the care of the Mathematical Programming Glossary from Greenberg, bringing it under the auspices of the ICS. Holder was also an important contributor to another years-long effort under the auspices of ICS—the Education Committee.

5.3.2 TutORials

This subsection starts with some personal reflection. I was invited in 2004 to serve as the Tutorials track chair for the 2005 INFORMS conference that was ultimately held in San Francisco. Except for the unenviable task of choosing four out of the 16 speakers that year to present on Wednesday (the last day) of the conference, it was an exciting role to play for a young professor. Tutorials at the INFORMS meeting are 90-min talks by exceptional OR/MS researchers. Some are given by accomplished senior personnel and others by mid-career researchers who have a knack for explaining emerging fields in a clear and compelling manner. They are designed to be accessible lectures. On more than one occasion, I attended a tutorial at INFORMS on a topic about which I was (ostensibly) an expert, just to hear how someone else would explain what I thought I knew so well. Getting an "orthogonal" understanding of the material is always valuable, as is the chance to fill in some additional gaps that would benefit my research career down the line.

Thus for the 2005 conference, I had a chance to shape the topics I wanted to learn more about. To get me started, Greenberg and I sat down in a small room in Denver at the 2004 INFORMS meeting. Greenberg introduced himself and congratulated me warmly on becoming the Tutorials track chair. Then with a sly grin (and surely knowing that the answer was negative), he asked me, "Did we tell you about the book?"

"The book" was an idea of Greenberg's dating back many years (and in fact attempted in some form several years prior to 2004). The Tutorials track is very popular at INFORMS, yet many people remark that they cannot find the time to go to the tutorials they wanted to see. Anyone attending a professional meeting understands the conflicting demands on time; thus, the proposal was made to somehow document these tutorials. In 2004 Greenberg addressed exactly this problem by assembling an edited book composed of chapters authored by a subset of the tutorial speakers. In his own words [11], the vision for this tutorials book was "to provide a reference for practitioners and academics who seek a clear, concise presentation of developing methodologies, hence providing themselves with the capability to apply these methods to new problems."

This edited volume [11] was a labor of love for him, as Greenberg did the recruiting of the authors, the screening and refereeing, and LaTeX typesetting for this volume. This collection consists of eight chapters:

- "Heuristic Search for Network Design," by I. Gamvros, B. Golden, S. Raghavan, and D. Stanojević
- "Polyhedral Combinatorics," by R. D. Carr and G. Konjevod
- "Constraint Languages for Combinatorial Optimization," by P. Van Hentenryck and L. Michel
- "A Tutorial on Radiation Oncology and Optimization," by A. Holder and B. Salter
- "Parallel Algorithm Design for Branch and Bound," by D. A. Bader, W. E. Hart, and C. A. Phillips
- "Computer-Aided Design for Electrical and Computer Engineering," by J. W. Chinneck, M. S. Nakhla, and Q. J. Zhang
- "Nonlinear Programming and Engineering Applications," by R. J. Vanderbei
- "Connecting MRP, MRP II and ERP—Supply Chain Production Planning Via Optimization Models," by S. Voss and D. L. Woodruff

From these chapters, it is easy to see three things. The first is Greenberg's knack for finding diverse and interesting topics that would excite the OR/MS community as a whole. Indeed, several of these chapters are relevant today, and several more were on the leading edge of research at the time. Two, there is a consistent focus on computing within these chapters (in fact, three of these authors had served or would serve as editor-in-chief of JOC as of 2019). Three, Greenberg's vast network of colleagues, students, and other friends he had made through the preceding three decades is clearly on display through these eight chapters.

By the time I took on the Tutorials track chair responsibility, Greenberg was excited to launch the first volume of the *TutORials in Operations Research* book series, published by INFORMS, where he would serve as the founding series editor-in-chief and I as the volume editor. Looking back on that volume, there were several chapters that remain among my favorites in the series today. The process of reading the chapters and, in some cases, converting them to LaTeX, gave me the perfect excuse to read what my colleagues were working on. In terms of the book series, that first chapter was essential in determining where the challenges would arise in producing these volumes each year. The most pressing is the need to review the chapters and get them to the publishers in time for the book to appear by the time of the INFORMS Annual Meeting. It is common to see special issue deadlines for journals to have three-month extensions or longer, and the idea of hard deadlines for papers goes somewhat against the culture of our field.

Harvey steered this book series initially, before handing it off to Paul Gray as series editor in 2006. Gray served two full terms, with his last term ending after the 2011 volume. It was Gray who addressed many of the foundational challenges associated with this volume and stabilized its presence and long-term viability at the conference. During his tenure as series editor, Gray set expectations on procuring more chapters per volume, seeing that these chapters were refereed, and analyzing

Table 5.2 Summary of *TutORials* volumes

Year	Theme	Volume editor(s)
2005	Emerging theory, methods, and applications	J. Cole Smith
2006	Models, methods, and applications for innovative decision making	Michael P. Johnson, Bryan Norman, and Nicola Secomandi
2007	OR tools and applications: glimpses of future technologies	Ted Klastorin
2008	State-of-the-art decision-making tools in the information-intensive age	Zhi-Long Chen and S. Raghavan
2009	Decision technologies and applications	Mohammad R. Oskoorouchi
2010	Risk and optimization in an uncertain world	John J. Hasenbein
2011	Transforming research into action	Joseph Geunes
2012	New directions in informatics, optimization, logistics, and production	Pitu B. Mirchandani
2013	Theory driven by influential applications	Huseyin Topaloglu
2014	Bridging data and decisions	Alexandra M. Newman and Janny Leung
2015	The operations research revolution	Dionne M. Aleman and Aurélie C Thiele
2016	Optimization challenges in complex, networked, and risky systems	Aparna Gupta and Agostino Capponi
2017	Leading developments from INFORMS communities	Rajan Batta and Jiming Peng
2018	Recent advances in optimization and modeling of contemporary problems	Esma Gel and Lewis Ntaimo
2019	Operations research and management science in the age of analytics	Serguei Netessine

how the volumes could be more broadly disseminated. A CD of the volumes was produced for several years, before it became more practical to simply offer the chapters electronically through the INFORMS website. I took two terms as series editor from 2012 to 2017 and have been succeeded by Doug Shier starting in 2018.

Each volume has its own theme, either linked to the conference theme, based on the volume editors' creative direction or stated generically enough to encompass the breadth of chapters that appear in the volume. Table 5.2 lists the volumes in print so far, along with the corresponding volume editors.

Most volumes have about ten chapters, with two notable exceptions. In 2008 (for the Washington, DC INFORMS conference), Zhi-Long Chen and Raghu Raghavan managed to procure 15 chapters, a feat duplicated by Aparna Gupta and Agostino Capponi in 2016. In the 14 volumes published by INFORMS between 2005 and 2014, plus the one published by Springer in 2004, there are a total of 156 tutorial chapters appearing in the volumes. I could not do justice to a list of the most

impactful or best written chapters in this series, but below are ten of the many that appeal to me because of my own research and personal interests in the field.

– The very first chapter of volume 1 includes an article by Hicks et al. [13] on branch and tree decomposition techniques. As volume editor, I was a little apprehensive about having a more technical chapter appear, because of my worry about the accessibility of this work to a large audience. After reading it, though, I became so interested in it that I began a collaboration with two of the authors of this chapter (and with a third one a decade later). This in fact is a clear work that helps anyone with a background in graphs understand the area of branch and tree decompositions. With the popularity of binary decision diagrams today, it is useful background for researchers examining combinatorial structures that may enable elegant algorithms for difficult problems.
– In 2009, Dr. Suvrajeet Sen asked me to present a 90-min presentation on robust optimization to a group of researchers at a workshop in Banff. I agreed without thinking further about it, although I later realized that the workshop crowd would consist of extraordinarily accomplished researchers and that I did not really know robust optimization very well. This is where Bertsimas and Thiele's [5] tutorial article helped orient my efforts. The first two chapters clearly lay out the concept of (static) robust optimization, how a robust counterpart is formulated, and how one might use this in applications like portfolio optimization. From there, much of the rest of this literature becomes far more accessible. They cover dynamic optimization in their paper as well, a topic that would be expanded in great depth in Delage and Iancu's [8] excellent tutorial on multistage robust optimization almost a decade later.
– I was familiar with the basics of chance-constrained programming by the time Ahmed and Shapiro's 2008 work [2] was published, and how these problems could be approximately modeled using stochastic programming. This chapter explained to me some key convergence properties of Sample Average Approximation in language that I was able to understand given my limited depth of knowledge at the time. Their work helped me in a paper that I would write later with Ahmed and with Dr. Siqian Shen, a PhD student at the time who became an expert in this very field.
– Assessing the quality of a solution within stochastic programming was covered by Bayraksan and Morton [4] in a tutorial the very next year. The concept is that it is possible to determine point and interval estimates on the optimality gap with respect to a feasible solution to a stochastic program and then leverage those bounds within an exact optimization scheme. The topic is inherently complex, but the authors deliver an exceptionally accessible treatment of the material with direct relevance for optimizers.
– As someone with an inherent interest in history, I was very excited to read Gass and Assad's [9] brief chapter on the history of operations research. I loved the stories my advisor, Dr. Hanif Sherali, would tell our class about OR development, simply because the history was recent and the characters relatable. Quoting the authors in [9], "Many of its developers are still alive and records

of their accomplishments are available from them and/or from colleagues and friends. Similarly, for those who have passed on, writings and reports of their OR activities are still reasonably accessible, and can be amplified with the memories of close collaborators or friends." Few in our field are as well equipped to tell the history of OR like Gass and Assad, and this chapter is a treasure in our field.

- Roughly a decade before Alessandro Agnetis's chapter [1] was published for the 2012 INFORMS Annual Conference, Dr. Pitu Mirchandani (also the volume editor of the 2012 book and that year's Tutorials chair) introduced me to Agnetis and to his work on competitive scheduling. This chapter shows the applications of multiagent scheduling along with several complexity results. The complexity results presented in this chapter are truly comprehensive. For those wanting a practical example of multiagent scheduling, though, the focused examination of scheduling problems in a leading industrial district in Tuscany, Italy, is certainly worth reading.
- The tutorial of Alderson et al. [3] is on risk, resilience, interdiction, fortification, and applications of these concepts. This is not the first tutorial in this general area, but the depth and completeness of the story told here is impressive. These authors have a particular knack for translating a complex application into one whose pieces can more easily be comprehended, which is perfect for a tutorial chapter. This tutorial chapter appeals to practitioners and academicians alike, an achievement that is possible because the authors have rich experience in real-world development and implementation of optimization risk-assessment and decision-making models pertaining to critical infrastructure systems.
- The theme of the chapter mentioned above was continued by Dr. Laura Albert [16] in her work on disaster preparedness and recovery. Her work splits the field into vulnerability analysis, mitigation, preparedness, emergency response, and recovery in response to disasters. The chapter's treatment of this material uses a broad spectrum of approaches ranging from stochastic models to mathematical optimization and draws from the author's research in screening security (especially for air transportation and cargo application), ambulance pre-positioning, and location-allocation models for disaster recovery.
- The operations research community's intersection with other fields occasionally results in overlapping discoveries written in slightly different languages. These parallel developments represent a potential missed opportunity in integrating discoveries that could afford deeper knowledge in interdisciplinary fields. I attended a workshop at the Rutgers University in 2012 on "A Conversation Between Computer Science and Operations Research on Stochastic Optimization," hosted by Santinder Singh and Warren Powell, intended to help each field understand one discipline's contributions in a common context. Powell's work in [17] is a continuation of this effort. This massive and ambitious chapter is notable for its efforts in linking stochastic programming, optimal control, dynamic programming, and other fields.
- Last and most recent is a chapter by Brooks and Holder [7] on OR challenges that arise in metabolic networks, an area in the general field of computational biology that Greenberg helped to develop and promote. This tutorial touts the importance

of deep interdisciplinary collaborations, as opposed to one disciplinary expert simply executing standard tools out of their own discipline to apply across other fields. In this case, interdisciplinary collaboration is difficult because the biological field itself is so complex. OR researchers trying to make contributions in this field simply must do so in collaboration with others. As Brooks and Holder attest regarding this difficulty, "those in OR should know that biology is a rapidly changing science—so much so that biologists themselves are constantly facing the same sense of overwhelming unfamiliarity." Their chapter is a perfect example of the computational and applied OR focus that Greenberg and his colleagues sought to promote throughout their careers.

5.4 Summary and Acknowledgments

Having been born too late to recall many of the events in this chapter, I sincerely appreciate the comments of many colleagues who helped point me to the literature, gave extended reflections, or called me to recount some of these stories. I will inevitably overlook some help that I received, but I particularly wish to thank Gerald Brown, John Chinneck, Bruce Golden, Al Holder, David Kelton, Alice Smith, and David Woodruff for their help.

Al Holder, in his moving tribute in [14], states that Greenberg "was an affable and gracious friend to many, and while he targeted magnanimous pursuits in knowledge, education, and service, he cherished the camaraderie of the quest. Working or studying with Harvey could be exhilarating, tense, friendly, fun, tiring, and acute." This recollection dovetails perfectly with everyone who communicated with me for the purposes of this chapter (and many who just very fondly remembered him). A recurring trait mentioned by those who knew him best regards his sincerity and his willingness to collaborate with, mentor, or assist anyone who needed it. This chapter hopefully provides the reader a sense of what his leadership helped to bring to his professional communities and his colleagues over his many decades of work in the field. Perhaps it will also convince the reader of the very real possibility for making impactful changes in complex organizations, given the right amount of patience, stubbornness, and dedication to the profession.

References

1. A. Agnetis, Multiagent scheduling problems, in *TutORials in Operations Research: New Directions in Informatics, Optimization, Logistics, and Production*, ed. by P.B. Mirchandani (INFORMS, Catonsville, 2012), pp. 151–170
2. S. Ahmed, A. Shapiro, Solving chance-constrained stochastic programs via sampling and integer programming, in *TutORials in Operations Research: State-of-the-Art Decision-Making Tools in the Information-Intensive Age*, ed. by Z.L. Chen, S. Raghavan (INFORMS, Catonsville, 2008), pp. 261–269

3. D.L. Alderson, G.G. Brown, W.M. Carlyle, Assessing and improving operational resilience of critical infrastructures and other systems, in *TutORials in Operations Research: Bridging Data and Decisions*, ed. by A. Newman, J. Leung (INFORMS, Catonsville, 2014), pp. 180–215
4. G. Bayraksan, D.P. Morton, Assessing solution quality in stochastic programs via sampling, in *TutORials in Operations Research: Decision Technologies and Applications*, ed. by M.R. Oskoorouchi (INFORMS, Catonsville, 2009), pp. 102–122
5. Bertsimas, D., Thiele, A.: Robust and data-driven optimization: modern decision making under uncertainty, in *TutORials in Operations Research: Models, Methods, and Applications for Innovative Decision Making*, ed. by M.P. Johnson, B. Norman, N. Secomandi (INFORMS, Catonsville, 2006), pp. 95–122
6. B. Borchers, J.P. Brooks, L. McLay (eds.), *Operations Research and Computing: Algorithms and Software for Analytics* (INFORMS, Catonsville, 2015)
7. J.P. Brooks, A. Holder, Metabolic networks and modern research problems in operations research, in *TutORials in Operations Research: Leading Developments from INFORMS Communities*, ed. by R. Batta, J. Peng (INFORMS, Catonsville, 2017), pp. 115–130
8. E. Delage, D.A. Iancu, Robust multistage decision making, in *TutORials in Operations Research: The Operations Research Revolution*, ed. by D.M. Aleman, A.C. Thiele (INFORMS, Catonsville, 2015), pp. 20–46
9. S.I. Gass, A.A. Assad, History of operations research, in *TutORials in Operations Research: Transforming Research into Action*, ed. by J. Geunes (INFORMS, Catonsville, 2011), pp. 1–14
10. F. Glover, Tabu search, part I. ORSA J. Comput. **1**(3), 190–206 (1989)
11. Greenberg, H.J. (ed.): *Tutorials on Emerging Methodologies and Applications in Operations Research* (Springer, New York, 2004)
12. H.J. Greenberg, A personal history of ICS. ORMS Today **35**(5) (2006). https://www.informs.org/ORMS-Today/Archived-Issues/2006/orms-10-06/A-Personal-History-of-ICS
13. I.V. Hicks, A.M.C.A. Koster, E. Kolotoglu, Branch and tree decomposition techniques for discrete optimization, in *TutORials in Operations Research: Emerging Theory, Methods, and Applications*, ed. by J.C. Smith (INFORMS, Catonsville, 2005), pp. 1–29
14. A. Holder, F. Murphy, W. Pierskalla, A memorial to Harvey J. Greenberg, founding editor of the *INFORMS Journal on Computing*. INFORMS J. Comput. **30**(3), 421–423 (2018)
15. P. Horner, History lesson: the evolution of INFORMS. ORMS Today **44**(1) (2007). https://www.informs.org/ORMS-Today/Public-Articles/February-Volume-44-Number-1/History-Lesson-The-evolution-of-INFORMS
16. L.A. McLay, Discrete optimization models for homeland security and disaster management, in *TutORials in Operations Research: The Operations Research Revolution*, ed. by D.M. Aleman, A.C. Thiele (INFORMS, Catonsville, 2015), pp. 111–132
17. W.B. Powell, A unified framework for optimization under uncertainty, in *TutORials in Operations Research: Optimization Challenges in Complex, Networked and Risky Systems*, ed. by A. Gupta, A. Capponi (INFORMS, Catonsville, 2016), pp. 45–83

Chapter 6
Parametric Stochastic Programming with One Chance Constraint: Gaining Insights from Response Space Analysis

Harvey J. Greenberg, Jean-Paul Watson, and David L. Woodruff

Abstract We consider stochastic programs with discrete scenario probabilities where scenario-specific constraints must hold with some probability, which we vary parametrically. We thus obtain minimum cost as a function of constraint-satisfaction probability. We characterize this trade-off using Everett's response space and introduce an efficient construction of the response space frontier based on tangential approximation, a method introduced for one specified right-hand side. Generated points in the response space are optimal for a finite set of probabilities, with Lagrangian bounds equal to the piece-wise linear functional value. We apply our procedures to a number of illustrative stochastic mixed-integer programming models, emphasizing insights obtained and tactics for gaining more information about the trade-off between solution cost and probability of scenario satisfaction. Our code is an extension of the PySP stochastic programming library, included with the Pyomo (*Python Optimization Modeling Objects*) open-source optimization library.

Electronic Supplementary Material: The online version of this chapter (https://doi.org/10.1007/978-3-030-56429-2_6) contains supplementary material, which is available to authorized users.

The author "Harvey J. Greenberg" is deceased at the time of publication.

H. J. Greenberg
Mathematics Department, University of Colorado, Denver, CO, USA

J.-P. Watson (✉)
Center for Applied and Scientific Computing and Global Security Directorate, Lawrence Livermore National Laboratory, Livermore, CA, USA
e-mail: jeanpaulwatson@llnl.gov

D. L. Woodruff
Graduate School of Management, University of California Davis, Davis, CA, USA
e-mail: dlwoodruff@ucdavis.edu

A. Holder (ed.), *Harvey J. Greenberg*, International Series in Operations Research & Management Science 295, https://doi.org/10.1007/978-3-030-56429-2_6

6.1 Introduction and Background

We consider stochastic programs with discrete probabilities where one or more constraints must hold jointly with some probability, β, which we vary parametrically between zero and one. We refer to the probability requirement as a *chance constraint* [20, 25]. Chance constraints make sense in many settings for various reasons, among them: (1) when constraints represent adherence to policies rather than laws of physics, it may be deemed too expensive to comply with all constraints under all circumstances; (2) when the discrete probabilities are the result of sampling from continuous distributions or from simulation realizations, it is simply a form of false advertising to claim that constraints will hold with probability 1, so it may make sense to relax away from 1 under the control of a parameter. The usefulness of chance constraints has led to a large body of research directed at solving these sorts of problems for a given value of β (see, e.g., [1, 20, 21, 24, 30, 31]).

Let $S = \{1, \ldots, N_S\}$ represent the set of scenario indexes. Each of the N_S scenarios gives a full set of the data for a constrained minimization problem, and we associate the symbol p_s with the probability that scenario $s \in S$ will be realized, where $\sum_{s\in S} p_s = 1$. Following [28], we assume that the problem formulation includes N_S binary variables, δ, that take the value one if there must be compliance with scenario-s constraints. Although $\delta_s = 0$ allows violation (i.e., non-compliance), we discuss a model extension to allow the converse: $\delta_s = 0$ only if some scenario-s constraint is violated (see Sect. 6.6.2).

There are different formulations that fit under this rubric. For example, consider a two-stage, chance-constrained, stochastic program where the first-stage variables, x, are constrained to be in a set X. The second-stage variables, $\{y_s\}_{s\in S}$, are constrained by $y_s \in Y_s(x, \delta)$. In particular, suppose the function to be minimized is $c(x) + \sum_{s\in S} p_s h_s(x, y_s)$, where c and $\{h_s\}_{s\in S}$ are functionals and

$$Y_s(x, \delta) = \{y_s \in \mathcal{Y}^s : A_s x + B_s y_s \geq \delta_s d_s - (1 - \delta_s) M_s\}, \text{ for } x \in X, \tag{6.1}$$

where M_s is sufficiently large to render scenario-s constraints redundant for $\delta_s = 0$; $\mathcal{Y}^s$ may be simply $\mathbb{R}^{m_s}$ or it may constrain some variables to be integer-valued.

Only a proper subset of the constraints form the joint chance constraint in some applications. In order to capture a wide range of chance-constrained models, we express the general idea by using $z^*(\delta)$ to represent the result of solving the extended minimization problem with an indicator vector, δ. We thus define the chance-constrained problem as:

$$\text{CC}: \quad \min z^*(\delta) : p\delta \geq \beta, \ \delta \in \{0, 1\}^n, \tag{6.2}$$

where $p\delta \stackrel{\text{def}}{=} \sum_{s\in S} p_s \delta_s$.

We think of CC computationally as a decomposition with an outer problem to select scenarios by setting their corresponding $\delta_s = 1$; the inner problem is

what defines z^*. Specifically, the two-stage stochastic program with joint chance constraints uses Y_s as defined in (6.1) to obtain

$$z^*(\delta) = \min\left\{c(x) + \sum_{s\in S} p_s h_s(x, y_s) : x \in X,\ y_s \in Y_s(x, \delta),\ \forall s \in S\right\}.$$

To compute solutions under parametric variation of β, we form the Lagrangian of CC:

$$L^*(\lambda) \stackrel{\text{def}}{=} \min\left\{z^*(\delta) - \lambda p\delta : \delta \in \{0,\ 1\}^n\right\}. \tag{6.3}$$

Each Lagrangian gives a lower bound on the minimum cost:

$$f^*(\beta) \stackrel{\text{def}}{=} \min\{z^*(\delta) : p\delta \geq \beta,\ \delta \in \{0,\ 1\}^n\} \ \geq\ L^*(\lambda) + \lambda\beta. \tag{6.4}$$

The optimal multiplier, λ^*, gives the tightest bound:

$$L^*(\lambda^*) + \lambda^*\beta = \max_{\lambda\geq 0}\{L^*(\lambda) + \lambda\beta\},$$

which is the weak Lagrangian dual. The Lagrangian gap is the difference in optimal objective values:

$$G(\beta) \stackrel{\text{def}}{=} f^*(\beta) - (L^*(\lambda^*) + \lambda^*\beta).$$

Let $\delta^* \in \text{argmin}\{z^*(\delta) - p\delta : \delta \in \{0,\ 1\}^n\}$. We have $G(\beta) = 0$ if, and only if, complementary slackness holds: $\lambda^* > 0 \Rightarrow p\delta^* = \beta$. This follows from $f^*(\beta) = z^*(\delta^*)$, and hence $G(\beta) = \lambda^*(p\delta^* - \beta)$.

If $\beta = 0$, no scenarios need to be selected, so $\delta = 0$ is optimal and $\lambda = 0$ is an optimal multiplier. Otherwise, if the optimal solution satisfies $p\delta^* = \beta$, then it solves the original problem (6.2). In the more typical cases, either the probabilities are such that there is no vector $\delta \in \{0, 1\}^n$ for which $p\delta = \beta$, or such vectors are suboptimal. There are two alternative Lagrangian optima in these cases, δ^L and δ^U, such that $b^L = p\delta^L < \beta < p\delta^U = b^U$. The interval (b^L, b^U) is called the *gap region*.

The best feasible solution corresponds to b^U, with min-cost $z^U = z^*(\delta^U)$. The Lagrangian duality gap is bounded by

$$G(\beta) = f^*(\beta) - \left(L^*(\lambda^*) + \lambda^*\beta\right) = f^*(\beta) - \left(z^U - \lambda^* b^U + \lambda^*\beta\right) \leq \lambda^*\left(b^U - \beta\right),$$

where the last inequality follows from the fact that $\beta < b^U \Rightarrow f^*(\beta) \leq f^*(b^U) = z^U$. If we think of λ^* as a unit price, then the bound value is the total cost of the discrepancy, $b^U - \beta$. We use a dimensionless measure of solution quality, called the *relative Lagrangian gap*:

$$g(\beta) = \frac{\lambda^* \left(b^U - \beta\right)}{z^U}. \tag{6.5}$$

While our main goal is to use a chance-constraint stochastic programming model in support of decision-making, we go beyond the model and algorithm descriptions by emphasizing a maxim of good decision support: *The purpose of mathematical programming is insight, not numbers*[6]. We envision an environment where the mathematical program without the chance constraint is computationally difficult, so a best algorithm is one that needs the fewest Lagrangian solutions. Furthermore, we see the user as an analyst who wants to see a broad range of the efficient frontier, f^*, but not necessarily those points that add significant computational difficulty. Thus, seeing the convex envelope, F^*, presents a useful graph in its own right. Besides the generated points, where $f^* = F^*$, we provide a visual of how close the cost is for some particular β. The user can then choose regions for which the gap, $f^* - F^*$, needs to be tightened. The restricted flipping heuristic offers a framework for doing this, and the analyst could specify regions of search or use our automatic search based on uncertainty measured by the length of the gap interval, $b^U - b^L$.

There are cases where a probability is (or appears to be) specified. For example, consider the case of a government regulation on sulfur emissions. A company may want parametric analysis to substantiate a challenge based on how much the regulation costs, particularly if a small relaxation of the regulation costs much less. The government may want to analyze consideration of a tax that incentivizes compliance with the impact of keeping emissions and cost low. The Lagrange multipliers provide bounds on a tax that associates cost with compliance probability. (See LP Myth 23 in [12] to avoid seeing the tax as equivalent to the optimal multiplier.)

The rest of this chapter is organized as follows. The response space in which trade-offs are displayed is defined in Sect. 6.2. An algorithm that finds the optimal Lagrange multiplier is described in Sect. 6.3. Some of our computational search can be mitigated by the pre-processing methods in Sect. 6.4, and we emphasize the insight that tells us when a scenario must be selected. Examples based on instances of three models are given in the Supplementary Material for this chapter (https://github.com/DLWoodruff/GWW). These are used to illustrate methods for finding additional points in the response space in Sect. 6.5. Section 6.6 provides information about details that arise when implementing algorithms that map the trade-offs between probability and cost. The chapter closes with a summary and conclusions. The methods described in this chapter have been implemented as an extension to the PySP stochastic programming library [32], which is distributed as part of the Pyomo [16, 17] algebraic modeling language.

6.2 Response Space Analysis

Everett's seminal paper [5] introduced the Payoff-Resource (PR) space, which is the range of the objective and constraint functions. His mathematical program was a maximization of a payoff subject to resource limits. Our model is a minimization of cost subject to a probability of scenario satisfaction, so we call it more simply the response space (RS). (See *Mathematical Programming Glossary* [18].) He also introduced the term "gap," which is now entrenched in our vocabulary, to mean the difference between the primal and dual objective values. There was a stream of foundational papers that deepened our knowledge of general (non-convex) duals based on Everett's Generalized Lagrange Multiplier method (GLM)—see [2, 4, 7–9, 15, 29, 33]. We present the main concepts focused on one joint chance constraint with uncertainties that can be involved in both the left-hand side matrix and the right-hand side vector. Our purpose is to elucidate the results, particularly the search for an optimal Lagrange multiplier and the source of a Lagrangian duality gap, to gain insight.

The set of feasible right-hand sides for the chance-constraint problem is $B = [0, P^{\max}]$. For now, assume $P^{\max} = 1$. Since scenarios may compete for common resources it may not be possible to achieve $P^{\max} = 1$, so it is important to consider $P^{\max} < 1$, and it may be the reason for a chance-constraint model. However, in the interest of clarity, we defer this point until after we present the main results.

The response space compares the range of probability to cost over scenario-selection values, δ:

$$\text{RS} = \{(b, z): b = p\delta,\ z = z^*(\delta) \text{ for some } \delta \in \{0,\ 1\}^n\}. \tag{6.6}$$

It is helpful to realize that each Lagrangian contour in response space is a line, regardless of the structure of decision space and objective function. Furthermore, the transition from decision space to response space makes evident that the maximum-Lagrangian is the *convex envelope function*, F^* (also called the second convex conjugate of f^*) [14]:

$$F^*(\beta) = \max_{\lambda \geq 0} \min_{b \in B} \{f^*(b) - \lambda b + \lambda \beta\}. \tag{6.7}$$

The epigraph of $[F^*, B]$ is geometrically the closed convex hull of the epigraph of (f^*, B), denoted by

$$\text{epi}(F^*, B) = \text{convh}(\text{epi}(f^*, B)). \tag{6.8}$$

Figure 6.1 illustrates this epigraph, where each point is the probability, $b = p\delta$, and cost, $z = z^*(\delta)$. Each line supports its epigraph:

$$\text{epi}(F^*, B) = \{(b, z): b \in B,\ z \geq F^*(b)\}. \tag{6.9}$$

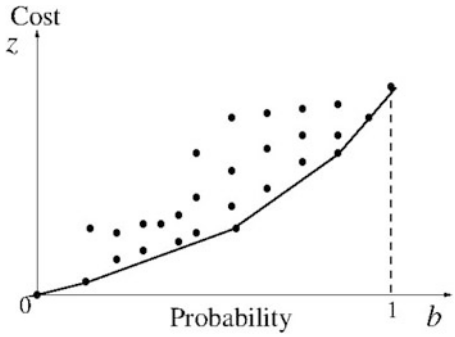


Fig. 6.1 Response Space as the range of $\delta \rightarrow \big(b = p\delta,\ z = z^*(\delta)\big)$

At each change in slope, the (b, z) point corresponds to an integer optimum that defines endpoints of the line segment whose slope is the Lagrange multiplier that produces the support for epi(f^*, B). If $\beta^* \in (b^L, b^U)$ (i.e., not one of the endpoints), then it is theoretically possible to find it, but to do so requires an enumeration of alternative optimal δ values. Only the endpoints are generated because they have alternative optimal multipliers. (Note that it is possible that an initial solution happens to obtain $(\beta, f^*(\beta))$, but once the iterations begin, only the endpoints are generated.) Thus, every $\beta \in (b^L, b^U)$ is essentially in a gap even though the gap value may be zero.

Because the Lagrangian approach provides a decomposition of scenarios, we can fit it into the PySP framework by simply adjusting the stage-two objective function to include the Lagrangian penalty cost. Luedtke [22] takes an alternative decomposition approach designed to obtain points on $[f^*, B]$, the efficient frontier of cost versus probability. Our Lagrangian approach focuses on computational efficiency by first obtaining points on $[F^*, B]$, followed by exploratory analysis of RS that includes sub-optimal solutions.

Here is a summary of the main points about response space.

- Each point in decision space, $\delta \in \{0, 1\}^{N_S}$, maps to a point in response space, $(b, z) \in$ RS.
- A Lagrangian contour in RS is a line with slope = λ.
- The bound, $f^*(\beta) \geq L^*(\lambda) + \lambda\beta$, is the support-line value at $b = \beta$.
- The Lagrangian dual gives the tightest Lagrangian bound, $\lambda^* \in \text{argmax}\{L^*(\lambda) + \lambda\beta\}$.
- The optimal multiplier, λ^*, is unique if, and only if, β is in a gap, in which case $\beta \in (b^L, b^U)$ and $\lambda^* = \dfrac{z^U - z^L}{b^U - b^L}$.

6.3 Multiplier Search

We now review the method of tangential approximation [10] to find an optimal Lagrange multiplier and then extend it to find the entire envelope function.

6.3.1 Search for One Optimal Multiplier

There are several ways to search for one optimal Lagrange multiplier, but tangential approximation was proposed as an efficient scheme [10]. For CC, it converges finitely to λ^* whether β is in a gap or not.

The general class of interval reduction algorithms includes bisection and linear interpolation, analyzed in [10]. Unlike tangential approximation, they are not guaranteed to converge finitely although it is possible to construct numerical examples for which they converge immediately. For example, suppose the initial interval of the multiplier search is $\lambda \in (0, \lambda^{\max})$ and $\lambda^* = \lambda^{\max}/2$. If we assume β is in a gap, which is likely in our binary model, then the optimal multiplier is unique—only extreme values of (b, z) yield a range, $\lambda^* \in [\lambda^L, \lambda^U]$ for $\beta \in (b^L, b^U)$. The multipliers are the left and right derivatives of F^*, respectively:

$$\lambda^L = \frac{\partial^- F^*(\beta)}{\partial \beta} \leq \frac{\partial^+ F^*(\beta)}{\partial \beta} = \lambda^U. \tag{6.10}$$

One optimal search for λ^* is Fibonacci, which minimizes the maximum number of functional evaluations (i.e., Lagrange solutions). One problem is with initialization: setting $\lambda^U = \infty$ (some big number). Another problem is getting close to λ^* but not converging finitely, in which case the computed gap region could be much wider than the actual value.

The tangential approximation search for one optimal multiplier, $\lambda^*(\beta)$, begins with the search intervals $(0, z^*(\vec{0}))$ and $(1, z^*(\vec{1}))$. These are obtained by $\delta \overset{\text{fix}\vec{}}{=} 0$ and $\delta \overset{\text{fix}\vec{}}{=} 1$, respectively. (We address the case where $\delta \overset{\text{fix}\vec{}}{=} 1$ is infeasible in Sect. 6.6.3.) At a general iteration we have $(b^L, z^L), (b^U, z^U) \in \text{RS}$ such that $b^L < \beta < b^U$, $z^L = f^*(b^L) < f^*(b^U) = z^U$. We set λ equal to the slope of the line segment joining these two points:

$$\lambda = \frac{z^U - z^L}{b^U - b^L}. \tag{6.11}$$

Computing $L^*(\lambda)$ yields the point on the support: $(b = p\delta^*,\ z = z^*(\delta^*)) \in \text{RS}$ so that $b \in [b^L, b^U]$. If $b = \beta$, then we are done and λ is an optimal multiplier, and the chance-constraint instance is solved. If $b = b^L$ or $b = b^U$, then we terminate with the gap region, (b^L, b^U), which contains β. We otherwise shrink the interval of search by replacing (b^L, z^L) or (b^U, z^U) according to whether $b < \beta$ or $b > \beta$,

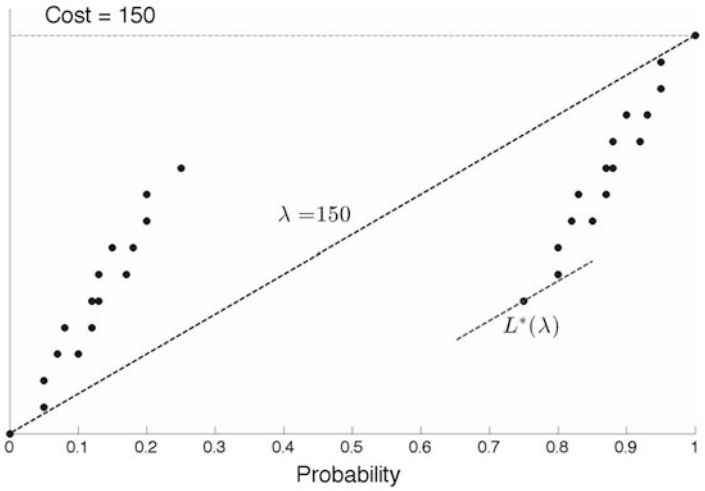


Fig. 6.2 Complete Response Space for Example 6.1 (32 points)

respectively. Because RS is finite, this must converge in a finite number of iterations, and our experiments indicate that it requires very few iterations.

Example 6.1 Suppose $z^*(\delta) = c\delta$ and we have the following five scenarios:

	Scenario				
	1	2	3	4	5
Probability (p)	0.05	0.05	0.07	0.08	0.75
Cost (c)	10	20	30	40	50

Figure 6.2 shows the complete response space, which has 32 points, corresponding to the 2^5 subsets of selections.

The slope of the line segment joining $(0, 0)$ and $(1, 150)$ is the initial Lagrange multiplier, $\lambda = 150$. Minimizing $L(\delta, \lambda) = z^*(\delta) - \lambda p\delta$ moves the line down (parallel) to become the support of epi(f^*, B) and of epi(F^*, B) at $(b, z) = (0.75, 50)$. We would terminate with the exact solution (no gap) if $\beta = 0.75$. Otherwise, the left point is replaced and the interval becomes $[0.75, 1]$ if $\beta > 0.75$; the right point is replaced and the interval becomes $[0, 0.75]$ if $\beta < 0.75$.

6.3.2 Parametric Search Algorithm

We extend tangential approximation to parametric analysis of $\beta \in [\beta^{\min}, \beta^{\max}] \subseteq [0, 1]$. The trade-off between cost and probability is a decision support tool that helps a policy analyst understand impacts, notably the proverbial: *What if* I loosen/tighten the probability? *How* does it affect cost? The analyst may also want to explore: *Why* was this scenario selected and that one not?

The Lagrangian approach uses multiplier values as a trade-off between cost and compliance probability. Varying λ generates $\beta^1, \ldots, \beta^K$, such that $f^*(\beta^k) = F^*(\beta^k) = L^*(\lambda^k) + \lambda^k \beta^k$, thus creating points on the efficient frontier of the bi-objective problem, Pareto-min$\{-b, z\} : (z, b) \in$ RS; see [26, 27] for alternative approaches. MOP Myth 2 in [12] also shows how a Lagrangian duality gap relates to Pareto-frontier generation. The difference with our Lagrangian approach is that it can be done efficiently and can provide additional perspectives of the multiplier values—viz., each break point on the piece-wise linear convex envelope has a range of multiplier values. See, e.g., [3, 34] for early connections between parametric linear programming and multiple objectives.

One approach is to specify a sample of target probabilities. This may be adequate if the Lagrangian problem is solved within a few minutes. Our applications, however, require many minutes (sometimes more than an hour) to solve one Lagrangian problem, so our PySP extension is designed to obtain the convex envelope of the response function for more computer-intensive reference models.

For a specified probability, tangential approximation is efficient among interval reduction methods [10], but it is not dominant. Shen [30] uses bisection, which may obtain an optimal multiplier in just one iteration, once there are two initial solutions with $b^L < \beta < b^U$. It may be (due to the problem instance) that $\lambda^* = \frac{1}{2}(\lambda^L + \lambda^U)$. In a worst case, however, bisection may not generate any new RS point, and it may not confirm the region as the gap region for β. The reason is that if $\beta \in (b^L, b^U)$ is in a gap and $|\lambda_i - \lambda^*|$ is sufficiently small, but $\lambda_i \neq \lambda^*$ for any (finite) i, then $\lambda_i < \lambda^* \rightarrow b_i = b^L$ and $\lambda_i > \lambda^* \rightarrow b_i = b^U$. Only tangential approximation is guaranteed to set $\lambda_i = \lambda^* = (z^U - z^L)/(b^U - b^L)$ once b^L and b^U are generated, thus terminating with the confirmation that β is in the gap region, (b^L, b^U).

Our method is an extension of tangential approximation that computes the minimum number of Lagrangian solutions to obtain the breakpoints in the piece-wise linear envelope. Other methods may compute solutions that provide no new information, for example, by generating a point on the convex envelope already generated by another Lagrange multiplier. This occurs if the probabilities are in the same gap region. None of the target probabilities are known to be on the convex envelope except for $\beta = 0$ and $\beta = 1$, so choosing a sparse set of targets could provide little information to the analyst.

Initialization Set $\lambda = 0$, fix $\delta_s = 0$ for all $s \in S$, and solve the Lagrangian problem (6.3). If the Lagrangian is infeasible, so is CC problem (6.2) for all β. Otherwise, the solution yields the point $(0, z_0) \in$ RS.

Next, fix $\delta_s = 1$ for all $s \in S$ and solve the Lagrangian with $\lambda = 0$, making $L^*(\lambda)$ to be the cost. If the Lagrangian is unbounded, then so is CC problem (6.2) for all β. If it is infeasible, then set λ to some large value and solve to obtain the maximum probability attainable (see Sect. 6.6.3). The solution otherwise yields $(1, z_1) \in \text{RS}$. Initialization ends with two points in RS : $(0, z_0)$ and $(1, z_1)$. Set $\mathcal{I} = \{[0, 1]\}$ and $L_1 = 0$.

Fathoming Gap Intervals At a general iteration we have a sequence of intervals, $\mathcal{I} = \{[b_0, b_1], [b_1, b_2], \ldots, [b_{n-1}, b_n]\}$, with associated min-costs, $\{z_i\}_0^n$, and truth labels, $\{L_i\}_1^n \in \{0, 1\}$. $L_i = 1$ indicates the ith interval is fathomed, meaning that it is the gap region for $\beta \in (b_{i-1}, b_i)$. Otherwise, the associated interval needs to be searched if $L_i = 0$.

Choose an interval that is not fathomed. There are tactical selections such as choosing an interval with the greatest Lagrangian gap value. Such tactics are important if each Lagrangian minimization takes so much time that termination may need to occur before the parametric solution is complete. Set λ as one iteration of tangential approximation:

$$\lambda = \frac{z_i - z_{i-1}}{b_i - b_{i-1}}.$$

Solve the Lagrangian to obtain the response space point (b, z), where $b \in [b_{i-1}, b_i]$. If $b = b_{i-1}$ or $b = b_i$, set $L_i = 1$ and $\lambda_i = \lambda$. Otherwise, do one of the following:

Case 1: $b < \beta^{\min}$ (must have selected the interval $[b_0, b_1]$). Replace $b_0 = b$.
Case 2: $b > \beta^{\max}$ (must have selected the interval $[b_{n-1}, b_n]$). Replace $b_n = b$.
Case 3: $\beta^{\min} \le b_{i-1} < b < b_i \le \beta^{\max}$. Split the interval into $[b_{i-1}, b]$ and $[b, b_i]$. Re-index to maintain $b_0 < b_1 < \cdots < b_n$.

This update maintains $b_0 \le \beta^{\min} \le b_1 < \cdots < b_{n-1} \le \beta^{\max} \le b_n$. We are done when all intervals are fathomed. The scheme terminates in a finite number of iterations since there is a finite number of gap regions, each detected by tangential approximation of its endpoints.

The result is the sequence of successive points in the response space, $\{(b_i, z_i)\}_0^n$, and their associated, optimal multipliers, $\{\lambda_i\}_0^n$:

$$\lambda_0 = 0, \ \lambda_i = \frac{z_i - z_{i-1}}{b_i - b_{i-1}} \text{ for } i = 1, \ldots, n.$$

We provide a function that computes the Lagrangian bound and best feasible solution for each $\beta \in [\beta^{\min}, \beta^{\max}]$ from the algorithm's terminal information. Specifically, find the interval that contains β: $b_{i-1} \le \beta \le b_i$. Then, (b_i, z_i) is the best feasible solution, and the Lagrangian bound is $F^*(\beta) = L^*(\lambda_i) + \lambda_i \beta = z_i + \lambda_i(\beta - b_i)$. The relative Lagrangian gap is thus $g(\beta) = 1 - F^*(\beta)/z_i \in (0, 1]$. Note that $g(\beta) > 0$ because $F^*(\beta) < z_i$ for $\beta < b_i$. We now have the following property.

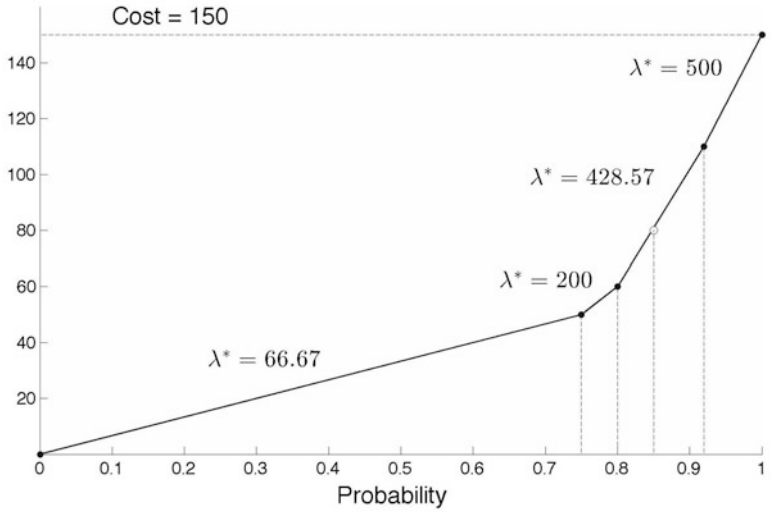


Fig. 6.3 Result of parametric multiplier search for Example 6.1 (c.f., Fig. 6.2)

Property 6.1 *Parametric Tangential Approximation solves the complete parametric CC model with the minimum number of Lagrangian optimizations.*

Our parametric algorithm reduces to the method of tangential approximation for one β. This follows because $\beta^{\min} = \beta^{\max}$ implies we always have either Case 1 or Case 2, thus shrinking the one interval of search and never splitting the interval. At the other extreme, if $\beta^{\min} = 0$ and $\beta^{\max} = 1$, Case 3 always applies and the interval is split. Hence, the number of Lagrangian minimizations is equal to the number of gap regions.

Figure 6.3 shows the result of parametric search for $\beta \in [0, 1]$. The only points generated are the endpoints of each gap interval. The point on F^* at $b = 0.85$ is an alternative optimum to the Lagrangian defined by the slope of the line segment, and this point is not generated. In Sect. 6.5 we describe techniques for generating additional points.

The computational time is dominated by the time it takes to minimize the Lagrangian to obtain z^*. Initialization requires two computations, but all δ values are fixed, so it is the time needed to solve the instance without the chance constraint ($\delta = \vec{0}$) plus the time needed to solve the complete extended form with all scenarios being satisfied ($\delta = \vec{1}$). Each subsequent iteration solves the original instance with the N_S additional binary variables, δ, plus all scenario constraints present with associated δ to indicate whether to require their satisfaction. Each Lagrangian solution yields a point on the envelope, so the total time is the initialization time plus the average time to solve the model instance multiplied by the number of points generated.

We emphasize the novelty of parametric tangential approximation. First, there are no superfluous computations like those of other methods. Each Lagrangian solution either generates a new point on the envelope function or it fathoms a gap region. Our parametric tangential approximation algorithm is optimal in the sense that it requires the minimum number of Lagrangian optimizations to generate the complete convex envelope. Second, there is no a priori specification of target probabilities except for $\beta = 0$ and $\beta = 1$ and all envelope points are generated a posteriori.

6.4 Pre-processing

Connections between chance constraints and knapsack constraints have been exploited by numerous authors (e.g., [19, 23, 28]) and there are knapsack properties that can be used for our application. We found the following property useful in reducing the number of indicator variables when solving the CC problem (6.2).

Property 6.2 *If $p_s > 1 - \beta$, then $\delta_s = 1$ in every feasible solution.*

A proof is straightforward. If $\delta_s = 0$, then the probability is at most $\sum_{i \neq s} p_i$, which equals $1 - p_s$. We thus require $1 - p_s \geq \beta$, which is equivalent to $p_s \leq 1 - \beta$. We let $\alpha \stackrel{\text{def}}{=} 1 - \beta$ for notational convenience in the remainder of this section.

If the scenarios are equally likely, then Property 6.2 yields an all-or-nothing situation. If $\alpha < \frac{1}{N_S}$, then all scenarios are forced to be selected; otherwise, no scenario is forced. In practice, the distribution is generally not uniform and there are scenarios that must be selected for sufficiently large β. For example, if there are only 20 scenarios (maybe during model development), then some $p_s \geq 0.05$—in which case the scenario must be selected for $\beta > 0.95$.

Pre-processing with a specified probability includes fixing $\delta_s = 1$ for all forced selections, i.e., for $p_s > \alpha$. Figure 6.4 shows the reduced response space for Example 6.1 with $\beta = 0.5$. The response space has only 16 of the 32 points, and the left endpoint is (0.75, 60), corresponding to setting $\delta_5 = 1$.

In some cases forced selections solve the problem using the following property.

Property 6.3 *Let $\widehat{S}$ be a set of scenarios for which $\delta_s = 1$ for all $s \in \widehat{S}$. Suppose $P(\widehat{S}) = \sum_{s \in \widehat{S}} p_s \geq \beta$. Then, we can fix $\delta_s = 0$ for all $s \notin \widehat{S}$ without loss in optimality.*

We can use these two properties to limit the intervals over which we must search. Let the scenarios be sorted by non-decreasing probability, and suppose $\widehat{S}$ contains all s for which $p_s > \alpha$. Further suppose that k is the smallest index in the set (so $p_{k-1} \leq \alpha$). Combining Properties 6.2 and 6.3, we find that the chance-constraint instance is solved for $\alpha \in [1 - P(\widehat{S}),\ p_k)$. We use this solution to find probability intervals that solve the chance-constraint instance with forced selections. Let $\mathcal{I}_s = \left[\sum_{i=1}^{s-1} p_i,\ p_s\right)$. We have $\mathcal{I}_1 = [0, p_1) \neq \emptyset$ (assuming $p > 0$). Let $\mathcal{A} = \cup_s \mathcal{I}_s$,

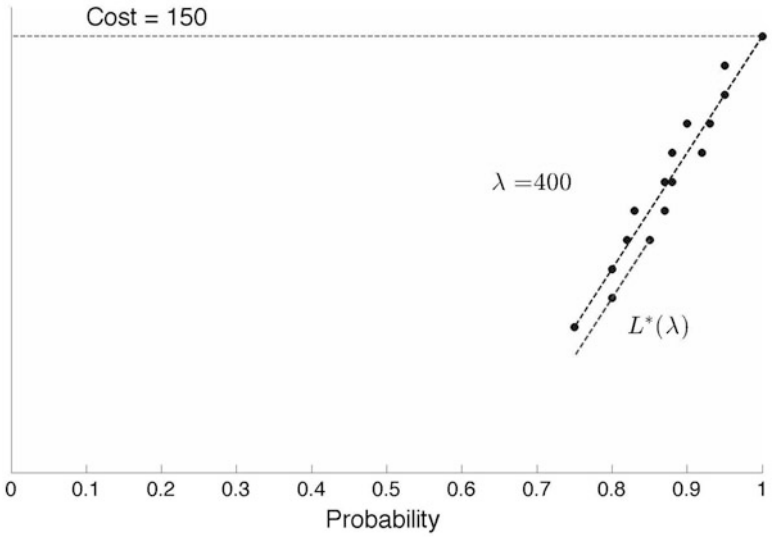


Fig. 6.4 Reduced response space for Example 6.1, fixing the selection of scenario 5

so that the chance-constraint instance is solved by forced selections if, and only if, $\alpha \in \mathcal{A}$.

Example 6.2 This is to demonstrate that solved intervals can be separated by empty ones.

s	$\sum_{i=1}^{s-1} p_i$	p_s	$\mathcal{I}_s$
1	0	0.05	$\neq\emptyset$
2	0.05	0.10	$\neq\emptyset$
3	0.15	0.10	$=\emptyset$
4	0.25	0.20	$=\emptyset$
5	0.45	0.55	$\neq\emptyset$

Thus, $\mathcal{A} = [0, 0.10) \cup [0.45, 0.55)$.

In summary, we find $\mathcal{A}$ for parametric processing, which is a non-empty union of intervals, and we intersect it with

$$[\alpha^{\min} = 1 - \beta^{\max},\ \alpha^{\max} = 1 - \beta^{\min}].$$

This process is used in the parametric version of the Lagrange multiplier search by fathoming intervals contained in the forced-selection interval. We can force at the outset selections for scenarios such that $p_s > \alpha^{\max}$. Although the conditions are simple to establish, they can have a significant impact.

We emphasize that our algorithmic goal is to provide an advanced understanding of the chance-constraint model. An analyst needs to know *why* some scenarios are selected while others are not—is it due to economic benefit or are they restricted by other constraints? Can the analyst deduce some scenario dependence—e.g., $\delta_s = 1 \rightarrow \delta_t = 0$. Such analysis could occur during a debugging stage or during data development, but in a mature model our analysis could add clarity concerning what the scenario constraints mean and how they relate to the rest of the model.

6.5 Gap Closing

The procedures of the previous section provide the lower convex envelope for RS, denoted F^*; however, analysts may benefit from seeing more points in the space even if they are not on this frontier. We seek additional information about solutions in gap regions by fixing δ, thus providing points above the envelope function. It is natural for a good analyst to ask, "How close are suboptimal solutions?" (which may have other favorable properties to present options for management).

Consider a gap region $[b^L, b^U]$ with $\beta \in (b^L, b^U)$ and $g(\beta) > \tau^{\text{gap}}$ (a tolerance). We present some heuristics to search for a feasible solution, (b, z), where b is in the interior of the gap region—i.e., $b \in [\beta, b^U)$. Let δ^L and δ^U be optimal selection values associated with the endpoints, and define the partition of scenarios:

$$\begin{aligned} S^{00} &= \{s: \delta_s^L = 0,\ \delta_s^U = 0\} \\ S^{01} &= \{s: \delta_s^L = 0,\ \delta_s^U = 1\} \\ S^{10} &= \{s: \delta_s^L = 1,\ \delta_s^U = 0\} \\ S^{11} &= \{s: \delta_s^L = 1,\ \delta_s^U = 1\}. \end{aligned}$$

We must have $\left|S^{01} \cup S^{10}\right| > 0$ because the two solutions differ. Our first heuristic is called *restricted flipping* and it fixes values in $S^{00} \cup S^{11}$ and flip values in $S^{01} \cup S^{10}$, moving from b^L to β and/or moving from b^U to β.

If $\left|S^{01}\right| = 1$, restricted flipping takes us from b^L to b^U, so suppose $\left|S^{01}\right| > 1$. We then select a sequence to flip until the total probability, b, is at least β. If $b = b^U$, then this flipping sequence fails, and we order the sequence by probability, leaving the minimum value for last. If that last flip is necessary to reach β—i.e., if $\sum_{s \in S} p_s < \beta$ for all $S \underset{\neq}{\subset} S^{01}$, restricted flipping fails. We otherwise fix $\delta_s = 1$ for those flipped. Those not flipped are fixed at 0, their current value. This gives us a new point in the response space, $(b, z^*(\delta))$.

Initialize $z^{\text{Best}} = z^U$ and $b^{\text{Best}} = b^U$. If $z^*(\delta) < z^{\text{Best}}$, then update $z^{\text{Best}} = z^*(\delta)$ and $b^{\text{Best}} = p\delta$. Test for termination using a gap tolerance, $g(\beta) \leq \tau^{\text{gap}}$, and a probability tolerance: $\left|b^{\text{Best}} - \beta\right| \leq \tau^{\text{prob}}$. If we do not terminate, flip from b^U, fixing $\delta_s = 0$ for a sequence of $s \in S^{01}$, ordered by probability, until $p\delta < \beta$. Let b be the probability just before reaching this condition. As above, we must have

Table 6.1 Points in the response space and their associated scenario selections

	Subset selected	Probability	Cost
	S	$\sum_{s\in S} p_s$	$\sum_{s\in S} c_s$
1	{5}	0.750	50
2	{1, 5}	0.800	60
3	{2, 5}	0.800	70
4	{3, 5}	0.820	80
5	{4, 5}	0.830	90
6	{1, 2, 5}	0.850	80
7	{1, 3, 5}	0.870	90
8	{2, 3, 5}	0.870	100
9	{1, 4, 5}	0.880	100
10	{2, 4, 5}	0.880	110
11	{3, 4, 5}	0.900	120
12	{1, 2, 3, 5}	0.920	110
13	{1, 2, 4, 5}	0.930	120
14	{1, 3, 4, 5}	0.950	130
15	{2, 3, 4, 5}	0.950	140
16	{1, 2, 3, 4, 5}	1.000	150

Table 6.2 Illustration of calculations for points in the interval (0.85, 0.92)

b	z^*	S	
0.85	80	{1, 2, 5}	$S^{10} = \emptyset,\ S^{11} = \{1, 2, 5\}$
0.87	90	{1, 3, 5}	$L^*(\lambda^*) + \lambda^* b = 90$
0.88	100	{1, 4, 5}	$L^*(\lambda^*) + \lambda^* b = 95$
0.90	120	N/A[a]	$L^*(\lambda^*) + \lambda^* b = 105$
0.92	110	{1, 2, 4, 5}	$S^{01} = \{4\},\ S^{00} = \{3\}$

[a] $f^*(0.90) = f^*(0.92)$

$b > b^L$ to obtain a new point in the response space, and if that is the case, then compute $z^*(\delta)$ and apply the same tests to update z^{Best} and terminate.

Table 6.1 enumerates the 16 points of the reduced response space from Fig. 6.4, plotted in Fig. 6.5 (spread out to see the points more distinctly). The envelope function, F^*, is the piece-wise linear function, with $B = [0.75, 1]$. We can restrict $\beta \in B$ because the parametric range is $\alpha \in [0.05, 0.25)$.

Suppose we want to close the gap in the interval $(0.85, 0.92)$, with $\lambda^* = (110 - 80)/(0.92 - 0.85) = 428.57$. The two circled points are the only non-dominated, feasible points with a better solution than $z^U = 110$ as documented in Table 6.2.

Restricted flipping fails because once we fix the common selections, $\delta_1 = \delta_2 = \delta_5 = 1$, only $\delta_4 = 1$ flips from b^L, which gets us to b^U; and, flipping $\delta_4 = 0$ from b^U gets us to b^L. However, if we relax fixing all common selections, we can reach $(0.88, 100)$ from b^U by flipping δ_2, resulting in $\delta = (1, 0, 0, 1, 1)$. This is the optimal value, but all we can confirm is that the best feasible solution, with $z = 100$, has relative gap value $g(0.88) = 1 - 95/100 = 0.05$. This is a significant

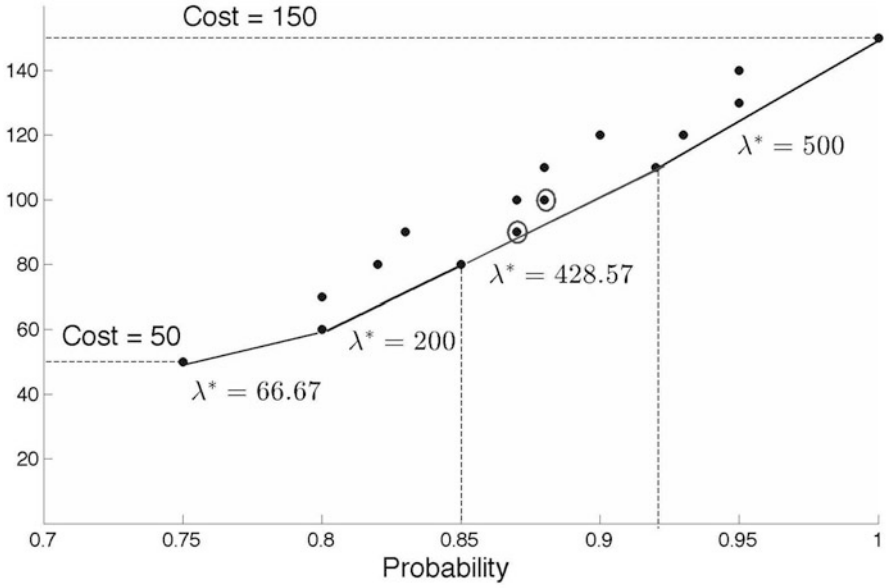


Fig. 6.5 Reduced response space of Example 6.1 (c.f., Fig. 6.4) after our parametric search algorithm finds $F^*(\beta)$ for $\beta \geq 0.75$

improvement over the original value $g(0.88) = 1 - 95/110 = 0.1364$, and it is the best we can do.

We also cannot reach $(0.87, 90)$ by restricted flipping because $S = \{1, 3, 5\} \not\subset S^U = S^{01} \cup S^{11}$ implies that we cannot flip from (b^U, z^U). Similarly, $S \not\subset S^L = S^{10} \cup S^{11}$ means we cannot flip from (b^L, z^L). However, if we relax fixing common exclusions, we can then flip $\delta_3 = 1$ and consider flipping others in S^L. Heuristics that relax fixing common exclusions remain as future research.

It is in general inexpensive and potentially valuable to consider flipping only scenarios that are selected by one endpoint and not the other. Contrary to the particular example, common selections may be a form of evidence, and there is little computational cost to try it first. That is, we need not solve any new minimization problem to discover if this flipping generates a new probability; we simply loop through a sorted list of probabilities. If this fails, then relaxed flipping is tried, which may generate a new feasible response space point, (b, z) with $b \geq \beta$ and $z < z^U$. If this is the case, then we decrease the gap by setting $z^{\text{Best}} = z$.

Suppose restricted flipping fails to yield an acceptable solution—i.e., the best solution is not within tolerances: $\left|b^{\text{Best}} - \beta\right| > \tau^{\text{prob}}$ or $g(\beta) > \tau^{\text{gap}}$. We then begin to enlarge the space of candidates to flip. For parametric chance constraint we use gap-closing heuristics to generate additional points in RS. The purpose is to learn about the cost-probability trade-offs.

We applied our methods to the three models as described in the Supplementary Material attached to the electronic version of this chapter. Two of the models capture

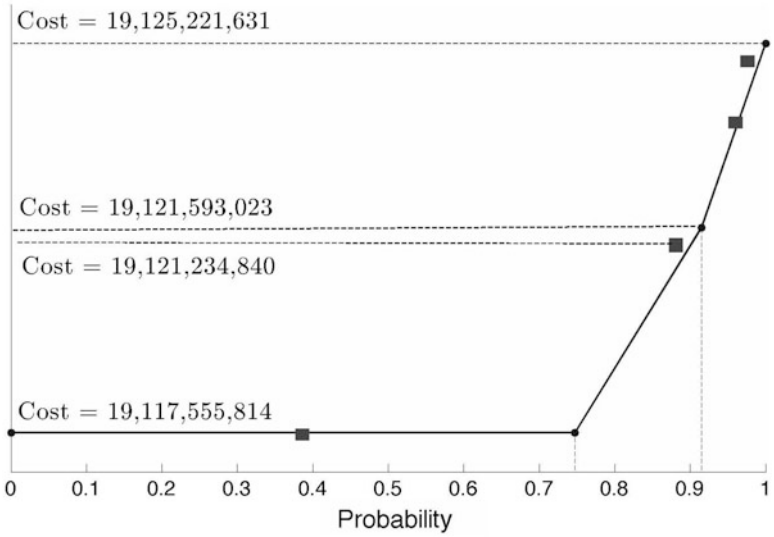


Fig. 6.6 Adding response space points to the 10-scenario Midwest GEP model

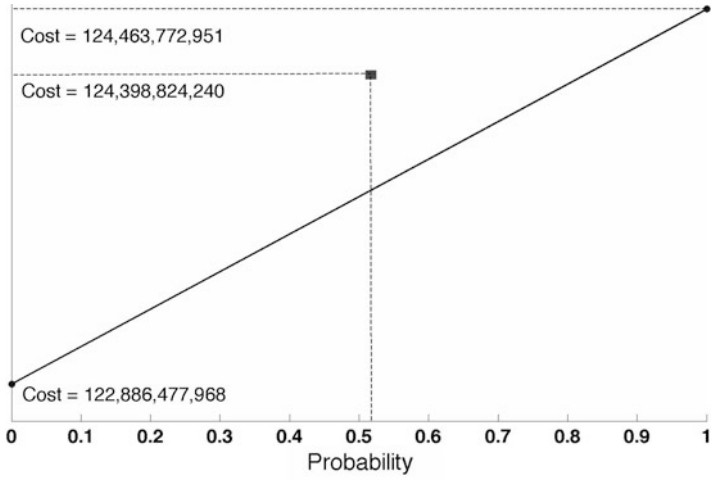


Fig. 6.7 Adding to the response space for the 70-scenario Korean GEP model

features of an electricity generation expansion planning (GEP) problem, and the third model is a network flow, capacity-planning model. We refer to the models as Midwest GEP, Korean GEP, and Network Flow, respectively. Figures 6.6, 6.7, and 6.8 show response space points added to each of the instances by flipping selections of each upper endpoint ($\delta_s^U = 1$) not selected in the interval's lower endpoint ($\delta_s^L = 0$).

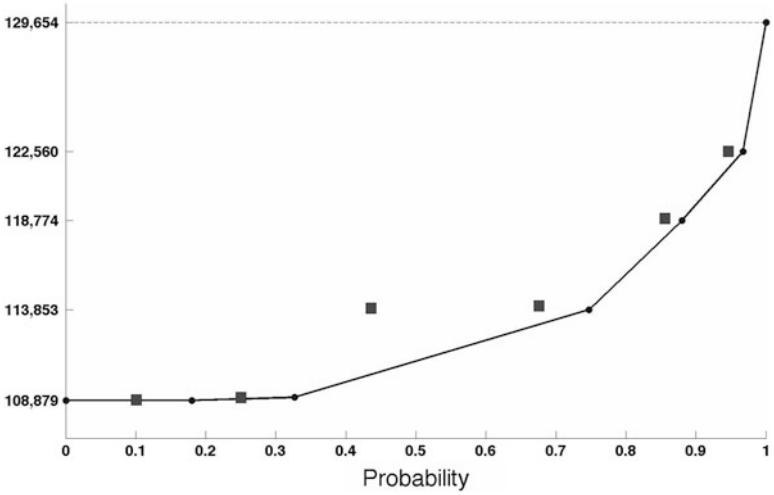


Fig. 6.8 Adding response space points to the 10-scenario network flow model

The one point added to RS in the Korean model (Fig. 6.7) tells us $f^*(0.512) \leq 124,398,824,240$, which is a slight improvement over the upper endpoint, $z^U = 124,463,772,951$. The relative gap is reduced by an order of magnitude to 0.000111 (from 0.006237).

Here is the algorithm to generate additional points after F^* is constructed. First, form the list of gap intervals, $\{(i, w_i, m_i)\}$, where $w_i = b_i - b_{i-1}$ is the width, and $m_i = \frac{1}{2}(b_i + b_{i-1})$ is the midpoint. Sort this list by width and drop intervals with $w_i \leq 2\tau^{\text{prob}}$.

If the number of (sufficiently wide) intervals is greater than a specified maximum, we simply drop the last few intervals. If we have fewer than the specified number of intervals, then we use the sort-order to split the first (i.e., widest) interval:

$$(i, w, m) \rightarrow \left(i, \frac{1}{2}w, m - \frac{1}{4}w\right), \left(i, \frac{1}{2}w, m + \frac{1}{4}w\right).$$

Note that the original index is retained when splitting. We then re-sort until we either reach the maximum number of points specified or the split would make the width too small—i.e., stop once $w \leq 4\tau^{\text{prob}}$.

We have in the end abscissa points, $\{m_k\}$, plus associated widths and gap-region indexes, for $k = 1, \ldots, K$, where K is within the specified maximum and $w_k > 2\tau^{\text{prob}}$. For each k, initialize selections from z_{i_k}, the upper endpoint of the i_k-th gap region, and flip $s_1, s_2, \ldots$ (in probability-order) until reaching $b = \sum_{j=1}^{v} p_{s_j} \geq m_k$ and $b - p_{s_v} < m_k$. If this is reached before $b^L = b_{i_k-1}$, we then compute $z^*(\delta)$ to obtain the new RS point, $(b, z^*(\delta))$. Otherwise, we simply go to the next interval.

We can combine the gap intervals with pre-processing intervals of the form $\cup_k(\bar{p}_{s_k}, \bar{a}_k]$, where $\bar{v} \stackrel{\text{def}}{=} 1 - v$ for any $v \in [0, 1]$. We know $b = \frac{1}{2}(\bar{a}_k + \bar{p}_{s_k})$ is solved by a forced selection (that fixes δ). If the forced selection has $b = \bar{a}_k$ and the selection is already an endpoint of a gap interval, then the solver regenerates b_i and we do not obtain a new point. However, if $\min_i |\bar{a}_k - b_i| > \tau^{\text{prob}}$, we then compute $(b = p\delta,\ z^*(\delta))$ for δ corresponding to the forced selections by solving for $z^*(\delta)$.

A major advantage of doing this is that $f^*(b) = z^*(\delta)$. This optimality cannot be guaranteed with a gap-closing heuristic, like restricted flipping. On the other hand, an advantage of using gap intervals to determine the abscissa values is that we have a more distributed collection of response space points, which gives a sense of how the chance constraint affects the solution. Further experimentation with this avenue of solution insights from a response space is warranted.

6.6 Some Pitfalls to Consider

Our implementation has identified pitfalls that merit some attention. For convenience, assume $z > 0$ for all $(b, z) \in \text{RS}$, so relative cost values can be used without absolute values.

6.6.1 Tolerance Relations

We can increase the optimality tolerance, τ^{opt}, to reduce the time to minimize the Lagrangian. The effect of this change depends on the solver [11] and relates to two tolerances that can be set as options in our Python program:

- τ^{prob}: two probabilities, b and b', are equal if $\left|b - b'\right| \leq \tau^{\text{prob}}$.
- τ^{gap}: (b, z) is acceptable (i.e., z is sufficiently close to $f^*(b)$) if the gap between z and the Lagrangian bound, $F^*(b) = L^*(\lambda) + \lambda b$, satisfies $1 - F^*(b)/z \leq \tau^{\text{gap}}$. Recall we use this when exploring gap regions with $z = z^U$.

We cannot be sure exactly what near-optimality means, but we can suppose $\mathcal{L}$ is a lower bound on the (unknown) optimum because $\mathcal{L}(\lambda) \leq L^*(\lambda) \leq z - \lambda b$. The solver terminates if

$$\frac{z - \lambda b - \mathcal{L}(\lambda)}{\mathcal{L}(\lambda)} \leq \tau^{\text{opt}}.$$

Equivalently (as implemented), $z - \lambda b \leq L^*(\lambda)(1 + \tau^{\text{opt}})$.

Figure 6.9a shows an alternative optimum for the Lagrangian with $\lambda = z_1 - z_0$, which is the slope of line segment joining the two initial points, $(0, z_0)$ and $(1, z_1)$. A solver should begin by checking the optimality of the endpoint that is still resident, but some will begin anew. That is the only way the alternative solution

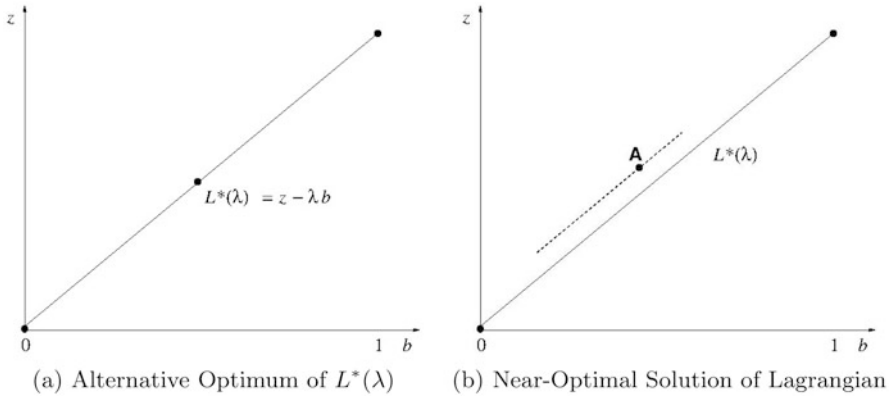


Fig. 6.9 Inexact alternative Lagrange optimum in $(b^L + \tau^{\text{prob}}, b^U - \tau^{\text{prob}})$. (**a**) Alternative optimum of $L^*(\lambda)$. (**b**) Near-optimal solution of Lagrangian

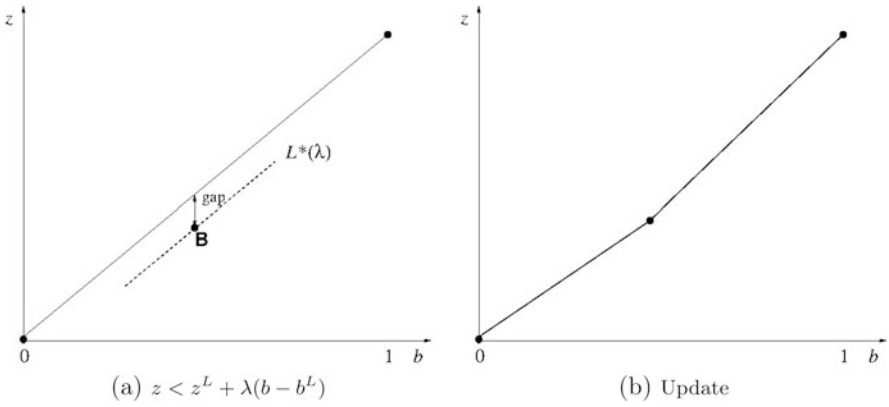


Fig. 6.10 Accepting an inexact alternative Lagrange optimum as a new RS point. (**a**) $z < z^L + \lambda(b - b^L)$. (**b**) Update

would be reached. If this occurs, then we save the generated RS point because b is not within tolerance of either endpoint—i.e., $b \in (b^L + \tau^{\text{prob}},\ b^U - \tau^{\text{prob}})$. However, the interval is fathomed because there are no points below the line.

Figure 6.9b shows a situation where a new point is generated by being within (relative) tolerance of optimality: $z - \lambda b \leq L^*(\lambda)(1 + \tau^{\text{opt}})$. With cost above the line (i.e., $z > z^L + \lambda(b - b^L)$), depicted as point A, we save (b, z), but we fathom the interval, as in the case of the exact optimum.

Figure 6.10a shows the near-optimum of the Lagrangian below the line, labeled point B. We consider B to be a new point if $\left|b - b^L\right| > \tau^{\text{prob}}$, $\left|b - b^U\right| > \tau^{\text{prob}}$, and the Lagrangian gap exceeds tolerance: $1 - (z^L + \lambda(b - b^L))/2 > \tau^{\text{gap}}$. We otherwise treat B the same as if (b, z) is above the line.

The value τ^{opt} pertains to the Lagrangian optimum, whereas τ^{gap} and τ^{prob} pertain to the cost and probability, respectively. The action to treat (b, z) as a new point (splitting the interval, rather than fathoming it) depends on how these tolerances relate. Their meanings are different and can cause anomalous behavior, even if $\tau^{\text{opt}} = \tau^{\text{gap}}$. See [11] for elaboration and examples of how tolerances can interact. A pitfall to avoid is setting τ^{opt} in a way that is inconsistent with other tolerances. More analysis of the interaction among tolerances, including some solver tolerances, is an avenue for further research.

6.6.2 *Measuring Probability*

Using the indicator variable in (6.1), we find that it is possible to have $\delta_s = 0$ but still have scenario-s constraints satisfied. Therefore, $p\delta$ is not an exact measure of the probability of being feasible. Letting Q_s denote the feasible values of (x, y_s), the real chance constraint is $\sum_s \Pr\big((x, y_s) \in Q_s \,|\, s\big)\, p_s \geq \beta$. Our probability value is thus an underestimate:

$$p\delta = \sum_{s=1}^{N_S} p_s \delta_s \leq \sum_{s=1}^{N_S} \Pr\big((x, y_s) \in Q_s \,|\, s)\big)\, p_s.$$

To see this, consider x restricted to satisfy scenario s. That is the case if $\delta_s = 1$, and we have in this case that $\Pr\big((x, y_s) \in Q_s \,|\, s)\big) = 1$. However, x can satisfy the constraint when $\delta_s = 0$, so in general $\delta_s \leq \Pr\big((x, y_s) \in Q_s \,|\, s)\big)$. Being an underestimate means that the chance-constraint model is conservative because $p\delta \geq \beta$ implies that the true chance constraint is satisfied.

A modeler can add violation variables to measure actual violation and enforce the converse: $\delta_s = 0$ implies scenario s is violated. Let the auxiliary variable v_i^s measure violation of the ith constraint in scenario s:

$$\textstyle\sum_j a_{ij}^s x_j^s - b_i^s \geq -v_i^s, \qquad\qquad 0 \leq v_i^s \leq (1 - \delta_s)M.$$

The scenario constraint is satisfied if $v^s = 0$, which is forced by $\delta_s = 1$ (as in first model).

For $\delta_s = 0$, the solver could produce a solution with $v^s \neq 0$ even if there is no violation as long as the cost is not greater than the minimum. In fact, if an interior solution is computed, then both the surplus variable and v_i^s are positive if the ith constraint is over satisfied in *some* optimal solution. To ensure $v \neq 0$ except when necessary, define a nuisance cost, $\varepsilon > 0$, and add $\varepsilon \sum_{i,s} v_i^s$ to the objective. If there are alternative optima, then favor is given to $v = 0$. Notice that ε must be small enough to preserve minimality of the original cost. Then, an optimal solution will have $v_i^s = \max\left\{0,\ b_i^s - \sum_j a_{ij}^s x_j^s\right\}$, which equals the amount of violation of the ith constraint.

For some models, like our Network Flow, setting $\delta_s = 0$ has no effect on the objective function, but in other models, this could mislead an analyst who uses scenario violation to support one decision over another. Moreover, the exact form is important to answer questions like, "What is the impact of having a chance constraint?" The level of violation may be of interest, which is not obtained in the first (underestimate) model. Further analysis of the level of constraint violation can be supported by taking a large number of additional samples for the purpose of a better estimate of the actual probability of violation (see, e.g., [24]).

6.6.3 When It Is Infeasible to Select All Scenarios

We have assumed for notational convenience that it is feasible to select all scenarios—i.e., $\delta_s = 1$ for all $s \in S$. This may not be the case, and we might need to find

$$P^{\max} = \max_{\delta \in \{0,\,1\}^{N_S}} \{p\delta : \exists x \in X \ni Y_s(x, \delta) \neq \emptyset\}. \tag{6.12}$$

If we seek a solution for $\beta > P^{\max}$, then the specified chance-constraint instance is infeasible. We otherwise need to find λ such that $L^*(\lambda)$ yields the RS point, $(P^{\max}, z^*(\delta))$ for some optimal δ (not necessarily the selections computed if we only maximized $p\delta$ without regard for cost). Tangential approximation is initialized with this point to bracket the search in this case.

Here is how we find such a λ. Let Z be the cost for the computed solution of $P^{\max}$, and consider $\lambda > Z/\tau \geq z^*(\delta)/\tau$, where τ is sufficiently small to ensure that $L^*(\lambda) = z^*(\delta) - \lambda p\delta \Rightarrow p\delta \geq P^{\max} - \tau^{\text{prob}}$. To help intuition, consider $z^*(\delta) = c\delta$. Then, $c_s - \lambda p_s < 0$ for all s for $\lambda > \max_s \left\{c_s/p_s\right\}$. This means $\delta_s = 1$ unless it is not feasible to select scenario s. Minimization of the Lagrangian takes care of the trade-off, making $p\delta$ a maximum over all feasible selections.

6.6.4 Low Probabilities

The general range for the parametric tangential approximation algorithm is $[\beta^{\min}, \beta^{\max}]$. If $\beta^{\min} = 0$ is infeasible, then our code terminates, as this means the original model instance is infeasible without the scenario constraints. One usually imagines low values of β as being of little interest, but we assert that this is a pitfall because low values of β can also provide some information, starting with $\beta = 0$:

- What is my minimum cost with no scenario compliance?

- How much computational time is due to adding the joint chance constraint? (That is, what is a baseline for how much computational time to expect as we search for optimal multiplier values, which adds N_S binary variables to the model?)

The full range is useful for debugging a model and testing its validity even before analysis support.

6.7 Summary and Conclusions

Each Lagrangian solution generates a point that is an exact optimum for the Lagrangian problem (6.3) and its associated parametric program. The piece-wise linear function connecting those points is the envelope function that yields the Lagrangian bound. We have shown that the complete parametric tangential approximation minimizes the number of Lagrangian solutions to generate the convex envelope. Each iteration of parametric tangential approximation yields a new RS point that either confirms a gap region (immediately, as the resident solution is optimal) or causes it to shrink or split. The terminal succession of RS points, $\{(b_i, z_i)\}_{i=0}^{N}$, covers $[0, P^{\max}]$ with $b_0 = 0 < b_1 < \cdots < b_N = P^{\max}$.

Tangential approximation for a single β was introduced decades ago, and it is not necessarily an optimal algorithm. For example, bisection could reach the one optimal multiplier faster. However, our extension to a complete parametric search is optimal in that it minimizes the number of Lagrangian solutions needed for complete parametrization.

Each interval (b_{i-1}, b_i) contains Lagrangian duality gaps, where there may be a solution above (or on) the convex envelope function for which $z < z_U$. Gap intervals can be explored by a variety of heuristics. We presented one approach, called restricted flipping, that seeks a better feasible solution for $\beta \in (b_{i-1}, b_i)$ than z_i by flipping optimal values of δ that differ between the endpoint solutions. Once δ is specified, we compute the RS point, $(p\delta,\ z^*(\delta))$. We presented a heuristic for choosing probability values, based on the midpoints of intervals that have been sorted by $b_i - b_{i-1}$.

Each Lagrangian solution to a general problem, $L(x) = f(x) - \lambda g(x)$, solves two programs:

$$\min\ f(x)\colon g(x) \geq b \stackrel{\text{def}}{=} g(x^*) \text{ and } \max\ g(x)\colon f(x) \leq c \stackrel{\text{def}}{=} f(x^*)$$

for any $x^* \in \operatorname{argmin} L(x)$. Varying $\lambda \in [0, \infty)$ yields parametric solutions $\{b,\ f^*(b)\}_b$ and $\{g^*(c),\ c\}_c$. They are precisely the same set in RS. These are also equivalent to using a weighted sum of the bi-criteria program:

$$\min\ \alpha f(x) + (1 - \alpha)(-g(x)).$$

Varying $\alpha \in (0, 1)$ generates Pareto-optimal points. Each α is equivalent to the Lagrangian with $\lambda = 1/(1-\alpha)$ (so $\min\ L = f - \lambda g \leftrightarrow \min\ \alpha f + (1-\alpha)g$). This generates the same portion of the efficient frontier. The weighted sum fails to generate some Pareto-optimal points—viz., non-convex segments of the frontier. This is precisely the Lagrangian duality gap.

One reason to point this out is that there has been a vast literature on Lagrangian duality and bi-criteria programming (separately and jointly) in the last several decades. Most of it assumes a special structure, notably convexity or separability, which we do not. Moreover, we go beyond the algorithmics, focusing on the use of the convex envelope to support analysis. Our gap-resolution method demonstrates an effective computational approach not only to reduce the gap, but also to provide a better understanding of the cost-probability trade-off, including sub-optimal solutions.

Most importantly, Everett's Generalized Lagrange Multiplier Method advances analysis with efficient computation implemented as an open-source extension of PySP. Output includes tables of RS points, which can be used by software to provide graphical support. The goal is insight into the trade-off between cost and scenario-satisfaction probability. It is for that reason that we provide additional code to explore RS, not only for near-optimal solutions, but also for gaining information about alternative (sub-optimal) solutions. A motive for such exploration is to consider alternative solutions that have properties not represented in the model—e.g., ease of policy implementation.

Our current and future research extends our work to multiple chance constraints. Our foundation is Everett's paper and its derivatives, notably [2] and [13, 14].

Acknowledgments The research in this article was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under contract number KJ0401000 through the Project "Multifaceted Mathematics for Complex Energy Systems".

References

1. S. Ahmed, A. Shapiro, Chapter 12: solving chance-constrained stochastic programs via sampling and integer programming, in *TutORials in Operations Research*, ed. by Z.-L. Chen, S. Raghavan (INFORMS, Catonsville, 2008), pp. 261–269
2. R. Brooks, A. Geoffrion, Finding Everett's Lagrange multipliers by linear programming. Oper. Res. **14**(6), 1149–1153 (1966)
3. A. Charnes, W.W. Cooper, Systems evaluation and repricing theorems. Manag. Sci. **9**(1), 33–49 (1962)
4. J.P. Evans, F.J. Gould, S.M. Howe, A note on extended GLM. Oper. Res. **19**(4), 1079–1080 (1971)
5. H. Everett, III, Generalized Lagrange multiplier method for solving problems of optimum allocation of resources. Oper. Res. **11**(3), 399–417 (1963)
6. A.M. Geoffrion, The purpose of mathematical programming is insight, not numbers. Interfaces **7**(1), 81–92 (1976)
7. F.J. Gould, Extensions of Lagrange multipliers in nonlinear programming. SIAM J. Appl. Math. **17**(6), 1280–1297 (1969)
8. H.J. Greenberg, Lagrangian duality gaps: Their source and resolution. Technical Report CP-69005, Southern Methodist University, Dallas (1969). http://math.ucdenver.edu/~hgreenbe/pubs.shtml
9. H.J. Greenberg, Bounding nonconvex programs by conjugates. Oper. Res. **21**(1), 346–348 (1973)
10. H.J. Greenberg, The one dimensional generalized Lagrange multiplier problem. Oper. Res. **25**(2), 338–345 (1977)
11. H.J. Greenberg, Supplement: tolerances, in [Holder, A. (ed.), *Mathematical Programming Glossary*. INFORMS Comput. Soc. (2014)]. Posted 2003. Also appears at *Optimization Online*. http://www.optimization-online.org/DB_HTML/2012/05/3486.html
12. H.J. Greenberg, Supplement: myths and counterexamples in mathematical programming, in [A. Holder (ed.), *Mathematical Programming Glossary*. INFORMS Comput. Soc. (2014)]. Posted 2010
13. H.J. Greenberg, Supplement: Lagrangian saddle point equivalence, in [A. Holder (ed.), *Mathematical Programming Glossary*. INFORMS Comput. Soc. (2014)]. Transcribed from 1969 Course Notes
14. H.J. Greenberg, Supplement: response space, in [A. Holder (ed.), *Mathematical Programming Glossary*. INFORMS Comput. Soc. (2014)]. Transcribed from 1969 Course Notes
15. H.J. Greenberg, T. Robbins, Finding Everett's Lagrange multipliers by generalized linear programming. Technical Report CP-70008, Southern Methodist University, Dallas, 1970. http://math.ucdenver.edu/~hgreenbe/pubs.shtml
16. W.E. Hart, J.P. Watson, D.L. Woodruff, Python optimization modeling objects (Pyomo). Math. Program. Comput. **3**(3), 219–260 (2011)
17. W.E. Hart, C. Laird, J.-P. Watson, D.L. Woodruff, *Pyomo—Optimization Modeling in Python* (Springer, Berlin, 2012)
18. A. Holder (ed.), *Mathematical Programming Glossary*. INFORMS Comput. Soc. (2014). http://glossary.computing.society.informs.org
19. S. Küçukyavuz, On mixing sets arising in chance-constrained programming. Math. Program. **132**(1–2), 31–56 (2012). ISSN 0025-5610. https://doi.org/10.1007/s10107-010-0385-3
20. M.A. Lejeune, S. Shen, Multi-objective probabilistically constrained programming with variable risk: new models and applications. Eur. J. Oper. Res. **252**(2), 522–539 (2016)
21. J. Luedtke, An integer programming and decomposition approach to general chance-constrained mathematical programs, in *Integer Programming and Combinatorial Optimization*, ed. by F. Eisenbrand, F. Shepherd. Lecture Notes in Computer Science, vol. 6080 (Springer, Berlin, 2010), pp. 271–284. ISBN: 978-3-642-13035-9

22. J. Luedtke, A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support. Math. Program. A **146**, 219–244 (2014)
23. J. Luedtke, S. Ahmed, G.L. Nemhauser, An integer programming approach for linear programs with probabilistic constraints. Math. Program. **122**(2), 247–272 (2010). ISSN: 0025-5610. https://doi.org/10.1007/s10107-008-0247-4
24. A. Nemirovski, A. Shapiro, Convex approximations of chance constrained programs. SIAM J. Optim. **17**(4), 969–996 (2006)
25. A. Prekopa, Probabilistic programming, in *Handbooks in Operations Research and Management Science, Volume 10: Stochastic Programming*, ed. by A. Ruszczyński, A. Shapiro (Elsevier, Amsterdam, 2003)
26. T. Rengarajan, D. P. Morton, Estimating the efficient frontier of a probabilistic bicriteria model, in *Proceedings of the 2009 Winter Simulation Conference*, ed. by M.D. Rossetti, R.R. Hill, B. Johansson, A. Dunkin, R.G. Ingalls (2009), pp. 494–504
27. T. Rengarajan, N. Dimitrov, D.P. Morton, Convex approximations of a probabilistic bicriteria model with disruptions. INFORMS J. Comput. **25**(1), 147–160 (2013)
28. A. Ruszczyński, Probabilistic programming with discrete distributions and precedence constrained knapsack polyhedra. Math. Program. **93**(2), 195–215 (2002)
29. J.F. Shapiro, Generalized Lagrange multipliers in integer programming. Oper. Res. **19**(1), 68–76 (1971)
30. S. Shen, Using integer programming for balancing return and risk in problems with individual chance constraints. Comput. Oper. Res. **49**, 59–70 (2014)
31. J.-P. Watson, R.J.-B. Wets, D.L. Woodruff, Scalable heuristics for a class of chance-constrained stochastic programs. INFORMS J. Comput. **22**(4), 543–554 (2010). ISSN: 1526-5528. https://doi.org/10.1287/ijoc.1090.0372
32. J.-P. Watson, D.L. Woodruff, W.E. Hart, Modeling and solving stochastic programs in Python. Math. Program. Comput. **4**(2), 109–149 (2012)
33. W.B. Widhelm, Geometric interpretation of generalized Lagrangian multiplier search procedures in the payoff space. Oper. Res. **28**(3), 822–827 (1980)
34. P.L. Yu, M. Zeleny, Linear multiparametric programming by multicriteria simplex method. Manag. Sci. **23**(2), 159–170 (1976)

Chapter 7
An Analysis of Multiple Contaminant Warning System Design Objectives for Sensor Placement Optimization in Water Distribution Networks

Jean-Paul Watson, William E. Hart, Harvey J. Greenberg, and Cynthia A. Phillips

Abstract A key strategy for protecting municipal water supplies is the use of sensors to detect the presence of contaminants in associated water distribution systems. Deploying a contamination warning system involves the placement of a limited number of sensors—placed in order to maximize the level of protection afforded. Researchers have proposed several models and algorithms for generating such placements, each optimizing with respect to a different design objective. The use of disparate design objectives raises several questions: (1) What is the relationship between optimal sensor placements for different design objectives? and (2) Is there any risk in focusing on specific design objectives? We model the sensor placement problem via a mixed-integer programming formulation of the well-known p-median problem from facility location theory to answer these questions. Our model can express a broad range of design objectives. Using three large test networks, we show that optimal solutions with respect to one design objective are often highly sub-optimal with respect to other design objectives. However, it is sometimes possible to construct solutions that are simultaneously near-optimal with respect to a range of design objectives. The design of contamination warning

The author "Harvey J. Greenberg" is deceased at the time of publication.

J.-P. Watson
Center for Applied and Scientific Computing and Global Security Directorate, Lawrence Livermore National Laboratory, Livermore, CA, USA
e-mail: jeanpaulwatson@llnl.gov

W. E. Hart (✉) · C. A. Phillips
Computing Research Center, Sandia National Laboratories, Albuquerque, NM, USA
e-mail: wehart@sandia.gov; caphill@sandia.gov

H. J. Greenberg
Mathematics Department, University of Colorado, Denver, CO, USA

A. Holder (ed.), *Harvey J. Greenberg*, International Series in Operations Research & Management Science 295, https://doi.org/10.1007/978-3-030-56429-2_7

systems thus requires careful and simultaneous consideration of multiple, disparate design objectives.

7.1 Introduction

A series of municipal water contamination incidents and increased awareness of water supply vulnerabilities led the newly formed United States Department of Homeland Security to invest in research to better protect municipal water systems after the 9/11 terrorist attacks. See [22] for more information on some of the history of such incidents and threats. There has been a contemporaneous increase in water-security research around the world over the last two decades. See the recent survey [14], which also describes two other contamination incidents outside the USA.

There has been considerable research to mitigate this risk in the design and deployment of contaminant warning systems (CWSs) for water distribution networks based on real-time sensors that provide continual water quality monitoring. In the short-to-moderate term, complete protection of distribution networks is unrealistic due to budgetary constraints. Optimization is consequently required to maximize the level of protection afforded by a limited number of sensors.

The efficacy of sensor placement can be quantified using a variety of measures. Researchers have developed algorithms for optimizing sensor placements with respect to a number of design objectives, including the proportion of population exposed prior to detection [2], the volume of contaminated water consumed prior to detection [15], and the time to detection [17]. Researchers may choose an objective because it sounds reasonable and they are most able to determine or estimate correct parameters for that objective. Researchers might also implicitly assume that the selected design objective is the "best" objective for sensor placement; in some instances researchers have made this argument explicitly, e.g., see Kumar et al. [17]. However, there may be no single best design objective in reality. The variety of design objectives introduced by researchers supports this view, as there are valid arguments for the importance of all such objectives. Furthermore, existing objectives are not obviously redundant. For example, the number of failed detections (missed incidents) and the proportion of population exposed seem to provide complementary information.

Following the 9/11 terrorist attacks, Harvey Greenberg worked closely with researchers at Sandia National Laboratories and the US Environmental Protection Agency to develop effective optimization strategies for large-scale contamination sensor placement. This collaboration led to the first publication discussing multi-objective trade-offs in sensor placement for water security [31], which appeared in the Proceedings of the ASCE/EWRI Congress. This chapter is an extension of that prior research and the associated paper, which was co-authored with Harvey but not published. Specifically, this research extension considers the analysis of multi-objective trade-offs on larger distribution networks. Most researchers at the time had

considered sensor placement only in the context of small test networks, e.g., on the order of 100s of junctions. Concurrently, this research extension demonstrates the application of sensor placement techniques on large-scale networks. We chose not to rerun the original experiments, as the IP solver of the time obtained globally optimal solutions. In Sect. 7.2, we describe how this work still provides a contribution with respect to the current state of the art.

We explore in this chapter the trade-offs between different design objectives for sensor placement optimization in water distribution networks. We specifically pose and answer the following two research questions: (1) What is the relationship between optimal sensor placements for different design objectives? and (2) Is there any risk in focusing on specific design objectives? Our analysis considers the following six design objectives: population exposed, time to detection, volume of contaminated water consumed, mass of contaminant consumed, number of failed detections, and extent of contamination. See Sect. 7.3 for detailed descriptions of the objectives. The corresponding sensor placement optimization problem can be casted as the well-known p-median problem from facility location theory in each case. Using mixed-integer programming (MIP) models and commercial MIP solvers, we identify optimal solutions for all design objectives across a range of sensor budgets on each of three large test networks.

Our analysis of the resulting placements indicates that optimal solutions with respect to one design objective are often highly sub-optimal with respect to complementary design objectives. In other words, there may be significant risk associated with focusing a priori on specific design objectives. Our results reinforce the view that multiple objectives should be considered during the design of sensor placements for CWSs, as there are significant trade-offs that should be exposed to decision-makers. There is fortunately evidence that this risk can be mitigated in some circumstances; by sacrificing optimality in some design objectives, we demonstrate that it is possible to develop solutions that are more robust (i.e., higher-quality) with respect to secondary design objectives.

We review recent literature on multi-objective sensor placement for water security in Sect. 7.2 and discuss the continued relevance of the results presented in this chapter. In Sect. 7.3, we document the p-median formulation of the sensor placement problem and detail the computation of the various design objectives in our analysis. Section 7.4 describes our test networks, contamination scenarios, and aspects of our experimental methodology. We then analyze the behavior of individual design objectives on our test networks. The remainder of the section addresses multiple-objective analysis, focusing on the relationship between different design objectives. Section 7.5 reviews the implications of our analysis, including a discussion of the extent to which it may be possible to generate solutions that simultaneously yield high-quality solutions with respect to a range of design objectives.

7.2 Background and Overview

As noted previously, Watson et al. [31] was the first publication to analyze multi-objective trade-offs for sensor placement in water distribution systems [26]. At that time there was a growing literature on sensor placement techniques for CWSs, but papers typically reported results for individual design objectives. Propato [27] presented a similar modeling approach, using MIP to model the sensor placement optimization problem. His optimization formulations can represent different objectives by changing the formulation of the linear objective function, much like the models described here and in Watson et al. [31].

The *Battle of the Water Sensor Networks* (BWSN) was a comparison of sensor placement techniques that catalyzed significant interest in multi-objective sensor placement [25]. The BWSN focused on a comparison of sensor placement techniques considering four independent design objectives. A variety of multi-objective optimization techniques were consequently developed for this comparison. For example, Ostfeld and Salomons [24] and Preis and Ostfeld [26] describe multi-objective evolutionary algorithms, and Dorini et al. [8] developed a constrained multi-objective optimization framework based on a cross-entropy methodology. The final comparison considered in the BWSN involved an assessment of Pareto optimal points, even for methods that did not explicitly optimize multiple objectives. Researchers reporting on their methods used in BWSN subsequently emphasized their ability to support multi-objective analysis. For example, Krause et al. [16] describe fast sensor placement methods that optimize weighted multi-objective optimization, and they exploit the submodular structure of this problem to ensure near-optimality.

Several reviews of sensor placement research have been published since the BWSN [1, 12, 14, 28, 29]. These reviews illustrate several trends. First, the number of relevant design objectives has continued to increase, including design for water quality management, contaminant source identification, and risk measures that account for uncertainty. Second, researchers continue to explore the design of new multi-objective algorithms although most research considers heuristic methods—especially evolutionary algorithms. Finally, researchers have increasingly focused on fast methods with modest computer memory requirements that are robust to the limited information available to water utilities around the world. These last considerations reflect practical realities for water engineers who need to design contamination warning systems with limited resources and often with practical limitations on their ability to model their systems.

Considering the volume and focus of these subsequent works, the research in this chapter continues to be relevant. The methods we consider for sensor placement are based on MIP models and solvers, so we can guarantee Pareto optimal solutions. For example, these methods were used to generate solutions included in the BWSN [25], and in all cases these solutions were not dominated by solutions from other optimizers. It is noteworthy that few authors have considered MIP models and solvers subsequent to the BWSN comparison.

The focus of the research in this chapter is on multi-objective trade-offs. Thus, the fact that MIP solvers guarantee optimality ensures that our conclusions reflect the structure of the Pareto set. By comparison, multi-objective solutions generated by heuristic algorithms like evolutionary algorithms are not guaranteed to reflect Pareto optimality.

7.3 Problem, Objectives, and Mixed-Integer Formulation

The problem of placing contaminant sensors in a water distribution network to maximize the degree of afforded protection can be expressed as any of a number of standard problems in discrete location theory [20]. The specific selection depends on modeling decisions, e.g., whether sensor installation costs should be considered, or if the objective is to minimize expected or worst-case impact. We base our analysis on the well-known p-median facility location problem [7], building on our prior research efforts involving sensor placement optimization [3].

There are n customers and m potential facility locations in the p-median problem. Exactly p of the m potential facilities are actually "opened," where $1 \leq p \leq m$, and each customer is "served" by the nearest open facility. For any fixed p, the objective is to determine the subset of p open facilities that minimizes the sum of the distances between each customer and the nearest open facility. The p-median problem is NP-hard for unbounded $p \leq m$ although the p-median problem is not NP-hard for any fixed p [10] since there are $O(m^p)$ possible choices for p sensors. However, it is computationally intensive to determine optimal solutions for instances with even modest n and m [6] even for fixed p. The p-median problem is closely related to the p-center problem, in which the objective is to minimize the *maximum* distance between a customer and the nearest open facility; the latter, however, is significantly more difficult to solve in practice.

In the context of contamination sensor placement for water distribution networks, a "customer" corresponds to a particular contamination scenario, i.e., an injection of contaminant into the network. We assume sensors are placed at network junctions, including tanks and reservoirs, to mirror nearly all prior research on sensor placement optimization (see Berry et al. [2] for a noteworthy exception). The set of potential facility locations then corresponds to the set of network junctions. We assume a fixed budget of p general contaminant sensors, each placed at a specific junction. Let $\mathcal{S}$ denote the set of potential contamination scenarios, and let $\mathcal{L}$ denote the set of network junctions. We additionally define a "dummy" network junction q corresponding to an abstract location at which detection occurs via mechanisms external to the sensor network, e.g., through observation of population behaviors. The introduction of q reflects the fact that not all contamination scenarios are detectable by a physical sensor.

Let P denote the subset of junctions with installed sensors, where $|P| = p$ and $P \subseteq \mathcal{L}$. For each combination of $s \in \mathcal{S}$ and $j \in \mathcal{L}$, we define d_{sj} as the aggregate, network-wide "damage" incurred if scenario s is first detected by a

sensor at junction j, *assuming* a sensor is actually located at junction j. We further define d_{sq} for each $s \in \mathcal{S}$ as the network-wide damage incurred if scenario s is not detectable by any sensor in P. As discussed below, precise quantification of damage depends on the optimization objective; for illustrative purposes, the values d_{sj} can be interpreted as the number of people exposed to the injected contaminant. The design objective is then to minimize

$$\sum_{s \in \mathcal{S}} d_{sf(s,P\cup\{q\})}, \tag{7.1}$$

where $f(s, P \cup \{q\})$ denotes a $j \in P \cup \{q\}$ that minimizes d_{sj}.

To determine an optimal sensor placement P and the corresponding minimum impact quantity, we formulate the p-median problem as a mixed-integer (linear) program (MIP), which we then solve using a commercially available MIP solver. The MIP-related terms used throughout this paper are defined in the *Mathematical Programming Glossary* [11]. A MIP formulation of the p-median problem is

$$\text{Minimize} \quad \sum_{s \in \mathcal{S}} \sum_{j \in \mathcal{L}\cup\{q\}} d_{sj} x_{sj} \tag{7.2}$$

$$\text{Subject to} \quad \sum_{j \in \mathcal{L}\cup\{q\}} x_{sj} = 1 \qquad \forall s \in \mathcal{S} \tag{7.3}$$

$$x_{sj} \leq y_j \qquad \forall j \in \mathcal{L}, \forall s \in \mathcal{S} \tag{7.4}$$

$$\sum_{j \in \mathcal{L}} y_j = p \tag{7.5}$$

$$y_j \in \{0, 1\} \qquad \forall j \in \mathcal{L} \tag{7.6}$$

$$0 \leq x_{sj} \leq 1 \qquad \forall s \in \mathcal{S}, j \in \mathcal{L} \cup \{q\}. \tag{7.7}$$

The binary y_j variables determine whether a sensor is placed at a junction $j \in \mathcal{L}$. Linearization of Eq. (7.1) is achieved through the introduction of auxiliary variables x_{sj}, which indicate whether a sensor placed at junction j is the first to detect scenario s. Constraint (7.4) ensures that detection is possible only if a sensor exists at junction j. The x_{sj} variables are implicitly binary due to a combination of binary y_j, Constraint (7.4), and the objective-function pressure induced by Eq. (7.2). Constraint (7.3) guarantees that each scenario $s \in \mathcal{S}$ is first detected by exactly one sensor, either at q or in the set $\mathcal{L}$; ties are broken arbitrarily. Finally, the objective function (Eq. (7.2)) ensures that detection of a scenario s is assigned to the sensor-bearing junction $j \in \mathcal{L} \cup \{q\}$ that minimizes d_{sj} or to the non-detected cost d_{sq} if no sensor can detect s. The objective therefore minimizes the average or cumulative damage taken over all the scenarios.

We determine the impact of a potential contamination scenario via transport simulation. Specifically, we use EPANET [30] to generate a time-series τ_{sj} of

contaminant concentration at each junction $j \in \mathcal{L}$ for each scenario $s \in \mathcal{S}$. We use the resulting time-series to compute the network-wide impact d_{sj} of the scenario s assuming first detection via a sensor placed at junction j. More formally, let γ_{sj} denote the earliest time t at which a sensor at junction j can detect the contaminant associated with scenario s, e.g., when contaminant concentration reaches a specific detection threshold. If the contaminant from scenario s fails to reach junction j, then $\gamma_{sj} = t^*$, where t^* denotes either the end of the simulation or an appropriate user-specified delay. We next define $d_{sj} = d_s(\gamma_{sj})$, i.e., the aggregate, network-wide damage incurred if scenario s is first detected at time γ_{sj}. In our analysis, $d_{sq} = d_s(t^*)$. We assume without loss of generality that a sensor placed at a junction $j \in \mathcal{L}$ is capable of immediately detecting any scenario $s \in \mathcal{S}$ at j once non-zero concentration levels of a contaminant are present. We finally assume that both consumption and propagation of contaminant are immediately terminated once detection occurs. The model can handle a delay for termination of damage accumulation, but we are not aware of realistic alarm procedures and mitigation strategies to give a reasonable approximate delay value.

By isolating objective-specific information to the d_{sj} coefficients, the p-median MIP seamlessly allows for optimization of disparate design objectives. We consider the following objectives in our analysis, variants of which have previously been considered by at least one research group—we briefly consider any key factors in the computation of these objectives from the set of τ_{sj} where necessary:

Population Exposed (*pe*) This objective quantifies the number of people *sickened* by exposure to the injected contaminant, as defined by the demand-based model described in Murray et al. [21]. The authors of this chapter can provide specific values for the numerous parameters in the dosage-response computation upon request. Alternative models of population exposure have assumed the availability of population estimates on a per-junction basis [2, 31]. While correcting the obvious deficiency of demand-based models, reliable estimates of population density over time are generally unavailable.

Time to Detection (*td*) This objective quantifies the time, measured in minutes, between the initiation of an injection and the earliest presence of non-zero contaminant concentration at a junction with a sensor. Watson et al. [31] previously considered this objective in the context of a flow-averaged model of the sensor placement problem.

Volume of Contaminated Water Consumed (*vc*) This objective quantifies the total volume of water, measured in gallons, extracted from the system prior to detection of non-zero contaminant concentration at a junction with a sensor. Extraction occurs at any junction—excepting tanks and reservoirs—with a positive, non-zero demand; the computation is independent of the magnitude of contaminant concentration. This objective is among the most widely studied, having previously been examined in [15, 23, 31].

Mass of Contaminant Consumed (*mc*) This objective quantifies the total "mass" of injected contaminant, quantified in terms of the number of biological organisms,

extracted from the system prior to the presence of non-zero contaminant concentration at a junction with a sensor. In contrast to vc, the mc objective is sensitive to the concentration of the contaminant. This objective was previously considered in [3, 31].

Number of Failed Detections (*nfd*) This objective quantifies the proportion of contamination scenarios for which no sensor detects non-zero contaminant concentration, i.e., the contamination scenario is "detected" by a sensor at the dummy q junction. Watson et al. [31] previously examined this objective.

Extent of Contamination (*ec*) This objective quantifies the length of pipe in a distribution system, measured in linear feet that has been directly exposed to non-zero contaminant concentration. The entire length of an individual pipe is considered to be contaminated if (1) non-zero contaminant concentration is present at an end-point junction j and (2) water flow (as determined via contaminant transport simulation) enters the pipe from j; consequently, the measure is conservative. Watson et al. [31] introduced this objective.

The pe, vc, and mc objectives are arguably related, in that they attempt to quantify—either implicitly or explicitly—the impact of a contamination scenario on a population. However, the relationship between these and the remaining objectives is less clear.

The straightforward formulation of the p-median MIP above is computationally tractable for small water distribution networks, but it suffers from serious scalability limitations when the number of network junctions exceeds several thousand. We solve a more complex variant of the above formulation to facilitate analysis of large-scale distribution networks, a tactic that yields identical solutions in significantly shorter run-times. We use the improved MIP formulation from Berry et al. [3], which is described here for completeness, to generate the results presented in Sect. 7.4.

The improved MIP formulation from Berry et al. [3] exploits a common property of water quality simulations run with a coarse time step: for a given scenario s, there are frequently many locations i and j such that $d_{sj} = d_{si}$. Informally, this indicates that contamination in scenario s reaches locations i and j within the same time step. We call such locations i and j *equivalent* with respect to detecting scenario s. In the previous MIP model we let $\mathcal{L}$ denote the set of all network locations. We now let $\hat{\mathcal{L}}_s$ denote the set of locations impacted by contamination from scenario s such that for all $i, j \in \hat{\mathcal{L}}_s$, we have $d_{sj} \neq d_{si}$. That is, the set $\hat{\mathcal{L}}_s$ contains exactly one representative location from any set of equivalent locations for scenario s. The new MIP model is then

$$\text{Minimize} \quad \sum_{s \in \mathcal{S}} \sum_{j \in \hat{\mathcal{L}}_s \cup \{q\}} d_{sj} x_{sj} \tag{7.8}$$

$$\text{Subject to} \quad \sum_{j \in \hat{\mathcal{L}}_s \cup \{q\}} x_{sj} = 1 \qquad \forall s \in \mathcal{S} \tag{7.9}$$

$$x_{sj} \leq y_j + \sum_{i \in \mathcal{L} \setminus \hat{\mathcal{L}}_s : d_{sj} = d_{si}} y_i \qquad \forall j \in \hat{\mathcal{L}}_s, \forall s \in \mathcal{S} \tag{7.10}$$

$$\sum_{j \in \mathcal{L}} y_j = p \tag{7.11}$$

$$y_j \in \{0, 1\} \qquad \forall j \in \mathcal{L} \tag{7.12}$$

$$0 \leq x_{sj} \leq 1 \qquad \forall s \in \mathcal{S},\ j \in \mathcal{L} \cup \{q\}. \tag{7.13}$$

This formulation has fewer x_{sj} variables for a given scenario s: one for each unique value of d_{sj}. The constraints operate similarly to those in the previous MIP model except for Constraint (7.10). Suppose location j is the representative for all of its equivalent locations with respect to scenario s. Variable x_{sj} is allowed to take the value of 1, signaling an observation of scenario s at time γ_{sj}, if there is a real sensor at any location with the same timing value for scenario s. The placement variable for every such equivalent location is on the right-hand side of Constraint (7.10) for each scenario s.

We now discuss various assumptions underlying our p-median MIP formulation of the sensor placement problem. The most critical and unrealistic assumption is the availability of general-purpose, perfect contaminant sensors. We base our decision to proceed under this assumption on two factors. First, sensor performance characteristics are not currently well-understood (although there is ongoing research in this area, e.g., see Liu et al. [18] and McKenna et al. [19]). Furthermore, data for real-world distribution networks is limited. Second, the tractability of computational techniques for imperfect-sensor variations of the p-median problem lags that of the perfect-sensor formulation [4], making extensive studies of the form presented in Sect. 7.4 infeasible. Although not considered here, one can easily extend the p-median formulation to handle a number of real-world constraints and factors, including fixed and/or invalid sensor locations, delays in raising a general alarm after detection by a sensor, thresholds on contaminant concentration, a probability distribution on the attack scenarios, and installation costs [3, 5]. Many of these extensions involve straightforward modifications to the computation of the d_{sj} damage coefficients, which should not greatly affect the tractability of the MIP models. These extensions require more data that utilities may not maintain or be able to acquire. But in some cases, researchers are suggesting guesses. For example He et al. [13] suggest attack probability distributions based on demands, flow rates, or pipe length.

7.4 Experimental Results and Analysis

We now use the p-median model introduced in Sect. 7.3 to analyze the relationship between different design objectives for three large, real-world water distribution

networks. We coded the p-median MIP formulation of the sensor placement optimization problem using the AMPL modeling language [9] and solved the resulting MIPs to optimality using ILOG's AMPL/CPLEX 9.1 commercial solver package.[1] We defer to Berry et al. [3] for a discussion of the computational characteristics of the p-median MIP model and alternative heuristic techniques for its solution. We describe the test networks and contamination scenarios in Sect. 7.4.1. In Sect. 7.4.2 we discuss the nature of individual design objectives for the test networks given a range of sensor budgets. In Sect. 7.4.3 we analyze the impact of optimization for a single design objective on complementary design objectives.

7.4.1 The Test Networks and Contamination Scenarios

We report computational results for three real, large-scale municipal water distribution networks. The networks are denoted simply as Network1, Network2, and Network3; the identities of the corresponding municipalities are withheld due to security concerns. Network1 consists of roughly 400 junctions, 500 pipes, and a small number of tanks and reservoirs. Network2 consists of roughly 3000 junctions, 4000 pipes, and roughly 50 tanks and reservoirs. Network3 consists of roughly 12,000 junctions, 14,000 pipes, and a handful of reservoirs; there are no tanks or well sources in this municipality. All of the models are skeletonized although the degree of skeletonization in Network1 and Network2 is much greater than in Network3.

Figures 7.1, 7.2, and 7.3 depict graphical representations of Network1, Network2, and Network3, respectively. We manually "morphed" or altered (e.g., through pipe lengthening or coordinate translation/rotation) key topological features of each original network structure to inhibit identification of the source municipality. Local topologies were largely preserved in this process, such that the graphics faithfully capture the overall characteristics of the underlying network structures. Sanitized versions of all three networks, in the form of EPANET input files, are available from the authors. While these files contain no coordinate information, all data other than that relating to labels (which have been anonymized) are unaltered. All computed hydraulic and water quality information thus accurately reflects (within the fidelity limits of the data and the computational model) the dynamics of the source municipality.

We simulated network hydraulics for 96 h, representing multiple iterations of a typical daily demand cycle. We defined a single contamination scenario for each junction with non-zero demand. Injection for each scenario starts at time $t = 0$ and continues for 12 h. We model scenarios as biological mass injections with a constant rate of $5.78e^{10}$ organisms per minute. Although not considered here, the p-median

[1] At the time of this writing, the latest version of CPLEX Optimization Studio is 12.9—now available from IBM.

Fig. 7.1 Graphical depiction of Network1 topology. See text for details

Fig. 7.2 Graphical depiction of Network2 topology. See text for details

Fig. 7.3 Graphical depiction of Network3 topology. See text for details

formulation—via the d_{sj}—allows arbitrarily complex attack scenarios. For example, one could model multiple simultaneous injection sites with different contaminants at variable injection strengths and durations.

We assume uniform scenario probabilities, so that all results (defined by Eq. (7.1)) are normalized by the number of non-zero demand junctions to obtain an expectation. We ran water quality simulations for each scenario with a time-step resolution of 5 min. We used the resulting τ_{sj} to compute the impact parameters d_{sj} for the various design objectives, as previously described in Sect. 7.3. We used EPANET [30] for all hydraulic and water quality simulations.

7.4.2 Characteristics of Individual Design Objectives

Most of the design objectives introduced in Sect. 7.3 have either been considered only in the context of small test networks or on less accurate formulations of the sensor placement problem. Furthermore, optimal-performance case studies of sensor configurations for large-scale, real-world distribution networks are of interest to practitioners and researchers but are absent in the broader literature.

Table 7.1 Optimal values of design objectives for a range of p on Network1

p	pe	td	vc	mc	nfd	ec
0	2445	5760	1,288,000	3.95e+13	1.0	41,268
5	143	1600	8357	1.69e+13	0.13	4084
10	63	985	3010	1.10e+13	0.05	2444
25	20	219	660	3.43e+12	0.0	982
50	5	41	158	3.79e+11	0.0	375
100	0	0	2	4.36e+08	0.0	0
200	0	0	0	0	0.0	0
400	0	0	0	0	0.0	0

The units of measure for the various design objectives are, respectively, number of individuals, minutes, gallons, organisms, proportion of total contamination scenarios not detected, and linear feet

Consequently, we first consider the nature of the individual design objectives on our test networks.

Table 7.1 reports the performance of optimal sensor placements for each of our design objectives on Network1, over a range of sensor budgets p. In the absence of sensors, the mean impact of a contamination scenario is significant, especially for the pe, vc, and ec objectives. Placing even $p = 5$ sensors yields an order-of-magnitude or larger reduction in many of the objectives, including pe, vc, nfd, and ec. Independent of objective, a budget of only $p = 10$ is sufficient to yield impacts of at most 28% of the $p = 0$ solution; for many objectives, e.g., pe and nfd, the impact is approximately 5% of the $p = 0$ solution. For all but the mc objective, $p = 50$ is sufficient to achieve near-perfect protection. In all cases, excellent performance can be achieved with a budget p equal to a small fraction of the total number of network junctions.

We report in Table 7.2 the performance of optimal sensor placements for each of our design objectives on Network2; recall that Network2 is roughly an order-of-magnitude larger in terms of the number of network elements than Network1. Considering the $p = 0$ solution, there is a marked growth in impact relative to Network1 for the objectives that do not have a fixed maximum (nfd with a maximum of 100% and td with a maximum equal to the (consistent) length of simulation). This growth in maximum average impact for Network2 relative to Network 1 is consistent given the differences in network size and roughly equivalent degrees of skeletonization. In terms of pe, over 14,000 individuals are sickened *on average* across the range of possible contamination scenarios, while large numbers of specific scenarios (not reported) impact far larger numbers of individuals. Similarly, approximately 66 miles of pipe are exposed to the contaminant on average, while 11.7 million gallons of contaminated water are extracted from the distribution network prior to detection. Despite the network size, $p = 5$ is still sufficient to yield an order-of-magnitude or greater reduction in damage relative to the $p = 0$ solution for the pe, vc, and ec objectives. Relative to the results for Network1, a large ($p = 100$) number of sensors are required to reduce impacts

Table 7.2 Optimal values of design objectives for a range of p on Network2

p	pe	td	vc	mc	nfd	ec
0	14,217	5760	11,667,200	3.90e+13	1.0	344,376
5	1709	3218	162,640	2.71e+13	0.47	38,822
10	1061	2860	66,241	2.44e+13	0.41	22,062
50	347	2028	13,675	1.74e+13	0.29	6382
100	205	1632	7549	1.42e+13	0.23	3604
500	50	124	1527	2.51e+12	0.0	754
1000	14	11	272	7.18e+10	0.0	84
2000	0	0	0	0	0.0	0

Table 7.3 Optimal values of design objectives for a range of p on Network3

p	pe	td	vc	mc	nfd	ec
0	2249	5760	978,487	4.15e+13	1.0	138,543
5	764	4523	98,751	3.63e+13	0.69	41,623
10	498	4134	52,354	3.41e+13	0.62	26,973
50	169	3224	10,112	2.72e+13	0.46	8801
100	103	2832	5305	2.43e+13	0.39	5424
500	34	1642	1311	1.58e+13	0.20	1820
1000	20	987	672	1.08e+13	0.09	1153
2000	11	310	287	5.14e+12	0.0	684

to about 28% or less than that of the $p = 0$ solution for all objectives except mc, while slightly over 1000 sensors are required to achieve near-perfect protection. We observe that both results are consistent with the differences in network size.

Finally, we consider individual design objective results for Network3, reported in Table 7.3. Despite the larger size relative to Network2, the mean impacts under $p = 0$ for the network-dependent performance measures (pe, vc, mc, and ec) are comparable to those reported for Network1. This is due to the lesser degree of skeletonization used in the development of the Network3 model. Given the absolute network size, a very small budget of 5 sensors yields significant protection relative to the baseline $p = 0$ solution. However, very large numbers of sensors ($p \approx 1000$) are required to yield impacts of at most 28% of the $p = 0$ solution. While small relative to the total number of network junctions, such large budgets would represent a significant investment for a water utility.

Overall, the results in Tables 7.1 through 7.3 demonstrate that independent of objective, a very small number of sensors can yield large—and often order-of-magnitude—reductions in impacts relative to the $p = 0$ solution. In all of our test networks, a sensor budget equal to at most 10% of the total number of network junctions yields reductions in the mean impact of an contamination scenario, quantified by any design objective, of 80% or more relative to the $p = 0$ solution. In other words, a very small number of sensors, in both relative and absolute terms, provides a significant degree of protection. As the number of sensors grows, however, the per-sensor benefit diminishes greatly.

7.4.3 The Impact of Optimization on Competing Objectives

Several researchers have argued for the use of specific design objectives to develop sensor placements in water distribution networks [15, 17]. Populations would undoubtedly prefer minimization of pe as the primary objective. In contrast, potential economic impacts influence both adversaries and decision-makers, leading to an argument for minimization, or at least consideration, of ec. Similarly, it is not unreasonable to expect a CWS to detect a large proportion of possible contamination scenarios within a reasonable time-frame (nfd). In any case, no single view is likely to prevail, and planners will realistically have to understand the trade-offs between the various design objectives. The objectives of interest are ideally highly correlated so that optimal solutions with respect to one objective yield near-optimal solutions with respect to others.[2] Unfortunately, as we now discuss, we do not necessarily observe this behavior on our test networks.

It is important to note before proceeding that solution quality with respect to secondary objectives is entirely ignored if optimizing any individual objective. So given a problem with multiple globally optimal solutions with respect to the primary objective, we are not selecting among the "best" with respect to any particular secondary objective. Rather, we are simply using the particular solution returned by our MIP solver. However, there is no evidence that such consideration would influence the results presented below.

We begin with an example in which we investigate how minimization of the nfd objective on Network1 impacts solution quality with respect to the other design objectives. Given $p = 25$, $nfd = 0.0$, i.e., all contamination scenarios are detected within the 96-h simulation period. However, the solution that yields an optimal value of nfd given $p = 25$ also results in $pe = 234$, which represents a 1170% deviation from the optimal value of $pe = 20$ given $p = 25$ (as shown in Table 7.1). Similarly large deviations are observed for ec and vc (677% and 1017%, respectively), while the values of td and mc represent "only" 180% and 223% deviations from optimality, respectively. Although such large deviations were unexpected, we acquired a qualitative understanding of the underlying causes via straightforward *a posteriori* analyses. For example, minimization of nfd subject to a limited sensor budget tends to yield sensor placements near the leaves, or end-points, of the distribution network. In doing so, many upstream nodes are exposed to contaminant for longer durations, resulting in greater overall ingestion.

Characterizing and analyzing the interactions between all of the design objectives introduced in Sect. 7.3 are beyond the scope of this paper. Rather, we examine illustrative cases, one for each of our test networks. In the first case, we consider how optimization of pe on Network1 impacts solution quality, in terms of deviation from optimality, with respect to the complementary design objectives. The results are shown in Table 7.4 for a range of sensor budgets; for each complementary

[2] We use this informal notion of correlation throughout, as opposed to the more familiar concept of statistical correlation.

Table 7.4 Percentage and absolute deviations from optimality for complementary design objectives on Network1, given a pe-optimal solution

	td		vc		mc		nfd		ec	
p	%	Abs.	%	Abs.	%	Abs.	%	Abs.	%	Abs.
5	73	1160	206	17,212	30	5.10e+12	246	0.32	42	1707
10	96	944	42	1263	58	6.40e+12	440	0.22	46	1119
25	400	875	31	205	174	5.98e+12	∞	0.15	68	663
50	1685	513	58	91	1396	5.29e+12	∞	0.11	107	403
100	∞	74	700	14	87,385	3.81e+11	0	0.0	∞	106
200	0	0	0	0	∞	1.16e+8	0	0.0	∞	26
400	0	0	0	0	0	0	0	0.0	0	0

objective, we report both the absolute and percentage difference from optimality (e.g., determined in part using the data recorded in Table 7.1). We observe that large absolute deviations do not necessarily correspond to large percentage deviations, and vice versa. Values of ∞ in a percentage-difference column indicate the optimal value for the corresponding objective given p is 0. The results indicate two specific trends. First, for small-to-moderate p ($5 \leq p \leq 50$), pe is not significantly correlated with any of the other objectives; both percentage and absolute deviations from the optimal values of the complementary objectives are unexpectedly and uniformly large. Second, for large p ($p \geq 100$), the correlation between pe and the complementary objectives begins to improve, reaching near-perfect correlation once $p \approx 200$. The convergence as $p \rightarrow |\mathcal{L}|$ is expected: as a larger number of network junctions are covered by sensors, the similarity between the optimal placements for different objectives necessarily increases. The differences are unfortunately greatest in the most likely regime for CWS deployment—when the sensor budgets p are small.

We next consider a similar analysis on the larger Network2 model, in which we analyze how optimization of nfd impacts performance in terms of secondary design objectives. Table 7.5 gives the results, presented in an analogous form to those shown in Table 7.4. In contrast to the results for Network1, we observe fairly strong correlation between some objectives. Specifically, optimal nfd solutions yield near-optimal td performance, and only moderately worse mc performance for up to about 100 sensors. However, by 1000 sensors, the deviations are quite large. For small-to-moderate sensor budgets ($5 \leq p \leq 100$), optimal nfd solutions yield significant absolute and percentage deviations from optimality for the pe, ec, and vc. These deviations further persist even for large sensor budgets of $p = 1000$ and greater. Comparison of the Network1 and Network2 results further reinforces the general observation that optimization with respect to a specific design objective can yield highly sub-optimal performance with respect to secondary objectives. However, the results also indicate that the degree of sub-optimality appears to be dependent upon both the test network and the design objectives under consideration.

Finally, for Network3 we consider the impact of optimization of vc on secondary design objectives; the results are shown in Table 7.6. Relative to the results shown in

Table 7.5 Percentage and absolute deviations from optimality for complementary design objectives on Network2, given a nfd-optimal solution

	pe		td		vc		mc		ec	
p	%	Abs.	%	Abs.	%	Abs.	%	Abs.	%	Abs.
5	472	8062	11	342	819	1,332,100	28	7.60e+12	567	220,233
10	773	8204	12	338	1651	1,093,519	37	9.10e+12	980	216,231
50	933	3237	10	203	1779	243,245	44	7.60e+12	1467	93,647
100	1030	2111	8	127	1473	111,161	37	5.30e+12	1742	62,793
500	844	422	72	89	824	12,578	91	2.20e+12	1490	11,234
1000	1586	222	1100	121	2526	6872	3897	2.80e+12	6636	5574
2000	∞	58	∞	53	∞	2385	∞	1.10e+12	∞	2077

Table 7.6 Percentage and absolute deviations from optimality for complementary design objectives on Network3, given a vc-optimal solution

	pe		td		mc		nfd		ec	
p	%	Abs.	%	Abs.	%	Abs.	%	Abs.	%	Abs.
5	4	30	3	115	<1	3.00e+11	10	0.07	10	4324
10	7	33	5	190	<1	2.00e+11	15	0.09	17	4518
50	14	24	9	283	2	5.00e+11	24	0.11	35	3101
100	22	23	12	344	2	6.00e+11	33	0.13	46	2475
500	35	12	37	612	12	1.90e+12	80	0.16	75	1371
1000	50	10	73	723	25	2.70e+12	200	0.18	94	1089
2000	55	6	240	743	63	3.25e+12	∞	0.16	121	829

Tables 7.4 and 7.5, the deviations from optimality for the secondary objectives are significantly lower in both percentage and absolute terms (e.g., most deviations are less than 100%), and there is stronger correlation between many of the objectives (e.g., pe, mc, and ec). Network3 is supplied strictly through reservoirs, in contrast to Network1 and Network2. The lack of tanks strongly limits the flow dynamics, which partially explains both the lower deviations from optimality and the stronger correlations observed between many of the objectives.

Our results support three general conclusions. First, there are significant risks associated with optimization of sensor placements with respect to any particular design objective. The results in Tables 7.4, 7.5, and 7.6 demonstrate that optimal solutions with respect to any specific design objective can be far from optimal with respect to a range of complementary objectives. Second, and counter-intuitively, the lack of significant correlation between objectives may not improve with small-to-moderate increases in p. In other words, a large sensor budget does not necessarily mitigate the risk associated with optimization with respect to a single design objective. Third, the nature of the correlation between various objectives is highly problem-dependent, suggesting that a comprehensive analysis is required on a per-network basis.

7.5 Compromise Solutions

While there is significant risk associated with focusing on any individual design objective in sensor placement optimization, it is unclear whether it may be possible to construct a solution that more carefully balances a range of design objectives, or whether there exist objectives that are strictly conflicting. We again consider minimization of pe on Network1 given $p = 10$ to explore this question, but we impose additional constraints on the corresponding p-median MIP described in Sect. 7.3 so that the values of td, mc, and vc are constrained to be no greater than 50, 20, and 30% of their optimal values, respectively (as recorded in Table 7.1). While the optimal value of pe increases 60% relative to the baseline MIP without the side constraints, the deviations from optimality for all but one of the other design objectives are significantly reduced relative to the baseline MIP results, as reported in Table 7.4. Specifically, the deviation from optimality given the additional side constraints is 29% for td (down from 96%), 28% for vc (down from 42%), 20% for mc (down from 58%), and 180% for nfd (down from 440%). However, the deviation from optimality for ec grows from 46 to 78%, indicating that at least for Network1, the pe and ec objectives are strongly complementary.

We omit a broader analysis of the trade-offs between the various design objectives due to space limitations. We rather observe that by sacrificing solution quality with respect to a primary design objective, it is possible to gain significant improvements with respect to secondary objectives and avoid some of the brittleness associated with solutions that are optimal with respect to individual design objectives. However, due to the competing nature of some design objectives on some networks, it is not always possible to improve the performance of all secondary objectives simultaneously. Ultimately, a detailed understanding of the relationship between various design objectives is required for decision-makers to develop robust sensor placements for CWS deployment. Such understanding is even more crucial in the early phases of CWS deployment, where sensor budgets are small and the correlation between the optimality of different design objectives is usually weak. As the sensor budgets grow to cover a significant fraction of the network, this correlation tends to increase, and the need for multiple-objective analysis is less crucial.

7.6 Conclusions

Most research on contaminant sensor placement optimization in water distribution networks presupposes a given, fixed design objective. Several disparate design objectives have been proposed, and there are associated arguments—both implicit and explicit—for why one particular objective should be preferred over another. Yet, preference for any fixed objective is potentially risky given the current lack of understanding of the relationships among the proposed objectives. We have characterized

some of the inter-dependencies among a range of optimization objectives on three large-scale test networks. The majority of these objectives are uncorrelated, in that optimal solutions with respect to any one objective are often highly sub-optimal with respect to complementary objectives. Furthermore, increasing the number of sensors frequently fails to improve the correlation. However, these risks can be mitigated in some circumstances by considering solutions that are sub-optimal with respect to all performance objectives, which in turn requires a thorough understanding of how different objectives are related. Overall, the implications of our results for both researchers and planners are clear: algorithms for the sensor placement problem must carefully and simultaneously consider multiple design objectives.

Acknowledgments This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes. This work was performed in part under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

References

1. O.S. Adedoja, Y. Hamam, B. Khalaf, R. Sadiku, A state-of-the-art review of an optimal sensor placement for contaminant warning system in a water distribution network. Urban Water J. **15**(10), 985–1000 (2018)
2. J. Berry, L. Fleischer, W.E. Hart, C.A. Phillips, J.-P. Watson, Sensor placement in municipal water networks. J. Water Resour. Plann. Manag. **131**(3), 237–243 (2005)
3. J. Berry, W.E. Hart, C.A. Phillips, J.G. Uber, J.-P. Watson, Sensor placement in municipal water networks with temporal integer programming models. J. Water Resour. Plann. Manag. **132**(4), 218–224 (2006)
4. J. Berry, R.D. Carr, W.E. Hart, V.J. Leung, C.A. Phillips, J.-P. Watson, Designing contamination warning systems for municipal water networks using imperfect sensors. J. Water Resour. Plann. Manag. **135**(4), 253–263 (2009)
5. J.W. Berry, W.E. Hart, C.A. Phillips, J.G. Uber, T.M. Walski, Water quality sensor placement in water networks with budget constraints, in *Proceedings of the ASCE/EWRI Congress* (2005)
6. T.G. Crainic, M. Gendreau, P. Hansen, N. Mladenovic, Cooperative parallel variable neighborhood search for the *p*-median. J. Heuristics **10**, 293–314 (2004)

7. M.S. Daskin, *Network and Discrete Location: Models, Algorithms, and Applications* (Wiley, New York, 1995)
8. G. Dorini, P. Jonkergouw, Z. Kapelan, F. di Pierro, S.T. Khu, D. Savic, An efficient algorithm for sensor placement in water distribution systems, in *Proceedings of the 2006 Symposium on Water Distribution Systems Analysis* (2006)
9. R. Fourer, D.M. Gay, B.W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*, 2nd edn. (Duxbury Press, 2002)
10. M.R. Garey, D.S. Johnson, *Computers And Intractability: A Guide to the Theory of NP-Completeness* (W.H. Freeman and Company, New York, 1979)
11. H.J. Greenberg, *Mathematical Programming Glossary*. World Wide Web (1996–2018). https://glossary.informs.org
12. W.E. Hart, R. Murray, A review of sensor placement strategies for contamination warning systems in drinking water distribution systems. J Water Resour. Plan. Manag. **136**(6), 611–619 (2010)
13. G. He, T. Zhang, F. Zheng, Q. Zhang, An efficient multi-objective optimization method for water quality sensor placement within water distribution systems considering contamination probability variations. Water Res. **143**, 165 – 175 (2018). ISSN: 0043-1354. https://doi.org/10.1016/j.watres.2018.06.041
14. C. Hu, M. Li, D. Zeng, S. Guo, A survey on sensor placement for contamination detection in water distribution systems. Wirel. Netw. **24**(2), 647–661 (2018). ISSN: 1022-0038. https://doi.org/10.1007/s11276-016-1358-0
15. A. Kessler, A. Ostfeld, G. Sinai, Detecting accidental contaminations in municipal water networks. J. Water Resour. Plan. Manag. **124**(4), 192–198 (1998)
16. A. Krause, J. Leskovec, C. Guestrin, J. Vanbriesen, C. Faloutsos, Efficient sensor placement optimization for securing large water distribution networks. J. Water Resour. Plan. Manag. **134**, 516–526 (2008)
17. A. Kumar, M.L. Kansal, G. Arora, Discussion of "Detecting accidental contaminations in municipal water networks". J. Water Resour. Plan. Manag. **124**(4), 308–310 (1998)
18. S. Liu, H. Che, K. Smith, L. Chen, Contamination event detection using multiple types of conventional water quality sensors in source water. Environ. Sci.: Process. Impacts **16**, 2028–2038 (2014). https://doi.org/10.1039/C4EM00188E
19. S.A. McKenna, D.B. Hart, L. Yarrington, Impact of sensor detection limits on protecting water distribution systems from contamination events. J. Water Resour. Plan. Manag. **132**(4), 305–309 (2006)
20. P.B. Mirchandani, R.L. Francis (eds.), *Discrete Location Theory* (Wiley, New York, 1990)
21. R. Murray, J. Uber, R. Janke, Model for estimating acute health impacts from consumption of contaminated drinking water. J. Water Resour. Plan. Manag. **132**(4), 293–299 (2006)
22. R. Murray, W. E Hart, C.A. Phillips, J. Berry, E.G. Boman, R.D. Carr, L.A. Riesen, J.-P. Watson, T. Haxton, J.G. Herrmann, et al., US Environmental Protection Agency uses operations research to reduce contamination risks in drinking water. Interfaces **39**(1), 57–68 (2009)
23. A. Ostfeld, E. Salomons, Optimal layout of early warning detection stations for water distribution systems security. J. Water Resour. Plan. Manag. **130**(5), 377–385 (2004)
24. A. Ostfeld, E. Salomons, Sensor network design proposal for the battle of the water sensor networks (BWSN), In *Proceedings of the 2006 Symposium on Water Distribution Systems Analysis* (2006)
25. A. Ostfeld, J.G. Uber, E. Salomons, J.W. Berry, W.E. Hart, C.A. Phillips, J.-P. Watson, G. Dorini, P. Jonkergouw, Z. Kapelan, F. di Pierro, S.-T. Khu, D. Savic, D. Eliades, M. Polycarpou, S.R. Ghimire, B.D. Barkdoll, R. Gueli, J.J. Huang, E.A. McBean, W. James, A. Krause, J. Leskovec, S. Isovitsch, J. Xu, C. Guestrin, J. VanBriesen, M. Small, P. Fischbeck, A. Preis, M. Propato, O. Piller, G.B. Trachtman, Z.Y. Wu, T. Walski, The battle of the water sensor networks (BWSN): a design challenge for engineers and algorithms. J. Water Resour. Plan. Manag. **134**(6), 556–568 (2008)

26. A. Preis, A. Ostfeld, Multiobjective contaminant sensor network design for water distribution systems. J. Water Resour. Plan. Manag. **134**(4), 366–377 (2008)
27. M. Propato, Contamination warning in water networks: general mixed-integer linear models for sensor location design. J. Water Resour. Plan. Manag. **132**(4), 225–233 (2006)
28. S. Rathi, R. Gupta, A critical review of sensor location methods for contamination detection in water distribution networks. Water Quality Res. J. **50**(2), 95–108 (2014)
29. S. Rathi, R. Gupta, L. Ormsbee, A review of sensor placement objective metrics for contamination detection in water distribution networks. Water Supply **15**(5), 898–917 (2015)
30. L.A. Rossman, The EPANET programmer's toolkit for analysis of water distribution systems, in *Proceedings of the Annual Water Resources Planning and Management Conference* (1999)
31. J.-P. Watson, H.J. Greenberg, W.E. Hart, A multiple-objective analysis of sensor placement optimization in water networks, in *Proceedings of the ASCE/EWRI Congress* (2004)

Chapter 8
A Simplex Approach to Solving Robust Metabolic Models with Low-Dimensional Uncertainty

Allen Holder and Bochuan Lyu

Abstract We address the problem of solving difficult metabolic models that arise in the study of flux balance analysis (FBA). FBA problems are regularly linear due to simplifying assumptions although quadratic, combinatorial, and robust extensions are pragmatic variations. All such extensions inherit an underlying computational difficulty from the linear model, although in many instances this concern can be avoided by selecting an appropriate solution algorithm. Robust extensions unfortunately lack a trustworthy computational standard and are thus difficult to solve and problematic to employ. We show that a robust model's optimal value can be calculated by coupling standard nonlinear schemes with a technique of successive linear approximation, and we further indicate how the computational outcome might differ from the intent of the original robust model. We test our algorithm on two simple, motivating examples and on a standard FBA problem.

8.1 Introduction

Flux balance analysis (FBA) is the study of metabolic systems that have been expressed as computational and mathematical models. Most FBA models are optimization problems principled on the biological assumptions of maximal growth rate and steady metabolic state. Such models were proposed as early as the 1980s [11, 25], and the field has flourished due to its wide-ranging efficacy to mimic and predict biological outcomes [20, 22]. For instance, numerous models accurately predict gene knockouts [20, 22], while others identify how chemical pathways change to accommodate biochemical occlusion [8], advance synthetic biological design [18, 23], vet new biochemical systems [24], or suggest how to hinder cancer migration [27]. The applications are many and burgeoning.

A. Holder (✉) · B. Lyu
Department of Mathematics, Rose–Hulman Institute of Technology, Terre Haute, IN, USA
e-mail: holder@rose-hulman.edu

A. Holder (ed.), *Harvey J. Greenberg*, International Series in Operations Research & Management Science 295, https://doi.org/10.1007/978-3-030-56429-2_8

The success of FBA is due in large part to our ability to accurately and efficiently solve optimization models that represent metabolic systems. The authors of [9] challenged this success by claiming that much of FBA's research stemmed from computational tolerances that obscured a model's lack of fidelity to represent an intended metabolic network. The authors specifically argued that many metabolic models were incapable of biomass production with exact arithmetic, and thus, these models did not approximate a living organism's metabolism. The work in [10] refuted these claims and demonstrated agreement between exact and approximate solvers for the majority of models in question. The primary issue seems to have been unclear standards with regard to model generation.

Related to this computational discussion is the fact that FBA models are generally understood to be difficult instances within their problem class, and computational performance is regularly sensitive to the choice of algorithm and implementation. A primary concern is the amount of degeneracy, a topic that has been studied algorithmically with Monte Carlo sampling by Wiback et al. [26]. Algorithm choice is particularly problematic for robust analysis of metabolic pathways (RAMP), which is an FBA adaptation that relaxes the dubitable assumption of steady state [19]. RAMP models are difficult second-order cone problems (SOCPs) that do not lend themselves to interior-point algorithms, and since SOCP solvers are interior-point schemes, RAMP models are without a steadfast computational platform. This computational hindrance is sidestepped in [19] by restricting uncertainty to the biomass equation, and while this limitation provides computational efficacy with a simplex algorithm after a model reformulation, it also reduces viability. We address this limitation by combining nonlinear schemes with a simplex algorithm to solve general RAMP models. The resulting algorithm is generally appropriate for robust models with low-dimensional uncertainty for which the objective value, and not the optimal solution, is paramount.

8.2 Motivation and Problem Statement

Consider the SOCP,

$$\max \left\{ c^T x : x \geq 0,\ A_i x + \|R_i\, x\| \leq b_i,\ \text{for } i = 1, 2, \ldots, m \right\}, \tag{8.1}$$

where A_i is the i-th row of the $m \times n$ matrix A, and R_i is a $p_i \times n$ matrix. See [4–6] as informative references on SOCPs. A common re-expression of (8.1) is based on the fact that

$$\|R_i\, x\| = \max\{u_i^T R_i\, x : \|u_i\| \leq 1\}, \tag{8.2}$$

where u_i is a p_i-vector. The subsequent re-expression is

$$\left.\begin{array}{l} \max c^T x \\ \quad \text{such that} \\ \quad A_i x + u_i^T R_i \, x \le b_i, \quad \text{for all } u_i \text{ satisfying } \|u_i\| \le 1, \\ \qquad\qquad\qquad\qquad \text{for } i = 1, 2, \ldots, m, \\ \quad x \ge 0. \end{array}\right\} \tag{8.3}$$

Modeling uncertainty benefits from the inequality $\|u_i\| \le 1$ in (8.2), as the SOCP is interpreted as optimizing against all constraints of the form

$$(A_i + u_i^T R_i)\, x \le b_i, \text{ with } \|u_i\| \le 1,$$

where i indexes the constraint. This format includes the linear constraint $A_i x \le b_i$ with $u_i = 0$, and hence, this formulation imparts that we are optimizing against a collection of uncertainty surrounding the data of A_i.

The inequality in (8.2) can be tacitly replaced with the equality $\|u_i\| = 1$ because the maximum is always achieved on the boundary. This observation permits the reformulation,

$$\min\{z(u_1, u_2, \ldots, u_m) : \|u_i\| = 1 \text{ for } i = 1, 2, \ldots, m\}, \tag{8.4}$$

where

$$z(u_1, \ldots, u_m) = \max\{c^T x : x \ge 0, \; A_i x + u_i^T R_i x \le b_i, \text{ for } i = 1, 2, \ldots, m\}.$$

This second reformulation differs from the first with regard to its decision space. Models (8.1) and (8.3) share x as their common decision vector, and their argument maximums agree. Model (8.4) instead has $(u_1, \ldots, u_m)$ as its decision vector, and this model only agrees with models (8.1) and (8.3) in its optimal value. Problems (8.1) and (8.3) are convex and are commonly solved by software packages with an interior-point algorithm. Problem (8.4) is not convex, and it is thus in a different, and typically more difficult, problem class. However, any value of $z(u_1, \ldots, u_m)$ can be calculated by solving a linear program (LP), a fact that we promote as an advantage in some circumstances—especially those in which interior-point methods are less than performant and those for which an optimal value is more important than an optimal solution.

RAMP models motivate both circumstances. First, standard interior-point solvers have not proven trustworthy [19], and second, the most common computational task requires the calculation of the optimal value and not an optimal solution. In particular, FBA models, as well as their extensions like RAMP, are benchmarked on their ability to predict gene essentiality, which is determined by calculating the maximum growth rate with a gene knockout. A gene is essential if the maximum growth rate is sufficiently small. Most FBA models, including RAMP

with limited uncertainty, predict gene essentiality with at least 90% accuracy, see for example [21].

RAMP's computational success to date follows from the restrictive assumption that uncertainty be narrowed to the biomass equation, a source of uncertainty that we discuss more thoroughly in Sect. 8.5. Restricting uncertainty to the biomass equation means that each linear constraint of an FBA model has at most a single uncertain parameter, and subsequently, the SOCP of RAMP reduces to an LP. RAMP benefits computationally because the resulting LP solves reliably with a simplex method, providing a stable computational model. We note that the LPs, just like their corresponding SOCPs, do not lend themselves to interior-point algorithms, and hence, there is a computational preference for simplex based approaches [19]. That said, interior-point solutions have interpretive advantages should an interior-point algorithm solve the particular problem of interest [2, 3].

Restricting uncertainty to the biomass equation limits RAMP's scope, but fulfilling RAMP's promise as an SOCP does not mandate a substantial increase in uncertainty. Indeed, the vast majority of a metabolic model's data is based on standard stoichiometry and is thus perfectly determined. So little is truly uncertain, and all metabolic models known by the authors would only need an extension to two or less uncertain parameters per constraint. We address these extensions by developing an algorithm to calculate the optimal value of (8.1) by solving (8.4) under the restriction that each constraint has at most two uncertain parameters. We say that such problems have low-dimensional uncertainty. The new algorithm combines a nonlinear search to solve the nonconvex problem in (8.4) with a simplex algorithm to evaluate $z(u)$.

Restricting uncertainty to at most two parameters per constraint allows additional modeling assumptions. We can first assume that each R_i has at most two columns with nonzero elements, and we can secondly assume that each R_i has at most two rows, i.e. $p_i \leq 2$ for each i. Suppose to the contrary that $p_i > 2$. Then the maximum rank of R_i being two guarantees that we can row reduce R_i to R_i' with all but the top two rows being zero. Let $\hat{R}_i$ be the submatrix of R_i' that contains the top two rows of R'. Since the row spaces of R_i and $\hat{R}_i$ are the same, we can replace $u_i^T R_i$ with $\hat{u}_i^T \hat{R}_i$, where $\hat{u}_i$ is a two element vector, and hence, we can assume $p_i \leq 2$. If $p_i = 1$, then u is a scalar with the value of either 1 or -1, and in this case we can simply add both linear constraints and forego deciding u_i. Lastly, u_i is a two element unit vector if and only if

$$u_i = (\cos(\theta_i),\ \sin(\theta_i))^T$$

for a unique θ_i in $[0, 2\pi)$.

The RAMP formulation with at most two uncertain parameters per constraint prompts us to study the model,

$$\min\{z(\theta) : \theta \in [0, 2\pi)^q\}, \tag{8.5}$$

where

$$\begin{aligned} z(\theta) = \max\, & c^T x \\ & \text{such that} \\ & A_i x + (\cos(\theta_i), \sin(\theta_i)) R_i\, x \le a_i, \ \text{ for } i = 1, 2, \ldots, q, \\ & Bx \le b, \\ & x \ge 0. \end{aligned}$$

The $(m - q) \times n$ system $Bx \le b$ contains the constraints without uncertain parameters and those that have a single uncertain parameter, the latter of which have been re-expressed as two linear inequalities. Constraints with exactly two uncertain parameters are modeled by the system

$$A_i x + (\cos(\theta_i), \sin(\theta_i)) R_i\, x \le a_i, \ \text{ for } i = 1, 2, \ldots, q,$$

where each θ_i ranges over $[0, 2\pi)$. Each R_i is a $2 \times n$ matrix of rank 2, with the only nonzero columns aligning with the two uncertain parameters of A_i.

The optimal value of (8.5) is the same as the optimal value of the SOCP,

$$\max\{c^T x : x \ge 0,\ Bx \le b,\ A_i x + \|R_i x\| \le a_i, \ \text{for } i = 1, 2, \ldots q\},$$

but we again stress that the decision spaces are different. The SOCP seeks an optimal x, whereas the minimization of z seeks an optimal θ. We comment that the SOCP format naturally arises as a robust linear program, and any linear model may fit the restricted form in (8.5) if the sources of uncertainty are limited to two per constraint.

8.3 Algorithmic Development

Approximate solutions to a general SOCP like (8.1) are possible by generating a set of u_i vectors per constraint satisfying $\|u_i\| = 1$ and then formulating an approximate LP from (8.3). This tactic linearly approximates the boundaries of the ellipsoids

$$\{A_i + u_i^T R_i : \|u_i\| \le 1\}.$$

The number of constraints unfortunately grows exponentially in the dimension of the associated Lorentz cone even for crude approximations [7]. This conundrum is overcome in [7] by showing that a polyhedral approximation is possible with size, i.e. the number of variables and constraints, no greater than a constant multiple of the size of the SOCP multiplied by $\ln(1/\varepsilon)$, where ϵ is a tolerance parameter defining the approximation. This reduction of an SOCP to an approximate LP has

the advantage of approximately solving the SOCP with a linear solver, but it has two potential downsides with regard to RAMP. First, FBA models are typically fraught with redundancy, and the suggested approximating polyhedron adds additional redundancy and exacerbates this concern. Second, deciding if a maximum growth rate is zero is sensitive to a cutoff allowance that would need to account for the approximating parameter ε, with computational confidence being gained as ε decreases. However, the number of constraints increases as ε decreases, which again complicates a robust extension of an FBA model.

We promote a different solution procedure that iteratively maintains the number of constraints of an FBA model as it extends to its robust counterpart. Moreover, our solution technique technically reduces the number of variables from n to q, i.e. the number of variables lowers to the number of constraints with two uncertain parameters, which is a substantial reduction in RAMP models. The downside is that we solve a sequence of LPs whose optimal values tend to converge to the optimal value of the SOCP. So instead of modeling an approximate SOCP as a single LP to solve once with a guaranteed accuracy, we define a sequence of smaller LPs whose solutions are intended to converge to the optimal value of the SOCP.

The necessary and sufficient conditions of optimality for the general SOCP in (8.1) are:

$$A_i x + \|R_i x\| \leq b_i, \quad \text{for } i = 1, 2, \ldots m$$

$$c_j - \sum_{i=1}^{m} \lambda_i \, e_j^T \left(A_i^T + \left(\frac{R_i x}{\|R_i x\|} \right) R_i^T \right) = 0, \quad \text{for } j = 1, 2, \ldots n$$

$$\sum_{i=1}^{m} \lambda_i \, (b_i - \|R_i x\| - A_i x) = 0, \quad \text{and}$$

$$x, \ \lambda \geq 0,$$

where e_j is a vector of zeros except for a one in position j. The second condition is the requirement that the gradient of the Lagrangian be zero, and this equality indicates the relationship between an optimal x to problems (8.1) and (8.3) and the vector u in problem (8.4). In particular, these necessary and sufficient conditions equate with those of the LP in (8.3) if we set

$$u_i = \frac{R_i x}{\|R_i x\|}, \tag{8.6}$$

which subsequently implies that $\|R_i x\| = u_i^T R_i \, x$. This relationship provides an optimality test for a feasible x, say $\hat{x}$, of the SOCP. In particular, $\hat{x}$ is optimal for the SOCP if

$$\hat{x} \in \operatorname{argmax}\left\{c^T x : \left(A_i + \left(\frac{R_i \hat{x}}{\|R_i \hat{x}\|}\right)^T R_i\right) x \leq b_i,\ x \geq 0\right\}. \tag{8.7}$$

So feasible SOCP solutions can be verified as optimal solutions by solving an LP.

The optimality test in (8.7) suggests the possibility of solving LPs to generate solutions to the SOCP, but this intuition is only partially fulfilled by problem (8.4), which reduces to problem (8.5) in our restricted setting. The issue is that solving LPs does not necessarily provide a feasible solution to the SOCP. Suppose $\hat{u}$ solves (8.4) and $\hat{x}$ solves the LP,

$$\max\left\{c^T x : x \geq 0,\ A_i x + \hat{u}_i^T R_i x \leq b_i,\ \text{ for } i = 1, 2, \ldots, m\right\},$$

then we are not guaranteed that $\hat{x}$ is feasible for the SOCP even though the argument maximum of this LP contains an optimal solution to the SOCP. The resulting consequence is that the optimal value of the LP defined by $\hat{u}$ agrees with the optimal value of the SOCP, while an optimal solution $\hat{x}$ to the LP might not be feasible to the SOCP. We can thus calculate the optimal value of the SOCP by solving an appropriate LP, but we cannot guarantee that an optimal solution to the LP is feasible for the SOCP.

We use the relationship in (8.6) to initiate our algorithm, and the first calculations are:

Initialization Process

1. Solve the certain LP with each $u_i = (0, 0)^T$. Let x^0 be an optimal solution.
2. Calculate for each i the unique θ_i^0 in $[0, 2\pi)$ so that

$$\left(\cos\left(\theta_i^0\right),\ \sin\left(\theta_i^0\right)\right)^T = \frac{R_i x^0}{\|R_i x^0\|}.$$

3. Calculate $z(\theta^0)$ as an initial candidate to solve problem (8.5).

The goal being to minimize $z(\theta)$ suggests that we calculate $\nabla z(\theta)$, which we could then use in an algorithm like gradient descent or BFGS. Notice that

$$z(\theta) = \max\{c^T x : Ax \leq a - \delta a,\ Bx \leq b,\ x \geq 0\},$$

with

$$\delta a = \begin{pmatrix} (\cos(\theta_1),\ \sin(\theta_1))\, R_1 x^* \\ (\cos(\theta_2),\ \sin(\theta_2))\, R_2 x^* \\ \vdots \\ \left(\cos\left(\theta_q\right),\ \sin\left(\theta_q\right)\right) R_q x^* \end{pmatrix}, \tag{8.8}$$

where x^* is an optimal solution for θ, e.g. x^* could be x^0 for θ^0. From the chain rule we have that $\partial z/\partial\theta_i$ is

$$\frac{\partial}{\partial(a-\delta a)_i}\left(\max\{c^T x : Ax \le a-\delta a,\ Bx \le b,\ x \ge 0\}\right)\frac{\partial(a-\delta a)_i}{\theta_i}, \tag{8.9}$$

so long as both partials exist. Unfortunately, neither partial derivative is guaranteed to exists in the presence of degeneracy, although we can ensure the existence of directional derivatives even in this case.

Both partials have been studied in the area of sensitivity analysis, see for example [1, 12–16], and both have directional counterparts expressed in terms of optimization problems. The right-sided partial derivative of the second partial in (8.9) is

$$\left(\frac{\partial(a-\delta a)_i}{\theta_i}\right)_+ = \max_{x^*}\ (-\sin(\theta_i), \cos(\theta_i))\, R_i\, x^*,$$

where x^* ranges over the optimal set of the LP defining $z(\theta)$. The existence of this right-sided derivative can be ensured by establishing that the optimal set is bounded, which can subsequently be guaranteed by satisfying Slater's interiority condition. The value of the derivative can be computed by solving the stated LP. The left-sided partial is

$$\left(\frac{\partial(a-\delta a)_i}{\theta_i}\right)_- = \min_{x^*}\ (-\sin(\theta_i), \cos(\theta_i))\, R_i\, x^*,$$

where x^* again ranges over the optimal set of the LP defining $z(\theta)$. Notice that the partial derivative itself exists if the minimum and maximum values agree, which is assured if x^* is unique.

The left- and right-sided partial derivatives of the first partial in (8.9) are

$$\left(\frac{\partial}{\partial(a-\delta a)_i}\right)_+ \left(\max\{c^T x : Ax \le a-\delta a,\ Bx \le b,\ x \ge 0\}\right) \tag{8.10}$$

$$= \min\left\{\mu_i : A^T\mu + B^T\sigma \ge c, (a-\delta a)^T\mu + b^T\sigma = z(\theta), \mu \ge 0, \sigma \ge 0\right\}.$$

and

$$\left(\frac{\partial}{\partial(a-\delta a)_i}\right)_- \left(\max\{c^T x : Ax \le a-\delta a,\ Bx \le b,\ x \ge 0\}\right) \tag{8.11}$$

$$= \max\left\{\mu_i : A^T\mu + B^T\sigma \ge c, (a-\delta a)^T\mu + b^T\sigma = z(\theta), \mu \ge 0, \sigma \ge 0\right\}.$$

Solutions to these LPs are again guaranteed by satisfying an appropriate constraint qualification such a Slater's interiority condition. These directional derivatives are the same if and only if the dual multiplier μ_i for the constraint $A_i x \leq (a - \delta a)_i$ has a unique optimal value. We conclude that $\nabla z(\theta)$ exists if the LP defining $z(\theta)$ has a unique primal and dual solution. This result is stated in Theorem 8.1.

Theorem 8.1 *Define δa as in (8.8), and assume the LP*

$$z(\theta) = \max\{c^T x : Ax \leq a - \delta a,\ Bx \leq b, x \geq 0\}$$

has a unique primal and dual solution. Then,

$$\nabla z(\theta) = \begin{pmatrix} \mu_1^*(\cos(\theta_1), \sin(\theta_1) R_1 x^* \\ \mu_2^*(\cos(\theta_2), \sin(\theta_2) R_2 x^* \\ \vdots \\ \mu_q^*(\cos(\theta_q), \sin(\theta_q) R_q x^* \end{pmatrix},$$

where x^* is the unique solution to the LP defining $z(\theta)$ and μ^* is the unique vector of dual multipliers for the constraints $Ax \leq a - \delta a$.

FBA problems are highly degenerate, and making an assumption of uniqueness to ensure the existence of $\nabla z(\theta)$ is suspect. This fact prompts alternatives to the expression in Theorem 8.1. One apparent option is to solve the LPs defining the directional derivatives, which can then be used to construct a search direction along which $z(\theta)$ decreases. This scheme has a mathematical elegance that is difficult to realize numerically since the LPs require the computational identification of the optimal set defining $z(\theta)$. One common mathematical strategy is to add an equality constraint that holds the objective to its optimal value, which is the tactic used in (8.10) and (8.11). One problem with this tactic is that numerical round off can lead to an infeasible system. Tolerable inequality replacements also necessitate additional numerical considerations and can be difficult computationally. A second mathematical alternative is to describe the optimal set of the LP defining $z(\theta)$ by calculating the optimal partition, but this calculation can also be computationally troublesome.

We originally attempted to solve the LPs defining the partial directional derivatives of $z(\theta)$, but this mathematically elegant solution routinely proved problematic. Moreover, this scheme required solving four LPs for each θ_i to decide a search direction, adding further computational burden. A more stable and simplistic approach is the finite difference approximation,

$$\frac{\partial z}{\partial \theta_i}(\theta) \approx \frac{z(\theta + \varepsilon\, e_i) - z(\theta)}{\varepsilon},$$

where ε is some suitably small, positive value. This finite difference exists as long as both $z(\theta + \varepsilon\, e_i)$ and $z(\theta)$ exist, and this calculation only requires the additional solution of the single LP defining $z(\theta + \varepsilon\, e_i)$. This is a 75% reduction in the number of LPs being solved for the cases in which all the LPs associated with the partial derivatives stem from degenerate problems. Possible inaccuracies are from the approximation itself, from the potential loss of the existence of the partial derivative, and from the numerical round off of the LP solver. Degeneracy occurs on a set of measure zero, and hence, we are unlikely to realize the mathematical loss of the derivative computationally, especially within an approximating computational scheme. This suggests that the finite difference approximation will be accurate within our algorithmic framework for sufficiently small ε. We have indeed found this to be the case and have thus selected the finite difference approximation for our forthcoming numerical work. We denote the approximate gradient as $\nabla_{\approx} z(\theta)$.

Our overriding algorithmic framework to minimize $z(\theta)$ follows a standard nonlinear approach that seeks to move from iteration k to $k+1$ so that

$$\theta^{k+1} \in \operatorname{argmin}\{z(\theta^k + \alpha d^k) : \alpha \geq 0\}.$$

We test two search directions, those being gradient descent and BFGS. Gradient descent uses

$$d^k = -\nabla_{\approx} z(\theta^k),$$

and BFGS solves

$$H_{k+1} d^{k+1} = -\nabla_{\approx} z(\theta^k),$$

with

$$H_{k+1} = \left(H_k + \frac{\Delta D z_k (\Delta D z_k)^T}{(\Delta D z_k)^T \Delta\theta_k} - \frac{H_k \Delta\theta_k (\Delta\theta_k)^T H_k}{(\Delta\theta_k)^T H_k \Delta\theta_k} \right), \tag{8.12}$$

$$\Delta D z_k = \nabla_{\approx} z(\theta_k) - \nabla_{\approx} z(\theta_{k-1}), \quad \text{and}$$

$$\Delta\theta_k = \theta_k - \theta_{k-1}.$$

We further test the common inverse version of BFGS to bypass the need to solve $H_{k+1} d^{k+1} = -\nabla_{\approx} z(\theta^k)$ per iteration. In this case we set

$$H_{k+1}^{-1} = H_k^{-1} + \left(1 + \frac{\Delta D z_k^T H_k^{-1} \Delta D z_k}{\Delta D z_k^T \Delta x_k} \right) \frac{\Delta x_k \Delta x_k^T}{\Delta x_k^T \Delta D z_k}$$

$$- \frac{\Delta x_k \Delta D z_k^T H_k^{-1} + H_k^{-1} \Delta D z_k \Delta x_k^T}{\Delta x_k^T \Delta D z_k}, \tag{8.13}$$

and the search direction is

$$d^{k+1} = -H_{k+1}^{-1} \nabla_{\approx} z_k(\theta^k).$$

All BFGS algorithms initiate with $H^0 = I$, making the first search directions of gradient descent and BFGS agree.

The line search requires special consideration since the standard convergence criteria of $\|\nabla_{\approx} z(\theta)\| \leq \varepsilon$ can be impossible to achieve at the optimal solution, a situation illustrated by the first example of the next section. We promote a line search that allows α to range over a predefined interval, say $[0, \hat{\alpha}]$. We use a line search that accepts the step size of $\hat{\alpha}$ if

$$\nabla_{\approx} z(\theta^k + \hat{\alpha} d^k)^T d^k < 0 \text{ and } z(\theta^k + \hat{\alpha} d^k) < z(\theta^k);$$

that is, we accept the maximum step if we predict continued improvement along d^k beyond $\hat{\alpha}$. There is no mathematical guarantee that $z(\theta^k + \alpha d^k)$ decreases over the interval $[0, \hat{\alpha}]$, although this is the case that motivates the rule. If $\nabla_{\approx} z(\theta^k + \hat{\alpha} d^k)^T d^k > 0$, then we employ a binary search for an α in $[0, \hat{\alpha}]$ at which $\nabla_{\approx} z(\theta^k + \alpha d^k)^T d^k$ changes sign. There is no guarantee of continuity, and it is consequently unrealistic to search for an α in $[0, \hat{\alpha}]$ satisfying $\nabla_{\approx} z(\theta^k + \alpha d^k)^T d^k = 0$. If $z(\theta^k + \hat{\alpha} d^k) \geq z(\theta^k)$, then we decrease $\hat{\alpha}$ and repeat the decision process. Pseudocode for our calculation scheme is listed in Algorithm 1.

The lack of rigorous guarantees during the search is disquieting. Most of the mathematical concerns are due to potential changes in the rank of $A + \delta a$ as θ adjusts, although the loss of convexity is also problematic. That said, nonlinear schemes like those that we suggest are often motivated by reasonable, albeit not verifiable, perspectives, and an algorithm's merit lies in its efficacy and not its mathematical comfort. The computational advantage of our algorithmic development is that both $z(\theta)$ and $\nabla_{\approx} z(\theta)$ can be computed by solving linear programs of the same size as the certain LP from which the SOCP is generated. In particular, any algorithm may be used to solve the LPs, which means that we can leverage the capabilities of the simplex method to solve FBA and RAMP problems.

8.4 Illustrative Examples

We investigate a couple of simple problems to motivate the practicality of our computational scheme. These examples are purposefully small to support geometric reassurance and algebraic certainty with regard to optimality.

Algorithm 1 Pseudocode to solve an SOCP model with low-dimensional uncertainty

0. **Initialize:** Calculate an optimal solution, x^0, to the certain LP. Set

$$\theta_i^0 = \begin{cases} \tan^{-1}\left([R_i x^0]_2/[R_i x^0]_1\right), & [R_i x^0]_1 > 0 \\ \tan^{-1}\left([R_i x^0]_2/[R_i x^0]_1\right) + \pi, & [R_i x^0]_2 < 0 \\ \pi/2, & [R_i x^0]_1 = 0 \text{ and } [R_i x^0]_2 > 0 \\ 3\pi/2, & [R_i x^0]_1 = 1 \text{ and } [R_i x^0]_2 < 0. \end{cases}$$

Calculate $z(\theta^0)$, and set $k = 0$.

1. **Calculate Search Direction:** Calculate $\nabla_{\approx} z(\theta^k)$ and then calculate d^k to satisfy $H_k d^k = -\nabla_{\approx} z(\theta^k)$. The matrix H_k is the identity in the method of steepest descent, and H_k is the expression in (8.12) for the BFGS method. If the BFGS algorithm is instead calculating H_k^{-1}, then set $d_k = -H_k^{-1}\nabla_{\approx} z(\theta^k)$ after updating the inverse according to (8.13).
2. **Line Search:** Search for α^k in $[0, \hat{\alpha}]$ so that

$$\theta^k + \alpha^k d^k \in \text{argmin}\{z(\theta^k + \alpha\, d^k) : 0 \le \alpha \le \hat{\alpha}\}.$$

Case 1: Set $\alpha^k = \hat{\alpha}$ if $z(\theta^k + \hat{\alpha} d^k) < z(\theta^k)$ and $\nabla_{\approx} z(\theta^k + \hat{\alpha} d^k)^T d^k < 0$.
Case 2: Reduce $\hat{\alpha}$ and repeat the line search if $z(\theta^k + \hat{\alpha}\, d^k) \ge z(\theta^k)$.
Case 3: Conduct a binary search to locate a sign change in

$$\nabla_{\approx} z(\theta^k + \alpha\, d^k)^T d^k$$

if $\nabla_{\approx} z(\theta^k + \hat{\alpha} d^k)^T d^k \ge 0$. Set α^k to the value at which the sign change occurs, including the case that $\alpha^k = \hat{\alpha}$ if $\nabla_{\approx} z(\theta^k + \hat{\alpha} d^k)^T d^k = 0$.

3. **Update:** Set $\theta^{k+1} = \theta^k + \alpha^k d^k$ and calculate $z(\theta^{k+1})$ and $\nabla_{\approx} z(\theta^{k+1})$.
4. **Check for Termination:** Terminate the algorithm if α^k is decided by the binary search in the third case of the line search or if $\nabla_{\approx} z(\theta^{k+1})$ is sufficiently small. Otherwise return to step 1) with $k = k + 1$.

8.4.1 Example 1

We solve the parameterized SOCP,

$$\max\{x_1 + x_2 : x_1 + x_2 + \sigma\|(x_1, x_2)\| \le 1,\ x_1 \ge 0,\ x_2 \ge 0\},$$

for which

$$A = [1, 1],\quad a = 1,\quad B = 0,\quad b = 0,\quad \text{and}\quad R = \sigma I.$$

The amount of the uncertainty is controlled by the nonnegative parameter σ, and the problem is certain if $\sigma = 0$. The elements of the coefficient matrix are otherwise uncertain and range within the ellipsoid,

$$\{[1, 1] + \sigma[u_1, u_2] : \|(u_1, u_2)\| \leq 1\}.$$

The geometry of the problem is depicted in Fig. 8.1. A straightforward calculation shows that the unique optimal solution and the optimal value are, respectively,

$$x^* = \frac{1}{2 + \sigma\sqrt{2}}\,(1, 1)^T \text{ and } z^* = \frac{2}{2 + \sigma\sqrt{2}}.$$

Problem (8.5) re-expresses the SOCP relative to the single variable θ, with the result being

$$\min\{z(\theta) : 0 \leq \theta < 2\pi\},$$

where

$$z(\theta) = \max\{x_1 + x_2 : (1 + \sigma\cos(\theta))x_1 + (1 + \sigma\sin(\theta))x_2 \leq 1,\ x_1 \geq 0,\ x_2 \geq 0\}.$$

Figure 8.2 illustrates a few of the LPs defining $z(\theta)$, and Figures 8.3, 8.4, and 8.5 depict $z(\theta)$ for two different values of σ. We plot the value of $z(\theta)$ over the unit circle in Figs. 8.3 and 8.4 to highlight its periodic nature. The minimum value of $z(\theta)$ is $z(\pi/4) = 2/(2 + \sigma\sqrt{2})$.

The geometry of $z(\theta)$ deserves inspection, especially near its minimum, see Figs. 8.3 and 8.5. Notice that $z(\theta)$ is not differentiable at its minimum for either value of σ although both its left- and right-derivatives exist. The function $z(\theta)$ is neither convex nor continuous although it does have a unique minimum. The discontinuities with $\sigma = 1$ occur because the LP is unbounded for $\theta = \pi$ and $\theta = 3\pi/2$. A couple of observations are:

1. small amounts of uncertainty might have computational advantages because problem dynamics can be more varied as data is less certain, and
2. a reasonable starting solution could be paramount.

The first observation is mathematically justified by the fact that $\text{rank}(M) \leq \text{rank}(M + \Delta M)$ so long as $\|\Delta M\|$ is sufficiently small, and moreover, $\text{rank}(M + \beta\Delta M)$ is constant for small, positive β. We have for the example that

$$\text{rank}([1 + \sigma\cos(\hat{\theta}),\ 1 + \sigma\sin(\hat{\theta})]) \leq \text{rank}([1 + \sigma\cos(\theta),\ 1 + \sigma\sin(\theta)])$$

if θ is sufficiently close to $\hat{\theta}$, and we further have that rank is constant if θ is in a sufficiently small neighborhood of the form $(\hat{\theta}, \hat{\theta} + \delta)$ or $(\hat{\theta} - \delta, \hat{\theta})$, where $\delta > 0$ is sufficiently small. This rank argument extends to show that the optimal partition, and subsequently the dimension of the optimal set and its algebraic description, does not change over these neighborhoods, see [15].

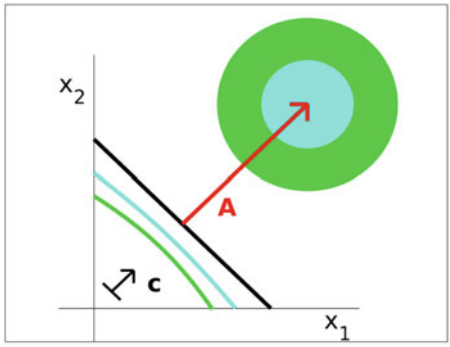


Fig. 8.1 Smaller and larger uncertainty sets with their corresponding SOCP constraints in cyan and green, respectively

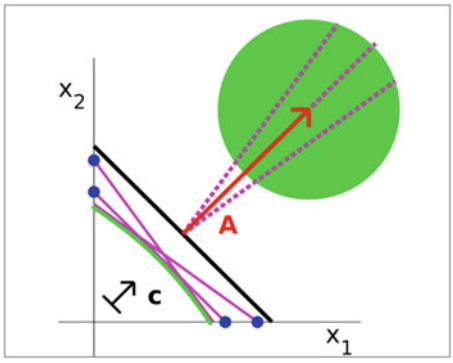


Fig. 8.2 LPs defining $z(\theta)$ for three different values of θ, with basic optimal solutions as blue dots

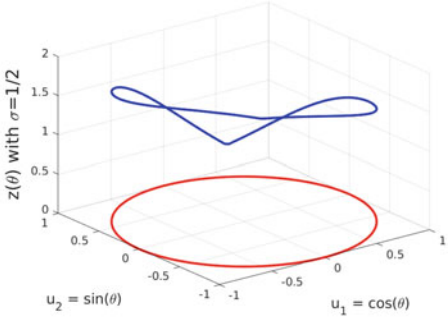


Fig. 8.3 $z(\theta)$ for $0 \leq \theta < 2\pi$ with $\sigma = 1/2$

The calculation of $z(\theta)$ can be accomplished with any LP algorithm, which can be a computational advantage since different problems lend themselves to different solvers. The argument maximum of the LP defining $z(\theta)$ with $\sigma = 1$ is

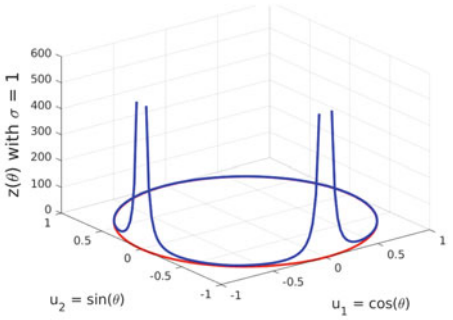


Fig. 8.4 $z(\theta)$ for $0 \leq \theta < 2\pi$ with $\sigma = 1$

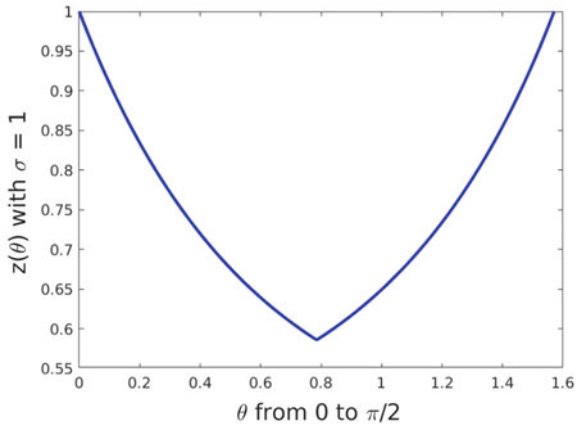


Fig. 8.5 $z(\theta)$ for $0 \leq \theta < \pi/2$ with $\sigma = 1$

$$\operatorname{argmax}\{x_1 + x_2 : (1 + \cos(\theta))x_1 + (1 + \sin(\theta))x_2 \leq 1,\ x_1 \geq 0,\ x_2 \geq 0\}$$

$$= \begin{cases} \left\{(1/(1+\cos(\theta)), 0)^T\right\}, & \cos(\theta) < \sin(\theta) \\ \left\{(0, 1/(1+\sin(\theta)))^T\right\}, & \cos(\theta) > \sin(\theta) \\ \left\{(1-\beta)\,(1/\gamma, 0)^T + \beta(0, 1/\gamma)^T : 0 \leq \beta \leq 1\right\}, & \cos(\theta) = \sin(\theta) \\ & \quad = \gamma - 1. \end{cases}$$

The argument maximum does not intersect the feasible region of the SOCP except for the optimal value of $\theta^* = \pi/4$. So an optimal x for any $z(\theta)$ other than $z(\pi/4)$ is infeasible for the SOCP. Since our algorithmic goal is to construct a sequence θ^k so that $z(\theta^k) \to z(\pi/4)$, we see that our calculation scheme is, in some manner, an infeasible algorithm from the perspective of the SOCP.

The argument maximum with $\theta = \pi/4$ intersects the feasible region of the SOCP at the unique optimal solution x^*. However, x^* is not a basic optimal solution, and it is thus impossible to evaluate $z(\pi/4)$ with a simplex method and be left with a feasible solution to the SOCP. This observation sharpens the fact that we are

calculating the optimal value of the SOCP by minimizing z as a function of θ and not x. We comment that x^* is the analytic center of the argument maximum for $\theta = \pi/4$ in this specific example, and hence, a path-following interior-point algorithm for the LP would (theoretically) converge to a feasible solution of the SOCP. The fact that an interior-point algorithm would (theoretically) converge to an optimal solution of the SOCP is due to this example's simple symmetry and its lack of redundancy, and interior-point algorithms are not generally guaranteed to provide feasible SOCP solutions.

The infeasible nature of our algorithm can give x solutions, i.e. those that define the minimum value of $z(\theta)$, that are arbitrarily distant from the SOCP's argument maximum. If we let the right-hand side of this example be a instead of the fixed value 1, then the distance between an optimal basic solution of the LP defining $z(\theta/4)$ and the unique SOCP solution is

$$\frac{a}{\sqrt{2}+\sigma}.$$

We conclude that a simplex based routine can give a solution of any distance from the unique SOCP solution depending on a and σ. Notice that the distance in this example diminishes as σ increases, which follows because the feasible region collapses onto the origin as uncertainty grows.

Some emblematic numerical outcomes for various combinations of σ and θ^0 are listed in Table 8.1. All values of $z(\theta)$ are calculated with the dual-simplex algorithm in MATLAB's optimization toolbox, and the finite difference approximation uses $\varepsilon = 10^{-6}$. We set $\hat{\alpha} = 1$ in all cases, and we forego the initialization scheme because it terminates with the optimal value of $\theta^* = \pi/4$.

The geometry in Fig. 8.3 with $\sigma = 1/2$ would seem to better lend itself to our computational framework than would the geometry in Fig. 8.4, but this is not the case. One difference is that we fail to convergence to an optimal solution with $\sigma = 1/2$ if we initiate any of the algorithms within the interval $[\pi, 3\pi/2]$. Indeed, both versions of BFGS extend this interval to the left and have wider ranges of false convergence. The geometry in Fig. 8.4 with $\sigma = 1$ exhibits no such computational concern by comparison. In particular, all algorithms converge to an optimal solution with $\sigma = 1$ so long as θ^0 is not one of the discontinuities. We illustrate this curiosity by setting $\sigma = 1$ and initializing the search with $\theta^0 = 5\pi/4$, which is a local minimum trapped between the unbounded LPs with $\theta = \pi$ and $\theta = 3\pi/2$. The favorable outcome of all algorithms in this case follows because the forward difference approximation results in $\nabla_{\approx} z(5\pi/4) = -8.2427$, and each algorithm escapes what appears to be a local catchment region of the local minimum as it accepts the full step. The innate periodicity of searching over the unit circle then leads to optimal convergence, albeit with the terminating value of θ being outside the standard reference of $0 \leq \theta < 2\pi$. We remind that none of the basic optimal solutions of the LPs are feasible for the SOCP, but nonetheless, the algorithm converges to the correct optimal value over a wide range of settings.

Table 8.1 Representative numerical outcomes for Example 1

σ	Algorithm	θ^0	θ^*	Opt. Val.	Iter.	Time (s)
1	Grad. desc.	0	0.7854	0.5858	2	0.53
	BFGS	0	0.7854	0.5858	2	0.55
	BFGS inv.	0	0.7854	0.5858	2	0.53
1/2	Grad. desc.	0	0.7854	0.7388	3	0.55
	BFGS	0	0.7854	0.7388	3	0.59
	BFGS inv.	0	0.7854	0.7388	3	0.59
1	Grad. desc.	$5\pi/4$	-5.4978	0.5858	3	0.62
	BFGS	$5\pi/4$	-5.4978	0.5858	3	0.61
	BFGS inv.	$5\pi/4$	-5.4978	0.5858	3	0.61

8.4.2 Example 2

The second example is

$$\max\{x_1 + x_2 : 2x_1 + x_2 - \|x\| \leq 2,\ x_1 + 2x_2 - \|x\| \leq 2,\ x_1 \geq 0,\ x_2 \geq 0\},$$

for which

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix},\ a = \begin{pmatrix} 2 \\ 2 \end{pmatrix},\ B = 0,\ b = 0,\ \text{and}\ R_1 = R_2 = I.$$

This problem has two constraints with two uncertain parameters instead of the sole constraint of the first example. In this case the function $z(\theta)$ is

$$\begin{aligned} z(\theta) = \max\{&x_1 + x_2 : (2 + \cos(\theta_1))x_1 + (1 + \sin(\theta_1))x_2 \leq 2, \\ &(1 + \cos(\theta_2))x_1 + (2 + \sin(\theta_2))x_2 \leq 2,\ x_1 \geq 0,\ x_2 \geq 0\}, \end{aligned}$$

and a straightforward calculation shows that the minimizer of z is

$$\theta^* = (\pi/4,\ \pi/4)^T \ \text{ with } \ z^* = z(\theta^*) = \frac{4}{3 + \sqrt{2}} \approx 0.9062.$$

The (unique) optimal solution to the LP for θ^* is

$$x^* = \left(\frac{2}{3 + \sqrt{2}},\ \frac{2}{3 + \sqrt{2}}\right)^T,$$

and unlike the first example, this LP solution is feasible and (uniquely) optimal for the SOCP. Figure 8.6 illustrates the geometry of the certain LP and its relationship to the SOCP, and Figs. 8.7 and 8.8 depict the landscape of z.

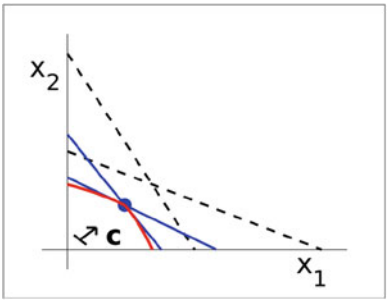


Fig. 8.6 The certain LP has dashed constraints, the SOCP has red, and the LP for the optimal $z(\theta)$ has blue. The dot is the unique optimal solution to the SOCP

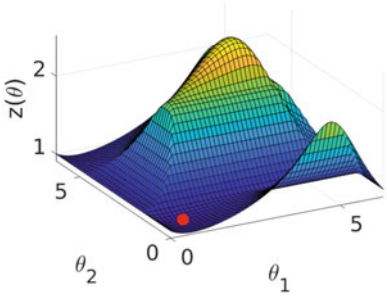


Fig. 8.7 The surface of $z(\theta)$ for $\theta \in [0, 2\pi]^2$. The red dot is the minimum

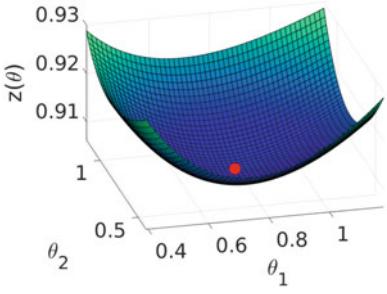


Fig. 8.8 The surface of $z(\theta)$ for $\theta \in [\pi/8, 3\pi/8]^2$. The red dot is the minimum

The geometry of $z(\theta)$ shows that it is neither convex nor differentiable, although it is continuous. The surface is convex near the optimal solution, and a technique such as BFGS should have favorable convergence properties if initiated sufficiently close to the optimal solution. The geometry also adumbrates wide applicability of gradient descent with $-\nabla_{\approx} z(\theta)$, as all such approximations would appear to converge to the optimal solution. BFGS lacks this trust because it might converge to the maximum of z as it seeks to satisfy a first order condition, or our adaption thereof, if initiated near the maximum. The surface is periodic, and the depicted geometry replicates itself along each axial direction. We note that the function is not differentiable for $\theta_1 = 0$ due to primal degeneracy in Theorem 8.1.

The initialization process terminates with the optimal θ of $(\pi/4, \pi/4)^T$ similar to the first example, and we again start with different θ^0 values to illustrate our algorithm's efficacy. We compute $z(\theta)$ with the dual-simplex algorithm in

MATLAB, and we use $\varepsilon = 10^{-6}$ in the calculation of $\nabla_{\approx} z(\theta)$. We initiate each algorithm over a uniform 400 point grid in $[0, 2\pi)^2$ to experimentally assess convergence properties. Statistics of this computational venture are in Table 8.2. Gradient descent proves the most trustworthy although it requires about twice the time as either BFGS or BFGS inverse. Gradient descent also requires a tenfold increase in the number of iterations over either version of the BFGS, but each iteration requires less computation. MATLAB warned on three occasions that the system defining the search direction in BFGS was nearly singular.

Representative results from the computational test are in Table 8.3. The three algorithms converge to the optimal solution and perform similar to their statistical averages if initiated at $(0, 0)^T$, a point at which $z(\theta)$ is not differentiable. The benefit of both BFGS algorithms if the starting point is near the optimal solution is highlighted with $\theta^0 = (\pi/3, \pi/3)^T$. All algorithms converge in this case, but the BFGS algorithms significantly outperform gradient descent. Initializing with $(3\pi/2, 3\pi/2)^T$ epitomizes how the algorithms can fail. Both BFGS algorithms converge to the maximum value in this case, whereas gradient descent terminates with a near optimal solution. Gradient descent's near convergence follows because it migrates from near the top of $z(\theta)$ toward to the "valley" along $\theta = 2\pi$, which is sufficiently flat to cease further progress. Gradient descent converges to a value of 1.0001 or less 159 of the 163 failures, meaning that it converges to a near optimum of 1.0001 or less in 396 of the 400 trials. This small example promotes

Table 8.2 Gross statistical results for each algorithm tested over a uniform, 400 point grid over $[0, 2\pi)$

Algorithm	Number correct	Min/max/mean time (s)	Min/max/min itrs.
Grad. desc.	237/400 (59.25%)	0.05/4.51/2.11	0/125/59.31
BFGS	166/400 (41.50%)	0.01/3.87/1.06	0/13/5.93
BFGS inv.	164/400 (41.00%)	0.01/3.27/1.05	0/13/5.90

Table 8.3 Representative numerical outcomes for Example 2

θ^0	Algorithm	θ^*	Opt. val.	Iter.	Time (s)
$(0, 0)^T$	Grad. desc.	$(0.7854, 0.7854)^T$	0.9062	85	2.61
	BFGS	$(0.7854, 0.7854)^T$	0.9062	11	1.60
	BFGS inv.	$(0.7854, 0.7854)^T$	0.9062	11	1.59
$(\pi/3, \pi/3)$	Grad. desc.	$(0.7854, 0.7854)^T$	0.9062	71	2.25
	BFGS	$(0.7854, 0.7854)^T$	0.9062	10	0.38
	BFGS inv.	$(0.7854, 0.7854)^T$	0.9062	10	0.36
$(3\pi/2, 3\pi/2)$	Grad. desc.	$(6.2832, 4.7124)^T$	**1.0000**	14	0.73
	BFGS	$(4.7124, 4.7124)^T$	**2.0000**	2	0.57
	BFGS inv.	$(4.7124, 4.7124)^T$	**2.0000**	2	0.50

Results in blue are suboptimal

a rule-of-thumb, which is to use BFGS if the starting solutions can be guaranteed to be near the optimal solution, but otherwise, the extra time of gradient descent is likely worthwhile.

8.5 RAMP Studies

We now turn to the biological problems motivating our algorithmic development. A metabolic model is a list of biochemical reactions that describe an organism's cellular metabolism. These reactions are subsequently represented by a system of ordinary differential equations through the principle of mass action, with steady state solutions being algebraically defined by a corresponding linear system of equations. The rows of the system express the rates at which metabolic concentrations change, and the columns contain chemical reactions.

We consider a simple example to illustrate the progression from a collection of biochemical equations to a linear system. The following three equations describe relationships among metabolites A, B, C, and D,

$$2A + B \xrightarrow{k_+^1} C + D, \quad A + C + D \xrightarrow{k_+^2} 3B, \quad \text{and} \quad C + B \xrightarrow{k_+^3} 2A.$$

The principle of mass action asserts that the reaction rates k_+^1, k_+^2, and k_+^3 are defined in terms of metabolic concentrations, which are denoted by brackets. So $[A]$ is the concentration of A, and $[D]$ is the concentration of D. The resulting system of differential equations is

$$\begin{aligned}
\tfrac{d[A]}{dt} &= 2\big(k_+^3[C][B]\big) - 2\big(k_+^1[A]^2[B]\big) - \big(k_+^2[A][C][D]\big) &&= 2v_3 - 2v_1 - v_2\\
\tfrac{d[B]}{dt} &= 3\big(k_+^2[A][C][D]\big) - \big(k_+^1[A]^2[B]\big) - \big(k_+^3[C][B]\big) &&= 3v_2 - v_1 - v_3\\
\tfrac{d[C]}{dt} &= \big(k_+^1[A]^2[B]\big) - \big(k_+^2[A][C][D]\big) - \big(k_+^3[C][B]\big) &&= v_1 - v_2 - v_3\\
\tfrac{d[D]}{dt} &= \big(k_+^1[A]^2[B]\big) - \big(k_+^2[A][C][D]\big) &&= v_1 - v_2.
\end{aligned}$$

The colored parenthetical groupings, which are products of reaction rates and concentrations, define fluxes represented by v_1, v_2, and v_3. These equations define a linear relationship between the fluxes and the rates at which metabolic concentrations change, which for this example is

$$\frac{d}{dt}\begin{pmatrix}[A]\\[B]\\[C]\\[D]\end{pmatrix} = \begin{bmatrix}-2 & -1 & 2\\ -1 & 3 & -1\\ 1 & -1 & -1\\ 1 & -1 & 0\end{bmatrix}\begin{pmatrix}v_1\\v_2\\v_3\end{pmatrix} = Sv.$$

The matrix S is called the stoichiometric matrix.

Reactions are not generally unidirectional, and it is common for reactions to have different forward and backward rates. Full metabolic systems commonly have thousands of reactions and hundreds of metabolites. For instance, we solve the

iJO1366 model of E. coli with 2583 fluxes and 1805 metabolites [21], see also the BiGG repository of FBA models [17]. We assume steady state by requiring $Sv = 0$, an equation that does not uniquely define the flux state since the FBA system is underdetermined in real models.

The most common FBA paradigm adds an empirical reaction that creates biomass as one of its products, and hence, a faux biochemical process describes the generation of biomass in terms of other metabolites. Let v_g be the flux of this growth reaction. We can then calculate the maximum growth rate over all possible steady state solutions by solving

$$\max\{v_g : Sv = 0,\ L \leq v \leq U\},$$

where L and U bound the fluxes to ensure directionality, to limit nutritional resources, and to account for non-metabolic processes like cellular repair.

Stoichiometric coefficients are innately integer, and the preponderance of data comprising the matrix S is consequently unambiguous. However, the growth equation is not standard stoichiometry, and its coefficients are uncertain. The default growth equation of the iJO1366 model incorporates 72 metabolites and has coefficients ranging in absolute value from 2×10^{-6} to 54.1248. This seven orders of magnitude difference draws suspicion and raises an interest in studying a model's dependence on small adjustments. Two other sources of uncertainty are the environmental bounds and the loss of adenosine triphosphate (ATP) to non-metabolic processes. These sources of uncertainty correspond with variable bounds that can be remodeled as uncertainty in matrix coefficients [4, 6]. We note that most applications of FBA assume a limiting carbon source and an unlimited amount of other nutritional elements.

RAMP is an FBA adaptation that permits uncertainty and provides a stochastic interpretation of FBA [19]. The model is

$$\max\{v_g : -M_i + \|R_i v\| \leq S_i v \leq M_i - \|R_i v\|,\ \forall i,\ L \leq v \leq U\},$$

where S_i is the i-th row of S, M_i sets a maximum deviation from the steady state assumption, and R_i determines the structure of uncertainty. Only one of the 72 metabolites in the growth equation, that being kdo2lipid4, has a transport reaction through the cellular membrane, and hence, this RAMP model has a single row with two uncertain parameters—one in the growth column and one in the transport column, the latter of which permits the metabolite to enter the cell from the environment. These two uncertain parameters are in two inequalities, one for the upper bound and one for the lower bound, making the RAMP model similar to the second example of the previous section. A third reaction can produce the kdo2lipid4 metabolite, but the coefficients of this reaction are certain. Solving the default model draws no kdo2lipid4 from the environment and instead creates the needed metabolite from this third reaction. The upper and lower bounds for the kdo2lipid4 transport reaction are 0 and 1000, with the transport coefficient being 1. Adjusting the coefficient equates to adjusting these bounds. The coefficient in the

growth equation is -0.0195, which means that a unit of biomass requires 0.0195 units of kdo2lipid4.

We vet our algorithm against two models, both of which assume certain parameters except for the two suspicious coefficients of the kdo2lipid4 metabolite, i.e. those coefficients corresponding with the metabolite's transport into the cell and with its consumption during the formation of biomass. We assume in both cases that $M_i = 0.01$, and hence,

$$-0.01 + \|R_i v\| \leq S_i v \leq 0.01 - \|R_i v\|,$$

which assures that

$$\left| \frac{d[\text{kdo2lipid4}]}{dt} \right| \leq 0.01.$$

The submatrix of R_i corresponding with the two uncertain parameters has the form

$$\begin{bmatrix} 0.02 & \pm 1 \\ \pm 1 & 0.02 \end{bmatrix}.$$

A standard probabilistic interpretation of this information follows if we assume the coefficients to be standard normal variables. In this case we are assuming:

- the rate of kdo2lipid4 transport into the cell is $\mathcal{N}(1, 0.02)$,
- the rate of kdo2lipid4 consumption by growth is $\mathcal{N}(-0.0195, 0.02)$, and
- the covariance of the random variables is ± 1.

The two constraints have subsequent probabilistic interpretations if we accept the aforementioned stochastic assumption.

We are not advocating these models as auspicious biological paragons but are instead using them as reasonable computational prototypes that are commensurate with the problem's data. The only difference between the two models is that one assumes positive off-diagonals and the other assumes negative off-diagonals. The computational differences are significant as we discuss below.

The landscapes of $z(\theta)$, of which the minimum values are the maximum growth rates of the SOCPs, are shown in Figs. 8.9 and 8.10 for the two models. The maximum growth rate decreases from 0.9824 without uncertainty to 0.00999 with positive off-diagonals and 0.01001 with negative off-diagonals. The case with negative off-diagonals is significantly more difficult to solve with regard to computational time. The positive case solves to optimality in about 0.18 s with either version of BFGS or with gradient descent, but the negative case requires about 26 s with either BFGS algorithm or about 18 s with gradient descent. So the negative diagonal case has at least a one-hundred-fold increase in computational time. All algorithms only require four or less iterations after initiation with the process from Sect. 8.3. The approximate gradient calculation uses $\varepsilon = 10^{-6}$.

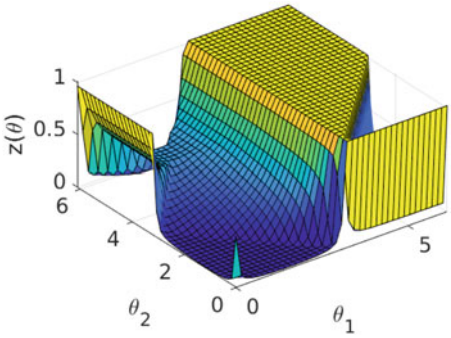


Fig. 8.9 The landscape of $z(\theta)$ over a $[0, 2\pi)^2$ for the positive off-diagonal case

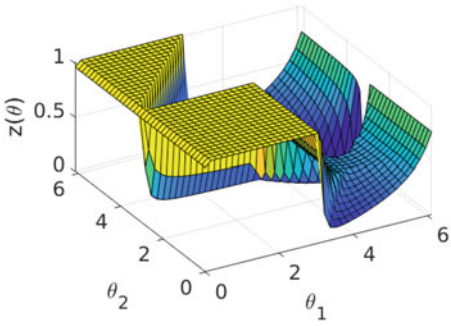


Fig. 8.10 The landscape of $z(\theta)$ over a $[0, 2\pi)^2$ for the negative off-diagonal case

All LPs solve with MATLAB's dual-simplex algorithm, but as the first example of Sect. 8.4 shows, the terminal flux state need not be feasible. However, all algorithms terminate with numerically viable flux vectors for both models, with the maximum deviation from feasibility for the SOCP being on the order of 10^{-17} for the model with positive off-diagonals and 10^{-5} for the model with negative off-diagonals. MATLAB's interior-point algorithm fails to solve the LPs required to evaluate $z(\theta)$, including the original FBA model, a fact that substantiates the earlier work in [19] even though MATALB's optimization algorithms have since been updated.

We conclude by predicting gene essentiality. Gene knockouts force resulting collections of fluxes to be zero, and a gene is essential if its knockout results in an optimal growth rate of zero. The model with negative off-diagonals requires over 3.5 h to complete the knockouts, whereas the model with positive off-diagonals only requires a couple of minutes—again highlighting the numerical difficulty of the model with negative off-diagonals. We use the gradient descent algorithm for all knockouts, and all solutions converge to optimality. Results of the comparison are in Table 8.4.

The gene knockout predictions are similar to those in [19], with RAMP tending to improve prediction of non-essentiality and degrade prediction of essentiality. FBA's accuracy is 91.14%, which fell, respectively, to 90.2 and 90.27% for the models with positive and negative off-diagonals.

Table 8.4 Gene essentiality prediction for FBA and RAMP

Computational	Experimental: Essential	Experimental: Nonessential
Essential	True Positive FBA | 171 RAMP (pos. off-diag.) | **153** RAMP (neg. off-diag.) | **154**	False Positive FBA | 44 RAMP (pos. off-diag.) | **39** RAMP (neg. off-diag.) | **39**
Nonessential	False Negative FBA | 77 RAMP (pos. off-diag.) | **95** RAMP (neg. off-diag.) | **94**	True Negative FBA | 1074 RAMP (pos. off-diag.) | **1080** RAMP (neg. off-diag.) | **1080**

8.6 Conclusions

Our nonlinear algorithms are performant for solving the example SOCPs associated with RAMP extensions of FBA. Such extensions have limited uncertainty and lend themselves to low-dimensional investigations that rely on accurate solutions to LPs. The advantage of the solution technique herein is that it uses a simplex algorithm to iteratively calculate the optimal value of an SOCP, providing a stable calculation scheme for cases such as RAMP in which native interior-point solvers regularly fail. Our nonlinear algorithms only require a few iterations to identify the optimal solution if initialized as in Sect. 8.3, making them reasonably efficient solution schemes. Moreover, terminal flux states are computationally feasible to the robust model even though feasibility is not mathematically ensured, and hence, the terminal flux vector solves the RAMP model in our experiments.

Several avenues for continued numerical study and computational science exist. We have not compared our nonlinear algorithms to the linear model in [7], but such a comparison would be worthwhile since it would likely establish a computational preference for RAMP models. The examples of this article only establish a proof of concept and do not yet verify broad applicability with regard to RAMP applications. Conducting a wide-scale numerical study seems prudent. Lastly, adapting our nonlinear tactic to problems with general uncertainty could provide alternative calculation schemes for challenging SOCPs.

References

1. I. Adler, R.D.C Monteiro, A geometric view of parametric linear programming. Algorithmica **8**, 161–176 (1992)
2. E. Almaas, A. Holder, K. Livingstone, Introduction to systems biology for mathematical programmers, in *Optimization in Medicine and Biology*, ed. by G. Lim, K. Lee Eva, chapter 11 (Taylor & Francis Group, Park Drive, 2008)
3. E. Almaas, Optimal flux patterns in cellular metabolic networks. Chaos Interdiscip. J. Nonlinear Sci. **17**(2), 026107 (2007)
4. A. Ben-Tal, A. Nemirovski, Robust convex optimization. Math. Oper. Res. **23**(4), 769–805 (1998)
5. A. Ben-Tal, A. Nemirovski, Robust solutions of uncertain linear programs. Oper. Res. Lett. **25**(1), 1–13 (1999)
6. A. Ben-Tal, A. Nemirovski, Robust solutions of linear programming problems contaminated with uncertain data. Math. Program. **88**(3), 411–424 (2000)
7. A. Ben-Tal, A. Nemirovski, On polyhedral approximations of the second-order cone. Math. Oper. Res. **26**(2), 193–205 (2001)
8. A.P. Burgard, P. Pharkya, C.D. Maranas, Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. Biotechnol. Bioeng. **84**(6), 647–657 (2003)
9. L. Chindelevitch, J. Trigg, A. Regev, B. Berger, An exact arithmetic toolbox for a consistent and reproducible structural analysis of metabolic network models. Nat. Commun. **5**, 4893 (2014)
10. A. Ebrahim, E. Almaas, E. Bauer, A. Bordbar, A.P. Burgard, R.L. Chang, A. Dräger, I. Famili, A.M. Feist, R.M.T. Fleming, S.S. Fong, V. Hatzimanikatis, M.J. Herrgård, A. Holder, M. Hucka, D. Hyduke, N. Jamshidi, S.Y. Lee, N. Le Novère, J.A. Lerman, N.E. Lewis, D. Ma, R. Mahadevan, C. Maranas, H. Nagarajan, A. Navid, J. Nielsen, L.K. Nielsen, J. Nogales, A. Noronha, C. Pal, B.O. Palsson, J.A. Papin, K.R. Patil, N.D. Price, J.L. Reed, M. Saunders, R.S. Senger, N. Sonnenschein, Y. Sun, I. Thiele, Do genome-scale models need exact solvers or clearer standards? Mol. Syst. Biol. **11**(10), 831 (2015)
11. D.A. Fell, J.R. Small, Fat synthesis in adipose tissue. An examination of stoichiometric constraints. Biochem. J. **238**(3), 781–786 (1986)
12. T. Gal, H.J. Greenberg (eds.), *Advances in Sensitivity Analysis and Parametric Programming* (Springer, Berlin, 1997)
13. H.J. Greenberg, An analysis of degeneracy. Naval Res. Logist. Q. **33**(4), 635–655 (1986)
14. H.J. Greenberg, The use of the optimal partition in a linear programming solution for postoptimal analysis. Oper. Res. Lett. **15**(4), 179–185 (1994)
15. A. Holder, Parametric LP analysis, in *Wiley Encyclopedia of Operations Research and Management Science*, ed. by J.J. Cochran, L.A. Cox, P. Keskinocak, J.P. Kharoufeh, J.C. Smith (Wiley, London, 2011)
16. B. Jansen, C. Roos, J.P. Vial, Interior-point methodology for linear programming: duality, sensitivity analysis and computational aspects, in *Optimization in Planning and Operation of Electric Power Systems*, ed. by K. Frauendorfer, H. Glavitsch, R. Bacher (Physica, Heidelberg, 1993), pp. 57–123
17. Z.A. King, J. Lu, A. Dräger, P. Miller, S. Federowicz, J.A. Lerman, A. Ebrahim, B.O. Palsson, N.E. Lewis, BiGG models: a platform for integrating, standardizing and sharing genome-scale models. Nucl. Acids Res. **44**(D1), D515–D522 (2015)
18. K.H. Lee, J.H. Park, T.Y. Kim, H.U. Kim, S.Y. Lee, Systems metabolic engineering of *Escherichia coli* for L-threonine production. Mol. Syst. Biol. **3**(1), 149 (2007)
19. M. MacGillivray, A. Ko, E. Gruber, M. Sawyer, E. Almaas, A. Holder, Robust analysis of fluxes in genome-scale metabolic pathways. Sci. Rep. **7**, 268 (2017)
20. C.D. Maranas, A.R. Zomorrodi, *Optimization Methods in Metabolic Networks* (Wiley, London, 2016)

21. J.D. Orth, T.M. Conrad, J. Na, J.A. Lerman, H. Nam, A.M. Feist, B.O. Palsson, A comprehensive genome-scale reconstruction of *Escherichia coli* metabolism—2011. Mol. Syst. Biol. **7**, 535 (2011)
22. B.O. Palsson, *Systems Biology: Constraint-based Reconstruction and Analysis* (Cambridge University Press, Cambridge, 2015)
23. J.H. Park, K.H. Lee, T.Y. Kim, S.Y. Lee, Metabolic engineering of *Escherichia coli* for the production of L-valine based on transcriptome analysis and *in silico* gene knockout simulation. Proc. Nat. Acad. Sci. **104**(19), 7797–7802 (2007)
24. T.D. Vo, H.J. Greenberg, B.O. Palsson, Reconstruction and functional characterization of the human mitochondrial metabolic network based on proteomic and biochemical data. J. Biol. Chem. **279**(38), 39532–39540 (2004)
25. M.R. Watson, Metabolic maps for the Apple II. Biochem. Soc. Trans. **12**(6), 1093–1094 (1984)
26. S.J. Wiback, I. Famili, H.J. Greenberg, B.Ø. Palsson, Monte Carlo sampling can be used to determine the size and shape of the steady-state flux space. J. Theoret. Biol. **228**(4), 437–447 (2004)
27. K. Yizhak, S.E. Le Dévédec, V.M. Rogkoti, F. Baenke, V.C. de Boer, C. Frezza, A. Schulze, B. van de Water, E. Ruppin, A computational study of the Warburg effect identifies metabolic targets inhibiting cancer migration. Mol. Syst. Biol. **10**(8), 744 (2014)