Myths and Counterexamples in Mathematical Programming

Harvey J. Greenberg

hjgreenberg@gmail.com

March 10, 2009
(Previous: January 5, 2009)

Available at the INFORMS Computing Society
Mathematical Programming Glossary
http://glossary.computing.society.informs.org/.

Contact the author to be notified of updates.

Cartoon courtesy of John Zakour ▷

Contents

Introduction **1**

Linear Programming **2**

LP 1. All redundant constraints can be removed. 3
LP 2. A degenerate basis implies there is a (weakly) redundant constraint. 3
LP 3. If an LP has an optimal solution, there is an extreme point of the feasible . . 3
LP 4. If one knows that an inequality constraint must hold with equality in every . 4
LP 5. In a dynamic LP, each period should be the same duration. 4
LP 6. If the optimal value of a slack variable is zero, the associated constraint is . . 4
LP 7. It is a good idea to convert free variables to the standard form by the 4
LP 8. The standard simplex method does not select a dominated column to enter . 5
LP 9. ***new*** At optimality, $\pi^* b = cx^*$ — that is, the inner product of the optimal . . 6
LP 10. Once the simplex method reaches an optimal vertex, it terminates. 6
LP 11. In the absence of degeneracy, the standard simplex method does not repeat . 7
LP 12. The standard simplex method does not revisit a basic feasible solution (that . 8
LP 13. ***new*** A simplex method does not cycle for an assignment problem. 9
LP 14. Suppose LP is solved and π_i is the dual price associated with the i^{th} 13
LP 15. Let $z(t) = \min\{cx : x \geq 0,\ Ax = b + th\}$, where h is a (fixed) m-vector. . . . 15
LP 16. ***new*** Alternative, integer-valued optima in a shortest path problem 15
LP 17. In a standard assignment problem, it is always optimal to assign the person- . 16
LP 18. Given an assignment problem with a non-optimal (but feasible) assignment, . 16
LP 19. A transportation problem with unique shipping costs has a uniquely 17
LP 20. The optimal dual price of a demand constraint equals the increase in the . . 17
LP 21. An increase in a demand requirement (with concomitant increase in supply) . 18
LP 22. ***new*** The line-drawing step of the Hungarian method for the assignment . . . 19
LP 23. The Stepping Stone Method always produces an optimal distribution. 20
LP 24. ***new*** The standard free-float formula for an activity in an activity-on-arc . . 21
LP 25. The maximum flow of commodities through a network equals the capacity . 22
LP 26. Undirected arcs can be replaced by a pair of oppositely oriented arcs, and . . 23
LP 27. The maximum two-way flow of a commodity through a network equals its . . 23
LP Background — Gomory-Hu Cut Trees 24
LP 28. Every connected network has a cut-tree. 24
LP 29. Removing an arc in a network cannot decrease users' latency or cost. 26
LP 30. Given strict improvement in the objective value, the standard simplex 26
LP 31. The worst-case time complexity of the simplex method is exponential and . . 27
LP 32. ***new*** The c-diameter of a non-empty polytope of dimension d with f facets . 27
LP 33. In employing successive bound reduction in a presolve, we can fix a variable . 28
LP 34. A factored form of the basis contains less error for FTRAN after reinversion. . 29
LP References 30

Integer Programming and Combinatorial Optimization **33**

IP 1. The integer solution is a rounding of some LPR solution. 33

IP 2. If a basic solution of the LPR of a MILP is not integer-valued, no rounding . . 34
IP 3. The LPR solves its associated ILP if, and only if, it has an optimal basic . . 34
IP 4. The number of extreme points of the integer hull is at least as great as the . . 34
IP 5. The number of extreme points of the integer hull is bounded by some 34
IP 6. ***new*** Every integral vector of an n-dimensional integral polyhedral pointed . . 35
IP 7. If some activities in an LP have a fixed charge, a valid MILP model is to . . 36
IP 8. If an ILP has an unbounded LPR, the ILP is also unbounded. 36
IP 9. ***new*** If an ILP has an unbounded LPR, the ILP is feasible. 37
IP 10. Suppose $x^*(t)$ is an optimal solution to the parametric ILP: 37
IP 11. One can solve an ILP finitely by adding cuts of the form 38
IP 12. For any 0-1 program with a single constraint, there exists a B&B algorithm . . 38
IP 13. An optimal schedule of jobs with deadlines on a single machine is given by . . 39
IP 14. A no-wait flow-shop's makespan cannot be worse by increasing the speed . . 39
IP 15. ***new*** The worst-case solution quality of the First-Fit bin-packing algorithm . . 40
IP 16. Suppose the edge weights satisfy the triangle inequality in a general routing . . 41
IP 17. In B&B it is best to branch from the largest upper bound (for 42
IP 18. If we increase the number of processors in a parallel B&B algorithm, the . . . 43
IP Background — Parallel Computation of a Schedule 43
IP 19. If we reduce the computation time of each processor, the makespan cannot . . 44
IP 20. If we add another processor, the makespan cannot increase. 44
IP 21. If we remove some precedence constraints, the makespan cannot increase. . . 45
IP 22. ***new*** Given jobs with unit time, an optimal schedule on n processors is 46
IP Background — Metaheuristics 46
IP 23. ***new*** Computing the depth of a discrete optimization problem P with 47
IP 24. ***new*** Computing the depth of a discrete optimization problem P with 47
IP 25. ***new*** In a metaheuristic search, it is better to start with a best-possible 48
IP 26. For N sufficiently large, simulated annealing visits the global minimum . . . 48
IP 27. In simulated annealing, it is always better to let the temperature decrease. . . 49
IP References 50

Dynamic Programming **51**
DP 1. Given a separable objective function, the Principle of Optimality enables . . 53
DP 2. The Principle of Optimality is a necessary condition for the optimality of . . 54
DP 3. In a dynamic lot size inventory model, decreasing setup costs does not . . . 54
DP 4. In a dynamic lot size inventory model, decreasing setup costs does not . . . 55
DP 5. ***new*** The Federgruen-Lee algorithm produces an optimal solution to the . . 56
DP 6. Consider a dynamic lot size problem in which lead times are stochastic, . . . 57
DP 7. ***new*** In a dynamic lot size problem, a stochastically greater lead time results . 58
DP 8. ***new*** In a multi-item, base-stock inventory system, the total order fill rate . . 59
DP 9. ***new*** Given continuous control over arrival and service rates in a network . . 59
DP Background — Infinite Horizon 60
DP 10. If there is a nearly optimal solution to a discounted DP with a finite number . 62
DP 11. For the discounted-return DP, there exists an ε-optimal policy for all $\varepsilon > 0$. . 62

DP 12. There exists a stationary policy that is B-opt. 63
DP 13. If the average-return DP has an optimal policy, there exists a stationary . . 64
DP 14. There exists an average ε-optimal policy for $\varepsilon > 0$ that is optimal for the . . 65
DP 15. If the optimal average-return is finite, there is an ε-optimal stationary policy . 66
DP 16. If a policy is B-opt, it optimizes the average return. 67
DP 17. If a stationary policy is B-opt, it is average-overtaking. 67
DP 18. Every stationary, 1-optimal policy is average-overtaking. 68
DP 19. If a policy is both B-opt and average-overtaking, it is liminf average 69
DP 20. If a policy is both B-opt and average-overtaking, it is limsup average 69
DP 21. If a policy is B-opt among stationary policies, it optimizes the average . . . 70
DP 22. If a policy is average-overtaking among stationary policies, it optimizes the . 70
DP 23. We can approximate an infinite horizon Markov decision process with a . . . 70
DP 24. A discounted-return stationary DP with finite decision space for each state . 71
DP 25. ***new*** Denardo's policy improvement algorithm computes a 1-optimal policy. . 72
DP References 73

Nonlinear Programming **74**
NLP 1. If f has continuous n^{th}-order derivatives, local behavior of f can be 75
NLP 2. Given differentiable functions, an optimal point must satisfy the Lagrange . 75
NLP 3. A point that satisfies the Lagrange multiplier necessary conditions is a local 76
NLP 4. Suppose f is analytic and x is a minimum of f. Then, $\nabla f(x) = 0$, and if . 76
NLP 5. Given $\min\{f(x,y) = g(x) + h(y) : ay = g(x)\}$, we can equivalently solve . . 77
NLP 6. ***new*** A smooth surface with one critical point that is a local, but not a . . 78
NLP 7. If f is continuous, the closure of its strict interior equals its level set. That 78
NLP Background — Semi-continuity 78
NLP 8. Given the objective is continuous and the feasible region is non-empty and 79
NLP 9. Given the objective is continuous and the feasible region is non-empty and 79
NLP 10. If the feasibility region and optimal value function are lsc at $b = 0$, so is . . 80
NLP 11. If the feasibility region and optimal value function are usc at $b = 0$, so is . 80
NLP 12. A set is convex if it contains the midpoint of any pair of its points. 81
NLP 13. A convex function is continuous. 81
NLP 14. A convex function is upper semi-continuous on its boundary. 82
NLP 15. Suppose $x^* \in \operatorname{argmax}\{f(x) : x \in X,\ g(x) \leq 0\}$ and $g(x^*) < 0$. Then, ... 82
NLP 16. ***new*** Cauchy's steepest ascent either diverges or converges to a relative . . 82
NLP 17. If f is concave on $[a,b]$, the truncated gradient algorithm converges to an . 83
NLP 18. ***new*** Rosen's projected gradient algorithm with linear constraints converges 83
NLP 19. A strictly quasiconvex function is quasiconvex. 84
NLP 20. Let f be convex on $X \neq \emptyset$, where $X \subseteq \mathbb{R}^n$, and the range of f is in $\mathbb{R}^m$. . . 84
NLP 21. Newton's method converges to a stationary point if the starting point is . . 85
NLP 22. Newton's method has converged when the change in the iterate value is less 85
NLP 23. Cyclic descent produces a local minimum. 86
NLP 24. If one algorithm has a higher order of convergence than another, it is better. 86
NLP 25. For a convex program, the Generalized Lagrange Multiplier Method 87

NLP 26. Ritter's method to solve a QP converges to a global optimal solution. . . . 87
NLP 27. Tui's method to maximize a convex function subject to linear constraints . 87
NLP 28. The Nelder-Mead method converges to a local optimum. 88
NLP 29. Rosen's decomposition method converges to an optimal solution for convex 89
NLP 30. ***new*** In methods of feasible directions, it is better to improve the objective 90
NLP 31. ***new*** Sequential Quadratic Programming (SQP) is quadratically convergent 91
NLP 32. ***new*** A barrier algorithm is globally convergent if the functions are smooth 91
NLP 33. ***new*** If every optimum in the trajectory of a barrier function satisfies strong 93
NLP 34. The central path converges to the analytic center of the optimality region . 94
NLP 35. If an NLP is infeasible, one can sequentially test for the deletion of 95
NLP References 95

Multiple-Objective Programming **97**
MOP 1. For LP, the Pareto frontier is a convex set. 98
MOP 2. Varying the weights of a convex combination of objectives generates the . 98
MOP 3. ***new*** Adding an objective function expands the set of Pareto-optima. . . . 99
MOP 4. Consider the multiple-objective convex program: 99
MOP 5. Consider a multiple-objective LP in standard form. A Pareto-maximum . 99
MOP 6. Let U be an increasing utility function on the range of f on X, and . . . 100
MOP 7. In a multiple-objective LP, one should put the greatest weight on the most 101
MOP 8. All Pareto-optimal solutions are equal. 101
MOP Background — Pareto-optimum Graph 102
MOP 9. The Pareto-minimum graph for spanning trees is connected. 102
MOP 10. The Pareto frontier is closed. 103
MOP 11. If the Pareto frontier contains an interior point, it must contain all interior 104
MOP 12. The Pareto frontier is edge-connected. 104
MOP Background — Lexico-optima and Bilevel Programs 105
MOP 13. Every Pareto-optimum is a solution to the lexico-optimum of some lexico- 105
MOP 14. A solution to the bilevel mathematical program can be found with some . 106
MOP 15. ***new*** An optimal solution to the linear bilevel program is either Pareto- . 107
MOP 16. ***new*** A linear bilevel optimum is Pareto-optimal if the coefficient vectors . 107
MOP 17. ***new*** The Parametric Complementary Pivot algorithm obtains a bilevel . . 108
MOP References 108

Special Forms of Mathematical Programs **110**
SF 1. A good cluster results from maximizing inter-cluster distances and 110
SF 2. A chance-constraint stochastic program has a solution if its certainty 110
SF 3. Someone with constant risk aversion always selects the less risky of two . . . 111
SF 4. There is always a maximum likelihood estimator. 112
SF 5. If demands for substitute products are pooled by a centralized system, the . . 112
SF 6. If a team has the greatest win percentage before and after some specified . . 114
SF 7. In revenue management, it is always better to re-solve dynamic allocations . . 114
SF 8. ***new*** Among no-memory rules to order list items, the move-to-front rule . . . 115

SF Background — Data Envelopment Analysis 116
SF 9. ***new*** A DMU that exhibits increasing return to scale continues to do so up . . 117
SF 10. ***new*** When using LP for DEA, it does not matter if you obtain a basic or . . 118
SF 11. ***new*** When entering a parallel queueing system, it is optimal to join the . . . 119
SF References 120

Index **122**

Acknowledgments

I thank those who contributed clarifications, suggestions, and updates: Hossein Arsham, Michel Balinski, András A. Benczúr, Harold Benson, Lorenz Biegler, Charles Blair, Winfried Bruns, Sharlee Climer, William L. Cooper, Marbelly Davila, John Dennis, Matthias Ehrgott, James Flynn, Bennett Fox, Saul I. Gass, Fred Glover, Jochen Gorski, Alan J. Hoffman, X.-D. Hu, Kathrin Klamroth, Ed Klotz, Sven Leyffer, Jan-Karel Lenstra, Ken McKinnon, Nimrod Megiddo, Sheldon Ross, Romeo Rizzi, Tim Roughgarden, Galen H. Sasaki, András Sebö, Hanif Sherali, Danny Sleator, Ralph Steuer, Philippe L. Toint, Craig Tovey, Frank Vanden Berghen, Andreas Wächter, Richard Weber, H. Paul Williams, Gerhard Woeginger, Henry Wolkowicz, and Majid Zohrehbandian. I am also grateful for the support and encouragement from Allen Holder.

Morality Code

 **Enjoy!**

Introduction

This is an update of what I began in 1996 and is posted with the ICS *Mathematical Programming Glossary*[3]. I follow the terms and notation in the *Glossary*, presenting examples in Linear Programming (LP), Integer Programming (IP), Dynamic Programming (DP), Nonlinear Programming (NLP), as well as Multiple-Objective (MOP) and Special Forms (SF). These comprise the sections that follow, but they are not a partition of mathematical programming in general. Many problems overlap; for example, a problem could be represented as an LP and a DP. Further, network problems are scattered in all of these. I placed an entry where I thought it should go for what we teach. Thus, most network problems are in the LP section, and a dynamic problem is in DP only if dynamic programming is the underlying methodology, not just that the model is dynamic.

The use of counterexamples to disprove some result that *seems* as though it is true is an old technique to deepen our understanding of the underlying concepts. One of the most impressive books I read in graduate school was *Counterexamples in Analysis*[1], by Gelbaum and Olmsted. Since then, similar books have appeared[2, 4, 5, 6, 7, 8]. Pedagogically, one could put a theorem out to the students of the form: $P \rightarrow Q$, then list some counterexamples to Q. The goal is for the student to discover P that makes Q true. What are the properties of the pathologies? Some myths are counterexamples to previously-published claims. Although that renders the original claim obsolete (unless repaired), it is included to demonstrate the construction of a counterexample in what appeared to be a valid result, not only to the author but also to at least two referees and one editor. What property did they all miss, and where does it present a flaw in the "proof?"

However, the myths and counterexamples I present here are not restricted to mathematical constructs. I have also included some practices that have grown in the folklore to dispel myths about "good" models, solutions, and computational efficiency. One class of myth to challenge our intuition is that the objective value cannot worsen when we improve resources and/or relax constraints. I list these as *better is worse* in the index. A related type of myth is *more for less*.

I use fairly standard notation for mathematical objects (though they have no universal standard), some of which are shown in Table 1.

Table 1: Notation

(a, b)	open interval	$\{x : a < x < b\}$
$[a, b]$	closed interval	$\{x : a \le x \le b\}$
$\mathbb{R}$	set of real values	$(-\infty, \infty)$
$\mathbb{Z}$	set of integer values	$\{\ldots, -2, -1, 0, 1, 2, \ldots\}$
$\mathbb{Q}$	set of rational values	$\left\{\frac{p}{q} : p, q \in \mathbb{Z} : q > 0\right\}$

I use $\mathbb{R}_+, \mathbb{Z}_+, \mathbb{Q}_+$ to restrict the values to be non-negative. For example, $\mathbb{R}_+ = [0, \infty)$. I use $\mathbb{R}^n, \mathbb{Z}^n, \mathbb{Q}^n$ to denote n-vectors whose coordinates belong to the indicated set. For example, $\mathbb{Z}^n = \{x = (x_1, \ldots, x_n) : x_j \in \mathbb{Z} \text{ for } j = 1, \ldots, n\}$. These can be combined. For example, $\mathbb{Q}^n_+ = \{x \in \mathbb{R}^n_+ : x_j \in \mathbb{Q} \text{ for } j = 1, \ldots, n\}$.

Following the *Glossary* notation, the general form of a mathematical program is given by:

$$\min f(x) : x \in X,\ g(x) \geq 0,\ h(x) = 0,$$

where $\emptyset \neq X \subseteq \mathbb{R}^n$, $f : X \to \mathbb{R}$, $g : X \to \mathbb{R}^m$, $h : X \to \mathbb{R}^M$. (The sense of optimization could be max.) The functional relations are called *constraints*.

I welcome suggestions for future versions.

General References

[1] B. R. Gelbaum and J. M. H. Olmsted. *Counterexamples in Analysis*. Holden-Day, San Francisco, CA, 1964.

[2] B. R. Gelbaum and J. M. H. Olmsted. *Theorems and Counterexamples in Mathematics*. Springer-Verlag, New York, NY, 1990.

[3] A. Holder, editor. *Mathematical Programming Glossary*. INFORMS Computing Society, http://glossary.computing.society.informs.org, 2006–08. (Originally authored by Harvey J. Greenberg, 1999–2006.)

[4] J. P. Romano and A. F. Siegel. *Counterexamples in Probability and Statistics*. Wadsworth & Brooks/Cole, Monterey, CA, 1986.

[5] S. Ya. Serovaiskii. *Counterexamples in Optimal Control Theory*. Inverse and Ill-Posed Problems. VSP, The Netherlands, 2004.

[6] L. A. Steen and J. A. Seebach, Jr. *Counterexamples in Topology*. Holt, Rinehart and Winston, Inc., New York, NY, 1971.

[7] J. M. Stoyanov. *Counterexamples in Probability*. John Wiley & Sons, New York, NY, 2nd edition, 1997.

[8] G. L. Wise and E. B. Hall. *Counterexamples in Probability and Real Analysis*. Oxford University Press, New York, NY, 1993.

Linear Programming

The general form of a linear program (LP) is the optimization of a linear function subject to a system of linear equations and inequalities. The *standard form* is

$$\min\ cx : Ax = b,\ x \geq 0,$$

where $\text{rank}(A) = m =$ number of equations. This form is particularly useful when considering the simplex method.

When talking about duality, I use the *canonical form*:

$$\min\ cx : Ax \geq b,\ x \geq 0.$$

(No rank condition on A.) This renders the dual prices non-negative, giving the dual canonical form:

$$\max\ \pi b : \pi A \leq c,\ \pi \geq 0.$$

Unless stated otherwise, or implied from context, the LP in question could be any linear system; it need not be in standard or canonical form.

The *standard simplex method* is the original pivot-selection rule by Dantzig, applied to the standard form — a variable with the greatest reduced cost (rate of improvement) is chosen to enter the basis. An alternative is the *best-gain* criterion, which evaluates the actual gain of each candidate to enter the basis by computing its change in level and multiplying by the rate of improvement.

A constraint is *redundant* if its removal does not change the set of feasible points. An inequality is an *implied equality* if it must hold with equality in every feasible solution.

LP Myth 1. *All redundant constraints can be removed.*

The reason this is incorrect is that once a redundancy is removed, the other constraints may no longer be redundant.

Counterexample. $x, y \geq 0$ and $x - y = 0$. Each non-negativity constraint is redundant, but they cannot both be removed. The redundancy of $x \geq 0$ follows from the equation and the non-negativity of y: $x = y \geq 0$.

Practical use was first reported by Tomlin and Welch[51], and that led to a theory of *common dependency sets* by Greenberg[29].

LP Myth 2. *A degenerate basis implies there is a (weakly) redundant constraint.*

Counterexample. Consider $y \geq 0$, $x \geq 1$, $x+y \leq 1$. The only feasible point is $(x, y) = (1, 0)$ with slack and surplus variables both 0. Thus, each of the possible feasible bases is degenerate, but no constraint is redundant.

Sierksma and Tijssen[46] generalized this: If a face of dimension $n-1$ or $n-2$ is degenerate, the defining linear inequalities are not minimal — that is, the system must contain either a redundant inequality or an implied equality. Note the special conditions on dimension. For $n \geq 3$, it cannot apply generally to an extreme point (face of 0 dimension). A pyramid is a counterexample for $n = 3$. The pyramid's top extreme point is degenerate because it is the intersection of 4 planes, but none of the defining inequalities is redundant or an implied equality.

LP Myth 3. *If an LP has an optimal solution, there is an extreme point of the feasible region that is optimal.*

Counterexample. Arsham[2, #rlpnov] provides the following: $\max x_1 + x_2 : x_1 + x_2 \leq 5$. The feasible set is a polyhedron with no extreme point. This occurs because we do not require the variables to be non-negative.

The myth is true when the LP is in standard form. Converting the example to standard form increases the dimension:

$$\begin{gathered} \max u_1 - v_1 + u_2 - v_2 : \\ u_1 - v_1 + u_2 - v_2 + x_3 = 5, \\ u_1, v_1, u_2, v_2, x_3 \geq 0, \end{gathered}$$

where we have augmented the slack variable, x_3, and we have partitioned each of the original variables into their positive and negative parts:

$$x_1 = u_1 - v_1 \text{ and } x_2 = u_2 - v_2.$$

(Be sure to see LP Myth 7.)

In this higher-dimensional space, it is true that an extreme point is optimal — in particular, $(u_1, v_1, u_2, v_2, x_3) = (5, 0, 0, 0, 0)$. In fact, there are three extreme points; the other two are (0,0,5,0,0) and (0,0,0,0,5). Each of these three extreme points is optimal for some objective value coefficients, spanning all that render the LP optimal (vs. unbounded).

LP Myth 4. *If one knows that an inequality constraint must hold with equality in every optimal solution, it is better to use the equality in the constraint because it will reduce the solution time.*

First, it is not necessarily the case that it will reduce the solution time — the solver could get a first feasible solution faster with the inequality formulation. Second, even if the tighter version solves faster (perhaps by pre-solve reduction), it is generally better to let the model tell you the answer than for you to wire the result. Your intuition could be wrong, or there could be a data entry error that goes undetected with the equality constraint. A better approach is to attach a back-end report to examine all things "known" to be true and flag the violations. Thus, if an inequality is slack and you expected it to be tight, you can investigate why the model did what it did.

LP Myth 5. *In a dynamic LP, each period should be the same duration.*

This is tacitly implied in many textbook examples. The reality is that we know more about what is likely to happen tomorrow than next year. In general, data can provide forecasts for demands, supplies, and other model parameters, but the accuracy tends to be less as the time is further into the future. One may have, for example, a 5-year planning model with the first 12 time periods being months, the next 4 periods being quarters, and the last 3 being years.

LP Myth 6. *If the optimal value of a slack variable is zero, the associated constraint is binding.*

As suggested by H. P. Williams, this myth reflects confusion in terminology. An inequality constraint is *active* at a point if it holds with equality; it is *binding* if its removal changes the solution.

Counterexample. $\max\ x_1 : x \geq 0,\ x_1 + 2x_2 \leq 3,\ 2x_1 + x_2 \leq 3,\ x_1 + x_2 \leq 2.$

The (unique) optimal solution is at $x^* = (1, 1)$, and all slack variables are zero. However, the last constraint is not binding; it is redundant.

LP Myth 7. *It is a good idea to convert free variables to the standard form by the expression: $x = u - v$, where u is the positive part and v is the negative part of x.*

Too often students (and new graduates) do this, perhaps thinking it is necessary due to the text they used. However, all solvers handle free variables directly.

For a simplex method, the conversion requires a change in basis whenever x needs to change sign. This is an unnecessary pivot, wasting time and space. Recognition of free variables allows the solver to put all free variables into the basis at the start (dealing with linear dependence, if that should be a problem). Once in the basis, a free variable cannot block an entrant, so it simply stays there. Some solvers also use the free variable to eliminate a row (and restore it after a solution is obtained). Thus, it is never a good idea to perform this conversion when using a simplex method.

For an interior method, this causes the optimality region to be unbounded (if it is not empty). Whatever the value of x^*, there is an infinite number of values of u^* and v^* that yield the same difference, $u^* - v^*$. During the iterations, it is not unusual for u and v to diverge, while maintaining a constant difference, and this divergence can cause numerical problems for the algorithm (especially for convergence detection).

LP Myth 8. *The standard simplex method does not select a dominated column to enter the basis.*

Consider LP in canonical form:

$$\max\ cx : x \geq 0,\ Ax \leq b.$$

A column, j, is *dominated* if there exists $k \neq j$ such that

$$c_k \geq c_j \text{ and } A_k \leq A_j.$$

Counterexample. Blair[10] provides the following:

$$\begin{array}{rrrrrrrrrr}
\max & 5x_1 & + & 3x_2 & + & x_3 & + & x_4 & & \\
 & x_1 & - & x_2 & + & 5x_3 & + & 3x_4 & \leq & 10 \\
 & 3x_1 & + & x_2 & + & x_3 & + & x_4 & \leq & 40 \\
 & -2x_1 & + & x_2 & - & 3x_3 & - & 3x_4 & \leq & 10 \\
 & & & & & x \geq 0. & & & &
\end{array}$$

After adding slack variables to convert to standard form, the first simplex tableau is:

	Level	x_1	x_2	x_3	x_4	s_1	s_2	s_3
$\leftarrow$	10	1	−1	5	3	1	0	0
	40	3	1	1	1	0	1	0
	10	−2	1	3	3	0	0	1
	0	5	3	1	1	0	0	0
		$\uparrow$						

The first pivot exchange is $s_1 \leftarrow x_1$:

	Level	x_1	x_2	x_3	x_4	s_1	s_2	s_3
	10	1	−1	5	3	1	0	0
$\leftarrow$	10	0	4	−14	−8	−3	1	0
	30	0	−1	13	9	2	0	1
	50	0	8	−24	−14	−5	0	0
			$\uparrow$					

Column 3 is dominated by column 4, but it enters the basis next:

Level	x_1	x_2	x_3	x_4	s_1	s_2	s_3
$12\frac{1}{2}$	1	0	$1\frac{1}{2}$	1	$\frac{1}{4}$	$\frac{1}{4}$	0
$2\frac{1}{2}$	0	1	$-3\frac{1}{2}$	-2	$-\frac{3}{4}$	$\frac{1}{4}$	0
$32\frac{1}{2}$	0	0	$9\frac{1}{2}$	7	$1\frac{1}{4}$	$\frac{1}{4}$	1
69	0	0	4	2	1	-2	0
			↑				

One way to look at Blair's example is that the dominance conditions are not generally preserved as the basis changes. This is evidenced by $[B^{-1}A_4]_2 = -2 \not< -3\frac{1}{2} = [B^{-1}A_3]_2$.

Another view is to drop the first two columns entirely and consider a 2-variable LP with an initial basis that is slack. The values of A do not affect the selection of the basis entrant. With equal reduced costs, the first variable, which is dominated.

Level	x_3	x_4	s_1	s_2	s_3
10	5	3	1	0	0
40	1	1	0	1	0
10	3	3	0	0	1
0	1	1	0	0	0
	↑				

LP Myth 9. **new** *At optimality, $\pi^* b = cx^*$ — that is, the inner product of the optimal dual variables on the constraints and the right-hand side values equals the optimal primal objective value.*

While this is true for standard and canonical forms, it fails when primal bounds are handled directly. Consider the primal-dual LPs:

Primal	Dual
$\min\ cx : 0 \le x \le U,\ Ax \ge b.$	$\max\ \pi b - \mu U : \pi, \mu \ge 0,\ \pi A - \mu \le c.$

At optimality, $cx^* = \pi^* b - \mu^* U$, so one must be careful to subtract $\mu^* U$ from $\pi^* b$ to obtain the correct equation.

Support for handling bounds directly, rather than including them in other constraints, is an example of how optimization software may use different conventions than in the theory. Such deviations from theory in the world of optimization software include reporting dual prices and/or reduced costs as the negative of their theoretically-correct values. One must check the manual or run a small test case to see how they are reported in any particular solver. (ANALYZE[27] reports theoretically-correct values, changing solver-values as needed.)

LP Myth 10. *Once the simplex method reaches an optimal vertex, it terminates.*

The fallacy is that the Basic Feasible Solution (BFS) reached must be both primal and dual optimal for the tableau to be terminal.

Counterexample. Gerard Sierksma provided the following (converted to standard form):

$$\begin{array}{rcccccc} & \max\ x_1 + x_2 : x, s \geq 0 & & & & & \\ x_1 & & & + & s_1 & = & 1 \\ & + & x_2 & + & s_2 & = & 1 \\ x_1 & + & x_2 & - & s_3 & = & 2 \end{array}$$

The extreme point $(1, 1)$ is optimal and corresponds to three BFSs:

basic	level	s_2	s_3
x_1	1	-1	-1
x_2	1	1	0
s_1	0	1	1
$-z$	2	0	1
			$\uparrow$

basic	level	s_1	s_3
x_1	1	1	0
x_2	1	-1	-1
s_2	0	1	1
$-z$	2	0	1
			$\uparrow$

basic	level	s_1	s_2
x_1	1	1	0
x_2	1	0	1
s_3	0	1	1
$-z$	2	-1	-1
		Terminal	

Only the third of these is both primal and dual optimal; the other two are not terminal. The reason is the myopic nature of rates, oblivious to the degeneracy:

Tableau 1

Δx_1	=	Δs_3
Δx_2	=	0
Δs_1	=	$-\Delta s_3$
Δz	=	Δs_3

Tableau 2

Δx_1	=	0
Δx_2	=	Δs_3
Δs_2	=	$-\Delta s_3$
Δz	=	Δs_3

Tableau 3

Δx_1	=	$-\Delta s_1$
Δx_2	=	$-\Delta s_2$
Δs_3	=	$-\Delta s_1 - \Delta s_2$
Δz	=	$-\Delta s_1 - \Delta s_2$

Tableau 1 sees a rate of change in the objective value as $+1$ per unit of increase in s_3 (keeping $s_2 = 0$). The linear equations show that the net rate of change in the objective value (z) is $+1$, which is its reduced cost. Similarly, tableau 2 sees a rate of change in the objective value as $+1$ per unit of increase in s_3 (keeping $s_1 = 0$). The linear equations show that the net rate of change in the objective value (z) is $+1$, which is its reduced cost. The third tableau has s_3 in the basis, so it responds to changes in either of the first two slack variables. Any increase in one slack value causes a decreases in its corresponding variable while keeping the other primary variable at 1 — for example,

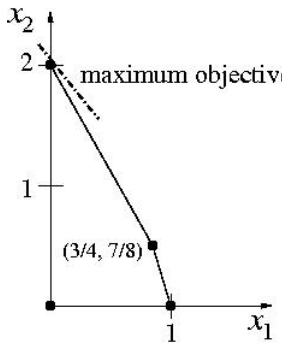
$$\Delta s_1 > 0 \Rightarrow \Delta x_1 = -\Delta s_1 < 0 \text{ and } \Delta x_2 = 0.$$

(The value of s_3 also decreases at the same rate, which does not affect the objective value.) The net effect is that the objective value decreases at that same unit rate, as indicated by the reduced cost. The same analysis applies to increasing s_2.

LP Myth 11. *In the absence of degeneracy, the standard simplex method does not repeat a basis exchange.*

Saaty[45] presented this conjecture with some supporting intuition. In the absence of degeneracy, this has a unique choice of departing variable for the exchange. However, Goldman and Kleinman[25] found the following:

Counterexample. This is a special case of the family of counterexamples in [25]:



$$\max\ 3x_1 + 2x_2 :$$
$$7x_1 + 2x_2 \le 7$$
$$3x_1 + 2x_2 \le 4$$
$$x_1, x_2 \ge 0.$$

Adding slack variables $s = (s_1, s_2)$, and starting at $x = (0, 0)$, the standard simplex iterations are:

Iteration	Vertex	Basic Variables	Basis Exchange
0	$(0,0)$	s_1, s_2	$s_1 \leftarrow x_1$
1	$(1,0)$	x_1, s_2	$s_2 \leftarrow x_2$
2	$(3/4, 7/8)$	x_1, x_2	$x_1 \leftarrow s_1$
3	$(0,2)$	s_1, x_2	

LP Myth 12. *The standard simplex method does not revisit a basic feasible solution (that is, cycle) as it pivots to an optimum.*

Hoffman[31] gave the first example of cycling in the standard simplex method, which has 11 variables and 3 equations. (Also see Gass and Vinjamuri[21] for elaboration and a collection of cycling examples.)

Counterexample. The following is due to Beale[5], with only 7 variables and 3 equations.

x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
$(\mathbf{1/4})$	-60	$\mathbf{-1/25}$	9	1			0
$\mathbf{1/2}$	-90	$\mathbf{-1/50}$	3		1		0
		1				1	1
$\mathbf{-3/4}$	150	$\mathbf{-1/50}$	6	•	•	•	0
↑							

1	-240	$\mathbf{-4/25}$	36	4			0
	$(\mathbf{30})$	$\mathbf{3/50}$	-15	-2	1		0
		1				1	1
•	-30	$\mathbf{-7/50}$	33	3	•	•	0
	↑						

1		$(\mathbf{8/25})$	−84	−12	8		0
	1	$\mathbf{1/500}$	$\mathbf{-1/2}$	$\mathbf{-1/15}$	$\mathbf{1/30}$		0
		1				1	1
•	•	$\mathbf{-2/25}$	18	1	1	•	0

↑

$\mathbf{25/8}$		1	$\mathbf{-525/2}$	$\mathbf{-75/2}$	28		0
$\mathbf{-1/160}$	1		$(\mathbf{1/40})$	$\mathbf{1/120}$	$\mathbf{-1/60}$		0
$\mathbf{-25/8}$			$\mathbf{525/2}$	$\mathbf{75/2}$	−25	1	1
$\mathbf{1/4}$	•	•	−3	−2	3	•	0

↑

$\mathbf{-125/2}$	10500	1		$(\mathbf{50})$	−150		0
$\mathbf{-1/4}$	40		1	$\mathbf{1/3}$	$\mathbf{-2/3}$		0
$\mathbf{-125/2}$	−10500			−50	150	1	1
$\mathbf{-1/2}$	120	•	•	−1	1	•	0

↑

$\mathbf{-5/4}$	210	$\mathbf{1/50}$		1	−3		0
$\mathbf{1/6}$	−30	$\mathbf{-1/150}$	1		$(\mathbf{1/3})$		0
		1				1	1
$\mathbf{-7/4}$	330	$\mathbf{1/50}$	•	•	−2	•	0

↑

Next tableau is same as first.

LP Myth 13. new *A simplex method does not cycle for an assignment problem.*

"A simplex method" is taken to mean any sequence of (adjacent) basic feasible solutions that enters a basic variable with negative reduced cost. This need not be the standard simplex method, which selects one with the most negative reduced cost.

Counterexample. Gassner[22] provides a 4×4 with costs:

$$c = \begin{bmatrix} 3 & 5 & 5 & 11 \\ 9 & 7 & 9 & 15 \\ 7 & 7 & 11 & 13 \\ 13 & 13 & 13 & 17 \end{bmatrix}.$$

Begin with the diagonal assignment: $x_{11} = x_{22} = x_{33} = x_{44} = 1$. Let the additional 3 basic (degenerate) variables be x_{12}, x_{23}, and x_{34}. Here is the initial (abbreviated) tableau:

			Nonbasic								
	Basic	level	x_{13}	x_{14}	x_{21}	x_{24}	x_{31}	x_{32}	x_{41}	x_{42}	x_{43}
	x_{11}	1	0	0	1	0	1	0	1	0	0
	x_{22}	1	−1	−1	1	0	1	1	1	1	0
	x_{33}	1	0	−1	0	−1	1	1	1	1	1
	x_{44}	1	0	0	0	0	0	0	1	1	1
	x_{12}	0	1	1	−1	0	−1	0	−1	0	0
←	x_{23}	0	**1**	1	0	1	−1	−1	−1	−1	0
	x_{34}	0	0	1	0	1	0	0	−1	−1	−1
		38	−2	2	4	4	0	−2	2	0	−2
			↑								

There are three candidates for entering the basis; select x_{13}. Then, there are two candidates to leave the basis; select x_{23}. The pivot results in the following tableau:

			Nonbasic								
	Basic	level	x_{14}	x_{21}	x_{23}	x_{24}	x_{31}	x_{32}	x_{41}	x_{42}	x_{43}
	x_{11}	1	0	1	0	0	1	0	1	0	0
	x_{22}	1	0	1	1	1	0	0	0	0	0
	x_{33}	1	−1	0	0	−1	1	1	1	1	1
	x_{44}	1	0	0	0	0	0	0	1	1	1
←	x_{12}	0	0	−1	−1	−1	0	1	0	**1**	0
	x_{13}	0	1	0	1	1	−1	−1	−1	−1	0
	x_{34}	0	1	0	0	1	0	0	−1	−1	−1
		38	4	4	2	6	−2	−4	0	−2	−2
										↑	

The next entering variable is x_{42}, which has reduced cost = −2 (not the most negative). In each of the subsequent tableaux, Gassner selects an entrant with reduced cost = −2, although some have a reduced cost = −4, which would be selected by the standard simplex method.

			Nonbasic								
	Basic	level	x_{12}	x_{14}	x_{21}	x_{23}	x_{24}	x_{31}	x_{32}	x_{41}	x_{43}
	x_{11}	1	0	0	1	0	0	1	0	1	0
	x_{22}	1	0	0	1	1	1	0	0	0	0
	x_{33}	1	−1	−1	1	1	0	1	0	1	1
	x_{44}	1	−1	0	1	1	1	0	−1	1	1
	x_{42}	0	1	0	−1	−1	−1	0	1	0	0
	x_{13}	0	1	1	−1	0	0	−1	0	−1	0
←	x_{34}	0	1	1	−1	−1	0	0	**1**	−1	−1
		38	2	4	2	0	4	−2	−2	0	−2
									↑		

			Nonbasic								
	Basic	level	x_{12}	x_{14}	x_{21}	x_{23}	x_{24}	x_{31}	x_{34}	x_{41}	x_{43}
	x_{11}	1	0	0	1	0	0	1	0	1	0
	x_{22}	1	0	0	1	1	1	0	0	0	0
	x_{33}	1	−1	−1	1	1	0	1	0	1	1
	x_{44}	1	0	1	0	0	1	0	1	0	0
←	x_{42}	0	0	−1	0	0	−1	0	−1	**1**	1
	x_{13}	0	1	1	−1	0	0	−1	0	−1	0
	x_{32}	0	1	1	−1	−1	0	0	1	−1	−1
		38	4	6	0	−2	4	−2	2	−2	−4
										↑	

			Nonbasic								
	Basic	level	x_{12}	x_{14}	x_{21}	x_{23}	x_{24}	x_{31}	x_{34}	x_{42}	x_{43}
	x_{11}	1	0	1	1	0	1	1	1	−1	−1
	x_{22}	1	0	0	1	1	1	0	0	0	0
	x_{33}	1	−1	0	1	1	1	1	1	−1	0
	x_{44}	1	0	1	0	0	1	0	1	0	0
	x_{41}	0	0	−1	0	0	−1	0	−1	1	1
←	x_{13}	0	1	0	−1	0	−1	−1	−1	1	**1**
	x_{32}	0	1	0	−1	−1	−1	0	0	1	0
		38	4	4	0	−2	2	−2	0	2	−2
											↑

			Nonbasic								
	Basic	level	x_{12}	x_{13}	x_{14}	x_{21}	x_{23}	x_{24}	x_{31}	x_{34}	x_{42}
	x_{11}	1	1	1	1	0	0	0	0	0	0
	x_{22}	1	0	0	0	1	1	1	0	0	0
	x_{33}	1	−1	0	0	1	1	1	1	1	−1
	x_{44}	1	0	0	1	0	0	1	0	1	0
←	x_{41}	0	−1	−1	−1	**1**	0	0	1	0	0
	x_{43}	0	1	1	0	−1	0	−1	−1	−1	1
	x_{32}	0	1	0	0	−1	−1	−1	0	0	1
		38	6	2	4	−2	−2	0	−4	−2	4
						↑					

Standard simplex enters x_{31}

			Nonbasic								
	Basic	level	x_{12}	x_{13}	x_{14}	x_{23}	x_{24}	x_{31}	x_{34}	x_{41}	x_{42}
	x_{11}	1	1	1	1	0	0	0	0	0	0
	x_{22}	1	1	1	1	1	1	−1	0	−1	0
	x_{33}	1	0	1	1	1	1	0	1	−1	−1
	x_{44}	1	0	0	1	0	1	0	1	0	0
	x_{21}	0	−1	−1	−1	0	0	1	0	1	0
	x_{43}	0	0	0	−1	0	−1	0	−1	1	1
←	x_{32}	0	0	−1	−1	−1	−1	**1**	0	1	1
		38	4	0	2	−2	0	−2	−2	2	4
								↑			

			Nonbasic								
	Basic	level	x_{12}	x_{13}	x_{14}	x_{23}	x_{24}	x_{32}	x_{34}	x_{41}	x_{42}
	x_{11}	1	1	1	1	0	0	0	0	0	0
	x_{22}	1	1	0	0	0	0	1	0	0	1
	x_{33}	1	0	1	1	1	1	0	1	−1	−1
	x_{44}	1	0	0	1	0	1	0	1	0	0
	x_{21}	0	−1	0	0	1	1	−1	0	0	−1
	x_{43}	0	0	0	−1	0	−1	0	−1	1	1
←	x_{31}	0	0	−1	−1	−1	−1	1	0	1	1
		38	4	−2	0	−4	−2	2	−2	4	6
							↑				

			Nonbasic								
	Basic	level	x_{12}	x_{13}	x_{14}	x_{21}	x_{23}	x_{32}	x_{34}	x_{41}	x_{42}
	x_{11}	1	1	1	1	0	0	0	0	0	0
	x_{22}	1	1	0	0	0	0	1	0	0	1
	x_{33}	1	1	1	1	−1	0	1	1	−1	0
	x_{44}	1	1	0	1	−1	−1	1	1	0	1
	x_{24}	0	−1	0	0	1	1	−1	0	0	−1
←	x_{43}	0	−1	0	−1	1	1	−1	−1	1	0
	x_{31}	0	−1	−1	−1	1	0	0	0	1	0
		38	2	−2	0	2	−2	0	−2	4	4
							↑				

			Nonbasic								
	Basic	level	x_{12}	x_{13}	x_{14}	x_{21}	x_{32}	x_{34}	x_{41}	x_{42}	x_{43}
	x_{11}	1	1	1	1	0	0	0	0	0	0
	x_{22}	1	1	0	0	0	1	0	0	1	0
	x_{33}	1	1	1	1	−1	1	1	−1	0	0
	x_{44}	1	0	0	0	0	0	0	1	1	1
←	x_{24}	0	0	0	1	0	0	1	−1	−1	−1
	x_{23}	0	−1	0	−1	1	−1	−1	1	0	1
	x_{31}	0	−1	−1	−1	1	0	0	1	0	0
		38	0	−2	−2	4	−2	−4	6	4	2
					↑						

			Nonbasic								
	Basic	level	x_{12}	x_{13}	x_{21}	x_{24}	x_{32}	x_{34}	x_{41}	x_{42}	x_{43}
	x_{11}	1	1	1	0	−1	0	−1	1	1	1
	x_{22}	1	1	0	0	0	1	0	0	1	0
	x_{33}	1	1	1	−1	−1	1	0	0	1	1
	x_{44}	1	0	0	0	0	0	0	1	1	1
	x_{14}	0	0	0	0	1	0	1	−1	−1	−1
	x_{23}	0	−1	0	1	1	−1	0	0	−1	0
←	x_{31}	0	−1	−1	1	1	0	1	0	−1	−1
		38	0	−2	4	2	−2	−2	4	2	0
								↑			

			Nonbasic								
	Basic	level	x_{12}	x_{13}	x_{21}	x_{24}	x_{31}	x_{32}	x_{41}	x_{42}	x_{43}
	x_{11}	1	0	0	1	0	1	0	1	0	0
	x_{22}	1	1	0	0	0	0	1	0	1	0
	x_{33}	1	1	1	−1	−1	0	1	0	1	1
	x_{44}	1	0	0	0	0	0	0	1	1	1
←	x_{14}	0	1	1	−1	0	−1	0	−1	0	0
	x_{23}	0	−1	0	1	1	0	−1	0	−1	0
	x_{34}	0	−1	−1	1	1	1	0	0	−1	−1
		38	−2	−4	6	4	2	−2	4	0	−2
			↑								

The next pivot brings us back to the initial tableau, thus completing the cycle. (Also see Gass[20, Chap. 10].)

Gassner proved that a simplex method cannot cycle for $n < 4$, so the above is an example of a smallest assignment problem for which a simplex method cycles. To my knowledge, there is no example of an assignment problem that cycles with the standard simplex method. ***Please let me know if you have one.***

LP Myth 14. *Suppose LP is solved and π_i is the dual price associated with the i^{th} constraint. Then, the same solution is obtained when removing the constraint and subtracting $\pi_i A_{i\bullet} x$ from the objective.*

The reason this incorrect is because other solutions might exist to the revised LP. This error has caused some to say that a tax is equivalent to a prohibition in the sense that the dual price can be used as a tax in an LP that adds the tax to the objective and removes the prohibition constraint.

Counterexample. $\min\ x + 2y : 0 \le x, y \le 10,\ x + y = 1$. The solution is $(x^*, y^*) = (1, 0)$ with dual price, $\pi = 1$ for the last constraint. Then, the *tax equivalent* is:

$$\min y : 0 \le x, y \le 10.$$

The solutions are of the form $(x, 0)$, where x is arbitrary in $[0, 10]$. Using a simplex method, the solution obtained will be one of the extremes: $x = 0$ or $x = 10$, neither of which is the original solution. In fact, the basic solution $(10, 0)$ violates the original constraint.

A motivating application is the control of emissions of some pollutant. In an LP, there may be a prohibition constraint:

$$\max\ cx :\ x \ge 0,\ Ax = b,\ dx \le \delta,$$

where d_j is the rate of emission caused by activity j, and δ is the limit. The tax model has the form:

$$\max\ cx - \tau dx :\ x \ge 0,\ Ax = b,$$

where τ is the shadow price associated with the prohibition constraint (equal to an extreme dual-variable value). Although the prohibition solution is optimal in this tax model, there may be other optimal solutions that violate the limit.

Consider a numerical example for electricity generation by three sources: scrubbed coal, oil, and uranium. The variables are fuel purchases and generation. The prohibition is a limit

on sulfur emissions (LSU) while satisfying electricity demand (DEL). The B-rows balance fuels.

	Purchase			*Generate*					*Dual*
	PCL	POL	PUR	GSC	GOL	GUR			*Price*
$COST$	18	15	20	0.9	0.6	0.4	=	min	
BCL	1			−1			≥	0	18
BOL		1			−1		≥	0	15
BUR			1			−1	≥	0	20
DEL				0.3	0.3	0.4	≥	10	67.5
LSU				0.2	0.6		≤	6	−8.25
bound					25	10			
level	15	5	10	15	5	10			

The solution to this LP generates all the electricity it can from uranium, which is 4 units, and the remaining 6 units from the only combination of oil and scrubbed coal to satisfy both the demand and the sulfur limit: $GSC = 15$ and $GOL = 5$. The issue is whether the sulfur-limit constraint can be replaced by a tax on sulfur emissions.

The tax model adds 8.25 times the LSU coefficients to the objective:

$$COST + 8.25(0.2\,GSC + 0.6\,GOL).$$

The tax model and its two optimal solutions are:

	Purchase			*Generate*					*Dual*
	PCL	POL	PUR	GSC	GOL	GUR			*Price*
$COST$	18	15	20	2.55	5.55	0.4	=	min	
BCL	1			−1			≥	0	18
BOL		1			−1		≥	0	15
BUR			1			−1	≥	0	20
DEL				0.3	0.3	0.4	≥	10	67.5
bound					25	10			
level[1]	20	0	10	20	0	10			
level[2]	0	20	10	0	20	10			

The tax LP has alternative solutions with extremes that contain the original limit of 6 units of sulfur emissions. At one extreme (level[1]), the company uses no oil; it generates the 6 units of remaining electricity (after nuclear generation) by scrubbed coal. This complies with the sulfur limit with slack: the amount of sulfur emitted is only 4 units. At the other extreme (level[2]), the company uses no scrubbed coal. This violates the sulfur limit: the amount emitted is 12 units. (This is the solution to the original model without the sulfur limit constraint; the prohibition was specified to disallow this.)

Because the 'equivalent' tax model could result in a violation, the tax might be levied at slightly more than the dual price of \$8.25. In that case, however, the result is overly conservative, resulting in much less sulfur emission than was deemed necessary for good health while raising the cost above its minimum.

The problem is the bang-bang phenomenon with linear models: solutions respond to data changes by an all-or-nothing principle. This reflects the fact that constant rates of substitution

cause trade-offs that are marginally beneficial to be globally beneficial; only a constraint can stop the negotiation.

LP Myth 15. *Let $z(t) = \min\{cx : x \geq 0,\ Ax = b + th\}$, where h is a (fixed) m-vector. Then, z is piece-wise linear, where the break-points occur wherever there must be a basis change.*

The fallacy is the last sentence. The reason that this is not correct is that not every change in basis implies the slope must change.

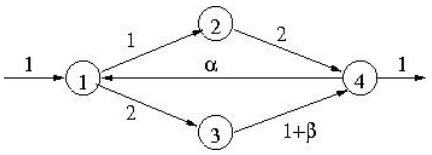
Counterexample. $\min x - y : x, y \geq 0,\ x - y = t$. Because $z(t) = t$ for all t, there is only one linearity interval (no breakpoints). However, for t positive, we must have x basic, and for t negative, we must have y basic. At $t = 0$ there are two optimal bases, and the basis must change as t varies in either of the two directions. Thus, although the basis must change (to be feasible), the point at which this occurs (namely, at $t = 0$) is not a breakpoint of z.

Note: the interior approach gives the correct answer (that is, the slope changes when the optimal partition changes). In the example, the optimal support has both $x > 0$ and $y > 0$, no matter what the value of t. Thus, the optimal partition does not change.

LP Myth 16. **new** *Alternative, integer-valued optima in a shortest path problem correspond to alternative shortest paths.*

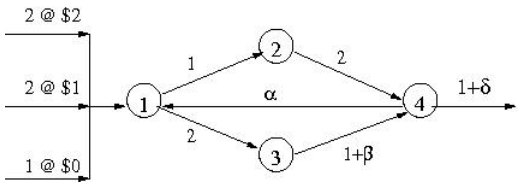
Counterexample. Consider the following network, where the LP is to ship one unit from node 1 to node 4 along the least costly route. An optimal solution is the shortest path, $1 \to 2 \to 4$, with a cost of \$3. There are two parameters, α, β, whose values can create alternative optima. We assume $\alpha \geq -3$ to avoid a negative cycle, and we assume $\beta \geq 0$.

If $\beta = 0$, another shortest path is $1 \to 3 \to 4$. The two shortest paths correspond to two basic optima in the LP formulation, consistent with the myth. However, when $\alpha = -3$, we have a zero-cost cycle: $1 \to 2 \to 4 \to 1$. Any solution can be augmented by an arbitrary amount of flow around the cycle without changing the total cost.



The essence of the myth rings true — there are two *simple* paths corresponding to two basic optima. However, the alternative optima with positive flow around the cycle spoils the result being literally true. One must consider zero-cost cycles as a caveat in how the statement is worded. The issue runs deeper in separating true alternative optima from *frivolous* ones. In particular, the dual always has alternative optima of the form $\pi' = \pi + K$, where π is any dual solution and $K > 0$. This is frivolous because they do not convey any true alternatives in the underlying economics.

To illustrate the difference between true versus frivolous alternative dual optima, consider a 3-tier supply, shown on the right. The dual price at node 4 depends on the demand parameter $\delta \geq 0$.



For $\delta = 0$, the initial supply step can be basic, giving a basic dual price of $\pi_4 = 3$ (and $\pi_1 = 0$). Another basic optimum has the initial supply step out of the basis at its upper bound of one unit, and the second supply step is in the basis (at zero level), giving $\pi_1 = 1$. The price at node 4 then becomes $\pi_4 = 4$. We have another interval of optimal prices at $\delta = 2$. Optimal dual prices are never unique, but when $\delta \neq 0, 2, 4$, alternatives are frivolous in that we could simply add any constant to all of them to obtain an alternative optimum. That notion of "alternative" does not correspond to a real alternative; it is an artifact of the modeling.

To summarize, we have the following cases (for $\alpha \geq -3$, $\beta \geq 0$, $\delta \geq 0$):

	Primal	Dual
unique	$\alpha > -3$, $\beta > 0$	never
frivolous	$\alpha = -3$, $\beta > 0$	$\delta \neq 0, 2, 4$
true alternatives	$\alpha > -3$, $\beta = 0$	$\delta = 0, 2, 4$

LP Myth 17. *In a standard assignment problem, it is always optimal to assign the person-to-job that has the least cost.*

If this were true, we would have a greedy algorithm that recursively assigns the pair of least cost among unassigned pairs. As illustrated with the following counterexample, the optimality of an assignment depends upon *relative* costs. The one with least cost may eliminate an alternative savings that is greater when considering second-least costs.

Counterexample.

1	2
10	15

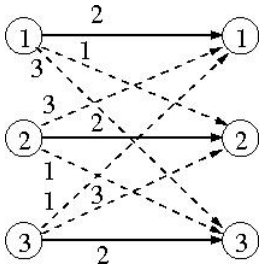
This is a 2×2 problem, and the issue is whether to assign Person 1 to Job 1 since that is the least cost.

If we assign Person 1 to Job 1, that cost is only 1, but we must then assign Person 2 to Job 2. That yields a total cost of 16. The optimal assignment is to assign Person 1 to Job 2 and Person 2 to Job 1, for a total cost of 12.

LP Myth 18. *Given an assignment problem with a non-optimal (but feasible) assignment, its cost can be reduced by swapping some pair of assignments.*

The following counterexample is adapted from Bertsekas[8].

Counterexample. There are 3 people to be assigned to 3 jobs. The current assignment is shown below with the solid arcs, having total cost = 6.



Numbers next to arcs are costs.

Here are the possible pair-wise swaps:

Old	New	Δcost
{1-1, 2-2}	{1-2, 2-1}	0
{1-1, 3-3}	{1-3, 3-1}	0
{2-2, 3-3}	{2-3, 3-2}	0

Every pair of swaps leaves the cost unchanged, but an optimal assignment is {1-2, 2-3, 3-1}, having total cost = 3.

LP Myth 19. *A transportation problem with unique shipping costs has a uniquely optimal shipment.*

Counterexample. Rubin and Wagner[44] pointed this out after noticing that managers apply this myth in practice. They provided the following:

	Supplier 1	Supplier 2	Demand
Market 1	55 0 10	5 10 0	10
Market 2	65 5 5	15 10 10	15
Market 3	75 10 0	25 0 10	10
Supply	20	20	

The upper number in each cell is the unit shipping cost. For example, each unit shipped from Supplier 1 to Market 1 is $55. The lower-left number is the shipment in one optimal solution, and the lower-right number is the shipment in another optimal solution.

Note that the unit costs are all different, yet there are alternative optimal shipments. (The minimum total cost is $1,275.)

LP Myth 20. *The optimal dual price of a demand constraint equals the increase in the minimum total cost if that demand is increased one unit.*

This fails if the solution is not at a *compatible basis*[30] (in the presence of primal degeneracy).

Counterexample. The following is taken from Rubin and Wagner[44].

	Supplier 1	Supplier 2	Demand	Price
Market 1	55 10	10 $0^{\ddagger}$	10	$55^{\dagger}$,$55^{\ddagger}$
Market 2	65 $0^{\dagger}$	15 10	10	$65^{\dagger}$,$60^{\ddagger}$
Market 3	80 0	25 10	10	$75^{\dagger}$,$70^{\ddagger}$
Supply	20	20		†Basis 1
Price	0, 0	50, 45		‡Basis 2

The cell values are unit costs and the (unique) optimal shipment levels. Two (basic) dual prices are shown.

If Market 2 demand increases, the first basis is compatible, and the change in the minimum total cost is indeed $65. This can be achieved by sending one unit from Supplier 1 (which has excess). The basis is compatible with this change because the shipment level, x_{12}, can

increase from its degenerate basic value, 0. On the other hand, if the solver obtains Basis 2, the \$60 dual price understates the increase in minimum total cost.

However, if we want to know the rate of savings from decreasing the demand in Market 2, we obtain the minimum optimal dual price (among the alternative optima) of the demand constraint. It is given by Basis 2 by letting the basic shipment level, x_{21}, increase by 1, balanced by decreasing x_{11} and x_{22} to 9.

The importance of using the wrong dual price for a marginal demand change is that the computed change in the minimum total cost may not be correct. One must have the maximum dual price to compute the effect of a demand increase, and one must have the minimum dual price to compute the effect of a demand decrease. (More details are in [28].)

For non-network LPs the myth can fail by having the correct slope (that is, $\partial f^*(b)/\partial b_i = \pi_i$), but the slope changes at $\Delta b_i < 1$, so the effect of a full unit change cannot be measured precisely with the shadow price.

LP Myth 21. *An increase in a demand requirement (with concomitant increase in supply) increases the minimum total cost.*

This is called the "more-for-less paradox." The following transportation problem is from Charnes and Klingman[17] (also see [49]).

Counterexample. There are 3 suppliers, with supplies shown in the last column, and 4 destinations, with demands shown in the last row. The cell values are optimal flows (blank is zero) and the boxed cell values in the NW corner are costs. The modified problem is to increase demand 1 and supply 2 by 9 units. The new optimal flow is shown on the right, and the total cost has decreased from \$152 to \$143, despite the increase in total flow, from 55 to 64.

[1] 11	[6]	[3] 7	[5] 2	20
[7]	[3]	[1] 10	[6]	10
[9]	[4] 13	[5]	[4] 12	25
11	13	17	14	55

Original Problem
Min Cost = \$152

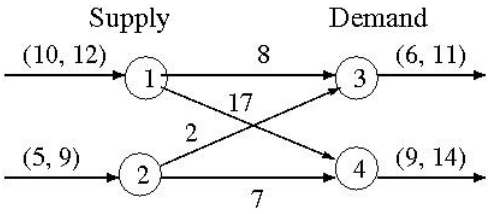
[1] 20	[6]	[3]	[5]	20
[7]	[3] 2	[1] 17	[6]	19
[9]	[4] 11	[5]	[4] 14	25
20	13	17	14	64

Modified Problem
Min Cost = \$143

The underlying economics is that the greater flow can take advantage of low-cost activities. In the transportation example, shipments from supplier 1 to destination 1 have the lowest cost, but the original demand is not enough to ship all of the availability supply; supplier 1 must ship to other destinations. In the revised problem, supplier 1 can ship all of its units to destination 1, and the other destinations can meet their requirements from other suppliers less expensively.

Dĭneko, B. Klinz, and G. J. Woeginger[18] provide the following 3×3 transportation problem: supply: $s = (0, 1, 1)$, demand: $d = (1, 1, 0)$, and cost: $c_{ij} = 2^{|i-j|}$. The minimum total cost is 4. Increasing the first supply and last demand to $s' = d' = (1, 1, 1)$, the minimum total cost is only 3. They proceed to develop a key condition under which this paradox cannot occur: *there does not exist* i, j, p, q *such that* $c_{ij} + c_{pq} < c_{iq}$. If this condition does not hold, the more-for-less paradox may apply, depending on the data.

Glover[9, p. 37] gives another example:



The supplies and demands are required ranges, and the arc numbers are unit flow costs.

The minimum feasible flow is 15 units, and the least costly way to send that minimum is $x_{13} = 6$, $x_{14} = 4$, and $x_{24} = 5$, for a total cost of \$151. However, we can ship $x_{13} = 10$ and $x_{24} = 9$, for a total cost of \$143. We thus ship more for less!

Another form of the more-for-less paradox also arises with modeling requirement constraints as equations, rather than with inequalities. The problem need not be a network.

Counterexample. The following is a diet problem with 3 foods and 2 nutrient requirements, given by Arsham[4, 1]:

$$\begin{gathered} \min 40x_1 + 100x_2 + 150x_3 : \\ x_1 + 2x_2 + 2x_3 = 10 \\ 3x_1 + x_2 + 2x_3 = 20 \\ x_1, x_2, x_3 \geq 0. \end{gathered}$$

The optimal diet is $x = (6, 2, 0)$ with a minimum total cost of \$440. If we increase the second nutrient requirement to 30, the optimal diet becomes $x = (10, 0, 0)$ with a minimum total cost of \$400.

The diet problem usually has the canonical form:

$$\min\ cx :\ Ax \geq b,\ x \geq 0$$

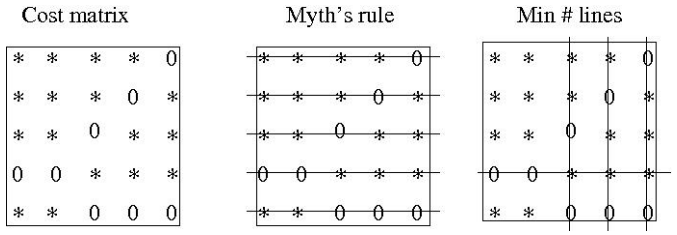
(perhaps with bounds on the levels of foods, as $L \leq x \leq U$). To require $Ax = b$ does not give the flexibility of allowing over-satisfaction of nutrient requirements, even though it could be quite healthy to do so. This principle carries over to other situations, where modeling with equations is not the appropriate representation. (Also see Charnes, Duffuaa, and Ryan[14].)

Arsham[3] provides another vantage, with some focus on production problems.

LP Myth 22. new *The line-drawing step of the Hungarian method for the assignment problem can be replaced by:* cover as many zeroes as possible with each line.

There have been several variants of the Hungarian algorithm — see Kuhn[35]. The original Hungarian method is to cover the zeroes with a minimum number of lines. This myth suggests another criterion, which turns out not to guarantee an optimal solution.

Counterexample. Storøy and Sørevik[52] provide the following 5×5 (* denotes non-zero):



The line-drawing rule starts by covering the three zeroes in row 5, followed by covering the two zeroes in row 4. Thus, a total of five lines must be drawn to cover all zeroes. Since this equals the number of rows (and columns), the Hungarian method's next step is to create an optimal solution from the covered zeroes. This is not possible.

The minimum number of lines is four, and the Hungarian method continues to subtract the minimum uncovered element (adding it to those covered by two lines).

LP Myth 23. *The Stepping Stone Method always produces an optimal distribution.*

This clever, early algorithm by Charnes and Cooper[13] specifically requires equality constraints (with total supply equal to total demand). It was extended to the general *node-bounded problem* by Charnes and Klingman[16]:

$$\begin{aligned} &\min \textstyle\sum_{i,j} c_{ij}x_{ij} : x \geq 0 \\ &\underline{s}_i \leq \textstyle\sum_j x_{ij} \leq \overline{s}_i, \ \forall i \\ &\underline{d}_j \leq \textstyle\sum_i x_{ij} \geq \overline{d}_j, \ \forall j, \end{aligned}$$

where $0 \leq \underline{s} \leq \overline{s}$ (supply out-flow bounds) and $0 \leq \underline{d} \leq \overline{d}$ (demand in-flow bounds).

Charnes, Glover, and Klingman[15] illustrated that the Stepping Stone Method need not terminate with an optimal solution if the constraints are the following special case of the node-bounded problem:

$$\min \textstyle\sum_{i,j} c_{ij}x_{ij} : x \geq 0, \ \sum_j x_{ij} \geq a_i, \ \sum_i x_{ij} \geq b_j.$$

Counterexample. Charnes, Glover, and Klingman gave a counterexample for each case:

$\sum_i a_i = \sum_j b_j$

1	6	3	5	20
7	3	1	6	10
8	3	4	3	25
11	13	17	14	

$\sum_i a_i < \sum_j b_j$

2	4	3
1	1	1
2	5	1
3	4	

$\sum_i a_i > \sum_j b_j$

1	1	2	5
6	5	1	6
2	7	1	

Each table gives the data in the form:

c_{11}	$\dots$	c_{1n}	a_1
$\vdots$		$\vdots$	$\vdots$
c_{m1}	$\dots$	c_{mn}	a_m
b_1	$\dots$	b_m	

The solutions given by the Stepping Stone Method are the associated x_{ij}:

$\sum_{i,j} c_{ij}x_{ij} = 127$

11	0	9	0
0	2	8	0
0	11	0	14

$\sum_{i,j} c_{ij}x_{ij} = 13$

2	1
0	3
1	0

$\sum_{i,j} c_{ij}x_{ij} = 27$

2	3	0
0	4	2

Here are feasible solutions with lower costs:

$\sum_{i,j} c_{ij}x_{ij} = 118$

20	0	0	0
0	2	17	0
0	11	0	14

$\sum_{i,j} c_{ij}x_{ij} = 12$

3	0
0	4
1	0

$\sum_{i,j} c_{ij}x_{ij} = 15$

2	7	0
0	0	6

LP Myth 24. new *The standard free-float formula for an activity in an activity-on-arc network equals the maximum leeway for scheduling the activity without affecting any the earliest start time of any later activity.*

The standard formula for the free float (FF) activity (i, j) is:

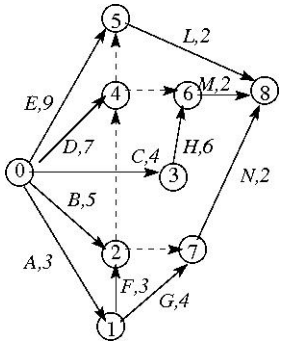
$$FF_{ij} = ES_j - EC_i \tag{LP.1}$$

where ES = earliest start time, EC = earliest completion time.

The statement is true in the absence of dummy arcs, but it can be an underestimate when all successors of some activity in the activity-on-arc network are dummy arcs.

Counterexample. Zhao and Tseng[53] provide the following (numbers on arcs are activity durations):

Activity	(i, j)	incorrect FF_{ij}	correct FF_{ij}
B	$(0, 2)$	1	2
D	$(0, 4)$	0	2
F	$(1, 2)$	0	1



The incorrect values are from (LP.1). For example, $FF_{02} = ES_2 - (ES_0 + 5) = 6 - (0 + 5) = 1$. The maximum leeway, however, is 2. If we delay starting activity B by 2 time units,

that will delay reaching node 2 by 2 time units. But since all arcs out of node 2 are dummy arcs, no activity is immediately affected. Instead, the float limit of 2 comes from tracing the paths out of node 2. Path $2 \to 7 \to 8$ gives a limit of 2 time units — that is, increasing the start of activity B by t delays the start of activity N by $t-2$ for $t \geq 2$. Similarly, the path $2 \to 4 \to 5 \to 8$ reveals that the start of activity L will be delayed by $t-9$, and the path $2 \to 4 \to 6 \to 8$ reveals that the start of activity M will be delayed by $t-10$. The binding limit is from the first path, which yields the correct float value of 2.

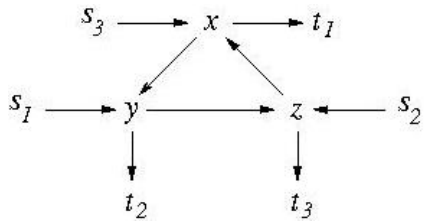
Similarly, applying (LP.1) to arc $(0,4)$, we have the incorrect value: $FF_{04} = ES_4 - (ES_0 + 7) = 7 - (0+7) = 0$. The correct value is obtained by tracing the paths $4 \to 5 \to 8$ and $4 \to 6 \to 8$. The former path yields a float limit of 2 time units (since activity L earliest start time $= ES_5 = 9$); the latter yields a float limit of 3 time unit (since activity M earliest start time $= ES_6 = 10$). The least of these limits is 2, which is the correct float value.

Zhao and Tseng developed this into an algorithm that follows dummy arcs from a rooted tree to obtain the correct free float values.

LP Myth 25. *The maximum flow of commodities through a network equals the capacity of a minimum disconnecting set.*

This is correct when there is only one commodity and for special cases of more than one. The failure for general numbers of commodities on networks of arbitrary topology was recognized in the 1950's — see Zullo[55] and her bibliography through 1995. The following example is from Ford and Fulkerson[19], and is further discussed by Bellmore, Greenberg, and Jarvis[6].

Counterexample. In the following network, all capacities are 3.



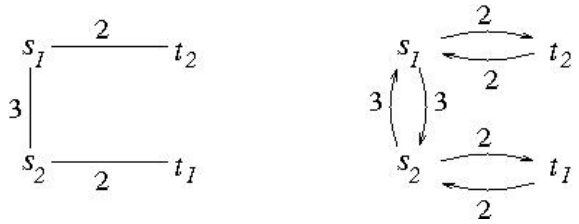
The max-flow is to send $^3/_2$ units along each path from its source to its sink, for a total of $^9/_2$ units. Here are the (unique) paths for each commodity: $s_1 \to y \to z \to x \to t_1$; $s_2 \to z \to x \to y \to t_2$; $s_3 \to x \to y \to z \to t_3$.

The minimum disconnecting is just to break the cycle, say with arc $(x,\ y)$, and the supply arc for the one remaining commodity, which is $(s_1,\ y)$, for a total of 6 units of capacity. There is no 1-arc disconnecting set, so this is a minimum, which implies max-flow $<$ min-cut.

LP Myth 26. *Undirected arcs can be replaced by a pair of oppositely oriented arcs, and there is no loss in generality in obtaining a max-flow or a min-cut.*

This is true for a single-commodity network[19], but it generally fails for multi-commodity networks. The following is given by Bellmore, Greenberg, and Jarvis[6].

Counterexample. In the following network, capacities are shown next to each edge.



In the undirected graph, the max-flow is only 3, sending $^3/_2$ units of each commodity (the min-cut is also 3). After the replacement of each edge with opposite arcs, the max-flow becomes 4 units (also the min-cut value).

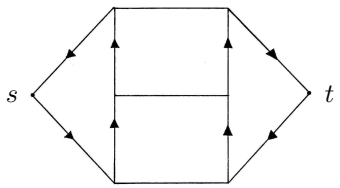
(Note: for a single commodity there is no advantage to sending flow across both arcs since they would cancel out in computing the total flow.)

LP Myth 27. *The maximum two-way flow of a commodity through a network equals its min-cut.*

In this variation of capacitated network flow, some links may be directed (arcs) and some may be undirected (edges). The flow on edges may be in either direction. Two-way flow from node s to node t, denoted $s \leftrightarrow t$, means two paths, one from s to t, denoted $s \to t$, and one from t to s, denoted $t \to s$. A two-way flow is a pair of paths, one in each direction, and the value of the flow is the minimum of all capacities of the links in the paths. A two-way cut for (s, t) is a set of links whose removal removes all paths in both directions, $s \to t$ and $t \to s$.

Rothschild and Whinston[41] provide the following:

Counterexample. In the following network (taken from [41]), all capacities are one. We have: two-way max-flow = 1 < two-way min-cut = 2.



LP Background — Gomory-Hu Cut Trees

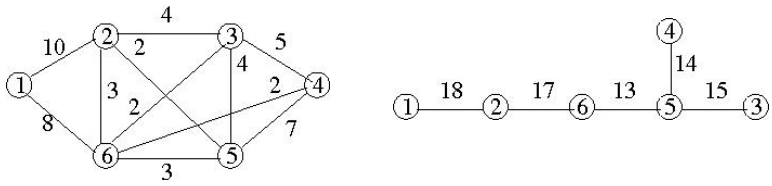
Consider an undirected graph with distinguished nodes s, t. Each edge e has a capacity, c_e, so there is a maximum flow from s to t, which equals the minimum cut that disconnects s from t. The *multi-terminal max-flow/min-cut problem* is to find the max-flow/min-cut between each s, t in the graph. This could be done by solving each of the $\binom{n}{2}$ min-cut problems, but the Gomory-Hu algorithm[26] does this with only $n-1$ min-cut solutions.

Let V_{st} denote the max-flow/min-cut value between s and t. The Gomory-Hu algorithm produces a *cut-tree* (sometimes called a *Gomory-Hu tree*), whose nodes are those of the original graph and whose edges satisfy:

$$V_{st} = \min_{(i,j)\in P_{st}} V_{ij}, \tag{LP.2}$$

where P_{st} = edges in s-t path. The Gomory-Hu algorithm computes the $n-1$ cuts, from which (LP.2) yields all of the $\binom{n}{2}$ min-cut values in the original graph.

Example (taken from [26]):



Capacitated network Gomory-Hu cut-tree

For example, $V_{14} = 13 = \min\{18, 17, 13, 14\}$. The cut set is $\{(2,3), (2,5)\,(6,3), (6,4), (6,5)\}$, with graph partition = $\{1, 2, 6 \,|\, 3, 4, 5\}$.

A cut-tree has two key properties:

1. Each max-flow/min-cut value in the original graph equals the minimum of the edge values along the unique path connecting them in the cut-tree (that is, equation (LP.2)).
2. Removal of any edge from the cut-tree partitions the original graph into two sets of nodes that comprise a cut set whose value equals the cut-tree edge value.

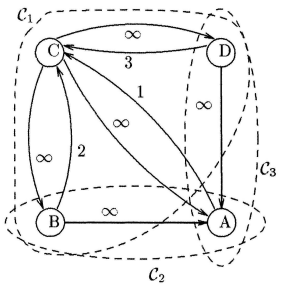
The first property gives the correct value of the min-cut, and hence the max-flow, and the second property gives the actual cut-set for any pair of nodes.

LP Myth 28. *Every connected network has a cut-tree.*

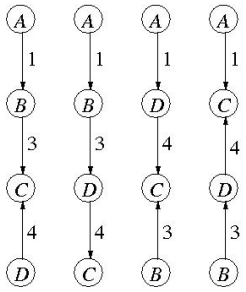
The classical algorithm by Gomory and Hu[26] constructively establishes the existence of a cut-tree for every connected, ***undirected*** graph. This was allegedly extended to directed graphs for the *symmetric case*: the min-cut between two nodes is the lesser of the min-cut from one to the other:

$$\mathcal{V}_{st} = \min\{V_{st}, V_{ts}\}$$

Counterexample. Benczúr[7] provides the following:



Only 3 (of 7) cut-sets are min



Flow trees do not encode min cut-set

Here are the min-cut values:

$$V = \begin{bmatrix} 0 & 1 & 1 & 1 \\ \infty & 0 & 3 & 3 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & 4 & 0 \end{bmatrix} \Rightarrow \mathcal{V} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 3 & 3 \\ 1 & 3 & 0 & 4 \\ 1 & 3 & 4 & 0 \end{bmatrix}.$$

Since the min-cut value of A is 1 and all other min-cut values are greater than 1, any cut-tree must have A as a leaf. That leaves 9 trees to consider. Of these, 4 are shown with the edge values equal to the associated min-cut values: $V(\mathcal{C}_1 = (A \,|\, B, C, D)) = 1$, $V(\mathcal{C}_2 = (A, B \,|\, C, D)) = 3$, and $V(\mathcal{C}_3 = (A, D \,|\, B, C)) = 4$. Each tree violates the second property to be a cut-tree: the cut-set obtained upon breaking an edge of minimum value in the path between two nodes is not their min-cut.

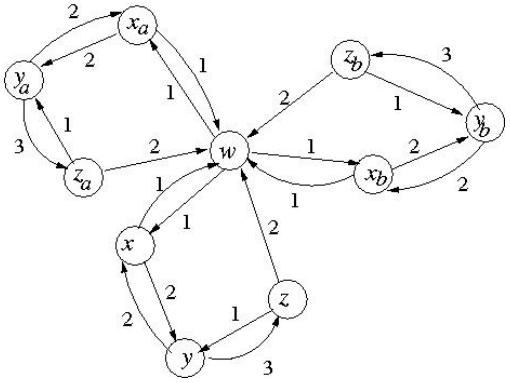
Going from left-to-right, the first two trees' violation is with (D, C). The cut-sets obtained from the edge is $(D \,|\, C, B, A)$ and $(C \,|\, D, B, A)$, respectively, but the min cut-set between D and C is $\mathcal{C}_3$. The third tree's violation is with (B, C). The cut-set obtained from the edge is $(B \,|\, C, D, A)$, but the min cut-set between B and C is $\mathcal{C}_2$. The fourth tree's violation is with (B, D). The cut-set obtained from the edge is $(B \,|\, D, C, A)$, but the min-cut is $\mathcal{C}_2$.

Now consider the other possible trees. Separating C and D makes their path value 3, which is not the value of their min-cut. The four shown are the only ones satisfying the first property of a cut-tree, showing the correct values of the min-cut using equation (LP.2). Since min-cut = max-flow, these are called *flow trees*.

Rizzi[40] provides the following with additional insight.

Counterexample.

In any tree there must be a leaf. Any cut-tree for this network must therefore have a star cut, $(v \,|\, \{u \neq v\})$. Suppose z is a leaf and its neighbor is y. The edge value of (z, y) is the star cut value $V(z \,|\, x, y, w, x_a, \dots) = 3$. If it were a cut-tree, this partition must be the min-cut between z and y. This is not the case, as the min-cut between z and y is $V(z, x \,|\, y, w, x_a, \dots) = 2$.

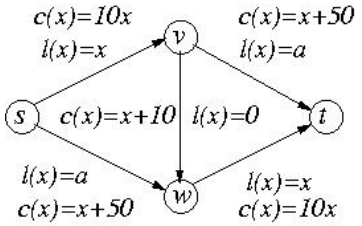


The key property identified by Rizzi is the notion of a *good pair*: (s,t) such that the star cut at t is a min-cut of (s,t), or there is no min-cut of (s,t) (that is, no path $s \to t$ or $t \to s$). Rizzi's network has no good pair, and that is why a cut-tree does not exist.

LP Myth 29. *Removing an arc in a network cannot decrease users' latency or cost.*

This is *Braess' Paradox*[11] applied to traffic flow.

Counterexample. The following is the classical example[12] — also see http://supernet.som.umass.edu/facts/braess.html.



$\ell(x)$ is the latency function of flow, x;
$c(x)$ is the cost.

The equilibrium flow is determined by each driver using the min-latency path. For n users, such that $n < a$, this is $s \to v \to w \to t$. (The users are indifferent among the three paths if $n = a$.) This results in each user experiencing $2n$ units of latency. If we remove arc $(v,\ w)$, the drivers evenly split the use of the two different paths: $s \to v \to t$ and $s \to w \to t$. Their latencies thus reduce to $\frac{1}{2}n + a$ each.

Using the same graph, Steinberg and Zangwill[48] provide the rest of the counterexample, using the cost functions shown. With arc $(v,\ w)$, 6 users evenly split each of the three paths from s to t, so that $x_{sv} = x_{wt} = 4$, while the other arc flows are 2. Thus, each user pays \$92, and the system cost is \$552. Without arc $(v,\ w)$, 6 users split evenly between the two paths. Thus, each user pays \$83, and the system cost is \$498.

A great deal of literature has developed since Braess introduced his paradox in 1968. It has become a cornerstone of traffic equilibrium, as reflected in modern books by Nagurney[36, 39, 37] and Roughgarden[42]. Also see Nagurney[38] and Roughgarden[43] for focus on the Braess paradox and its relatives.

LP Myth 30. *Given strict improvement in the objective value, the standard simplex method does not visit an exponential number of vertices of the feasible polyhedron.*

The falsity of this was first demonstrated by Klee and Minty[34]. The so-called *Klee-Minty polytope* causes the standard simplex method to visit every extreme point, which grows exponentially with the number of variables.

Counterexample. The LP has n variables, n constraints, and 2^n extreme points. The elementary simplex method, starting at $x = 0$, goes through each of the extreme points before reaching the optimum solution at $(0, 0, \ldots, 0, 5^n)$.

$$\begin{array}{rcrcccrcrcr}
\max\ 2^{n-1}x_1 & + & 2^{n-2}x_2 & + & \ldots & + & 2x_{n-1} & + & x_n: & & \\
x_1 & & & & & & & & & \leq & 5 \\
4x_1 & + & x_2 & & & & & & & \leq & 25 \\
8x_1 & + & 4x_2 & + & x_3 & & & & & \leq & 125 \\
\vdots & & & & & & & & & & \vdots \\
2^n x_1 & + & 2^{n-1}x_2 & + & \ldots & + & 4x_{n-1} & + & x_n & \leq & 5^n
\end{array}$$

$$x \geq 0.$$

Another interesting example of exponential growth is due to Blair[9].

Jeroslow[32] was the first to present the construction of a class of examples for the best-gain basis entrance rule to visit an exponential number of vertices. (Also see Blair[9].)

LP Myth 31. *The worst-case time complexity of the simplex method is exponential and hence worse than the worst-case time complexity of the interior-point method.*

There are several things wrong with this statement. The first thing to note is that there is no "***the*** simplex method" and there is no "***the*** interior-point method." We know that both the standard simplex method and the best-gain rule have exponential time complexity (see LP Myth 30). However, the *Hirsch Conjecture*[54] leaves open the prospect for some simplex method to be linear in the numbers of variables and constraints. Also, there are interior-point methods that behave better than Karmarkar's original[33] in practice, but have no proof of polynomial complexity.

The second thing to note is the perturbation analysis by Spielman and Teng[47]. In fact, many coefficients are subjected to "random" perturbation due to rounding in their computations from other data.

Now suppose we are talking about the standard simplex method and one of the interior-point methods with a proof of polynomial complexity in the length of the data. Then, the third thing to consider is that the length of the data could be an exponential function of the number of variables. One example of this is a *Linear Programming Relaxation* (LPR) whose coefficients are computed from an aggregation algorithm[24]. The length of the coefficients (number of digits) can be an exponential function of the numbers of variables and constraints.

Thus, one must be careful in how to compare the (theoretical) worst-case time complexities of simplex versus interior methods.

LP Myth 32. **new** *The c-diameter of a non-empty polytope of dimension d with f facets cannot exceed $f - d$.*

Let P denote the polytope (that is, bounded polyhedron), and let $V^*(P, c)$ denote the set of vertices that minimize a linear form, cx, on P. The *c-diameter* from a vertex $v \in P$ for a given linear form is defined as the maximum distance from v to $V^*(P, c)$. The *distance* is defined to be the minimum number of edges in a path joining v to $V^*(P, c)$ along which cx is non-increasing. (In terms of LP, the c-diameter is an upper bound on how many vertices the simplex method visits before reaching an optimal vertex.) Denote the c-diameter from v

by $\Delta(v,c)$, and the myth asserts $\Delta(v,c) \le f - d$. This is known as the *monotonic bounded Hirsch conjecture*.

Counterexample. Todd[50] provides the following:

$$P = \{x \in \mathbb{R}^4_+ : Ax \le b\}, \text{ where } A = \begin{bmatrix} 7 & 4 & 1 & 0 \\ 4 & 7 & 0 & 1 \\ 43 & 53 & 2 & 5 \\ 53 & 43 & 5 & 2 \end{bmatrix}, \; b = \begin{pmatrix} 1 \\ 1 \\ 8 \\ 8 \end{pmatrix}.$$

This is a 4-dimensional polytope with 8 facets, so the myth asserts that the c-diameter cannot exceed 4 for any linear form. Let $c = (1,1,1,1)$, so $V^*(P,c) = \{(0,0,0,0)^\intercal\}$. Todd proves that all non-increasing paths from $v = \frac{1}{19}(1,1,8,8)^\intercal$ to 0 have a distance of 5.

LP Myth 33. *In employing successive bound reduction in a presolve, we can fix a variable when its bounds are within a small tolerance of each other.*

The myth is that we can pick a tolerance, say $\tau > 0$, such that if we infer $L \le x \le U$ and $U - L \le \tau$, we can fix x to some value in the interval, such as the midpoint, $\frac{1}{2}(L+U)$. There are a few things wrong with this, as reported by Greenberg[27].

Counterexample. Consider $x \ge 0$ and

$$\begin{array}{rcrcl} \frac{1}{2}x_1 & + & x_2 & = & 1 \\ x_1 & + & x_2 & = & 2. \end{array}$$

This has the unique solution, $x = (2,0)$, and it is this uniqueness that causes a problem with greater implications.

In successive bound reduction, the most elementary tests evaluate rows to see if just one row alone can tighten a bound on a variable. Initially, the bounds are the original ones: $L^0 = L = (0,0)$ and $U^0 = U = (\infty,\infty)$. The first iteration results in the inference that $x_1 \le 2$, from the first equation and the fact that $x_2 \ge 0$. It similarly produces an upper bound, $x_2 \le 1$, so $U^1 = (2,1)$. Still in iteration 1, the second equation causes the inference, $x_1 \ge 1$, because we already have $x_2 \le 1$ when we get there. Thus, $L^1 = (1,0)$.

At a general iteration, we will have inferred $L^k_1 \le x_1 \le 2$ and $0 \le x_2 \le U^k_2$, where $L^k_1 < 2$ and $U^k_2 > 0$. At the end of iteration k, the inferred bounds are:

$$2 - (\tfrac{1}{2})^k \le x_1 \le 2 \;\text{ and }\; 0 \le x_2 \le (\tfrac{1}{2})^k.$$

This converges to the unique solution, but it does not reach it finitely. If the iterations go far enough, the bounds become within the tolerance $\tau > 0$. At that point, suppose x is fixed to the interval's midpoint: $x = \frac{1}{2}(L^k + U^k)$.

To see a consequence of this, suppose that the presolve tests feasibility with another tolerance, μ. Let the constraints be of the form $Ax = b$. The rule is: *Declare infeasibility if, for some equation, i,*

$$y_i^{\max} = \max_{L^k \le x \le U^k} A_{i\bullet}x < b_i - \mu \quad \textit{or}$$
$$y_i^{\min} = \min_{L^k \le x \le U^k} A_{i\bullet}x > b_i + \mu.$$

In our example, when $k = \lceil -\log_2 \tau \rceil$, both variables are fixed:

$$x_1 = 2 - (\tfrac{1}{2})^{k+1}\,,\; x_2 = (\tfrac{1}{2})^{k+1}\,.$$

Equation 2 passes the feasibility test, but equation 1 has

$$y_1^{\max} = y_1^{\min} = 1 - (\tfrac{1}{2})^{k+2} + (\tfrac{1}{2})^{k+1} = 1 + (\tfrac{1}{2})^{k+2}\,.$$

Thus, $y_1^{\min} = 1 + (\frac{1}{2})^{k+2}$, so we declare infeasibility if $(\frac{1}{2})^{k+2} > \mu$. Taking logs, this is equivalent to $-(k+2) > \log_2 \mu$. Replacing k, we have that a false infeasibility is declared if

$$-\lceil -\log_2 \tau \rceil - 2 > \log_2 \mu.$$

For example, if $\tau = 2^{-20}$, we declare a *false infeasibility* if $\mu < 2^{-22}$.

This example highlights two things:

1. Tolerances are related. The tolerance to fix a variable should not be substantially less than the infeasibility tolerance.
2. Fix a variable judiciously. When having inferred $x_j \in [L_j, U_j]$, such that $U_j - L_j$ is within tolerance of fixing x_j, do so in the following order of choice:
   (1) If L_j is an original bound, fix $x_j = L_j$;
   (2) If U_j is an original bound, fix $x_j = U_j$;
   (3) If $[L_j, U_j]$ contains an integer, p, fix $x_j = p$;
   (4) If all of the above fail, fix $x_j = \frac{1}{2}(L_j + U_j)$.

LP Myth 34. *A factored form of the basis contains less error for FTRAN after reinversion.*

The Forward Transformation (FTRAN) algorithm solves the forward system, $Bx = b$, by factoring B and updating it after each basis change. Consider the elementary product form: $B = E_1 E_2 \cdots E_k$, where each E_i is an elementary matrix.

Algorithm: Forward Transformation with PFI
Initialize. Set $x^0 = b$.
for $i = 1 : k$ **do**
  Solve $E_i x^i = x^{i-1}$
end for
Exit with x^k the (computed) solution to $Bx = b$.

During the pivoting process, k increases and there are more factors than needed. *Reinversion* is the process of restarting to obtain the minimum number of factors, which equals the number of variables in the basis (except slacks). One reason to reinvert is to "cleanup" the errors that accumulate, which affects the accuracy of solving $Bx_B = b$. (Another reason is to reduce the FTRAN time.)

The essence of the counterexample is cancelation of errors in the first factors that does not cancel in the reinverted factorization.

Counterexample. Consider the 2×3 system:

$$\begin{array}{ccccccc} a_{11}x_1 & + & a_{12}x_2 & + & a_{13}x_3 & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & a_{23}x_3 & = & b_2 \end{array}$$

Pivoting x_1 on equation 1, then x_2 on equation 2 into the basis, then replacing x_1 with x_3 yields the following elementary factors:

$$E_1 = \begin{bmatrix} a_{11} & 0 \\ a_{21} & 1 \end{bmatrix};\ E_2 = \begin{bmatrix} 0 & a_{12}/a_{11} \\ 1 & a_{22} - a_{21}a_{12}/a_{11} \end{bmatrix}$$

$$E_3 = \begin{bmatrix} a_{13}/a_{11} - a_{22} - a_{12}a)21/a_{11})((a_{23} - a_{13}a)21/a_{11})/a_{12}/a_{11} & 0 \\ \big(a_{23} - a_{13}a_{21}/a_{11}\big)/(a_{12}/a_{11}) & 1 \end{bmatrix}.$$

Collecting computed values and substituting c with a new index whenever there is a new computation, we obtain:

$$E_1 = \begin{bmatrix} a_{11} & 0 \\ a_{21} & 1 \end{bmatrix};\ E_2 = \begin{bmatrix} 0 & c_1 \\ 1 & c_2 \end{bmatrix};\ E_3 = \begin{bmatrix} c_3 & 0 \\ c_4 & 1 \end{bmatrix}.$$

Then, executing FTRAN for b (to get basic levels):

$$\begin{aligned} x^1 &= \begin{pmatrix} b_1/a_{11} \\ b_2 - (b_1/a_{11})a_{12} \end{pmatrix} = \begin{pmatrix} c_5 \\ c_6 \end{pmatrix} \\ x^2 &= \begin{pmatrix} x_1^1 - (x_2^1/c_2)c_1 \\ x_2^1/c_2 \end{pmatrix} = \begin{pmatrix} c_7 \\ c_8 \end{pmatrix} \\ x^3 &= \begin{pmatrix} x_1^2/c_3 \\ x_2^2 - (x_1^2/c_3)c_4 \end{pmatrix} = \begin{pmatrix} c_9 \\ c_{10} \end{pmatrix} \end{aligned}$$

After reinversion, the elementary matrices have the form:

$$E_1 = \begin{bmatrix} a_{13} & 0 \\ a_{23} & 1 \end{bmatrix};\ E_2 = \begin{bmatrix} 0 & c_{11} \\ 1 & c_{12} \end{bmatrix}.$$

Now the FTRAN algorithm yields computed levels:

$$\widehat{B^{-1}b} = \begin{pmatrix} c_{13} \\ c_{14} \end{pmatrix}.$$

Suppose $\beta = B^{-1}b$, the true value of the levels. The issue is whether

$$\left|\left|\begin{pmatrix} \beta_1 - c_{13} \\ \beta_2 - c_{14} \end{pmatrix}\right|\right| = ||\beta - \zeta'|| < \left|\left|\begin{pmatrix} \beta_1 - c_9 \\ \beta_2 - c_{10} \end{pmatrix}\right|\right| = ||\beta - \zeta||\,,$$

where ζ the accumulated error before reinversion, and ζ' is the accumulated error after reinversion.

It is possible that $\zeta = 0$ while $\zeta' \neq 0$ — that is, that we obtain an error-free solution with the original factorization and reinversion introduces error. This can happen by error cancelation. However, even if $||\zeta'|| < ||\zeta||$, the computed levels could have less error, at least for some particular b. For example, let $\beta = (100, 100)^\intercal$, $\zeta = (2, 2)^\intercal$, and $\zeta' = (1, -1)^\intercal$. Then, $||\zeta|| > ||\zeta'||$, yet $||\beta - \zeta|| \approxeq 138.6 < 141.4 \approxeq ||\beta - \zeta'||$.

LP References

[1] H. Arsham. On the more-for-less paradoxical situations in linear programs: A parametric optimization approach. *Journal of Information & Optimization Sciences*, 17(3):485–500, 1996.

[2] H. Arsham. Tools for modeling validation process: The dark side of LP. Web note, Merrick School of Business, University of Baltimore, http://home.ubalt.edu/ntsbarsh/opre640A/partv.htm, 1996–2008.

[3] H. Arsham. Construction of the largest sensitivity region for general linear programs. *Applied Mathematics and Computation*, 189(2):1435–1447, 2007.

[4] H. Arsham and M. Oblak. Perturbation analysis of general LP models: A unified approach to sensitivity, parametric, tolerance, and more-for-less analysis. *Mathematical and Computer Modelling*, 13(8):79–102, 1990.

[5] E. M. L. Beale. Cycling in the dual simplex method. *Naval Research Logistics Quarterly*, 2(4), 1955.

[6] M. Bellmore, H. J. Greenberg, and J. J. Jarvis. Multi-commodity disconnecting sets. *Management Science*, 16(6):427–433, 1970.

[7] A. A. Benczúr. Counterexamples for directed and node capacitated cut-trees. *SIAM Journal on Computing*, 24(3):505–510, 1995.

[8] D. P. Bertsekas. *Linear Network Optimization: Algorithms and Codes*. MIT Press, Cambridge, MA, 1991.

[9] C. Blair. Some linear programs requiring many pivots. Technical Report 867, College of Commerce and Business Administration, University of Illinois, Champaign, IL, 1982. Available at https://netfiles.uiuc.edu/c-blair/www/papers/papers.html.

[10] C. Blair. Random linear programs with many variables and few constraints. *Mathematical Programming*, 34(1):62–71, 1986.

[11] D. Braess. Über ein paradoxon aus der verkehrsplanung. *Unternehmensforschung*, 12:258–268, 1968. English translation in [12]

[12] D. Braess, A. Nagurney, and T. Wakolbinger. On a paradox of traffic planning. *Transportation Science*, 39(4):446–450, 2005. English translation of [11]

[13] A. Charnes and W. W. Cooper. The stepping stone method of explaining linear programming calculations in transportation problems. *Management Science*, 1(1):49–69, 1954.

[14] A. Charnes, S. Duffuaa, and M. Ryan. Degeneracy and the more-for-less paradox. *Journal of Information & Optimization Sciences*, 1(1):52–56, 1980.

[15] A. Charnes, F. Glover, and D. Klingman. A note on a distribution problem. *Operations Research*, 18(6):1213–1216, 1970.

[16] A. Charnes and D. Klingman. The distribution problem with upper and lower bounds on the node requirements. *Management Science*, 16(9):638–642, 1970.

[17] A. Charnes and D. Klingman. The "more for less" paradox in the distribution model. *Cahiers du Centre d'Etudes de Recherche Opérationnelle*, 13(1):11–22, 1971.

[18] V. G. Dĭneko, B. Klinz, and G. J. Woeginger. Which matrices are immune against the transportation paradox. *Discrete Applied Mathematics*, 130(3):495–501, 2003.

[19] L. Ford and D. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, NJ, 1963.

[20] S. I. Gass. *Linear Programming: Methods and Applications*. McGraw-Hill, New York, NY, fifth edition, 1975.

[21] S. I. Gass and S. Vinjamuri. Cycling in linear programming problems. *Computers & Operations Research*, 31(2):303–311, 2004.

[22] B. J. Gassner. Cycling in the transportation problem. *Naval Research Logistics Quarterly*, 11(1):43–58, 1964.

[23] F. Glover. *Netforms in Management Science and Decision Support.* University of Colorado, School of Business, Boulder, CO, 1983.

[24] F. Glover and D. A. Babayev. New results for aggregating integer-valued equations. *Annals of Operations Research*, 58:227–242, 1995.

[25] A. J. Goldman and D. Kleinman. Examples relating to the simplex method. *Operations Research*, 12(1):159–161, 1964.

[26] R. E. Gomory and T. C. Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):551–570, 1961.

[27] H. J. Greenberg. *A Computer-Assisted Analysis System for Mathematical Programming Models and Solutions: A User's Guide for ANALYZE*, volume 1 of *Operations Research/Computer Science Interfaces Series.* Springer, New York, NY, 1993.

[28] H. J. Greenberg. How to analyze the results of linear programs — Part 2: Price interpretation. *Interfaces*, 23(5):97–114, 1993.

[29] H. J. Greenberg. Consistency, redundancy, and implied equalities in linear systems. *Annals of Mathematics and Artificial Intelligence*, 17:37–83, 1996.

[30] H. J. Greenberg. Chapter 3: Linear programming 1: Basic principles. In T. Gal and H. J. Greenberg, editors, *Advances in Sensitivity Analysis and Parametric Programming.* Kluwer Academic Press, 1997.

[31] A. J. Hoffman. Cycling in the simplex algorithm. Report 2974, National Bureau of Standards, Gaithersburg, MD, 1953.

[32] R. G. Jeroslow. The simplex algorithm with the pivot rule of maximizing criterion improvement. *Discrete Mathematics*, 4(4):367–377, 1973.

[33] N. Karmarkar. A new polynomial time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984.

[34] V. Klee and G. J. Minty. How good is the simplex algorithm? In O. Shisha, editor, *Inequalities III*, pages 159–175, New York, NY, 1972. Academic Press.

[35] H. W. Kuhn. Variants of the Hungarian method for assignment problem. *Naval Research Logistics Quarterly*, 3:253–258, 1956.

[36] A. Nagurney. *Network Economics: A Variational Inequality Approach*, volume 10 of *Advances in Computational Economics.* Springer, Dordrecht, The Netherlands, revised edition, 1999.

[37] A. Nagurney. *Supernetworks: Decision-Making for the Information Age.* Edward Elgar Publishing, Cheltenham, England, 2002.

[38] A. Nagurney. Supernetworks: Paradoxes, challenges, and new opportunities. In W. H. Dutton, B. Kahin, R. O'Callaghan, and A. W. Wyckoff, editors, *Transforming Enterprise*, pages 229–254, Cambridge, MA, 2004. MIT Press.

[39] A. Nagurney. *Supply Chain Network Economics: Dynamics of Prices, Flows And Profits.* New Dimensions in Networks. Edward Elgar Publishing, Cheltenham, England, 2006.

[40] R. Rizzi. Excluding a simple good pair approach to directed cuts. *Graphs and Combinatorics*, 17(4):741–744, 2001.

[41] B. Rothschild and A. Whinston. Maximal two-way flows. *SIAM Journal on Applied Mathematics*, 5(5):1228–1238, 1967.

[42] T. Roughgarden. *Selfish Routing and the Price of Anarchy.* MIT Press, Cambridge, MA, 2005.

[43] T. Roughgarden. On the severity of Braess's paradox: Designing networks for selfish users is hard. *Journal of Computer and System Sciences*, 72(5):922–953, 2006.

[44] D. S. Rubin and H. M. Wagner. Shadow prices: Tips and traps for managers. *Interfaces*, 20(4):150–157, 1990.

[45] T. L. Saaty. A conjecture concerning the smallest bound on the iterations in linear programming. *Operations Research*, 11(1):151–153, 1963.

[46] G. Sierksma and G. A. Tijssen. Degeneracy degrees of constraint collections. *Mathematical Methods of Operations Research*, 57(3):437–448, 2003.

[47] D. A. Spielman and S.H. Teng. Smoothed analysis: why the simplex algorithm usually takes polynomial time. *Journal of the Association for Computing Machinery*, 51(3):385–463, 2004.

[48] R. Steinberg and W. I. Zangwill. The prevalence of braess' paradox. *Transportation Science*, 17(3):301–318, 1983.

[49] W. Szwarc. The transportation paradox. *Naval Research Logistics Quarterly*, 18:185–202, 1971.

[50] M. J. Todd. The monotonic bounded Hirsch conjecture is false for dimension at least 4. *Mathematics of Operations Research*, 5(4):599–601, 1980.

[51] J.A. Tomlin and J.S. Welch. A pathological case in the reduction of linear programs. *Operations Research Letters*, 2(2):53–57, 1983.

[52] S. Storøy and T. Sørevik. A note on an orthogonal systolic design for the assignment problem. *Parallel Computing*, 17(4–5):523–525, 1991.

[53] T. Zhao and C-L. Tseng. A note on activity floats in activity-on-arrow networks. *Journal of the Operational Research Society*, 54(12):1296–1299, 2003.

[54] G. M. Ziegler. *Lectures on Polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer, 1995. See pp. 83–93.

[55] H. Zullo. *Feasibile Flows in Multicommodity Graphs*. Ph.D. thesis, University of Colorado at Denver, Mathematics Department, Denver, CO, 1995. Available at http://www-math.cudenver.edu/graduate/thesis/hzullo.pdf.

Integer Programming and Combinatorial Optimization

The general form of an *Integer Program* (IP) is the optimization (min or max) of a function over a domain such that the variables are required to have integer values. An *Integer Linear Program* (ILP) has the form of LP with $x \in \mathbb{Z}^n$. If only some of the variables must be integer, it is called a *Mixed-Integer Program* (MIP). If it has the form of LP, but with $x_j \in \mathbb{Z}$ for $j \in J \neq \emptyset$, it is a *Mixed-Integer Linear Program* (MILP). The LPR of a MILP is the *Linear Programming Relaxation* of $x_j \in \mathbb{Z}$, allowing non-integer solutions for all variables.

We include combinatorial optimization problems, even those that are not usually modeled with IP.

IP Myth 1. *The integer solution is a rounding of some LPR solution.*

It is possible that every MILP solution could be far from the relaxed solution. In fact, it is possible that no rounding is feasible.

Counterexample. max $21x_1 + 11x_2 : x \geq 0, 7x_1 + 4x_2 \leq 13$.

The relaxed solution is at $\left(\frac{13}{7}, 0\right)$, and the optimal integer solution is at (0, 3).

Glover and Sommer[12] provide more meaningful examples, including a conditional transportation problem. Additional examples and discussion are in Glover[9] and Glover, Klingman and Phillips[11].

IP Myth 2. *If a basic solution of the LPR of a MILP is not integer-valued, no rounding is possible.*

This clever paradox was noted by Glover and Sommer[12] with the following "proof:"

> The only fractional variables, which could be rounded, are basic, but the basic equations, $Bx_B = b$, have a unique solution. Hence, no rounding is possible!

The flaw is the assumption that all non-basic variables must remain fixed at zero (or an upper bound). In particular, slack variables may change to offset the rounding.

IP Myth 3. *The LPR solves its associated ILP if, and only if, it has an optimal basic solution that is integer-valued.*

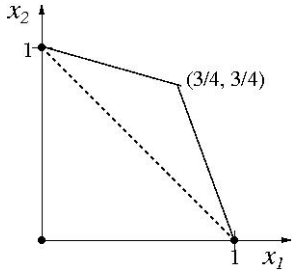
The sufficiency is always true, but the necessity of an integer-valued optimum that is basic applies to binary programs (whose LPR is in standard form) and may not hold otherwise. The following counterexample has an optimality region with non-integer extreme points but an optimal integer point in its interior.

Counterexample. max $0x : 0 \le x \le b\,\boldsymbol{e},\ x \in \mathbb{Z}$, where $b \notin \mathbb{Z}$ and $\boldsymbol{e}$ is a vector of ones. Dropping the integer requirement, the LP solution is any feasible point. For $b > 1$, $\boldsymbol{e}$ is feasible and hence optimal for the LP. Therefore, it is optimal for the ILP, but it is not a basic optimum for the LPR.

IP Myth 4. *The number of extreme points of the integer hull is at least as great as the number of extreme points of the LPR polyhedron.*

Counterexample.

The LPR polyhedron can have regions with no integer points, as illustrated to the right. The integer hull has 3 extreme points, whereas the LPR polyhedron has 4 (and could have any arbitrary number).



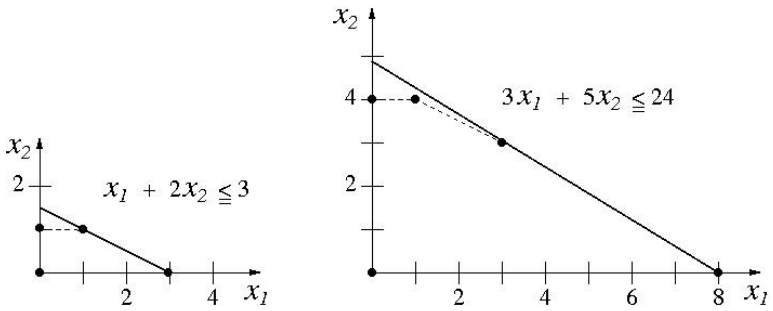
$$3x_1 + x_2 \le 3$$
$$x_1 + 3x_2 \le 3$$
$$x \ge 0$$

The same myth and counterexample applies if "facets" replaces "extreme points."

IP Myth 5. *The number of extreme points of the integer hull is bounded by some multiple of those of the LPR, where the multiple depends upon the number of variables and constraints.*

Rubin[28] provides the following counterexample that shows the integer hull can have any number of extreme points with only one constraint in $\mathbb{R}^2_+$. (Also see Jeroslow[17].)

Counterexample. Define a polytope in 2 variables and 1 constraint plus non-negativity: $P = \{x \in \mathbb{R}^2_+ : a_1x_1 + a_2x_2 \leq b\}$, where $a, b > 0$. This has three extreme points. We can choose a, b such that its integer hull, convh$(P \cap \mathbb{Z}^2)$, has N extreme points for any $N \geq 3$. The following figures show two such polytopes:



(a) $N = 4$ (b) $N = 5$

The following table shows more, and you may note a pattern that Rubin discovered.

N	a_1	a_2	b
4	1	2	3
5	3	5	24
6	8	13	168
7	21	34	1,155
8	55	89	7,920
9	144	233	54,288
10	377	610	372,099

Let F_k be the k^{th} Fibbonacci number, and $P = \{x \in \mathbb{R}^2_+ : F_{2k}x_1 + F_{2k+1}x_2 \leq F^2_{2k+1} - 1\}$. Then, convh$(P \cap \mathbb{Z}^2)$ has $k + 3$ extreme points.

Rubin gives other ways to generate the polytope for one constraint in $\mathbb{R}^2_+$ such that it has any number of extreme points.

The same myth and counterexample applies if "facets" replaces "extreme points."

IP Myth 6. new *Every integral vector of an n-dimensional integral polyhedral pointed cone C can be expressed as a non-negative integral combination of at most n elements of the Hilbert basis of C.*

It it were true, this would be an extension of Carathéodory's theorem. Let $z^1, \ldots, z^k \in \mathbb{Z}^n$ be *generators* of

$$C = \{z : z = \textstyle\sum_{i=1}^{k} \lambda_i z^i \text{ for some } \lambda \in \mathbb{Z}^k_+\}.$$

Counterexample. Bruns et al.[1] provide the following:

$$\begin{aligned} z^1 &= (0,1,0,0,0,0), & z^6 &= (1,0,2,1,1,2),\\ z^2 &= (0,0,1,0,0,0), & z^7 &= (1,2,0,2,1,1),\\ z^3 &= (0,0,0,1,0,0), & z^8 &= (1,1,2,0,2,1),\\ z^4 &= (0,0,0,0,1,0), & z^9 &= (1,1,1,2,0,2),\\ z^5 &= (0,0,0,0,0,1), & z^{10} &= (1,2,1,1,2,0). \end{aligned}$$

The generators form a Hilbert basis for C, and the myth asserts that every integral vector in C is a conical combination of only 6 of the 10 generators. Consider

$$g = (9, 13, 13, 13, 13, 13) = z^1 + 3z^2 + 5z^4 + 2z^5 + z^8 + 5z^9 + 3z^{10}.$$

A minimum number of generators can be obtained by the ILP:

$$\min \textstyle\sum_{i=1}^{10} u_i : \sum_{i=1}^{10} \lambda_i z^i = g,$$
$$u_i \in \{0,\, 1\},\ 0 \le \lambda_i \le 13u_i,\ \lambda_i \in Z \text{ for } i = 1, \ldots, 10.$$

Bruns et al. solved this and found that seven generators are needed. They show how to generate more counterexamples, giving insight into why more than six are necessary.

Bruns et al. also prove that for $n \ge 6$ there exists some $C \subseteq \mathbb{Z}^n_+$ for which at least $\lfloor \frac{7}{6} n \rfloor$ vectors are needed to span its integral vectors.

IP Myth 7. *If some activities in an LP have a fixed charge, a valid MILP model is to introduce a binary variable, z, for each such activity and include constraints of the form, $0 \le x \le Uz$, where U is a given or derived upper bound on x. The fixed charge, K, enters the objective with the linear term Kz.*

The model is $\min\{f(x) + Kz : x \in X,\ z_j \in \{0,1\} \text{ and } 0 \le x_j \le z_j U_j \text{ for } j \in J\}$, where J is the set of variables with fixed charge. The idea is that $z_j = 0$ forces $x_j = 0$, whereas $z_j = 1$ presents no additional constraint on x_j, and allows $x_j > 0$, in which case it incurs the fixed charge. The issue arises when $K < 0$, sometimes called a *fixed benefit*.

Counterexample. Let $K = -1$ in the following: $\min\ 5x - z : 0 \le x \le 10z$. The optimum sets $z = 1$, but $x = 0$, contrary to what is intended.

This is an example of the *MIP-Representable problem*, introduced by Meyer[25] and advanced by Jeroslow and Lowe[19]. For a fixed charge, the minimization renders $z = 1$ as an optimal binary value if $x > 0$ is optimal. For a fixed benefit, however, the minimization could render $z = 1$ with $x = 0$, thus not representing the problem correctly.

Ed Klotz points out another problem, using software with imperfect arithmetic. Suppose there is no a priori upper bound, and you use a "big-M" for the constraint: $0 \le x \le Mz$. If M is chosen large enough that $x \le M$ is redundant, the model is theoretically correct (for $K > 0$). However, the integrality tolerance allows $z = \tau$ to be considered integer-valued (CPLEX® uses $\tau = 10^{-5}$). Suppose you set $M = 10^9$. Then, the solver can set $x = 100$ and $z = 100/10^9 = 10^{-7} < \tau$, thus allowing $x > 0$ with a net fixed-charge of only $K \times 10^{-7}$.

IP Myth 8. *If an ILP has an unbounded LPR, the ILP is also unbounded.*

The following counterexample is due to Byrd, Goldman and Heller[2], based on the work of Meyer[24], who showed this cannot happen with rational data and a feasible ILP.

Counterexample. $\max x_1 : x \ge 0,\ x \in \mathbb{Z}^4,\ x_3 - \sqrt{2}\,(x_1 - x_2) = 0,\ x_2 + x_4 = 1.$

The constraint set for the LP relaxation contains the ray, $\{(t, 0, t\sqrt{2}, 1) : t \ge 0\}$. Thus, the LPR is unbounded. The integer solutions, however, must have $x_1 = x_2$ in {0,1} and $x_3 = 0$. Thus, the only feasible solutions to the ILP are (0,0,0,1) and (1,1,0,0).

Ed Klotz points out that the IP can be bounded even with rational data if you allow type 2 SOS declarations, as in CPLEX.

Counterexample. max $x_1 : x \geq 0,\ x_2 \leq 1,\ x_3 - 1.41421x_1 + 1.41421x_2 = 0,$

```
SOS
S2:: x_1: 1 x_2: 2 x_3 : 3
End
```

In the LP relaxation, the ray $(t, 0, t*1.41421)$ remains feasible. However, the SOS requirement allows only 2 consecutive variables in the SOS set to take on nonzero values, so it cuts off this unbounded direction when enforced. As a result, the MIP has a bounded, optimal solution of $x_1 = x_2 = 1$. Of course, you could model the SOS2 condition with regular integer variables, but those would involve binary variables and big-M values, which would make the associated LP relaxation bounded by the value of M.

IP Myth 9. new *If an ILP has an unbounded LPR, the ILP is feasible.*

This was motivated by a question from Marbelly Davila.

A polyhedron is unbounded if it contains a feasible half-line — that is, $\{x^0 + th : t \geq 0\} \subseteq P$, where $x^0 \in P$ and $h \neq 0$.

Counterexample. Let $P = \{(x, y) : x \geq \frac{3}{4},\ y \geq \frac{1}{2},\ x - y = \frac{1}{4}\} = \{(\frac{3}{4}, \frac{1}{2}) + t(1, 1) : t \geq 0\}$. This does not contain any integer point since

$$(x, y) = (\tfrac{3}{4} + t, \tfrac{1}{2} + t) \rightarrow x - y = \tfrac{1}{4} \rightarrow x \notin \mathbb{Z} \text{ or } y \notin \mathbb{Z}.$$

IP Myth 10. *Suppose $x^*(t)$ is an optimal solution to the parametric ILP:*

$$\min\{cx : Ax \geq b + td,\ x \in \mathbb{Z}^n_+\},\ \textit{for } t \geq 0.$$

If $t' < t''$ and $x^*(t') = x^*(t'')$, then $x^*(t)$ is optimal for all $t \in [t', t'']$.

Wang and Horng[33] provide the following:

Counterexample. $\min 3x_1 + 2x_2 : x \in \mathbb{Z}^2_+,$

$$2x_2 \leq 9,\ 25x_1 + 10x_2 \geq 129 + 2t,\ 5x_1 + 20x_2 \geq 82 - 4t.$$

At $t = 0$ and $t = 1$, the optimal solution is $x^*(0) = x^*(1) = (4, 4)$; however, $x^*(\frac{1}{2}) = (4, 3)$.

IP Myth 11. *One can solve an ILP finitely by adding cuts of the form*

$$\sum_{j \in N^k} x_j \geq 1$$

where N^k is the set of non-basic variables in the k^{th} LP relaxation having a fractional basic solution.

This is known as the *Dantzig cut*[3]. Gomory and Hoffman[13] showed that the Dantzig cuts need not converge finitely to an optimal solution with the following:

Counterexample.

$$\max\ z = 4x_1 + 3x_2 + 3x_3 : x \in \{0,1\}^3,\ 3x_1 + 4x_2 + 4x_3 \leq 6.$$

The optimal integer solution is at $x = (1, 0, 0)$, with $z = 4$. Let s_j be the slack variable for the upper bound, $x_j + s_j = 1$, and let s_0 be the slack variable for the constraint, $3x_1 + 4x_2 + 4x_3 + s_0 = 6$. The LPR solution is at $x = (1, \frac{3}{4}, 0)$, with $z = 6\frac{1}{4}$ and $s = (0, 0, \frac{1}{4}, 1)$.

The following table gives five iterations, introducing a slack variable, t_k, when the k^{th} cut is constructed.

x_1	x_2	x_3	s_0	s_1	s_2	s_3	t_1	t_2	t_3	t_4	t_5	z	cut
1	$\frac{3}{4}$	0	0	0	$1/4$	1						$6\,1/4$	$x_3 + s_0 + s_1 - t_1 = 1$
$\frac{6}{7}$	0	$\frac{6}{7}$	0	$1/7$	1	$1/7$	0					6	$x_2 + s_0 + t_1 - t_2 = 1$
1	$\frac{2}{7}$	$\frac{2}{7}$	$\frac{5}{7}$	0	$\frac{5}{7}$	$\frac{2}{3}$	0	0				$5\frac{5}{7}$	$s_1 + t_1 + t_2 - t_3 = 1$
1	0	$\frac{1}{2}$	1	0	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0			$5\frac{1}{2}$	$x_2 + s_1 + t_2 + t_3 - t_4 = 1$
$\frac{6}{13}$	$\frac{6}{13}$	$\frac{9}{13}$	0	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{4}{13}$	$\frac{3}{13}$	$\frac{3}{13}$	0	0		$5\frac{7}{13}$	$s_0 + t_3 + t_4 - t_5 = 1$

The cuts keep going, never terminating finitely.

IP Myth 12. *For any 0-1 program with a single constraint, there exists a B&B algorithm that can determine if it is feasible in polynomial time.*

The following is due to Jeroslow[18]:

Counterexample. $\max\ x_1 : x \in \{0,\ 1\}^n,\ 2x_1 + 2x_2 + \cdots + 2x_n = n.$

This is infeasible for n odd, but any Branch-and-Bound (B&B) algorithm (that is, with ***any*** rule for fixing values of fractional variables in the LP relaxation) must evaluate at least $2^{\lceil n/2 \rceil}$ nodes before it discovers (and certifies) that it is infeasible.

Ed Klotz, a longtime member of the ILOG CPLEX technical staff, points out that modern B&B algorithms are more broadly construed to include preprocessing, among other things, that would solve this example without exhaustive search. The counterexample does emphasize the need for such things. (This is another example of how optimization software may use different conventions than in the theory — see LP Myth 9.)

IP Myth 13. *An optimal schedule of jobs with deadlines on a single machine is given by the Ratio Rule.*

Thanks to Jan-Karel Lenstra for contributing this myth.

Smith[30] asserted this in the very early years, when simple rules were sought for special cases. He proposed four ideas:

1. *Shortest Processing Time (SPT) Rule*: schedule jobs in non-decreasing order of processing times.
2. If jobs are weighted, let t_j/w_j be the processing time over the (positive) weight. To minimize total weighted completion time, schedule jobs in non-decreasing order of t_j/w_j. This is the *Ratio Rule*.
3. If each job j must be finished by a given deadline d_j, one minimizes total completion time by selecting from all jobs j that are eligible for the last position (that is, i for which $d_i \geq \sum_j t_j$) the one with largest t_i; put that job in the last position and repeat. If, at any point, there is no eligible job, there is no feasible schedule.
4. If each job j must be finished by a given deadline, one minimizes the total weighted completion time by combining the ideas of (3) and (4) — that is, by applying the Ratio Rule to the eligible job from the end of the schedule backwards.

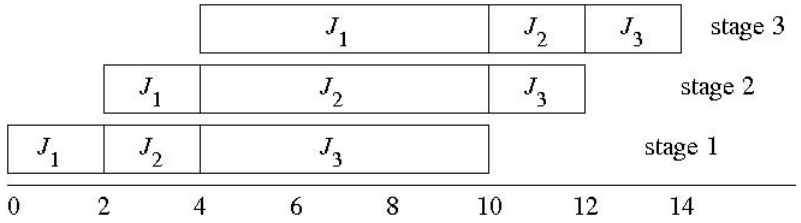
Lenstra, Rinnooy Kan, and Brucker[23] prove that problem 4 is NP-hard, so the assertion is a myth (unless $P = NP$). Many thanks to Jan-Karel Lenstra for providing the following:

Counterexample. We have three jobs with process times: $t = (2, 1, 1)$, deadlines: $d = (4, 4, 3)$, and weights: $w = (7, 4, 1)$. The Ratio Rule yields the schedule $(2, 3, 1)$ with objective value 34. The optimal schedule is $(1, 3, 2)$ with objective value 33.

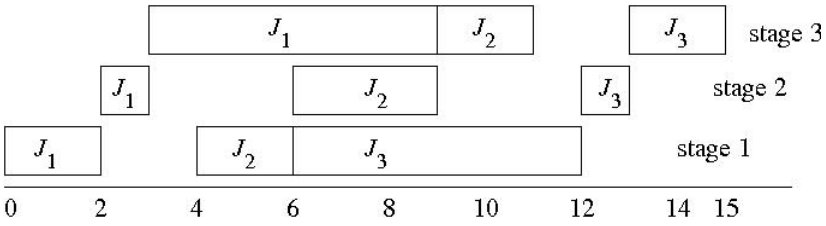
IP Myth 14. *A no-wait flow-shop's makespan cannot be worse by increasing the speed of some machines.*

Spieksma and Woeginger[31] provide the following:

Counterexample. Jobs 1, 2, and 3 are scheduled, each with three stages of operations as shown in the following figure. The minimum makespan is 14.



Now suppose the time spent on each machine in stage 2 is cut in half. Because the problem is a ***no-wait*** flow-shop, there cannot be any idle time between the processing of consecutive operations of the same job. Thus, the same job order yields a makespan of 15.



This is the minimum makespan for the new problem with speedup in stage 2. (The job orders 1-3-2 and 2-1-3 also have makespans of 15; job orders 2-3-1 and 3-1-2 have makespans of 17; and, job order 3-2-1 has a makespan of 19.)

Spieksma and Woeginger provide variations on the speedup and establish the following:

For every real number $r \geq 1$, there exists an instance of the no-wait flow-shop problem with minimum makespan C^* a speedup of processing time for some jobs and machines such that the makespan is at least $r\,C^*$.

IP Myth 15. **new** *The worst-case solution quality of the First-Fit bin-packing algorithm equals the maximum feasible decomposition of the bin size.*

Let α denote the bin size, and let $L = \{a_1, \ldots, a_n\}$ be an ordered list of items with sizes $0 < s(a_i) \leq 1$. Let $OPT(L, \alpha)$ denote the minimum number of bins needed to pack the items, and let $FF(L, 1)$ denote the number of bins of size 1 needed for a First-Fit packing of L. The *worst-case solution quality* is the ratio:

$$R(\alpha) = \limsup_{N \to \infty} \max_{L} \left\{ \frac{FF(L,1)}{N} : OPT(L,\alpha) = N \right\}.$$

A *feasible decomposition* of α is an ordered sequence of integers $p_1 \leq p_2 \leq \ldots$, such that

$$\sum_i \frac{1}{p_i} = \alpha, \; p_1 \geq 2, \text{ and } \left|\{i : p_i > 2\}\right| \geq 2.$$

For example, for $\alpha = 1$, $p = (2, 3, 6)$ is a feasible decomposition. (Note that $p_2 > 2$ to satisfy the last condition.) Once we set $p_3 = 6$, we are done since

$$\frac{1}{p_1} + \frac{1}{p_2} + \frac{1}{p_3} = \frac{1}{2} + \frac{1}{3} + \frac{1}{6} = 1 = \alpha.$$

Let $\mathcal{P}(\alpha)$ equal the set of feasible decompositions of α, and define the *maximum feasible decomposition*:

$$W(\alpha) = \max_{p \in \mathcal{P}(\alpha)} \sum_i \frac{1}{p_i - 1}.$$

For example, $W(1) = 1 + \frac{1}{2} + \frac{1}{5} = \frac{17}{10}$.

The myth asserts $R(\alpha) = W(\alpha)$. This was conjectured by Garey, Graham, and Johnson[8], upon noticing its truth for the special case of a bin size of 1: $R(1) = W(1) = \frac{17}{10}$, a curious equation. Further, they found an efficient algorithm to compute $W(\alpha)$, so if the conjecture proved true, we could compute the worst-case solution quality without solving the worst-case.

Counterexample. Shearer[29] provides the following. Let $\alpha = \frac{1}{3} + \frac{1}{7} + \frac{1}{62} = \frac{641}{1302} = \frac{2564}{5208}$. Consider a list L of 120 items with sizes:

$$\begin{aligned} s(a_i) &= \tfrac{745}{5208}, \quad 1 \le i \le 30 \\ s(a_i) &= \tfrac{869}{5208}, \quad 31 \le i \le 60 \\ s(a_{2i-1}) &= \tfrac{1695}{5208}, \quad 31 \le i \le 60 \\ s(a_{2i}) &= \tfrac{1819}{5208}, \quad 31 \le i \le 60 \end{aligned}$$

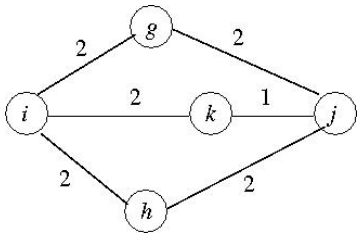
The First-Fit algorithm packs L into 41 bins of size 1. The first five bins each contain 6 items of size $\frac{745}{5208}$; the next six bins each contain 5 items of size $\frac{869}{5208}$; and, the remaining 30 bins each contain 1 item of size $\frac{1695}{5208}$ and 1 item of size $\frac{1819}{5208}$. An optimal packing uses 60 bins of size α, so $R(\alpha) \ge \frac{41}{60}$. However, $W(\alpha) = \frac{1}{2} + \frac{1}{6} + \frac{1}{61} = \frac{2500}{3660} < \frac{41}{60}$.

IP Myth 16. *Suppose the edge weights satisfy the triangle inequality in a general routing problem. Consider required nodes i, j, k such that $(i,j) \notin E$ and $\big[(k,\ell) \in E \leftrightarrow \ell \in \{i,j\}\big]$. Then, the required nodes can be replaced by one required edge (i,j) with weight $w_{ij} = w_{ik} + w_{kj}$.*

Let $G = [N, E]$ be an undirected graph with edge weights $w \ge 0$. The *General Routing Problem* (GRP) is to find a tour with minimum total weight that contains a specified subset of nodes, $\widehat{N}$, and a specified subset of edges, $\widehat{E}$. (Note that this specializes to the TSP if $\widehat{N} = N$ and $\widehat{E} = \emptyset$ and to the Chinese Postman Problem if $\widehat{N} = \emptyset$ and $\widehat{E} = E$.) The myth assumes $w_{ij} \le w_{ik} + w_{kj}$.

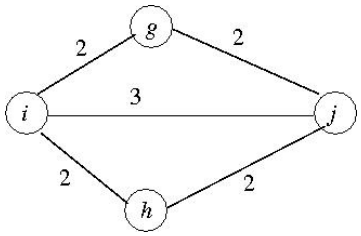
Orloff[26] introduced the reduction rule with the intuition that the added edge represents the path $i \to k \to j$. Lenstra and Rinnooy Kan[22] provide the following:

Counterexample. Let $\widehat{N} = N$ and $\widehat{E} = \emptyset$ in the following graph.



An optimal tour is
$k \to j \to g \to i \to h \to j \to k$
with total weight = 10.

Applying the reduction, the new GRP has $\widehat{N} = \{g, h\}$ and $\widehat{E} = \{(i,j)\}$ with $w_{ij} = w_{ik} + w_{kj}$.



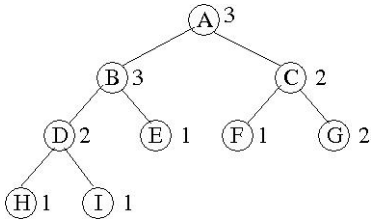
An optimal tour is
$i \to j \to g \to i \to h \to j \to i$
with total weight = 14.

(See Orloff's rejoiner[27] and [22] for the merit of using the reduction rule as a heuristic.)

IP Myth 17. *In B&B it is best to branch from the largest upper bound (for maximization).*

Fox et al.[6] provide the following:

Counterexample. The numbers next to each node in the following search tree are the upper bounds. Assume node G contains the maximum whose value is 2.



The *largest-upper-bound* (LUB) branching rule searches the nodes in one of the orders:

1. A, B, C, D, E, F, G (H & I not generated)
2. A, B, D, C, E, H, I, F, G

The particular order depends upon the expansion rule and how ties are broken. After B is expanded with children D & E, order 1 uses breadth-first search and chooses C; order 2 uses depth-first search and chooses D. Order 1 is better because it searches fewer nodes, but order 2 could be the one generated by the LUB branching rule.

A key to whether LUB is in some sense an optimal branching rule partly depends upon how ties are broken and the order in which the siblings are expanded. If the right-child is expanded first (among those with the same upper bound), the orders become:

1'. A, B, E, D, C, G

2'. A, B, E, D, I, H, C, G

Order 1' checks only 6 nodes, which is better than the left-child order of expansion.

In any case, the shortest path to the solution node is A, C, G, which does not follow the LUB rule. Node B must still be expanded to confirm optimality at node G, so the full sequence is 6 nodes: A, C, G, B, D, E (or E, D).

One alternative to LUB is to branch on the node with the *least ambiguity* — that is, fewest binary variables that are not fixed[15]. The rhetoric for this choice is that we can reach closure quickly, and a smart implementation computes look-ahead implications, generally arising from logical conditions in the model. For example, selecting one project may force other projects to be rejected, scheduling some job may force other schedule assignments, and so on. Thus, suppose we are given two nodes with the following properties: node A has LPR bound 100 and 75 binary variables that have not been fixed, of which 20 are fractional; node B has LPR bound 101 and 10 variables that have not been fixed, one of which is fractional. The LUB rule expands A and ignores the other information; the least-ambiguity rule expands B and ignores the bound. (Hybrid rules use multiple criteria, of which these are two.)

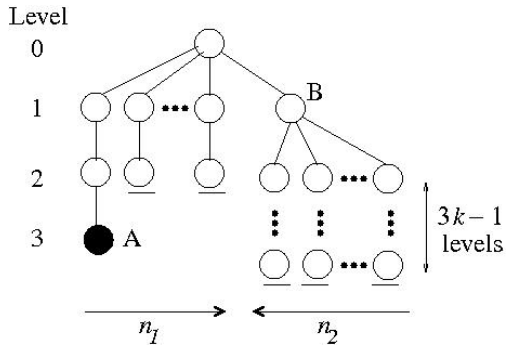
Ties for node selection, whether with LUB or not, do occur in practice, partly because the underlying problem has alternative optima, and partly due to a naive modeler ignoring symmetries. For example, in graph coloring, let $x_{ij} = 1$ if we color node i with color j. For

any solution, we can swap colors: x' represents an equivalent coloring as x, but in the model $x'_{i,\text{blue}} = x_{i,\text{green}}$ and $x'_{i,\text{green}} = x_{i,\text{blue}}$. Thus, in the model, these are alternative solutions since $x \neq x'$, and they have the same objective values, so if one does not add "symmetry exclusion constraints," ties are inevitable.

IP Myth 18. *If we increase the number of processors in a parallel B&B algorithm, the number of generated nodes decreases or remains the same.*

Lai and Sahni[21] measure performance by the number of iterations, $I(n)$, for n processors. An iteration of an n-processor model with N open nodes expands $\min\{n, N\}$ nodes. Assuming maximization, each node is evaluated by computing an upper bound (such as with LPR). Those nodes that are feasible and have an upper bound that is greater than the current best value enter the pool; those that are infeasible or cannot have a better objective value are discarded. Lai and Sahni provide the following:

Counterexample. Assume that the bound of each node is the optimum value (but not confirmed as an optimal solution value). The following shows the state tree. The n_1-processor model selects the left portion, resulting in reaching solution node A in 3 iterations (at which point node B is closed without expansion). The n_2-processor model selects the right portion, expanding the sub-tree rooted at node B for $3k-1$ more levels before closing those leaves and finally evaluating node A.



They use this construction to prove:

Let $n_1 < n_2$. For every $k > 0$, there exists a problem instance such that $k\,I(n_1) < I(n_2)$.

The construction in the state tree has $I(n_2) = 3k + 1 = k\,I(n_1) + 1 > k\,I(n_1)$. They also prove that this cannot happen if the bound is not the optimum value (which allows node B and the right-tree expansion to be candidate selections that are selected before node A).

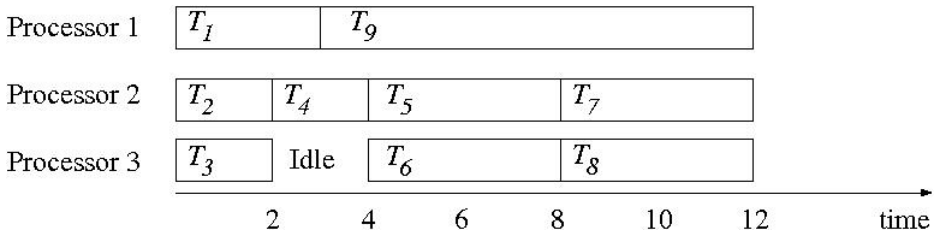
IP Background — Parallel Computation of a Schedule

In IP Myths 19–21 suppose we have n identical processors to perform computations in parallel. Tasks are presented at once with known precedence relations: $T_i \prec T_j$ means task T_i must be finished before task T_j can start. The order of the tasks is given by the list $L = \{T_{i_1}, \ldots, T_{i_r}\}$, and the rule is that a processor takes the next task in L that is ready (that is, all predecessors are finished). The processor time to perform task T_i is denoted by t_i.

To illustrate, let $L = \{T_1, \ldots, T_9\}$ with associated process times, $t = (3, 2, 2, 2, 4, 4, 4, 4, 9)$. The precedence relations are:

$$T_1 \prec T_9,\ T_4 \prec T_5, T_6, T_7, T_8.$$

Here is the time line for three processors:



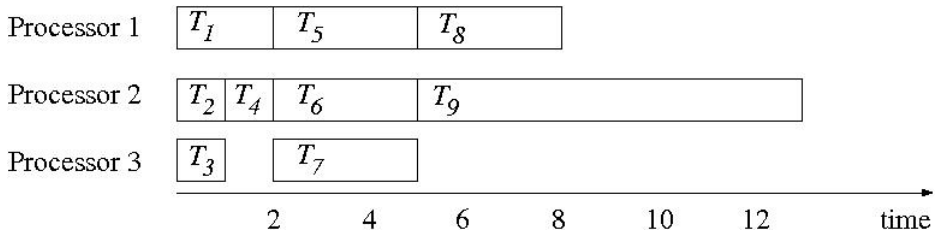
The makespan is 12.

IP Myths 19–21 are given by Graham[14], who also derives bounds on the makespan ratio for the improved system to the old, where "improved" is any combination of time reduction, added processors, precedence relaxation, and list-order rearrangement.

IP Myth 19. *If we reduce the computation time of each processor, the makespan cannot increase.*

Graham[14] provides the following:

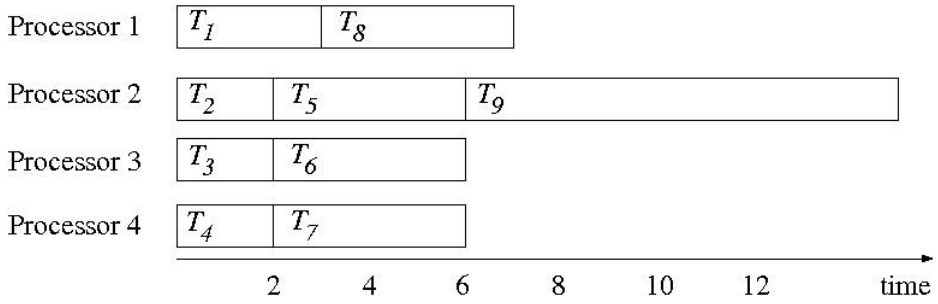
Counterexample. Change the previous example to have $t' = t - 1$. The result is a makespan of 13:



IP Myth 20. *If we add another processor, the makespan cannot increase.*

Graham[14] provides the following:

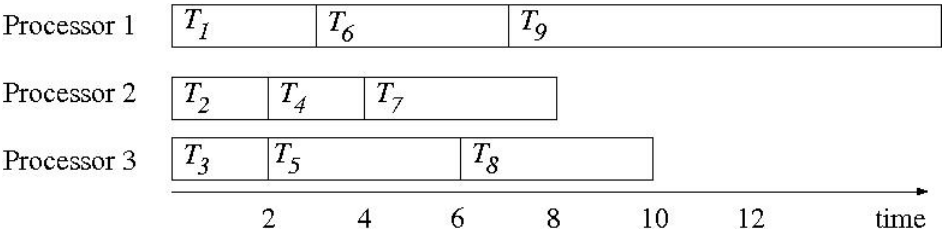
Counterexample. The fourth processor results in a makespan of 15:



IP Myth 21. *If we remove some precedence constraints, the makespan cannot increase.*

Graham[14] provides the following:

Counterexample. Remove the precedence constraints, $T_4 \prec T_5$ and $T_4 \prec T_6$. This results in a makespan of 16:



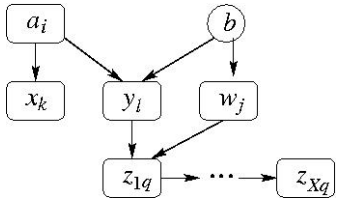
Tovey[32] extended Graham's example as follows.

Counterexample. Using Tovey's notation (nearly), the jobs are denoted: a_i, b, w_j, x_k, y_ℓ, z_{pq}, where $i = 1, \ldots, A$, $j = 1, \ldots, W$, $k = 1, \ldots, X$, $\ell = 1, \ldots, Y$, $p = 1, \ldots, X$, and $q = 1, \ldots, n+1$ (recall n = number of processors). The precedence relations are:

$$a_i \prec x_k,\ a_i \prec y_\ell \prec z_{1q},\ b \prec w_j \prec z_{1q},\ b \prec y_\ell,\ z_{pq} \prec z_{p+1,q}$$

for all i, j, k, ℓ, q. (In Graham's example, $A = 6$, $W = 1$, $Y = 6$, and $X = 4$.)

The figure on the right (taken from Tovey) shows the precedence relations.



In any optimal schedule with $n = 2$ processors, b must precede some a_i, but with $n = 3$ processors, all a_i must precede b. If all a_i must precede b, a schedule is not optimal for n processors if, and only if, $A + 1 \not\equiv 0 \mod n$. If b precedes some a_i, a schedule is not optimal for $n + 1$ processors if, and only if, $A \equiv 0 \mod (n + 1)$ and $W < n$.

In particular, consider $n = 2$ and Graham's dimensions: $a = (a_1, \ldots, a_6)$, $w = (w_1)$, $y = (y_1, \ldots, y_6)$, and $x = (x_1, \ldots, x_4)$. In this example, b does not precede any a_i in an

optimal schedule. For those same dimensions, increasing n to 3, b must precede every a_i for the schedule to be optimal.

IP Myth 22. **new** *Given jobs with unit time, an optimal schedule on n processors is obtained by assigning compatible jobs in a minimal partition.*

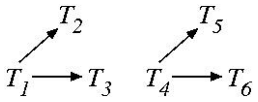
This refers to an early algorithm by Fujii, Kasami, and Ninomiya[7], which is valid for two processors and they conjectured extends to $n > 2$ processors.

Two jobs are *compatible* if neither must precede the other (that is, not adjacent in the transitive closure of the precedence graph). The algorithm is to form a minimum number of subsets of compatible jobs, such that each subset contains no more than n jobs. These are then assigned sequentially, and the minimum makespan is the number of subsets. For example, suppose $T_1 \prec T_2 \prec \cdots \prec T_N$. Then, there are no compatible jobs, and the subsets are $\{T_1\}, \ldots, \{T_N\}$, giving a minimum makespan of N, using only one processor (and having the other $n-1$ processors idle). On the other hand, if the precedence relations are $T_1 \prec T_2 \prec \cdots \prec T_{\frac{N}{2}}$ and $T_{\frac{N}{2}+1} \prec T_2 \prec \cdots \prec T_N$ (with N even), then with two processors, the jobs can be partitioned into subsets $\{T_1, T_{\frac{N}{2}+1}\}, \{T_2, T_{\frac{N}{2}+2}\}, \ldots, \{T_{\frac{N}{2}}, T_N\}$. Then, the minimum makespan is $\frac{N}{2}$, obtained from the algorithm by assigning:

processor 1	T_1	T_2	$\cdots$	$T_{\frac{N}{2}}$
processor 2	$T_{\frac{N}{2}+1}$	$T_{\frac{N}{2}+2}$	$\cdots$	T_N

The issue is whether this is valid for $n > 2$ processors. In the above example, for $n = 3$ suppose $N = 3k$ and the precedence relations are $T_1 \prec \cdots \prec T_k$, $T_{k+1} \prec \cdots \prec T_{2k}$, $T_{2k+1} \prec \cdots \prec T_N$. Then, we can partition the jobs into k subsets, and assign the jobs to achieve the minimum makespan of $\frac{N}{3}$.

Counterexample. Kaufman[20] provides the following. Let $T_1 \prec T_2, T_3$ and $T_4 \prec T_5, T_6$. For 3 processors, the minimum makespan is 3.



processor 1	T_1	T_2	T_6
processor 2	T_4	T_3	
processor 3		T_5	

The algorithm, however, obtains the partition $\{\{T_1, T_5, T_6\}, \{T_4, T_2, T_3\}\}$, giving the incorrect minimum makespan of 2. The partition satisfies the properties: each subset has no more than 3 jobs, and they are compatible.

IP Background — Metaheuristics

A *metaheuristic* is a top-level general strategy that guides other heuristics to search for feasible solutions in domains where the task is NP-hard. Examples include genetic algorithms, simulated annealing, and tabu search. The *state* is a vector defined by the problem representation; often the state is a solution. A key to any metaheuristic is the definition of *neighborhood* of a state, denoted $\mathcal{N}(s)$. (Unlike the neighborhood defined in real analysis, we typically have $s \notin \mathcal{N}(s)$.) One example is removing and/or adding an object to a knapsack.

Another example is replacing two arcs in a travelling salesman tour. A common neighborhood is complementing one binary value:

$$\mathcal{N}(x) = \cup_j \{x' : x'_i = x_i \text{ for } i \neq j,\ x'_j = 1 - x_j\}. \tag{IP.3}$$

Let f be the objective value (or some measure of fitness used in a metaheuristic), which we seek to maximize. The *depth* of a non-optimal feasible solution, x, is the minimum value $d(x)$, such that there exists a sequence $< x^0 = x, x^1, \ldots, x^k >$ that satisfies the following conditions:

1. x^i is feasible and $x^i \in \mathcal{N}(x^{i-1})$ for $i = 1, \ldots, k$
2. $f(x^k) > f(x^0)$.
3. $f(x^i) \geq f(x^0) + d(x)$ for $i = 1, \ldots, n$.

The depth of a problem instance P with respect to a neighborhood [and fitness function] is $d(P) = \max\{d(x) : x \in X\}$, where X is the set of feasible solutions.

IP Myth 23. new *Computing the depth of a discrete optimization problem P with respect to a neighborhood is at least as hard as solving P.*

Woeginger[34] provides the following:

Counterexample. Let P be an instance of the Satisfiability Problem (SAT), which is NP-complete. Let x be a truth setting and $L(x)$ a logical expression whose truth value we seek. Let $f(x)$ be the truth value of $L(x)$ (that is, 1 if true; 0 if false), so we seek to maximize f over the 2^n binary values.

Define the neighborhood as in (IP.3). Then, the depth of any non-optimal feasible solution is 0 (with $f(x^i) = f(x^{i-1}) = 0$ for $i = 1, \ldots, k-1$ and $f(x^k) = 1$), so $d(P) = 0$. Further, the depth of x is trivial to compute.

IP Myth 24. new *Computing the depth of a discrete optimization problem P with respect to a neighborhood is at most as hard as solving P.*

Woeginger[34] provides the following:

Counterexample. Let P be an instance of the Satisfiability Problem (SAT). Let x be a truth setting and $L(x)$ a logical expression whose truth value we seek. Let the state of the system be bit strings in $\{0, 1\}^{n+2}$, where $s = (x, s_{n+1}, s_{n+2})$ and $f(s) = -c(s)$, where

$$\begin{aligned} s_{n+1} = s_{n+2} = 0 \quad &\Rightarrow\ c(s) = 0 \\ s_{n+1} = s_{n+2} = 1 \quad &\Rightarrow\ c(s) = 1 \\ s_{n+1} \neq s_{n+2},\ L(x) = 1 \quad &\Rightarrow\ c(s) = 1 \\ s_{n+1} \neq s_{n+2},\ L(x) = 0 \quad &\Rightarrow\ c(s) = 2 \end{aligned}$$

We seek to minimize c, and $\mathcal{N}(s)$ is defined to be the 1-bit flip (IP.3) plus s.

Woeginger proves: *If L is satisfiable, $d(P) = 0$. If L is not satisfiable, $d(P) = 1$.* This proves that determining the depth is NP-hard. Further, a global minimum is found simply by any output whose last two bits are zero. Thus, the myth is false (unless $P = NP$).

IP Myth 25. new *In a metaheuristic search, it is better to start with a best-possible objective value, even if it is not the global optimum.*

Consider a binary IP and the neighborhood as (IP.3) plus complementing all ($x' = 1 - x$), if that is feasible.

Counterexample. Glover and Hao[10] provide the following:

$$\max\ nx_1 - \textstyle\sum_{j=2}^{n} x_j : x \in \{0,\, 1\}^n,\ (n-1)x_1 - \sum_{j=2}^{n} x_j \le 0.$$

The worst feasible solution is $x = (0, 1, 1, \ldots, 1)$. Complementing each x yields the global optimum in one iteration. Starting at some other feasible solution, such as $x = 0$, causes the search to re-visit $x = 0$ many times before reaching the worst solution (followed by the global maximum).

IP Myth 26. *For N sufficiently large, simulated annealing visits the global minimum within N iterations with probability 1.*

Fox[4, 5] provides the following:

Counterexample. Let $X = \{1, 2, 3\}$ and $f(X) = (1, 3, 0)$, so, $x^* = 3$ is the global minimum. The system state is the value of x, and the neighborhoods are: $N(1) = \{2\}$, $N(2) = \{1, 3\}$, and $N(3) = \{2\}$. The acceptance probability of an uphill move from x_1 to x_2 is given by:

$$P(X(k+1) = x_2 \,|\, X(k) = x_1) = e^{-\frac{f(x_2)-f(x_1)}{T_k}} = e^{-\frac{2}{T_k}},$$

where T_k is the temperature at iteration k. Once the state moves from x_1 to x_2, it then moves to x_3, the global minimum. So, not visiting x_3 is equivalent to remaining at x_1 forever. That probability is given by:

$$P(X(1) = X(2) = \cdots = X(k) = x_1 \,|\, X(0) = x_1) = \prod_{i=1}^{k} \left(1 - e^{-\frac{2}{T_i}}\right).$$

Thus, the system does not reach the global minimum with probability 1 within any *finite* number of iterations.

As the general theory goes, the example does converge to the global minimum asymptotically with probability 1. However, the expected number of iterations is infinite. Specifically, for a standard cooling schedule, Fox shows

$$\lim_{k \to \infty} (-k + \mathbf{E}[N \,|\, N > k]) = \infty.$$

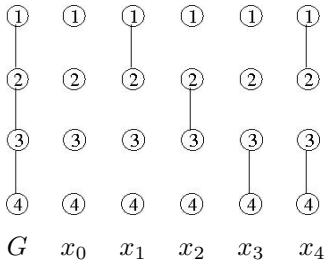
In words, the longer the search has been unsuccessful in reaching the global minimum, the longer the expected remaining time to reach it.

Fox provides variations that escape this difficulty (among others).

IP Myth 27. *In simulated annealing, it is always better to let the temperature decrease.*

Hajek and Sasaki[16] provide sufficient conditions for which no cooling temperature sequence is better than a constant temperature. They then show how the conditions apply to a matching problem, for which the following counterexample is a special case.

Counterexample. Let G be a simple path with 4 nodes, for which there are five matchings, denoted $x_0 = \emptyset,\ x_1 = \{(1,2)\},\ x_2 = \{(2,3)\},\ x_3 = \{(3,4)\},\ x_4 = \{(1,2),\ (3,4)\}$.



Let x be a matching, and let its neighborhood be any matching that differs by exactly one edge:

$$\begin{aligned}
\mathcal{N}(x_0) &= \{x_1, x_2, x_3\} \\
\mathcal{N}(x_1) &= \{x_0, x_4\} \\
\mathcal{N}(x_2) &= \{x_0\} \\
\mathcal{N}(x_3) &= \{x_0, x_4\} \\
\mathcal{N}(x_4) &= \{x_1, x_3\}.
\end{aligned}$$

The transition from x_i to x_j consists of two steps: (1) Select $x_j \in \mathcal{N}(x_i)$ with probability R_{ij}; (2) Accept x_j according to the following SA rule:

$$\mathbf{Pr}(X(k+1) = x_j \,|\, X(k) = x_i) = \begin{cases} 1 & \text{if } x_j \supset x_i; \\ e^{-\frac{1}{T_k}} & \text{if } x_j \subset x_i, \end{cases}$$

where T_k is the temperature. If x_j is not accepted, set $X(k+1) = x_i$.

Let each neighbor be equally-likely to be selected in step 1: $R_{ij} = \frac{1}{|\mathcal{N}(x_i)|}$. Then, the process is a Markov chain with the following transition probabilities:

$$Q(T_k) = \begin{array}{c} \begin{matrix} x_0 & x_1 & x_2 & x_3 & x_4 \end{matrix} \\ \begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ \frac{1}{2}\,e^{-\frac{1}{T_k}} & \frac{1}{2}(1-e^{-\frac{1}{T_k}}) & 0 & 0 & \frac{1}{2} \\ e^{-\frac{1}{T_k}} & 0 & 1-e^{-\frac{1}{T_k}} & 0 & 0 \\ \frac{1}{2}\,e^{-\frac{1}{T_k}} & 0 & 0 & \frac{1}{2}(1-e^{-\frac{1}{T_k}}) & \frac{1}{2} \\ 0 & \frac{1}{2}\,e^{-\frac{1}{T_k}} & 0 & \frac{1}{2}\,e^{-\frac{1}{T_k}} & 1-e^{-\frac{1}{T_k}} \end{bmatrix} \end{array} \begin{matrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{matrix}$$

The issue is whether it is better to let $\{T_k\}$ decrease or remain constant.

For $T_k = \infty$, the search is completely random, and for $T_k = 0$ the local-maximum matching x_2 is an absorbing state.

$$Q(\infty) = \begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} \\ 1 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{bmatrix}. \quad Q(0) = \begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Keeping the temperature constant at $T_k = \infty$ (or any value large enough to ensure acceptance), the system eventually reaches the global maximum, x_4. Whenever the system reaches the local maximum x_2, it moves to x_0. On the other hand, as $T_k \to 0$, the system could be absorbed at x_2. Thus, cooling is worse than the constant temperature.

Hajek and Sasaki conjecture the existence of other problem classes for which it is not optimal to cool the temperature.

IP References

[1] W. Bruns, J. Gubaladze. M. Henk, A. Martin, and R. Weismantel. A counterexample to an integer analogue of Carathéodory's theorem. *Journal für die reine und angewandte Mathematik (Crelle's Journal)*, 510:179–185, 1999.

[2] R. H. Byrd, A. J. Goldman, and M. Heller. Recognizing unbounded integer programs. *Operations Research*, 35(1):140–142, 1987.

[3] G. B. Dantzig. Note on solving linear programs in integers. *Naval Research Logistics Quarterly*, 6(1):75–76, 1959.

[4] B. L. Fox. Faster simulated annealing. *SIAM Journal on Optimization*, 5(3):488–505, 1995.

[5] B. L. Fox. Simulated annealing: Folklore, facts, and directions. In H. Niederreiter and P. J. Shieu, editors, *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing: Proceedings of a Conference at the University of Nevada*, volume 106 of *Lecture Notes in Statistics*, pages 17–48, Berlin, FRG, 1995. Springer.

[6] B. L. Fox, J. K. Lenstra, A. H. G. Rinnooy Kan, and L. Schrage. Branching from the largest upper bound: Folklore and fact. *European Journal of Operational Research*, 2:191–194, 1978.

[7] M. Fujii, T. Kasami, and K. Ninomiya. Optimal sequencing of two equivalent processors. *SIAM Journal on Applied Mathematics*, 17(4):784–789, 1969.

[8] M.R. Garey, R.L. Graham, and D. S. Johnson. On a number-theoretic bin packing conjecture. In *Proceedings of the 5th Hungarian Combinatorial Colloquium*, pages 377–392, Amsterdam, 1978. North-Holland. Conference was held 1976 in Budapest.

[9] F. Glover. *Netforms in Management Science and Decision Support.* University of Colorado, School of Business, Boulder, CO, 1983.

[10] F. Glover and J-K. Hao. Hard optimization problems and strategic oscillation. Working paper, Laboratoire d'Etude et de Recherche en Informatique (LERIA), Université d'Angers, Angers, France, 2008.

[11] F. Glover, D. Klingman, and N. V. Phillips. *Network Models in Optimization and Their Applications in Practice.* Wiley-Interscience, New York, NY, 1992.

[12] F. Glover and D. C. Sommer. Pitfalls of rounding in discrete management decision problems. *Decision Sciences*, 6:211–220, 1975.

[13] R. E. Gomory and A. J. Hoffman. On the convergence of an integer-programming process. *Naval Research Logistics Quarterly*, 10:121–123, 1963.

[14] R. L. Graham. Bounds on multiprocessing timing anomalies. *SIAM Journal on Applied Mathematics*, 17(2):416–429, 1969.

[15] H. J. Greenberg. MIP tactics. In R. H. F. Jackson and R. P. O'Neill, editors, *Special Issue: Mixed Integer Programming in Mathematical Programming Systems*, pages 3–11. ORSA Computer Science Technical Section and Mathematical Programming Society Committee on Algorithms, 1984.

[16] B. Hajek and G. Sasaki. Simulated annealing — to cool or not. *Systems & Control Letters*, 12:443–447, 1989.

[17] R. G. Jeroslow. Comments on integer hulls of two linear constraints. *Operations Research*, 19(4):1061–1069, 1971.

[18] R. G. Jeroslow. Trivial integer programs unsolvable by branch-and-bound. *Mathematical Programming*, 6(1):105–109, 1974.

[19] R. G. Jeroslow and J. K. Lowe. Modeling with integer variables. *Mathematical Programming Study*, 22:167–184, 1984.

[20] M. T. Kaufman. Counterexample to a conjecture of Fujii, Kasami and Ninomiya. *SIAM Journal on Applied Mathematics*, 26(2):370–371, 1974.

[21] T-H. Lai and S. Sahni. Anomalies in parallel branch-and-bound algorithms. *Communications of the ACM*, 27(4):594–602, 1984.

[22] J. K. Lenstra and A. H. G. Rinnooy Kan. On general routing problems. *Networks*, 6(3):273–280, 1976.

[23] J. K. Lenstra, A. H. G. Rinnooy Kan, and P. Brucker. Complexity of machine scheduling problems. *Annals of Discrete Mathematics*, 1:343–362, 1977.

[24] R. R. Meyer. On the existence of optimal solutions to integer and mixed-integer programming problems. *Mathematical Programming*, 7:223–235, 1974.

[25] R. R. Meyer. Integer and mixed-integer programming models: General properties. *Journal of Optimization Theory and Applications*, 16(3/4):191–206, 1975.

[26] C. S. Orloff. A fundamental problem in vehicle routing. *Networks*, 4(1):35–64, 1974.

[27] C. S. Orloff. On general routing problems: Comments. *Networks*, 6(3):281–284, 1976.

[28] D. S. Rubin. On the unlimited number of faces in integer hulls of liner programs with a single constraint. *Operations Research*, 18(5):940–946, 1970.

[29] J. P. Shearer. A counterexample to a bin packing conjecture. *SIAM Journal of Algebraic and Discrete Methods*, 2(3):309–310, 1981.

[30] W. E. Smith. Various optimizers for single-state production. *Naval Research Logistics Quarterly*, 3:59–66, 1956.

[31] F. C. R. Spieksma and G. J. Woeinger. The no-wait flow-shop paradox. *Operations Research Letters*, 33(6):603–608, 2005.

[32] C. A. Tovey. A simplified anomaly and reduction for precedence constrained multiprocessor scheduling. *SIAM Journal on Discrete Mathematics*, 3(4):582–584, 1990.

[33] H-F. Wang and J-S Horng. Structural approach to parametric analysis of an IP on the case of the right-hand side. *European Journal of Operational Research*, 92(1):148–156, 1996.

[34] G. J. Woeginger. A note on the depth function of combinatorial optimization problems. *Discrete Applied Mathematics*, 108:325–328, 2001.

Dynamic Programming

A dynamic program (DP) is one that can be solved as a sequence of state-dependent optimization problems. When the underlying problem is dynamic, time provides the natural ordering for sequential optimization. However, DP is also a technique used to decompose a static problem into a sequence of lower-dimensional decision problems. A classical example of this decomposition is the *knapsack problem*:

$$\max \textstyle\sum_j c_j x_j : \sum_j a_j x_j \leq b,\ x \in \mathbb{Z}_+^n,$$

where $a, c > 0$. (See Martello and Toth[15] for a more extensive introduction.) A DP formulation of this is the *forward recursion*:

$$f_k(s) = \max \{c_j x_j + f_{k-1}(s - a_j x_j) : x_j \in \mathbb{Z}_+,\ a_j x_j \leq s\} \text{ for } s = 0, 1, \ldots, b,$$

for $k = 1, \ldots, n$ and $f_0(s) = 0$ for all $s = 0, \ldots, b$. The DP algorithm starts with $k = 0$ (with $f_0(s) = 0$ for $s \geq 0$), and it proceeds forward: $k = 1, 2, \ldots, n$. The solution value to the original problem is $f_n(b)$, and x^* is computed by backtracking through the 1-dimensional optimal solutions.

At the foundation is Bellman's *Principle of Optimality*[2]:

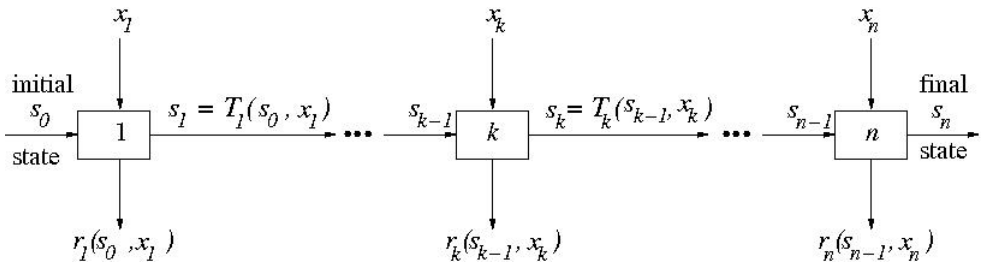
> "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision."

This is what enables us to decompose an n-variable problem into a sequence of 1-variable problems. If the number of states and decisions is finite, this is equivalent to a *shortest path* through a network. The nodes are the (state, stage) pairs and the arcs are the transitions resulting from the decision.

In its natural time-ordered form, DP represents a *sequential decision process*: a discrete-time process characterized by a sequence of states, where the next state depends upon the current state and decision. (It does not upon the rest of the history of states and decisions.) At each time period, the decision yields a return and a state transition. Here is the *backward DP recursion*:

$$f_k(s) = \max_{x \in X_k(s)} \{r_k(s, x) + f_{k+1}(T_k(s, x))\},$$

where $X_k(s)$ is the set of decisions upon entering time period k in state s; $r_k(s, x)$ is the immediate return for choosing $x \in X_k(s)$; and, the last term is the total future return after transitioning to the new state, $T_k(s, x)$. This is illustrated in the following diagram:

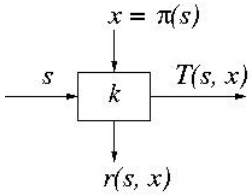


If the time periods are long enough, the present value is used with a *discount factor*, $\beta \in (0, 1]$:

$$f_k(s) = \max_{x \in X_k(s)} \{r_k(s, x) + \beta f_{k+1}(T_k(s, x))\}.$$

So, $f_0(s_0) = \sum_{k=1}^{n} \beta^{k-1} r_t(s_k, x_k^*)$, where $\{x_k^*\}$ are the decisions made at each time period, and the state sequence is given by $s_k = T_k(s_{k-1}, x_k^*)$ for $k = 1, \ldots, n$. Denote a *policy* by $\pi_k(s)$ = decision made at time k upon entering in state s. An *optimal policy* maximizes $f_0(s)$; equivalently, $\pi_k^*(s) = x_k^*$ for some $x_k^* \in \text{argmax}_{x \in X_k(s)}\{r_k(s, x) + \beta f_{k+1}(T_k(s, x))\}$.

The DP is *stationary* if the decision set and functions are independent of time: $X_k = X$, $r_k = r$, and $T_k = T$. A *stationary policy* is a function of state, but not of time: $\pi(s) \in X(s)$. It specifies the decision to be taken. This is illustrated on the right.



A *randomized policy* is one that is specified by $P_t(s, x)$ = probability that $\pi_t(s) = x$ when the system is in state s at time t. The actual decision is determined by some random selection method according to P. In general, $P_t(s, x) \in [0, 1]$ and for finite or denumerable decision sets, $\sum_{x \in X_t(s)} P_t(s, x) = 1$ for all s. ($P_t(s, x) = 0$ for $x \notin X_t(s)$.) The *non-randomized policy*, a.k.a., *pure policy*, is the special case: $P_t(s, \pi_t(s)) = 1$ for all s, t; otherwise, $P_t(s, x) < 1$ for at least one $x \in X_t(s)$ for some s, t.

DP Myth 1. *Given a separable objective function, the Principle of Optimality enables the decomposition into a series of smaller optimization problems over a state space. In particular, suppose*

$$R(x) = r_1(x_1) \oplus r_2(x_2) \oplus \cdots \oplus r_n(x_n)$$

over the separable domain, $X = X_1 \times X_2 \times \cdots \times X_n$. Further, we have a simple limit constraint, $\sum_{j=1}^{n} x_j \le b$. Then,

$$\max_{x \in X}\{R(x) : \sum_{j=1}^{n} x_j \le b\} = \max_{s \le b} f_n(s),$$

where

$$f_j(s) = \max_{x_j \in X_j} \{r_j(x_j) \oplus f_{j-1}(s - x_j) : s - x_j \le b\} \text{ for } j = 1, \ldots, n,\ s \le b.$$

with $f_0(s) = i$ = identity element for $\oplus$ (= 0 if $\oplus$ is ordinary addition; = 1 if $\oplus$ is ordinary multiplication).

The *Principle of Optimality* was originally developed for additive processes (where $\oplus$ is simple addition). Mitten[16] pointed out that non-additive processes may not decompose directly, as he developed a general framework.

Counterexample. $f(x_1, x_2) = x_1 x_2$ and $X_1 = X_2 = [-2, 1]$.

For $b \ge -4$, the 2-variable maximum value is 4. However, $f_1(s) = 1$ for all $s \le 1$, which yields $f_2(s) = 1$ for all $s \le 2$. The DP solution is thus $x = (1, 1)$ with $R(x) = 1$.

The problem is that $\oplus$ violates *Mitten's monotonicity condition* on this domain.

DP Myth 2. *The Principle of Optimality is a necessary condition for the optimality of a policy.*

This myth and the following counterexample are given by Porteus[18].

Counterexample. Let the state space be the interval [0, 1]. For each state there are two possible decisions: $X(s) = \{0, 1\}$. The immediate return is the same for each state: $r(s, x) = x$. Regardless of the decision and current state, the state transition is a uniform random variable. The objective is the discounted total return with discount factor $\beta < 1$.

It is optimal to set $x_n(s) = 1$ for all n and all states, s. Consider the alternative policy that sets $x_k(s) = 1$ for $s \neq 1$. This yields the same expected total discounted return, but it violates the necessity of the Principle of Optimality.

Because the probability of any one return is zero, what is done for just one decision for one state has no effect on the objective value.

DP Myth 3. *In a dynamic lot size inventory model, decreasing setup costs does not increase total inventory.*

The intuition behind this is that inventory is caused by the setup cost. In the *Economic Order Quantity* (EOQ) model, we have

$$Q = \sqrt{\frac{2Kd}{h}},$$

where Q is the min-cost order quantity, K is the setup cost, d is the demand, and h is the holding cost. Thus, reducing the setup cost does reduce the EOQ. However, this does not carry over to the dynamic lot size problem, where costs and demands occur over time, and the decision variables are how much to produce in each period to satisfy the demands.

The DP recursion is given by:

$$f_t(y) = \min_{x \geq 0} \{p_t(x) + h_t(x + y - d_t) + f_{t-1}(x + y - d_t) : x + y \geq d_t\}, \quad \text{for } t = 1, \ldots, N$$

$$f_0(0) = 0;\ f_0(y) = \infty \text{ for } y > 0,$$

where y is the inventory level (state), starting with $y_0 = 0$, x is the production level, p_t is the production cost in period t, d_t is the demand, and h_t is the 1-period holding cost in period t for the new inventory level, $x + y - d_t$.

Zangwill[25] provides further discussion and the following:

Counterexample. A plant runs two shifts a day, a morning shift and a night shift. Consider two days of its operation which we divide into four periods. Designate period 1 as the morning shift of day 1, period 2 as the night shift of day 1, with periods 3 and 4 the day and night shifts, respectively, of day 2. Suppose the product demand during each shift is 3 units. Let the variable production cost be linear and stationary: $p_t(x_t) = px_t$ for all t. Because total production equals total demand, this form eliminates variable production cost as a factor in the objective. What remains is the setup cost, $\sum_t K_t \delta(x_t)$, where

$$\delta(x) = \begin{cases} 1 & \text{if x} > 0; \\ 0 & \text{otherwise.} \end{cases}$$

Let the holding cost be $h_t(y_t) = y_t$, for all t, where y_t is the inventory at the end of period t. Thus, the dynamic lot size model is given by:

$$\min \sum_{t=1}^{4} (K_t \delta(x_t) + y_t) : x, y \geq 0,$$
$$y_{t-1} + x_t - y_t = d_t,$$

with $y_0 = 0$.

Scene 1. At present the plant is quite busy during the day, and the setup costs during the day are higher than at night. In particular $K_1 = K_3 = 8$, $K_2 = K_4 = 5$. The optimal production schedule is $x^* = (3, 6, 0, 3)$ with associated inventory levels $y^* = (0, 3, 0, 0)$.

Scene 2. The engineering department undertakes to reduce setup costs and thereby move closer to a *Zero-Inventory* system. After considerable analysis, they conceive how to use the greater range of talent available during the day, which enables them to reduce setup costs more during the day than at night. After the engineering department completes its task, the setup costs are significantly reduced and become $K_1 = K_3 = 1$, $K_2 = K_4 = 4$. All other costs remain the same. The new optimal production schedule is $x' = (6, 0, 6, 0)$ with associated inventory levels $y' = (3, 0, 3, 0)$. Even though all setup costs have been cut (and no other changes made), the total inventory level has doubled.

Zangwill provides conditions under which decreasing setup costs results in decreasing inventory. The key feature of the counterexample is that the day-shift reduction is different from the night-shift reduction.

DP Myth 4. *In a dynamic lot size inventory model, decreasing setup costs does not increase minimum total cost.*

See DP Myth 3 for the description of the dynamic lot size problem.

Zangwill[25] provides further discussion and the following:

Counterexample. To manufacture a particular product requires three separate operations, call them I, II, and III. Suppose also that five workstations exist, and each workstation can do the operations enumerated:

A={I}, B={II}, C={III}, D={I, II}, E={II, III}.

For example, workstation A can do only operation I whereas station D can accomplish both operation I and operation II. Since all three operations are required to complete the product, there are three possible *routings*: A→B→C, A→E, and D→C. For example, route A→E accomplishes operation I at station A and station E does II and III. Generally, the various workstations are scheduled carefully with the work flow balanced and optimally allocated.

Upon occasion, an emergency rush order for the product arises, which is costly since it disrupts operations. The more emergency orders that occur during a day the more costly it becomes because additional disruptions cause the regular schedule to become increasingly rushed. Initially, suppose for an emergency order on a workstation that the cost for the x^{th} emergency order that day is

$$q_A(x) = q_C(x) = 10 + 10x, \; q_D(x) = q_B(x) = q_E(x) = 31 + x.$$

Here $q_D(x) = 31 + x$ means that for workstation D, the setup cost is 31 for processing an emergency order, and x is the additional cost if $x - 1$ emergency orders have already been processed at workstation D. The cost increases as more emergency orders are processed at a workstation, as mentioned, because of the increased disruption.

The expeditor is the individual who juggles the work and tries to process the emergency as inexpensively as possible by selecting the routing. The cost depends not only on which workstations are along the route but also on how many emergencies a workstation has already had to contend with that day. Given the costs, here is the minimum cost for processing if there are one or two emergencies in a day:

If one emergency occurs, an optimal route is A→E at a cost of 52.

If a second emergency occurs, an optimal route is D→C, a cost of 52.

Thus, if one emergency occurs, the total cost is 52. Should two emergencies occur, the total cost is 104.

Suppose the setup cost for an emergency on workstation B is cut from 31 to 10: $q_B(x) = 10 + x$. All other costs remain the same. Now we have:

If one emergency occurs, an optimal route is A→B→C at a cost of 51.

If a second emergency occurs, an optimal route is A→E, at a cost of 62.

Thus, the cost of one emergency during the day is 51, but if two emergencies occur, the cost is 113. If we are unfortunate enough to get two emergencies during the day, the cost is higher after the setup cost reduction, so the setup cost reduction has actually increased the minimum total cost.

DP Myth 5. **new** *The Federgruen-Lee algorithm produces an optimal solution to the dynamic lot size model with quantity discount.*

Federgruen and Lee[5] proposed a DP algorithm, but there are special cases for which it does not necessarily produce an optimal solution. Notation:

D_t	demand in period t
K_t	fixed setup cost in period t
c_t	unit purchase price in period t
h_t	unit holding cost
N	discount quantity
r	discount rate
x_t	amount purchased in period t

The model has $x_t \geq N \rightarrow$ purchase cost $= c_t(1 - r)$ and holding cost $= h_t(1 - r)$.

Counterexample. Xu and Lu[24] provide the following: $n = 4$, $D = (10, 40, 80, 20)$, $N = 75$, $r = 0.1$, $c = (8, 8, 8, 8)$, $K = (150, 150, 150, 150)$, and $h = (5, 5, 5, 5)$. The Federgruen-Lee algorithm obtains $x = D$ with a total cost of \$1,736. An optimal solution is $x^* = (10, 40, 100, 0)$ with a total cost of \$1,660.

Xu and Lu give more insights into the cause of the algorithm's failure. Another cause is given by the following:

Counterexample. $n = 3$, $D = (30, 30, 10)$, $N = 60$, $r = 0.1$, $c = (10, 10, 10)$, $K = (60, 60, 60)$, and $h = (2, 2, 2)$. The Federgruen-Lee algorithm obtains $x = (60, 0, 10)$ with a total cost of \$814. An optimal solution is $x^* = (70, 0, 0)$ with a total cost of \$762.

Xu and Lu presented a modified algorithm to overcome such counterexamples.

DP Myth 6. *Consider a dynamic lot size problem in which lead times are stochastic, and shortages are backlogged. Optimal production levels still satisfy the property that they are zero with positive incoming inventory and otherwise equal the sum of successive demands.*

The appeal of this myth is that the property holds with zero lead times. In that case, whenever there is zero inventory upon entering period t, the optimal production level is $x_t^* = \sum_{k=t}^{t'} d_k$ for some $t' \geq t$. If the entering inventory is positive, it is enough to meet the demand and $x_t^* = 0$. More generally, if the lead time of production in period t is L_t, the zero-inventory point is at period $t' = \min\{k : k \geq t + L,\ d_k > 0\}$. Thus, $x_t^* y_{t'-1}^* = 0$ is the optimality property in question.

Anderson[1] provides the following:

Counterexample. The horizon is 9 periods with $d_5 = 2$, $d_8 = 3$, and $d_t = 0$ for $t \neq 5, 8$. Setup costs are all zero, and the unit production costs are $p_1 = 2$, $p_2 = 5$, $p_4 = 30$, $p_8 = 3$, and $p_t = 1000$ for $t = 3, 6, 7, 8, 9$. The holding costs are all zero, and the 1-period unit shortage costs are nonzero for $s_5 = s_9 = 1000$.

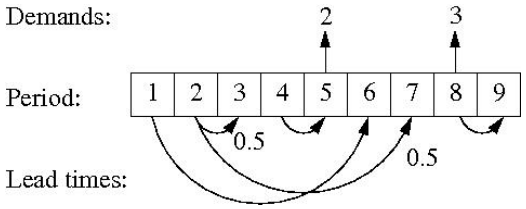


Figure taken from [1].

Production cost of 1000 is enough to render $x_3^* = x_5^* = x_6^* = x_7^* = x_9^* = 0$ in every optimal solution, so lead times for those periods are not shown. The total production cost is then $2x_1 + 5x_2 + 30x_4 + 3x_8$.

The total shortage costs for each of the two random lead times are given by:

$$L_2 = 1\colon\ -1000(x_2 + x_4 - 2)^- - 1000(x_1 + (x_2 + x_4 - 2)^+ + x_8 - 3)^-$$
$$L_2 = 5\colon\ -1000(x_4 - 2)^- - 1000(x_1 + (x_2 + x_4 - 2)^+ + x_8 - 3)^-$$

To avoid the 1000-unit costs for shortages, every optimal policy sets $x_2^* + x_4 \geq 2$, and $x_1 + x_2 + x_4 + x_8 \geq 1$.

Since this is a DP, the production levels x_4, x_8 are determined *after* the lead time from period 2 becomes known. The optimal policy is to set $x_1^* = 1$, $x_2^* = 2$, and

$$x_4^* = 2, \quad x_8^* = 0 \quad \text{if lead time} = 5$$
$$x_4^* = 0, \quad x_8^* = 2 \quad \text{if lead time} = 1.$$

Since the lead times in period 2 are equally-likely, the total expected cost is

$$2 + 10 + \tfrac{1}{2}(30 \times 2 + 3 \times 2) = 45.$$

Notice that the zero-inventory point for period 1 occurs at period 8 for the arrival pattern in which the lead time of x_2^* is 5 periods — that is, $x_1^* > 0$ and $y_7 > 0$ (where the zero-inventory point of period 1 is period 8), thus violating the myth's indicated property.

DP Myth 7. new *In a dynamic lot size problem, a stochastically greater lead time results in a greater optimal average cost.*

Let L_1, L_2 be random lead times. Then, L_1 is *stochastically greater* than L_2, denoted $L_1 \geq_{\text{st}} L_2$, if

$$\mathbf{Pr}[L_1 \geq \ell] \geq \mathbf{Pr}[L_2 \geq \ell] \text{ for all } \ell.$$

The underlying model is continuous-time, single-item, where demands form a compound Poisson process — demands occur at epochs with random batch size. Here we assume the batch size is 1. An stationary policy is optimal, where the decision variables are the target inventory levels. (This is called a *base-stock policy*, and "target" is used when the demand structure is random.)

Notation:

h	unit holding cost rate
p	unit shortage cost rate
D	lead time demand
Ψ	cdf of D
y	target inventory level for base-stock policy

Only the holding and penalty costs depend upon lead time demand, so we ignore ordering costs for purposes of policy comparisons. The expected average cost is: *odering cost* +

$$\text{lead time-dependent costs} = E[h(y - D)^+ + p(D - y)^+].$$

Counterexample. Song[21] provides the following: $\mathbf{Pr}[L_1 = 1] = 1$ and $\mathbf{Pr}[L_2 = 1] = 0.7$, $\mathbf{Pr}[L_2 = 0.1] = 0.3$. Note that $L_1 \geq_{\text{st}} L_2$. Let $h = 2$, $p = 9$, and the two cdfs:

$$\begin{array}{lll} \Psi_1(0) = 0.3679 & \Psi_1(1) = 0.7358 & \Psi_1(2) = 0.9197 \\ \Psi_2(0) = 0.5290 & \Psi_2(1) = 0.8137 & \Psi_2(2) = 0.9437 \end{array}$$

The optimal target inventory for each lead time is $y_1^* = y_2^* = 2$. Their costs differ by

$$E[h(y - D_1)^+ + p(D_1 - y)^+] - E[h(y - D_2)^+ + p(D_2 - y)^+] = 3.14 - 3.24 < 0.$$

Hence, the stochastically greater lead time has lower average cost, contrary to the myth.

Song also analyzes the effect of more variable lead time (with equal means).

DP Myth 8. new *In a multi-item, base-stock inventory system, the total order fill rate cannot exceed the independent fill rate.*

The underlying model is a continuous-time, multi-item inventory system, where items are consumed by *demand types*. Demand rates, define a Poisson process, for which a stationary base-stock policy (see p. 7) minimizes the average cost. An *order fill rate* is the probability of satisfying demand immediately.

The total order fill rate is given by:

$$FT = \sum_{k=1}^{K} q_k f_k,$$

where f_k is the type-k fill rate, and q_k is the probability that the demand is of type k. The independent fill rate, which is used to approximate FT, assumes all demands are independent of all other demands:

$$FI = \sum_{i=1}^{n} Q_i F_i,$$

where F_i is the fill rate of item i, and Q_i is the demand rate for item i. Q is determined by q:

$$Q_i = \frac{1}{\kappa} \sum_{k \in S(i)} q_k,$$

where $S(i)$ is the set of types that consume item i, and $\kappa = \sum_{i=1}^{n} \sum_{k \in S(i)} q_k$. The myth asserts $FT \leq FI$.

Counterexample. Song[22] provides the following: Let $n = 3$, all lead times are 1, and the overall demand rate is 1. Let $K = 7$ with $q = (0.01, 0.01, 0.85, 0.03, 0.01, 0.01, 0.08)$ and

$$S(1) = \{1, 4, 5, 7\},\ S(2) = \{2, 4, 6, 7\},\ S(3) = \{3, 5, 6, 7\}.$$

Let the base-stock levels be $(1, 1, 4)$. Then, $f = (0.878, 0.878, 0.984, 0.861, 0.868, 0.868, 0.858)$, so $F = (3.465, 3.465, 3.578)$, which yield the contradiction: $FT = 0.965 > FI = 0.961$.

DP Myth 9. new *Given continuous control over arrival and service rates in a network of queues, the optimal arrival rate into a queue does not increase with its size.*

Weber and Stidham[23] consider a network of m queues in which customers arrive at queue i in a Poisson stream with rate λ_i and complete service at a rate μ_i. The completion may be rejected, so the customer may remain at the same queue; otherwise, the customer may leave the system or move to another queue.

The arrival and service rates are subject to continuous control over intervals, $[\underline{\lambda}, \overline{\lambda}]$ and $[\underline{\mu}, \overline{\mu}]$, respectively. Service rates incur costs, $c_i(\mu_i)$, and arrival rates bring rewards, $r_i(\lambda_i)$. The state of the system is $s = (s_1, \ldots, s_m) \geq 0$, where s_i is the number of customers in queue i. There is a holding (or waiting) cost, $\sum_{i=1}^{m} h_i(s_i)$.

Here is the DP recursion for minimizing the total expected cost:

$$f_{n+1}(s) = \sum_{i=1}^{m} \min_{\substack{\lambda_i \in [\underline{\lambda}_i, \overline{\lambda}_i] \\ \mu_i \in [\underline{\mu}_i, \overline{\mu}_i]}} \big(h_i(s_i) + c_i(\mu_i) - r_i(\lambda_i) + \lambda_i f_n(s + \boldsymbol{e}_i) + \mu_i \mathbf{E}[f_n(T_i(s, \mu_i))] \big),$$

where $\boldsymbol{e}_i$ is the i^{th} unit vector and $T_i(s, \mu_i)$ is the new state resulting from completion events at queue i:

$$\begin{array}{ll} T_i(s,\mu_i) = s - \boldsymbol{e}_i & \text{if customer leaves system;} \\ T_i(s,\mu_i) = s - \boldsymbol{e}_i + \boldsymbol{e}_k & \text{if customer joins queue } k; \\ T_i(s,\mu_i) = s & \text{if customer remains in queue } i. \end{array}$$

The myth says that an optimal rate satisfies:

$$\lambda_i^*(s + \boldsymbol{e}_i) \le \lambda_i^*(s).$$

Weber and Stidham[23] call this property *transition-monotone*, and they prove it holds under certain assumptions. The intuition is that it is less costly to slow down the entrance of new arrivals if the queue grows.

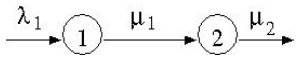
Counterexample. Weber and Stidham[23] provide the following:

$\lambda_1 \in [0, 0.01], \lambda_2 = 0,\ \mu_1 = 1, \mu_2 \in [0, 2]$,

$c_1 = c_2 = 0,\ r_1(\lambda_1) = 0.944\lambda_1$,

$$h_1(s_1) = \begin{cases} 0 & \text{if } s_1 = 0, 1, 2 \\ \infty & \text{if } s_1 > 2, \end{cases}$$

$$h_2(s_2) = \begin{cases} 0 & \text{if } s_2 = 0, 1, 2, 3, 4 \\ \infty & \text{if } s_2 > 4, \end{cases}$$



An optimal solution has $\mu_2^*(s) = 2$ for all s, and

$$\lambda^*(s) = \begin{cases} 0.01 & \text{if } s \in S \\ 0 & \text{otherwise,} \end{cases}$$

where $S = \{(0,0), (1,0), (0,1), (1,1), (1,2)\}$. This violates the myth with $s = (0,2)$ since

$$\lambda^*(1,2) \not\le \lambda^*(0,2).$$

DP Background — Infinite Horizon

The *infinite horizon* DP has two basic models:

$$\begin{array}{rrll} \text{Discounted:} & V_\beta(s,\pi) & = \lim_{n\to\infty} V_\beta^n(s,\pi) & \text{for } \beta < 1 \\ \text{Average:} & A(s,\pi) & = \lim_{n\to\infty} \dfrac{V_1^n(s,\pi)}{n}, & \end{array}$$

where

$V_\beta^n(s,\pi) = \sum_{t=1}^{n} \beta^{t-1} r_t(s_t, \pi_t(s_t))$

π is a policy — that is, a decision rule with $\pi_t(s_t) \in X_t(s_t)$

β is the discount factor $\in [0, 1]$.

For the deterministic model, the state transition is given by $s_{t+1} = T_t(s_t, \pi_t(s_t))$. The stochastic transition is given by:

$$\mathbf{Pr}[s_{t+1} = s | s_t, x] = q_t(s_t, s_{t+1}; x) \text{ for } x \in X_t(s_t).$$

The stochastic models use the expected returns in the objective, and the current state, s_t, is known at the time of the decision, $x = \pi_t(s_t) \in X_t(s_t)$. This is a *Markov decision process* with discrete time.

In words, V_β is the total discounted return when starting in state s and using policy π; A is the longterm, undiscounted average return. The former has an optimal solution under mild assumptions (but see DP Myth 11); the latter is approached under certain circumstances by letting $\beta \to 1$ from below.

The DP recursion for the discounted model is given by:

$$f_t(s) = \max_{x \in X_t(s)} \left\{ r_t(s,x) + \beta \textstyle\sum_{s'} q_t(s, s'; x) f_{t+1}(s') \right\}.$$

The deterministic model is included with

$$q_t(s, s'; x) = \begin{cases} 1 & \text{if } s' = T_t(s,x); \\ 0 & \text{otherwise.} \end{cases}$$

If the maximum exists for each state, an optimal policy is to let

$$\pi_t^*(s) \in \operatorname*{argmax}_{x \in X_t(s)} \left\{ r_t(s,x) + \beta \textstyle\sum_{s'} q_t(s, s'; x) f_{t+1}(s') \right\}.$$

In a stationary DP, the recursion may be regarded as *value iteration* whose limiting function is the solution to

$$f(s) = \max_{x \in X(s)} \left\{ r(s,x) + \beta \textstyle\sum_{s'} q(s, s'; x) f(s') \right\}.$$

The value function, f, is a fixed-point of the mapping, where $\beta < 1$ makes it a contractor.

Let V_β^* and A^* denote suprema values of the discounted and average-return models, respectively. The existence of optimal policies is not guaranteed, especially for the average-return model, where the limit may not exist. There are several variations of the DP objective for π^* to be an optimal policy:

B-opt:	$\exists \bar\beta \in (0,1) : V_\beta(s, \pi^*) \geq V_\beta^*(s)$	$\forall \beta \in (\bar\beta, 1)$;
nearly optimal:	$\lim_{\beta \to 1^-} \left(V_\beta(s, \pi^*) - V_\beta^*(s) \right) = 0$;	
1-optimal:	$\liminf_{\beta \to 1^-} \left(V_\beta(s, \pi^*) - V_\beta(s, \pi) \right) \geq 0$	$\forall \pi$;
discount ε-optimal:	$V_\beta(s, \pi^*) \geq V_\beta^*(s) - \varepsilon$	for $\varepsilon > 0$;
average ε-optimal:	$A(s, \pi^*) \geq A^*(s) - \varepsilon$	for $\varepsilon > 0$;
liminf average optimal:	$\liminf_{n \to \infty} \frac{1}{n} V_1^n(s, \pi^*) \geq \liminf_{n \to \infty} \frac{1}{n} V_1^n(s, \pi)$	$\forall \pi$;
limsup average optimal:	$\limsup_{n \to \infty} \frac{1}{n} V_1^n(s, \pi^*) \geq \limsup_{n \to \infty} \frac{1}{n} V_1^n(s, \pi)$	$\forall \pi$;
average-overtaking:	$\liminf_{N \to \infty} \frac{1}{N} \sum_{n=1}^{N} \left(V_1^n(s, \pi^*) - V_1^n(s, \pi) \right) \geq 0$	$\forall \pi$.

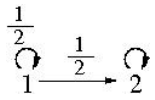
Each defining condition applies to all states, s. (See Lippman[12] and Flynn[8] for succinct introductions and how the objectives relate to each other.)

The remaining DP Myths pertain to infinite horizon, stationary DPs.

DP Myth 10. *If there is a nearly optimal solution to a discounted DP with a finite number of states and decisions, it is an optimal policy.*

Blackwell[3] provides the following:

Counterexample. The state space is $S = \{1, 2\}$, and the decision sets are $X(s) = \{1, 2\}$ for each $s \in S$. The return functions are $r(1, x) = x$ and $r(2, x) = 0$. The state transition probabilities are: $q(1, 1; 1) = q(1, 2; 1) = \frac{1}{2}$, $q(1, 2; 2) = 1$, and $q(2, 2; x) = 1$.



State transitions for $x = 1$

State transitions for $x = 2$

Let π^x be any policy for which $\pi^x(1) = x$ for $x = 1, 2$ Then, starting in state 1:

$$V_\beta(\pi^1) = 1 + \tfrac{1}{2}\beta + \tfrac{1}{4}\beta^2 + \cdots = \frac{2}{2 - \beta}$$
$$V_\beta(\pi^2) = 2.$$

Thus, π^2 is an optimal policy for each β, and $\lim_{\beta \to 1^-} V_\beta(\pi^1) = 2$. So, π^1 is nearly optimal but not optimal for any $\beta < 1$.

DP Myth 11. *For the discounted-return DP, there exists an ε-optimal policy for all $\varepsilon > 0$.*

Blackwell[4] provides the following:

Counterexample. Let the state space be the unit interval: $S = [0, 1]$. For each $s \in S$, the decision set is $X(s) = [0, 1]$, and the state remains unchanged: $T(s, x) = s$ for all $s \in S, x \in X(s)$. Let B be a Borel subset of $[0, 1]^2$, and let D be the projection of B on S. Choose B such that D is not a Borel set, and define the return function:

$$r(s, x) = \begin{cases} 1 & \text{if } (s, x) \in B; \\ 0 & \text{otherwise.} \end{cases}$$

An optimal policy, π^*, is such that $(s, \pi^*(s)) \in B$, so the optimal value is $V_\beta(s, \pi^*) = \frac{1}{1-\beta}$. For any other policy, π, there exists $s \in D$ such that $r(s, \pi(s)) = 0$, so

$$V_\beta(s, \pi) \leq \beta + \beta^2 + \cdots = \frac{\beta}{1 - \beta}.$$

Hence, there is no ε-optimal policy for $0 < \varepsilon < 1$.

DP Myth 12. *There exists a stationary policy that is B-opt.*

While this is true for finite decision sets and state space, it fails for non-finite decision sets. Maitra[13] proved that finiteness is not necessary if the objective is the discounted model — that is, V_β is well defined for any particular $\beta \in [0,1)$. He provided the following for this myth:

Counterexample. Let the state space be $\{1,2\dots\}$, and let the decision set be binary: $X = \{0,1\}$, independent of the state. The returns are $r(s,0) = c_s$ and $r(s,1) = 0$; and, the state transitions are $T(s,0) = s$ and $T(s,1) = s+1$. Choose $c_s > 0$ such that $\{c_s\} \uparrow c < \infty$ (for example, $c_s = c - \frac{1}{s}$).

Note that an advantage of choosing decision 0 is the positive immediate return, and the advantage of choosing decision 1 is the transition to the next state, which has a greater immediate return for the next decision. To illustrate, suppose $\pi^x(s) = x$ for all s. Then, $V_\beta(1,\pi^1) = 0$ and

$$V_\beta(1,\pi^0) = \sum_{t=1}^{\infty} \beta^{t-1} c_1 = \frac{c_1}{1-\beta}.$$

More generally, suppose $\pi(s) = 0$ for all $s \in S \neq \emptyset$ and $\pi(s) = 1$ for $s \notin S$. Then, with $S = \{s_1, s_2, \dots\}$,

$$V_\beta(1,\pi) = \sum_{t=1}^{\infty} \beta^{t-1} c_{s_1} = \frac{1-\beta^{s_1}}{1-\beta} c_{s_1}.$$

Here is an optimal (stationary) policy for any fixed β:

$$\pi(s) = \begin{cases} 0 & \text{if } \beta^k c_{s+k} < c_s \text{ for all } k \geq 1; \\ 1 & \text{otherwise.} \end{cases}$$

This defines s_1 in the above equation for $V_\beta(1,\pi)$ as

$$s_1 = \min\{s : \beta^k c_{s+k} < c_s \text{ for all } k \geq 1\}.$$

There is thus an optimal policy for each fixed $\beta \in (0,1)$. However, there is no B-opt policy, as Maitra proves by contradiction. Suppose π^* is B-opt. He first proves

$$\lim_{\beta \to 1^-} (1-\beta) V_\beta(1,\pi^*) = c.$$

An implication of this equation is that $\pi^*(s)$ selects decision 0 a finite number of times when in state s in order to advance to the greater immediate returns that converge to c.

Maitra constructs another policy, π', that contradicts the optimality of π^* by showing $V_\beta(1,\pi') > V_\beta(1,\pi^*)$ for all β sufficiently close to 1.

Maitra[14] later provided the following:

Counterexample. Let there be just one state, and let the decision set at each time period be given by $X = \{1,2,\dots\}$. Let the return function be $r(x) = 1 - \frac{1}{x}$, so there is no optimal policy.

Thus, non-finite state or decision sets can result in there being no B-opt policy.

DP Myth 13. *If the average-return DP has an optimal policy, there exists a stationary policy that is optimal.*

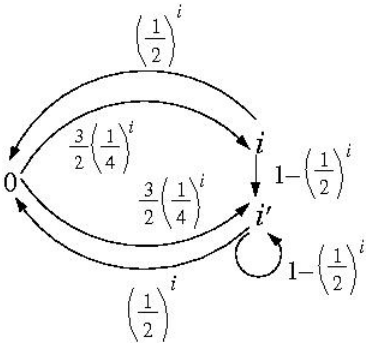
Fisher and Ross[6] provide the following:

Counterexample. Let $S = \{0, 1, 1', 2, 2', 3, 3', \ldots, \}$, $X(s) = \{1, 2\}$ for $s = 1, 2, 3 \ldots$, and $X(s) = \{1\}$ for $s = 0, 1', 2', 3', \ldots$. The return values are $r(0, x) = -1$ and $r(s, x) = 0$ for $s \neq 0$. The state transition probabilities are:

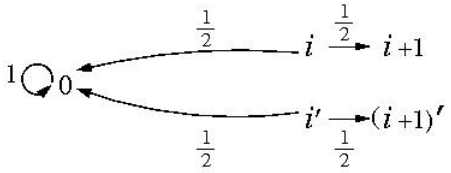
For $s = i > 0$: $q(0, i; 1) = q(0, i'; 1) = \frac{3}{2}\,(\frac{1}{4})^i$

For $s = i$: $q(i, 0; 1) = q(i', 0, 1) = (\frac{1}{2})^i = 1 - q(i, i'; 1) = 1 - q(i', i'; 1)$

For $s = i$: $q(i, 0; 2) = q(i, i + 1; 2) = \frac{1}{2}$



State transitions for $x = 1$



State transitions for $x = 2$

Let $M_{ij}(\pi)$ denote the expected number of periods to reach state j, starting in state i and following policy π. For example, suppose π always selects decision 2. Then,

$$M_{00}(\pi) = 1; \quad \text{for } s \neq 0\text{: } M_{0s}(\pi) = \infty \text{ and } M_{s0}(\pi) = \textstyle\sum_{j=1}^{\infty} j(\frac{1}{2})^j = 2.$$

Let π^m be the policy that selects decision 2 at states $0 < i < m$ and decision 1 otherwise. Then,

$$M_{00}(\pi^m) = 1 + \sum_{i=1}^{\infty} \tfrac{3}{2}\,(\tfrac{1}{4})^i\, M_{i0}(\pi^m) + \sum_{i=1}^{\infty} \tfrac{3}{2}\,(\tfrac{1}{4})^i\, 2^i.$$

Fisher and Ross derive the fact that $M_{00}(\pi^m) < 5$ for all m, and $\lim_{m\to\infty} M_{00}(\pi^m) = 5$.

Let π be any stationary policy, and let $p_i = \mathbf{Pr}(x = 1 | s = i)$ define a randomized policy. Then,

$$\begin{aligned} M_{00}(\pi; p) &= 1 + \sum_{j=1}^{\infty} \tfrac{3}{2}\,(\tfrac{1}{4})^j\, M_{j0}(\pi^m; p) + \sum_{j=1}^{\infty} \tfrac{3}{2}\,(\tfrac{1}{4})^j\, 2^j. \\ M_{j0}(\pi; p) &= \sum_{m=j}^{\infty} p_m \prod_{k=j}^{m-1} (1 - p_k)\, M_{j0}(\pi^m; p) + 2 \prod_{k=j}^{\infty} (1 - p_k) < (2 + 2^j). \end{aligned}$$

Hence, $M_{00}(\pi; p) < 5$ for all stationary policies randomized by p. They proceed to construct the following non-stationary policy and prove that it has the optimal value of 5 for

some $\{N_i\}$:

$$\pi_t(s) = \begin{cases} \pi^1(s) & \text{for} \quad 0 < t \le N_1 \\ \pi^2(s) & \text{for} \quad N_1 < t \le N_1 + N_2 \\ \vdots & \\ \pi^m(s) & \text{for} \quad \sum_{i=1}^{m-1} N_i < t \le \sum_{i=1}^{m} N_i \\ \vdots & \end{cases}$$

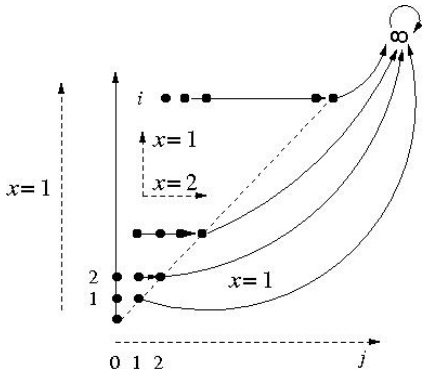
Ross[19] provides the theory needed to establish this, with application to the replacement process.

DP Myth 14. *There exists an average ε-optimal policy for $\varepsilon > 0$ that is optimal for the discounted-return model for β sufficiently close to 1.*

The intuition is that $\liminf_{\beta \to 1^-} V_\beta(s,\pi) = A(s,\pi)$, but Ross[19] provides the following:

Counterexample. Let $S = \{(i,j) : 0 \le j \le i,\ i \ge 1\} \cup \{\infty\}$ and $X(s) = \{1,2\}$ for $s = (i,0)$, $X(s) = \{1\}$ for $s = (i, j > 0), \infty$. The state transitions are deterministic:

$$\begin{aligned} T((i,0),1) &= (i+1,0) \\ T((i,0),2) &= (i,1) \\ T((i,j),1) &= (i,j+1) \quad \text{for } 0 < j < i \\ T((i,i),1) &= \infty = T(\infty,1). \end{aligned}$$



Starting at state $(i,0)$, one can move upward ($x = 1$) to some point, say $(i+h,0)$, then go right ($x = 2$). There is no choice from that point; after reaching the 45° line (where $j = i + h$), one jumps to ∞ and stays there.

The immediate returns are $r(s,x) = 0$, except $r((i,0),x) = 2$ and $r(\infty,x) = 1$. Suppose we start at state $(1,0)$. Letting π^1 be the policy that always selects decision 1, there is no right turn, so the average return is 2. Otherwise, the average return is 1 (reaching ∞ and staying there forever).

Letting π be the policy that selects action 2 at state $(n,0)$ and decision 1 otherwise, we have:

$$\begin{aligned} V_\beta((1,0),\pi) &= \sum_{k=0}^{n-1} \beta^k + 2 \sum_{k=2n}^{\infty} \beta^k = \frac{1 - \beta^n + 2\beta^{2n}}{1-\beta} \\ &< \frac{1}{1-\beta} \text{ for } n \text{ sufficiently large} \\ &= V_\beta((1,0),\pi^1). \end{aligned}$$

Hence, for $\beta \in (0,1)$, $V_\beta((1,0),\pi^1) \neq V_\beta^*((1,0))$. This implies that the discounted-return model optimum does not approach the optimal average return, so it cannot become average ε-optimal.

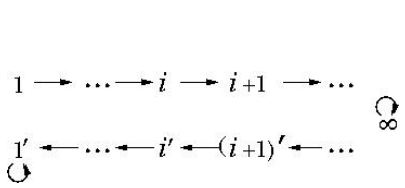
DP Myth 15. *If the optimal average-return is finite, there is an ε-optimal stationary policy for any $\varepsilon > 0$.*

Ross[20] provides the following:

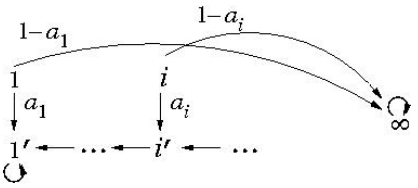
Counterexample. Let the state space be $S = \{1, 1', 2, 2', \ldots, i, i', \ldots, \infty\}$. For each $i \in S$, the decision set is binary: $X(i) = \{0, 1\}$. For each $i' \in S$, the decision set is a singleton: $X(i') = \{0\}$. The state-transition probabilities are:

$$\begin{array}{ll} \text{For } s = i: & q(i, i+1; 0) = 1,\ q(i, i'; 1) = a_i = 1 - q(i, \infty; 1), \\ \text{For } s = i': & q(i', (i-1)'; 0) = 1 \text{ for } i \geq 2,\ q(1', 1; 0) = 1, \\ \text{For } s = \infty: & q(\infty; \infty, s) = 1 \text{ for all } s \in S, \end{array}$$

where a_i satisfy: $0 < a_i < 1$ and $\prod_{i=1}^{\infty} a_i = \frac{3}{4}$.



State transitions for $x = 0$



State transitions for $x = 1$

The returns are $r(i, x) = 2$ and $r(i', x) = 0$ for all $i, i' \in S$ and all x.

Let the initial state be $s = 1$. Then, every stationary policy has a return of 2 in all but a finite number of time periods. This implies (by the bounded convergence theorem) that the average expected return is 2.

Let π be a (non-stationary) policy such that:

$$\begin{aligned} \pi_1(1) &= 1 \\ \pi_t(1') &= 0 \text{ for } t = 2, \ldots, T \\ \pi_{T+1}(1') &= 1. \end{aligned}$$

The average return equals:

$$\begin{array}{lll} 2 \text{ with probability} & 1 - \prod_{i=1}^{\infty} a_i & = \frac{1}{4} \\ 1 \text{ with probability} & \prod_{i=1}^{\infty} a_i & = \frac{3}{4}. \end{array}$$

Hence, the expected average return is $\frac{1}{2} + \frac{3}{4} = \frac{5}{4}$, so there is no ε-optimal stationary policy for $\varepsilon < \frac{3}{4}$.

DP Myth 16. *If a policy is B-opt, it optimizes the average return.*

Flynn[7] proved this for finite state spaces and provides the following for a non-finite state space:

Counterexample. Let $\{s_j\}_{j=1}^{\infty}$ be a real sequence such that

$$s^* \stackrel{\text{def}}{=} \liminf_{\beta \to 1^-} (1-\beta) \sum_{j=1}^{\infty} \beta^{j-1} s_j \;>\; \liminf_{n \to \infty} \frac{\sum_{j=1}^{n} s_j}{n} \stackrel{\text{def}}{=} s_*.$$

(Flynn establishes existence by an earlier theorem.) Let the state space be $\{0, s_1, s_2, \dots\}$. The decision sets are binary, independent of the state: $X(s) = \{0,1\}$ for all s. The state transition functions are deterministic: $T(s_j, x) = s_{j+1}$, $T(0,0) = 0$, and $T(0,1) = s_1$. The immediate returns are independent of the decision: $r(s_j, x) = s_j$ and $r(0,x) = \frac{1}{3}(s^* + 2s_*)$.

Let π^x denote a policy that always selects $x \in \{0,1\}$. We have $V_\beta(s, \pi^1) = V_\beta(s, \pi)$ for $s \neq 0$. For $\pi(0) = 0$,

$$(1-\beta) V_\beta(0,\pi) = \tfrac{1}{3}(s^* + 2s_*) < s^* = (1-\beta) \sum_{j=1}^{\infty} \beta^{j-1} s_j = (1-\beta) V_\beta(0, \pi^1).$$

Hence, $V_\beta(0,\pi) < V_\beta(0,\pi^1)$, so π^1 is B-opt. However, we also have

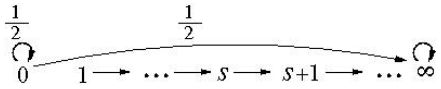
$$A(0,\pi^0) = \tfrac{1}{3}(s^* + 2s_*) > s_* = A(0,\pi^1),$$

so π^1 does not maximize the average return, starting in state 0.

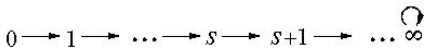
DP Myth 17. *If a stationary policy is B-opt, it is average-overtaking.*

This is true for finite state spaces, and Flynn[8] provides the following:

Counterexample. Let $S = \{0, 1, \dots, \infty\}$ and $X(s) = \{0,1\}$ for all $s \in S$. State transitions are stochastic only for $s = 0$ and $x = 0$: $q(0,0;0) = q(\infty, 0; 0) = \frac{1}{2}$. Otherwise, the transition is deterministic: $T(0,1) = 1$ and $T(s,x) = s+1$ for all $s > 0$, $x \in X(s)$ (note: $T(\infty, x) = \infty$).



State transitions for $x = 0$



State transitions for $x = 1$

Flynn establishes the existence of a sequence $\{a_j\}_{j=1}^{\infty}$ that satisfies:

$$\liminf_{N \to \infty} \frac{1}{N} \sum_{n=1}^{N} \sum_{j=1}^{n} a_j = -1 \tag{DP.4}$$

$$\liminf_{\beta \to 1^-} \sum_{j=1}^{\infty} \beta^{j-1} a_j = \limsup_{N \to \infty} \frac{1}{N} \sum_{n=1}^{N} \sum_{j=1}^{n} a_j = 0 \tag{DP.5}$$

Using this sequence, the returns are defined as: $r(s,x) = a_{s+1}$ for $s > 0$; $r(0,1) = a_1$, $r(0,0) = -\frac{1}{4}$, and $r(\infty, x) = 0$.

Let π^x denote the policy that always selects x. A B-opt policy is π^1 because $V_\beta^*(s) = V_\beta(s,\pi^1)$ for all s. (For $s = 0$, $V_\beta(0,\pi^1) = 0$ from (DP.5), whereas if $\pi^*(0) = 0$, $V_\beta(0,\pi^*) = \frac{-1/4}{1-\beta}$.) However, π^1 is not average-overtaking because the defining inequality fails for $s = 0$:

$$\liminf_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \sum_{j=1}^{n} \left(V_1^n(0,\pi^1) - V_1^n(0,\pi^0) \right) = \liminf_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \sum_{j=1}^{n} a_j + \tfrac{1}{2} = -\tfrac{1}{2}.$$

The last step uses (DP.4).

DP Myth 18. *Every stationary, 1-optimal policy is average-overtaking.*

Flynn[8] establishes this for finite state spaces and provides the following:

Counterexample. Let $S = \{0, 1, 2, \dots\}$ and $X(s) = \{0, 1\}$ for all $s \in S$. The state transitions are deterministic: $T(0,0) = 0$, $T(0,1) = 1$, $T(s,x) = s + 1$ for $s > 0$.

State transitions for $x = 0$ State transitions for $x = 1$

The immediate returns are $r(0,0) = 0$, $r(0,1) = a_1$, and $r(s,x) = a_{s+1}$ for $s > 0$.

Let π^x denote the policy that selects x each time. Flynn proves that π^0 is both 1-optimal and average-overtaking, whereas π^1 is 1-optimal but not average-overtaking.

For $s > 0$, $V_\beta(s,\pi^0) = V_\beta(s,\pi^1) = V_\beta(s,\pi)\,\forall\pi$, so the defining inequality for 1-optimal is valid. Now consider $s = 0$. If $\pi^*(0) = 0$, $V_\beta(s,\pi^0) = V_\beta(s,\pi), \forall\pi$ and π^0 is optimal. Otherwise, applying (DP.5), we have:

$$\lim_{\beta\to 1^-} \left(V_\beta(0,\pi^0) - V_\beta(0,\pi) \right) = -\lim_{\beta\to 1^-} \sum_{j=1}^{\infty} \beta^{j-1} a_j = 0.$$

Hence, π^0 is 1-optimal. Similarly, if $\pi^*(0) = 1$, $V_\beta(s,\pi^1) = V_\beta(s,\pi)$ and π^1 is optimal. Otherwise, we have:

$$\lim_{\beta\to 1^-} \left(V_\beta(0,\pi^1) - V_\beta(0,\pi) \right) = \lim_{\beta\to 1^-} \sum_{j=1}^{\infty} \beta^{j-1} a_j = 0.$$

Hence, π^1 is 1-optimal.

For any policy, π, $V_1^n(s,\pi) = V_1^n(s,\pi^0) = V_1^n(s,\pi^1)$ for $s > 0$. Consider $s = 0$. If $\pi(0) = 0$, π^1 is not average-overtaking because (DP.4) yields:

$$\liminf_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \left(V_1^n(0,\pi^1) - V_1^n(0,\pi) \right) = -1.$$

Whereas, if $\pi(0) = 1$,

$$\liminf_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \left(V_1^n(0,\pi^0) - V_1^n(0,\pi)\right) = 1,$$

so π^0 is average-overtaking.

DP Myth 19. *If a policy is both B-opt and average-overtaking, it is liminf average optimal.*

The assertion is true for finite state spaces, and Flynn[8] provides the following:

Counterexample. Let the state space, decision sets, and state transition functions be the same as in DP Myth 18, but with returns: $r(0,x) = 0$ and $r(s,x) = v_s$ for $s > 0$, where Flynn establishes the existence of $\{v_j\}_{j=1}^{\infty}$ that satisfies:

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \sum_{j=1}^{n} v_j = \infty, \qquad \liminf_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} v_j < 0. \tag{DP.6}$$

Let π^x denote the policy that always selects decision x. π^1 is average-overtaking because $\forall\pi$:

$$\begin{aligned} s > 0 &\to V_1^n(s,\pi^1) = V_1^n(s,\pi) \\ s = 0, \pi(0) = 1 &\to V_1^n(0,\pi^1) = V_1^n(0,\pi) \\ s = 0, \pi(0) = 0 &\to V_1^n(0,\pi) = 0 \text{ and } V_1^n(0,\pi^1) = \sum_{j=1}^{n} v_j > 0 \text{ for } n \text{ sufficiently large.} \end{aligned}$$

Further, π^1 is B-opt because $\forall\pi$:

$$\liminf_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} V_1^n(s,\pi) \le \liminf_{\beta\to 1^-} V_\beta(s,\pi) \le \limsup_{\beta\to 1^-} V_\beta(s,\pi) \le \limsup_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} V_1^n(s,\pi).$$

However, π^1 is not liminf average optimal because (DP.6) yields:

$$\liminf_{n\to\infty} \frac{1}{n} V_1^n(0,\pi^1) = \liminf_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} v_j < 0 = \liminf_{n\to\infty} \frac{1}{n} V_1^n(0,\pi^0).$$

DP Myth 20. *If a policy is both B-opt and average-overtaking, it is limsup average optimal.*

The assertion is true for finite state spaces, and Flynn[8] provides the following:

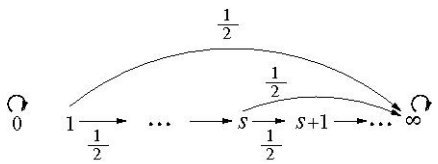
Counterexample. Let the state space, decision sets, and state transition functions be the same as in DP Myth 19, but with returns: $r(0,x) = 0$ and $r(s,x) = -v_s$ for $s > 0$, satisfying (DP.6). Using a similar proof, π^0 is both B-opt and average-overtaking, but it is not limsup average optimal because:

$$\limsup_{n\to\infty} \frac{1}{n} V_1^n(0,\pi^0) = 0 < -\liminf_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} v_j = \limsup_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} -v_j = \limsup_{n\to\infty} \frac{1}{n} V_1^n(0,\pi^1).$$

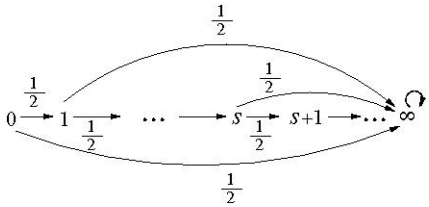
DP Myth 21. *If a policy is B-opt among stationary policies, it optimizes the average return among stationary policies.*

Flynn[8] establishes this for finite state spaces and provides the following:

Counterexample. Let $S = \{0, 1, \ldots, \infty\}$ and $X(s) = \{0, 1\}$ for all $s \in S$. State transitions are: $q(\infty, \infty; x) = q(0, 0; 1) = 1$, $q(0, 1; 1) = q(0, \infty; 1) = \frac{1}{2}$, and $q(s, s+1; x) = q(s, \infty; x) = \frac{1}{2}$ for $s > 0$.



State transitions for $x = 0$



State transitions for $x = 1$

Letting $\{v_j\}$ satisfy (DP.6), define the returns independent of the decisions: $r(0, x) = r(\infty, x) = 0$ and $r(s, x) = -2^s v_s$ for $s > 0$ ($s \neq \infty$).

Under any stationary policy, the system is absorbed in state ∞ with probability 1. Following the same arguments as in the counterexample to DP Myth 19, Flynn proves π^0 is B-opt, but not limsup average optimal.

DP Myth 22. *If a policy is average-overtaking among stationary policies, it optimizes the average return among stationary policies.*

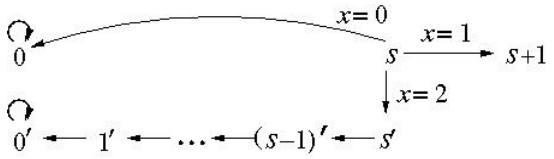
Flynn[8] establishes this for finite state spaces and provides the following:

Counterexample. Let $S = \{0, 1, \ldots, \infty\}$ and $X(s) = \{0, 1\}$ for all $s \in S$. State transition functions are as in DP Myth 21, but the immediate returns are $r(s, x) = 2^{s+1} a_{s+1}$, $r(\infty, x) = 0$, $r(0, 0) = -\frac{1}{4}$, and $r(0, 1) = 2a_1$, where $\{a_j\}$ satisfies (DP.4). Using the same arguments as in DP Myth 17, π^1 is B-opt, but not average-overtaking.

DP Myth 23. *We can approximate an infinite horizon Markov decision process with a sufficiently long, finite horizon.*

Hinderer[10] first raised this issue for both discounted and average return models. Flynn[9] provides the following:

Counterexample. Let $S = \{0, 0', 1, 1', 2, 2', \ldots\}$ and $X(s) = \{0, 1, 2\}$ for all $s \in S$. The state transitions are deterministic: $T(s, 0) = 0$, $T(s, 1) = s + 1$, $T(s, 2) = s'$, $T(s', x) = (s - 1)'$ for $s = 1, 2, \ldots$.

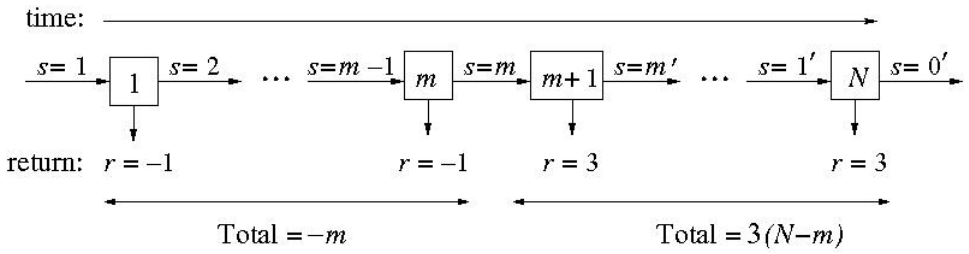


The immediate returns are $r(0,x)$, $r(s,x) = -1$ for $s = 0', 1, 2, \ldots$, and $r(s,x) = 3$ for $s = 1', 2', \ldots$

Let π^0 be the policy of always selecting $x = 0$, and note that the infinite horizon solution is π^0 with $A^*(s) = 0$ for all s. Let x^N be an optimal policy for N time periods, and let $m = \lfloor \frac{N}{2} \rfloor$. Then,

$$x^N(s) = \begin{cases} 1 & \text{if } s \le m; \\ 2 & \text{if } s > m. \end{cases}$$

So, starting in state 1, we have $V_1^N(1, x^N) = N$ if N is even, and $V_1^N(1, x^N) = N + 2$ if N is odd.



Optimal State and Return Sequences for N-period Horizon

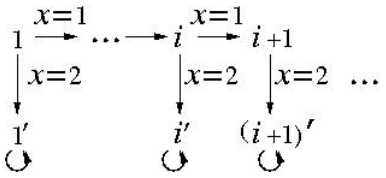
Consider x^N as a finite approximation for the infinite horizon. As N becomes large, $\{V_1^N(1, x^N) - V_1^N(1, \pi^0)\} \uparrow \infty$, so x^N is a poor approximation. From the other view, π^0 becomes increasingly less desirable as an approximation to the N-horizon DP as $N \to \infty$. Moreover, the average return for the N-horizon approaches 1, whereas the average return for the infinite horizon DP is 0.

DP Myth 24. *A discounted-return stationary DP with finite decision space for each state has a pure, stationary policy that is optimal.*

Hordijk and Tijms[11] provide the following:

Counterexample. Let the state space be given by the denumerable set, $S = \{1, 1', 2, 2', \ldots\}$.

Let the decision sets be $X(s) = \{1, 2\}$ for $s = 1, 2, \ldots$ and $X(s) = \{1\}$ for $s = 1', 2', \ldots$ Let the state transitions be $T(i,1) = i{+}1$ and $T(i,2) = T(i',1) = i'$ for $i = 1, 2, \ldots$ The immediate returns are $r(s,1) = 0$ for all s and $r(i,2) = \beta^{-i}\left(1 - \frac{1}{i}\right)$ for $i = 1, 2, \ldots$



State Transitions

There are two pure, stationary policies:

$$\pi^1(s) = 1 \text{ and } \pi^1(s') = 1$$
$$\pi^2(s) = 2 \text{ and } \pi^2(s') = 1$$

Then,

$$\begin{array}{ll} V_\beta(s,\pi^1) & = 0 \qquad\qquad\qquad\qquad\qquad\qquad\quad \text{for all } s \\ V_\beta(i,\pi^2) & = \beta^{-i}\left(1-\frac{1}{i}\right) \text{ and } V_\beta(i',\pi^2)=0 \quad \text{for } i=1,2,\ldots \end{array}$$

π^2 is optimal among pure, stationary policies, and $V_\beta(i,\pi^2) < \beta^{-i}$ for all $i = 1, 2\ldots$

Now consider the following randomized policy: $P_i(t)$ = probability that $\pi_t(i) = 1$ when the system is in state i at time t. Suppose $P_i(t) < 1$ for at least one t. Then,

$$V_\beta(i,\pi) = \sum_{t=0}^{\infty} \beta^t (1-P_i(t)) \beta^{-(i+t)} \prod_{n=0}^{t-1} P_i(n) \left(1-\frac{1}{i+t}\right).$$

This yields

$$V_\beta(i,\pi) = \beta^{-i} \sum_{t=0}^{\infty} \beta^t (1-P_i(t)) \prod_{n=0}^{t-1} P_i(n) \left(1-\frac{1}{i+t}\right).$$

Using the identity

$$\sum_{t=0}^{\infty} \beta^t (1-P_i(t)) \prod_{n=0}^{t-1} P_i(n) = 1 - \prod_{t=0}^{\infty} P_i(t),$$

we obtain

$$V_\beta(i,\pi) < \beta^{-i}.$$

Consider the policy with

$$\pi(i) = 1 \text{ for } i = 1,\ldots,m-1 \text{ and } \pi(i) = 2 \text{ for } i = m, m+1,\ldots$$

Then, $V_\beta(i,\pi) = \beta^{-i}\left(1-\frac{1}{m}\right)$ for all $i \geq 1$. Letting $m \to \infty$, we see that

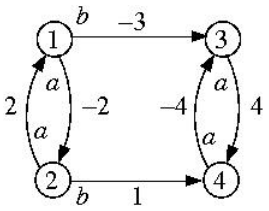
$$\sup_\pi V_\beta(i,\pi) = \beta^{-i}.$$

Thus, the supremum cannot be achieved (finitely), so there is no optimal policy.

DP Myth 25. new *Denardo's policy improvement algorithm computes a 1-optimal policy.*

Counterexample. O'Sullivan[17, Appendix A] provides the following: Let $S = \{1,2,3,4\}$, $X(1) = X(2) = \{a,b\}$, and $X(3) = X(4) = \{a\}$. The state transitions are deterministic:

$$\begin{array}{ll} T(1,a)=2 & r(1,a)=-2 \\ T(1,b)=3 & r(1,b)=-3 \\ T(2,a)=1 & r(2,a)=\ \ 2 \\ T(2,b)=4 & r(2,b)=\ \ 1 \\ T(3,a)=4 & r(3,a)=\ \ 4 \\ T(4,a)=3 & r(4,a)=-4 \end{array}$$



The only 1-optimal policy is $\pi^a = (a,a,a,a)$ because

$$\liminf_{\beta\to 1^-} \left(V_\beta(s,\pi^a) - V_\beta(s,\pi)\right) = \begin{cases} \frac{2}{1+\beta} - 1 > 0 & \text{if } \pi(s) = b \text{ and } s = 1,2 \\ 0 & \text{otherwise.} \end{cases}$$

Starting with $\pi^0 = (b, b, a, a)$, the algorithm computes a solution to $Q^0 v^1 = v^0$, where $Q^0 = P(\pi^0) - I$ ($P(\pi^0)$ is the state transition probability matrix using policy π^0), and

$$v^0 = \lim_{\beta \to 1^-} V_\beta(\bullet, \pi^0) = \lim_{\beta \to 1^-} \begin{pmatrix} -3 + \frac{4\beta}{1+\beta} \\ 1 + \frac{4\beta}{1+\beta} \\ \frac{4\beta}{1+\beta} \\ -\frac{4\beta}{1+\beta} \end{pmatrix} = \begin{pmatrix} -1 \\ 3 \\ 2 \\ -2 \end{pmatrix}.$$

Such a solution is given by $v^1 = (1, -1, 0, 2)^\intercal$:

$$\begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{pmatrix} 1 \\ -1 \\ 0 \\ 2 \end{pmatrix} = \begin{pmatrix} -1 \\ 3 \\ 2 \\ -2 \end{pmatrix}.$$

Then, π is the unique solution to the associated maximum-reward-rate problem. Finally, the algorithm seeks a policy that is transient on states where π is transient — in particular, states 1 and 2. This means $\pi(s) \neq a$ because states 1 and 2 are recurrent under a.

DP References

[1] E. J. Anderson. A note on the dynamic lot-size model with uncertainty in demand and supply processes. *Management Science*, 35(5):635–640, 1989.

[2] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

[3] D. Blackwell. Discrete dynamic programming. *The Annals of Mathematical Statistics*, 33(2):719–726, 1962.

[4] D. Blackwell. Discounted dynamic programming. *The Annals of Mathematical Statistics*, 36(1):226–235, 1965.

[5] A. Federgruen and C. Lee. The dynamic lot size model with quantity discount. *Naval Research Logistics*, 37:707–713, 1990.

[6] L. Fisher and S. M. Ross. An example in denumerable decision processes. *The Annals of Mathematical Statistics*, 39(2):674–675, 1968.

[7] J. Flynn. Averaging versus discounting in dynamic programming: A counterexample. *Annals of Statistics*, 2(2):411–413, 1974.

[8] J. Flynn. Conditions for the equivalence of optimality criteria in dynamic programming. *Annals of Statistics*, 4(5):936–953, 1976.

[9] J. Flynn. On optimality criteria for dynamic programs with long finite horizons. *Journal of Mathematical Analysis and Applications*, 76(1):202–208, 1980.

[10] K. Hinderer. Estimates for finite-stage dynamic programs. *Journal of Mathematical Analysis and Applications*, 55(1):207–238, 1976.

[11] A. Hordijk and H. C. Tijms. A counterexample in discounted dynamic programming. *Journal of Mathematical Analysis and Applications*, 39(2):455–457, 1972.

[12] S. A. Lippman. Criterion equivalence in discrete dynamic programming. *Operations Research*, 17(5):920–923, 1969.

[13] A. Maitra. Dynamic programming for countable state systems. *Sankhyā, The Indian Journal of Statistics, Series A*, 27:241–248, 1965.

[14] A. Maitra. A note on undiscounted dynamic programming. *The Annals of Mathematical Statistics*, 37(4):1042–1044, 1966.

[15] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, New York, NY, 1990.

[16] L. G. Mitten. Composition principles for synthesis of optimal multistage processes. *Operations Research*, 12(4):610–619, 1964.

[17] M. J. O'Sullivan. *New Methods for Dynamic Programming Over an Infinite Time Horizon*. Ph. D. thesis, Stanford University, Stanford, CA, 2002.

[18] E. Porteus. An informal look at the principle of optimality. *Management Science*, 21(11):1346–1348, 1975.

[19] S. M. Ross. Non-discounted denumerable Markovian decision models. *The Annals of Mathematical Statistics*, 39(2):412–423, 1968.

[20] S. M. Ross. On the nonexistence of ε-optimal randomized stationary policies in average cost Markov decision models. *The Annals of Mathematical Statistics*, 42(5):1767–1768, 1971.

[21] J-S. Song. The effect of leadtime uncertainty in a simple stochastic inventory model. *Management Science*, 40(5):603–613, 1994.

[22] J-S. Song. On the order fill rate in a multi-item, base-stock inventory system. *Operations Research*, 46(6):831–845, 1998.

[23] R. R. Weber and S. Stidham, Jr. Optimal control of service rates in networks of queues. *Advances in Applied Probability*, 19(1):202–218, 1987.

[24] J. Xu and L. L. Lu. The dynamic lot size model with quantity discount: Counterexamples and correction. *Naval Research Logistics*, 45:419–422, 1998.

[25] W. I. Zangwill. From EOQ towards ZI. *Management Science*, 33(10):1209–1223, 1987.

Nonlinear Programming

A nonlinear program (NLP) has the general form:

$$\max\ f(x) :\ x \in X,\ g(x) \leq 0,\ h(x) = 0,$$

where $\emptyset \neq X \subseteq \mathbb{R}^n$, $f : X \to \mathbb{R}$, $g : X \to \mathbb{R}^m$, $h : X \to \mathbb{R}^M$. This is the most general form with no restrictions on the domain (X) or the nature of the functions, except to have at least one of the functions be nonlinear. Historically, the domain is a simple convex set, like all of $\mathbb{R}^n$ or $\mathbb{R}^n_+$. If the functions are differentiable, methods of calculus are used to establish optimality conditions and provide a foundation for algorithm design.

We refer to some special NLPs:

Convex (CP). X is closed and convex, f is concave, g is convex, and h is affine.

Quadratic (QP). $\max\ x^\intercal Qx + cx :\ Ax \leq b$.

Typically, Q is assumed to be symmetric, but this is no restriction because the same quadratic form results upon replacing Q with $\frac{1}{2}(Q + Q^\intercal)$.

NLP Myth 1. *If f has continuous n^{th}-order derivatives, local behavior of f can be approximated by Taylor's series:*

$$f(x+th) = f(x) + t\nabla f(x)h + \tfrac{1}{2}t^2 h^\intercal \nabla^2 f(x)h + \ldots,$$

where h is a vector with $||h|| = 1$, t is a scalar, $\nabla f(x)$ is the gradient of f, and $\nabla^2 f(x)$ is the hessian of f.

The reason this is not correct is that, although Taylor's series might converge under the stated assumptions, it need not be to the correct value.

Counterexample. Let $f(x) = e^{-\frac{1}{x^2}}$ for $x \neq 0$, and $f(0) = 0$ (x is a scalar). It can be shown that f is infinitely differentiable everywhere. At $x = 0$, the n^{th} derivative is 0 for all n. Thus, the Taylor series converges to 0, which gives the approximation, $f(t) = 0$ (with $h = 1$) for all t. This is incorrect for $t \neq 0$.

This myth is used all too often in textbooks. The correct assumption is that f is *analytic*. Then, by definition, the Taylor's series does converge to the correct value of the function, so it can be used for approximation when proving theorems — viz., that necessary conditions for x to be an unconstrained minimum are: $\nabla f(x) = 0$ and $h^\intercal \nabla^2 f(x)h \geq 0$ for all h.

NLP Myth 2. *Given differentiable functions, an optimal point must satisfy the Lagrange Multiplier Rule.*

We are given

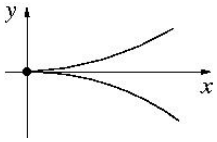
$$\text{NLP: } \max\ f(x) : x \in \mathbb{R}^n,\ g(x) \leq 0,\ h(x) = 0,$$

where f, g, h are differentiable functions on $\mathbb{R}^n$. For just equality constraints (g vacuous), the Lagrange Multiplier Rule (LMR) states: x^* is optimal only if there exists λ such that:

$$\nabla f(x^*) - \lambda \nabla h(x^*) = 0.$$

Counterexample. Consider $\max\ -x : x^3 - y^2 = 0$. The optimum is at $(x^*, y^*) = (0, 0)$. The LMR requires $-1 - \lambda\, 0 = 0$ for some λ, which is impossible.

cusp: $h(x, y) = x^3 - y^2$



The LMR for equality constraints is valid with the *constraint qualification*: $\nabla h(x^*)$ has full row rank. This is what Lagrange assumed, using the Implicit Function Theorem to prove the necessity of the LMR. (Affine functions need no constraint qualification.)

One extension of the LMR to include inequality constraints is simple: there exists λ, μ such that:

$$\mu \geq 0,\ \mu_i > 0 \rightarrow g_i(x^*) = 0$$
$$\nabla f(x^*) - \mu \nabla g(x^*) - \lambda \nabla h(x^*) = 0$$

The extended Lagrange constraint qualification is simply

$$\text{rank}\left(\begin{bmatrix}\nabla g_A(x^*)\\ \nabla h(x^*)\end{bmatrix}\right) = |A| + M,$$

where A is the set of *active constraints* among the inequalities — that is, $A = \{i : g_i(x^*) = 0\}$ — and M is the number of equality constraints.

The classical extension and deeper meaning into saddlepoint equivalence by Kuhn and Tucker[28] gave a weaker constraint qualification, but it is violated by the following:

Counterexample. max $x : x \geq 0,\ y - (1-x)^3 \leq 0,\ -y \leq 0$. The solution is at $(1,0)$. The LMR requires $(\mu_1, \mu_2) \geq 0$ to satisfy:

$$1 - \mu_1\, 3(1-x)^2 = 0$$
$$-\mu_1 + \mu_2 = 0$$

The first equation is impossible at $(1,0)$.

Here is another counterexample with g convex: max $x : x^2 \leq 0$.

NLP Myth 3. *A point that satisfies the Lagrange multiplier necessary conditions is a local optimum.*

Most people know this is a myth because the Lagrange (a.k.a., Kuhn-Tucker-Karush) conditions hold at stationary points that are not minima or maxima (for example, at a saddle point). This is included here, however, because it appears too often in textbooks and even some research articles. Those not expert in mathematical programming are told that an algorithm converges to a local optimum when, in fact, it converges to a point that satisfies the Lagrange multiplier conditions. (Methods of descent can rule out converging to a "pessimal" solution — that is, to a max when seeking a min — if it moves from its initial point.)

Counterexample. min $x^2 - y^2 : -1 <= x, y <= 1$. A Lagrange point is at $(x, y) = (0, 0)$ with all four multipliers = 0, but this is not a local min (or max) of the NLP. It is a saddlepoint.

NLP Myth 4. *Suppose f is analytic and x is a minimum of f. Then, $\nabla f(x) = 0$, and if $h^\intercal \nabla^2 f(x) h = 0$ for all h, it is necessary that all terms of the third derivative shall vanish. In that case, if the fourth-order term is positive, the point is a minimum.*

This is a classical error, made by the great Lagrange. A complete discussion is given by Hancock[25]. (Qi[35] provides a qualification that makes an "extended Lagrange claim" valid.)

The proposition is a natural extension of the (correct) result for one variable: *the first non-vanishing derivative must be of even order; and, it is positive for a minimum and negative for a maximum.* For two variables, however, we have a problem with the ambiguous case.

For notational convenience, translate the solution to the origin, and suppose $f(0) = 0$. Then, Taylor's expansion is:

$$f(h,k) = \tfrac{1}{2}(Ah^2 + 2Bhk + Ck^2) + 3^{\text{rd}}\text{-order terms},$$

where the quadratic form coefficients (A, B, C) are the associated second partial derivatives of f, evaluated at (0,0). When $B^2 - 4AC > 0$, the origin is a proper (local) minimum; when $B^2 - 4AC < 0$, the origin is a proper (local) maximum. The ambiguous case, when $B^2 - 4AC = 0$, is at issue. Here is where Lagrange claimed the 3^{rd}-order term must vanish, and that the sign of the 4^{th}-order term (if it does not vanish) can then determine whether the point is a (local) minimum or a maximum. The following is a special case of a counterexample found by Peano.

Counterexample. $f(x, y) = (y - x^2)(y - 2x^2)$. We have first derivatives: $f_x = -6xy + 8x^3$ and $f_y = 2y - 3x^2$. These vanish at (0,0), so we proceed to the second derivatives:

$$\nabla^2 f(x, y) = \begin{bmatrix} -6y + 24x^2 & -6x \\ -6x & 2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix} \text{ at } (0, 0).$$

This is the ambiguous case, where the hessian is positive semi-definite at the origin. Let the change in the y-direction be zero, and let the change in x be t, so the quadratic form is $(t, 0)\nabla^2 f(0)(t, 0)^\intercal = 0$ for all t. We proceed to third derivatives, but since we maintain no change in the y-direction, we need to compute derivatives of only x:

$$f_{Ben-AyedBlair90x} = 48x \text{ and } f_{Ben-AyedBlair90Ben-AyedBlair90} = 48,$$

so $f(t, 0) = 48t^4 > 0$ for all t. According to the myth, this implies f achieves a minimum at $(0, 0)$; however, consider $y = {}^3\!/_2\, x^2$. Along this parabola, $f(x, y) = -{}^1\!/_4\, x^4$, which is negative for $x \neq 0$. Thus, $(0, 0)$ is not a local minimum of f.

NLP Myth 5. *Given* $\min\{f(x, y) = g(x) + h(y) : ay = g(x)\}$*, we can equivalently solve* $\min\{ay + h(y)\}$.

Counterexample. The following is given by Bloom[4]. Determine the shortest distance from the point $(0, 5)$ to the parabola defined by $16y = x^2$. Using the square distance as the objective function, our problem is:

$$\min\ x^2 + (y - 5)^2 : 16y = x^2.$$

Substituting x^2, the unconstrained "equivalent" is given by:

$$\min\ 16y + (y - 5)^2.$$

The only critical point (where $f' = 0$) is at $y = -3$. However, this produces an imaginary value of x, so the minimum does not exist. The problem is that we cannot simply replace x^2 with $16y$; we must divide the problem into the cases: $x \geq 0$ and $x \leq 0$.

The Lagrange Multiplier rule does not run into any problem. The Lagrange conditions for the original problem are:

$$2x + \lambda 2x = 0,\ 2(y - 5) - 16\lambda = 0,\ \text{and } 16y = x^2.$$

With $x = 0$ we obtain $y = 0 \Rightarrow \lambda = -\frac{5}{8}$. With $x \neq 0$, we obtain $\lambda = -1 \Rightarrow y = -3 \Rightarrow$ contradiction. Hence, the minimum is at $(0, 0)$.

NLP Myth 6. new *A smooth surface with one critical point that is a local, but not a global, minimum must have a second critical point.*

Let the surface be given by $(x, y, f(x, y))$ for $x, y \in \mathbb{R}$. Define "smooth" as f is infinitely differentiable. The intuition stems from the fact that the statement is true in one variable.

Counterexample. Ash and Sexton[3] provide the following:

$$f(x,y) = -\frac{1}{1+x^2} + (2y^2 - y^4)\left(e^x + \frac{1}{1+x^2}\right).$$

The origin is a local, but not a global, minimum with

$$\nabla f(0,0) = (0,0),\ \nabla^2 f(0,0) = \begin{bmatrix} 2 & 0 \\ 0 & 8 \end{bmatrix},\ f(0,0) = -1 > f(0,2) = -17.$$

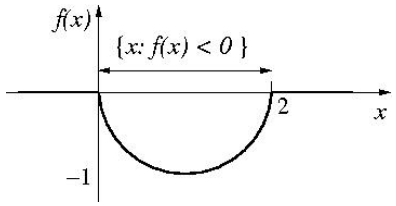
There are no other critical points.

NLP Myth 7. *If f is continuous, the closure of its strict interior equals its level set. That is,* $\mathrm{cl}\{x : f(x) < 0\} = \{x : f(x) \le 0\}$.

One importance of this in stability — see NLP Myth 8.

Counterexample. Let f be the following function on $\mathbb{R}$:

$$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ (x-1)^2 - 1 & \text{if } 0 \le x < 2 \\ 0 & \text{if } 2 \le x \end{cases}$$



f is continuous (and quasiconvex). However, the strict interior of the 0-level set is $(0, 2)$, so its closure is only $[0, 2]$. We lose the flat portions in the tails.

NLP Background — Semi-continuity

Some myths involve continuity properties of the optimal value as a function of the right-hand side. This requires us to consider the feasibility region a *point-to-set map*, as follows. Let $X(b) = \{x \in X : g(x) \le b\}$ denote the feasible region, and let $B = \{b : X(b) \ne \emptyset\}$. The optimal value function is $f^*(b) = \sup\{f(x) : x \in X(b)\}$, and the optimality region is $X^*(b) = \{x \in X(b) : f(x) = f^*(b)\}$. Unless stated otherwise, we are interested in continuity properties at $b = 0$, and we assume $0 \in B$.

The optimal value function is *lower semi-continuous* (lsc) at $b = 0$ if

$$\liminf_{b \to 0} f^*(b) \ge f^*(0).$$

The optimal value function is *upper semi-continuous* (usc) at $b = 0$ if

$$\limsup_{b \to 0} f^*(b) \leq f^*(0).$$

The optimal value function is *continuous* if it is both lsc and usc.

The neighborhood of a set $S \subseteq \mathbb{R}^n$ is given by:

$$\mathcal{N}_\varepsilon(S) = \{y \in \mathbb{R}^n : ||y - x|| \leq \varepsilon \text{ for some } x \in S\},$$

where $\varepsilon > 0$ and $||\bullet||$ is any norm of interest.

A point-to-set map $S(b)$ is *lower semi-continuous* (lsc) at $b = 0$ if for $b^k \to 0$ and $\varepsilon > 0$

$$\exists K \ni S(0) \subset \mathcal{N}_\varepsilon(S(b^k)) \text{ for } k > K.$$

$S(b)$ is *upper semi-continuous* (usc) at $b = 0$ if for $b^k \to 0$ and $\varepsilon > 0$

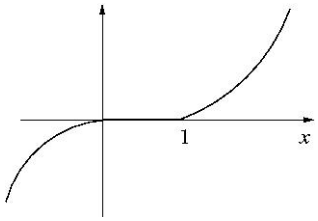
$$\exists K \ni S(b^k) \subset \mathcal{N}_\varepsilon(S(0)) \text{ for } k > K.$$

NLP Myth 8. *Given the objective is continuous and the feasible region is non-empty and compact at $b = 0$, the optimal value function is lsc at $b = 0$.*

Evans and Gould[10] provide the following:

Counterexample. max $x : g(x) \leq 0$, where g is given by:

$$g(x) = \begin{cases} x^3 & \text{if } x \leq 0 \\ 0 & \text{if } 0 \leq x \leq 1 \\ (x-1)^3 & \text{if } x \geq 1. \end{cases}$$

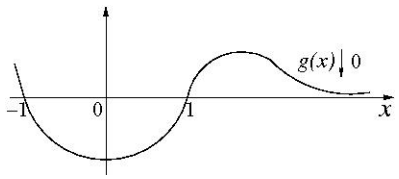


Then, for $b^k = -\frac{1}{k^3}$, we have $f^*(b^k) = -\frac{1}{k} \to 0$, but $f^*(0) = 1$. The key to this discontinuity is that $\text{cl}\{x : g(x) < 0\} \neq \{x : g(x) \leq 0\}$.

NLP Myth 9. *Given the objective is continuous and the feasible region is non-empty and compact at $b = 0$, the optimal value function is usc at $b = 0$.*

Evans and Gould[10] provide the following:

Counterexample. max $x : g(x) \leq 0$, where g has the following shape:



Then, for $b^k = g(k)$, we have $f^*(b^k) = k \to \infty$, but $f^*(0) = 1$. The key to this discontinuity is that $\{x : g(x) \le b\}$ is unbounded for all $b > 0$ (even though $\{x : g(x) \le 0\}$ is bounded).

NLP Myth 10. *If the feasibility region and optimal value function are lsc at $b = 0$, so is the optimality region.*

Greenberg and Pierskalla[21] provide the following:

Counterexample. Consider max $x : g(x) \le 0$, where g is given in NLP Myth 9. Specifically, let $g(x) = \min\{x^2 - 1, e^{-x^2}\}$ (they cross at about $x = \pm 1.3$). We have

$$X(b) = \begin{cases} [-\sqrt{b+1}, \sqrt{b+1}\,] & \text{if } -1 < b < 0; \\ [-1, 1\,] & \text{if } b = 0; \\ [-\sqrt{b+1}, \sqrt{b+1}\,] \cup [\sqrt{-\ln b}, \infty) & \text{if } 1 > b > 0. \end{cases}$$

We have

$$\big(-1 \ge -\sqrt{b+1} - \varepsilon \text{ and } 1 \le \sqrt{b+1} + \varepsilon \big) \leftrightarrow 1 - \varepsilon \le \sqrt{b+1} \leftrightarrow b \ge \varepsilon^2 - 2\,\varepsilon\,.$$

Hence, for any $\varepsilon > 0$, let $b \ge \varepsilon^2 - 2\,\varepsilon$ to have $X(0) \subset \mathcal{N}_\varepsilon(X(b))$. This proves that X is lsc at $b = 0$. Further,

$$f^*(b) = \begin{cases} \sqrt{b+1} & \text{if } -1 < b < 0; \\ 1 & \text{if } b = 0; \\ \infty & \text{if } 1 > b > 0. \end{cases}$$

Hence, $\liminf_{b \to 0} f^*(b) = 1 = f^*(0)$, so f^* is lsc at $b = 0$.

Now consider the optimality region:

$$X^*(b) = \begin{cases} \{\sqrt{b+1}\} & \text{if } -1 < b < 0; \\ \{1\} & \text{if } b = 0; \\ \emptyset & \text{if } 1 > b > 0. \end{cases}$$

Let $b^k = e^{-k}$, so $X^*(b^k) = \emptyset$ for all k. Then,

$$X^*(0) = \{1\} \not\subset \mathcal{N}_\varepsilon(X^*(b^k)) = \emptyset,$$

so X^* is not lsc at $b = 0$.

Also see Dantzig, Folkman, and Shapiro[7].

NLP Myth 11. *If the feasibility region and optimal value function are usc at $b = 0$, so is the optimality region.*

Greenberg and Pierskalla[21] provide the following:

Counterexample. Consider max $x : g(x) \le 0$, where g is given in NLP Myth 8. We have

$$X(b) = \begin{cases} (-\infty, \sqrt[3]{b}\,] & \text{if } b < 0; \\ (-\infty, 1] & \text{if } b = 0; \\ (-\infty, 1 + \sqrt[3]{b}\,] & \text{if } b > 0. \end{cases}$$

Since $\mathcal{N}_\varepsilon(X(0)) = (-\infty, 1+\varepsilon]$, we have

$$X(b) \subset \mathcal{N}_\varepsilon(X(0)) \text{ for } b \leq \varepsilon^3 .$$

Hence, X is usc at $b = 0$. Further,

$$f^*(b) = \begin{cases} \sqrt[3]{b} & \text{if } b < 0; \\ 1 & \text{if } b = 0; \\ 1 + \sqrt[3]{b} & \text{if } b > 0. \end{cases}$$

Hence,

$$\limsup_{b \to 0} f^*(b) = 1 = f^*(0),$$

so f^* is usc at $b = 0$.

Now consider the optimality region:

$$X^*(b) = \begin{cases} \{\sqrt[3]{b}\} & \text{if } b < 0; \\ \{1\} & \text{if } b = 0; \\ \{1 + \sqrt[3]{b}\} & \text{if } b > 0. \end{cases}$$

We have $\mathcal{N}_\varepsilon(X^*(0)) = [1-\varepsilon, 1+\varepsilon]$. Let $b \uparrow 0$, so $X^*(b) = \{\sqrt[3]{b}\} \not\subset \mathcal{N}_\varepsilon(X^*(0))$ for $\varepsilon < 1$. Hence, X^* is not usc at $b = 0$.

Also see Dantzig, Folkman, and Shapiro[7].

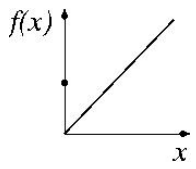
NLP Myth 12. *A set is convex if it contains the midpoint of any pair of its points.*

Counterexample. The set of rational values.

NLP Myth 13. *A convex function is continuous.*

This is true in the interior of its effective domain, but not necessarily on its boundary.

Counterexample. Let $f : \mathbb{R}_+ \to \mathbb{R}$, with $f(0) = 1$ and $f(x) = x$ if $x > 0$.



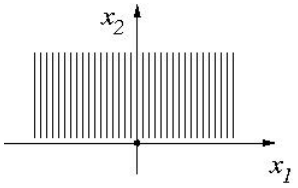
NLP Myth 14. *A convex function is upper semi-continuous on its boundary.*

Fenchel[12, 13] shows that $\liminf_{y \to x} f(y) \leq f(x)$, and his example, which follows, shows f need not be upper semi-continuous on the boundary.

Counterexample. Consider the following:

$$X = \{x \in \mathbb{R}^2 : x_2 > 0 \vee x = (0,0)\}$$

$$f(x) = \begin{cases} \dfrac{x_1^2 + x_2^2}{2x_2} & \text{if } x_2 > 0; \\ 1 & \text{if } x = (0,0). \end{cases}$$



No points on x_1-axis, except $(0,0)$.

f is convex on X and $\liminf_{y \to (0,0)} f(y) = 0$. By letting x^k take the nonlinear path such that $x_2^k = (x_1^k)^3 = \frac{1}{k}$,

$$\limsup_{y \to (0,0)} f(y) = \lim_{k \to \infty} f(x^k) = \lim_{k \to \infty} \frac{(x_1^k)^2 + (x_1^k)^6)}{2(x_1^k)^3} = \lim_{k \to \infty} \left(\frac{k}{2} + \frac{1}{2k^3} \right) = \infty.$$

NLP Myth 15. *Suppose* $x^* \in \operatorname{argmax}\{f(x) : x \in X,\ g(x) \leq 0\}$ *and* $g(x^*) < 0$. *Then,* $x^* \in \operatorname{argmax}\{f(x) : x \in X,\ g(x) \leq b\}$ *for any* $b > 0$.

This is true if f is concave and g is convex on X, in which case x^* is the (unconstrained) maximum of f on X.

Counterexample. $\max x_1 + 2x_2 : x_1, x_2 \in \{x : x(1-x) = 1\},\ 2x_1 + 4x_2 - 3 \leq b$.

At $b = 0$, $x^* = (1,0)$ and $g(x*) < 0$; however, at $b = 4$, the optimal solution is $x^* = (0,1)$, which is not optimal for $b = 0$.

Other examples, which are not integer-valued, include the case where x^0 is a global maximum for $b = 0$, but it is only a local maximum for $b > 0$. The objective function (f) decreases for a while, but then it turns back upward to a maximum at $x^* > x^0$ with $f(x^*) > f(x^0)$.

NLP Myth 16. **new** *Cauchy's steepest ascent either diverges or converges to a relative maximum.*

We seek to maximize $f(x)$, and the iterations are:

$$x^{k+1} = x^k + s_k \nabla f(x^k), \text{ where } s_k > 0.$$

Wolfe[39] presented an insightful analysis, but he later corrected some statements that seemed intuitive at first, such as this Myth.

Counterexample. Wolfe[40] provides the following: $f(x,y) = -\frac{1}{3}x^3 - \frac{1}{2}y^2$, which is concave for $x > 0$. Starting at (x^0, y^0) such that $0 < x^0 < 1$, the sequence satisfies $0 < x^k < 1$ for all k. Moreover, $\{(x^k, y^k)\} \to (0,0)$, which is not a relative maximum. Wolfe discusses this further, giving more insight into underlying behavior when studying the differential equation $\dot{x} = \nabla f(x)$.

NLP Myth 17. *If f is concave on $[a,b]$, the truncated gradient algorithm converges to an optimal solution. That is, $x' = x + sd$ yields a sequence for which $d = 0$ in the limit, where d is the projected steepest ascent direction:*

$$d_j = \begin{cases} \dfrac{\partial f(x)}{\partial x_j} & \text{if} \quad a_j < x_j < b_j \\ \max\left\{0, \dfrac{\partial f(x)}{\partial x_j}\right\} & \text{if} \quad a_j = x_j \\ \min\left\{0, \dfrac{\partial f(x)}{\partial x_j}\right\} & \text{if} \quad x_j = b_j. \end{cases}$$

Note: $d = 0$ if, and only if, x satisfies the first-order optimality conditions, which is equivalent to x being optimal in the case of a concave maximand. If $d \neq 0$, $s \in \operatorname{argmax}_{t>0}\{f(x+td)\}$.

Wolfe[41] provided the following:

Counterexample. Let $f(x,y,z) = -\frac{4}{3}(x^2 - xy + y^2)^{\frac{3}{4}} + z$, on the cube, $[0,100]^3$. It can be shown (non-trivially) that f is concave, and that the truncated gradient algorithm converges to the non-optimal point, $(0,0,c)$, where $c < 100$, depending on the starting point (in particular, $c = 0$ for $z^0 = 0.1$). See Dussault and Fournier[9] and Greenberg[19] for some details.

The basic problem is that the zig-zagging can cause non-finite convergence on some face, but the optimum lies on another face.

NLP Myth 18. **new** *Rosen's projected gradient algorithm with linear constraints converges to a Kuhn-Tucker point with inexact line search.*

The NLP is max $f(x) : Ax \leq b$, where f is continuously differentiable. The *active set* of constraints is denoted $I(x) = \{i : A_{i\bullet}x = b_i\}$, and $A_{I(x)}$ is the submatrix whose rows are $I(x)$. At a general iteration, Rosen's projected gradient method is to set the (feasible) direction: $d(x) = P(x)\nabla f(x)$, where $P(x)$ is the projection matrix onto the active face:

$$P(x) = I - A_{I(x)}^{\mathsf{T}}\left[A_{I(x)}A_{I(x)}^{\mathsf{T}}\right]^{-1}A_{I(x)}.$$

If $d(x^k) = 0$, the first-order (Lagrangian) conditions are satisfied, and the algorithm terminates. If $d(x^k) \neq 0$ and $A_{i\bullet}d(x^k) \leq 0$ for all $i \notin I(x^k)$, the problem is unbounded, and the algorithm terminates. Otherwise, let $\mathcal{I}(x) = \{i \notin I(x^k) : A_{i\bullet}d(x^k) > 0\}$ $(\neq \emptyset)$, and $\bar{s} = \min\limits_{i\in\mathcal{I}(x^k)} \dfrac{b_i - A_{i\bullet}x^k}{A_{i\bullet}d(x^k)}$. Then, the iteration is given by:

$$x^{k+1} = x^k + s_k d(x^k),$$

where s_k is the step size, limited by $0 \leq s_k \leq \bar{s}$. An *inexact line search* is specifying s_k without optimizing along the direction, such as using Armijo's rule[1] (See NLP Myth 30.) Also, successive directions are not orthogonal, so the zig-zag phenomenon does not apply, as in NLP Myth 17.

Counterexample. Hu[26] provides the following:

$$\max\ -x_1^2 + x_2^2 :\ x \geq 0,\ x_1 - x_2 \leq 1.$$

The step size is determined by a near-optimal line search:

$$f(x^k + s_k d(x^k)) \geq \max_{0<s\leq \overline{s}} \{f(x^k + sd(x^k))\} - \varepsilon_k,$$

where $\varepsilon_k = 2^{1-k}$. Starting at $x^0 = (2,1)$, Hu's inexact line search generates the sequence $\{x^k\} = \{(1 + 2^{-k}, 2^{-k})\} \to (1,0)$ (with $I(x^k) = \{3\}$, $d(x^k) = (-x_1^k + x_2^k, -x_1^k + x_2^k) = (-1,-1)$, and $s_k = 2^{1-k}$). The optimal step size is $s^* = \overline{s} = 2^{-k}$, so

$$f(x^k + s_k d(x^k)) - f(x^k + s^* d(x^k)) = -(1 + 2^{1-k}) - (-1) = -2^{1-k} = -\varepsilon_k\,.$$

This dispels the myth because the only Kuhn-Tucker point is at $x = 0$.

NLP Myth 19. *A strictly quasiconvex function is quasiconvex.*

The definition of strictly quasiconvex says that f is defined on a convex set X, and $f(\alpha x + (1-\alpha)y) < \max\{f(x), f(y)\}$ for $x, y \in X$ such that $f(x) \neq f(y)$ and $\alpha \in (0,1)$. (Note that the definition imposes no restriction if $f(x) = f(y)$.) Karamardian[27] found the following:

Counterexample. $X = \mathbb{R}$ and $f(x) = 0$ for $x \neq 0$, $f(0) = 1$. It can be shown that f is strictly quasiconvex, but the level set, $\{x : f(x) \leq 0\}$, is not convex, so f is not a quasiconvex function.

This is what led to the definition of an *explicitly quasiconvex* function by Martos[31]. Details and further properties are given by Greenberg and Pierskalla[20].

NLP Myth 20. *Let f be convex on $X \neq \emptyset$, where $X \subseteq \mathbb{R}^n$, and the range of f is in $\mathbb{R}^m$. Then, either there exists $x \in X$ such that $f(x) \leq 0$, or there exists $y \in \mathbb{R}^m$ such that $y^\mathsf{T} f(x) > 0$ for all $x \in X$. Further, the two alternatives exclude each other.*

The reason this seems reasonable is due to the theorem by Fan, Glicksburg and Hoffman[11], where the first system is $f(x) < 0$, and the alternative is $y \in \mathbb{R}^m \setminus \{0\}$ such that $y^\mathsf{T} f(x) \geq 0$ for all $x \in X$. The myth "seems reasonable," considering related transposition theorems in linear systems.

Counterexample. Let $X = \{(x_1, x_2) : x_2 > 0 \vee (x_2 = 0 \text{ and } x_1 > 0)\}$ and $f(x) = x^\mathsf{T}$. Then, $f(x) \leq 0$ has no solution in X. The (fallacious) alternative is $y \geq 0$ and $y^\mathsf{T} f(x) = y_1 x_1 + y_2 x_2 > 0$ for all $x \in X$. If $y_1 > 0$, let $x_2 = 1$ and $x_1 \leq -\frac{y_2}{y_1}$, so $x \in X$, but $y^\mathsf{T} x \leq 0$. If $y_1 = 0$, let $x_2 = 0$ and $x_1 > 0$, so $y^\mathsf{T} x = 0$. Thus, the alternative system also has no solution.

NLP Myth 21. *Newton's method converges to a stationary point if the starting point is sufficiently close.*

Newton's method is applied to the root-finding problem, $f(x) = 0$, with the iterations:

$$x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)}.$$

(In optimization, this is applied to the derivative of the objective function.)

A simple counterexample to the myth is any quadratic with two distinct roots, a, b, and $x^0 = \frac{a+b}{2}$. In that case $f'(x^0) = 0$, so Newton's method is undefined. (We can make a, b arbitrarily close to each other to satisfy the condition of the myth.)

A more interesting example is analyzed by Ascher[2], where Newton's method cycles — that is, it generates x^0 after a finite number of iterations. (The analysis goes beyond this simple example.)

Counterexample. Let $f(x) = x^2 + 3$, so $x^{k+1} = \frac{1}{2}x^k - \frac{3}{2x^k}$. Then, for $x^0 = \pm 1$, the iterates cycle in two iterations.

NLP Myth 22. *Newton's method has converged when the change in the iterate value is less than some specified, small tolerance.*

Let $f : \mathbb{R} \to \mathbb{R}$ be a function in C^1, for which we seek a root, $f(x) = 0$. Let $\{x^k\}$ be generated by Newton's method:

$$x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)}.$$

The stopping criterion in the statement says that we terminate when

$$|x^{k+1} - x^k| < \tau,$$

where τ is the tolerance. Donovan, Miller and Moreland[8] provided the following:

Counterexample. $f(x) = \sqrt[3]{x}\, e^{-x^2}$. The generated sequence satisfies the iteration equation:

$$x^{k+1} = x^k - \frac{3x^k}{1 - 6(x^k)^2},$$

which does not converge. Yet, $|x^{k+1} - x^k| < \tau$ is equivalent to: $\left|\frac{3x^k}{1 - 6(x^k)^2}\right| < \tau$, which is eventually satisfied since the authors prove that $\{x^k\} \to \infty$.

They also derive properties of f and insight into its construction for the counterexample. In particular, they note that the first part, $\sqrt[3]{x}$, fails Newton's method on its own ($x^{k+1} = -2x^k$ implies $x = 0$ is a repelling fixed point). The second part, e^{-x^2}, gives the "false convergence" property.

NLP Myth 23. *Cyclic descent produces a local minimum.*

We are given the mathematical program: min $f(x) : x \in X$, where X is a non-empty, closed subset of $\mathbb{R}^n$. Cyclic descent proceeds as follows (where $\boldsymbol{e}_i$ is the i^{th} unit vector):

Set $y = x$
for i=1:n **do**
  find $t^* \in \text{argmin}\{f(y + t\boldsymbol{e}_i) : (y + t\boldsymbol{e}_i) \in X\}$
  Set $y \leftarrow y + t^*\boldsymbol{e}_i$
end for
if $||y - x|| \leq \tau$, exit; **else** set $x \leftarrow y$ and repeat.

The problem is that while f may not be increased by a change in any one variable, it could increase with simultaneous changes in more than one variable.

Counterexample. $f(x) = (x_2 - x_1^2)(x_2 - 2x_1^2)$; start at $x = (0, 0)$.

$\min_t f(t, 0) = \min_t 2t^4 = 0$, so there is no change after $i = 1$. Similarly, $\min_t f(0, t) = \min_t t^2 = 0$, so there is no change after $i = 2$. Hence, cyclic descent terminates after one iteration with the same point with which it started, $x = (0, 0)$. This is not a minimum, even locally, because we can let $x_2 = \frac{3}{2}x_1^2$. Then, for x_1 arbitrarily close to 0, but $x_1 \neq 0$, $f(x) = -\frac{1}{4}x_1^2 < 0$.

Also see Powell[34].

NLP Myth 24. *If one algorithm has a higher order of convergence than another, it is better.*

The reason that this is wrong is that the goodness of a solution (for example, how close it is to optimal) cannot be accurately described by one number. Greenberg[17] provides the following:

Counterexample. Let $\{x^k\}$ be a sequence of solutions converging to x^*, and let $f(x^*)$ be the optimal objective value. Define the deviations, $\{e^k = f(x^k) - f(x^*)\}$. For definiteness, suppose $e^k > 0$ for all k and we are minimizing (so $\{f(x^k)\}$ is approaching from above, as in a primal algorithm). Define the "goodness" of x^k to be e^k — that is, how close $f(x^k)$ is to the optimal objective value. Now suppose another algorithm generates the sequence $\{X^k\}$ whose associated goodness is $\{E^k\}$, where

$$E^k = \begin{cases} \min\{e^k, e^{k+1}\}/k & \text{if } k \text{ is odd;} \\ E^{k-1} & \text{if } k \text{ is even.} \end{cases}$$

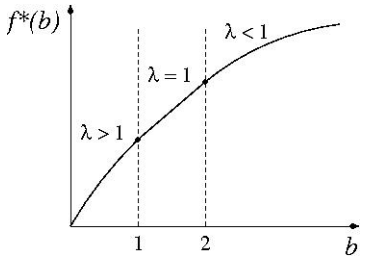
The result is that the second sequence is sublinear (the worst possible for a monotonically decreasing sequence), but X^k is always better since $E^k < e^k$ for all k.

An algorithm that has plateaux exhibits this behavior — no improvement for an iteration, then a sharp improvement. Some measures of the order of convergence take constant plateaux into account, but the example can be revised to have a plateau of length k at iteration k, so the order of convergence is still sublinear.

NLP Myth 25. *For a convex program, the Generalized Lagrange Multiplier Method converges to an optimal solution.*

Counterexample. Let $f^*(b) = \max\{f(x) : 0 \le x \le b\}$ for $b \ge 0$, where

$$f(x) = \begin{cases} 2\sqrt{x} & \text{if} \quad x \le 1 \\ x + 1 & \text{if} \quad 1 \le x \le 2 \\ 4 - e^{-(x-2)} & \text{if} \quad x \ge 2 \end{cases}$$



(Note: $f^*(b) = f(b)$.) Using any interval-reduction method[18] that does not terminate finitely, the left endpoint converges to 1 and the right endpoint converges to 2.

Finite termination occurs when the two endpoints equal the linearity portion, so the next iteration chooses the multiplier equal to the slope ($\lambda = 1$). Then, the set of optimal solutions is the interval $[1, 2]$, so that any b in this interval is generated by searching the set of alternative optima. Without finite termination, no b in (1,2) is a limit point. There is thus a *pseudo-gap*[16] for $b \in (1, 2)$ in that the algorithm cannot reach the solution, but there is no duality gap.

NLP Myth 26. *Ritter's method to solve a QP converges to a global optimal solution.*

The counterexample was found by Zwart[42]. The problem is that the sequence of feasible regions (with a cut added each iteration) does not approach the optimality region. A non-global optimum point persists in the sequence of optima.

Counterexample.

$$\max\ 2x_1^2 + x_1x_2 + 2x_2 : -x_1 \le 0,\ x_1 + x_2 \le 1,\ 1.5x_1 + x_2 \le 1.4,\ -x_2 \le 10.$$

Each cut has the form $\frac{1}{2^k}x_1 + x_2 \le \frac{1}{2^k}$, and the optimal point is at the extreme point, $(0, \frac{1}{2^k})$. Ritter's method does not eliminate (0,0), so it cannot converge to the global optimum, which is at $(7.6, -10)$.

NLP Myth 27. *Tui's method to maximize a convex function subject to linear constraints converges to a global optimal solution.*

This counterexample was found by Zwart[42]. The problem is that Tui's algorithm can cycle — that is, repeat the generated subproblems.

Counterexample.

$$\begin{aligned} \max\ x_1^2 + x_2^2 + (x_3 - 1)^2 :\ & x_2 \geq 0 \\ x_1 + x_2 - x_3 &\leq 0 \\ -x_1 + x_2 - x_3 &\leq 0 \\ 12x_1 + 5x_2 + 12x_3 &\leq 22.8 \\ 12x_1 + 12x_2 + 7x_3 &\leq 17.1 \\ -6x_1 + x_2 + x_3 &\leq 1.9 \end{aligned}$$

Zwart gives the following generated sequence, starting at $x = (0, 0, 0)$.

THE CYCLING OF TUI'S METHOD

q	k	y^{q,k_1}	y^{q,k_2}	y^{q,k_3}	$x_{\text{opt}}^{q,k}$	$fx_{\text{opt}}^{q,k}$
1	1	(1, 0, 1)	(0, 1, 1)	(−1, 0, 1)	(0, 0, 1.90)	−0.81
2	1	(0, 0, 2)	(0, 1, 1)	(−1, 0, 1)	(0.05, 0.60, 1.60)	−0.7225
2	2	Not required because $\lambda_2^{1,1}=0$				
2	3	(1, 0, 1)	(0, 1, 1)	(0, 0, 2)	(0.05, 0.60, 1.60)	−0.7225
3	1	(0.055, 0.66, 1.75)	(0, 1, 1)	(−1, 0, 1)	*	
3	2	(0, 0, 2)	(0.055, 0.66, 1.75)	(−1, 0, 1)	*	
3	3	(0, 0, 2)	(0, 1, 1)	(0.055, 0.66, 1.75)	(−0.95, 0, 0.95)	−0.905
3	4	(0.055, 0.66, 1.75)	(0, 1, 1)	(0, 0, 2)	(−0.95, 0, 0.95)	−0.905
3	5	(1, 0, 1)	(0.055, 0.66, 1.75)	(0, 0, 2)	*	
3	6	(1, 0, 1)	(0, 1, 1)	(0.055, 0.66, 1.75)	*	
4	1	(−1, 0, 1)	(0, 1, 1)	(0.055, 0.66, 1.75)	*	
4	2	(0, 0, 2)	(−1, 0, 1)	(0.055, 0.66, 1.75)	*	
4	3	(0, 0, 2)	(0, 1, 1)	(−1, 0, 1)	(0.05, 0.60, 1.60)	−0.7225

* Optimal LP objective has value ≦1.

Notation: q indexes auxiliary problem; k_q indexes solution generated for q^{th} auxiliary problem; y^{qk_q} is Tui's search direction. See [42] for details.

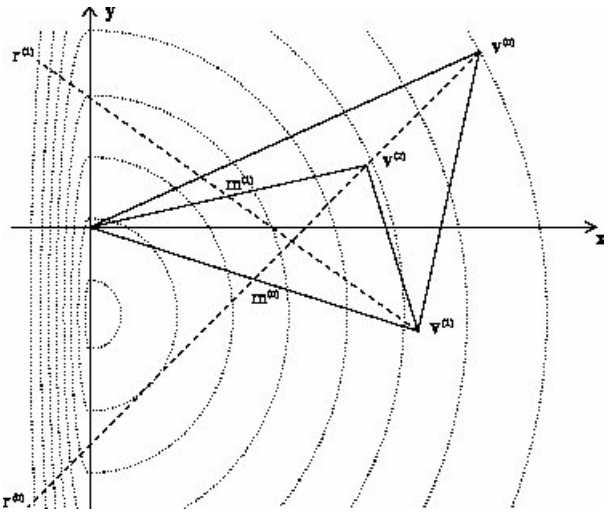
NLP Myth 28. *The Nelder-Mead method converges to a local optimum.*

The Nelder-Mead method is a very good heuristic that does well in many hard nonlinear problems. For a long time after its publication in 1965, many thought it converges to a local optimum, but McKinnon[32] provided the following:

Counterexample. Let

$$f(x, y) = \begin{cases} AB|x|^c + y + y^2 & \text{if } x \leq 0 \\ B\,x^c + y + y^2 & \text{if } x \geq 0, \end{cases}$$

where A, B, c are positive constants. Also, f is convex and has continuous first derivatives for $c > 1$.



Contours of McKinnon's function for $A = 6$, $B = 60$, $c = 2$, taken from [32]. Note: $(0, -1)$ is a descent direction from $(0, 0)$.

McKinnon proves (nontrivially) that for certain choices of these constants, the algorithm repeats the inside contraction step with the best vertex remaining fixed. In particular, with $A = 6$ and $B = 60$, the counterexample works for $0 \le c \le \bar{c}$, and it does not work for $c > \bar{c}$, where $\bar{c} \approxeq 3.06$, from McKinnon's derivation.

NLP Myth 29. *Rosen's decomposition method converges to an optimal solution for convex programs.*

We are given the convex program:

$$\min_{(x,y)} \; cx : \; Ax \ge b(y),$$

where b is a convex function. Rosen's decomposition is to separate x and y problems and proceed as follows. For any fixed $y = \bar{y}$, we obtain $\bar{x}$ by solving:

$$\text{LP}(\bar{y}): \; \min_{x} \; cx : \; Ax \ge b(\bar{y}).$$

Partition $[A\; b]$ into the tight and surplus constraints at the solution:

$$B\bar{x} = b_B(\bar{y}), \; N\bar{x} > b_N(\bar{y}).$$

(So $A = \left(\begin{smallmatrix} B \\ N \end{smallmatrix}\right)$ and $b = \left(\begin{smallmatrix} b_B \\ b_N \end{smallmatrix}\right)$.) We suppose B is nonsingular and use the tight constraints to eliminate $x = B^{-1}b_B(y)$ for any choice of y. To maintain feasibility of the surplus constraints, we require

$$b_N(y) - N^{\intercal}B^{-1}b_B(y) \le 0.$$

Using the Taylor expansion at $\bar{y}$ to linearize the constraints, Rosen's method solves the nonlinear program:

$$\text{NLP(B}, \bar{y}): \; \min_{y} \; cB^{-1}b_B(y) : \; b_N(\bar{y}) + \nabla b_N(\bar{y})(y - \bar{y}) - N^{\intercal}B^{-1}\big(b_B(\bar{y}) + \nabla b_B(\bar{y})(y - \bar{y})\big) \le 0.$$

Rosen's method is to start with y^0, then solve iteratively:

1. Solve LP(y^k) and obtain B
2. Solve NLP(B, y^k) and obtain y^{k+1}.

Subject to some details about step 2, the idea is to solve a sequence of problems that are decomposed, rather than tackle the whole nonlinear problem.

Grossmann[23] provided the following:

Counterexample. min $x : x \geq y^2,\ x \geq y$. Let the starting value satisfy $y^0 > 1$. Grossmann proves $y^k > 1$ for all $k = 0, 1, \ldots$, but the optimum is at $(x^*, y^*) = (0, 0)$.

Proceeding inductively, suppose $y^k > 1$. Then, $x^k = (y^k)^2$, and the optimal basis has the slack variable $s = x - y$. Therefore, y^{k+1} is determined by

$$y^{k+1} \in \operatorname{argmin}\{y^2 : (2y^k - 1)y \geq (y^k)^2\}.$$

Since $y^k > 1$, the solution is $y^{k+1} = \frac{(y^k)^2}{2y^k - 1} > 1$, and that completes the induction proof.

NLP Myth 30. new *In methods of feasible directions, it is better to improve the objective function each iteration than allow it to worsen.*

Counterexample. Grippo, Lampariello, and Lucidi[22] illustrated the use of their *nonmonotone method*

$$x^{k+1} = x^k - s_k[\nabla^2 f(x^k)]^{-1}\nabla f(x^k),$$

where $s_k = \operatorname{sign}\big(\nabla f(x^k)^{\mathsf{T}}[\nabla^2 f(x^k)]^{-1}\nabla f(x^k)\big)$.

The counterexample applies this to an unconstrained minimization in $\mathbb{R}^2$ using Rosenbrock's function: $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$. The minimum is at $x^* = (1, 1)$, and the starting point is $x^0 = (-1.2, 1)$. Their nonmonotone method converges in 7 iterations with $f(x^4) > f(x^3)$:

k	x_1^k	x_2^k	$f(x^k)$
0	−1.2000000000	1.0000000000	24.200
1	−1.1752808989	1.3806741573	4.732
2	0.7631148712	−3.1750338547	1.412
3	0.7634296789	0.5828247755	0.056
4	0.9999953111	0.9440273239	0.3132
5	0.9999956957	0.9999913913	1.853×10^{-11}
6	1.0000000000	1.0000000000	3.433×10^{-20}
7	1.0000000000	1.0000000000	$< 10^{-38}$

Steepest descent using optimal step size takes 33 iterations to get as close. Armijou's[1] descent (without optimal step size) takes 22 iterations. This illustrates that improving the objective function every iteration is not necessarily a most effective way to reach the optimum.

Generally, Rosenbrock's function is used to illustrate profuse zig-zagging in Cauchy's steepest descent with $s_k \in \operatorname{argmin}\{f(x^k + s\nabla f(x^k)) : s \geq 0\}$. The nonmonotone method in this counterexample highlights the need to capture curvature information, as does Newton's method. (Also see MacMillan[29] for mixing steepest descent with Newton's method.)

NLP Myth 31. **new** *Sequential Quadratic Programming (SQP) is quadratically convergent when it is sufficiently close to the solution.*

Given $\max\{f(x) : x \in \mathbb{R}^n,\ g(x) = 0\}$, where f, g have continuous second derivatives, the SQP subproblem is given by the iteration subproblem:

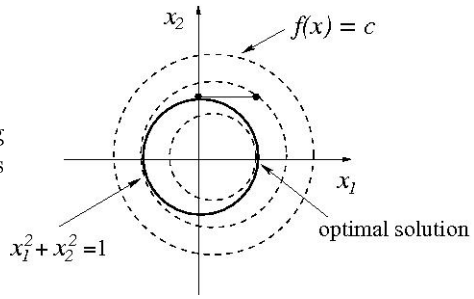
$$\max\ f(x^k) + \nabla f(x^k)(x - x^k) + \tfrac{1}{2}(x - x^k)^\intercal H(x^k)(x - x^k) : g(x^k) + \nabla g(x^k)(x - x^k) = 0,$$

where $H(x^k)$ is the Hessian of the Lagrangian: $H(x^k) = \nabla^2 f(x^k) - \lambda^k \nabla^2 g(x^k)$. The subproblem thus requires both x^k and λ^k as input. The constraint is replaced by its linear approximation from the Taylor series, so the iterates need not be feasible.

The intuition is that SQP behaves like Newton's method when it is close to the solution. The problem is that the step size can become so close to zero that it slows the convergence. This is known as the *Maratos effect*, and it can prevent convergence entirely — see Maratos[30], Panier and Tits[33], Bonnans et al.[5], and Vanden Berghen[36].

Counterexample. Vanden Berghen[36] provides the following: $\min\ 2(x_1^2 + x_2^2 - 1) - x_1 : x_1^2 + x_2^2 = 1$.

The optimum is at $x^* = (1, 0)$, and the starting point is $x^0 = (0, 1)$. The SQP algorithm finds $x^1 = (1, 1)$, and the next step size is zero.



Taken from Vanden Berghen[36].

The Maratos effect can be overcome by *Second Order Correction* and sometimes by filtering — see Fletcher, Leyffer, and Toint[14].

NLP Myth 32. **new** *A barrier algorithm is globally convergent if the functions are smooth and there is a unique stationary point that is the global optimum.*

Counterexample. Wächter and Biegler[38] provide the following:

$$\min\ x_1 : x_1^2 - x_2 = 1,\ x_1 - x_3 = 0.5,\ x_2, x_3 \geq 0.$$

The barrier problem is

$$\min\ x_1 - \mu\big(\ln(x_2) + \ln(x_3)\big) : x_1^2 - x_2 = 1,\ x_1 - x_3 = 0.5,\ x_2, x_3 > 0.$$

Note that the non-negativity constraints are replaced by positivity constraints. That is the formal statement of the barrier NLP, but the positivity constraints are ignored (and sometimes omitted, with risk of confusion) because we must use a *continuous-trajectory algorithm*, starting with a feasible point. This excludes, for example, applying the Nelder-Mead algorithm.

Starting with $x^0 = (-2, 3, 1)$, Wächter[37] gives the iterates for a particular barrier algorithm:

k	x_1^k	x_2^k	x_3^k	s_k
0	-2	3	1	–
1	-1.8077165354330709	2.2308661	0.01	0.338
2	-1.1941467232510745	0.0223087	0.0115512	0.264
3	-1.1827848033469259	0.223E-03	0.157E-02	0.125E-01
4	-1.1825077076013453	0.223E-05	0.160E-04	0.109E-02
5	-1.1825049104784042	0.223E-07	0.160E-06	0.111E-04
6	-1.1825048825066040	0.223E-09	0.160E-08	0.111E-06
7	-1.1825048822268860	0.223E-11	0.160E-10	0.111E-08
8	-1.1825048822240889	0.223E-13	0.160E-12	0.111E-10
9	-1.1825048822240609	0.223E-15	0.160E-14	0.111E-12

The algorithm aborts because the step size becomes too small. (see NLP Myth 31 for the Maratos effect.)

Wächter and Biegler note that the example has no degeneracy hidden in the equality constraints, the Jacobian is nonsingular everywhere, and the minimum satisfies second-order sufficient conditions and is strictly complementary. Hence, the counterexample is well posed, not some esoteric pathology.

Wächter's thesis[37] provides a deeper analysis of the above counterexample and shows that seemingly reasonable barrier algorithms from a generic class cannot be globally convergent under mild assumptions. The root of the problem is that those methods compute search directions that satisfy the linearization of the constraints (for example, line-search methods) and are later cut short to keep the iterates positive. Only algorithms that deviate from this paradigm (such as certain trust-region or filter methods) can be shown to have good convergence properties.

Larry Biegler adds the following points.

1. Failure of this example occurs for barrier methods where the search direction satisfies linearization of the equality constraints, followed by a line search, using any merit function (for example, a line-search based Newton method). Because of this restriction and the need to remain feasible to the bounds, the algorithm eventually terminates because it is too constrained to find a search direction to reduce the infeasibility of the equalities. [This is the Maratos effect, which affects SQP and Newton-based methods — see NLP Myth 31).]

2. Wächter's thesis mentions that convergence proofs for Newton-based line search barrier methods (from earlier studies) require boundedness of the multipliers (or similar regularity assumption). This assumption turns out to be violated for this example.

3. There are other barrier methods that can solve this counterexample. For instance, the trust region method (for example, in KNITRO) generates search directions that are not restricted by the constraint linearization, generate search directions that improve the constraint infeasibility, and avoid this failure of the counterexample.

NLP Myth 33. **new** *If every optimum in the trajectory of a barrier function satisfies strong second-order necessary conditions, so does its limit.*

Phillip Toint brought this to my attention. The intuition behind the myth is that many unconstrained algorithms can guarantee convergence to a stationary point that satisfies weak second-order conditions if each point in the trajectory does. The issue is whether the same could be said of the strong second-order conditions.

The NLP is min $f(x) : g(x) \geq 0$, for which we consider the trajectory of the logarithmic barrier function:

$$x^*(\mu) \in \operatorname{argmin} \left\{ f(x) - \mu \textstyle\sum_i \log g_i(x) : g(x) > 0 \right\} \text{ for } \mu > 0.$$

The myth assumes that $x^*(\mu)$ satisfies strong second-order conditions:

$$h^\tau \left[\nabla^2 f(x^*(\mu)) - \mu \sum_i \nabla^2 g_i(x^*(\mu)) \right] h > 0 \text{ for all } h \neq 0.$$

Counterexample. Gould and Toint[15] consider the following:

$$\min_{x \in \mathbb{R}^n_+} \ \tfrac{1}{2} x^\tau Q x,$$

where Q is symmetric and indefinite. The *strong second-order conditions* are that the Hessian of the Lagrangian be positive semi-definite over the space of feasible directions, strengthened by disallowing change in $x_i^* = 0$ when its associated Lagrange multiplier, λ_i, is positive:

$$h^\tau Q h \geq 0 \text{ for all } h : \begin{cases} h_i = 0 \text{ for } x_i^* = 0, \ \lambda_i > 0 \\ h_i \geq 0 \text{ for } x_i^* = 0, \ \lambda_i = 0. \end{cases} \tag{NLP.7}$$

We apply the logarithmic barrier (which is a special case of [15]). For $x > 0$:

$$b(x;\mu) = \tfrac{1}{2} x^\tau Q x - \mu \textstyle\sum_i \log x_i \tag{NLP.8}$$
$$\nabla b(x;\mu) = \quad Qx - \mu X^{-1} \boldsymbol{e} \tag{NLP.9}$$
$$\nabla^2 b(x;\mu) = \quad Q \ + \mu X^{-2} \boldsymbol{e}, \tag{NLP.10}$$

where $\boldsymbol{e}$ is a vector of ones and $X = \operatorname{diag}\{x_i\}$.

Define $Q = I - \frac{3}{2} \dfrac{z \otimes z^\tau}{||z||^2}$, where $z = \boldsymbol{e} - n e_1$ and $||\bullet||$ is the Euclidean norm. Then, $x^*(\mu) = \sqrt{\mu}\, \boldsymbol{e}$. This follows from $Q\boldsymbol{e} = \boldsymbol{e}$:

$$\nabla b(x;\mu) = 0 \Rightarrow \sqrt{\mu} \boldsymbol{e} = \mu \operatorname{diag} \left(\frac{1}{\sqrt{\mu}} \right) \boldsymbol{e}.$$

To show that the strong second-order conditions hold, substitute in (NLP.10):

$$\nabla^2 b(x;\mu) = \tfrac{1}{2} I + \tfrac{3}{2} \left(I - \frac{z \otimes z^\tau}{||z||^2} \right).$$

This is positive definite, thus satisfying the condition in the myth. However, $\lim_{\mu \to 0} x^*(\mu) = 0$, and e_1 is an admissible direction for the strong second-order conditions. Hence, for the myth to be true, we require:

$$0 \leq {e_1}^\intercal Q e_1 = 1 - \tfrac{1}{2}\frac{({e_1}^\intercal \otimes z)^2}{||z||^2} = \frac{n - \frac{3}{2}(n-1)}{n}.$$

This is violated for $n \geq 4$.

NLP Myth 34. *The central path converges to the analytic center of the optimality region of a semidefinite program.*

This is true for a linear program, but an attempted extension failed to assume a strictly complementary solution. Halická, de Klerk, and Roos[24] provide the following:

Counterexample.

$$\min\ x_{44}:$$
$$X = \begin{bmatrix} 1 - x_{22} & x_{12} & x_{13} & x_{14} \\ x_{12} & x_{22} & -\frac{1}{2}x_{44} & -\frac{1}{2}x_{33} \\ x_{13} & -\frac{1}{2}x_{44} & x_{33} & 0 \\ x_{14} & -\frac{1}{2}x_{33} & 0 & x_{44} \end{bmatrix} \succeq 0.$$

The optimality region consists of all positive semidefinite matrices of the form:

$$X^* = \begin{bmatrix} 1 - x_{22} & x_{12} & 0 & 0 \\ x_{12} & x_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

In particular, a positive definite optimum is given by setting $x_{22} = \frac{1}{2}$, $x_{33} = x_{44} = \frac{1}{4}$, and $x_{ij} = 0$ for $i \neq j$. Its analytic center is

$$X^* = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Halická et al. prove that the central path satisfies

$$\lim_{\mu \downarrow 0} X(\mu) = \begin{bmatrix} 0.4 & 0 & 0 & 0 \\ 0 & 0.6 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

NLP Myth 35. *If an NLP is infeasible, one can sequentially test for the deletion of constraints: if its deletion renders the system feasible, keep it; if its deletion maintains the system infeasible, remove it. What remains is an IIS.*

The indicated method is *deletion filtering*, introduced by Chinneck, to compute an *Irreducible Infeasible Subsystem* (IIS). His recent book[6] provides all background analysis, including the following:

Counterexample. $y - \sqrt{x} = 0,\ x \geq 0,\ y \leq -1.$

If the algorithm drops $x \geq 0$, the solver issues an error message and the algorithm cannot proceed.

Unlike LP, such logical constraints may be needed in NLP. Another source of failure is the inability of the NLP solver to determine whether a nonlinear system is feasible. This is more difficult than for an LP.

NLP References

[1] L. Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1):1–3, 1966.

[2] M. Ascher. Cycling in the Newton-Raphson algorithm. *International Journal of Mathematical Education in Science and Technology*, 5:229–235, 1974.

[3] J. M. Ash and H. Sexton. A surface with one local minimum. *Mathematics Magazine*, 58(3):147–149, 1985.

[4] D. M. Bloom. FFF #34. The shortest distance from a point to a parabola. *The College Mathematics Journal*, 22(2):131, 1991. *Fallacies, Flaws, and Flimflam* is edited by E. Barbeau.

[5] J. F. Bonnans, E. R. Panier, André Tits, and J. L. Zhou. Avoiding the Maratos effect by means of a nonmonotone line search II: Inequality constrained problems — feasible iterates. *SIAM Journal on Numerical Analysis*, 29(4):1187–1202, 1992.

[6] J. W. Chinneck. *Feasibility and Infeasibility in Optimization: Algorithms and Computational Methods*, volume 118 of *International Series in Operations Research & Management Science*. Springer, New York, NY, 2008.

[7] G. B. Dantzig, J. Folkman, and N. Shapiro. On the continuity of the minimum set of a continuous function. *Journal of Mathematical Analysis and Applications*, 17(3):519–548, 1967.

[8] G. C. Donovan, A. R. Miller, and T. J. Moreland. Pathological functions for Newton's method. *Teaching Numerical Analysis*, 100(1):53–58, 1993.

[9] J. P. Dussault and G. Fournier. On the convergence of the projected gradient method'. *Journal of Optimization Theory and Applications*, 77(1):197–208, 1993.

[10] J. P. Evans and F. J. Gould. Stability in nonlinear programming. *Operations Research*, 18(1):107–118, 1970.

[11] K. Fan, I. Glicksburg, and A. J. Hoffman. Systems of inequalities involving convex functions. *American Mathematical Society Proceedings*, 8(3):617–622, 1957.

[12] W. Fenchel. On conjugate convex functions. *Canadian Journal of Mathematics*, 1(1):73–77, 1949.

[13] W. Fenchel. Convex cones, sets and functions. Lecture notes, Princeton University, Department of Mathematics, Princeton, NJ, 1953.

[14] R. Fletcher, S. Leyffer, and Ph. Toint. A brief history of filter methods. Preprint ANL/MCS-P1372-0906, Argonne National Laboratory, Mathematics and Computer Science Division, Argonne, IL, 2006.

[15] N. I. M. Gould and Ph. L. Toint. A note on the convergence of barrier algorithms to second-order necessary points. *Mathematical Programming A*, 85(2):433–438, 1999.

[16] H. J. Greenberg. Lagrangian duality gaps: Their source and resolution. Technical Report CP-69005, CS/OR Center, Southern Methodist University, Dallas, Texas, 1969.

[17] H. J. Greenberg. Interpreting rate analysis. *SIGMAP Bulletin*, 17:33–37, 1974.

[18] H. J. Greenberg. The one dimensional generalized Lagrange multiplier problem. *Operations Research*, 25(2):338–345, 1977.

[19] H. J. Greenberg. Wolfe's example and the zigzag phenomenon. In A. G. Holder, editor, *Mathematical Programming Glossary*, http://glossary.computing.society.informs.org/notes/wolfe.pdf, 1996. INFORMS Computing Society.

[20] H. J. Greenberg and W. P. Pierskalla. A review of quasi-convex functions. *Operations Research*, 19(7):1553–1570, 1971.

[21] H. J. Greenberg and W. P. Pierskalla. Extensions of the Evans-Gould stability theorems for mathematical programs. *Operations Research*, 20(1):143–153, 1972.

[22] L. Grippo, F. Lampariello, and S. Lucidi. Nonmonotone line search technique for Newton's method. *SIAM Journal on Numerical Analysis*, 23(4):707–716, 1986.

[23] Ch. Grossmann. A counterexample to Rosen's decomposition method. *Optimization*, 13(4):511–514, 1982.

[24] M. Halická, E. de Klerk, and C. Roos. On the convergence of the central path in semidefinite optimization. *SIAM Journal on Optimization*, 12(4):1090–1099, 2002.

[25] H. Hancock. *Theory of Maxima and Minima*. Dover, New York, NY, 1960. Note: original version, 1917

[26] X.-D. Hu. A counterexample of the convergence of Rosen's algorithm with inexact line search. *Operations Research Letters*, 13(2):95–97, 1993.

[27] S. Karamardian. *Duality in Mathematical Programming*. Ph.D. thesis, University of California, Berkely, CA, 1965.

[28] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In J. Neyman, editor, *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, Berkeley, CA, 1951. University of California Press.

[29] D. MacMillan. *Relaxing Convergence Conditions to improve the Convergence Rate*. Ph. D. thesis, University of Colorado Denver, Department of Mathematical Sciences and Statistics, Denver, CO, 1999. available at http://www-math.cudenver.edu/graduate/thesis/macmillan.pdf.

[30] N. Maratos. *Exact Penalty Function Algorithms for Finite Dimensional and Control Optimization Problems*. Ph. D. thesis, University of London, London, UK, 1978.

[31] B. Martos. Quasi-convexity and quasi-monotonicity in non-linear programming. *Studia Scientiarum Mathematicarum Hungarica*, 2:265–273, 1967.

[32] K. I. M. McKinnon. Convergence of the nelder-mead simplex method to a nonstationary point. *SIAM Journal on Optimization*, 9(1):148–158, 1998.

[33] E. R. Panier and André Tits. Avoiding the Maratos effect by means of a nonmonotone line search I: General constrained problems. *SIAM Journal on Numerical Analysis*, 28(4):1183–1195, 1991.

[34] M. J. D. Powell. On search directions for minimization algorithms. *Mathematical Programming*, 4:193–201, 1973.

[35] L. Qi. On an extended Lagrange claim. *Journal of Optimization Theory and Applications*, 108(3):685–688, 2001.

[36] F. Vanden Berghen. *CONDOR: a constrained, non-linear, derivative-free parallel optimizer for continuous, high computing load, noisy objective functions.* Ph. D. thesis, Université Libre de Bruxelles, Faculté des Sciences Appliquées, Brussels, BE, 2004.

[37] A. Wächter. *An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering.* Ph. D. thesis, Carnegie Mellon University, Department of Chemical Engineering, Pittsburgh, PA, 2002.

[38] A. Wächter and L. T. Biegler. Failure of global convergence for a class of interior point methods for nonlinear programming. *Mathematical Programming*, A 88(3):565–574, 2000.

[39] P. Wolfe. Convergence conditions for ascent methods. *SIAM Review*, 11(2):226–235, 1969.

[40] P. Wolfe. Convergence conditions for ascent methods. II: Some corrections. *SIAM Review*, 13(2):185–188, 1971.

[41] P. Wolfe. On the convergence of gradient methods under constraint. *IBM Journal of Research and Development*, 16(4):407–411, 1972.

[42] P. B. Zwart. Nonlinear programming: Counterexamples to two global optimization algorithms. *Operations Research*, 21(6):1260–1266, 1973.

Multiple-Objective Programming

The multiple-objective mathematical program has the range of the objective function in $\mathbb{R}^N$ with $N > 1$. An optimal solution is defined as follows. A point $x \in X$ is *dominated* by $x' \in X$ if $f(x') \geq f(x)$ and $f_i(x') > f_i(x)$ for some i. (Reverse the inequalities for minimization.) A *Pareto-optimum* is a feasible solution that is not dominated. This is denoted:

$$\text{Pareto-max } f(x) : x \in X,\ g(x) \leq 0,\ h(x) = 0.$$

The *Pareto frontier* is the set of Pareto-optima, denoted

$$\text{argPareto-max}\{f(x) : x \in X,\ g(x) \leq 0,\ h(x) = 0\}.$$

One way to generate a Pareto-optimal is by taking a strictly-positive combination of the objective functions and solving:

$$\max \textstyle\sum_{i=1}^{N} w_i f_i(x) : x \in X,\ g(x) \leq 0,\ h(x) = 0,$$

where $w > 0$. This is sometimes called the *weighted-objective* model, and each optimal solution is Pareto-optimal. Typically, but not always, the weights are normalized by $\sum_{i=1}^{N} w_i = 1$.

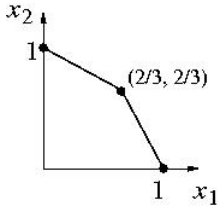
For a multiple-objective standard *Linear Program* (LP), the form is given by:

$$\text{Pareto-max } Cx : Ax = b,\ x \geq 0,$$

where C is $N \times n$.

MOP Myth 1. *For LP, the Pareto frontier is a convex set.*

Counterexample. The following has its Pareto frontier along the two edges defined by $2x_1 + x_2 = 2$ and $x_1 + 2x_2 = 2$, respectively. In particular, points $(1, 0)$ and $(0, 1)$ are Pareto-optimal, but their midpoint, $(1/2, 1/2)$, is dominated by $(2/3, 2/3)$

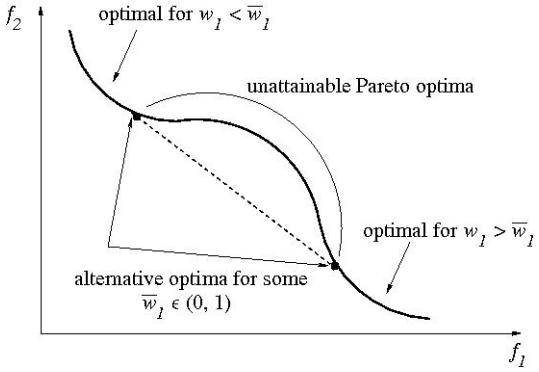


$$\text{Pareto-max} \quad \left\{ \begin{matrix} x_1 \\ x_2 \end{matrix} \right\} : x \geq 0$$
$$2x_1 + x_2 \leq 2$$
$$x_1 + 2x_2 \leq 2$$

MOP Myth 2. *Varying the weights of a convex combination of objectives generates the Pareto frontier.*

Although it is trivial to show that a solution to the weighted-objective (with $w > 0$) is a Pareto-optimum, the myth asserts the converse: each Pareto-optimum can be generated by some positive weight.

The problem is the same as the duality gap. In particular, the Lagrangian has a duality gap when the optimal response function is not convex (for minimization). That is what happens when the Pareto frontier does not produce a convex function in f_1-f_2 space. (See figure on right.)



Das and Dennis[5] provide the following:

Counterexample.

$$\text{Pareto-min} \quad \left\{ \begin{matrix} f_1(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2, \\ f_2(x) = 3x_1 + 2x_2 - \frac{1}{3}x_3 + 0.01(x_4 - x_5)^3 \} \end{matrix} \right\} :$$
$$x_1 + 2x_2 - x_3 - 0.5x_4 + x_5 = 2$$
$$4x_1 - 2x_2 + 0.8x_3 + 0.6x_4 + 0.5x_5^2 = 0$$
$$x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 \leq 10.$$

Also see Steuer[21, p. 439] .

MOP Myth 3. **new** *Adding an objective function expands the set of Pareto-optima.*

Let X_N^* be the set of Pareto-optima for $f = (f_1, \ldots, f_N)^\intercal$. The Myth asserts $X_N^* \subseteq X_{N+1}^*$.

Counterexample. Lowe, Thisse, Ward, and Wendell[16] provide the following. Let f_1 have two maxima, x^1 and x^2, such that $f_2(x^1) < f_2(x^2)$. Then, $x^1 \in X_1^*$ but $x^1 \notin X_2^*$.

MOP Myth 4. *Consider the multiple-objective convex program:*

$$\text{Pareto-min } f(x) : x \in \mathbb{R}^n,\ g(x) \leq 0,$$

where f and g are convex and differentiable. Then, x^* is a Pareto-optimal if $g(x^*) \leq 0$, and there exists $w, \lambda \geq 0$ such that $\lambda g(x^*) = 0$ and

$$w\nabla f(x^*) + \lambda \nabla g(x^*) = 0.$$

The basis for this is that these are the Lagrangian (Kuhn-Tucker-Karush) conditions for the weighted model. (The sufficiency is due to the convexity assumptions.)

Kim, Lee, and Cho[14] provide the following:

Counterexample.

$$\text{Pareto-min } \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} : x_1 \geq 0,\ x_1(x_1 - 1) \leq x_2.$$

Consider $x^* = (0, 1)$. This is not a Pareto-optimum because it is dominated by $(0, 0)$. Let $w = \lambda = (1, 0)$, so the conditions stated in the myth are satisfied with:

$$(1,0)\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + (1,0)\begin{bmatrix} -1 & 0 \\ -1 & -1 \end{bmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

MOP Myth 5. *Consider a multiple-objective LP in standard form. A Pareto-maximum can be obtained from a weighted objective, where the weights (w) are obtained from a solution to:*

$$\min b^\intercal u : u^\intercal A - wC \geq 0,\ w \geq 1.$$

Isermann[12] proposed this with the intuition that this is a sort-of dual to the original Pareto-maximum in the sense that the weighted objective wCx yields this LP (but with w fixed). The goal here is to obtain some initial Pareto-maximum, then find others. The following is due to Ecker and Hegner[6]:

Counterexample.

$$\text{Pareto-max } \begin{pmatrix} -x_1 \\ x_4 \end{pmatrix} : x \geq 0,$$

$$\begin{array}{rcrcrcl} x_1 & - & x_2 & & & = & 1 \\ x_1 & & & + & x_3 & = & 2 \\ & & & & x_4 & = & 1 \end{array}$$

A Pareto-maximum is $x = (1, 0, 1, 1)$. The LP to obtain the weights is given by:

$$\min\ u_1 + 2u_2 + u_3 :\ w_1, w_2 \geq 1$$

$$\begin{array}{rcrcrcl} u_1 & + & u_2 & & + & w_1 & \geq\ 0 \\ -u_1 & & & & & & \geq\ 0 \\ & & u_2 & & & & \geq\ 0 \\ & & & u_3 & - & w_2 & \geq\ 0 \end{array}$$

This is unbounded because we can let $(u, w) = (-2t, 0, t, 2t, t)$, which is feasible for all $t \geq 1$. The minimand is $-t$, which diverges to $-\infty$ as $t \to \infty$.

Ecker and Kouada[7] give the correct result as follows. Suppose $Cx^0 \neq 0$ for some feasible x^0. Then, there exists a Pareto-maximum if, and only if, the following LP has an optimum:

$$\max \textstyle\sum_{i=1}^{N} s_i :\ Cx = s + Cx^0,\ Ax \leq b,\ x, s \geq 0.$$

(Omitting Cx^0, the dual is Isermann's LP. The counterexample shows that Cx^0 cannot be omitted.)

In the counterexample, x^0 does not exist because $Cx \neq 0$ for any feasible x. See Benson[3] for additional discussion and another way to get an initial Pareto-maximum that is an extreme point of the feasible polyhedron.

MOP Myth 6. *Let U be an increasing utility function on the range of f on X, and*

$$\max\ U(f(x)) :\ x \in X.$$

Then, an extreme point with greatest utility value is Pareto-optimal.

Steuer[21, p. 157] provides the following:

Counterexample.

$$\text{Pareto-max} \left\{ \begin{matrix} x_1 \\ x_2 \end{matrix} \right\} :$$

$$\begin{array}{rcrcr} x_1 & + & x_2 & \leq & 18 \\ 8x_1 & + & 6x_2 & \geq & 112 \\ 5x_1 & + & 7x_2 & \geq & 96 \end{array}$$

$$x \geq 0.$$

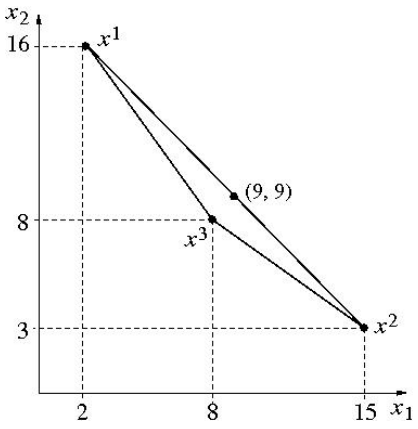


Figure taken from [21].

This has three extreme points:

$$x^1 = (2, 16),\ x^2 = (15, 3),\ x^3 = (8, 8).$$

x^3 is not Pareto-optimal because it is dominated by the non-extreme point $(9, 9)$. However, for $U(f_1, f_2) = f_1 f_2$, x^3 has the greatest utility value (64); the two Pareto-optimal extreme points have lower utility values:

$$U(f(x^1)) = 32,\ U(f(x^2)) = 45.$$

MOP Myth 7. *In a multiple-objective LP, one should put the greatest weight on the most important objective.*

Steuer[21, p. 198–9] provides the following:

Counterexample.

$$\text{Pareto-max} \begin{pmatrix} 2x_1 + 4x_2 \\ 3x_1 - 3x_2 \\ 5x_2 \end{pmatrix} : \begin{array}{r} x \geq 0,\ x_2 \leq 12 \\ 3x_1 + 5x_2 \leq 72 \\ 3x_1 - 5x_2 \leq 12 \end{array}$$

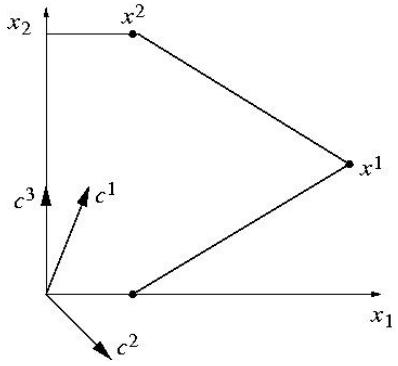


Figure taken from [21].

Assume the objectives are in order of importance, and consider the following two weights: $w^1 = (0.7, 0.2, 0.1)$ and $w^2 = (0, 0.1, 0.9)$. The first weight reflects the relative importance of the objectives and generates the Pareto-optimum point $x^1 = (14, 6)$, with objective values $(52, 33, 30)$. The second one is contrary to the relative importance and generates the Pareto-optimum point $x^2 = (4, 12)$, with objective values $(56, -24, 60)$.

These are counter-intuitive results because x^2 better reflects the objectives' relative importance. The outcome, particularly the latter, where x^1 is the solution, is due to the *correlation* between c^1 and c^3. By placing a large weight on c^3, it is not necessary to place a high weight on c^1.

MOP Myth 8. *All Pareto-optimal solutions are equal.*

The issue is that of *value trade-off*, say between two objectives that are in conflict. In particular, suppose $f_1(x)$ is cost (in USD) for decision x, and $f_2(x)$ is risk. Keeney[13] provides the following:

Counterexample. Suppose it costs \$3 billion annually if carbon monoxide concentrations are limited to 3 parts per million, and suppose that it costs \$6 billion if concentrations are

held to 2 parts per million. We must ask, "Is it worth \$3 billion to lower concentrations from 3 parts per million to 2 parts per million?" The appropriate way to address this is to model the causal relationship between pollutant concentrations and potential health effects. Then, one could deal directly with the value trade-offs between cost of the national air quality standard and the health effects averted.

The two Pareto-optima solutions are not equal, but the model focused attention on them for further analysis.

By its definition, points on the Pareto frontier are indifferent to each other as far as the model is concerned. Applying multiple-objective programming to decision-making, however, we must go beyond finding points on the Pareto frontier. Also, MOP Myth 7 demonstrates that using weights may not resolve the issue. It is important that Pareto-optima help to focus what can be achieved, but ultimately there is a trade-off in the value of one Pareto-optimum versus another. That trade-off could be subjective or with refined analysis. Keeney gives details on this, citing 12 common mistakes in making value trade-offs.

MOP Background — Pareto-optimum Graph

The following benefitted from comments by Jochen Gorski.

Define a *Pareto-optimum graph* whose nodes correspond to some finite set of Pareto-optimum solutions and whose edges correspond to their *adjacency*. The notion of the finite set of solutions and of their adjacency are not defined in general. For LP, it is natural to define Pareto-optimal adjacency the same as in LP: the nodes are basic optimal solutions, and their adjacency is that of their bases. For combinatorial problems, it is natural to use underlying combinatorial structures. For example, two spanning trees are adjacent if they differ by one edge (having $n-2$ edges in common).

The significance of a Pareto-optimum graph is its connectedness, raising the question if one can traverse the nodes without having to compute a solution that is not Pareto-optimal. If so, this enables neighborhood search to produce them. (See Gorski, Klamroth, and Ruzika[10] for a substantive description of this concept and an up-to-date review of results.)

MOP Myth 9. *The Pareto-minimum graph for spanning trees is connected.*

Ehrgott and Klamroth[8] provide the following:

Counterexample. The edge numbers are the costs of two objectives. In particular, $(0,0)$ is an edge with zero cost in both objectives, as the edge (s_1, s_{11}).

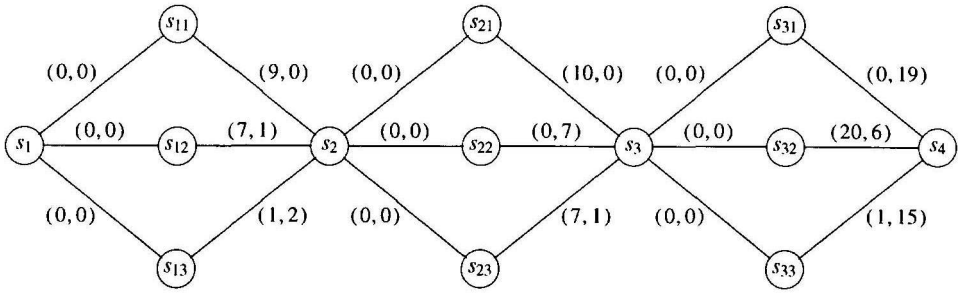


Figure taken from [8].

The following table lists the 12 Pareto-minimum spanning trees, showing the edges with positive costs (all edges with zero cost are in each tree).

Pareto-minimum Tree	Edges with Non-zero Cost	Objective Values
T_1	(s_{13}, s_2), (s_{22}, s_3), (s_{31}, s_4)	$(1, 28)$
T_2	(s_{13}, s_2), (s_{22}, s_3), (s_{33}, s_4)	$(2, 24)$
T_3	(s_{13}, s_2), (s_{23}, s_3), (s_{31}, s_4)	$(8, 22)$
T_4	(s_{13}, s_2), (s_{23}, s_3), (s_{33}, s_4)	$(9, 18)$
T_5	(s_{13}, s_2), (s_{21}, s_3), (s_{33}, s_4)	$(12, 17)$
T_6	(s_{11}, s_2), (s_{23}, s_3), (s_{33}, s_4)	$(12, 17)$
T_7	(s_{11}, s_2), (s_{21}, s_3), (s_{33}, s_4)	$(17, 16)$
T_8	(s_{12}, s_2), (s_{22}, s_3), (s_{32}, s_4)	$(20, 15)$
T_9	(s_{13}, s_2), (s_{23}, s_3), (s_{32}, s_4)	$(27, 14)$
T_{10}	(s_{13}, s_2), (s_{21}, s_3), (s_{32}, s_4)	$(28, 9)$
T_{11}	(s_{11}, s_2), (s_{23}, s_3), (s_{32}, s_4)	$(36, 7)$
T_{12}	(s_{11}, s_2), (s_{21}, s_3), (s_{32}, s_4)	$(39, 6)$

Tree T_8 is not adjacent to any other Pareto-minimum spanning tree.

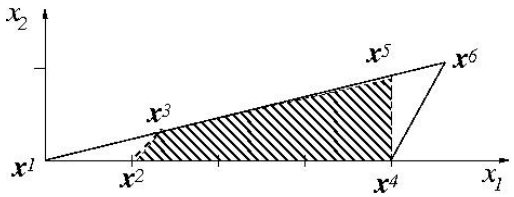
An implication is that to visit each Pareto-minimum spanning tree, we may need to visit a non-optimal spanning tree during the pivoting process. See Przybylski, Gandibleux, and Ehrgott[19] for how this invalidates a class of algorithms that seek to generate Pareto-optimal spanning trees and shortest paths.

MOP Myth 10. *The Pareto frontier is closed.*

The result is true for LP, but Kornbluth and Steuer[15] provide the following for a fractional program:

Counterexample.

$$\text{Pareto-max} \begin{pmatrix} \frac{x_1 - 4}{3 - x_2} \\ \frac{-x_1 + 4}{x_2 + 1} \\ -x_1 + x_2 \end{pmatrix} : \begin{array}{c} -x_1 + 4x_2 \leq 0 \\ x_1 - \frac{1}{2}x_2 \leq 4 \\ x \geq 0 \end{array}$$



$x^1 = (0,0)$ $\quad x^2 = (1,0)$
$x^3 = (\frac{4}{3}, \frac{1}{3})$ $\quad x^4 = (4,0)$
$x^5 = (4,1)$ $\quad x^6 = (\frac{32}{7}, \frac{8}{7})$

The feasible region is the convex hull of the extreme points, denoted $\text{convh}\{x^1, x^4, x^6\}$. The objective values for each of the points are:

$$f(x^1) = \begin{pmatrix} -\frac{4}{3} \\ 4 \\ 0 \end{pmatrix} \qquad f(x^2) = \begin{pmatrix} -1 \\ 3 \\ -1 \end{pmatrix}$$

$$f(x^3) = \begin{pmatrix} -1 \\ 2 \\ -1 \end{pmatrix} \qquad f(x^4) = \begin{pmatrix} 0 \\ 0 \\ -4 \end{pmatrix}$$

$$f(x^5) = \begin{pmatrix} 0 \\ 0 \\ -3 \end{pmatrix} \qquad f(x^6) = \begin{pmatrix} \frac{4}{13} \\ -\frac{4}{15} \\ -\frac{24}{7} \end{pmatrix}.$$

The Pareto frontier is given by the union of convex hulls minus two half-open line segments:

$$X^* = \text{convh}\{x^2, x^3, x^4, x^5\} \cup [x^1, x^2] \cup [x^5, x^6] - (x^2, x^3] - [x^4, x^5),$$

where $-$ denotes the set-minus. Points x^3 and x^4 are not Pareto-optimal, but they are in the closure of X^*. In particular, x^3 is dominated by x^2, but all feasible points on $\{(x^3, x^3 + (\varepsilon, 0)) : \varepsilon > 0\}$ are not dominated — they are Pareto-optimal. Thus, x^3 is a cluster point of X^*, so X^* is not closed.

MOP Myth 11. *If the Pareto frontier contains an interior point, it must contain all interior points.*

The result is true for LP, but the fractional program given by Kornbluth and Steuer[15] in MOP Myth 10 provides the following:

Counterexample. Interior points in $\text{convh}\{x^2, x^3, x^4, x^5\}$ are Pareto-optimal, but those in $\text{convh}\{x^1, x^2, x^3\} \cup \text{convh}\{x^4, x^5, x^6\}$ are not.

MOP Myth 12. *The Pareto frontier is edge-connected.*

The result is true for LP, but the fractional program given by Kornbluth and Steuer[15] in MOP Myth 10 provides the following:

Counterexample. Points x^1 and x^6 are Pareto-optimal, but they are not edge-connected because the edge $(x^2, x^3]$ is not in X^* (neither is $[x^4, x^5)$).

MOP Background — Lexico-optima and Bilevel Programs

Another way to generate Pareto-optimal solutions is by ordering the objective functions and solving sequentially. Suppose $f_1 \succ f_2 \succ \cdots \succ f_N$. Then, the *lexico-optimum* model is given by:

$$\begin{array}{rcl} X^1 & = & \text{argmax}\{f_1(x) : x \in X\} \\ X^2 & = & \text{argmax}\{f_2(x) : x \in X^1\} \\ \vdots & & \\ X^N & = & \text{argmax}\{f_N(x) : x \in X^{N-1}\} \end{array}$$

(The sequence stops if it is initially infeasible or encounters an unbounded solution.) The points in X^N are Pareto-optimal, with f lexio-graphically ordered: $f(x^*) \succeq f(x)$ for $x^* \in X^N$ and $x \in X$. (See Sherali and Soyster[20].)

As a practical matter, the sequence is modified by tolerances that allow a small amount of sub-optimality, which often results in X^k containing more than one point — that is, near-optimal solutions. Given $\tau = (\tau_1, \ldots, \tau_N) \geq 0$,

$$\begin{array}{ll} z_1 = \max\{f_1(x) : x \in X\} & X^1 = \{x \in X \quad : f_1(x) \geq z_1 - \tau_1\} \\ z_2 = \max\{f_2(x) : x \in X^1\} & X^2 = \{x \in X^1 \quad : f_2(x) \geq z_2 - \tau_2\} \\ \vdots & \vdots \\ z_N = \max\{f_N(x) : x \in X^{N-1}\} & X^N = \{x \in X^{N-1} \quad : f_N(x) \geq z_N - \tau_N\} \end{array}$$

Related to $N = 2$, we have the *bilevel* mathematical program:

$$\max\ f_2(x, y^*) : x \in X,\ y^* \in \text{argmax}\{f_1(x, y) : y \in Y(x)\}.$$

This also represents the ordered preference $f_1 \succ f_2$, but the inner optimality constraint is a restriction that y^* be optimal in the priority objective, whereas the lexico-optimum second problem would include x as:

$$\max\ f_2(x^*, y^*) : (x^*, y^*) \in \text{argmax}\{f_1(x, y) : x \in X,\ y \in Y(x)\}.$$

See MOP Myth 14 to avoid thinking the bilevel solution is the Pareto-optimum:

$$\text{Pareto-max} \begin{Bmatrix} f_1(x, y) \\ f_2(x, y) \end{Bmatrix} : (x, y) \in \mathcal{X},$$

where $\mathcal{X} = \{(x, y) : x \in X,\ y \in Y(x)\}$. Also, see Fliege and Vicente[9] for a recent analysis of their relationship.

MOP Myth 13. *Every Pareto-optimum is a solution to the lexico-optimum of some lexico-ordering.*

Counterexample. Consider

$$\text{Pareto-max} \begin{pmatrix} x \\ y \end{pmatrix} : x, y \geq 0,\ 2x + y \leq 2,\ x + 2y \leq 2.$$

Using the given order, we first maximize x and obtain $X^1 = \{(1,0)\}$. Since this is unique, the generated Pareto-optimum is $(1,0)$. Reversing the order, we maximize y and obtain $X^1 = \{(0,1)\}$. Again, since this is unique, the generated Pareto-optimum is $(0,1)$. Thus, the generated solutions are two points, but the Pareto frontier contains the point $(\frac{2}{3}, \frac{2}{3})$ (generated by weights $w = (\frac{1}{2}, \frac{1}{2})$).

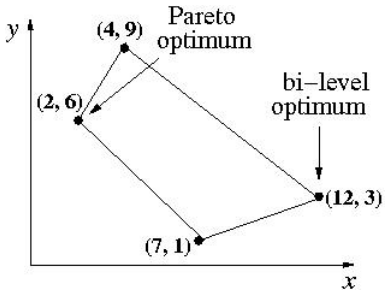
MOP Myth 14. *A solution to the bilevel mathematical program can be found with some weighted objective solution.*

The myth says there exists $w \in [0,1]$ such that a solution to the bilevel mathematical program is found by solving:

$$\max\ wf_1(x,y) + (1-w)f_2(x,y) : x \in X,\ y \in Y(x).$$

Counterexample. Haurie, Savard, and White[11] provide the following:

$$\begin{array}{rcrcr} \max & -x & -5y^* & : & x \geq 0, \\ y^* \in Y(x) = \text{argmax}\{y & : & y & \geq & 0 \\ x & + & y & \geq & 8 \\ -3x & + & 2y & \leq & 6 \\ 3x & + & 4y & \leq & 48 \\ 2x & - & 5y & \leq & 9\}. \end{array}$$



The optimal bilevel solution is at $(x^*, y^*) = (12,3)$, but a weighted-objective solution is given by other extreme points of the polyhedron. Here are the weighted-objective solutions for ranges of w:

w-range	optimal extreme point
$0 \leq w \leq 0.15$	$(2,6)$
$0.15 \leq w \leq 0.20$	$(4,9)$
$0.20 \leq w \leq 1$	$(7,1)$

The only extreme point with a greater value of the first-level objective is $(7,1)$, but $1 \notin Y(7)$, so it is not feasible in the bilevel model. The bilevel solution $(12,3)$ is not a solution to any of the weighted-objective models. If it were, it would be Pareto-optimal; Haurie et al. point out that bi-criteria solutions are generally not Pareto-optimal.

Also see the counterexamples by Candler[4] , Wen and Hsu[23]. Further, the counterexample by Ben-Ayed and Blair[2] is for the Grid Search Algorithm, which rests on this myth. See Marcotte[17] for a counterexample to a solution for the equilibrium network design problem based on the same myth.

MOP Myth 15. **new** *An optimal solution to the linear bilevel program is either Pareto-optimal, or it solves the outer LP.*

Counterexample. Wen and Hsu[23] provide the following:

$$\begin{array}{l}\max\ f_2(x,y) = -2x + 11y :\ x \geq 0,\\ y^* \in \operatorname{argmax}\{f_1(y) = -3y :\ y \geq 0,\\ \begin{array}{rcrcr} x & - & 2y & \leq & 4 \\ 2x & - & y & \leq & 24 \\ 3x & + & 4y & \leq & 96 \\ x & + & 7y & \leq & 126 \\ -4x & + & 5y & \leq & 65 \\ x & + & 4y & \geq & 8\}. \end{array}\end{array}$$

The bilevel solution is at $(x^*, y^*) = \frac{1}{11}(192, 120)$. It is not Pareto-optimal because it is dominated by $(x, y) = \frac{1}{11}(19, 108)$ with the objective values: $f(x^*, y^*) = (-32.727, 85.091)^{\mathsf{T}} < f(x, y) = (-29.45, 104.55)^{\mathsf{T}}$. The myth asserts that (x^*, y^*) must solve the "outer LP:"

$$\begin{array}{l}\max\ -2x + 11y :\ x, y \geq 0\\ \begin{array}{rcrcr} x & - & 2y & \leq & 4 \\ 2x & - & y & \leq & 24 \\ 3x & + & 4y & \leq & 96 \\ x & + & 7y & \leq & 126 \\ -4x & + & 5y & \leq & 65 \\ x & + & 4y & \geq & 8\}. \end{array}\end{array}$$

The optimal solution is $(x, y) = (5.333, 0.667)$ with $f_2(x, y) = -8.667 > f_2(x^*, y^*) = -32.727$.

MOP Myth 16. **new** *A linear bilevel optimum is Pareto-optimal if the coefficient vectors of the inner variable forms an acute angle.*

The linear bilevel program is:

$$\max\ cx + dy^* :\ x \geq 0,\ Ax \leq b,\ y^* \in \operatorname{argmax}\{fy :\ y \geq 0,\ Fx + Gy \leq g\}.$$

The myth asserts that if (x^*, y^*) is a bilevel optimum, it is Pareto-optimal if $df^{\mathsf{T}} > 0$.

The intuition behind this is as follows. From 1983[1]–1988[4, 17] it was believed that the linear bilevel program is Pareto-optimal for:

$$\max \begin{pmatrix} cx + dy \\ fy \end{pmatrix} :\ x, y \geq 0,\ Ax \leq b,\ Fx + Gy \leq g.$$

The rationale is that weights can purportedly be established using the Lagrange (Karush-Kuhn-Tucker) conditions, so that the bilevel program must solve

$$\max\ \lambda(cx + dy) + (1 - \lambda)fy :\ x, y \geq 0,\ Ax \leq b,\ Fx + Gy \leq g$$

for some $\lambda \in [0,1]$. Before that myth was dispelled (see MOP Myth 14), Ünlü[22] proceeded to correct this by pointing out that Pareto-optimality is not ensured for $\lambda = 1$, resulting in MOP Myth 15. That case is the outer LP — Wen and Hsu[23] gave a counterexample to show that this is not always true. Applying the Kuhn-Tucker conditions, they proposed that Ünlü's theorem is true if $df \geq 0$.

Counterexample. Marcotte and Savard[18] provide the following:

$$\max\ f_2(x,y) = -x - 2y_1 - 2y_2 : x \geq 0,\ x \leq 1,$$
$$y^* \in \operatorname{argmax}\{f_1(y) = y_1 - 2y_2 : y \geq 0, x + y_1 \leq 2\}.$$

We have $df^\tau = (2,2)(-1,2)^\tau > 0$. The bilevel optimum is at $(1,1,0)$ This is not Pareto-optimal because it is dominated by $(x,y) = (0.5, 1.1, 0)$ with $f_1(x,y) = -2.7 > f_1(x^*,y^*) = -3$ and $f_2(x,y) = 1.1 > f_2(x^*,y^*) = 1$.

MOP Myth 17. **new** *The Parametric Complementary Pivot algorithm obtains a bilevel solution.*

Counterexample. Ben-Ayed and Blair[2] provide the following:

$$\max\ 1.5x + 6y_1^* + y_2^* : 0 \leq x \leq 1$$
$$y^* \in \operatorname{argmax}\{y_1 + 5y_2 : y \geq 0,\ x + 3y_1 + y_2 \leq 5,\ 2x + y_1 + 3y_2 \leq 5\}.$$

The PCP algorithm searches for a solution to the feasibility and complementary slackness conditions:

$$\begin{array}{ccc} x + 3y_1 + y_2 + s_1 = 5 & 0.01y_1 + 3u_1 + u_2 - t_1 = 1 & 1.5x + 6y_1 + y_2 - z = 2 \\ 2x + y_1 + 3y_2 + s_2 = 5 & 0.01y_1 + u_1 + 3u_2 - t_2 = 5 & yt = u(s_1,s_2)^\tau = 0 \\ x + s_3 = 1 & x,y,s,t,u,z \geq 0. & \end{array}$$

The algorithm starts by ignoring the inner maximization (but does satisfy the constraints). That solution is $x = 0$ and $y = (1.667, 0)$, with $s = (0, 3.333, 1)$. The complementary slackness conditions require $t_1 = 0$ and $u = (0.328, 0)$. The middle equation then yields $t_2 = .01667 + 0.328 - 5 < 0$, so an artificial variable, w, is introduced:

$$0.01y_1 + u_1 + 3u_2 - t_2 + w = 5.$$

Entering x, y_2, u_2, s_1, or t_1 decreases w. However, u_2 cannot enter because $s_2 > 0$; similarly, neither s_1 nor t_1 can enter. If we choose y_2 to enter, s_2 leaves. At the next step, we may have u_2 enter (u_1 leaves), then s_1 enters to produce the system:

$$y_1 + 0.147x - 0.059s_2 - 0.176z = 0.059.$$

At this point, the PCP algorithm stops with the conclusion that the system has no solution. However, a solution is: $x = 1$, $y = (0,1)$, $s = (3,0,0)$, $u = (0, 1.663)$, $t = (0.663, 0)$, $z = 0.5$.

Ben-Ayed and Blair prove that the bilevel LP is NP-hard, so no polynomial algorithm can ensure optimality (unless $P = NP$).

MOP References

[1] J. F. Bard. An efficient point algorithm for a linear two-stage optimization problem. *Operations Research*, 31(4):670–684, 1983.

[2] O. Ben-Ayed and C. E. Blair. Computational difficulties of bilevel linear programming. *Operations Research*, 38(3):56–560, 1990.

[3] H. P. Benson. Finding an initial efficient extreme point for a linear multiple objective program. *Journal of the Operational Research Society*, 32(6):495–498, 1981.

[4] W. Candler. A linear bilevel programming algorithm: A comment. *Computers & Operations Research*, 15(3):297–298, 1988.

[5] J. Das and J. E. Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural Optimization*, 14(4):63–69, 1997.

[6] J. G. Ecker and N. S. Hegner. On computing an initial efficient extreme point. *Journal of the Operational Research Society*, 29(10):1005–1007, 1978.

[7] J. G. Ecker and I. A. Kouada. Finding efficient points for linear multiple objective programs. *Mathematical Programming*, 8:375–377, 1975.

[8] M. Ehrgott and K. Klamroth. Connectedness of efficient solutions in multiple criteria combinatorial optimization. *European Journal of Operational Research*, 97(1):159–166, 1997.

[9] J. Fliege and L. N. Vicente. Multicriteria approach to bilevel optimization. *Journal of Optimization Theory and Applications*, 131(2):209–225, 2006.

[10] J. Gorski, K. Klamroth, and S. Ruzika. Connectedness of efficient solutions in multiple objective combinatorial optimization. Preprint 310, Institute of Applied Mathematics, University of Erlangen-Nuremberg, Erlangen, Germany, 2006.

[11] A. Haurie, G. Savard, and D. J. White. A note on: An efficient point algorithm for a linear two-stage optimization problem. *Operations Research*, 38(3):553–555, 1990.

[12] H. Isermann. The enumeration of the set of all efficient solutions for a linear multiple objective program. *Operational Research Quarterly*, 28(3, Part2):711–725, 1977.

[13] R. L. Keeney. Common mistakes in making value trade-offs. *Operations Research*, 50(6):925–925, 2002.

[14] D. S. Kim, G. M. Lee, B. S. Lee, and S. J. Cho. Counterexample and optimality conditions in differentiable multiobjective programming. *Journal of Optimization Theory and Applications*, 109(1):187–192, 2001.

[15] S. H. Kornbluth and R. E. Steuer. Multiple objective linear fractional programming. *Management Science*, 27(9):1024–1039, 1981.

[16] T. J. Lowe, J.-F Thisse, J. E. Ward, and R. E. Wendell. On efficient solutions to multiple objective mathematical programs. *Management Science*, 30(11):1346–1349, 1984.

[17] P. Marcotte. A note on a bilevel programming algorithm by Leblanc and Boyce. *Transportation Research*, 22B(3):233–237, 1988.

[18] P. Marcotte and G. Savard. A note on the pareto optimality of solutions to the linear bilevel programming problem. *Computers & Operations Research*, 18(4):355–359, 1991.

[19] A. Przybylski, X. Gandibleux, and M. Ehrgott. The biobjective integer minimum cost flow problem — incorrectness of Sedeño-Noda and González-Martin's algorithm. *Computers & Operations Research*, 33(5):1459–1463, 2006.

[20] H. D. Sherali and A. L. Soyster. Preemptive and nonpreemptive multi-objective programming: Relationships and counterexamples. *Journal of Optimization Theory and Applications*, 39(2):173–186, 1983.

[21] R. E. Steuer. *Multiple criteria optimization: Theory, computation, and application.* John Wiley & Sons, New York, NY, 1986.

[22] G. Ünlü. A linear bilevel programming algorithm based on bicritera programming. *Computers & Operations Research*, 14(2):173–178, 1987.

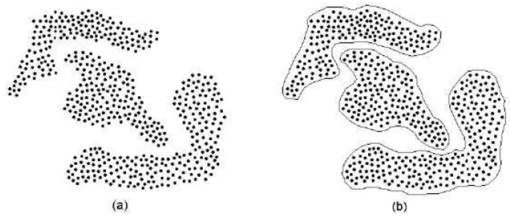
[23] U-P. Wen and S-T. Hsu. A note on a linear bilevel programming algorithm based on bicriteria programming. *Computers & Operations Research*, 16(1):79–83, 89.

Special Forms of Mathematical Programs

This section presents some myths and counterexamples for mathematical programs that do not fit easily into one of the other sections, notably some particular applications.

SF Myth 1. *A good cluster results from maximizing inter-cluster distances and minimizing cluster diameters.*

Counterexample. The following is taken from Climer and Zhang[3].



Using Euclidean distance as the similarity measure for (a), the "intuitive cluster" is (b). The myth fails because many points are closer to a different cluster than their own (so the diameters are not minimized), and the distance between clusters is less than maximal.

SF Myth 2. *A chance-constraint stochastic program has a solution if its certainty equivalent has a solution.*

The model is given by:

$$\max \mathbf{E}[f(x;\theta)] : x \in X, \mathbf{Pr}[g(x;\theta) \leq 0] \geq \alpha,$$

where θ is a vector of uncertain parameters and $\alpha \in (0,1)$. In words, this seeks a policy to maximize the expected value of the objective, subject to it being feasible with probability at least α. The probability and expected value operators are generally taken with respect to θ, and x is a *pure strategy* solution.

Greenberg[9] pointed out that one could allow *mixed-strategy* solutions, in which case the chance constraint could be violated a certain percentage of time. (Also see Blau[2].) The model becomes:

$$\max_H \int_{x \in X} \int_\theta f(x;\theta) dF(\theta) dH(x) : \mathbf{Pr}[g(x;\theta) \leq 0] \geq \alpha,$$

where H is a distribution function on X, subject to choice. The chance constraint is now a joint probability with respect to (x, θ). (Since x is selected before θ is known, the events are independent; hence, we see the product of their distributions.)

Counterexample. Suppose $\int_\theta f(x;\theta)dF(\theta)$ is unbounded over X: $\exists\{x^k\} \subseteq X$ such that $\int_\theta f(x^k;\theta)dF(\theta) \uparrow \infty$. Let x^0 be any *always-feasible solution* — that is, $g(x^0, \theta) \leq 0$ for all θ, and choose H such that $\mathbf{Pr}[x = x^0] = \alpha$, thus satisfying the chance constraint regardless of how we assign the remaining probability, $1 - \alpha$. Let $\{\phi_k\}_{k\geq 1}$ be any series such that $\phi_k \downarrow 0$, $\sum_k \phi_k = 1 - \alpha$, and

$$\sum_k \phi_k \int_\theta f(x^k;\theta)dF(\theta) \to \infty.$$

(Could be $\phi_k = \frac{K}{\int_\theta f(x^k;\theta)dF(\theta)}$ for the appropriate constant, $K > 0$.) Then, the objective is unbounded though the chance constraint is satisfied by the randomization of selecting the one feasible solution α portion of the time.

A realistic application of this model is with a government prohibition constraint. Suppose a chemical plant must limit the emissions of some toxic chemical, but it is not possible to have zero emissions (except by shutting down the plant). The regulation could be stipulated in the form of a chance constraint, and the plant could choose a randomized strategy to improve its expected value, even though that is not what the government had in mind. (See LP Myth 14.)

SF Myth 3. *Someone with constant risk aversion always selects the less risky of two investments.*

This can fail for small wealth, as shown by Lippman, McCall, and Winston[12].

Counterexample. Let s be the wealth of an investor. There are two possible investments with r_i = random return for the i^{th} investment. Letting $V(s)$ denote the maximum expected return for a wealth of s, the expected discounted return model is:

$$V(s) = \max_i\{1 - \mathbf{E}[e^{-\lambda(s+r_i)}] + \beta\,\mathbf{E}[V(s + r_i)]\},$$

where $\beta \in (0, 1)$ and λ is the constant risk aversion factor — that is, independent of the wealth.

Assume wealth and returns are integer-valued, and that investment continues indefinitely unless the investor becomes bankrupt — that is, $s = 0$, in which case $V(0) = 0$ and the process stops. Further, assume each investment produces either one positive return, $\overline{r}_i$, or a loss of 1 with probabilities:

$$\mathbf{Pr}[r_i = \overline{r}_i] = p_i > 0 \text{ and } \mathbf{Pr}[r_i = -1] = 1 - p_i > 0.$$

Set the parameter values as follows:

$$\lambda = 1,\ \beta = 0.9,\ \overline{r}_1 = 2,\ \overline{r}_2 = 1,\ p_1 = 0.5,\ p_2 = 0.6.$$

Then, $\mu_1 = 1.4268$ and $\mu_2 = 1.3080$, so we have

$$\begin{aligned} V(s) \quad = \max\big\{&1 - e^{-s}1.4268 + 0.9\,(0.5V(s+2) + 0.5V(s-1)),\\ &1 - e^{-s}1.3080 + 0.9\,(0.6V(s+1) + 0.4V(s-1))\big\}\end{aligned}$$

for $s = 1, 2, \ldots$

Note that $\mathbf{E}[r_1] = 0.5$, $\mathrm{Var}[r_1] = 5.625$, $\mathbf{E}[r_2] = 0.2$, and $\mathrm{Var}[r_2] = 0.544$. Hence, investment 1 has both the greater expected return and the greater risk. Lippman et al. showed that the investor chooses the risky investment when $s = 1$ — that is,

$$-e^{-1}1.4268 + 0.45V(3) > -e^{-1}1.3080 + 0.54V(2).$$

Thus, the optimality of risk aversion depends upon the level of wealth; it is possible for a risk-averse investor to choose a risky investment for a low level of wealth. Lippman et al. note that V is concave in their counterexample, so the counter-intuitive property is not due to any lack of convexity structure. They also prove that the myth remains if we allow no investment as a decision.

SF Myth 4. *There is always a maximum likelihood estimator.*

Wise and Hall[8] provide a counterexample such that the likelihood function is unbounded.

Counterexample. Let the density function be

$$f(x) = \frac{1-\varepsilon}{\sigma}\, h\left(\frac{x-\mu}{\sigma}\right) + \varepsilon\, h(x-\mu),$$

where μ = mean, σ^2 = variance, $h(\cdot)$ is the standard Gaussian density function, and ε is a (fixed) value in $(0, 1)$. We want to estimate (μ, σ^2).

The likelihood function for independent samples $\{x_1, \ldots, x_n\}$ is

$$L(\mu, \sigma^2) = \prod_{i=1}^{n} \left[\frac{1-\varepsilon}{\sigma}\, h\left(\frac{x_i-\mu}{\sigma}\right) + \varepsilon\, h(x_i-\mu)\right].$$

This is lower-bounded by

$$L(\mu, \sigma^2) = \frac{1-\varepsilon}{\sigma}\, h\left(\frac{x_1-\mu}{\sigma}\right) \prod_{i=2}^{n} \varepsilon\, h(x_i-\mu).$$

Consider $\mu = x_1$, so that $h\left(\frac{x_1-\mu}{\sigma}\right) = h(0) > 0$, independent of σ. Let $K = h(0) \prod_{i=2}^{n} \varepsilon\, h(x_i - x_1) > 0$, so the likelihood function is unbounded:

$$\limsup_{\sigma \to 0} L(x_1, \sigma^2) = \limsup_{\sigma \to 0} \frac{1-\varepsilon}{\sigma} K = \infty.$$

Thus, this distribution has no maximum likelihood estimator.

SF Myth 5. *If demands for substitute products are pooled by a centralized system, the optimal total inventory level cannot increase.*

We have n products with random demands, $D_1, \ldots, D_n$. Optimal inventories for each (ignoring the others) may use a simple model, such as the newsboy problem. Letting F_i denote the cumulative distribution function of D_i, its optimal inventory level is

$$F_i^{-1}(R) = \inf\{d : R \le F_i(d)\},$$

where $R = \frac{c_i^s}{c_i^s + c_i^e}$ (called the *newsboy ratio*), c_i^s = unit cost of shortage, and c_i^e = unit cost of excess.

Here we suppose the products could substitute for each other — that is, if there is a shortage in one, some portion of consumers are willing to buy any other. Then, the inventory model pools the inventories, and optimal levels depend upon various assumptions about the substitution. Letting $\mathcal{F}$ denote the c.d.f. of the sum, $D_1 + \cdots + D_n$, the myth asserts that $\mathcal{F}^{-1}(R) \leq \sum_{i=1}^{n} F_i^{-1}(R)$.

Gerchak and Mossman[8] provide the following:

Counterexample. Let $n = 2$ and let $F_1 = F_2 = F$ be the exponential distribution with common parameter λ. Then, $\mathcal{F}$ is the exponential distribution with parameter 2λ. Thus,

$$F(d) = 1 - e^{-\lambda d} \text{ and } \mathcal{F}(2d) = 1 - e^{-4\lambda d}.$$

Here are some distribution values for $\lambda = 1$:

d	$F(d)$	$\mathcal{F}(d)$
0	0.6321	0.3935
1	0.8647	0.6321
2	0.9502	0.7769
3	0.9807	0.8647

For $c^s = 4c^e$, the newsboy ratio is $R = 0.8$, which implies that the optimal level for each product without any substitution is $F^{-1}(0.8) = 1$, for a total inventory of 2. With full substitution, the optimal (pooled) level is $\mathcal{F}^{-1}(0.8) = 3$. Hence, the pooled inventory is greater than the total of the separate inventories, which violates the myth.

Another counterexample is with the Poisson distribution and $\lambda = 1$. This is perhaps more realistic for a demand distribution, giving some skewness to the right and limiting the demand to integer values. In this case, we have the following distribution values:

d	$F(d)$	$\mathcal{F}(d)$
0	0.3679	0.1353
1	0.7358	0.4060
2	0.9197	0.6767
3	0.9810	0.8571

For $c^s = 2.5c^e$, the newsboy ratio is $R = 0.7143$, which implies that the optimal level for each product without any substitution is $F^{-1}(0.7143) = 1$, for a total inventory of 2. With full substitution, the optimal (pooled) level is $\mathcal{F}^{-1}(0.7143) = 3$. Hence, the pooled inventory is greater than the total of the separate inventories, which violates the myth.

In both cases we choose R to satisfy $\underline{R} < R < \overline{R}$. Equivalently, we have a range on the cost ratio:

$$\frac{\underline{R}}{1 - \underline{R}} < \frac{c^s}{c^e} < \frac{\overline{R}}{1 - \overline{R}}.$$

For the two cases, these ratios are:

$$\text{Exponential:} \quad 3.482 = \frac{0.7769}{0.2231} < \frac{c^s}{c^e} < \frac{0.8647}{0.1353} = 6.391 \qquad \text{(I chose 4.)}$$

$$\text{Poisson:} \quad 2.093 = \frac{0.6767}{0.3233} < \frac{c^s}{c^e} < \frac{0.7358}{0.2642} = 5.998. \qquad \text{(I chose 2.5.)}$$

Yang and Schrage[19] establish a sufficient condition for the myth to be violated:

Suppose $D_1, \ldots, D_n$ are i.i.d. with common cumulative distribution function, F, and costs, c^s, c^e. Then, $\mathcal{F}^{-1}(R) > nF^{-1}(R)$ if there exists d such that

$$F(d) > \mathcal{F}(nd) \text{ and } F(d) \geq R \geq 0.5.$$

One can verify that the counterexamples satisfy this condition for the indicated range of cost ratios.

Yang and Schrage provide more analysis of the full substitution model, identifying skewness of the probability distribution as the key property that creates a counterexample. They extend the model to include *partial substitution*, which limits the percentages of substitution for each product.

SF Myth 6. *If a team has the greatest win percentage before and after some specified date, it also has the greatest overall win percentage.*

This is an instance of *Simpson's Paradox*, for which there is a vast literature. Cochran[4] used the baseball players' strike of 1981 as the demarcating date, and he used Simpson's Paradox to teach some elements of integer programming modeling, particularly the formation of objective functions.

Counterexample. Consider the following win-loss records:

	Pre-Strike				Post-Strike				Total			
Team	w	ℓ	$w+\ell$	$\frac{w}{w+\ell}$	w	ℓ	$w+\ell$	$\frac{w}{w+\ell}$	w	ℓ	$w+\ell$	$\frac{w}{w+\ell}$
A	17	18	35	**0.4857**	18	15	33	**0.5455**	35	33	68	0.5147
B	15	16	31	0.4839	19	16	35	0.5429	34	32	66	**0.5152**

(Teams A and B play other teams too.)

Team A stands above Team B in both the pre-strike and post-strike games, but Team B stands above Team A overall.

See the Wikipedia entry at http://en.wikipedia.org/wiki/Simpson's_paradox for more examples and further explanation.

SF Myth 7. *In revenue management, it is always better to re-solve dynamic allocations than use the planned allocations.*

The key to this myth is the definitions of *planned allocation* and *reallocation*. Cooper[5] provides the following:

Counterexample. Consider two fare classes in a flight with one leg and two seats available. Class 1 pays \$1,000 and class 2 pays \$200. The LP to plan allocations during the planning horizon, $[0, T]$ time periods is:

$$\max\ 10x_1 + 2x_2 : x_1 + x_2 \leq 2,\ 0 \leq x \leq T.$$

Assume demands, $D_i(t)$, for classes $i = 1, 2$, are independent Poisson processes, each with rate 1. For $T = 2$, an optimal solution to the above LP is $x^* = (2, 0)$. This is the planned solution: reserve 2 seats for class 1 and none for class 2.

The issue is whether to reallocate after one period, having had demand $d = (d_1, d_2)$:

$$\max\ 10y_1 + 2y_2 : y_1 + y_2 \le 2 - \big(\min\{d_1, x_1^*\} + \min\{d_2, x_2^*\}\big),\ 0 \le y \le \mathbf{E}[D(2)],$$

where $\mathbf{E}[D(2)]$ is the expected demand in period 2 for each class. Here are optimal reallocations for each possible value of d_1 (the value of d_2 is irrelevant since $x_2^* = 0$):

d_1	y_1^*	y_2^*	$\mathbf{Pr}(D_1(1) = d_1)$
0	1	1	0.3679
1	1	0	0.3679
≥ 2	0	0	0.2642

The expected revenue for this reallocation policy satisfies:

$$\begin{aligned}
&\mathbf{E}[10\,\min\{D_2(1), 1\} + 2\,\min\{D_2(2) - D_1(1), 1\} \,|\, D_1(1) = 0] \\
&\qquad = 10\,\mathbf{E}[\min\{X, 1\}] + 2\,\mathbf{E}[\min\{X, 1\}] = 7.59 \\
&< \mathbf{E}[10\,\min\{D_2(1), 2\} \,|\, D_1(1) = 0] = 10\,\mathbf{E}[\min\{X, 2\}] = 8.96,
\end{aligned}$$

where X is a random variable with Poisson distribution having rate 1.

Therefore, the expected remaining revenue is less by reallocation than by staying with the planned allocations, given the demand in the first period satisfies $D_1(1) = 0$.

SF Myth 8. **new** *Among no-memory rules to order list items, the move-to-front rule minimizes the average cost.*

The cost to access an item is its position in the list. For example, if the order does not change and P_i is the probability that item i is requested, the average cost for accessing n items is $\sum_{i=1}^{n} i\,P_i$. We assume that the probabilities are not known a priori. A *no-memory rule* is one that does not use any information about the history of requests. (This includes relative frequencies, so their probabilities cannot be estimated.) Rivest[13] introduced the *move-to-front rule*: replace the requested item with the one at the front of the line. For example, if the items are in their natural order, $1, 2, \ldots, n$, and there is a request for item m, the new order is $m, 1, 2 \ldots, m-1, m+1, \ldots, n$. The myth asserts that this has the least average cost among all no-memory rules. Rivest conjectured the myth using examples for some intuition.

Counterexample. Anderson, Nash, and Weber[1] provide the following. The request probabilities for six items are $P = (0.851, 0.146, 0.001, 0.001, 0.001)$. Consider the no-memory rule defined by six permutations, where Π_{ij} is the position of the j^{th} item after receiving a request for the i^{th} item.

$$\Pi = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 3 & 1 & 4 & 5 & 6 \\ 1 & 2 & 4 & 3 & 5 & 6 \\ 1 & 2 & 3 & 5 & 4 & 6 \\ 3 & 4 & 1 & 2 & 6 & 5 \end{bmatrix}.$$

For example, if the request is for item 1 or 2, the order does not change. If the request is for item 3, the new order is 231456.

The average cost for the move-to-front rule is approximately 1.26, whereas the average cost of this transposition rule is approximately 1.22. (Both calculations are done by forming the associated Markov chain — see Rivest[13].) See Sleator and Tarjan[16] for insight as to why the move-to-front rule is "approximately optimal" in practice.

SF Background — Data Envelopment Analysis

We are given features of each *Decision-Making Unit* (DMU), which we partition into inputs, denoted I, and outputs, denoted O. The *production possibility set*, P, is the convex hull of $\{(I_k, O_k)\}$, and we wish to evaluate the k^{th} DMU using the features of the others. There are several ways to approach this.

Assume, for our limited purposes here, that (I_k, O_k) is in the interior of the convex hull. Let c denote the cost vector associated with the DMUs, and our goal is to compare c_k with the other DMUs. To do so, we solve the *cost-comparison* LP:

$$\min \sum_{i \neq k} c_i x_i : \; x \in F_k(I_k, O_k), \tag{SF.11}$$

where

$$F_k(I_k, O_k) = \{x : \; x \geq 0, \; \textstyle\sum_{i \neq k} x_i = 1, \; \sum_{i \neq k} I_i x_i \leq I_k, \; \sum_{i \neq k} O_i x_i \geq O_k\}.$$

In words, we find a point in the production possibility set such that each of its inputs does not exceed the input for the k^{th} DMU, and each of its outputs is at least as great as that of the k^{th} DMU. We find the least costly point and compare that with c_k to evaluate how well the k^{th} DMU performs.

Two other LPs are used to evaluate a DMU:

$$\text{Input-oriented:} \quad \min \; \theta : \; x \in C_k(\theta I_k, O_k) \tag{SF.12}$$

$$\text{Output-oriented:} \quad \max \; \theta : \; x \in C_k(I_k, \theta O_k), \tag{SF.13}$$

where C_k is the set of conical combinations of DMU features, except the k^{th} DMU:

$$C_k(I_k, O_k) = \{x : \; x \geq 0, \; \textstyle\sum_{i \neq k} I_i x_i \leq I_k, \; \sum_{i \neq k} O_i x_i \geq O_k\}.$$

The Input-oriented LP asks for the minimum proportionate input for which the output could be satisfied. The Output-oriented LP asks for the maximum proportionate output within the input limit. (See Cooper, Gu, and Li[6] for alternative DEA models.)

Consider the Input-oriented model with $O_k > 0$ (so $x = 0$ is not feasible). Let θ^* be the minimum proportionate change with optimal weight set X^* for (SF.12). The *return to scale* exhibited by the k^{th} DMU is classified by the total of the solution weights in X^*:

Constant return to scale (CRS): $\sum_i x_i^* = 1$ for some $x^* \in X^*$;

Decreasing return to scale (DRS): $\sum_i x_i^* > 1$ for all $x^* \in X^*$;

Increasing return to scale (IRS): $\sum_i x_i^* < 1$ for all $x^* \in X^*$.

For example, an increasing return to scale means that the same output requirements can be satisfied with proportionally less input from the k^{th} DMU. The maximum total weight, among the optima that yield θ^*, is given by:

$$\sigma^{-1} = \max_{x \in X^*} \left\{ \sum_{i \neq k} x_k \right\}. \tag{SF.14}$$

(See Seiford and Zhu[14] for details and additional models.)

DEA Pitfalls and Protocols

Dyson et al.[7] describe pitfalls to avoid in using DEA. Here are some examples:

Homogeneity. Use clustering, if necessary, to have the DMUs comparable. For example, do not compare a science department with a language department.

Correlation. Reduce features to be as uncorrelated as possible. For example, if staff size is one input, total staff budget is correlated, so they should not be used as though they are two independent inputs.

Feature measurement. The inputs and outputs may be subject to measurement errors, and some may be qualitative. Several approaches have been considered (cited by Dyson et al.).

Linearity. The DEA models assume that we can represent a feature by taking a (non-negative) linear combination of the features of the DMUs.

Weight restrictions. There may be restrictions, such as simple bounds, $\underline{x} \leq x \leq \overline{x}$. These may depend upon the DMUs in the database. Removing or adding a DMU could change the weights, the manner of which needs explanation.

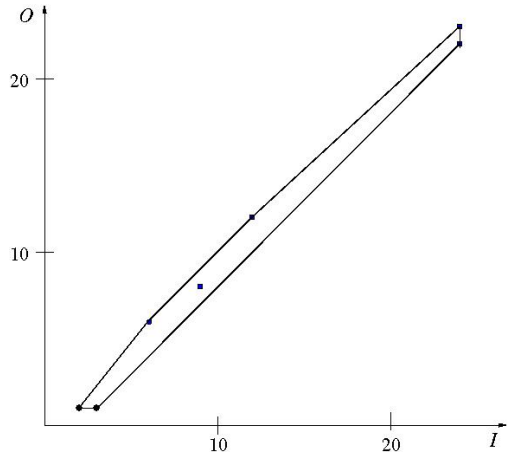
They provide further discussions and numerical examples.

SF Myth 9. new *A DMU that exhibits increasing return to scale continues to do so up to a proportional increase of all outputs equal to* $\alpha \in [1, \sigma)$.

This is one of the results by Seiford and Zhu[14] (also see [15] for further discussions and placing such errors in context).

Counterexample. Jahanshahloo, Lofti, and Zohrehbandian[11] provide the following:

	DMU						
	1	2	3	4	5	6	7
Input	2	3	6	12	24	24	9
Output	1	1	6	12	22	23	8



Let $k = 2$, so the Input-oriented LP is:

$$\begin{aligned} &\min\ \theta : x \geq 0 \\ &2x_1 + 6x_3 + 12x_4 + 24x_5 + 24x_6 + 9x_7 \leq 3\,\theta \\ &x_1 + 6x_3 + 12x_4 + 22x_5 + 23x_6 + 8x_7 \geq 1. \end{aligned}$$

An optimal solution is $\theta^* = \frac{1}{3}$ with $x^* = (0, \cdot, 0.0911, 0.0378, 0, 0, 0)$. This means that DMU_2 can produce its output with a combination of inputs from DMU_3 and DMU_4 that use only $\frac{1}{3}$ of the DMU_2 input. That combination sums to less than one, so DMU_2 exhibits an increasing return to scale.

Using (SF.14)[14], $\sigma = 6$, so the myth asserts that the IRS remains in effect if the output is increased to αO for $\alpha \in [1, \sigma)$. However, for $\alpha = 5 < 6$, we get $(I_2, O_2) = (3, 5) \notin P$. Thus, $(3, 5)$ does not exhibit IRS because it is not in the production possibility set.

SF Myth 10. new *When using LP for DEA, it does not matter if you obtain a basic or interior optimum.*

Given a solution, x^*, to (SF.11) the associated *peer group* is $\sigma(x^*) = \{i : x_i > 0\}$. If there are alternative optima, different peer groups can be generated, depending upon which solution is obtained. That raises the issue stated in the myth, given by Greenberg[10].

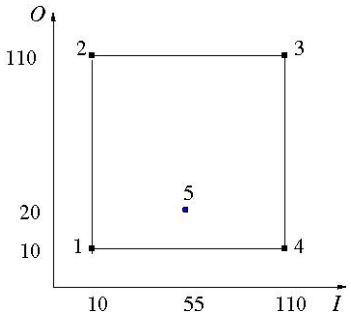
Any interior solution gives the union of all peer groups. Let x^0 denote an interior solution, and let $\{x^\ell\}_\ell = 1^L$ denote the basic solutions. Then,

$$\sigma(x^0) = \bigcup_{\ell=1}^{L} \sigma(x^\ell).$$

Because of the potential sensitivity of the evaluation process to the choice of peer group, the interior solution better serves the interests of full disclosure. If only one peer group is used in the evaluation, one may question whether another peer group should have been used.

Counterexample.

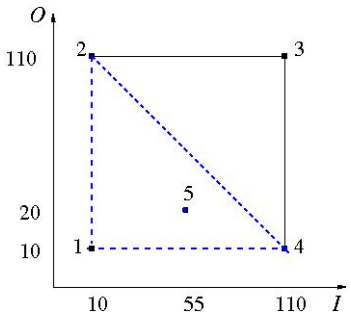
Consider just one input and one output, so the feature space is in the plane. There are five DMUs in the database, and we are evaluating number 5, illustrated on the right.



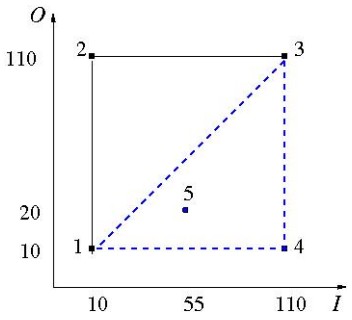
Let the cost vector be $c = (8, 6, 5, 7)$, so the LP is:

$$\begin{array}{lrcrcrcrcr} \min & 8x_1 & + & 6x_2 & + & 5x_3 & + & 7x_4 & : x \geq 0 \\ & x_1 & + & x_2 & + & x_3 & + & x_4 & = & 1 \\ & 10x_1 & + & 10x_2 & + & 110x_3 & + & 110x_4 & \leq & 55 \\ & 10x_1 & + & 110x_2 & + & 110x_3 & + & 10x_4 & \geq & 20 \end{array}$$

There are two optimal basic solutions: $x^1 = (0.45, 0.10, 0, 0.45)$ and $x^2 = (0.55, 0, 0.10, 0.35)$, with associated peer groups $\{1, 2, 4\}$ and $\{1, 3, 4\}$.



Basic Solution x^1



Basic Solution x^2

An interior solution is $x^0 = (0.5, .05, 0.05, 0.4)$ with peer group $\sigma(x^0) = \{1, 2, 3, 4\}$.

One may note, from the example, that the interior solution by itself does not provide all of the useful information. In particular, DMUs 1 and 4 must be in the peer group, whereas the third member could be either DMU 2 or 3 to form a basic feasible solution. Thus, the essential inclusion of DMUs 1 and 4 is lost if only one solution is obtained, regardless of whether it is basic or interior.

The bottom line is that the interior solution is preferred. Its peer group better fits the need for disclosure, and with a modest amount of additional computation, each *essential* member of the peer group can be identified (that is, fix $x_i = 0$ for each $i \in \sigma(x^0)$; i is essential if the LP becomes infeasible).

SF Myth 11. new *When entering a parallel queueing system, it is optimal to join the shortest queue.*

Assume the arrivals into the system follow a Poisson process, with rate λ. Each new arrival knows the queue lengths and must decide which queue to join to minimize expected time in the system.

One reason this myth is not true is that "shortest" need not be the least wait time. For example, in a supermarket checkout line people have different amounts of groceries. Let us assume, however, that our queueing system serves indistinguishable customers. (Each customer may be in one of several classes, requiring different service times, but the new arrival cannot determine to which class each customer belongs.)

Counterexample. Whitt[17] provides the following: Let there be two queues with independent service times. Let the common service-time distribution be given by the mass function:

$$\mathbf{Pr}(t = \tau) = 1 - \varepsilon \quad \text{and} \quad \mathbf{Pr}(t = 2) = \varepsilon,$$

where $0 \leq \tau \ll \varepsilon \ll 1$. We may consider this to be a 2-class population, but unlike the supermarket example, an arrival cannot determine the other customers' class. An example is a bank (with separate lines) such that a customer may have a very quick transaction, like a deposit, or may require a lot of time, like complicated transfers.

Define the system state (s_1, s_2) = queue sizes. The shortest-queue rule is optimal for states: (1) $s_1 = 0$ or $s_2 = 0$, and (2) $|s_1 - s_2| \leq 1$. However, if $s_1, s_2 > 0$ and $|s_1 - s_2| \geq 2$, Whitt proves that it is optimal to join the *longer* queue for τ, ε sufficiently small.

Special Forms References

[1] E. J. Anderson, P. Nash, and R. R. Weber. A counterexample to a conjecture on optimal list ordering. *Journal of Applied Probability*, 19(3):730–732, 1982.

[2] R. A. Blau. Stochastic programming and decision analysis: An apparent dilemma. *Management Science*, 21(3):271–276, 1974.

[3] S. Climer and W. Zhang. Rearrangement clustering: Pitfalls, remedies, and applications. *Journal of Machine Learning Research*, 7(6):919–943, 2006.

[4] J. J. Cochran. Bowie Kuhn's worst nightmare. *INFORMS Transactions on Education*, 5(1):18–36, 2004. Available at http://archive.ite.journal.informs.org/Vol5No1/Cochran/.

[5] W. L. Cooper. Asymptotic behavior of an allocation policy for revenue management. *Operations Research*, 50(4):720–727, 2002.

[6] W. W. Cooper, B. Gu, and S. Li. Comparisons and evaluations of alternative approaches to the treatment of congestion in DEA. *European Journal of Operational Research*, 132(1):62–74, 2001.

[7] R. G. Dyson, R. Allen, A. S. Camanho, V. V. Podinovski, C. S. Sarrico, and E. A. Shale. Pitfalls and protocols in DEA. *European Journal of Operational Research*, 132(2):245–259, 2001.

[8] Y. Gerchak and D. Mossman. On the effect of demand randomness on inventories and costs. *Operations Research*, 40(4):804–807, 1992.

[9] H. J. Greenberg. On mixed-strategy solutions to chance-constraint mathematical programs. Technical report, Southern Methodist University, Dallas, TX, 1968.

[10] H. J. Greenberg. The use of the optimal partition in a linear programming solution for postoptimal analysis. *Operations Research Letters*, 15:179–185, 1994.

[11] G. R. Jahanshahloo, F. H. Lofti, and M. Zohrehbandian. Notes on sensitivity and stability of the classifications of returns to scale in data envelopment analysis. *Journal of Productivity Analysis*, 23(3):309–313, 2005.

[12] S. A. Lippman, J. J. McCall, and W. L. Winston. Constant absolute risk aversion, bankruptcy, and wealth-dependent decisions. *Journal of Business*, 53(3):285–296, 1980.

[13] R. Rivest. On self-organizing sequential search heuristics. *Communications of the ACM*, 19(2):63–67, 1976.

[14] L. M. Seiford and J. Zhu. Sensitivity and stability of the classifications of returns to scale in data envelopment analysis. *Journal of Productivity Analysis*, 12(1):55–75, 1999.

[15] L. M. Seiford and J. Zhu. Notes on sensitivity and stability of the classifications of returns to scale in data envelopment analysis: A comment. *Journal of Productivity Analysis*, 23:315–316, 2005.

[16] D. D. Sleator and R. E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28(2):202–208, 1985.

[17] W. Whitt. Deciding which queue to join: Some counterexamples. *Operations Research*, 34(1):55–62, 1986.

[18] G. L. Wise and E. B. Hall. *Counterexamples in Probability and Real Analysis*. Oxford University Press, New York, NY, 1993.

[19] H. Yang and L. Schrage. Conditions that cause risk pooling to increase inventory. *European Journal of Operational Research*, 192(3):837–851, 2009.

Name Index

Allen, 117
Anderson, 57, 115
Armijou, 90
Arsham, vi, 3, 19
Ash, 78
Asher, 85

Babayev, 27
Balinski, vi
Bard, 107
Beale, 8
Bellmore, 22, 23
Ben-Ayed, 106, 108
Benczúr, vi, 25
Benson, vi, 100
Bertsekas, 16
Biegler, vi, 91
Blackwell, 62
Blair, vi, 5, 27, 106, 108
Bloom, 77
Bonnans, 91
Braess, 26
Brucker, 39
Bruns, vi, 35
Byrd, 36

Camanho, 117
Candler, 106, 107
Charnes, 18–20
Chinneck, 95
Cho, 99
Climer, vi, 110
Cochran, 114
Cooper, vi, 20, 114, 116

Dĭnekoa, 19
Dantzig, 38, 80, 81
Das, 98
Davila, vi, 37
de Klerk, 94
Denardo, 72
Dennis, vi, 98
Donovan, 85
Duffuaa, 19
Dussault, 83
Dyson, 117

Ecker, 99
Ehrgott, vi, 102, 103
Evans, 79

Fan, 84
Fenchel, 82
Fisher, 64
Fletcher, 91
Fliege, 105
Flynn, vi, 61, 67–70
Folkman, 80, 81
Ford, 22
Fournier, 83
Fox, vi, 42, 48
Fujii, 46
Fulkerson, 22

Gandibleux, 103
Garey, 40
Gass, vi, 8, 13
Gassner, 9
Gerchak, 113
Glicksburg, 84
Glover, vi, 19, 20, 27, 33, 34, 48
Goldman, 7, 36
Gomory, 38
Gorski, vi, 102
Gould, 79, 93
Graham, 40, 44, 45
Greenberg, 3, 17, 22, 23, 28, 42, 80, 84, 86, 118
Grippo, 90
Grossmann, 90
Gu, 116
Gubaladze, 35

Hajek, 49
Halická, 94
Hall, 112
Hao, 48
Haurie, 106
Hegner, 99
Heller, 36
Henk, 35
Hinderer, 70
Hoffman, vi, 8, 38, 84

Hordijk, 71
Horng, 37
Hsu, 106–108
Hu, vi, 84

Isermann, 99

Jahanshahloo, 117
Jarvis, 22, 23
Jeroslow, 27, 34, 36, 38
Johnson, 40

Karamardian, 84
Kasami, 46
Kaufman, 46
Keeney, 101
Kim, 99
Klamroth, vi, 102
Klee, 26
Kleinman, 7
Klingman, 18, 20, 33
Klinz, 19
Klotz, vi, 36–38
Kornbluth, 103, 104
Kuhn, 19, 76

Lai, 43
Lampariello, 90
Lee, 99
Lenstra, vi, 39, 41, 42
Leyffer, vi, 91
Li, 116
Lippman, 61, 111
Lofti, 117
Lowe, 36, 99
Lu, 56
Lucidi, 90

MacMillan, 90
Maitra, 63
Maratos, 91
Marcotte, 106–108
Martin, 35
Martos, 84
McCall, 111
McKinnon, vi, 88
Megiddo, vi
Meyer, 36
Miller, 85
Minty, 26
Mitten, 53
Moreland, 85
Mossman, 113

Nagurney, 26
Nash, 115
Ninomiya, 46

O'Sullivan, 72
Oblak, 19
Orloff, 41

Panier, 91
Phillips, 33
Pierskalla, 80, 84
Podinovski, 117
Porteus, 54
Powell, 86
Przybylski, 103

Qi, 76

Rinnooy Kan, 39, 41, 42
Rivest, 115, 116
Rizzi, vi, 25
Roos, 94
Ross, vi, 64–66
Rothschild, 23
Roughgarden, vi, 26
Rubin, 17, 34
Ruzika, 102
Ryan, 19

Sørevik, 20
Saaty, 7
Sahni, 43
Sarrico, 117
Sasaki, vi, 49
Savard, 106, 108
Schrage, 42, 114
Sebö, vi
Seiford, 117
Sexton, 78
Shale, 117
Shapiro, 80, 81
Shearer, 41
Sherali, vi, 105
Sierksma, 3, 7
Sleator, vi, 116
Smith, 39
Sommer, 33, 34
Song, 58, 59

Soyster, 105
Spieksma, 39
Spielman, 27
Steinberg, 26
Steuer, vi, 98, 100, 101, 103, 104
Stidham, 59, 60
Storøy, 20
Szwarc, 18

Tarjan, 116
Teng, 27
Thisse, 99
Tijms, 71
Tijssen, 3
Tits, 91
Todd, 28
Toint, vi, 91, 93
Tomlin, 3
Toth, 52
Tovey, vi, 45
Tseng, 21
Tucker, 76

Ünlü, 108

Vanden Berghen, vi, 91
Vicente, 105
Vinjamuri, 8

Wächter, vi, 91, 92
Wagner, 17
Wang, 37
Ward, 99
Weber, vi, 59, 60, 115
Weismantel, 35
Welch, 3
Wen, 106–108
Wendell, 99
Whinston, 23
White, 106
Whitt, 120
Williams, vi, 4
Winston, 111
Wise, 112
Woeginger, vi, 19, 39, 47
Wolfe, 82, 83
Wolkowicz, vi

Xu, 56

Yang, 114

Zangwill, 26, 54, 55
Zhang, 110
Zhao, 21
Zhou, 91
Zhu, 117
Zohrehbandian, vi, 117
Zullo, 22
Zwart, 87

Subject Index

active constraint, 4
activity-on-arc network, 21
alternative systems, 84
assignment problem, 9, 16, 19

barrier function, 91–93
base-stock policy, 58, 59
best-gain criterion, 3
better is worse, 39, 40, 43–45, 112
bilevel program, 97, 105–108
bin-packing, 40
Braess paradox, 26
branch-and-bound, 38, 42, 43
breakpoint, 15

Carathéodory's theorem, 35
Chinese postman problem, 41
clustering, 110
compatible basis, 17
complexity, 27
constraint, 2
  binding, 4
  equality vs. inequality, 4, 19
  implied equality, 3
  qualification, 75
  redundant, 3
convex function, 81
convex set, 81
cut-set, 24
cut-tree, 24
cyclic descent, 86
cycling, 8, 9

Dantzig cut, 38
Data Envelopment Analysis (DEA), 116–118
deletion filter, 95
depth, 47
disconnecting set, 22
dominance, 5
DP objective
  average-overtaking, 61, 67, 69, 70
  B-opt, 61–63, 65, 67–70
  ε-optimal, 61, 62, 65, 66
  1-optimal, 61, 68, 72
  average, 60, 67
  average return, 66
  Bopt, 63
  discounted, 60, 65
  liminf average optimal, 69
  limsup average optimal, 69
  nearly optimal, 61, 62
dual price, 13, 17
dynamic lot size, 54–59

economic order quantity (EOQ), 54
extreme point, 3

feasible directions, 90
filter method, 92
filtering, 91
First-Fit packing, 40
fixed charge, 36
float time, 21
forward transformation (FTRAN), 29
fourth-order conditions, 76
fractional program, 103
free variable, 4

General Routing Problem, 41
Gomory-Hu tree, 24
greedy algorithm, 16
Grid Search Algorithm, 106

Hirsch conjecture, 27
Hungarian method, 19

implied equality, 3
integer hull, 34
interior solution, 5, 118
interior-point method, 27
inventory, 54–59, 112
irreducible infeasible subsystem (IIS), 95

Klee-Minty polytope, 26
Kuhn-Tucker conditions, 75

Lagrange multiplier, 75, 76
lexico-optimum, 105
lexio-graphically order, 105
list ordering, 115
LP relaxation, 34

makespan, 39, 43–45

Maratos effect, 91, 92
Markov decision process, 61
max-flow, 22, 23, 26
maximum likelihood, 112
metaheuristic, 46, 48
min-cut, 22–24
MIP-representable, 36
more for less, 18–20, 26
move-to-front rule, 115
myopic solution, 16

Nelder-Mead method, 88, 91
network design, 106
newsboy problem, 112
Newton's method, 85
nonmonotone method, 90
notation, 1

order of convergence, 86

parallel computation, 43–45
parametric, 37
Parametric Complementary Pivot algorithm, 108
Pareto frontier, 97, 98
Pareto-optimum, 97, 99
Pareto-optimum graph, 102
partitioning, 89
Principle of Optimality, 53, 54
project scheduling, 21
pseudogap, 87

quasiconvex, 78, 84
queueing control, 59, 119

ratio rule, 39
revenue management, 114
risk analysis, 111
Ritter's method, 87
Rosen's decomposition method, 89
rounding, 33
routing, 41

scheduling, 39, 43–46
second-order correction, 91
semi-continuous, 80, 82
semidefinite programming, 94
sequential decision process, 52
Sequential Quadratic Programming (SQP), 91
shortest path, 103
simplex method, 5–7
Simpson's Paradox, 114
simulated annealing, 48
software anomaly, 6, 28, 36–38, 95
solution
    unbounded, 36, 37
    unique, 17
spanning tree, 102, 103
sports, 114
standard form, 7
standard form of LP, 2–4
standard simplex method, 3, 26, 27
stationary, 63
steepest ascent, 82, 83, 90
stepping stone method, 20
strict interior, 78

tax, 13
Taylor's series, 75
tolerance, 28, 85
transportation problem, 17
transposition rule, 116
transposition theorem, 84
travelling salesman problem (TSP), 41
triangle inequality, 41
truncated gradient, 83
trust region, 92
Tui's method, 87

weighted objective, 99

zig-zagging, 83, 90

If you know of some erroneous result, paradox, fallacy, anomaly, pitfall, or some counter-intuitive result in mathematical programming, please let me know.

Harvey J. Greenberg <hjgreenberg@gmail.com>

This is an ongoing project.