Introduction

The article "Roadmap for Optimization" (*WIREs: Computational Statistics*, Said and Wegman, 2009) purports "to provide in broad brush strokes a perspective on the area in order to orient the reader to more detailed treatments of specific subdisciplines of optimization throughout WIREs: Computational Statistics". Large sections of the article appear to have been copied from earlier-dated Wikipedia pages on the same topics. The writing is poor, including grammatical errors and even an occasional math error (not present in the antecedent Wikipedia entries). The references and further reading sections do not provide useful guidance to the reader seeking further understanding.

The article's poor writing and scattershot coverage disqualifies it from inclusion in a publication striving to provide authoritative summaries of topics in an interdisciplinary field. The apparent copying from other sources, if verified by other readers, disqualifies it from inclusion in the scholarly record. **The paper should be formally retracted and the editorial and reviewing processes that allowed it to be published in the first place should be changed.**

This note begins with a detailed analysis of the paper's section, "Linear Programming", showing the close relationship between that section and two Wikipedia pages published five months earlier. Having demonstrated the relationship in detail for one section, it then provides more concise overviews of several other sections where large portions share text with Wikipedia pages. Two entire pages (pp. 6–7) feature these strong similarities, with smaller regions of similarity on pages 4 and 8. This search has not been exhaustive; there may be other sources that I did not uncover.

For consistency, I have chosen to compare the Said and Wegman article with the versions of the corresponding Wikipedia pages for early February 2009. Many of the similarities to Wikipedia remain at time of writing (May 2011), although some Wikipedia entries have changed dramatically since then. For those pages that have changed, the February versions are much closer to the Said and Wegman text than the current pages.

Linear Programming

This section compares the contents of the "Linear Programming" section of "Roadmap for Optimization" (Said and Wegman, 2009), published on-line July 13, 2009, with the Wikipedia entries for "Linear Programming" and "Simplex Algorithm" from Feb 3 and 6, 2009.

The left column shows the text of the Said and Wegman section as written. The right column shows corresponding excerpts from the Wikipedia entries that have strikingly similar text. The order of the Wikipedia text on the right does not correspond to its original order, instead matching the similar text in Said and Wegman.

For each comparison, text is marked as either identical, different in only minor ways, wrong in Said and Wegman, not copied from Wikipedia into Said and Wegman, or original to Said and Wegman, as highlighted below. All text from the Said and Wegman section has strong similarity to counterpart text in Wikipedia, except for one paragraph describing feasibility and duality in linear programming.

Several aspects of the Said and Wegman text indicate that it was copied from the Wikipedia text, rather than the reverse:

1. The Wikipedia entries were published five months earlier.
2. Several key concepts are defined in the Wikipedia via hyperlinks on the terms, but never defined in Said and Wegman*: polytope, affine function, linear equality*, and *linear inequality constraints*. It seems odd that a "roadmap" article would not define a section's central terms.
3. The last paragraph of Said and Wegman's section, "Simplex Algorithm", incorrectly states that the number of vertices of a *d*-dimensional cube is 2*d*. The correct value is 2^d. This is exactly the kind of mistake that would arise from copying text from an HTML page and pasting it into a text editor such as Microsoft Word.

Identical text Minor difference Moved in S&W Wrong in S&W ~~Not copied from Wiki~~

Linear Programming: Said & Wegman vs Wikipedia page for Linear Programming, Feb. 3, 2009 (archived at http://en.wikipedia.org/w/index.php?title=Linear_programming&oldid=268212519). Said & Wegman was published on-line 13 July 2009, five months after this version of the Wikipedia page.

Said& Wegman, pp. 5–6:	Wikipedia, Linear Programming:
Linear programming is a technique for optimization of a linear objective function subject to linear equality and linear inequality constraints.	**From introductory section:** In mathematics, **linear programming** ~~(LP)~~ is a technique for optimization of a linear objective function, subject to linear equality and linear inequality constraints. Informally, linear programming determines the way to achieve the best outcome (such as maximum profit or lowest cost) in a given mathematical model and given some list of requirements represented as linear equations.
Given a polytope and a real-valued affine function	More formally, given a polytope ~~(for example, a polygon or a polyhedron)~~, and a real-valued affine function $f(x_1, x_2, \ldots, x_n) = c_1x_1 + c_2x_2 + \cdots$
on the polytope, a linear programming solution will be a point in the polytope where that function will have the smallest (or largest) value. Suppose φ(x) = c1x1 + c2x2 +· · ·+ cdxd = cTx where vectors c and x are both column vectors. In the standard maximum problem,	~~defined~~ on this polytope, a linear programming method will find a point in the polytope where this function has the smallest (or largest) value. Such points may not exist, but if they do, searching through the polytope vertices is guaranteed to find at least one of them. Linear programs are problems that can be expressed in canonical form: Maximize $\mathbf{c}^T\mathbf{x}$ Subject to $A\mathbf{x} \le \mathbf{b}$.
We wish to maximize cTx subject to some constraint of the form Ax ≤ b. x represents the vector of variables to be determined, c and b are vectors of known coefficients, and A is a m × d matrix of known coefficients. In this case, cTx is the objective function and Ax ≤ b and x ≥ 0 are the constraints that specify the convex polytope. The standard minimum problem is to find y = (y1, . . . , yd) to minimize bTy subject to Ay ≥ c.	$\mathbf{x}$ represents the vector of variables (to be determined), while $\mathbf{c}$ and $\mathbf{b}$ are vectors of (known) coefficients and A is a (known) matrix of coefficients. The expression to be maximized or minimized is called the objective function ($\mathbf{c}^T\mathbf{x}$ ~~in this case). The equations~~ $A\mathbf{x} \le \mathbf{b}$ are the constraints which specify a convex polyhedron over which the objective function is to be optimized.

Identical text Minor difference Moved in S&W Wrong in S&W ~~Not copied from Wiki~~

Said& Wegman (p. 6)

The following paragraph appears to be original:
A vector x in the standard maximum problem or y in the standard minimum problem is said to be feasible if it satisfies the corresponding constraints. The set of feasible vectors is the constraint set. A linear programming problem is said to be feasible if the constraint set is not empty; otherwise it is said to be infeasible. A feasible maximum (minimum) problem is said to be unbounded if the objective function can assume arbitrarily large positive (negative) values at feasible vectors. If a problem is not unbounded it is said to be bounded. A minimum problem can be changed to a maximum problem by multiplying the objective function by −1. Similarly constraints of the form ∑dj =1 aijxj ≥ bi can be changed to the form ∑dj =1(−aij)xj ≤ −bi. A variable xj may not be restricted to be non-negative. In this case we can replace xj by the difference of two variables uj − vj where both are restricted to be non-negative. Thus corresponding to every maximum problem, called the primal problem, there is a corresponding minimum problem which is said to be the dual problem.

Said& Wegman, p. 6

The dual of a dual linear problem is the original primal problem. Also every feasible solution of a linear problem gives a bound on the optimal value of objective function of its dual. The weak duality theorem states that the objective function value of the dual at any feasible solution is always greater than or equal to the objective function value of the primal at any feasible solution. The strong duality theorem says that if the primal has an optimal solution, x∗, then the dual also has an optimal solution, y∗, such that cTx∗ = bTy∗. Finally,

if the primal is unbounded, then the dual is infeasible and if the dual is unbounded, then the primal is infeasible. It is possible for both the primal and the dual to be infeasible.

Wikipedia, Linear Programming

From "Duality":
One is the fact that the dual of a dual linear program is the original primal linear program. Additionally, every feasible solution for a linear program gives a bound on the optimal value of the objective function of its dual. The weak duality theorem states that the objective function value of the dual at any feasible solution is always greater than or equal to the objective function value of the primal at any feasible solution. The strong duality theorem states that if the primal has an optimal solution, x*, then the dual also has an optimal solution, y*, such that $c^Tx^*=b^Ty^*$.
A linear program can also be unbounded or infeasible. Duality theory tells us that if the primal is unbounded then the dual is infeasible ~~by the weak duality theorem~~. Likewise, if the dual is unbounded, then the primal must be infeasible. However, it is possible for both the dual and the primal to be infeasible (See also Farkas' lemma).

Identical text Minor difference Moved in S&W Wrong in S&W ~~Not copied from Wiki~~

Simplex Algorithm: *Said & Wegman (200, pp. 6–7) vs Wikipedia page for Simplex Algorithm, Feb. 9 2009 version (archived at http://en.wikipedia.org/w/index.php?title=Simplex_algorithm&oldid=269565766). Note that Said & Wegman was published on-line 13 July 2009, five months after this version of the wikipedia page.*

Said& Wegman, p. 6:	Wikipedia
The simplex algorithm is an algorithm to find a solution to a linear programming problem and is due to George Dantzig. We consider the linear programming problem by maximizing cTx subject to Ax ≤ b and x ≥ 0. Geometrically each inequality specifies a half-space in d-dimensional Euclidean space and their intersection is the set of all feasible values the variables can assume. The region is either empty, unbounded, or is a convex polytope. The set of points on which the objective function obtains the value v is defined by the hyperplane cTx = v. The solution to the linear programming problem will be by finding the largest v such that the hyperplane still intersects the feasible region. As v increases, the hyperplanes translate in the direction of the vector c. The hyperplane corresponding to the largest v that still intersects the feasible region will intersect a vertex, a whole edge, or a face of the polytope. In the case of a edge or face, it is still the case that the endpoints of the edge or face will achieve the optimum value. Thus the optimum value of the objective function will always be achieved on one of the vertices of the polytope.	In mathematical optimization theory, the simplex algorithm, created by the American mathematician George Dantzig in 1947, is a popular algorithm for numerical solution of the linear programming problem. Consider a linear programming problem, maximize $c^T x$ subject to $Ax \leq b, x \geq 0.$ In geometric terms, each inequality specifies a half-space in *n*-dimensional Euclidean space, and their intersection is the set of all feasible values the variables can take. The region is either empty, unbounded, or a convex polytope. The set of points where the objective function obtains a ~~given~~ value *v* is defined by the hyperplane $c^T x = v$. We are looking for the largest *v* such that the hyperplane still intersects the feasible region. As *v* increases, the hyperplanes translates in the direction of the vector *c*. Intuitively, and indeed it can be shown by convexity, the last hyperplane to intersect the feasible region will either just graze a vertex of the polytope, or a whole edge or face. In the latter two cases, its still the case that the endpoints of the edge or face will achieve the optimum value. Thus, the optimum value will always be achieved on one of the vertices of the polytope.

Identical text Minor difference Moved in S&W Wrong in S&W ~~Not copied from Wiki~~

Said& Wegman, p. 6:

The simplex algorithm is based on rewriting the linear programming problem in an

augmented form. The augmented form changes the basic inequalities to equalities by introducing the so-called slack variables.
In matrix form the problem becomes

Maximize Z such that

$$\begin{bmatrix} 1 & -c^T & 0 \\ 0 & A & I \end{bmatrix} \begin{bmatrix} Z \\ x \\ x_s \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix} \text{ with } x, x_s \geq 0$$

where x are the variables from the standard form,
xs are the slack variables from the augmentation process,
c contains the optimization coefficients,
A and b describe the system of constraint equations, and
Z is the variable to be optimized.
Suppose in the standard form of the problem there are d variables and m constraints, not counting the n non-negativity constraints. Generally, a vertex of the simplex corresponds to making d of the m + d total constraints tight, while adjacent vertices share d − 1 tight constraints. In the augmented form, this corresponds to setting m of the m + d variables (d original and m slack) to 0. Such a setting of the variables is called a basic solution. The m variables that are set to 0 are called the nonbasic variables. One can then solve the remaining d constraints, called the basic variables, that will be uniquely determined. The simplex algorithm begins by finding a basic feasible solution. At each step, one basic and one nonbasic variable are chosen according to the pivot rule, and their roles are switched.

Wikipedia:
From "Overview":
The simplex algorithm leverages this insight by rewriting the problem so that one of the vertices of the (possibly unbounded) polytope is easy to find, or it is revealed that the problem is infeasible.

From "Implementation":
The simplex algorithm requires the linear programming problem to be in augmented form. The problem can then be written as follows in matrix form:

Maximize *Z* in:

$$\begin{bmatrix} 1 & -\mathbf{c}^T & 0 \\ 0 & \mathbf{A} & \mathbf{I} \end{bmatrix} \begin{bmatrix} Z \\ \mathbf{x} \\ \mathbf{x}_s \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{b} \end{bmatrix}$$

$$\mathbf{x},\ \mathbf{x}_s \geq 0$$

where x are the variables from the standard form,
xs are the ~~introduced~~ slack variables from the augmentation process,
c contains the optimization coefficients,
A and b describe the system of constraint equations, and
Z is the variable to be maximized.
Suppose in the standard form of the problem there are n variables and m constraints, not counting the n nonnegativity constraints. Generally, a vertex of the simplex corresponds to making n of the m+n total constraints tight, while adjacent vertices share n-1 tight constraints. ~~There is a little subtlety when such a point in n-space does not fall in feasible region. Ignoring that,~~ in the augmented form, this corresponds to setting n of the m+n variables (n original and m slack) to 0. We call such a setting of the variables a basic solution. The m variables which are ~~purposely~~ set to 0 are called the nonbasic variables. We can then solve ~~for~~ the remaining n constraints, called the basic variables, which will be uniquely determined~~, as we will be careful not to step out of the feasible region.~~
The simplex algorithm begins by finding a basic feasible solution. At each step, one basic and one nonbasic variable are chosen according to the pivot rule, and their roles are switched.

Identical text Minor difference ~~Not copied from Wiki~~

Said& Wegman, pp. 6–7:

Klee and Minty developed a linear programming problem in which the polytope P is a distortion of a d-dimensional cube. In this case, the simplex method visits all 2d vertices before arriving at the optimal vertex. Thus the worst-case complexity for the simplex algorithm is exponential time.

However, the simplex method is remarkably efficient in practice. The simplex algorithm has polynomial-time average-case complexity under various distributions. The computational formulation of the simplex algorithm will appear in another study.

Wikipedia:

From "Overview":

In 1972~~,~~ Klee and Minty[2] gave an example of a linear programming problem in which the polytope P is a distortion of an n-dimensional cube. They showed that the simplex method ~~as formulated by Dantzig~~ visits all 2^n vertices before arriving at the optimal vertex. This shows that the worst-case complexity of the algorithm is exponential time. Since then it has been shown that for almost every deterministic rule there is a family of simplices on which it performs badly. It is an open question if there is a pivot rule with polynomial time, or even sub-exponential worst-case complexity.

Nevertheless, the simplex method is remarkably efficient in practice. It has been known since the 1970s that it has polynomial-time average-case complexity under various distributions.

Identical text Minor difference Moved in S&W Wrong in S&W ~~Not copied from Wiki~~

Other sections with similarities to Wikipedia

In addition to the Linear Programming section, the following other sections of "Roadmap for Optimization" are strikingly similar to sections in Wikipedia:

1. Said and Wegman's "Langrange Multipliers and Karush-Kuhn-Tucker Conditions" is similar to the Wikipedia page "Karush-Kuhn-Tucker conditions" for Jan 17 2009 (the latest page before Feb 28, 2009). The portion of Said and Wegman on p. 4 beginning "If x* is a local minimum", extending to "Also MFCQ is not equivalent to CRCQ", corresponds nearly word for word to the Wikipedia page. The only substantive difference is the omission of the QNCQ condition from Said and Wegman. Furthermore, there is another instance of the wording in Said and Wegman being less clear—arguably wrong—compared to its Wikipedia counterpart. Said and Weman introduce the regularity conditions with "The regularity conditions for a minimum point to satisfy the Karush-Kuhn-Tucker conditions are given below". However, Wikipedia states clearly (albeit less grammatically) that each condition is *sufficient by itself* to ensure regularity: "In order for a minimum point x * be KKT, it should satisfy some regularity condition, the most used ones are listed below".
2. Said and Wegman's "Interior Point Methods and Karmarkar's Algorithm" has several sections identical to the Wikipedia pages for "Interior Point Methods" and "Karmarkar's Algorithm". In particular, Said and Wegman's pseudocode for the affine-scaling algorithm is identical in nearly every respect to its Wikipedia counterpart (both the May 8 2011 page and the one for Feb 25, 2009). In two small ways, Said and Wegman's version is *less complete* than Wikipedia: Said and Wegman do not include x_0 in their list of inputs, and their line beginning "$\alpha \leftarrow \gamma$…" should have a trailing "i" subscript after the second h_v.
3. Said and Wegman's "Dynamic Programming" has many passages closely matching the Feb 8 2009 Wikipedia page, "Dynamic Programming". In particular, there are two places where the correspondence is very close indeed but the Said and Wegman version has errors (see comparisons on next page). In the opening sentence, there is a typo as "Bellman" is repeated twice in close succession. More substantively, Said and Wegman's schema of a dynamic programming algorithm misrepresents it as working "iteratively", whereas Wikipedia uses the far more correct "recursively". Given how central the recursive nature of this algorithm is, the Said and Wegman version is substantially wrong. The remainder of this section of Said and Wegman also closely matches the remainder of the Wikipedia page, even using the rather awkward phase "not intuitive" to describe the process of recognizing subproblems.

***Dynamic programming**: Said and Wegman vs. Wikipedia page, "Dynamic Programming" (Feb 8 2009 version: http://en.wikipedia.org/w/index.php?title=Dynamic_programming&oldid=269373908).*

Said& Wegman, p 7:

Opening sentences:

Dynamic programming, pioneered by Richard Bellman in the 1940s Bellman[5,6], is a method of solving problems that exhibit the properties of overlapping subproblems and optimal substructure.

(Note that the two references following the second "Bellman" are to the hardback and paperback versions of the same book.)

Schema of algorithm:

Optimal substructure means that optimal solutions of subproblems can be used to find the optimal solutions of the overall problem. Three steps are necessary:

1. Decompose the problem into smaller subproblems
2. Optimally solve the subproblems using the three-step process iteratively
3. Use the optimal sub-solutions to build an optimal solution for the original problem.

Wikipedia:

Opening sentences:

In mathematics and computer science, dynamic programming is a method of solving problems that exhibit the properties of overlapping subproblems and optimal substructure (described below). The method takes much less time than naive methods.

The term was originally used in the 1940s by Richard Bellman to describe the process of solving problems where one needs to find the best decisions one after another.

Schema of algorithm:

Optimal substructure means that optimal solutions of subproblems can be used to find the optimal solutions of the overall problem. … In general, we can solve a problem with optimal substructure using a three-step process:

1. Break the problem into smaller subproblems.
2. Solve these problems optimally using this three-step process recursively.
3. Use these optimal solutions to construct an optimal solution for the original problem.

Identical text Minor difference Moved in S&W Wrong in S&W ~~Not copied from Wiki~~

4. Said and Wegman's "Calculus of Variations" section (p. 8) has several passages very similar to the Feb 4 2009 Wikipedia page, "Calculus of Variations" (http://en.wikipedia.org/w/index.php?title=Calculus_of_variations&oldid=268383585). The second sentence of Said and Wegman matches the opening sentence of the Wikipedia page. The second paragraph of Said and Wegman's section matches the Wikipedia section, "Weak and Strong Extrema", differing only in notation. The Said and Wegman definition of *supremum norm* is given by a sentence fragment, as is the introduction of the fundamental lemma of the calculus of variations (which is presented with no apparent connection to the paragraph immediately preceding it).
5. The remaining sections of the article are simply examples of terrible writing, featuring phrases such as "intensively mathematically based optimization", "binary or bit strings", and "a method of trial-and-error problem-solving technique". The opening of the "Genetic algorithms" subsection is horrible:

   > As with evolutionary algorithms, genetic algorithms are an attempt at using the evolutionary laws of Darwin to formulate an algorithm of optimization. What these algorithms boil down to are two simple underlying principles.

   The introduction to "Simulated Annealing" bungles its rhetorical question:

   > Annealing is defined as a process of heating then cooling for the purposes of softening or making a metal less brittle. The logical question that follows this definition is how this is related to optimization.

   And what is a reader supposed to make of the concluding sentence of "Tabu Search"?

   > As such a recently developed method attributed to Glover and Laguna[7], the full implications and implementations of tabu search remain unknown.

   I didn't search for antecedent versions of these sentences because I didn't want to admit that there might be other publications with writing this bad.
6. The references section is nonstandard for an overview article. The first three references support detailed points within specific sections. The fourth is the original description of an algorithm, while the remaining three are full books describing families of algorithms (with Bellman's book inexplicably featured twice). This short list of six distinct references is then followed by "Further Reading", a laundry list of readings ranging from specific points about algorithms (Granville et al.) to texts covering entire fields of optimization. The list appears to comprise whatever publications related to "optimization" that the authors found ready at hand when writing the article. The list provides no sense of a systematic overview of the literature in optimization.