Surveys in Operations Research and Management Science 21 (2016) 1–17

Contents lists available at ScienceDirect

Surveys in Operations Research and Management Science

journal homepage: www.elsevier.com/locate/sorms

Review

# Consistent notation for presenting complex optimization models in technical writing

Michael D. Teter [a], Alexandra M. Newman [a,*], Martin Weiss [b]

[a] Operations Research with Engineering PhD Program, Mechanical Engineering Department, Colorado School of Mines, Golden, CO 80401, United States
[b] European Commission - DG JRC, Institute for Energy and Transport, Sustainable Transport Unit, Via Enrico Fermi 2749 - TP 441, I-21010, Ispra, Italy

ARTICLE INFO

*Article history:*
Received 18 June 2015
Received in revised form
16 May 2016
Accepted 16 May 2016

ABSTRACT

With an increase in computational power, and with recent advances in software, practitioners are formulating ever more complicated optimization models, many of which are drawn from interdisciplinary applications and are designed to reflect the details of real-world systems. While such models have proven useful in providing implementable solutions with a verified impact, they are also more cumbersome to document in a clear and concise manner. In this paper, we recommend: (i) conventions for defining sets, parameters, and variables, (ii) ways of presenting the objective and constraints, and (iii) means by which to organize formulations. While other conventions may be perfectly acceptable, we suggest one set of guidelines for graduate students, academics, and practitioners in need of clearly and easily presenting a large, complex optimization model. Self-study and/or the introduction of these principles into an applied, advanced graduate class can remove ambiguity from model formulations and improve the communication between modelers and their intended audience.



**Contents**



* Corresponding author.
*E-mail addresses:* mteter@mines.edu (M.D. Teter), newman@mines.edu (A.M. Newman), martin.weiss@jrc.ec.europa.eu (M. Weiss).

http://dx.doi.org/10.1016/j.sorms.2016.05.001

## 1. Introduction and background

Implementable and implemented, complex applications of optimization have become a reality in the past three decades owing to advances in both hardware and software. Correspondingly, graduate students, academics and novice practitioners can struggle with the way in which to mathematically formulate these optimization models in a consistent and understandable manner. Our intended audience is not the practitioner working with an already-established lexicon. Rather, we seek to suggest possible means to clearly express optimization models for those struggling to do so. In particular, we address graduate students in engineering and applied science and those beginning to work in applied research settings who wish to present their mathematical models in formal and technical writing, e.g., as a technical paper, journal article, or conference presentation. We also target those researchers who have yet to establish a lexicon, in part, because they might have been working with more theoretical models. With this audience in mind, our article is intended for self-study or as reference in graduate-level courses for those unfamiliar with written formulation techniques of complex models. An instructor or primary investigator could direct students or research associates to this writing.

Mathematical programming now consists not of wiring "three patch-boards, which became like masses of spaghetti" [1], as it did during the first implementation of the Simplex method (and, at which, Dantzig was "appalled"), but of sophisticated hardware and software, including algebraic modeling languages e.g., AMPL [2,3], and solvers, e.g., CPLEX [4], MINLPBB [5], and the toolkit MINOTAUR [6,7].

Optimization models of three decades ago were simple, often expressed as linear programs of the form:

$$\begin{aligned}&\text{minimize } \mathbf{cx}\\&\text{subject to } A\mathbf{x} = \mathbf{b}\\&\mathbf{x} \geq 0\end{aligned}$$

or as (mixed) integer programs of the form:

$$\begin{aligned}&\text{minimize } \mathbf{cx} + \mathbf{dy}\\&\text{subject to } A\mathbf{x} + B\mathbf{y} = \mathbf{b}\\&\mathbf{x} \geq 0,\ \mathbf{y} \text{ binary (or integer)}\end{aligned}$$

or as nonlinear programs of the form:

$$\begin{aligned}&\text{minimize } \mathbf{f}(\mathbf{x})\\&\text{subject to } \mathbf{g}(\mathbf{x}) \leq \mathbf{b}\\&\mathbf{h}(\mathbf{x}) = \mathbf{c}\end{aligned}$$

where $A$ and $B$ are $m \times n$ and $m \times n'$ matrices of left-hand-side constraint coefficients, respectively, $\mathbf{c}$ and $\mathbf{d}$ are $1 \times n$ and $1 \times n'$ row vectors of objective function coefficients, respectively, $\mathbf{b}$ is an $m \times 1$ column vector of right-hand-side data values for each constraint, and $\mathbf{x}$ and $\mathbf{y}$ are vectors of continuous and discrete decision variables, respectively, that conform. While we later suggest differentiating parameters and variables in their notational appearance based on the specificity required to express an applied model, note here that classical mathematical programs in their most general form use notation that differentiates a matrix (which is expressed here with a capital letter) and a vector (which is expressed here with a boldface, lower case letter). The functions $\mathbf{f}(\mathbf{x})$, $\mathbf{g}(\mathbf{x})$, and $\mathbf{h}(\mathbf{x})$ represent nonlinear expressions. For methodological development and small models, these forms suffice. However, for large, complex models expressing real-world details such as those found in chemical engineering [8], civil engineering [9], electrical engineering [10], mechanical engineering [11], and mining engineering [12], for example, a reader can become lost in the weeds of notation, the objective function, and the constraints.

Authors have emphasized the need to write well technically, and have suggested conventions in this vein, e.g., [13,14], and the references contained therein. Both Gillman [15] and Higham [16] provide handbooks on expressing mathematics, the former author concentrating purely on notation, and the latter including the communication of mathematics through writing a textbook or journal article, or preparing a presentation. Here, we focus specifically on presenting optimization models with the operations research field in mind.

Authors have provided technical guidance as to how to appropriately formulate a mathematical model. Brown [17] and Brown and Rosenthal [18] describe good practice in applied optimization modeling, including writing, formulating models, using modeling languages and spreadsheets, and analyzing results. Brown and Dell [19] provide excellent examples of how to formulate a model based on its structure. Newman and Weiss [20] describe some of these papers in more detail, and point to tutorials on optimization modeling and associated algorithms in general. These papers cover the foundations of the mathematics associated with building and implementing optimization models. By contrast, Powell [21] discusses the dialect of dynamic programming and proposes a standard notation for stochastic (dynamic) programming problems. We also focus on the presentation of the mathematics within formal and technical writing, but within a more general framework, not on the accompanying description or the dissemination of a model's results to a wider audience. We emphasize that our conventions are only suggestions. Thus, our intentions mimic those of Halmos [22], who states that his essay on "How to write mathematics" is subjective, i.e., that it might have been called "How I write mathematics". He emphasizes the importance of good choice of specific mathematical notation as well as the way in which an entire paper should be written. Similarly, our article enables the reader with good mathematical knowledge of his formulation to write it in a cohesive format that is easy to parse and understand.

It is the authors' experience as instructors, research directors (in a lab or university environment), and as journal editors that practitioners will substitute code for a mathematical formulation. We suggest that a formulation, in its entirety, should contain a section on indices and sets, parameters (with units), variables (with units), the objective function, and the constraints, in that order, written with mathematical symbols, as appropriate. The writer should never include modeling language code such as GAMS or AMPL as a substitute for a mathematical formulation unless said code is for internal consumption, e.g., a project group within a research lab. Not only are these modeling languages not universally known among practitioners, but coding constructs can only approximate mathematical notation. Furthermore, poor coding conventions can lead to misinterpretation or errors of translation.