IFSR International Series on Systems Science and Engineering

George J. Friedman
Phan Phan

Constraint Theory

Multidimensional Mathematical Model Management

Second Edition

Springer

CONSTRAINT THEORY

MULTIDIMENSIONAL MATHEMATICAL MODEL MANAGEMENT

International Federation for Systems Research International Series on Systems Science and Engineering

IFSR was established "to stimulate all activities associated with the scientific study of systems and to coordinate such activities at international level." The aim of this series is to stimulate publication of high-quality monographs and textbooks on various topics of systems science and engineering. This series complements the Federations other publications.

A Continuation Order Plan is available for this series. A continuation order will bring delivery of each new volume immediately upon publication. Volumes are billed only upon actual shipment. For further information please contact the publisher.

Volumes 1–6 were published by Pergamon Press

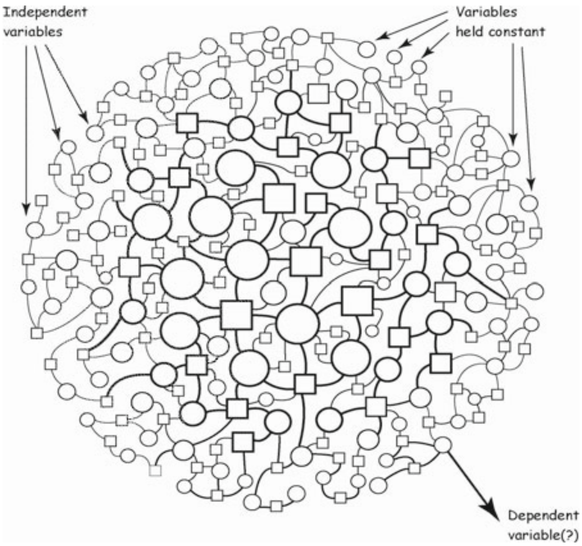
More information about this series at http://www.springer.com/series/6104

CONSTRAINT THEORY

MULTIDIMENSIONAL MATHEMATICAL MODEL MANAGEMENT

Second Edition

George J. Friedman • Phan Phan



Springer

George J. Friedman
Viterbi School of Engineering
University of Southern California
Los Angeles, CA, USA

Phan Phan
Viterbi School of Engineering
University of Southern California
Los Angeles, CA, USA

ISSN 1574-0463
IFSR International Series on Systems Science and Engineering
ISBN 978-3-319-54791-6 ISBN 978-3-319-54792-3 (eBook)
DOI 10.1007/978-3-319-54792-3

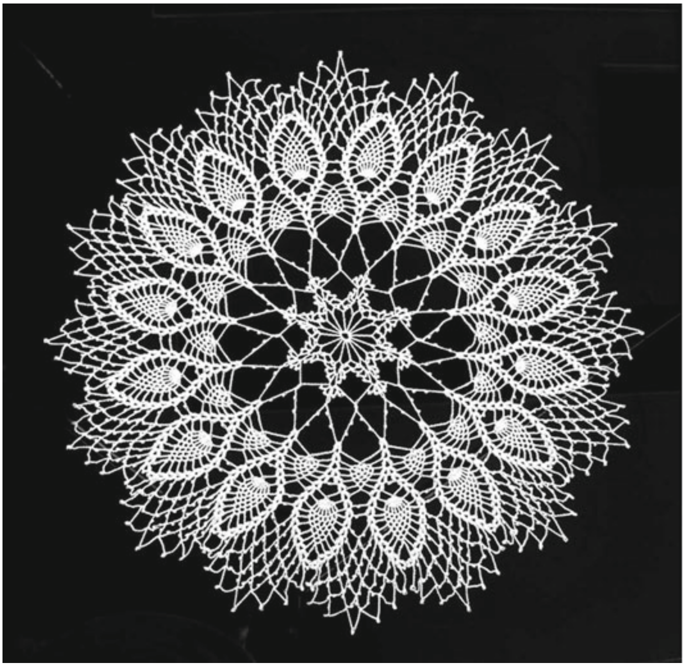
Library of Congress Control Number: 2017933682

Printed on acid-free paper

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

Fronticepiece

24” Doily, designed and constructed by Regina Oberlander (Dr. Friedman’s mother) circa 1915 in Chinedeev, near Muncacevo, Austro-Hungarian Empire.



A single circuit cluster with over 1,000 independent simple circuits (thus there are 1000 more edges than vertices)

This book is dedicated to our wives, who have helped us find the time and priority for our mathematical dreams.

Preface

At first glance, this might appear to be a book on mathematics, but it is really intended for the practical engineer who wishes to gain greater control of the multidimensional mathematical models which are increasingly an important part of his environment. Another feature of the book is that it attempts to balance left- and right-brain perceptions; the authors have noticed that many graph theory books are disturbingly light on actual topological pictures of their material.

Constraint Theory was originally defined by George Friedman in his PhD dissertation at UCLA in 1967 and subsequent papers written over the following decade. There was a dearth of constraint theory publication after the 1970’s as Dr. Friedman was working on several classified aerospace programs wherein publication of any kind was most difficult. The first edition of this book was published in 2005. Constraint Theory was further extended by Phan Phan in his PhD dissertation at USC in 2011, leading to this second edition.

Acknowledgments

Over the past several years, Constraint Theory has been a substantial part of a special graduate course in the Systems Architecture and Engineering program at the University of Southern California, where the authors are adjunct faculty members. The feedback from this group of bright graduate students was invaluable. Special thanks are given to these graduate students who performed research studies – summarized in Appendix A – on Constraint Theory: Leila Habibibabadi, Kathya Zamora-Diaz and Elliott Morgan [1]. Extra special thanks are given to Mr. Gary L. Friedman who provided the extremely valuable service of intellectual reflector, editor and digital format manager for this book.

Introduction

Many thousands of papers have been written about the accelerating pace of increased complexity and interactivity in virtually every walk of life in the developed world. Domains which previously could have been studied and managed separately – such as energy, the environment and economics – must now be dealt with as intimately intertwined disciplines. With its multitude of additional capabilities, complex systems also provide a treacherous array of fragile failure modes, and both the development and operation of new systems are an increasing challenge to the systems engineer. Advanced technology is the primary driving force behind the increasing complexity and the enthusiastic pushing of immature technologies is behind most of the early failures in the development phases.

Perhaps the most significant advanced technology employed in new complex systems is the computer science family with its ancillary disciplines of communications and software. Fortunately, computer science also represents a major opportunity to control the design and operation of complex systems because of its ability to perform multi-dimensional modeling to any level of detail desired. Math models have been used in support of every phase of systems engineering, including requirements management, preliminary design, interface and integration, validation and test, risk management, manufacturing, reliability and maintainability, training, configuration management and developing virtual universes to measure customer preferences prior to the implementation of the design. Properly used, the enormous power of modern computers can even furnish researchers with a synthetic world where theories can be tried and tested on validated models, thus avoiding far more expensive tests in the real world. A wide variety of questions – or "tradeoffs' – can be asked of the models and, at least in theory, the analyst has a free choice as to which computations he wishes to observe and which variables he desires to be independent. Philosophically, it can even be argued that the math model employed in this fashion provides the technologist a virtual extension of the scientific method itself.

Those who have actually wrestled with large-scale models will complain that the above description is far too rosy. Submodels which are developed by separate organizations are normally very difficult to integrate into an overall working model; they often must be dealt with as "islands of automation." The greatest of care must be taken to make sure that the definition of each variable is clear and agreeable to every member of the team. In general, it is difficult to distinguish between a model and the computer program, and if a computational request is made which reverses dependent and independent variables, then the model must be reprogrammed. To say the least, much diligent effort must be undertaken to obtain the many advantages promised by mathematical modeling.

However, even after the diligence, there exists a much deeper problem that often diminishes the utility of math modeling; it is associated with the traditional "well posed" problem in mathematics. We need to know whether the model is internally consistent and whether computational requests made on it are allowable. The alarming facts are that models constructed by diverse teams – and this is normally the case for very large models – have internal inconsistencies and that most of the possible computational requests which can be made on even consistent models are not allowable. This problem is the domain addressed by Constraint Theory and is the subject of this book.

Chapter 1 provides an example of low dimension, showing how problems of consistency and computational allowability can arise in even simple situations. The reader is introduced to the two main characters of the book – an experienced manager and an analyst – whose dialogue will hopefully illuminate the book's many concepts. The bipartite graph is introduced, as are a few simple rules. However, the analyst argues that, in order to expand the tools to models of very high dimension, and in order to trust the reliability of these tools, the theory must be based on a more rigorous foundation. "Only the simplest 5% of graph theory and set theory are required", he claims.

Chapter 2 begins to establish the rigorous foundation by defining four "views" of a mathematical model: 1) set theoretic, 2) submodel family, 3) bipartite graph, and 4) constraint matrix. The first two views are full models; the last two views are metamodels. Then, rigorous definitions of consistency and computational allowability are made in the context of these views.

Chapter 3 discusses the similarities between language and mathematics and provides some general consistency and computability results with respect to any class of relation. In order to provide a basis for the next three chapters, three classes of exhaustive and mutually exclusive relations are defined: discrete, continuum, and interval.

Due to the amount of new materials resultant from Dr. Phan's research as part of his doctoral dissertation, previous Chapter 4 in the first edition has been split into two new Chapters 4 and 5 in this second edition. Chapters 4 and 5 represent the core of constraint theory at its present stage.

As before, the new Chapter 4 addresses the constraint theoretic properties of *regular* relations, the most important type within the continuum class, and the most often employed in the development of multidimensional math models. The simple rules presented in Chapter 1 are rigorously proved employing the foundations of Chapters 2 and 3. The topological properties of the bipartite graph are analyzed to provide key conclusions of the model's consistency and computational properties.

A specific type of subgraph within the bipartite graph, called the *Basic Nodal Square (BNS)* is identified as the "kernel of intrinsic constraint" and is accused of being the culprit in model inconsistency and unallowable computability. Trivially easy computations on the bipartite graph – such as circuit rank and constraint potential – are shown to have enormous utility in locating the BNSs which hide in tangled circuit clusters.

Additionally, the newly updated Section 4.6 extends, in more prescriptive details with graphical illustrations, the step by step algorithm for locating BNSs within a model graph.

The new Chapter 5 discusses the general issue of constraint propagation through a connected model graph of regular relations. A detailed procedure for determining model consistency and computational allowability in such a model is introduced.

In particular, Section 5.4 introduces new mathematical definitions and theorems to enable the detection of overlapping BNSs. Section 5.5 outlines techniques to relieve over-constraint among them. Section 5.6 describes a precise procedure to expand their resultant constraint domains. Section 5.7 prescribes, in details with graphical illustrations, a step-by-step algorithm to process computation requests made on a model. And Section 5.8 provides a constraint theory toolkit to employ the rules and theorems in an orderly manner and which can find BNSs trillions of times faster than brute force approaches.

Chapter 6 addresses the constraint properties of *discrete* and *interval* functions such as those from Boolean algebra, logic and inequalities. These classes of relations are less important in support of modern math modeling, but strangely, it was the first that the author studied in his development of Constraint Theory. It was easier for him to imagine multidimensional sets of points than multidimensional sets of continuous functions. Interval relations require the greatest interaction between models and metamodels, and the concept of constraint potential is less useful than for regular relations.

Chapter 7 provides a compact structure of constraint theory. All postulates, definitions and theorems are listed and their logical interrelationships are displayed in the form of quasi-bipartite graphs.

Chapter 8 presents detailed examples of the application of constraint theory to the areas of operations analysis, kinematics of free-fall weapon delivery systems and the dynamics of deflecting asteroids with mass drivers.

Chapter 9 summarizes the book and provides the manager and analyst a final opportunity to dialogue and discuss their common background.

Problems for the interested student are presented at the end of most chapters, so this book could be employed as a text for a graduate course – or senior level undergraduate course – in Systems Engineering or mathematical modeling.

Of course, a complete list of references is provided, as well as an index.

Several appendices treat detailed material to a depth that would slow down the natural rhythm of the exposition if they were included in the chapters themselves. Appendix A is noteworthy in that it summarizes the research projects on "computational request disappointments." On models approximately the size of Chapter 1's "simple example" – eight variables – the percentage of allowable computational requests based on the total number of possible computational requests is only on the order of 10%. It is presently "Friedman's conjecture" that as the dimensionality, K, of the model increases, the number of allowable computational requests also increases, perhaps as fast as the square of the model's dimension or K^2. However, the number of possible computational requests increases far faster: 2^K. Thus, for a 100-dimension model, only 10^{-26} of all possible computational requests will be allowable! Models of thousands of dimensions have been built and are planned; so the ratio of allowable to possible computational requests is enormously worse that even this incredibly low number. The technologist who wishes to gain maximum benefit from asking his model to perform any computation his imagination conjures up will certainly be disappointed! A tool such as constraint theory which will lead him to the 10,000 computational requests (K=100) or 1,000,000 requests (K=1,000) which are allowable should be valuable.

Appendix B provides a very brief overview of graph theory with the objective of justifying why the bipartite graph was chosen as the primary metamodel for constraint theory.

Appendix C describes the rigorous logic of the difference between "if and only if" and "if" types of theorems. Most of constraint theory's theorems are of the latter category – a source of confusion to many students.

The newly updated Appendix D establishes fundamental algebraic structures which are essential to implement constraint theory. These include definitions and properties of general vector spaces, and binary set operations.

A Warmup Problem in Complexity

This book makes substantial use of a mathematical structure from graph theory called a bipartite graph. In the past, bipartite graphs have been employed to solve "pairing" problems associated with various social situations such as picnics or dinner parties.

Out of respect for this tradition, let us consider a set of five men – named Jack, Jake, Jude, Juan, and Jobe – and a set of five women – named Jane, Joan, June, Jean, and Jenn. Let us define a *relationship pattern* as a complete description of all heterosexual relationships between the five men and five women. For example:

- In the *communal* pattern, every man has a relationship with every woman. There is one such pattern.
- In the *celibacy* pattern, none of the men have a relationship with any of the women. Again, there is one such pattern.
- In the *male harem* patterns, one of the men has a relationship with each of the women, but all the other men are devoid of relationships, except perhaps to be eunuchs. There are five such patterns. Similarly, there are five possible *female harem* patterns.
- In the *monogamy* patterns, each man has a relationship with exactly one woman and vice versa. There are 5!=120 such distinct patterns.

And so on. There are many more patterns. The question is: *What is the total number of possible heterosexual relationship patterns between five men and five women?*

The answer – discussed in Chapter 4 and Appendix A – may surprise you: it's over 30 million (!). It certainly surprised the author and changed an important objective of his research agenda. Moreover it represents the hidden depths possible in apparently simple problems of low dimension as well as a challenge to one's belief in intuition or rational mathematics.

About the Authors

George Friedman is a Professor of Practice in the Astronautical Engineering Department of the Viterbi School of Engineering of the University of Southern California. He has developed and taught graduate courses in systems engineering with emphasis on the management of complexity and decision science. This book is the product of one of these courses.

He has had over 45 years of experience in industry, retiring from the Northrop Corporation as their Corporate Vice President of Engineering and Technology. He worked on a wide variety of aerospace programs and served as a consultant to all branches of the Department of Defense, NASA, the National Science Foundation, and Department of Energy as well as to the NATO industrial advisory group.

He was a founder of the International Council on Systems Engineering (INCOSE), served as its third president, was elected a fellow and is on the editorial board of INCOSE's journal, *Systems Engineering.*

He has also been a member of the Institute of Electrical and Electronic Engineers (IEEE) since its formation from the IRE and AIEE, was elected a fellow and was the vice president for publications of the *IEEE Transactions on Aerospace and Electronics Systems.* He received the Baker Prize for the best paper published by all societies of the IEEE in 1970 – the subject of the paper was Constraint Theory.

He was a former director of research at the Space Studies Institute at Princeton, and had supported several new technologies associated with the long range development of space.

He received the Bachelor's degree in engineering at the University of California at Berkeley and the Masters and Doctorate at UCLA. The topic of his PhD dissertation was constraint theory [2, 3].

Phan Phan is a Lecturer in the Astronautical Engineering Department of the Viterbi School of Engineering of the University of Southern California (USC). He has assisted and taught graduate courses in systems engineering, systems management, lean operations and economic analysis.

He has had over 36 years of technical and managerial experience in government, military and various industries, including oil & gas exploration, commercial and military aircraft, unmanned sensors, and major weapon systems.

His industry assignments included Mobil Research & Development, General Dynamics, Lockheed Aeronautical Systems, McDonnell Douglas and Boeing Integrated Defense Systems. As a registered Professional Engineer in California, he currently works as a reliability analyst with the Naval Surface Warfare Center – Corona Division.

He has also served in the U.S. Navy Reserve as an Engineering Duty Officer, attaining the rank of Captain, and currently assigned to Naval Sea Systems Command. His previous Navy assignments included Office of Naval Research/Naval Research Laboratory, Program Executive Office Integrated Warfare Systems, Naval Space & Warfare Systems Command, Mobile Mine Assembly Group, Naval Weapons Station Seal Beach and Naval Shipyard Long Beach.

He received his B.S. in Engineering from the University of Alabama in Huntsville, Master of Engineering from the University of Texas at Arlington, MBA from California State University – Fullerton, M.S. in Systems Architecture & Engineering from USC, Master of Engineering Acoustics from the Naval Postgraduate School, and Ph.D. in Industrial & Systems Engineering from USC.

The topic of his doctoral dissertation was "Expanding Constraint Theory to Determine Well-Posedness of Large Mathematical Models" [22], the main contribution to the second edition of this book.

Contents

Chapter 1 MOTIVATIONS

What is Constraint Theory and why is it important?

1.1 TRENDS AND PROBLEMS IN SYSTEM TECHNOLOGIES

Gone forever are the simple days! Virtually every identifiable trend is driving humanity's enterprises into more intimate interaction and conflict. Increased population, accelerated exploitation of resources, and expanded transportation have brought the previously decoupled worlds of economics, energy and the environment into direct conflict. With the greater efficiency of travel and communication, the emergence of global marketplaces and the revolution in military strategies, the international world is incredibly more interactive, multidimensional and complex than even a decade ago. Locally, we observe ever tighter coupling between emerging problems in crime, poverty, education, health and drug misuse. All these issues have been aggravated by an explosion of new technology and – especially in the United States – a compulsion to force these new technologies into early and often simultaneous application. The most vigorous of these advancing technologies – digital computation – brings with it an unexpected complexity challenge: software and the management of highly complex and multidimensional mathematical models.

Fortunately, this most rapidly advancing technology of computer science not only adds to the complexity of *designed* systems, it also contributes enormously to *designing* these systems themselves. A host of new "computer assisted" software packages are published each year, running the gamut from Computer Assisted Design (CAD), Computer Assisted Engineering (CAE), Computer Assisted Systems Engineering (CASE), Computer Assisted Manufacturing (CAM), Computer Assisted Instruction (CAI), and eventually

G.J. Friedman, P. Phan, *Constraint Theory*, IFSR International Series on Systems Science and Engineering, DOI 10.1007/978-3-319-54792-3_1

to Computer Assisted Enterprise Management (CAEM). This family of tools permits the design engineers to control a virtually unlimited number of variables, to predict behaviors and performance of systems still in their early conceptual stages, to optimize with respect to detailed criteria, to effect interdisciplinary integration and to perform design changes with unprecedented speed and accuracy. It is not an exaggeration to claim that without this array of computer-based tools, many systems that exist today would have been impossible to design and implement.

However, as most systems engineers who attempt to gain benefit from these tools are well aware, computer-based design is a mixed blessing. A common complaint is that the various programs which support facets of the total problem are "islands of automation" – they are difficult to integrate into a total system problem solving capability. Another problem is that the tools are virtually useless in sorting out the variety of languages and technical shades of meaning, especially on highly interdisciplinary systems. Yet another challenge for the engineering and program managers is the vigorous and frequent upgrading of every hardware and software package causing unprecedented costs of initial installation and training to the overall design process, not to mention the inevitable bugs in the early versions. Many companies have even established new organizations whose members are expert in computer-assisted programs, and not expert in the technical design itself.

Observers who watched the agonizing entry of computers over the last several decades into many diverse worlds such as financial management, stock market trading, airline ticketing, air traffic control, and education should be optimistic that eventually computer-based design will also become an efficient tool which will become so easy to use that investing in it will be clearly justified. But in order for this dream to occur, there are more problems to solve – deeper problems than getting the definitions sorted out and software packages to work together.

Even when the willing but cognitively challenged computational giants of computer based tools are completely manageable, several fundamental problems will still exist, mostly on the cognition and mathematical levels.

Nothing is said in any of the present set of textbooks on Systems Engineering about the regrettable "subdimensionality" of the human intellect. Despite the fact that a thorough description of a modern complex system requires the understanding and integration of hundreds to thousands of variables, cognitive scientists have known for decades that the human mind is limited in its perceptive powers to a mere half-dozen dimensions. Regardless of all our other miraculous gifts such as language, art, music, imagination, judgment, and conscience, our dimensional perceptive power is tiny compared to the challenges of designing complex systems, and sadly, it appears that this is a “wired in” shortcoming of our nervous system and thus we cannot hope to be trained to attain a higher perceptive capability. This “dimensionality gap” is severely aggravated by the habit many self-styled “decisive managers” have in further suppressing their limited perception by searching for simplifications such as “the bottom line,” “the long pole in the tent” or “getting the right angle” in attempting to make complex descriptions more comprehensible to them. Typically, when the dimensionality of the model overwhelms that of the decision maker, and he sees results which appear to be anti-intuitive, he will tend to distrust these results as the product of software bugs or other errors. Thus, major opportunities to learn from the enhanced power of modeling are lost because the operation of the computer becomes more and more opaque to the decision maker as the dimensionality increases.

As an aside, we humans also have problems with numbers: we cannot perceive 29 in the same way we perceive 5. The raw arithmetic perception of the average person is “the magic number seven, plus or minus two,” according to the cognitive scientist George Miller. However, after a lifetime of dependable experience with arithmetic algorithms, we have the illusion that we can understand and manage entities such as 29, or even 29,000,000,029.

The other fundamental problem was previously referred to as the “well-posed” problem in mathematics. That is, when a mathematical model is established, is it internally consistent? When computational requests are desired based on this model, are they allowable? If the answer to either question is “no,” then we have a situation which is not “well posed” and we can expect nonsensical results or jammed up attempts to program. This problem is made worse by the fact that in most digital computer programs, models are built with a unidirectional computational flow that was anticipated by the programmers, but is not necessarily responsive to the needs of the decision makers. It was a source of great irritation to this author to be told many times over his career that a computational request was “impossible” because the model was programmed with another computational flow in mind. However, when the reprogramming *was* done in an attempt to be more responsive, more fundamental problems frequently arose.

An example of these problems will be useful at this point. The example given in the next section was chosen to be as simple as possible, but still indicating aspects of the well posed problem that can arise even without our entering a dimension so high that our perceptions are boggled.

1.2 AN EXAMPLE OF LOW DIMENSION

A decision-making manager was authorized to initiate the preliminary design of a new system development by his board of directors. In the true spirit of systems engineering, he realized the importance of making the best decisions as early in the system development process as possible. Accordingly, he gathered a team of the best specialists available, along with a systems analyst to help organize the math model that he hoped would guide him to strategic systems tradeoffs and decisions.

The chief systems engineer stressed that, in order for an "optimum design" to exist, it was necessary to define a total systems optimization criterion, T:

$$T = PE/C \qquad (1)$$

where:

P was the political index of acceptability by the board of directors,
E was the system effectiveness, and
C was the life cycle cost of the system.

The operational chief, expressing a weariness with the overly aggressive use of new and unproved technology on most of his previous systems, wanted to stress that most of the total system cost should be applied to operations and support, not new systems development. Thus, he contributed this limitation:

$$D = k_1C, \text{ where } k_1=0.3 \qquad (2)$$

where D, the development cost, was to be limited to 30% of the total cost.

The operations and support specialist, attempting to predict the level of cost after production and delivery were complete, provided:

$$S = X + 0.5D \qquad (3)$$

where:

S is the total support cost
X is the cost of ops and support if there were no new technology
D is the development cost for the system, including new technology.

The systems costing and estimating specialist contributed the obvious: