BOOK REVIEW

of George J. Friedman:
Constraint Theory: Multidimensional Mathematical Model Management
IFSR International Series on Systems Science and Engineering, Volume 23
reviewed by John N. Warfield
June, 2005

If, as Ralph Waldo Emerson said, “a foolish consistency is the hobgoblin of little minds, adored by little statesmen and philosophers and divines” [*Essays. First Series. Self-Reliance],* what should be said about unrecognized inconsistencies in large system models that are responsible for the waste of millions or even billions of dollars, the loss of countless working days, and production delays of sorely needed products?

It is rare, essentially unique, that a high-level technical leader not only has made a unique contribution to a major theory related to consistency in modeling of systems, but also has had an opportunity both to observe the potential utility of that theory in a variety of large systems programs, and to teach the aggregated findings to students who work with him on the pedagogy of the material.

Yet this is what we find displayed in the book *Constraint Theory* by George J. Friedman. Developed in his doctoral work in the late 1960's, tested as he rose through the ranks at Northrop Corporation in ensuing decades to become Corporate Vice President of Engineering and Technology, and finally, in retirement from Northrop, taught as an Adjunct Professor in systems engineering at the University of Southern California; this book reflects insights that he gained over more than three decades of attention to his pioneering development that he christened “constraint theory” [not to be confused with other work having somewhat similar names, but very different content].

This period of activity has not been completely free of frustration for George Friedman. High-level fiscal managers are seldom receptive to what they are likely to perceive as thorny–something that is seen as a nuisance to attempt to apply: theory that may demand significant, not-readily obtained cooperation and integration of work carried out among diverse groups or even diverse organizations; theory that is best served with software which may not even be available. Yet to the extent that this frustration has had a telling impact on George Friedman, it may have been a very positive factor in the development of his book.

The book has emerged partly from his mind as a written dialog between a technical manager and an analyst: something he is well-qualified to do since he has lived in both roles (though probably not with the managerial attitudes portrayed, at least concerning constraint theory). The potential conflicts surface. The potential benefits and contributions of constraint theory, and the negatives that are almost certain to accrue by ignoring it, are highlighted in this dialog.

In the outer world of engineering, this type of presentation may forearm both the analyst who wishes to influence the technical manager in the direction of applying constraint theory, and the marginally-receptive technical manager who must get a budget line item to support the application of constraint theory, and who needs to have enough insight to argue knowledgeably for that line item.

In the process of presenting constraint theory, George Friedman offers major contributions and several supporting contributions that are essential. The major contributions are the ones that motivated the development of constraint theory in the first place:

- **Model Consistency Analysis and Inconsistency Location**: To demonstrate how to determine, from the bipartite structure of a model, whether the model is consistent and, if it is not, to determine from the structure of the model, where the inconsistency(ies) is (are) found; so that timely corrective measures can be taken, before large amounts of time and money (and ultimately, perhaps, even lives) are wasted

- **Computation Allowability Analysis.** To determine whether a computation that is requested from the model is allowable (in order to avoid brute force attempts to squeeze a computed result out of a model that is beyond the capacity of the model to produce); once again so that timely corrective measures can be taken, before large amounts of time and money (and ultimately, perhaps, even lives) are wasted

George Friedman recognizes that no matter how well constraint theory is formulated, in application it is subject to the foibles of initial human construction. Misuse of language, failure to define terms adequately, omission of important variables, built-in insensitivity to cognitive limitations—all of these (and other) behavioral factors have the potential to foil any analytical or synthetic system. So it is that the combination of sound management, sound analysis, sound theory, and the kind of excellent algorithms based in constraint theory can come together to enable effective systems engineering to occur. These recognitions lend importance to the key supporting contributions, which include the following:

- **Founder of Discursivity.** Even before model consistency and computational allowability can be assessed, it is necessary to construct a discursive language in which the necessary concepts can be embedded, and the necessary conversations can be enabled. George Friedman has provided this discursivity, and a herculean task it has surely been to do so; requiring extensive and highly-correlated inter-definition, with the precision available only through carefully-crafted mathematical language.

- **Graphical Interpretability.** With the extensive nature of the discursive algebraic language, a tremendous cognitive burden lurks. George Friedman has circumvented this burden significantly by tying the algebraic language to the graphical language through the isomorphism of the bipartite graph.

- **Computer Implementation.** The ultimate relief from the cognitive burden of the constraint theory (which is inherent in the nature of the material, and not a consequence of its formation) lies in transferring as much of it as possible to computer software. This is not a task that George Friedman could have expected to accomplish in his employment. He has, however, started to program the main principles of constraint theory, and it remains for his successors, on the one hand to understand the importance of this task and, on the other hand, to take the necessary steps to get it done; not only for use in the system design community, but also in the higher education community.

It is clear that Friedman has a strong bent toward systems engineering, where most of his professional life has been focused, but he is not blind to other issues in society, as he makes clear in Chapter 1 of the book. Still if the engineering community cannot be awakened to the merits in this work, one wonders where the leadership is going to come from in other social arenas.

Modeling. While the title “Constraint Theory” is the most appropriate one for the book, it could also be described by a longer title: “how to avoid wasting huge amounts of time with defective mathematical models”. In order to illuminate this longer phrase, it is necessary, as mentioned earlier, to introduce some more detailed language than has been commonplace in the modeling community, and Friedman has contributed to such a language.

Earlier Kaplan[1], in considering the general nature of inquiry, had distinguished two types of theory on the basis of internal primacy. He defined a **field theory** as one which takes as fundamental “a system of relations among certain elements, explaining the elements by reference to these relations”. He defined a **monadic theory** as one that gave primacy to the relations and illuminated the relations by reference to the attributes of what they relate. Friedman is not quite satisfied with this categorization of models. Instead he requires a triadic structure.

He prefers a three-level categorization, starting with **protomodel** in which a set of elements is recognized but the relations among the elements are too dimly noticed to be articulated; then a **model**, in which the detailed elements and relations are collectively and mathematically articulated and, finally, a **metamodel** in which those details are subsumed (though in a recoverable way, if necessary) in larger objects such as graphs or matrices upon which operations can be carried out and which have properties not observable by visual inspection of a model alone.

If one imagines *homo sapiens* observing a complicated situation and, at first, sensing it

---

[1] A. Kaplan (1964): ***The Conduct of Inquiry***, San Francisco: Chandler.

only through a protomodel in which a few elements are given names; later, while aging rapidly into the computer age, developing a model in which those and other elements are related and feeling quite competent but being unable to determine the consistency or allowability of computations involving those elements and their relations; and finally constructing metamodels from the models using constraint theory, whereupon it becomes feasible to determine the consistency and allowability of computations; we have traveled with this creature who has passed through the various stages in the evolution of the theory and utility of modeling, using the language of George Friedman. And a worthy language it is, too.

The Complexity Threshold and "Friedman's Conjecture". One of the most troublesome aspects of systems engineering, and of the world of analysis and synthesis as a whole, is the virtually universal state of denial that persists concerning the prevalence of a complexity threshold beyond which conventional assumptions about how analysis, modeling, and synthesis no longer apply. Your reviewer has dealt with this extensively in his own publications, and will not dwell on the subject here. But I bring it out to emphasize a special contribution of George Friedman in this book which he calls "Friedman's conjecture". Appendix A discusses this topic in more detail than I will treat it here, but I mention it here because if the existence of a complexity threshold cannot be established in the organizational power structure, the future of constraint theory or any other powerful methods of working with complexity will always be shaky. Friedman's conjecture, which he illustrates with a straightforward example is:

> "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 dimensions or K^2. However the number of possible computational requests increases far faster: 2^K."

He goes on to illustrate that for a model of dimension 100, only 10^{-26} of all possible computational requests will be allowable! Since models of thousands of dimensions have been built and are planned, the ratio of allowable to possible computational requests is enormously worse than even this incredibly low number.

In effect what Friedman is showing is roughly that, if his conjecture is correct, the likelihood that one will be able to compute anything from a model of high dimensionality is nil, unless one has applied constraint theory to assure the consistency of the model, and to determine what computations are allowable.

[I was advised some years ago that no one strives to find solutions from the high-dimensional econometric models, but rather one assumes solutions based on economic beliefs, and then adjusts parameters to see what they would have to be yield the assumed solutions. Did someone say something about model strategy?]

In light of the fundamental role of the concept of dimension in constraint theory, one must note that the term retains a highly intuitive flavor colored by the long-standing electrical network concept—a flavor that is not readily transferred to fields outside of engineering. On page 78, Friedman says that his Theorem 14 “essentially tells us the dimensionality of the circuit vector space”. His second index reference to dimensionality occurs on page 129 where the term is used as “dimensionality limitations of the human mind”. There may be a significant conceptual gap between these two views of dimensionality which can benefit by further reflection and research–research that is not likely to affect constraint theory in any negative way, but which may go a long way toward circumscribing its connection to model-building as opposed to model analysis.

Organization of the Book. After the helpful front material, which should be read to gain perspective, the book is organized into eight chapters, four appendices, a set of references, and an index. I will discuss now the specific content of these parts of the book, in the light of what has gone before in this review. In much of this discussion, I draw freely on the author’s preface, editing as I see fit, to keep the review to a reasonable length. With apologies to the author, I note that the reader can always refer to the book for the full description.

Chapter One 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.

Chapter Two begins to establish [a] rigorous foundation. Four views are introduced: 1) set theoretic model, 2) sub-model family, 3) bipartite graph metamodel, and 4) constraint matrix metamodel. ...rigorous definitions of consistency and computational allowability are made in the context of these views.

Chapter Three discusses the similarities between language and mathematics and provides some general consistency and computability results with respect to any class of relation. ...three classes of exhaustive and mutually exclusive relations are defined: discrete, continuum, and interval.

Chapter Four emphasizes the constraint theoretic properties of *regular* relations, the most important type in the continuum class, and the most often used in developing multidimensional math models. 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. A constraint theory toolkit is provided to help you use the rules and theorems in an orderly manner. It can help locate BNSs trillions of times faster than brute force approaches. This chapter is the longest in the book and represents the core of constraint theory in its present stage.

Chapter Five emphasizes constraint properties of *discrete* and *interval* functions such as those from Boolean algebra, logic and inequalities. Interval relations require the greatest interaction between models and metamodels. The concept of constraint potential is less useful than for regular relations.

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

Chapter Eight 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 used as a text for a graduate course -- or senior level undergraduate course -- in systems engineering or mathematical modeling.

A 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%. [As mentioned earlier in this review] 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 the theorems of constraint theory are of the latter category – a source of confusion to many students.

Appendix D develops the similarity between vector spaces and graph theory circuits. The concept of independence in both worlds is strikingly similar and the ability to analyze circuits in graph theory has powerful ramifications in constraint theory because basic nodal squares – the source of intrinsic constraint – are only found in circuit clusters. (A BNS is always within a circuit cluster, but not every circuit cluster contains a BNS.)

Concluding Remarks. There is ample evidence in the literature that three threads of research can be consulted to gain significant perspective on working effectively with complexity in modeling. One is microbiology, whose fruits remain to be harvested, and whose future is the subject of intense speculation. Another whose contributions are already in view is cognitive psychology and what has been learned about behavioral pathologies, as well as what has already been proved about means of overcoming or circumventing these limitations. The third is graph theory and the various ways in which the scanning powers of the human eye can be put to work in conjunction with the brain to develop pictures that are superior to what the ear can aggregate.

There are at least three developments in applying graph theory that have demonstrated prowess in different ways. George Friedman’s constraint theory has offered some unique and highly valuable ways to gain insights into already-constructed mathematical models. The other two developments have to do with the development of models. One of these is the work of Hartmanis and Stearns[2] which uses the lattice structure (another metamodel, in Friedman’s language) as the primary graphic device, through which they were able to solve a logic design component-minimization problem that had challenged designers for decades. The other is my work which uses the digraph structure (still another metamodel, in Friedman’s language) as the basis for structuring designs, and pointing the way to what kind of mathematical models or metamodels may be needed to analyze a design whose logic structure has been proposed.

These three developments are, at best, loosely coupled. It is my conjecture that a particularly worthwhile effort could be to link Friedman’s constraint theory with the lattice structure. I believe that most of the action in Friedman’s work lies at low levels in the

---

[2] J. Hartmanis and R. E. Stearns (1966): *Algebraic Structure Theory of Sequential Machines*, Englewood Cliffs, NJ: Prentice-Hall.

lattice, and that a few of the concepts would be cognitively easier to bear if envisaged as lattice components; but this remains to be seen. Further effort in all of these domains cries out for the use of creative and flexible wall displays, electronically controlled, for the purpose of supporting human insight.

In a sense the developments in graph theory can be compared to drilling in adjacent, prospective gold-mining areas. Several hardy prospectors have uncovered bonanza-type veins that are open both laterally and at depth. Mining has already begun in these different areas, but the different mines have yet to share much infrastructure (only a few roads, and a few power sources) even though it appears that the various veins are likely to be connectible, and to lead to even richer lodes as the drilling continues. Ironically, it is only when sufficient detail concerning the individual veins has been developed to produce a metamodel that gives adequate insight into the economics of the underground structure that a basis for financing the mining venture proves suitable to the banking community!!

The 1847 De Morgan theory of relations slept for more than a century until far-sighted people like George Friedman and Frank Harary[3] placed it in a practical context. Let us hope that others will pick up this torch and carry it forward–a challenge made easier to accept by the availability of George Friedman’s book.

C:\Documents and Settings\Administrator\My
Documents\FRIEDMANBookAnalysis\BOOK REVIEWof George J.wpd
June 4, 2005 (4:08pm)

---

[3] F. R. Harary; V. Norman; and D. Cartwright (1965): *Structural Models: An Introduction to the Theory of Directed Graphs*, New York: Wiley.