ARTIFICIAL INTELLIGENCE AND ENERGY SYSTEMS: SCOPE AND DEFINITIONS

Enrico Sciubba
University of Rome “La Sapienza,” Italy

Keywords: Artificial Intelligence; Expert Systems, Knowledge-Based Systems.

Contents

Summary

This chapter is an introduction to the field of artificial intelligence (AI). After a brief discussion of the characteristics that make AI useful for engineering applications, a

concise definition of terms and concepts is given. The presentation style has been tailored to provide readers with a general introduction to AI topics, without burdening them with excessive formalism. Since the goal of this topic is to describe engineering applications to thermal design, emphasis on the applicative side has been stressed in this introductory chapter.

1. Introduction: Engineering Design, Knowledge, and Artificial Intelligence

There is a consensus today that engineering design is a highly structured, interdisciplinary, *creative* process, and it is invariably the result of a close cooperation between several individuals (who formally or informally constitute the *design team*). Each person's diverse expertise and specialized skills must be blended together, and result in an engineering project (consisting of drawings, calculations, and all necessary technical documents) about a new solution to the problem that prompted the team into action. The product of a design activity is always an original answer to a technical, economic, social, or marketing problem. Naturally, past experience (both general and specific) plays an important role in the formulation of the solution, and this explains the intrinsic similarities between gas turbine plants, diesel engines, or airplanes manufactured by different producers. However, "design" is semantically linked to "innovation."

In analyzing the design activity as a whole, the concept that first comes to mind is *knowledge*:

- Knowledge about a (potential) specific problem is required to originate the need for a design
- General knowledge of the related engineering fields is needed to initiate the design activity
- Specific knowledge in a particular field is mandatory to participate in a design project
- Mutual communication of knowledge among individual members of a design team is necessary for successful performance
- In turn, the team has to impart this knowledge (or at least a portion of this knowledge) to production engineers to finalize the design (that is, to build a component, a machine, or a plan)
- Similarly, some other form of knowledge must be communicated to instruct operators and final users on how to make proper use of the designed item

Therefore, engineers should consider that a design activity, besides its *quantitative* side, has an equally important *qualitative* side. In addition to performing engineering calculations, it is necessary to be able to communicate to third parties not only the results of these calculations, but also the original problem definition and the method that was used to reach a solution. Finally, to increase the chance of future design improvements, it is more important to leave written records of why and how the design goal was achieved starting from certain premises, than to give an exact description of the numerical calculations which led to the definition of certain design parameters.

In this topic, we will show how to employ artificial intelligence techniques to this purpose. Moreover, we will describe how these techniques can be used to express in qualitative terms the logical steps of an engineering activity, to formalize and systematize a large body of knowledge, and to assist engineers by offering them logical guidance and support in their design endeavors.

1.1 What is Artificial Intelligence?

Artificial Intelligence (for a more precise definition see Section 3) is a cumulative denomination for a large body of techniques that have two general common traits: they are computer methods, and they try to reproduce non-quantitative human thought processes. For design applications, there is a third common trait: problems handled by AI-based techniques are usually ill structured; that is, they are difficult or impossible to tackle with pre-determined solution models.

The applications we will deal with in this chapter are a relatively small subset of general AI techniques: *knowledge based systems*, also called *expert systems*. An expert system (ES) can be roughly described as a specialized AI application aimed at the resolution of a single and well-identified class of problems. Some of the major benefits offered by ESs are as follows:

- They provide an efficient method for encapsulating and storing knowledge so that it becomes an asset for the ES user. For example, after an ES has been built and fielded by a company to assist its design engineers in the choice, design, and technical specifications of a certain component, all future designs will be carried out in a much shorter time, and with a less intensive use of both high-level human resources and computational hardware.

- They can make knowledge more widely available, and help overcome shortages of expertise. So, if an ES is developed by the research and development division of a company to overview the adaptive control of a line of production, it can be applied in all of that company's factories—as long as the same processes remain in use—without needing to dedicate highly qualified human resources to the proper communication of the technology.

- Knowledge stored in an ES is not lost when experts are no longer available. This is, for instance, the case of a numerical process simulator that is developed and implemented in a machine language that then becomes obsolete. If a language-independent ES has been employed in the development of the simulator, then far fewer high-level human resources need to be employed to re-implement the code in a different language or under a different operating system.

- An ES stores knowledge it can gather from experts in some field, even if they are not computer specialists (and even if they are not computer users!). For example, when prospecting for new raw resources, it is often indispensable to gather non-specific knowledge about the nature of the area to be scouted, or about the type and history of possible ores or flows that may have surfaced in the past. When performing this "information gathering" in remote areas, an ES assistant that prompts, collects, and critically analyzes answers by natives can drastically reduce both the time needed to make an educated decision, and the amount of resources invested in the prospecting activities.

- An ES can perform tasks that would be very difficult to perform with more conventional software: thus, it becomes possible to implement features which were impossible or very cumbersome with earlier systems. A case in point is the handling of approximate knowledge that cannot be hard-coded otherwise. For example, while designing a heat exchangers network (see below, and chapter *AI in Process Design*), it is at times convenient to relax the rule "no heat flux across the pinch," and reformulate it as "*some* small measure of heat flux across the pinch is allowed, provided the hot or cold utility is *conveniently* reduced by this measure." This construct would be impossible to translate into any rigidly framed type of computer instruction, but it can be easily managed by an ES that treats the word-concepts "some" and "conveniently" in a *relative* way, just as we mean it when we verbally formulate the relaxed rule.

Other practical advantages associated with the development of a knowledge-based system are:

- rapid prototyping
- explicit knowledge encoding
- possibility of dynamic and efficient further developments
- ease of alteration
- explanations capabilities for development and validation
- potential for rapid inclusion of new rules and techniques

All of these benefits come at a price: high requirements in computer resources, and low speed of computation. The amount of physical memory space required by a knowledge base is very high, because qualitative knowledge is not easily amenable to a simple numeric (binary) representation. Moreover, since qualitative rules are implemented in machine language by complex combinations of elementary logical operations, and the number of operations per clock tick remains constant for a given hardware, the actual execution time of a *rule* is much higher than that of a numerical operation. Furthermore, developing and installing a large ES is a high added-value operation, and therefore a very expensive enterprise: the provisional returns should always be carefully scrutinized before launching an ES project. Two main factors that can make knowledge based systems profitable are:

1. the specialized work load could be—or become—higher than that which can be reasonably performed by the available experts, or some of these experts might not be available in the foreseeable future; and
2. the existing—or foreseen—working conditions could lead to too high a demand on decision-making engineers; this, in turn, may result in what goes under the name of *cognitive overloading*. When cognitive overload sets in, increasing the number of human experts dedicated to the problem is not a proper solution, because communication among experts may deteriorate (too many information bits of very different kinds must be communicated to an increasing number of people).

Before proceeding any further, it is useful to present here our confutation of some of the more common misconceptions about AI in general and ESs in particular.

1. *It is not true that ESs are suitable for every type of design problem*. For instance, a very large class of structural, thermal, and fluid dynamic problems can be tackled very effectively with "exact" (and fully deterministic) algorithmic techniques, and rather poorly with an ES. On the other hand, it is not correct to describe an ES as "an alternative to conventional computer programming": in the majority of practical design cases, the fields of application display a substantial overlap.
2. *It is not true that AI users must be conversant with AI programming languages*. Neither is it true that creators of ESs must be familiar with high-level languages like PROLOG, LISP, and the like: actually, often all it takes to develop an effective ES is a sound knowledge of the principles of predicate logic, complemented by specific domain knowledge.
3. However sophisticated and knowledgeable in their field of expertise, *ES users are logically, functionally, and mentally distinct from knowledge engineers*. Similarly, a domain expert is a substantially different professional figure to a knowledge engineer (see Sections 3.15 and 3.16 below).
4. In turn, *not every design or process engineer, or technical operator, is a domain expert*: in reality, there is a remarkable scarcity of reliable and actually knowledgeable domain experts.
5. On the other hand, *an ES should not be tested "against the domain expert."* That is, its performance ought not be compared with that of the human expert in a similar situation.
6. *Complete modularity is rarely achieved in logically complex ESs*. Even in "small" ESs it is not possible to simply add to (or delete from) the knowledge base groups of rules and objects (or even a single rule or a single object), and be certain *a priori* that the solution or the inference process are not affected.

This chapter, as well as this entire topic, is a rather introductory, descriptive compilation of AI concepts and possible applications. We do not anticipate (and the reader should not expect) that upon completing the study of the topics contained here, non-AI-specialist engineers will be able to write their own AI application. To use a term that will be frequently encountered in the following pages, *much more specific knowledge* is needed to be able to write successful applications. In particular, notions of formal and symbolic logic, of predicate and propositional calculus, and of specific relational or object-oriented languages are needed. Our aim is to create instead informed and competent users. Readers interested in specialized AI monographs should consult the bibliography provided here.

1.2 AI is No Intelligence!

What is intelligence? Alternatively, putting the question in a form more suitable to the paradigms of propositional calculus, what are the attributes a (not necessarily human) being must possess to qualify as "intelligent"? There is considerable debate among specialists about an exact definition, but one that would be accepted by the majority of modern cognitive psychologists is the following.

An intelligent individual is one who:

1. reacts in a flexible way to different situations;
2. is able to take advantage of fortuitous circumstances;
3. can correctly assess the objective relative importance of the different elements of a newly encountered situation;
4. can solve with original (that is, hitherto personally unknown) means a problem not previously experienced;
5. can correctly understand thinking patterns different from the individual's own, and in particular, is capable of recognizing oblique and/or deceptive reasoning;
6. is open-minded (in other words, is ready to accept the possibility of being wrong both in causal and in "intuitive" thinking); and
7. can produce new and innovative ideas.

One does not need to be a computer expert, a logician, or an AI practitioner to see that, at least in the way we conceive computers and computer languages today, there cannot be an "intelligent android": however complex its hardware and software, it (the android) will in all likeliness lack (at least) qualities (d) and (g) in the list above.

(We must mention here that some AI researchers would strongly disagree with this last statement. In fact, there are examples of AI programs that have "discovered" new (albeit admittedly secondary) "laws of nature," and it would be difficult even for an expert to see where Deep Blue (the Chess Computer Master by IBM) lacks "intuition" in its chess moves. But the point we want to make here is somewhat less general, and limited to the realm of engineering design: within this realm, it is (yet?) unconceivable that an AI program "discovers" Fourier's law, or "invents" the Carnot cycle.)

In common language, we attribute the quality of "intelligence" both to individuals who must think because they do not know much, and to those who know much, and therefore can afford to think less. In the terms of AI, this can be rephrased as follows: we tend to attach the attribute "intelligent" both to individuals who possess a small amount of domain knowledge, and must therefore exercise a large quantity of inference to draw their conclusions, and to individuals who possess a large amount of domain knowledge, and can therefore reach the same conclusions by performing a rather limited inferential activity. Notice that in both definitions we have implicitly assumed that an intelligent reaction is neither spontaneous nor necessarily dictated by the available data. This point is important for two reasons. First, it makes a clear distinction between *potential* intelligence, intended as a "memorization of collected knowledge," and *dynamic* intelligence, which is at once the act and the result of "reflecting" upon the collected knowledge. Second, it reminds us that the concept of "thinking" reported in this topic refers to human thinking modes somehow hard- or soft-coded into a machine. A one year old human (or, for that matter, a one month old cat!) can "think" ("infer") and "learn" ("increase the amount of knowledge it has command of") much faster and more effectively than today's most advanced AI programs. On the other hand, AI methods produce codes that perform, with almost incredible reliability and speed, cognitive (qualitative) tasks that mimic very closely some traits of human thinking patterns. If we agree with the idea that potential and dynamic intelligence are operationally equivalent (in the sense that they can reach the same results), then clearly there is no limit to the "IQ" of an AI code: to increase the IQ just requires storing more (properly constructed and connected) knowledge in its database. In this sense, computers *are* intelligent, even by today’s standards.

2. Definitions of Concepts and Terms

2.1 Artificial Intelligence (AI)

There is no universally accepted definition of AI. We shall adopt the following one:

AI is that part of computer science that investigates symbolic, non-algorithmic reasoning processes, and the representation of symbolic knowledge for use in machine inference.

This is a modern and complete definition, which contains more than one point of interest for following this topic. Firstly, it stresses the symbolic character of the object AI. Though its meaning is obvious, its implications are not: AI’s field of interest does not contain numbers and operations on them, but logical objects and their mutual relationships. Secondly, it introduces the concept of non-algorithmic reasoning, which is usually very difficult to grasp for an engineer: a solution to a problem is not obtained by a predefined series of operations, but rather by a series of logical inferences, consisting of an *a priori* unknown number of steps, and whose outcome is in no way predictable in its entirety. Thirdly, it correctly states that the scope of AI is to generate a process of machine-implemented inference (that is, to construct a computer code which can infer a *cause* from an *effect*, possibly with the help of some additional (logical) constraints).

Historically, the first reference to the concept of artificial intelligence is generally credited to McCarthy, but his original definition was too broad for his times, and gave origin to a number of misinterpretations and misconceptions which still burden this branch of computer science: *AI is that compendium of computer techniques dedicated to the implementation of a simulating procedure for any mental process.*

Today, this definition—ambitious as it may seem—is being regarded with growing attention as our studies in the cognitive sciences (psychology and bioneurology) gain in depth and breadth. While many of the current schemes to explain mental artifacts like memory, selective memory, learning, and understanding are only crude approximations or primitive representations, we are now capable of constructing an approximate model of the complex chains of biochemical phenomena taking place in the brain. For the time being, though, the idea of “simulating any mental process” has been replaced by the following assumptions:

1. “intelligence” can be explained and represented as a symbol-manipulating activity;
2. this activity can be embodied in a physical symbol system; in particular, it can be both *described* and *implemented* in such a system;
3. the symbol manipulation necessary to the description and to the implementation of any “intelligent activity” can be carried out on digital computers;
4. various aspects of human “intelligence” (in particular, what is usually called “logical reasoning” in common language) can be modeled by such physical symbol systems; and
5. there may be a “general theory of intelligence” that may originate a “general symbol

> system," which in turn might result in a "universal computer code" capable of describing all "intelligent phenomena": however, even if its discovery should be a long range goal of AI research, it should not be sought to the detriment of "partial" or logically "local" symbols systems (which result in "specific" applications).

(This seemingly "technical" dispute (about the possibility of simulating "thought") has become a philosophical issue between those who negate the possibility of "strong AI" and underline the uniqueness of the human mental activity, and those who advocate a possible future essay implementation of a "universal virtual simulator," undistinguishable from a human brain. The topic obviously exceeds the limits of this essay.)

Recently, another purpose for AI has been put forth: as noted by D.A. Mechner, in spite of some humbling failures that helped bring things back into a more down-to-earth perspective, AI methods and devices can "serve as windows on the mind," (that is, help develop a scientifically pragmatic understanding of the very concept of "intelligence").

-
-
-

Bibliography

Brooks R. A. (1991). Intelligence without representation. *Artificial Intelligence* **47**(1). [This is a rather specialized essay on methods and techniques of non-algorithmic reasoning.]

Charniak E., McDemmott D. (1985). *Introduction to Artificial Intelligence*. Addison-Wesley. [An early, but complete and well thought out textbook, on AI topics. Some of the definitions it gives are dated, but it may be very useful to observe their developments in today's concepts.]

Deutsch D. (1997). *The Fabric of Reality*. Allen Lane. [This book was written for a large audience, and therefore it shows some contamination between scientific precision and imaginative scientific fiction. Nevertheless, the portion that deals with the representation of reality is very important for a correct understanding of the concept of qualitative modeling.]

Freeman J. A. and Skapura D. M. (1992). *Neural Networks*. Addison-Wesley. [A precise, complete, and self-contained specialized textbook.]

Garland W. M. J. (1996). The role of knowledge-based systems in heat exchanger selection, design and operation. *New Developments in Heat Exchangers* (ed. N. Afgan et al.). Gordon & Breach. [An interesting example of a direct application of AI techniques to an apparently "trivial" engineering problem. The specific results may be outdated at present, but the method is still valid.]

Giarratano J. and Riley G. (1989). *Expert Systems: Principles and Programming*. Boston, PWS Publ. Co. [A very useful guide to the practical development and implementation of expert systems. It is outdated as far as the programming language is concerned (at the time of writing, most AI programmers would use C++), but is very interesting for its insistence on a conceptually and formally structured approach.]

Gillies D. (1996). *Artificial Intelligence and Scientific Method*. [A confutation of the accusation that AI methods are 2a-scientific, because of their insistence on the importance of the qualitative side of a scientific paradigm.]

Gong M. and Wall G. (1997). *On Exergetics, Economics and Optimization of Technical Processes to meet Environmental Conditions*. (Proc. TAIES'97, World Publishing Corp., Beijing 1997). [Though written without explicit reference to AI methods, this is an interesting example of how engineering paradigms can be posed in a formally qualitative fashion. An interesting example of a "new" engineering approach.]

Gregory R. (1994). Visual intelligence. *What is Intelligence?* (ed. J. Khalfa). Cambridge, UK: Cambridge University Press. [By examining the logical structure of human vision, the author derives many fundamental conclusions about the act of "intelligent understanding."]

Hopfield J. J. (1982). Neural networks and physical systems with emergent collective computational capabilities. *Proceedings of the National Academy of Sciences* **79**. [A specialized paper describing the idea that an increase in the complexity of a system may bring about the appearance ("emersion") of discontinuous changes in the properties of the system.]

Ignizio J. P. (1991). *Introduction to Expert Systems*. McGraw Hill. [A complete and well written textbook on expert systems. Some of the definitions it gives are still in use today.]

Kowalski R. A. (1979). *Logic for Problem Solving*. North Holland. [An early, formal approach to the qualitative solution of practical engineering problems. The author was one of the earliest advocates of object-oriented programming.]

Liebowitz J. and De Salvo D. A., eds. (1989). *Structuring Expert Systems*. Yourdon Press. [A specialized essay with a strong accent on the necessity of using an intrinsically structured knowledge gathering approach to construct expert systems.]

Lloyd J. W., ed. (1990). *Computational Logic*. Springer Verlag Basic Res. Series. [A collection of articles on the transposition of relational logic rules into computer instructions.]

McBride R. D., O'Leary D. E. (1993). The use of mathematical programming with AI and ES. *European Journal of Oper. Res*. **70**(1). [An early precursor of MATHEMATICA, MAPLE, MATLAB, and the like. Strong emphasis was given to the possibility of adopting known mathematical/logical constructs to perform qualitative computations.]

McClelland J. and Rumelhart D. (1986). *Parallel Distributed Processing*, Vol.1–2. MIT Press. [An attempt to investigate the possibility of translating into cognitive science the mathematical properties of parallel computing.]

McCulloch W. S. and Pitts W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, **5**. [A pioneering work describing an attempt to formalize elementary nervous activity. Vastly outdated today, but strikingly innovative for its times.]

Mechner D. A. (1998). All systems GO. *The Sciences*. New York Academy of Sciences, January/February 1998. [A layman's exposition of some applications of AI techniques, together with extensive and useful comments and comparisons.]

Negoita C. V. (1993). *Expert Systems and Fuzzy Systems*. Benjamin Cummings Pub. [A complete and well thought out textbook on two AI topics that the author considers separately. More recently, fuzzy sets have been developed into autonomous techniques that can be incorporated in expert systems as a part of their inferential engine.]

Papalambros P. Y. and Wilde D. J. (1988). *Principles of Optimal Design: Modeling and Computation*. Cambridge, UK: Cambridge University Press. [An attempt to formalize "optimal design."]

Parsaye K. and Chignell M. (1988). *Expert Systems for Experts.* John Wiley & Sons, Inc. [A rather specialized book on expert systems. It contains a wealth of qualitative procedural information.]

Press W. C., Teukolsky S. A., Vetterling W. T., and Flannery B. P. (1985). *Numerical Recipes in C.* McGraw Hill. [A sort of "cookbook" of scientific computational routines. In this re-edition of an earlier FORTRAN version, even heavier accent is placed on the necessity of implementing intrinsically structured programming techniques. Though the authors never mention the words "object-oriented programming," this textbook may be seen as marking the borderline between deterministic and O-O programming techniques.]

Quine W. V. O. (1960). *Word and Object*. MIT Press. [An exceptionally deep and complete analytical critique of the concepts of cognitive science. The language (and its intelligence) is seen as the frame of reality.]

Rasmussen J. (1986). *Information Processing and Human-Machine Interaction: An Approach to Cognitive Engineering.* North Holland Ser. in Syst. Sci. and Eng. [An early description of the "cognitive" approach to solving engineering problems. Very limited emphasis on real design applications: this book is dedicated to the investigation (fundamental at that time) of the guiding principles of knowledge gathering and processing.]

Rettaroli M. and Sciubba E. (1994). MASAI: a code for the symbolic calculation of thermal plants. *ASME PD/64/3*. (Proc. ESDA Conf., London, 1994). [An attempt to substitute formal symbolic procedures to numerical solvers. Still, only entirely deterministic procedures are proposed.]

Rich E. (1983). *Artificial Intelligence*. McGraw-Hill. [A basic, older textbook on the fundamentals of artificial intelligence concepts and applications. Very few examples of applications to engineering design: in the 1980s there was still a lack of first-level applications, and the accent was being put on conceptual topics.]

Sciubba E. (1998). Toward automatic process simulators: Part II. An expert system for process synthesis. *J. Eng. for GT and Power* **120**(1). [A formal description and detailed explanation of how a process synthesizer works. The discussed applications are limited to fossil-fuelled power plants.]

Sciubba E. and Melli R. (1998). *Artificial Intelligence in Thermal Systems Design: Concepts and Applications*. New York: NOVA SCIENCE Publishers. [An introductory but complete and well written textbook on AI topics. A very strong bias towards applications results in a compression of the most theoretical topics, which are referred to but seldom thoroughly discussed. A textbook for AI users.]

Scott A. (1996). The hierarchical emergence of consciousness. *Math. & Comp. in Simul*. **40**. [A reflection on the emergence of a more complex, hyperlogical structure from a deeper analysis of living and non-living systems.]

Shank R. and Birnbaum L. (1994). Augmenting Intelligence. *What is Intelligence?* (ed. J. Khalfa). Cambridge, UK: Cambridge University Press. [Techniques to increase induction and deduction capabilities. The article contains some strong critique to the current misconceptions about "intelligence" and its representation.]

Shannon C. E., McCarthy J., and Ross A. W. (1956). Automata studies. *Annals of Mathematic Studies* **34**. Princeton University Press. [Early reflections on the implications of Turing's studies on "intelligent machines."]

Shoham Y. and Moses Y. (1989). *Belief as defeasible knowledge.* Proc. IJCAI-89, 1989. [An interesting essay on the possibility of incorporating an apparently vague and ambiguous statement of belief into a knowledge manipulating engine that may draw proper inferences from it.]

Sriram R. D. (1997). *Intelligent Systems for Engineering*. Springer Verlag. [A complete, well-written, well-balanced textbook on AI. For the quantity and quality of information contained, it constitutes a necessary reference for workers in this area.]

Widman L. E., Loparo K. A., and Nielsen N. R. (1989). *Artificial Intelligence, Simulation and Modeling*. J. Wiley. [A predecessor of Sriram's book. The emphasis here is more on applications, with special accent on the relationship between an "object" and its "model."]

Wittgenstein L. (1996). *Philosophie*. Rome: Donzelli Pub. (in Italian and German). [A collection of the philosopher's reflections on the relations between language, mind, and "reality." With extraordinary foresight, some of the issues raised today by knowledge gathering and representation methods are clearly stated and "discussed" in the peculiar style of Wittgenstein's latest works.]

Biographical Sketch

Enrico Sciubba is a professor at the Department of Mechanical and Aeronautical Engineering of the University of Rome 1 "La Sapienza" (UDR1), Rome, Italy. He received a master's degree in mechanical engineering from UDR1 in 1972. From 1972 to 1973, he was a research assistant at the Chair of Turbomachinery in the same university. From 1973 to 1975 he worked as research engineer in the research and development division of BMW, Munich, Germany, where his tasks included the design, development, and testing of advanced i.c. engines. After returning to UDR1 as a senior research assistant from 1975 to 1978, he enrolled in the Graduate School of Mechanical Engineering, majoring in thermal sciences, at the Rutgers University, New Brunswick, NJ, USA, where he was granted his Ph.D. in 1981. From 1981 to 1985 he was assistant professor at the Catholic University in Washington D.C., USA, teaching thermal sciences. He returned to the Department of Mechanical and Aeronautical Engineering of UDR1 as a faculty member in 1986. He lectures on turbomachinery and energy systems design, at both undergraduate and graduate level. His research activities are equally divided in three main fields: Turbomachinery design and CFD applications; Energy systems simulation and design; Applications of AI-related techniques and procedures to the design, synthesis, and optimisation of complex energy systems. His publications include more than thirty journal articles (mostly in international refereed journals in the field of energy and applied thermodynamics), and over eighty refereed papers at international conferences. He published one book on AI applications for NOVA Science, USA, and is writing a turbomachinery book for J. Wiley& Sons. Dr. Sciubba is associate editor for three major international journals in the field of energy conversion, and is a reviewer for several more.