IEEE SysCon 2009 —3rd Annual IEEE International Systems Conference, 2009
Vancouver, Canada, March 23–26, 2009

Multi-objective Stochastic Heuristic Methodology for Tradespace Exploration of a Network Centric System of Systems

Atmika Singh
Engineering Management and Systems Engineering
Missouri University of Science & Technology
Rolla, USA
asfff@mst.edu

Cihan H. Dagli
Engineering Management and Systems Engineering
Missouri University of Science & Technology
Rolla, USA
dagli@mst.edu

***Abstract*—System of systems (SoS) architecting techniques rely on traditional, static tools that were designed for classical stove-piped systems. There is a need for tools that can capture the complex adaptive nature of such SoS. An architecture search methodology using genetic algorithms and a fuzzy assessor was applied to the conceptual architecture design of a generic smart grid and a set of architectures with high fitness was obtained. This set of architectures is intended to serve as a starting point for a systems architect to ultimately be able to converge on the best system design. The SoS architecting process has largely remained heuristic in nature and there exists a need for quantitative and analytical models. The research presented in this paper provides a starting point for a mathematical basis to the SOS architecting process.**

Keywords-Net-centric; Conceptual Architecture; Computational Intelligence

I. INTRODUCTION

Conceptual design of network centric system of systems (SoS) is an extremely challenging problem exacerbated by the dynamic and continuously evolving nature of SoS and their component systems. Challenges in the SoS architecting process have been well documented in [1] and [2]. The emergent behaviors of a SoS cannot be captured and evaluated by the classical systems engineering process alone. Net-centricity or the ability of systems to interact with each other and evolve based on these interactions, is a key component of any modern SoS. Net-centric SoS dynamically change the organizational scope at run time unlike the static stove-piped systems designed using the traditional architecting tools. This open-ended communication within the SoS is a source of emergent behavior which makes it necessary for the SoS architecture to be flexible and extensible. For such systems operational efficiency takes a back seat to the ability to adapt to environmental changes and continue performing [2].

In summary, Net-centric SoS architecting problems involve dynamic open-ended systems which require the development of an entirely new set of system architecting tools that can capture their complex adaptive nature. Current systems of systems design processes follow an unstructured approach. Architectures are generated either by modifying legacy architectures or by following a few standard options that have been successfully implemented. Such a process does not support the top down 'big picture' focus needed for architecting these large scale SoS. Thus, SoS architecting frameworks are geared towards achieving feasibility instead of optimality [3]. This approach does not take into consideration all possible permutations and combinations of possible architectures. Thus many acceptable system architectures that satisfy performance and cost criteria remain overlooked. The research presented in this paper applies stochastic optimization techniques like multi-objective genetic algorithms to search the design tradespace of a SoS and generate a population of near optimal designs. The proposed methodology is intended to be used during the conceptual design phase to allow designers to search through the SoS design alternatives early in the architecting process, evaluate alternative architectures and develop a population of good solutions. This set of good solutions can then be used as inputs to the detailed design phase of systems engineering to ultimately converge on the best design.

SoS design problems are combinatorial in nature and have discrete variables in nonlinear problems with multi criteria objective functions [3]. Component systems have multiple intra and inter system trade-offs that cannot be fitted into the mold of a single objective and multiple constraint problem. Stochastic heuristic techniques like genetic algorithms and evolutionary computation will be used for design optimization because they work best for combinatorial problems with non-convex trade spaces [4].

Previous attempts to apply stochastic optimization to systems architecting generally included optimization of the design of individual components and then integrating them. However optimized subsystems may not necessarily combine to form an optimum system. Current SoS architecting frameworks dwell on descriptive techniques and heuristics without providing a mathematical basis to the design methodology [5]. The goal of this work is to provide a quantitative tool to converge on to the optimum SoS conceptual architecture early in the design life cycle.

The next section discusses the process of generating the conceptual architecture followed by a description of the

9781-4244-3463-3/09/$25.00 ©2009 IEEE

architecture search methodology including problem representation and architecture assessment. The methodology is then used to develop architecture alternatives for a generic smart power grid. A discussion of the results and future work concludes the paper.

II. Application of the Architecture Search

The SoS engineering process works by analyzing a system of systems at increasing levels of detail. It is a top down approach that begins with a high-level functional vision and proceeds by functionally breaking down the system to the component level. The early stages of design begin with requirements specification using stakeholder inputs and use-cases designed for the system. The systems architects identify the functional and business requirements. This is followed by the identification of technology and mechanisms that conform to the functional requirements and specifications. This process results in a series of engineering decisions which yield a generic solution that is platform-independent. This preferred architecture is known as the Conceptual Architecture. This conceptual architecture serves as input to the vendor evaluations, business case estimates, integration and testing plans, and provides a means to accelerate detailed design and engineering after trade studies have been completed to identify the exact technologies that will be used [6].

A. *Conceptual Design*

The Conceptual Architecture identifies which high-level systems, subsystems and components will form the SoS in order to fulfill the requirements specification. The inputs to the conceptual architecture design process include lists of actors, messages exchanged and data transferred. Using these inputs a high-level architecture is defined where all the systems, subsystems and components are characterized in a platform independent fashion. This allows the trade study process to evaluate all the technology options against a standard set of requirements. The Conceptual Architecture provides answers to the 'what' questions associated with the system design. It provides the design team with the necessary technical information to continue with detailed design. Fig. 1 shows the major steps that precede the conceptual architecture design process. The Conceptual Architecture determines how the system components are integrated into a logical architecture and which services must be provided to create a viable solution.

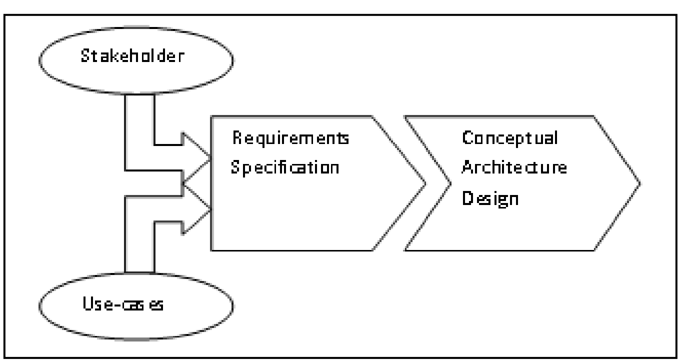

Figure 1. Conceptual Architecture Design Process

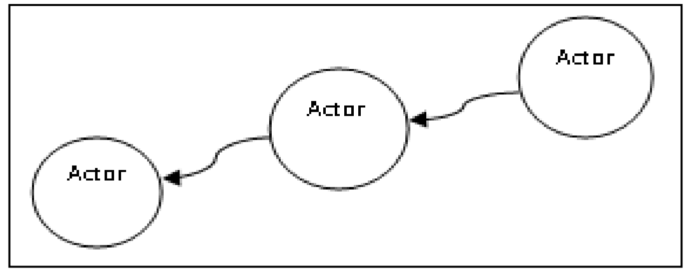

Figure 2. Actor Interface Diagram

The purpose of the interface diagram as shown in fig. 2 is to provide a single conceptual picture that can be used to express the flow and sequence of data within the system. The interface diagram can be derived from the interactions among the actors indicated by the activity diagram. A preliminary interface diagram can be created during the use case workshops by the architecture team to provide a high level conceptual view.

The proposed methodology is intended to be used during the conceptual design phase. It will use the actor list and actor interface diagrams generated during this phase to evaluate multiple architecture alternatives on the basis of customer or stakeholder preferences. At the higher levels the generated architectures will represent the actor/subsystems and the interfaces between them along with the services that will have to be associated with those interfaces. At the lower levels the methodology can combine the architecture search with the trade study function. The architectures will represent the components/services, their interfaces and the technology alternatives available. A combined evaluation can be performed to generate architectures whose fitness evaluation includes the technologies that will be used for implementing it.

III. Architecture Search Methodology

The architecture search problem will be formulated as an optimization problem for a genetic algorithm. The design vector will form the chromosomes for the genetic algorithm. The fitness of the architecture populations will be evaluated using a fuzzy assessor. Fig. 3 shows the general flow of the architecture search technique. The architectures to be generated are presented to the genetic algorithm which generates a population of solutions which are then assessed using a fuzzy assessor.

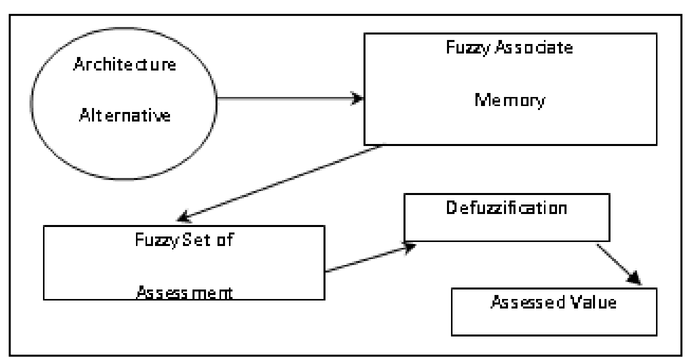

Figure 3. Architecture Search Methodology Block Diagram

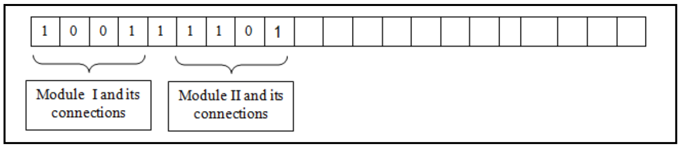

Figure 4. General chromosome structure

A. Architecture Representation

Fig. 4 shows a general chromosome structure used to represent an architecture solution. The architecture solution was coded as a binary string, where each bit represents the presence or absence of an interface or service.

Bit mutation operation is incorporated in the algorithm to aid in forestalling the problems of premature convergence associated with the repeated use of crossover. A binary tournament selection procedure was employed to select the chromosome for crossover. A tournament size of 2 was used. The selected chromosome in the population was crossed over with a randomly selected second chromosome. Crossover was performed with a fixed probability. Crossover was double, as each crossover produced two offspring. Mutation was performed with a low fixed probability at a randomly selected location. The elitism operator was active for the crossover and mutation. This means that the chromosome with the highest fitness in a generation was not crossed-over or mutated. The child population was ranked on the basis of fitness and the best chromosomes were chosen to form part of the next generation.

B. Architecture Assessment

Most real life decision problems are constrained by more than one objective. A typical multi-objective decision problem involves the selection of one alternative from a universe of alternatives. Each alternative is evaluated on a set of objectives important to the decision-maker based on how well it satisfies each objective. The decision-maker specifies objectives that impact the decision. For architecture assessment a Fuzzy Inference System (FIS) is used. The FIS evaluates the architecture based on its score for a set of key architectural attributes specified by the decision-maker. The inputs to the FIS are calculated using a set of definitions for the key attributes. These definitions are used to implement the constraints imposed on the architecture. Table 1 shows a list of sample key attributes. Membership functions for a few selected fuzzy performance attributes are detailed in the table 2. Fig. 5 shows membership function definitions for 2 of these attributes. Each attribute is represented by 3 membership functions. For most attributes the membership functions that will have to be defined are easily identifiable. For other attributes they can be determined by knowledge acquisition from an expert or group of experts.

TABLE I. LIST OF SOME KEY SYSTEM ATTRIBUTES

Flexibility	Reliability	Managebility
Survivability	Availability	Open-endedness
Security	Maintainability	Changeability
Affordability	Complexity	Controllability
Robustness	Adaptability	Stability

The membership function definitions were determined manually using decision-maker preferences. Once the membership function sets are chosen the associated membership functions shapes need to be defined. The membership functions shapes are domain dependent. The scale and shape of the membership functions will change based on the problem domain. For example, for the affordability functions, in case of a commercial project, the bounds will be much narrower and rigid. In such a case we may want to use a triangular or trapezoidal function shape. In a technology demonstration project like the moon landing, the affordability requirements are much lighter and in this case the shape of the Highly Affordable membership function will be much broader while the other two will be narrow. The approach adopted for acquiring the shape of any particular membership function is dependent on the attribute. Linear membership function shapes such as triangular are useful for measurable attributes. In other cases linear membership functions are not appropriate as they do not represent accurately the linguistic terms being modeled and so will have to be elicited directly from the expert, by a statistical approach or by automatic generation of the shapes.

The rules for evaluating the membership functions will also be problem specific. For, example if affordability is desirable above all other attributes except security then this can be built into the fuzzy inference rules. The table below shows some examples of the rules used for determining the FIS output.

IV. APPLICATION: SMART GRID

The need for energy independence, power conservation, and the customer need for high quality power are dramatically altering the business model for the electric power industry. The Smart Grid is the integration of the power infrastructure with the communication infrastructure to create an intelligent grid that can detect and address emerging problems before they impact service. The grid will respond to local and system-wide inputs and make protective relaying the last line of defense, not the only defense in case of cascading failures. This new and smarter grid will incorporate data inputs from sensors and smart meters, two-way data communications, and self diagnosing and healing abilities to respond to disturbances and to bring the system back to stability. It will automatically adapt to a changing situation. It will enable loads and distributed resources to participate in operations and will be designed and operated with reliability and security as key attributes [6]. Key functions of the grid as defined by [8] are,

- Self Heals (detects, analyzes, responds, restores)
- Motivates and includes the Consumer
- Resists Attack
- Provides Power Quality for 21st Century Needs
- Accommodates All Generation and Storage Options
- Optimizes Assets and Operates Efficiently

The IntelliGrid^{SM} Architecture was developed by the EPRI in an effort to create an industry level architecture standard to assist utilities in developing their own smart grids.

The smart grid is a large and complex net-centric system. This paper has used the guidelines and deliverables listed in [6], [7], [8] and [9] to formulate a generic smart grid example for implementing the architecture search methodology. The methodology was implemented on the high-level service architecture of a smart grid. The architecture was assessed based on 5 key success criteria that were identified with expert input. The problem representation and the assessment criteria are discussed in further detail in the next sections.

A. High-level Conceptual Architecture Representation

The main participants and function of the conceptual architecture are determined by the systems engineering process during the conceptual design phase. Table 3 lists the actors and participants that were used to identify the services or systems used for the purpose of this study. From these actors and functions 11 services or systems were identified to form the conceptual architecture. Each of these systems performs a key service responsible for the functioning of a smart grid. For the sake of simplicity the systems have been designated by labels from $m1$ through $m11$ as shown in table 3. The 11 systems were then grouped into 3 layers based on their operational focus. The groupings are listed in the table 2.

TABLE II. SYSTEM GROUPINGS BY LAYER

Layer 1	M1, M2, M3, M4, M5
Layer 2	M6, M7, M8
Layer 3	M9, M10, M11, M12

The interfaces between these systems are shown in fig. 5. The figure shows all possible interfaces. The optimal interfaces will be chosen by the architecture search algorithm. For the sake of simplicity, few constraints were applied to the presence or absence of interfaces between two systems. It is assumed that any system can be interfaced only with other systems in the layer immediately above it.

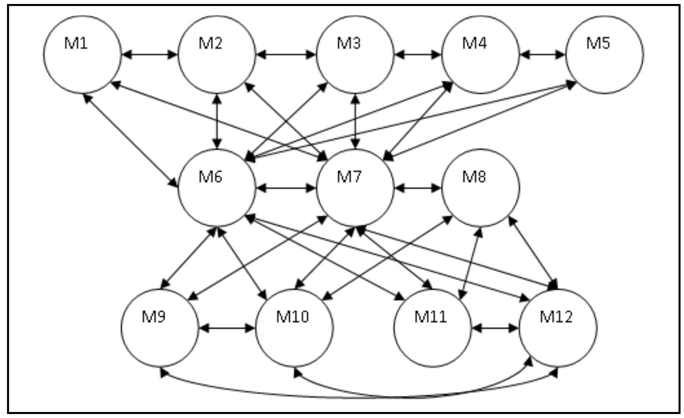

Figure 5. System interface diagram showing all possible interfaces connected.

The input to the genetic algorithm consists of a 58 bit chromosome. The digits in the chromosome are binary and indicate the presence or absence of a connection between the systems concerned. Each system is assigned a set of bits in the chromosome representing its allowed connections. Fig. 3 shows the interconnected systems, each with its maximum allowable connections. For example, the system M1 is allowed to connect to the systems M2, M7 and M6 so it is represented in the chromosome by 3 bits. The system M7 being the central control system should ideally connect to all the other systems and therefore will be represented by 11 bits in the chromosome. The bits for the other systems can be similarly derived.

TABLE III. SMART GRID ACTORS , FUNCTIONS AND SERVICES.

Actors/equipment	Smart Grid Functions	Systems/Services	Label
Power generators	Distributed generation resource management	Distributed resource management system	M2
Transmission operators		Distribution management and control system	M7
Demand response banks	Power storage		
Distribution control centers	Voltages support	Automated feeder management system	M3
Transmission substations	Reactive support		
Flexible AC transmission systems	Load-following support	Voltage and reactive support system	M5
Distribution substations	Power quality management		
Load serving entities	Distribution control and monitoring		
Distributed resource		Distributed substation management system	M4
Power storage device			
Market participants	Two-way data acquisition and communication	Smart meter operations system	M9
Industrial consumers		Meter and sensor data management system	M11
Residential consumers	Data storage		
Commercial consumers	Data analysis/forecasting	Data storage system	M10
Front and back offices	Demand side management	Data analysis and forecasting system	M8
Power safety and quality equipment	Equipment monitoring and maintenance	Equipment monitoring system	M6
		Equipment maintenance system	M1
Wired network	Network maintenance	Communication network maintenance system	M12
Wireless network			

TABLE IV. RULES FOR CALCULATING ATTRIBUTE VALUES

Attributes	***Rules***
Affordability	$\frac{100}{\text{total no. of connections}} \sum \text{cost weights}$
Complexity	$\frac{100}{\text{total no. of connections}} \sum \text{connection weights}$
Flexibility	1-Complexity
Maintainabili -ty	$\frac{100}{\text{total no. of connections}} \sum \text{connection weights to M6}$
Reliability	$\frac{100}{\text{total no. of connections}} \sum \text{connection weights to M7}$

TABLE V. RULES FOR COMBINING FUZZY OUTPUT

Rules for combining fuzzy output
If maintainability is low or affordability is low or flexibility is low or complexity is high then performance is unacceptable
If reliability is low and maintainability is low then performance is unacceptable
If flexibility is average and affordability is average then performance is tolerable
If complexity is average then performance is tolerable
If flexibility is high then performance is desirable
If reliability is high and affordability is high then performance is desirable
If flexibility is high and affordability is high then performance is desirable
If maintainability is high and flexibility and reliability is high then performance is desirable

B. *Architecture Assessment.*

The fitness of the generated architectures is calculated by a Fuzzy Inference System. The FIS calculates the fitness on the basis of 5 key attributes. These attributes and the rules for their calculation are shown in table IV. Each attribute is defined by three fuzzy sets and their corresponding membership functions. The membership functions and their shapes of the fuzzy set for the attribute *affordability* can be seen in fig. 6. Other attributes were similarly defined. The shapes were chosen on the basis of expert input and were modified based on experimentation. The outputs from the individual membership functions were then combined using a set of fuzzy rules which are listed in table V. The output from the fuzzy rules was then input to a final fuzzy attribute called fitness. The value obtained from the fitness attribute was used as a fitness for the genetic algorithm. The genetic algorithm ranks and selects architectures based on this final fitness value.

C. *Genetic Algorithm Implementation.*

The genetic algorithm was implemented with an initial population size of 50. The bitwise mutation and double crossover operations were used. Chromosomes for the mutation and crossover were selected by using binary tournament selection. Each chromosome had a length of 32 bits. This is the total number of connections in the interface diagram shown in fig. 5. The algorithm was run for a 100 generations and a mutation rate of 0.01 was used. The crossover rate was set to 0.80. Elitism was enabled to allow the architecture with the highest fitness to be preserved from generation to generation.

D. *Results*

The results from the architecture search are shown in table IV. The algorithm provided a population of architectures with assessment values over a maximum of 100 points. The architecture with the highest assessment value is shown in fig. 7. The architecture shown in fig. 7 had a fitness value of 63.3. Fig. 7 shows the highest fitness achieved for successive generations. The architecture generated can allow systems architects to study the impact of the attributes collectively and also individually. The systems can also be packed on the basis of the connections in the resulting architectures. A proliferation of connections can be inferred as a single subsystem that needs extensive communication capabilities and should be packaged together. Fewer connections can delineate the interface boundaries between 2 systems.

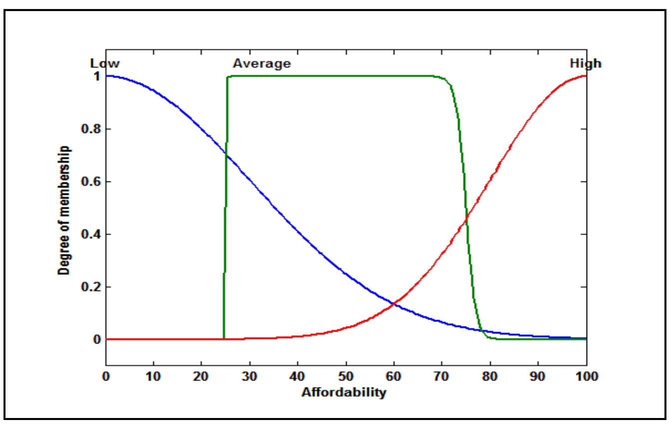

Figure 6. Membership function definitions for the affordability attribute.

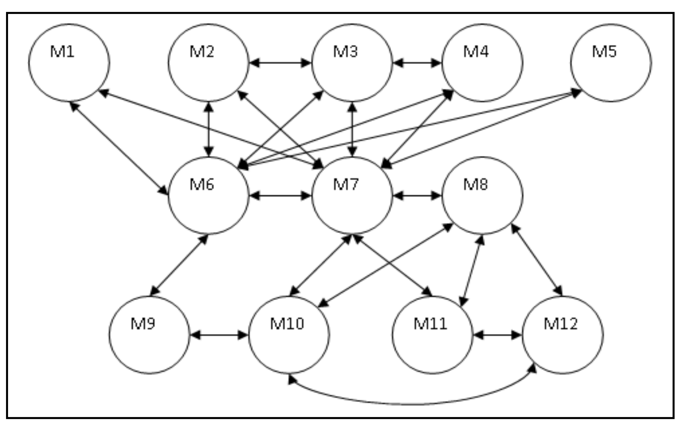

Figure 7. Architecture with highest fitness value.

V. CONCLUSIONS

An architecture search methodology was applied to a generic smart grid and a set of architectures with high fitness was obtained. The approach was an initial exercise that needs to further developed. However, it was demonstrated that stochastic heuristic techniques can assist in the systems architecting process by providing the systems architects with a set of feasible designs that can be developed into an optimal architecture. Further extensions of the proposed methodology will include the use of graph theory and constraint theory to evaluate model consistency. Decision theory will be needed to converge on the optimal designs that balance cost, performance and risk as required by the stakeholders. The SoS architecting process has largely remained heuristic in nature and there exists a need for quantitative and analytical models. The research presented here contributes greatly by providing a mathematical basis to SOS architecting.

REFERENCES

[1] M. W. Maier, "Research Challenges of System-of-Systems," IEEE International Conference on Systems Man and Cybernetics,Vol.4, 10-12 Oct. 2005, pp. 3149-3154.

[2] A. Meilich, "System of Systems Engineering & Architecture Challenges in a Net Centric Environment," Proceedings of the IEEE/SMC International Conference on System of Systems Engineering , Los Angeles, CA, USA, April 2006.

[3] C.D. Jilla, DW Miller and RJ Sedwick, "Application of mutidisciplinary design optimization techniques to distributed satellite systems.", Journal of Spacecraft and Rockets, Vol. 37, No. 4, July – August 2000.

[4] A. Konak, D. W. Coit, A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial", Reliability Engineering & System SafetyVolume 91, Issue 9, , Special Issue - Genetic Algorithms and Reliability, September 2006, Pages 992-1007.

[5] R. Valerdi, E. Axelband, T. Baehren, B. Boehm, D. Dorenbos, S. Jackson, A. Madni, G. Nadler, P. Robitaille and S. Settles, "A research agenda for systems of systems architecting ", Int. J. System of Systems Engineering, Vol. 1, Nos. 1/2, 2008.

[6] J. Hughes, Intelligrid[SM] Architecture Application Guide, www.icc.illinois.gov/downloads/public/edocket/219725.pdf.

[7] Advanced Metering Infrastructure- systems engineering intro, Technical Advisory Board, www.sce.com/NR/rdonlyres/B5866238-9196-4376-86B1-16B6A258FDEC/0/AMIIntroTAB11905.pdf

[8] Electrical Power Research Institute (EPRI) IntelliGrid[SM], http://intelligrid.epri.com/

[9] The Department of Energy, Modern Grid Initiative (MGI) http://www.netl.doe.gov/moderngrid/

[10] A. S. Fukunaga, S. Chien, D. Mutz, R. L. Sherwood, and A. D. Stechert. "Automating the Process of Optimization in Spacecraft Design." Proceedings of IEEE Aerospace Conference. Snowmass, CO, 1997.

[11] R. A. Hassan, and W. A. Crossley. "Multi-Objective Optimization of Communication Satellites with Two-Branch Tournament Genetic Algorithm." Journal of Spacecraft and Rockets 40, no. 2 (2003): 266-272. MDO.

[12] M. A. Walton and D. E. Hastings,"Applications of Uncertainty Analysis Applied to the Conceptual Design of Space Systems.", http://ocw.mit.edu.