INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED11
15 - 18 AUGUST 2011, TECHNICAL UNIVERSITY OF DENMARK

EQUILIBRIUM DESIGN PROBLEMS IN COMPLEX SYSTEMS REALIZATION

Jitesh H. Panchal
Washington State University, Pullman, WA, USA 99163

ABSTRACT

Equilibrium design is a class of problems where the design of complex systems is not directly controlled by designers but emerges from the self-interested decisions of stakeholders. While such problems have been common in economics and social sciences, they have not yet been addressed in engineering design. This is because the focus in engineering design is on technical performance with the assumption that designers directly control the design space. However, with the increasingly interconnected nature of the technical, social, economic and environmental aspects, equilibrium design problems become more important for designers. Instead of solving a specific equilibrium design problem, the goals in this paper are to highlight the importance and uniqueness of this class of problems and to present a general formulation within engineering design context. Specifically, we present a general formulation using concepts from non-cooperative game theory, mathematical tools for solving them, and various example problems relevant to engineering design that can be modeled as equilibrium design problems.

Keywords: Systems design, non-cooperative games, Nash equilibrium, mathematical programming

1 INTRODUCTION

How can we design large-scale complex systems whose structures and behaviors are not directly controlled by designers, but emerge dynamically from the local decisions and self-organization of individual entities? This question has been central to many parts of economics and social sciences. The design of markets, mechanisms, auctions, and organizations, all deal with essentially the same question. Increasingly, this question is also becoming relevant for engineering designers dealing with large-scale complex systems that involve technical, social, economic, and environmental aspects.

Traditionally, engineering design research has primarily been focused on systems whose design space is directly in control of the designers. However, there is an increasing importance of complex systems that are not designed, but emerge out of the individual decisions of different stakeholders. A prime example of such systems is the Internet which has evolved as a result of independent decisions of multiple stakeholders. Other examples include traffic systems, peer-to-peer networks, and communication networks. The key characteristic of such systems is that the overall performance is dependent on the design, which in turn is dependent on decisions made by individual decision-makers. The design (and hence, performance) of such systems can be directed by affecting the decisions of the individual stakeholders through different mechanisms such as the provision of incentives.

The natural framework for analyzing systems that involve multiple decision-makers is non-cooperative game theory [1]. Non-cooperative games have been used in engineering design, primarily as a way to represent decentralized design scenarios [2, 3] where designers are modeled as decision-makers. Decentralized design is characterized by four conditions [3]: a) designers have knowledge of only their own local objectives, b) designers act unilaterally to minimize their objective function, c) designers have complete control over specific local design variables, and d) designers communicate by sharing the current value of their local design variables. The decisions are in equilibrium if none of the designers can unilaterally improve their payoff by changing their own decisions. This equilibrium is referred to as the Nash Equilibrium. Current research on non-cooperative game theory in engineering design is focused on identifying the Nash equilibrium and its stability properties. However, the goal from a systems design standpoint is to achieve desired system performance by influencing stakeholder decisions. We refer to the corresponding problem as the “equilibrium design problem” in this paper.

The goals of this paper are three fold: 1) to define the equilibrium design problem, 2) to discuss mathematical tools that can be used for solving equilibrium design problems, and 3) to discuss various problems in engineering design that can be modeled as equilibrium design problems. The organization of the paper is as follows. In Section 2 we provide a background on non-cooperative game theory and the concepts of equilibrium in games. The general equilibrium design problem is formulated in Section 3. Two mathematical tools for solving the equilibrium design problem are discussed in Section 4. Examples of problems in engineering systems design that can be formulated as equilibrium design problems are discussed in Section 5. Closing thoughts are presented in Section 6.

2 BACKGROUND: NON-COOPERATIVE GAMES AND EQUILIBRIA

2.1 A Brief Overview of Non-Cooperative Games

In game theory, non-cooperative games [1] are models of situations where individuals make independent decisions without collaboration or communication. The individuals are referred to as players whose decisions may affect each other. A non-cooperative game consists of n players; each player has a finite set S_i of pure strategies. A combination of all the *strategies* of players in the product space $S = S_1 \times S_2 \times \ldots \times S_n$ is called the *strategy profile* of the game. Corresponding to each player i, there is a payoff function, p_i, which maps the player's strategies to real numbers. The payoffs capture the preferences of decision makers, with higher payoffs being more preferable to lower payoffs. In the literature on decision-making, payoffs are commonly represented as utility functions [4]. A *mixed-strategy* of player i is a probability distribution over the player's pure strategies.

At the core of non-cooperative games is the concept of equilibrium. A set of strategies is in equilibrium if no player has an incentive to unilaterally change the strategy. Mathematically, a strategy profile $s^* \in S$ is a *Nash equilibrium* if

$$\forall i, s_i \in S_i, s_i \neq s_i^* : p_i(s_i^*, s_{-i}^*) \geq p_i(s_i, s_{-i}^*) \qquad (1)$$

where $p(s_i, s_{-i}^*)$ represents a change in player i's strategy from s_i^* to s_i, while keeping all other players' strategies the same. The equilibrium is called strict Nash equilibrium if the symbol $\geq$ is replaced with $>$ in equation (1). Nash equilibrium can be defined either for pure strategies or for mixed strategies. Nash proved that every finite game has at least one mixed-strategy Nash equilibrium [1]. A game can have multiple Nash equilibria.

2.2 Challenges Associated with Nash Equilibria

Nash equilibrium is only one type of equilibrium for non-cooperative games. A generalization, referred to as *correlated equilibrium* was first suggested by Aumann [5]. In this case, the players choose their strategies based on a public signal from a trusted party. The trusted party chooses a strategy profile according to a probability distribution and informs it to the corresponding players. Individual players choose their strategies based on this information. If no player has an incentive to unilaterally deviate from his/her strategy, then the strategy set is called a correlated equilibrium. The advantages are that correlated equilibria always exist for finite games [6], they may be more efficient than Nash equilibria [7], and unlike Nash equilibria they can be efficiently computed and learnt [8, 9].

Despite the generality of the concept of correlated equilibrium, the concept of Nash equilibrium is used as the standard notion of equilibrium. Papadimitriou and Roughgarden [10] suggest that this is because "everybody uses it," it is used as a baseline for refinements and generalizations (such as the correlated equilibrium), and it is an open computational problem in computational game theory. Hence, in this paper, we focus on the Nash equilibrium to illustrate the problem of designing equilibria in decentralized systems design.

One of the key challenges in non-cooperative game theory is the *complexity of finding Nash equilibria*. Nash [1] commented that "The complexity of the mathematical work needed for a complete investigation increases rather rapidly…" Even with the developments in computers during the past 60 years, calculating Nash equilibria is still challenging. Various algorithms have been proposed for finding Nash equilibria but none of them is known to run in polynomial time [11]. Recently, the computational complexity of the problem of calculating Nash equilibria, even for a two player game, is classified as PPAD (Polynomial Parity Arguments on Directed graphs) complete [11-13].

The second challenge is related to the *dynamics of the processes* leading to the equilibria [14]. The dynamics refers to the sequence of decisions made by individual players in response to the decisions made by preceding players. Arguably the simplest and the most popular game dynamics is the "best response" (BR) dynamics [15] where at a given time, a small portion of players adjust their strategy to a strategy that is the best response to the current strategy of the other players. Other types of dynamics are also studied in the literature where the players anticipate future strategies of other players and respond accordingly. Convergence to an equilibrium depends on a) the game (i.e., the payoff functions of the individuals), and b) the dynamic process. Not all dynamic processes for a game converge to Nash equilibria. Similarly, a given dynamic process may or may not converge depending on the game. General results on the convergence of dynamics to Nash equilibria are available only for a small class of games. It has been shown that for a class of games called *potential games* [16] simple dynamics such as best response are guaranteed to converge to a Nash equilibrium. The rate of convergence and the dynamic stability of the equilibria are other related issues.

The third major challenge is related to the *inefficiency of equilibria* [17]. The prisoner's dilemma is a well-known example illustrating that the equilibrium achieved by the decentralized decisions of players may be less than the socially optimal solution that can be achieved by a central authority. The most commonly used notion of optimality is Pareto optimality. A set of strategies is Pareto optimal if it is impossible to strictly increase the payoff of a player without strictly decreasing the payoff of another player. The extent of inefficiency of the equilibria can be measured using measures such as a) the price of anarchy, and b) the price of stability. The price of anarchy is the ratio between the worst objective function of the equilibrium and that of an optimal outcome. On the other hand, the price of stability is the ratio of the best objective function value to that of an optimal outcome. Hence, price of anarchy is based on a pessimistic view whereas the price of stability is based on an optimistic view.

As a summary, the key challenges associated with equilibria in non-cooperative games are: a) efficiently determining the Nash equilibrium points, b) ensuring that the dynamics converges to the Nash equilibria, and c) ensuring that the Nash equilibria are closer to the efficient solutions.

2.3 Illustrative Example

In this section, a simple example with two designers (players), each with an objective function, is presented to illustrate the concept of Nash equilibrium. Consider two designers with payoff functions F_1 and F_2 as shown in Table 1. The payoff functions are polynomials in two variables x_1 and x_2. The strategy set is determined by the values of the design variables that each designer can control. The two designers are responsible for different variables; the first designer can choose values for x_1 and the second designer can choose the values for x_2. The strategy set for the first designer is x_1 = [0 1] and the strategy set for the second designer is x_2 = [0 1].

Table 1- Payoff functions and strategy sets of the two designers in the illustrative example

Designer 1	**Designer 2**
Payoff Function: Maximize: $F_1 = x_1x_2 - x_1^3$ *Strategy set:* x_1 = [0 1]	*Payoff Function:* Maximize: $F_2 = x_1x_2 - x_2^3$ *Strategy set:* x_2 = [0 1]

The strategy of each designer is to choose the values of corresponding design variables such that their payoffs are maximized for a given value of the design variable chosen by the other designer. This is referred to as the "best response" to the other designer's strategy. Hence, based on the first order optimality condition, Designer 1 chooses x_1 for a given value of x_2 such that $\frac{\partial F_1}{\partial x_1} = 0$ and Designer 2 chooses x_2 such that $\frac{\partial F_2}{\partial x_2} = 0$. The strategy profiles corresponding to these optimality conditions are referred to as the Best Response Correspondences (BRCs) of the designers. The BRCs for the example problem are shown in Figure 1. The points of intersection of the BRCs for the two designers are the Nash

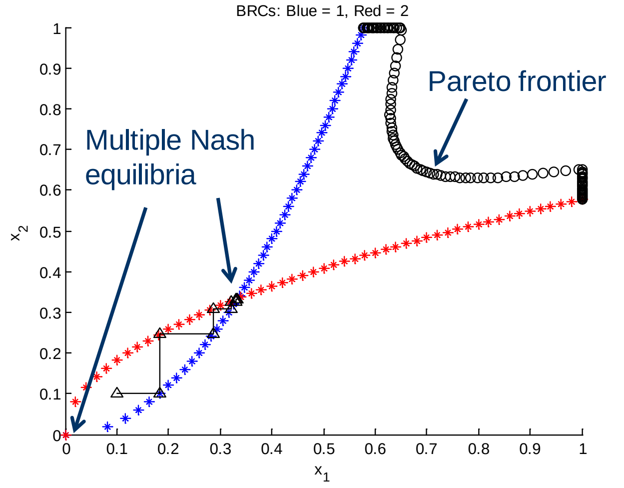

Figure 1 - Best response correspondence and Nash equilibria for the illustrative example

equilibria for the game. In the illustrative example, there are two Nash equilibria: (x_1, x_2) = (0, 0) and (1/3, 1/3).

If the designers sequentially choose the best response to the other designer, the dynamics is termed as a best response dynamics. The best response dynamic can be represented using the following iterated map: $x_{1,n+1} = \sqrt{\frac{x_{2,n}}{3}}$; $x_{2,n+1} = \sqrt{\frac{x_{1,n+1}}{3}}$. The iterated map may or may not converge to the Nash equilibrium. In Figure 1, it is shown that starting from an initial point ($x_1 = x_2 = 0.1$) the iterated map converges to one of the Nash equilibria: (x_1, x_2) = (1/3, 1/3). The Pareto optimal solutions for the game are shown in the figure. At these points none of the designers can improve their payoff without adversely affecting the other player's payoff. Comparing the Nash equilibria with the Pareto solutions, it is observed that the Nash equilibrium (1/3, 1/3) is closer to the Pareto frontier.

3 EQUILIBRIUM DESIGN PROBLEM IN NON-COOPERATIVE GAMES

The discussion in the previous section is focused on a set of fixed equilibrium points. Now consider a scenario where it is possible to modify the individual designers' payoffs through some incentives. In such cases, the individuals' strategies vary based on the payoffs. Hence, the corresponding Nash equilibria also change. The new Nash equilibria may have different efficiency (price of anarchy and price of stability), convergence, and stability characteristics compared to the original equilibria. By appropriately choosing the incentives to modify the designers' payoffs, the *resulting Nash equilibria can be designed* to possess the desired characteristics. This design problem is referred to as the equilibrium design problem. The higher level authority that has the power to provide incentives to modify individual payoffs is referred to as a "game designer."

For illustrative purposes, we extend the example from Section 2.3 to an equilibrium design problem. Assume that the payoff functions of the two designers contain parameters c_1 and c_2 that can be selected by the game designer (see Table 2). The ranges of these parameters are c_1 = [0 3], c_2 = [0 3]. For $c_1 = c_2 = 1$, the payoffs are similar to the previously discussed scenario. The best response strategies of the two designers for different values of the parameters are shown in Figure 2. The intersection of the best response correspondence of the two designers for different combinations of c_1 and c_2 are shown in Figure 3. Each intersection point corresponds to a combination of the parameters. The region highlighted in the figure is the sub-space of the joint strategy space where each point can be achieved as Nash equilibrium by choosing appropriate values of the parameters. We refer to this region as the *Nash feasible space* of the equilibrium problem. This notion is similar to the notion of feasible space in optimization problems because no point outside this space can be achieved as Nash equilibrium.

Table 2- Modified objective functions of the two designers

Designer 1	Designer 2
Objective Function: Maximize: $F_1 = c_1 x_1 x_2 - x_1^3$ *Strategy set:* x_1 = [0 1]	*Objective Function:* Maximize: $F_2 = c_2 x_1 x_2 - x_2^3$ *Strategy set:* x_2 = [0 1]

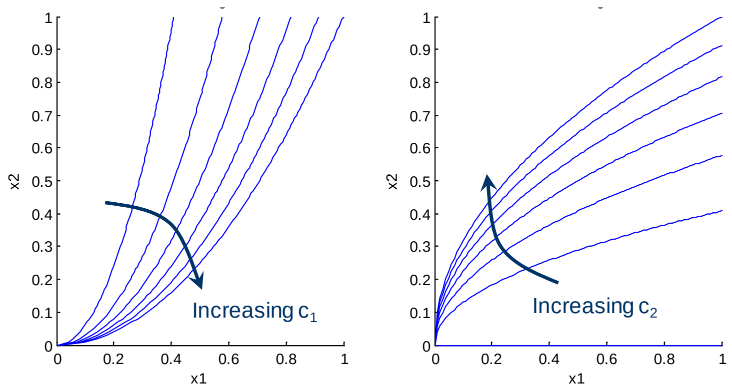

Figure 2 – The set of BRCs of the two designers (left: Designer 1 and right: Designer 2)

Based on the discussion in the previous section, the desired properties of the designed Nash equilibrium, and hence the **goals for the equilibrium design problems** are:

1. Closeness to the best solution: The Nash equilibrium should be close to the best possible (efficient) solution. The goodness of the solution can be defined in various different ways. Pareto efficiency, discussed in Section 2.2, is one of the most widely used concepts for quantifying the goodness of a solution in a decentralized system. If Pareto efficiency is used to define the ideal solution, the price of stability of the Nash equilibrium should be as close to 1 as possible. However, it is important to recognize that there are other ways of defining efficiency. Researchers in economics have proposed concepts such as Kaldor-Hicks efficiency, X-efficiency, allocative efficiency, distributive efficiency, dynamic efficiency, and productive efficiency. All of these notions of efficiency relate the individual preferences to the overall performance of the social system. Within systems design, the system level goal may or may not correspond to the efficiency of the solution (further discussed in Section 5.1). Hence, we do not limit ourselves to the notion of Pareto efficiency because within systems design, the quality of the solution may be dictated by system-level goals.

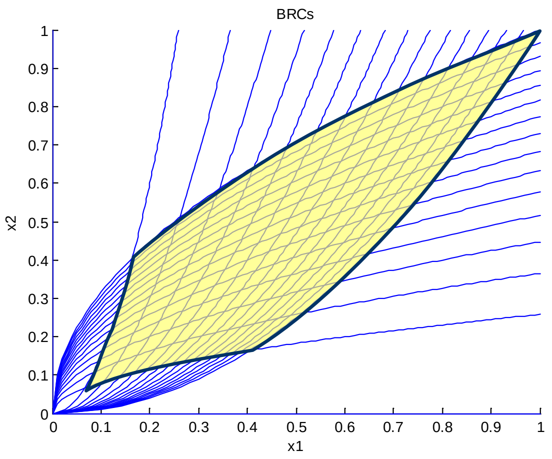

Figure 3 – The Nash feasible space resulting from the intersection of a range of BRCs

2. Convergence of the equilibrium: The dynamics of information exchange and decision making plays a significant role in the equilibrium design problems. There are various dynamic processes, the best response correspondence being the simplest one. Ideally, the dynamics of the process should result in the convergence of the solution to Nash equilibrium. However, it has been shown that depending on the problem, the dynamic processes may or may not converge. Additionally, for problems with multiple equilibria (as in Section 2.3), the process should converge to the one closer to the desired solution. Hence, achieving the convergence properties is an essential goal for the equilibrium design problems.

3. *Stability of the equilibrium*: The Nash equilibrium should be stable, i.e., small perturbations should not result in divergence from the equilibrium. Further details on stability are provided in Section 4.2.

As a summary, an equilibrium design problem can be defined by decision makers, their strategy space, individual payoffs, system-level goals, dynamic processes, and ways in which individual payoffs can be modified to affect the equilibrium and its characteristics (closeness to the best solution, convergence, and stability).

4 MATHEMATICAL TOOLS FOR EQUILIBRIUM DESIGN PROBLEMS

Having identified the key characteristics of an equilibrium design problem, the key question is: How can the equilibrium design problem be systematically formulated and solved? In this section, we discuss two tools, one from optimization theory and another from non-linear control theory that can be used as foundations for formulating and solving equilibrium design problems. In Section 4.1, we discuss mathematical programming with equilibrium constraints (MPEC) for finding the location of the best equilibria, and in Section 4.2 we discuss Lyapunov stability theory for assessing the stability and convergence characteristics of the equilibria.

4.1 Mathematical Programming with Equilibrium Constraints (MPEC)

The MPEC is a type of constrained nonlinear programming problem where some of the constraints are defined as parametric variational inequality or complementarity system [18]. These constraints arise from some equilibrium condition within the system, and hence, are called equilibrium constraints. MPEC is a special type of bi-level programming problems [19] consisting of a higher level optimization problem, whose constraints are defined in terms of solutions to lower-level optimization problems. MPEC is applicable to a variety of problems in engineering such as optimal design of mechanical structures, network design, motion planning of robots, and facility location and production. MPEC is also used to study equilibrium problems in economics. MPEC is closely related to the Stackelberg game [20] where a leader makes a decision first and then the followers make their decisions based on the leader's decision. The leader corresponds to the upper-level optimization problem in MPEC and the followers correspond to the lower-level problems. Examples of problems of economic equilibrium where MPEC has been used include maximizing revenue from tolls on a traffic system, optimal taxation, and demand adjustment problems.

Mathematically, a MPEC problem can be represented using two sets of variables, $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$. Here, *x* belongs to the upper-level problem and *y* solves the lower-level equilibrium problem. The solution of *y* depends on the value of *x* chosen for the upper-level problem. The overall objective function *f(x,y)* is minimized.

$$\min_{(x,y)} f(x,y) \quad (2)$$

$$\text{Subject to: } (x,y) \in \Omega, \text{ and } y \in S(x) \quad (3)$$

where Ω is the joint feasible region of *x* and *y*; and *S(x)* is a set of variational inequalities that represent the equilibrium problem. In the case of the equilibrium design problem discussed in Section 3, the set *x* is the set of parameters c_1 and c_2 affecting the individual payoffs and controlled by the game designer. Based on the values of these parameters, the players determine their best responses to the other players. Hence, the variables x_1 and x_2 in Section 3 correspond to the variable *y* in the MPEC formulation above. The intersection of the best responses is the Nash equilibrium point. Here, the function *f(x,y)* represents a system-level function that quantifies the goodness of the solution (as defined in Section 3).

The set *S(x)* corresponds to the feasible Nash space. As discussed earlier, the Nash equilibrium corresponds to the best response of each designer to the decisions made by other designers. The Nash equilibrium point can be formulated as a variational inequality using the first order necessary conditions for optimality such as Karush–Kuhn–Tucker (KKT) conditions [21]. Having formulated the equilibrium design problem as a MPEC, the next step is to solve it. Solving the MPEC problems is challenging because of the non-linearities in the problem, non-convex feasible space, combinatorial nature of constraints, disjointed feasible space, and multi-valued nature of the lower equilibrium problem [18]. There has been some progress in developing efficient algorithms for solving MPEC problems. Examples include variations of NLP algorithms, and interior point algorithms [22].

As a summary, MPEC can be used as a mathematical tool to model equilibrium design problems. We discussed how the formulation can be used to account for the goodness of the Nash equilibrium. MPEC results in the best Nash Equilibrium. However, this only addresses the first requirement listed in Section 3. It does not account for the dynamics of the problem. The solution does not provide any insight into the convergence and stability of the equilibrium. To address this limitation, we utilize some of the tools from non-linear control theory. We specifically focus on the Lyapunov stability theory in the next section.

4.2 Lyapunov Stability Theory

In Section 2.3, we illustrate how the best response dynamics can be modeled as an iterative map where the values of the variables during some iteration are given in terms of the values in the previous iteration. This represents a dynamic system for which the stability characteristics of the equilibrium points can be evaluated using the Lyapunov Stability Theory [23]. For nonlinear systems, various notions of stability have been developed. These include Lyapunov stability, asymptotic stability, exponential stability, and global asymptotic stability. Lyapunov stability means that trajectories in the phase space starting at two points close to equilibrium will stay sufficiently close to it. Asymptotic stability is a stronger notion of stability where in addition to Lyapunov stability the trajectories starting close to the equilibrium also converge to the equilibrium as time goes to infinity. An equilibrium point is exponentially stable if the trajectories converge to the origin faster than an exponential function.

The Lyapunov theory consists of a direct method and an indirect method for evaluating the stability of nonlinear dynamic systems. In the indirect method, the nonlinear system is approximated as a linear system near the equilibrium points and the stability of the linear system is determined. For a nonlinear system in the state space representation $\dot{x}_i = f_i(x_1, x_2, \dots, x_n)$ where x_i is a state variable, then the equilibrium point is given by $f_i(x_1, x_2, \dots, x_n) = 0 \ \forall\, i$. If the equilibrium is at the origin, the stability can be determined from the eigenvalues of the Jacobian matrix $J = \left(\frac{\partial F}{\partial X}\right)$ where F is the set of functions f_i and X is the set of state variables x_i. If all the eigenvalues are negative, then the system is stable. If at least one of the eigenvalues is positive, then the system is unstable. In contrast to the indirect method, the direct method accounts for the nonlinearities in the system using the notion of a Lyapunov function. For a system, if there is a positive-definite function of state variables which decreases along all state trajectories, the system is stable. That function is called a Lyapunov function of the system. A system can have more than one Lyapunov function. The key challenge in using the direct method is that there is no general method to find the Lyapunov function for a system. It requires intuition and trial-and-error. In the case of iterated maps, which are discrete-time systems, the concept of Lyapunov exponents is used to determine stability of a system. Assuming that two trajectories start at nearby points x_0 and $(x_0 + \delta_0)$. Then, the separation between the two points after n iterates is $|\delta_n| = |\delta_0|e^{n\lambda}$. The equilibrium point is stable if $\lambda < 0$. As elaborated by Strogatz [24], the Lyapunov exponent for a system trajectory starting at x_0 is:

$$\lambda = \lim_{n\to\infty}\left\{\frac{1}{n}\sum_{i=0}^{n-1} ln|f'(x_i)|\right\} \quad (4)$$

In order to incorporate stability as an integral part of the solution of the equilibrium design problem the Lyapunov stability criteria need to be integrated within the MPEC framework. One possibility is to develop constraints on the eigenvalues and to integrate them as constraints within the MPEC. Another potential approach is to determine a set of good solutions from MPEC and then to perform stability analysis to determine the best solution from the stability standpoint. The third approach is to determine basins of attraction within the design space and then use them as a feasible design space for the MPEC. Currently, there is a lack of methods that account for both stability and efficiency in an integrated manner for the equilibrium design problems. Further investigation is necessary.

5 EQUILIBRIUM DESIGN PROBLEMS WITHIN ENGINEERING DESIGN

The equilibrium design problem can be found in various problems related to engineering design. In this section, we discuss some classes of problems that have equilibrium design at their core. All of these classes of problems can be viewed from a collective systems perspective and it is possible to modify individual preferences through the provision of incentives. An overview of the characteristics of the problems is provided in Table 3.

Table 3 - Overview of the classes of equilibrium design problems in engineering design

	Decision Makers	**Individual Preferences**	**System-level goals**	**Dynamic Processes**	**Mechanisms for equilibrium design**
Requirements allocation	Designers working on different aspects of the system	Subsystem goals	System-level design goals	Updating individual decisions based on others' decisions	Requirements decomposition and targets
Evolutionary networks	Entities making decisions about linking with different nodes	Payoffs for individuals are network dependent	Performance of the network	Addition and removal of nodes and links	Incentives to individuals to affect linking behaviors
Collective Innovation	Individuals participating in collective innovation projects	Satisfying intrinsic and extrinsic needs	Development of the entire project	Self-organization of communities and growth of products	Incentives to individuals to participate in different activities
Decentralized Energy	Individual consumers	Minimize cost of owning and operating energy resources	Technical, Economic, Environmental, Social	Market processes of purchasing and selling energy	Taxes, incentives, laws, market rules

5.1 Requirements allocation in decentralized design

Requirements allocation, as used in systems engineering, is the process of decomposing system-level requirements into requirements for lower-level subsystems and components. Requirements allocation is a part of the requirements-engineering process. There are two types of requirements at the system level – a) requirements that can be directly assigned to individual subsystems and components, and b) requirements that need to be divided among multiple components. An example of the former type of requirements for an automotive system is “provide energy” which can be fulfilled by a lower level subsystem such as an “engine”. An example of the latter type is “weight should be lower than 10,000 kg” which can be divided into upper bounds of weights for individual systems. Such requirements are also referred to as allocable requirements. Here, weight is a system attribute which is a function of attributes of the components. Collopy [25] refers to these attributes as extrinsic attributes. Other examples of extrinsic attributes are cost, efficiency, and reliability.

The performance of the overall system is dependent on the allocation of requirements. Hence, the requirements allocation should be such that it maximizes the overall system performance. Traditionally, requirements allocation has been carried out by system designers based on their insights and the knowledge from prior systems. The effect of alternate requirements allocations on the system performance is rarely considered. The requirements allocation problem can be modeled from two different perspectives: optimization and non-cooperative game theory, as discussed next.

Optimization Perspective: Consider a single organization representing a completely collaborative scenario where the goals of all designers are to achieve the system-level objectives, and information can be freely shared among them at any stage during the design process. In such a scenario, the requirements allocation problem involves finding the best decomposition of requirements for extrinsic attributes. From an optimization perspective, the widely utilized approach for requirements allocation is to determine lower and upper bounds for the extrinsic attributes for subsystems. These bounds are such that if all the lower-level designers designed their subsystems to satisfy their corresponding bounds, the system level requirements are automatically satisfied. The bounds are used as constraints in sub-system level design problems. Collopy argues that using requirements for extrinsic attributes as constraints for subsystems and components results in inferior systems as compared to using them as parts of objective functions [25]. Whether the requirements are modeled as constraints or parts of objective functions, existing multi-disciplinary optimization approaches such as collaborative optimization [26], analytical target cascading [27], etc. can be used to model the scenario.

Non-cooperative Game Perspective: Consider another scenario of systems design carried out by multiple distributed entities (e.g., organizations or teams) where each entity has its own underlying goals and there are barriers (both organizational and technical) to complete information exchange throughout the design process. This scenario is representative of many complex automotive and aerospace systems designed by multiple organizations. Due to the limited information flow between designers, the resulting solution is equilibrium. Here, requirements allocation modifies individual payoffs and hence, acts as a way of modifying the equilibrium. In such a scenario, the requirements allocation problem can be modeled as an equilibrium design problem. The systems designer’s decision is to determine the best allocation of requirements such that when the individual designers make their decisions, the equilibrium is close to the desired solution. The approaches discussed in Sections 4.1 and 4.2 can be used to model the equilibrium design problem. Each equilibrium design problem has unique challenges. One of the challenges is the lack of detailed models of individual subsystems before the systems are designed. Without the availability of these subsystem models, it is challenging to model the impact of alternate ways of requirements allocation on the equilibrium and its properties.

5.2 Design of complex evolutionary networks

Recently a number of large-scale complex networks have been identified whose structures are not directly controlled by designers, but emerge dynamically from the local decisions and self-organization of individual entities. A prime example of such a network is the Internet, whose structure is a result of individual connection decisions made by individual entities. The overall topology of the Internet affects its reliability, the effectiveness of search, etc. The Internet is just one example of such networks. Other examples include social networks, ad hoc networks, trade networks etc. All of these networks are similar in the sense that: a) the topology is a result of local behaviors, and b) the topology has a significant effect on their performance. Such networks are also referred to as *endogenous networks* [28]. The system-level objective is to guide the evolution of such networks towards desired structures with desired behaviors and performance.

Existing approaches for network design are focused on centralized network design applied to networks such as transportation [29]. In these problems, the design variables are nodes and links, and the objectives are minimization of the cost of transportation, minimization of the distance travelled, etc. On the other hand, the design of complex evolutionary networks is governed by individual decisions which can be modified by providing incentives. The individual decisions are also based on the decisions made by decision makers. Hence, the decisions can be modeled as equilibrium problems. The individual decisions affect the formation of nodes and links, thereby affecting the network structure, and hence, the network performance. The goal is to maximize the performance of the overall network. For example, one of the performance goals for the Internet is to maximize the effectiveness of search. Hence, the design problem in such evolutionary networks is fundamentally different from traditional network design, and is more appropriately represented as an equilibrium design problem. The challenge associated with such problems is the presence of large and discrete design spaces.

5.3 Collective Innovation

Collective innovation is an emerging paradigm in product realization where complex systems are developed in a bottom-up manner by communities of independent individuals, as opposed to hierarchical organizations. The paradigm is epitomized by successful examples from open-source software development (e.g., Linux, Apache), crowdsourcing (e.g., Innocentive), and open encyclopedias (e.g., Wikipedia). The fundamental difference between traditional product realization processes and collective innovation processes is that the former are based on top-down decomposition and structured task assignment while the latter are based on self-organization of individually motivated participants into communities. The participants' contributions are not based on the pre-specified tasks, and the product evolves over time based on the contributions of the participants [30].

Collective innovation can be viewed from a complex systems perspective where individuals are decision makers with their own preferences, needs, and capabilities. The individual goals can range from enjoyment-based or community-based intrinsic motivation to extrinsic motivations such as career advancement and skill improvement [31]. Based on their interests and competencies, individuals make decisions such as whether to participate or not, which project to participate on, and whom to collaborate with. They self-select (or define) the activities they would like to participate in. Individuals interact with each other in two ways – directly and indirectly. The direct interactions are through one-on-one discussions, online forums, and other web-based mechanisms. The indirect interactions are mediated through the product structure, which is an essential aspect of the environment. Based on these decisions and interactions, the product evolves. At the same time, as the individuals decide to collaborate with each other on different tasks, the community structure also grows with time.

Panchal et al. [32] highlight that due to the interdependencies between different product modules, the product sequence in which they are developed has an impact on the growth rate of the product. The modules on which other modules depend should be developed first. In addition to the dependencies between modules, both the product structure and the community structure are also interdependent [32]. The growth of communities affects the way in which products evolve and the growth of products affect the evolution of communities. Some product structures and community structures are better than others in terms of collective innovation [33]. If the system-level goal is rapid growth of the product being designed, collective innovation is also associated with an equilibrium design problem. The individual behaviors can be modified by providing different types of incentives (such as awards and recognition). As the individuals make decisions to participate on different activities, the equilibrium design problem is to determine the incentives that can be provided to them at different points in time to participate on appropriate modules at appropriate time. Through a systematic design of the incentive structure, the dynamics of collective innovation processes can be directed towards faster growth and the achievement of targeted structures of products and communities.

5.4 Decentralized energy

The current energy infrastructure in the United States (and most other countries around the world) is primarily based on a centralized model of energy generation and distribution. The centralized energy model is characterized by large-scale power plants from which energy is distributed to the consumers through a centrally controlled network of cables. While the centralized energy model has been used for over two centuries, it has a number of limitations – a) waste of energy, b) large transmission losses, c) expensive distribution infrastructure, d) low resilience to failure, and e) impacts on the environment [34]. With the increasing use of small-scale energy generation from renewable sources and increasing deregulation of the energy sector, an alternative paradigm of energy generation and distribution is emerging. It is called *decentralized energy* and consists of distributed generation resources [34]. It is believed that decentralized energy can address the limitations of centralized energy. Since energy is generated closer to the consumers, the waste heat generated in the process can be used for space heating purposes. Since the energy does not need to be transmitted over long distances, the transmission losses are lower. Further, the cost of distribution infrastructure is lower. However, decentralized energy is faced with challenges such as poor control on the power quality due to the intermittency of renewable energy sources.

Energy infrastructure is associated with multiple levels of decisions such as network reconfiguration, service restoration, operation planning, and expansion planning [35]. These decision problems vary in their objectives and time horizons. The optimal network reconfiguration is an operation-related problem of finding the branches of the network to be opened to supply the loads with minimum energy losses. Optimal service restoration involves identifying the best strategy to meet the demands after a fault to minimize the effect of fault propagation. Operation planning involves choosing the optimum structural changes considering a constant load in the network. Expansion planning involves deciding how to grow the network considering future changes in demand. The timescale considered for expansion planning is about 20 years.

Within a centralized energy paradigm, all these decisions are made by central authorities who are in-charge of the infrastructure. However, in a decentralized infrastructure, the fundamental difference is that different stakeholders make their own decisions and the overall system-level performance is dependent on the individual decisions. For example, the consumers play an active role by acting as producers. They make their own decisions on a) which technologies to invest in, b) how much energy to generate, c) how much energy to buy and from whom, and d) how much energy to sell [36]. Other stakeholders include power producers (e.g., utility companies), grid operators, and regulators (e.g., government and other regulating authorities). The decisions made by different stakeholders are often conflicting. For example, consumers' decisions are often driven by economic aspects such as minimizing their energy costs. In addition to economic objectives, other objectives such as technical, environmental and social objectives are also important from a systems level. Technical objectives include minimum system losses, voltage stability, unbalance conditions, power quality, and energy needs. Environmental objectives include minimization of emissions and hazardous materials. Social objectives include fairness and quality of life.

Based on the decisions made by the individual stakeholders, the system reaches an equilibrium point which defines its overall behavior. The individual decisions can be directed through a number of mechanisms such as policy tools, incentives (e.g., tax breaks), penalties (e.g., tariffs), markets rules, and laws. The corresponding equilibria can be changed through these mechanisms. Hence, the decisions within decentralized energy can be modeled as an equilibrium design problem. The decisions of the policy makers can be represented as the higher-level optimization problem and the decisions of the individual consumers can be represented as lower-level equilibrium problem.

5.5 Other problems

There are various other examples of systems with similar structure. The sustainability standards development process such as LEED is a natural example of equilibrium design problem. The standards influence the design decisions made by architects and material decisions by builders. By appropriately choosing the standards, the decisions can be directed towards better environmental performance. Similarly, most of the problems that relate to policy design, determination of the right amount of taxes (e.g., carbon tax) are also equilibrium design problems.

6 CLOSING COMMENTS

As the scope of problems considered by engineering design researchers is extended beyond just technical design to include broader aspects such as organizational design, policy, and economics, the equilibrium design problem becomes pervasive. The first step towards addressing these challenges is to recognize the common structure of all these problems and the existence of tools in different fields that can be used to address some of the associated challenges. That is the primary goal of this paper.

Traditionally, problems related to equilibrium design are studied in economics, social sciences, and computer science. Specifically, the field of mechanism design [37] within economics deals with the "design of games" with desired outcomes. Design of multi-agent systems [38] involves designing the behaviors of individual decision-making agents to achieve the desired system-level behaviors. In this paper, we show that equilibrium design problems are also central to engineering systems design and hence, needs attention from the engineering design community.

Equilibrium design problems are complex in nature. One of the key challenges is the multi-objective nature of desired system-level outcome. Different problems are associated with unique challenges that require different ways of addressing them. As discussed in Section 5, equilibrium design problems in requirements allocation are challenging due to the lack of subsystem models before they are designed. The problems in evolutionary networks are challenging due to discrete and vast design spaces. There are various opportunities for research in addressing such challenges in specific classes of problems. Additionally, the classes of problems discussed in the paper are only representative examples of equilibrium design problems based on the author's own research. Many more such problems can be identified.

REFERENCES

[1] Nash, J. Non-Cooperative Games. *The Annals of Mathematics*, 1951, 54(2), 286-295.

[2] Marston, M. and Mistree, F. Game-Based Design: A Game Theoretic Extension to Decision-Based Design. In *ASME DETC, Design Theory and Methodology Conference*, Baltimore, MD, 2000, DETC2000/DTM-14578.

[3] Chanron, V. and Lewis, K. A Study of Convergence in Decentralized Design Processes. *Research in Engineering Design*, 2006, 16, 133-145.

[4] Fishburn, P.C. *Utility Theory for Decision Making*, 1970 (Wiley, New York).

[5] Aumann, R. Subjectivity and Correlation in Randomized Strategies. *Journal of Mathematical Economics*, 1974, 1(1), 67-96.

[6] Hart, S. and Schmeidler, D. Existence of Correlated Equilibria. *Mathematics of Operations Research*, 1989, 14(1), 18-25.

[7] Nisan, N. Correlated Equilibria — Some CS Perspectives [online]. 2009. Available from: http://agtb.wordpress.com/2009/05/10/correlated-equilibria-some-cs-perspectives/ [November 02, 2010]

[8] Foster, D.P. and Vohra, R.V. Calibrated Learning and Correlated Equilibrium. *Games and Economic Behavior*, 1997, 21(1-2), 40-55.

[9] Hart, S. and Mas-Colell, A. A Simple Adaptive Procedure Leading to Correlated Equilibrium. *Econometrica*, 2000, 68(5), 1127-1150.

[10] Papadimitriou, C.H. and Roughgarden, T. Computing Correlated Equilibria in Multi-player Games. *Journal of the ACM*, 2008, 55(3), 14(11-29).

[11] Daskalakis, C., Goldberg, P.W. and Papadimitriou, C.H. The Complexity of Computing a Nash Equilibrium. *SIAM Journal on Computing*, 2009, 39(1), 195-259.

[12] Papadimitriou, C.H. On the Complexity of the Parity Argument and Other Inefficient Proofs of Existence. *Journal of Computer and System Sciences*, 1994, 48(3), 498-532.

[13] Conitzer, V. and Sandholm, T. New Complexity Results about Nash Equilibria. *Games and Economic Behavior*, 2008, 63(2), 621-641.

[14] Skyrms, B. Chaos in Game Dynamics. *Journal of Logic, Language, and Information*, 1991, 1, 111-130.

[15] Hopkins, E. A Note on Best Response Dynamics. *Games and Economic Behavior*, 1999, 29(1-2), 138-150.

[16] Monderer, D. and Shapley, L.S. Potential Games. *Games and Economic Behavior*, 1996, 14(1), 124-143.

[17] Roughgarden, T. and Tardos, E. Introduction to the Inefficiency of Equilibria. In Nisan, N., Roughgarden, T., Tardos, E. and Vazirani, V.V., eds. *Algorithmic Game Theory*, 2007, pp. 441-457 (Cambridge University Press, New York, NY).

[18] Luo, Z.-Q., Pang, J.-S. and Ralph, D. *Mathematical Programs with Equilibrium Constraints*, 1996 (Cambridge University Press, Cambridge).

[19] Dempe, S. *Foundations of Bilevel Programming*, 2002 (Kluwer Academic Publishers, Dordrecht, The Netherlands).

[20] Fudenberg, D. and Tirole, J. *Game Theory*, 1993 (MIT Press, Cambridge, MA).

[21] Bertsekas, D.P. *Nonlinear Programming: Second Edition*, 1999 (Athena Scientific, UK).

[22] Dirkse, S.P. and Ferris, M.C. Modeling and solution environments for MPEC: GAMS and MATLAB. In Fukushima, M. and Qi, L., eds. *Reformulation: Nonsmooth, Piecewise Smooth,*

Semismooth and Smoothing Methods, 1999, pp. 127-148 (Kluwer Academic Publishers, Dordrecht, The Netherlands).
[23] Slotine, J.-J.E. and Li, W. *Applied Nonlinear Control*, 1991 (Prentice Hall, Upper Saddle River, NJ).
[24] Strogatz, S. *Nonlinear Dynamics and Chaos: With Applications To Physics, Biology, Chemistry And Engineering* 1994 (Westview Press, Cambridge, MA).
[25] Collopy, P. Adverse Impact of Extensive Attribute Requirements on the Design of Complex Systems. In *7th AIAA Aviation Technology, Integration, and Operations Conference (ATIO),* Belfast, Northern Ireland, 2007, pp. 1326-1332, No: 7820.
[26] Kroo, I. and Manning, V. Collaborative Optimization: Status and Directions. In *8th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Long Beach CA, 2000, AIAA-2000-4721.
[27] Liu, H., Chen, W., Kokkolaras, M., Papalambros, P.Y. and Kim, H.M. Probabilistic Analytical Target Cascading -- A Moment Matching Formulation for Multilevel Optimization Under Uncertainty. *ASME Journal of Mechanical Design*, 2006, 128(4), 991-1000.
[28] Hojmana, D.A. and Szeidlb, A. Endogenous networks, social games, and evolution. *Games and Economic Behavior*, 2006, 55(1), 112-130.
[29] Ahuja, R.K., Magnanti, T.L. and Orlin, J.B. *Network Flows: Theory, Algorithms, and Applications*, 1993 (Prentice Hall Inc., Upper Saddle River, New Jersey).
[30] Panchal, J.H. Agent-based Modeling of Mass Collaborative Product Development Processes. *Journal of Computing and Information Science in Engineering*, 2009, 9(3), 031007.
[31] Lakhani, K.R. and Wolf, R.G. Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. In Feller, J., Fitzgerald, B., Hissam, S. and Lakhani, K., eds. *Perspectives on Free and Open Source Software*, 2005, pp. 3-21 (MIT Press, Cambridge, MA).
[32] Panchal, J.H. Co-Evolution of Products and Communities in Mass-Collaborative Product Development - A Computational Exploration. In *International Conference on Engineering Design (ICED'09)*, Stanford, CA, 2009, ICED'09/147.
[33] Le, Q. and Panchal, J.H. Modeling the Effect of Product Architecture on Mass-Collaborative Processes. *Journal of Computing and Information Science in Engineering*, 2011, 11(1), 011003.
[34] Pepermans, G., Driesen, J., Haeseldonckx, D., Belmans, R. and D'haeseleer, W. Distributed Generation: Definition, Benefits and Issues. *Energy Policy*, 2005, 33(6), 787-798.
[35] Chicco, G. Challenges for Smart Distribution Systems: Data Representation and Optimization Objectives. In *12th International Conference on Optimization of Electrical and Electronic Equipment (OPTIM),* Brasov, Romania, 2010, pp. 1236 - 1244.
[36] Chicco, G. and Mancarella, P. Distributed Multi-Generation: A Comprehensive View. *Renewable and Sustainable Energy Reviews*, 2009, 13(3), 535-551.
[37] Hurwicz, L. and Stanley, R. *Designing Economic Mechanisms,* 2006 (Cambridge University Press, New York, NY).
[38] Wooldridge, M. *An Introduction to MultiAgent Systems*, 2002 (John Wiley & Sons Ltd., Glasgow, UK).

Contact: Jitesh H. Panchal
Washington State University
School of Mechanical and Materials Engineering
100 Dairy Road, Pullman, WA 99164 USA
Phone: +1-509-715-9241; Fax: +1-509-335-4662; E-mail: panchal@wsu.edu
URL: http://www.mme.wsu.edu/people/faculty/faculty.html?panchal

Jitesh H. Panchal is an Assistant Professor in the School of Mechanical and Materials Engineering at Washington State University. He received his B.Tech. from IIT Guwahati (India), and MS and PhD in Mechanical Engineering from Georgia Institute of Technology, Atlanta. His research interests are in the field of collective systems innovation and multilevel design. He is a member of ASME and ASEE.