IMPROVING CONVERGENCE TO AND LOCATION OF ATTRACTORS IN DYNAMIC GAMES

EDUARDO CAMPONOGARA

Departamento de Automação e Sistemas, Universidade Federal de Santa Catarina
Caixa Postal 476, 88040-900, Florianópolis, SC, Brasil
E-mail: `camponog@lcmi.ufsc.br`

SAROSH N. TALUKDAR, HAOYU ZHOU

Department of Electrical and Computer Engineering, Carnegie Mellon University
Pittsburgh, PA, 15213-3890, U.S.A.
E-mails: `talukdar@ece.cmu.edu, hzhou@cs.cmu.edu`

Resumo— A tarefa de operar um sistema complexo é usualmente delegada a vários agentes para, dessa forma, contornar os obstáculos impostos pela dimensão do problema. Os agentes, tendo habilidades limitadas e visões reduzidas do sistema, tipicamente competem uns com os outros e atingem decisões subótimas. Qualquer que seja o sistema complexo, o trabalho de seus agentes pode ser modelado por um jogo dinâmico. Este artigo ilustra como agentes altruísticos podem direcionar as decisões para atratores ótimos de uma família de jogos. Em particular, desenvolvemos um algoritmo para encontrar respostas altruísticas ótimas, capazes de melhorar a convergência e localização de atratores em jogos dinâmicos que surgem da otimização de funções quadráticas. Além disso, o artigo relata evidências experimentais de que os agentes podem aprender tais respostas altruísticas a partir de suas experiências.

Abstract— The task of controlling a complex enterprise is routinely delegated to several agents to cope with the curse of dimensionality. The agents, having limited abilities and views of the enterprise, typically compete with one another only to reach suboptimal decisions. Whatever the enterprise, the work of its agents can be modeled by a dynamic game. This paper illustrates how altruistic agents can drive the decisions to optimal attractors for a family of games. Specifically, it develops an algorithm to find optimal altruistic responses to improve convergence to, and location of, attractors in games arising from the optimization of quadratic functions. Further, the paper provides evidence that the agents can learn altruistic responses from past experience

Keywords— game theory; distributed control; optimization; automatic learning; linear matrix inequalities.

1 Distributed Decision Making and Control

The degree of distributed decision-making and control of today's systems, networks, and organizations is unprecedented. More than ever before, the decisions are entrusted to sizeable numbers of agents spread over vast areas, which compete and collaborate in using the resources to achieve their goals. The Internet, for instance, provides many services to the end-user through the aggregate effort of millions of agents, each with a share of the decisions and resources. Two forces that are driving the distribution of decision-making and control are:

- **Market pressure.**

  The break up of monopolies into open markets, such as telecommunications and transportation conglomerates, has resulted into lower prices and improved quality of services to customers around the globe. The benefits of competition are now demanded in segments as diverse as electric power networks and governmental agencies.

- **Technological constraints.**

  As systems are integrated within and across countries, the need of distribution of decision-making and control becomes more pronounced. The centralized control approach is limited in the size of the systems it can handle, even with the best of the foreseeable technology of hardware and software.

Regardless of the system, its operation by distributed agents can be modeled as a dynamic game (Talukdar and Camponogara, 2001). Consider the task of operating a system or enterprise. In the standard approach to the distribution of decision-making and control, the overall task is divided into a set of coupled subtasks, each consisting of an optimization problem that is assigned to a distributed agent (Camponogara, 2000; Camponogara et al., 2001). The dynamic game arises from the iterative, competitive effort of each agent at doing the best for itself, that is, affecting its variables to solve its problem whose outcome depends on the decisions of the other agents. As the agents react to one another's decisions, they trace a path in decision space that, if convergent, reaches an equilibrium point (Nash) that can be far off the best solutions, the so-called Pareto solutions (Basar and Olsder, 1999). (An ideal and centralized agent, one with unlimited computational power and full authority, has the capacity of finding

Pareto solutions.) Thus, two issues in the distributed operation of an enterprise are the convergence of its agents' decisions to attractors and their relative location. To this end, this paper delivers ways to improve convergence to and location of attractors by implementing altruistic behavior in the agents, whereby they account for the goals of the others and, from their interactions, learn the best altruistic responses. Even though the focus is on games originating from unconstrained quadratic problems, the results shed light on the issues of concern and seem extendable to more general games.

2 Quadratic Games

The potential benefits of altruism, in terms of improved convergence to Nash equilibria and their locations, will be confined to games whose agents optimize quadratic functions, hereafter denoted by quadratic games. More specifically, each of M agents solves a problem of the following form:

$$P_m: \ \underset{x_m}{\text{Min}}\ f_m(x) = ½x^T A_m x + b_m^T x + c_m$$

where:

- $x \in R^N$ is the vector with the decision variables of all agents,
- x_m is the vector with the decision variables over which agent-m has exclusive authority,
- f_m is the agent's objective function with the assumption that A_m is positive definite and, without loss of generality, symmetric.

After breaking up A_m into submatrices, the problem of the m-th agent becomes:

$$P_m: \ \underset{x_m}{\text{Min}}\ f_m(x) = ½ \sum_{i=1}^{M} \sum_{j=1}^{M} x_i^T A_{mij} x_j + \sum_{i=1}^{M} b_{mi}^T x_i + c_m$$

A competitive agent does the best for itself: given the values of the other agents' variables, it chooses values for its variables that solve its problem. These optimal values constitute the agent's reaction and the set of all reactions is its reaction set (Basar and Olsder, 1999). The concept of reaction set is of principal importance—it spells out the agent's decision-making process. For the m-th agent, these concepts can be formalized as follows.

- The exogenous variables, $y_m = [x_n \mid n \neq m]$, are the decision variables of the other agents.
- Agent-m's reaction, $R_m(y_m)$, is defined as: $R_m(y_m) = \underset{x_m}{\text{Argmin}}\ \{f_m(x_m,y_m)\}$, more specifically, $R_m(y_m) = -(A_{mmm})^{-1}[\sum_{n\neq m} A_{mmn} x_n + b_{mm}]$.
- The agent's reaction set is $R_m = \{R_m(y_m) \mid \text{for all } y_m\}$.

In reacting to one another's decisions, the agents iteratively update their variables tracing a trajectory in decision space that, if convergent, arrives at a fixed point referred to as Nash point, $\mathcal{N}$. At a Nash point, the competitive agent does not deviate from its decision so long as the others retain their decisions.

The agents can iterate in different ways. They can enforce precedence to work serially, synchronize the decisions to work in lock-step parallel, or simply iterate asynchronously. Although we concentrate on synchronous parallel iterations, some of the results are readily applicable to asynchronous iterations and the others are probably extendable. Putting all the reactions together, and letting $x(t)$ be the vector of decisions at iteration t, results in the following system of equations:

$$x_m(t+1) = R_m(y_m(t)) \quad \text{for } m = 1, \ldots, M \quad \text{or,}$$

more compactly, $x(t+1) = G(x(t))$.

The operator G, referred to as iteration function, describes the synchronous parallel work of the agents. A Nash point x^* is precisely a fixed point of G, i.e., $x^* = G(x^*)$. In the literature, it is well known that G has a unique fixed point and that convergence to it is guaranteed if G induces a contraction mapping for any norm $\|\bullet\|$—more formally, if $\|G(y) - G(x)\| \leq \gamma\|y - x\|$ for all x, y and $0 \leq \gamma < 1$ (Ortega and Rheinboldt, 1970). Asynchronous convergence is assured if G is a contraction mapping for the ∞-norm, $\|\bullet\|_\infty$ (Bertsekas, 1983; Pyo, 1985).

To better understand the issues of operating an enterprise with a network of agents, let P be the operating problem and $\{P_m\}$, its decomposition into a set of coupled problems, with the m-th agent assigned to P_m. For the iterative work of these agents to be meaningful, their decisions should converge to attractors and, for it to be satisfactory, the attractors should come close to Pareto solutions. A solution x^P is Pareto if it is not dominated by any other solution—i.e., if it does not exist a solution x such that $f_m(x) \leq f_m(x^P)$ for all m and $f_m(x) < f_m(x^P)$ for some m (Miettinen, 1999). In general, a centralized agent with full authority is in a position to find Pareto solutions, provided that it has enough computational power. The indiscriminate use of competitive agents to solve $\{P_m\}$ invariably results in suboptimal solutions, potentially falling far from the Pareto set, $\mathcal{P}$. Altruistic agents, those sympathetic to the goals of the others, however, hold the potential of drawing the attractors close to $\mathcal{P}$. The ultimately altruistic agent discards its own goal to take on the goals of the others.

In the work reported here, an agent implements altruism by assigning weights to the terms appearing in both its objective function and those of the other agents. More specifically, in the context of quadratic games, the m-th agent can become altruistic by augmenting its problem as follows:

$$P_m: \ \underset{x_m}{\text{Min}} \ f_m = \sum_{k=1}^{M} \{\alpha_{mk} x^T A_k x/2 + \beta_{mk}\, b_k^T x + \gamma_{mk} c_k\}$$

In a refined form, agent-m's problem is expressed as:

$$P_m: \underset{x_m}{\text{Min}} \ \sum_{k=1}^{M} \{\alpha_{mk} \sum_{i=1}^{M}\sum_{j=1}^{M} x_i^T A_{kij} x_j + \sum_{i=1}^{M} \beta_{mk}\, b_{ki}^T x_i + \gamma_{mk} c_k\}$$

Carrying the altruistic factors over the reactions, results in the following reaction function for this agent:

$$R_m(y_m) = -(\sum_{k=1}^{M} \alpha_{mk} A_{kmm})^{-1} \sum_{k=1}^{M} [\ \sum_{n\neq m} \alpha_{mk} A_{kmn} x_n + \beta_{mk}\, b_{km}]$$

In what follows, we identify the influence of the altruistic factors on both convergence to attractors and their locations. Before doing so, we group these factors in sets as follows:

- the factors of the m-th agent are $\alpha_m = \{\alpha_{mk}\}$, $\beta_m = \{\beta_{mk}\}$, and $\gamma_m = \{\gamma_{mk}\}$, and
- the factors of the entire game are $\alpha = \{\alpha_m\}$, $\beta = \{\beta_m\}$, and $\gamma = \{\gamma_m\}$.

Lemma 1. Only the factors from α influence the contraction property of the iteration function G.

Proof. Aggregate the reaction functions of the M altruistic agents to obtain the iteration function, G. Notice that G can be expressed as $G(x) = A(\alpha)x + b(\alpha,\beta)$. For any vector-norm, $\|\bullet\|$, $\|G(y) - G(x)\| = \|A(\alpha)y + b(\alpha,\beta) - A(\alpha)x - b(\alpha,\beta)\| = \|A(\alpha)y - A(\alpha)x\|$. ■

Lemma 2. The factors from β influence the location of the fixed point of G, without interfering with its contraction properties.

Proof. From Lemma 1, it follows that the factors in β do not affect contraction. The fixed point x^* can be determined by solving the equation $x^* = G(\alpha,\beta,x^*) = A(\alpha)x^*+b(\alpha,\beta)$, resulting in $x^*(\alpha,\beta) = (I-A(\alpha))^{-1}b(\alpha,\beta)$. Therefore, β can be utilized to relocate the attractor. ■

The upshot is that the altruistic agents can draw the decisions to attractors (by tweaking the values of α) and place these attractors at locations that are best for the agents as a whole (by tweaking the values of β).

3 Improving Convergence to Attractors

This section is concerned with the computation of values for α that ensure parallel convergence to attractors. To this end, it delivers a convex optimization problem with LMIs (linear matrix inequalities) (Boyd et al., 1994) that, if feasible, yields an iteration function inducing a contraction mapping for the 2-norm, $\|\bullet\|$.

Disregarding the altruistic factors in β, the iteration function becomes $G(\alpha) = A(\alpha)x + b(\alpha)$. By breaking $A(\alpha)$, it can be rewritten as $G(\alpha) = B(\alpha)^{-1}[C(\alpha)x + d]$. With this notation, a problem for finding contracting, altruistic factors can be posed as follows.

$$\begin{aligned} \text{PA:}\quad & \underset{\alpha,\ \lambda_B,\ \lambda_C}{\text{Maximize}} \ \ \lambda_B - \lambda_C \\ & \text{Subject to:} \\ & \qquad \lambda_B I - B(\alpha) < 0 \\ & \qquad \lambda_B > \lambda_C \\ & \qquad \lambda_B > 1 \\ & \qquad \begin{bmatrix} \lambda_C I, & C(\alpha)^T \\ C(\alpha), & I \end{bmatrix} \geq 0 \end{aligned}$$

Notice that $\lambda_B \leq \lambda_{\min}(B(\alpha))$ and $\lambda_C \geq \|C(\alpha)\|^2$, where: $\|A\| = \text{sqrt}(\lambda_{\max}(A^T A))$; $\lambda_{\max}(A)$ is the maximum eigenvalue of A; and $\lambda_{\min}(A)$ is its minimum.

Lemma 3. If $(\alpha,\lambda_B,\lambda_C)$ is a feasible solution to PA, then $G(\alpha)$ induces a contraction mapping for the 2-norm, $\|\bullet\|$.

Proof. By the symmetry of $B(\alpha)$, $\lambda_B < \lambda_{\min}(B(\alpha))$. This fact, together with $\lambda_B > 1$, leads to the following inequalities $\lambda_{\min}(B(\alpha)^2) = \lambda_{\min}(B(\alpha))^2 > \lambda_{\min}(B(\alpha)) > \lambda_B$. By Schur's complement, $\lambda_C I - C(\alpha)^T C(\alpha) > 0$ which implies that $\lambda_C > \lambda_{\max}\{C(\alpha)^T C(\alpha)\}$. Thus, $\lambda_C > \|C(\alpha)\|^2$.

Putting everything together, results in $\|A(\alpha)\|^2 = \|B(\alpha)^{-1}C(\alpha)\|^2 \leq \|B(\alpha)^{-1}\|^2.\|C(\alpha)\|^2 \leq \|B(\alpha)^{-1}\|^2\lambda_C = \lambda_{\max}(B(\alpha)^{-2})\lambda_C = \lambda_C/\lambda_{\min}(B(\alpha)^2) < \lambda_C/\lambda_B < 1$. Thus, $\|A(\alpha)\|^2 < 1$ for $A(\alpha) = B(\alpha)^{-1}C(\alpha)$. It is a simple exercise to verify that $\|A(\alpha)\| < 1$ defines an iteration function $G(\alpha)$ which induces a contraction mapping under the 2-norm. ■

4 Improving Location of Attractors

This sections illustrates how altruistic agents can converge to attractors located in the Pareto set of the game given in Fig. 1 by, first, applying the procedure of the preceding section to compute values for α and, second, developing and applying a procedure to compute values for β.

The optimization package sdpsol (Wu and Boyd, 2000) was used to solve the PA arising from the game of Fig. 1. It outputs the following values for α: α_{11} = 0.004834; α_{12} = 0.015966; α_{21} = 0.021461; α_{22} = 0.0. This α, with standard β, generates a $G(x) = A(\alpha)x + b(\alpha,\beta)$ inducing a contraction mapping with the fixed point x^* at (15.6668, 25.9597).

Finding optimal values for β is a two-stage procedure. First, compute a point x^p inside the Pareto. Second, solve the following convex optimization problem:

$$\text{PP:} \quad \underset{\beta}{\text{Minimize}} \ \|x^* - x^P\|^2$$
$$\text{Subject to:} \quad x^* = (I - A(\alpha))^{-1} b(\alpha,\beta)$$

The optimal values of β for PP are: β_{11} = 78.80; β_{12} = –4.27; β_{21} = –3.927; β_{22} = 36.45. Fig. 2 shows the resulting reaction curves of the agents, with the computed values of α and β. The new reaction curves intercept each other at the Pareto point x^P.

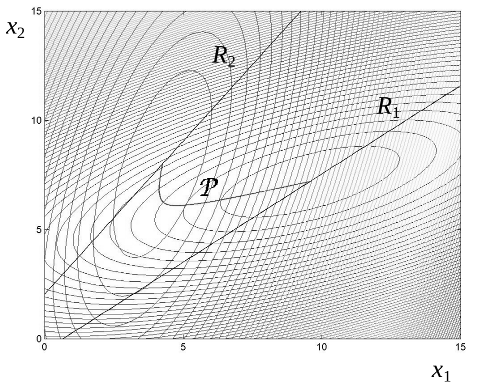

Figure 1. The objective landscape and reaction sets of a divergent 2-player game. (The game consists of the following problems:

$$P_1: \underset{x_1}{\text{Min}} \, f_1 = 9.11215x_1^2 - 22.5402x_1x_2 + 35.88785x_2^2 - 11.9718x_1 - 301.258x_2$$

$$P_2: \underset{x_2}{\text{Min}} \, f_2 = 47.0034x_1^2 - 22.428x_1x_2 + 7.99655x_2^2 - 219.8309x_1 - 32.6516x_2)$$

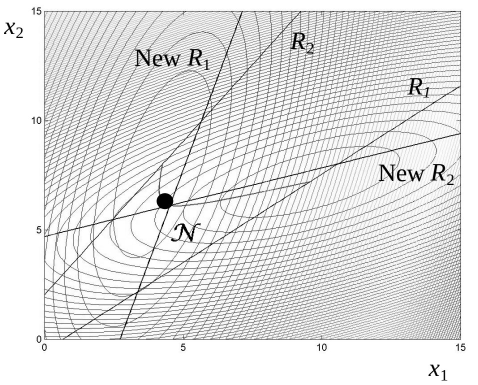

Figure 2. The reaction sets of altruistic agents for the divergent 2-player game. (The decisions of these altruistic agents are drawn to an attractor belonging to the Pareto set, x^P = (4.5179, 6.1197).).

5 Learning Altruistic Responses

The algorithms presented for calculating altruistic factors serve only the purpose of designing testing scenarios. If the agents relinquished their autonomy to a central agency, the need of altruism would vanish—the central agency would solve the overall problem to find a Pareto solution and pass the decisions down to the agents that, in turn, would implement them without contesting. It is imperative that the decisions remain shared among the distributed agents for, otherwise, the problems will be limited in size by the computational power of the central agency. Thus, we seek means by which the agents figure out altruistic responses, while working autonomously and sharing the burden of decision-making. Automatic learning techniques stand as promising ways of harnessing the agents' abilities. Not infrequently, these techniques are conceived from natural behavior or intuitive responses and utilize probabilistic or optimization frameworks, such as reinforcement learning, Bayesian nets, and maximum likelihood estimation to name a few (Mitchell, 1997).

Herein, we develop a search-based learning algorithm, akin to steepest descent (Nocedal and Wright, 1999), to empower altruistic agents to draw the decisions towards the Pareto set. The focus is on the location of the attractors, with the assumption that the games are convergent. This proposal is tested in a simple 2-player game and the results are reported thereafter.

Agent-m's Learning Algorithm

- Let t be the iteration number and set t to 0.
- Let $\beta_m(t) = \{\beta_{mk}(t)\}$ be the initial values of the altruistic factors, possibly disregarding the other agents' objectives.
- Let $\{\gamma_{mk}(t)\}$ contain bounds for changes on the values of the elements of β_m.
- δ is a decay factor, $0 < \delta < 1$.
- Iterate until the agents arrive at an attractor $x^*(t)$.
- Let x_{best} be the best attractor reached thus far, and set $x_{\text{best}} = x^*(t)$.
- Loop for ever
  - o Set $\beta_{mk}(t+1) = \beta_{mk}(t) + \text{rand}(-\gamma_{mk}(t),\gamma_{mk}(t))$ $\forall$ k.
  - o Iterate using altruistic factors from $\beta_m(t+1)$ to reach the attractor $x^*(t+1)$.
  - o If $\{\Sigma\ f_m(x(t+1)) < \Sigma\ f_m(x_{\text{best}})\}$ or $\{x^*(t+1)$ dominates $x_{\text{best}}\}$, then let $x(t+1)$ become the new x_{best},
    otherwise let $\beta_{mk}(t)$ become $\beta_{mk}(t+1)$ $\forall$ k
  - o $\gamma_{mk}(t+1) = \delta.\gamma_{mk}(t)$ $\forall$ k

The sequence of attractors obtained by two altruistic agents is reported in Table 1. The final attractor dominates the initial one, thereby making both agents better off. Fig. 3 depicts the test game, illustrating the contour lines and specifying the objective functions of the agents. Further, it shows the initial reaction sets and the ones the agents reach if they behave altruistically.

Tables 2 and 3 report the sequence of attractors reached by the agents when, respectively, agent-1 and agent-2 behave altruistically. In either case, and in all subsequent attractors, the altruistic agent is worse off while the competitive one is better off. Collectively, however, the agents operate at lower cost—the degradation experienced by the altruistic agent translates into higher benefits to the competitive agent.

Table 1. The series of best attractors reached by two altruistic agents.

Nash Sequence	Agents' objectives		Overall
	f_1	f_2	$f_1 + f_2$
1	–212.8966	–221.0169	–433.9135
2	–194.4710	–337.6601	–532.1310
3	–177.9596	–368.4986	–546.4582
4	–238.1919	–320.7309	–558.9227
5	–244.7565	–328.8900	–573.6464

Table 2. The series of best attractors reached by the agents, with an altruistic agent-1 and a competitive agent-2.

Nash Sequence	Agents' objectives		Overall
	f_1	f_2	$f_1 + f_2$
1	–212.9293	–221.1606	–434.0899
2	–220.9213	–246.2498	–467.1711
3	–153.1794	–369.1648	–522.3442
4	–189.7412	–342.2410	–531.9823
5	–194.8311	–337.2270	–532.0581

Table 3. The series of best attractors reached by the agents, with a competitive agent-1 and an altruistic agent-2.

Nash Sequence	Agents' objectives		Overall
	f_1	f_2	$f_1 + f_2$
1	–212.9376	–221.1657	–434.1032
2	–259.0347	–232.6204	–491.6550
3	–288.6574	–218.3329	–506.9903
4	–288.6436	–218.3467	–506.9903
5	–289.9235	–217.0724	–506.9959
6	–289.3253	–217.6967	–507.0220

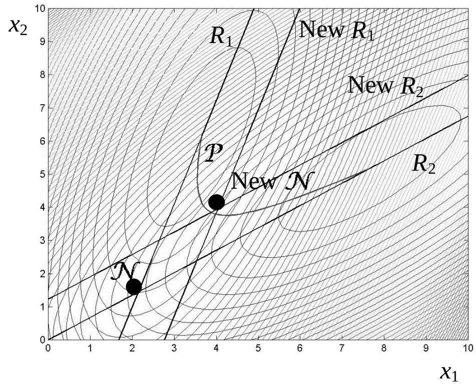

Figure 3. The objective landscape and reaction sets of competitive agents of a 2-player game, together with the reaction sets reached by altruistic agents. (The game consists of the following problems:

$$P_1: \min_{x_1} f_1 = 44.76x_1^2 - 28.87x_1x_2 + 10.24x_2^2 - 150x_1 - 20x_2$$
$$P_2: \min_{x_2} f_2 = 19.49x_1^2 - 34.48x_1x_2 + 25.51x_2^2 - 120x_1$$
).

6 Concluding Remarks

The operation of an enterprise with a network of distributed agents can be viewed as a dynamic game—the competitive agents, each with a share of the decision variables and an optimization task, iteratively react to each other's decisions until they arrive at an attractor, which typically induces a suboptimal solution. This paper has proposed the embedding of altruistic behavior in the agents to, first, improve convergence of their decision processes to attractors and, second, pull these attractors closer to optimal Pareto solutions. Even though the approach has been somewhat abstract, and confined to simple quadratic games, it sheds light on relevant issues and serves as a baseline along which improvements can be crafted for real-world enterprises.

Acknowledgments

This research was supported in part by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), under grant number 68.0122/01-0, and by the EPRI/DoD Complex Interactive Networks/Systems Initiative under contract number EPRI-W08333-05.

References

Basar, T. and Olsder, G.J. (1999). *Dynamic Noncooperative Game Theory*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Bertsekas, D.P. (1983). Distributed Asynchronous Computation of Fixed Points. *Mathematical Programming*, Vol. 27, pp. 107-120.

Boyd, S.; El Ghaoui, L.; Feron, E. And Balakrishnan, V. (1994). *Linear Matrix Inequalities in System and Control Theory*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Camponogara, E. (2000). *Controlling Networks with Collaborative Nets*. Ph.D. Dissertation, ECE Department, Carnegie Mellon University, Pittsburgh, PA, USA.

Camponogara, E.; Jia, D.; Krogh, B.H. and Talukdar, S.N. (2001). Distributed Model Predictive Control. Accepted to appear in *IEEE Control Systems Magazine*.

Miettinen, K.M. (1999). *Nonlinear Multiobjective Optimi-zation*. Kluwer Academic Publishers.

Mitchell, T.M. (1997). *Machine Learning*. McGraw-Hill.

Nocedal, J. and Wright, S.J. (1999). *Numerical Optimization*. Springer-Verlag.

Ortega, J.M. and Rheinboldt, W.C. (1983). *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York, USA.

Pyo, S.S. (1985). *Asynchronous Algorithms for Distributed Processing*. Ph.D. Dissertation, ECE Department, Carnegie Mellon University, Pittsburgh, PA, USA.

Talukdar, S.N. and Camponogara, E. (2001). Network Control as a Distributed, Dynamic Game. *Proc. 34th Hawaii Int. Conf. on System Sciences*. (*Best Paper Award in the Complex Systems Track*.)

Wu, S.-P. and Boyd, S. (2000). *SDPSOL: a parser/solver for determinant maximization (MAXDET) and semidefinite programming (SDP) problems with matrix structure*. Department of Electrical Engineering, Stanford University, CA, USA. (http://www.stanford.edu/~boyd)