ELSEVIER

Computers and Chemical Engineering 23 (1999) 1021–1030

Computers & Chemical Engineering

www.elsevier.com/locate/compchemeng

Robustness margin computation for large scale systems

Richard D. Braatz *, Evan L. Russell

Department of Chemical Engineering, Large Scale Systems Research Laboratory, University of Illinois at Urbana-Champaign, 600 South Mathews Avenue, Box C-3, Urbana, IL 61801-3782, USA

Received 6 January 1997; accepted 23 March 1999

Abstract

Large scale systems have large numbers of inputs and outputs, and include whole chemical plants as well as some unit operations, such as paper machines, polymer film extruders, and adhesive coaters. The importance of ensuring robustness of the closed loop system to model uncertainties increases as the process dimensionality increases; hence developing algorithms for computing robustness margins for large scale systems is of immense practical importance. Computational complexity is a tool of computer scientists which has had impact in understanding large scale optimization problems, both theoretically and in terms of finding computational solutions. Computational complexity theory is used to determine the level of accuracy and computational speed that are obtainable by algorithms for computing robustness margins, and as to which algorithms are likely for providing practical robustness margin computation for large scale systems. © 1999 Elsevier Science Ltd. All rights reserved.

Keywords: Large scale systems; Robustness; Closed loop system; Algorithms; Computational complexity

1. Introduction

Robust control for single loop and 2×2 chemical processes has been studied extensively over the past 15 years (Palmor & Shinnar, 1981; Song, Fisher & Shah, 1984; Laughlin, Jordan & Morari, 1986; Yousefpor, Palazoglu & Hess, 1988; Doyle, Packard & Morari, 1989; Kozub, MacGregor & Harris, 1989; Morari & Zafiriou, 1989; Campo & Morari, 1990; Zafiriou & Marchal, 1991; Arkun & Calvet, 1992; Schaper, Seborg & Mellichamp, 1992; Amann & Allgower, 1994; Horn, Arulandu, Gombas, VanAntwerp & Braatz, 1996; Skogestad & Postlethwaite, 1996; Stryczek & Brosilow, 1996). On the other hand, whole chemical plants as well as some unit operations, such as paper machines, polymer film extruders, and adhesive coaters have large numbers of inputs and outputs. More recently chemical engineers have become interested in applying rigorous control systems techniques to these *large scale systems* (Ricker & Lee, 1995; Dave, Willig, Kudva, Pekny & Doyle, 1997; Rao, Campbell, Rawlings & Wright, 1997; Rigopoulos, Arkun & Kayihan, 1997; Russell, Power & Braatz, 1997; Featherstone & Braatz, 1998a,b; Russell & Braatz, 1998a,b; VanAntwerp & Braatz, 1998). The importance of ensuring robustness of the closed loop system to model uncertainties increases as the process dimensionality increases (Featherstone & Braatz, 1997; Russell & Braatz, 1998a; Featherstone & Braatz, 1998b); hence developing algorithms for computing robustness margins for large scale systems is of immense practical importance. This explains why researchers have spent many man-centuries working to derive efficient numerical algorithms for computing robustness margins (a *man-century* refers to one man working 40 h per week for one century). In this paper *computational complexity theory* (described below) is used to determine the level of accuracy and computational speed that are obtainable by algorithms for computing robustness margins, and as to which algorithms are likely to provide practical robustness margin computation for large scale systems.

The purpose of computational complexity theory is to characterize the inherent difficulty of calculating the solution for a problem under study. This theory can be used to characterize computational problems as being in one of two classes: *P* and *NP-hard*. The class P refers to problems in which the exact time needed to solve the

* Corresponding author. Tel.: +1-217-3335073; fax: +1-217-3335052.

E-mail address: braatz@uiuc.edu (R.D. Braatz)

PII: S0098-1354(99)00268-9

problem can be bounded by a single function which is polynomial in the amount of data needed to define the problem. Such computational problems are said to be solvable in *polynomial-time*, and include linear and convex quadratic programs.

Although the exact consequences of a problem being NP-hard is still a fundamental open question in the theory of computational complexity, it is generally accepted that a problem being NP-hard means that it cannot be computed in polynomial time in the worst case. It is important to understand that being NP-hard is a property of the problem itself, not of any particular algorithm. Whether a problem is polynomial-time or NP-hard determines the highest computational efficiency which can be expected by any algorithm, and as to which classes of algorithms can provide practical solutions to the problem. For example, an engineer may look to interior point algorithms (Nesterov & Nemirovskii, 1994) for computing the solution to a polynomial-time optimization problem; whereas would consider using approximations, heuristics, or branch-and-bound techniques for computing the solution for an NP-hard optimization problem (Papadimitriou & Steiglitz, 1982; McCormick, 1983; Ryoo & Sahinidis, 1996; VanAntwerp, Braatz & Sahinidis, 1998). The solution to an NP-hard optimization problem takes much longer to compute, and for some large scale problems even a supercomputer may be unable to find the global solution to high accuracy within a reasonable amount of time (Garey & Johnson, 1983).

It is known that the exact computation of the robustness margin is NP-hard (Braatz, Young, Doyle & Morari, 1994). Although the general μ recognition problem is NP-hard, special cases (that is, with restrictions on the structure or field of M or Δ) may be simpler to compute. For example, when the M matrix is restricted to be rank one, the calculation of μ has sublinear growth in problem size, irrespective of the perturbation structure (Chen, Fan & Nett, 1991). In this manuscript we provide the simplest known proof that robustness margin computation is NP-hard even when all of the uncertainties are represented as independent real parameter variations. This is followed by a related result that robustness margin computation is NP-hard for the case of one complex scalar uncertainty and independent real parameter variations. These results indicate that computing the robustness margin even for systems with very simple uncertainty structures is NP-hard.

The main result of this manuscript is that even approximating the robust performance margin is a hard computational problem. This result is especially important in light of recently-proposed techniques for robustness margin computations using Monte Carlo methods and other methods of testing only a subset of the plants in the uncertainty set (Barmish & Lagoa, 1997). Although such algorithms are of interest, the main result implies that there exist large scale uncertain systems for which such algorithms will grossly underestimate the size of the robust performance margin. Practical algorithms for computing robustness margins should be based on rigorous upper and lower bounds that are tight for *typical* systems, although the main result implies that such bounds will not be tight for some problems.

All proofs require only basic linear algebra and follow from relating robustness margin computation with some specialized nonconvex programs. Given that computational complexity theory is not usually studied by chemical engineers, we provide a brief introduction to its basic principles. This is followed by the results, and a discussion of how the results relate to other results in the literature.

1.1. Notation

In what follows, matrices are upper case and vectors and scalars are lower case. The set of real numbers is R; the set of complex numbers is C, and the set of rationale is Q. The Euclidean 2-norm of vector x is defined by $\|x\|_2 = \sqrt{x^T x}$, whereas the vector ∞-norm of x is defined by $\|x\|_\infty = \max|x_i|$. The maximum singular value of matrix A is represented by $\overline{\sigma}(A)$, and I_r is the $r \times r$ identity matrix. An '0' will be used to represent either zero or a matrix of zeros.

2. Introduction to computational complexity theory

Here we provide a description of computational complexity theory that is concise and useful for practitioners. This includes a discussion of how this theory can be applied to ϵ-approximation problems, as such descriptions are not provided in the standard textbooks (Papadimitriou & Steiglitz, 1982; Garey & Johnson, 1983).

Computational complexity theory allows a characterization of the inherent difficulty of calculating the solution for a problem under study. Problems (or equivalent versions of the same problem) are generally characterized as being in one of two classes: *P* and *NP-hard*. The class P refers to problems in which the exact time needed to solve the problem can always be bounded by a single function which is polynomial in the amount of data needed to define the problem. Such problems are said to be solvable in *polynomial time*. Although the exact consequences of a problem being NP-hard is still a fundamental open question in the theory of computational complexity, it is generally accepted that a problem being NP-hard means that it cannot be computed in polynomial time in the worst case. It is important to understand that being NP-hard

is a property of the problem itself, not of any particular algorithm. It is also important to understand that having a problem be NP-hard does not imply that practical algorithms are not possible. Practical algorithms for NP-hard problems exist and typically involve approximation, heuristics, branch-and-bound, or local search (Papadimitriou & Steiglitz, 1982; Garey & Johnson, 1983; McCormick, 1983; Ryoo & Sahinidis, 1996). Determining whether a problem is polynomial-time or NP-hard tells the large scale systems engineer what kind of computational efficiency can be expected by any algorithm, and what kinds of algorithms to investigate for providing practical solutions to the problem.

2.1. Three versions of an optimization problem

Consider the optimization problem

$$\sup_{x\in\chi_w} f_w(x), \tag{1}$$

where w as the vector of data required to uniquely specify an instance of an optimization problem, and the feasible region χ_w and objective function $f_w(x)$ are functions of the problem data. For the optimization problems considered in this manuscript, the feasible region χ_w will be non-empty, and compact (closed and bounded). Then the supremum is achieved by at least one x in the feasible region:

$$\bar{f}_w = \sup_{x\in\chi_w} f_w(x) = \max_{x\in\chi_w} f_w(x) \tag{2}$$

An optimization problem can be written in three closely-related but different versions (Papadimitriou & Steiglitz, 1982):

P1. The *optimization version*: find the optimal feasible solution x, that is, such that $f_w(x)$ is maximized.

P2. The *evaluation version*: Find the value $\bar{f}_w = f_w(\hat{x})$ of the optimal solution.

P3. The *recognition version*: Given k, is there a feasible solution x such that $f_w(x) \geq k$?

The recognition version can also be written as: 'Given k, is $f_w(\hat{x}) \geq k$?' The recognition version is important for studying the complexity of the problem, because it is the type of problem traditionally studied by the theory of computation. Unlike the first two versions, the recognition version is a *question*, which can be answered by *yes* or *no*.

The *recognition version* is no harder than the *evaluation version* (P3 ≤ P2), since we can immediately deduce the answer to the *recognition version* from the answer to the *evaluation version*. In other words, the evaluation problem is at least as difficult to solve as the recognition problem, so if a recognition problem is NP-hard, then the corresponding evaluation problem must also be difficult.

If $f_w(x)$ is easy to compute, then the evaluation version cannot be significantly harder than the optimization version (P2 ≤ P1). Thus the ordering from least difficult to most difficult is P3 ≤ P2 ≤ P1.

2.2. The classes P and NP

The classes P and NP refer to the *recognition versions* only. The class P refers to the class of recognition problems that can be answered by a polynomial-time algorithm. An example would be the linear programming recognition problem.

Researchers believe that class NP is richer than the class P (P ≠ NP). For a problem to be in NP, it is not required that it can be answered in polynomial time by an algorithm. It is only required that, if x is a *yes* instance of the problem, then this x can be checked in polynomial time for validity. In other words, a problem is in NP if we can calculate $f_w(x)$ and check that $x\in\chi$ in polynomial time. This condition is usually referred to as *polynomial-time verifiability*. This definition automatically implies that P is a subset of NP.

If a recognition problem is in the class NP, then the complement is in the class co-NP. The co-NP problem related to our optimization problem would be: 'Given k, is $f_w(\hat{x}) < k$?' It is unclear how to verify if this problem is answered by *yes*, except by solving the evaluation version of the optimization problem. This certainly appears to be more difficult than verifying if the recognition version is in NP. For this reason researchers believe that NP ≠ co-NP.

2.3. NP-complete and NP-hard problems

NP-complete problems are the hardest problems in NP. Examples of problems that are NP-complete are the traveling salesman problem, the max-cut problem, and the indefinite quadratic programming problem. Almost all researchers believe that NP-complete problems are harder than P problems, but there is no proof.

Any problem that is at least as hard as an NP-complete problem is said to be *NP-hard*. An *NP-hard* problem can refer to much broader classes of problems than recognition problems (Garey & Johnson, 1983). In particular, if the recognition version of an optimization is NP-complete, then the corresponding evaluation and optimization versions are NP-hard.

2.4. Computational complexity of ϵ-approximation problems

Because we have assumed that the feasible region χ_w, in Eq. (1) is non-empty, and compact, the infimum is achieved by at least one x in the feasible region:

$$\underline{f}_w = \inf_{x\in\chi_w} f_w(x) = \min_{x\in\chi_w} f_w(x) \tag{3}$$

Then the *ϵ-approximation* problem for Eq. (1) is to compute a value $\tilde{f}_w \in R$ for which

$$|\bar{f}_w - \tilde{f}_w| \leq \epsilon|\bar{f}_w - \underline{f}_w|. \tag{4}$$

This definition is preferable to others in the literature in that it is invariant to translation and dilation (that is, multiplying by and adding a constant) of the objective function (Vavasis, 1992; Bellare & Rogaway, 1993; Vavasis, 1993). Quantifying the optimization objective in different units does not affect the quality of the approximation as measured by Eq. (4).

A 0-approximation algorithm provides the exact optimal solution, while a 1-approximation algorithm need only find any feasible point and compute its objective. We will study the existence of polynomial-time algorithms for computing an ϵ-approximation. Let n be a measure of the quantity of data needed to describe an instance of an optimization problem (for example, this could be the number of elements in the data vector w, or the number of rows in a matrix which contains most of the data). To provide the strongest results, ϵ will be selected to be a function of n—this allows the consideration of algorithms for which the accuracy of the approximation degrades as the size of the problem (measured by n) increases.

Even when the exact optimization problem is NP-hard, the ϵ-approximation problem can be either easy or hard (Ausiello, D'Atri & Protasi, 1980). For an example of a hard exact optimization problem that has an easy ϵ-approximation algorithm, consider the class of concave quadratic programming problems

$$\min_{Ax \le b} x^T H x + p^T x \tag{5}$$

where H is of rank one. The exact optimization problem is NP-hard (Pardalos & Vavasis, 1991), whereas an algorithm exists that computes an ϵ-approximation in polynomial-time as a function of problem size (Vavasis, 1992). As another example, while the exact knapsack problem is NP-hard, its solution can be approximated in polynomial-time within *every constant factor of the optimum* (Barland, Kolaitis & Thakur, 1996).

The well-known traveling salesman problem is NP-hard (Johnson & Papadimitriou, 1989), and its ϵ-approximation problem is also hard (Sahni & Gonzales, 1976). More specifically, finding a polynomial-time algorithm that approximates the optimum within a given constant is equivalent to establishing that $P = NP$ (Sahni & Gonzales, 1976; Kolaitis & Thakur, 1993; Reinelt, 1994; Yannakakis, 1994). As another example, consider the class of polynomial programs with box constraints

$$\max_{0 \le x \le 1} \sum_{k=1}^{t} \left\{ \left(\prod_{i \in \mathcal{A}_k} x_i \right) \left(\prod_{j \in \mathcal{B}_k} (1 - x_j) \right) \right\} \tag{6}$$

where t is the number of terms in the polynomial objective, n is the number of optimization variables, and for each k, A_k and B_k are disjoint subsets of {1, …, n}. The exact optimization problem is NP-hard. The ϵ-approximation problem is also hard, as stated in the following lemma.

Lemma 1. *(*Polynomial programming with box constraints*)* There is a constant $\gamma > 0$ such that the following is true. If there exists a polynomial-time algorithm that can compute an ϵ-approximation for Eq. (6), where $\epsilon(n) = 1 - n^{-\gamma}$, then P = NP.

Proof 1. Follows directly from the proof of Theorem *3.1* in *(*Bellare & Rogaway, *1993).* QED.

The general consensus in the computational community that $P \neq NP$ implies that the ϵ-approximation problem for polynomial programming with box constraints is also hard. Note that the particular form of Eq. (6) in Lemma 1 considers very weak forms of approximation, as it allows the quality of the approximation to degrade as a function of problem size. In particular, for fixed γ and large n, $\epsilon(n)$ in Lemma 1 approaches one, which only requires that the approximation algorithm be able to find a feasible point and evaluate the corresponding objective function. Thus Lemma 1 indicates that the existence of even a weak polynomial-time approximation algorithm for polynomial programs with box constraints is highly unlikely.

3. Robust stability and robust performance margins

Measuring the robustness of uncertain systems is of immense practical importance. The standard procedure for computing robust stability margins is to first convert the description of the closed loop system into a μ-computation problem (procedures for doing this are described in (Morari & Zafiriou, 1989; Russell & Braatz, 1996; Skogestad & Postlethwaite, 1996; Russell et al., 1997)).

Define the set Δ of block diagonal perturbations by

$$\mathbf{\Delta} = \Big\{ \mathrm{diag}\,\{\delta_1^r I_{r_1}, \ldots, \delta_k^r I_{r_k}, \delta_{k+1}^c I_{r_{k+1}}, \ldots, \delta_m^c I_{r_m}, \Delta_{r_{m+1}}, \ldots, \Delta_{r_l}\} \,\Big|\, \delta_i^r \in R;\ \delta_i^c \in C;\ \Delta_{r_i} \in C^{r_i \times r_i};\ \sum_{i=1}^{l} r_i = n \Big\}. \tag{7}$$

Let $M \in C^{n \times n}$ Then the structured singular value $\mu_\Delta(M)$ is defined as (Fan, Tits & Doyle, 1991)

$$\mu_\Delta(M) \equiv \{0 \text{ if there does not exist } \Delta \in \mathbf{\Delta} \text{ such that } \det(I - M\Delta) = 0,$$

$$\mu_\Delta(M) \equiv \left\{ \left[\min_{\Delta \in \mathbf{\Delta}} \{ \bar{\sigma}(\Delta) \,|\, \det(I - M\Delta) = 0 \} \right]^{-1} \quad \text{otherwise} \right. \tag{8}$$

Without loss of generality we have taken M and each subblock of Δ to be square.

In a similar manner, robust performance margins are computed by converting the system description into a skewed-μ computation problem, where *skewed-μ* has the same definition as μ, except with rows and/or columns of M (Eq. (8)) re-scaled (Smith, 1990; Braatz & Morari, 1991; Fan & Tits, 1992):

$$\mu^s_\Delta(M) \equiv \{0 \text{ if there does not exist } \Delta\in\mathbf{\Delta} \text{ such that } \bar\sigma(\Delta_1) \le 1 \text{ and } \det(I-M\Delta)=0,$$

$$\mu^s_\Delta(M) \equiv \left\{\left[\min_{\Delta\in\mathbf{\Delta}}\left\{\bar\sigma(\Delta_2)\middle|\bar\sigma(\Delta_1)\le 1;\det\left(I-M\begin{bmatrix}\Delta_1 & 0\\ 0 & \Delta_2\end{bmatrix}\right)=0\right\}\right]^{-1} \text{ otherwise}\right. \quad (9)$$

where $\Delta=\text{diag}\{\Delta_1,\Delta_2\}$ The reader is directed to the citations for more details; here we would just like to note that any algorithm for approximately computing μ can be used to directly approximately calculate skewed-μ and vice versa (Packard, 1988; Skogestad & Morari, 1988; Smith, 1990; Braatz & Morari, 1991; Fan & Tits, 1992; Braatz, Morari & Skogestad, 1996).

4. Computational complexity of exact robustness margin computation

Braatz et al. (1994) and Coxson and DeMarco (1992) proved that exact μ-computation is NP-hard. The Braatz et al. proof did not prove NP-hardness for classes of μ problems with independent real parameter variations. The Coxson and DeMarco proof did address independent real parameter variations, but was lengthy. The following approach addresses independent real perturbations, while providing a much shorter path between a basic computation problem (Lemma 2) and robust margin computation than provided by Coxson and DeMarco.

First we will recall a well-known NP-complete problem (Garey & Johnson, 1983):

Lemma 2. Given an integer $m>0$ and $a\in Q^m$, $\|a\|_2\le 0.1$, with positive entries, the task of determining whether the equation

$$a^T t=\sum_{i=1}^{m} a_i t_i = 0 \quad (10)$$

has a solution t with $t_i\in\{-1,1\}$, $\forall i=1,\ldots,m$, is NP-complete. Nemirovskii (Nemirovskii, 1993) used this result to prove that the following problem is NP-hard.

Lemma 3. Consider the set of matrices

$$M=\left\{\begin{bmatrix}C & z\\ y^T & 1\end{bmatrix}\middle| y,z\in Q^m;\|y\|_\infty\le\alpha;\ \|z\|_\infty\le\alpha\right\} \quad (11)$$

where $C\in Q^{m\times m}$ is a fixed constant invertible matrix with rational elements. The task of determining whether all matrices in M are nonsingular is NP-hard.

Poljak and Rohn (Poljak and Rohn, 1993) had earlier proved a closely related result, but had a much more complicated approach.

We will use Lemma 3 to show that exact μ computation is NP-hard for the class of μ problems with independent real parameter variations. Consider μ with $M\in Q^{n\times n}$, $k\in Q$, and mixed real/complex uncertainty blocks.

Theorem 1 (*NP-hardness of robust stability margin computation with independent real perturbations*). μ-computation with nonrepeated real perturbations is NP-hard.

Proof 2. Consider the class of matrices M defined in Eq. (11). All matrices in M are nonsingular if and only if

$$\det\begin{bmatrix}C & z\\ y^T & 1\end{bmatrix}\ne 0,\quad \forall\|y\|_\infty\le\alpha;\quad \forall\|z\|_\infty\le\alpha \quad (12)$$

$$\Leftrightarrow \det\begin{bmatrix}I_m & z\\ y^TC^{-1} & 1\end{bmatrix}\det\begin{bmatrix}C & 0\\ 0 & 1\end{bmatrix}\ne 0,$$
$$\forall\|y\|_\infty\le\alpha,\quad \forall\|z\|_\infty\le\alpha \quad (13)$$

$$\Leftrightarrow \det\left(I_{m+1}-\begin{bmatrix}0 & -z\\ -y^TC^{-1} & 0\end{bmatrix}\right)\ne 0,$$
$$\forall\|y\|_\infty\le\alpha,\quad \forall\|z\|_\infty\le\alpha \quad (14)$$

Define $z=-\Delta^r_z w$ and $y=-\Delta^r_y w$, where $w=-\alpha k\underline{1}$ and Δ^r_z and Δ^r_y are real diagonal perturbation matrices with independent real scalar uncertainties. Then Eq. (14) is equivalent to

$$\Leftrightarrow \det\left(I-\begin{bmatrix}0 & \Delta^r_z w\\ w^T\Delta^r_y C^{-1} & 0\end{bmatrix}\right)\ne 0,\quad \forall\|\Delta^r_z\|_\infty\le 1/k,$$
$$\forall\|\Delta^r_y\|_\infty\le 1/k \quad (15)$$

$$\Leftrightarrow \det\left(I-\begin{bmatrix}0 & I\\ w^T & 0\end{bmatrix}\begin{bmatrix}\Delta^r_y & 0\\ 0 & \Delta^r_z\end{bmatrix}\begin{bmatrix}C^{-1} & 0\\ 0 & w\end{bmatrix}\right)\ne 0,$$
$$\forall\|\Delta^r_z\|_\infty\le 1/k,\quad \forall\|\Delta^r_y\|_\infty\le 1/k \quad (16)$$

$$\Leftrightarrow \det\left(I-\begin{bmatrix}C^{-1} & 0\\ 0 & w\end{bmatrix}\begin{bmatrix}0 & I\\ w^T & 0\end{bmatrix}\begin{bmatrix}\Delta^r_y & 0\\ 0 & \Delta^r_z\end{bmatrix}\right)\ne 0,$$
$$\forall\|\Delta^r_z\|_\infty\le 1/k,\quad \forall\|\Delta^r_y\|_\infty\le 1/k \quad (17)$$

$$\Leftrightarrow \det\left(I-\begin{bmatrix}0 & C^{-1}\\ ww^T & 0\end{bmatrix}\begin{bmatrix}\Delta^r_y & 0\\ 0 & \Delta^r_z\end{bmatrix}\right)\ne 0,$$
$$\forall\|\Delta^r_z\|_\infty\le 1/k,\quad \forall\|\Delta^r_y\|_\infty\le 1/k \quad (18)$$

$$\Leftrightarrow \mu_{\Delta^r}(M) < k, \tag{19}$$

where

$$M = \begin{bmatrix} 0 & C^{-1} \\ ww^T & 0 \end{bmatrix} \tag{20}$$

and

$$\Delta^r = \begin{bmatrix} \Delta_y^r & 0 \\ 0 & \Delta_z^r \end{bmatrix}. \tag{21}$$

Thus there exists a singular matrix in M if and only if $\mu_{\Delta^r}(M) \geq k$. Determining whether there exists a singular matrix in M is NP-hard by Lemma 3, and the above equivalences show that the nonsingularity task polynomially reduces to a μ problem with real M and independent real scalars. QED.

Models for real systems always have unmodeled dynamics associated with them. Unmodeled dynamics correspond to having at least one complex uncertainty in the μ problem. The next result states that μ-computation is NP-hard for this practically-motivated class of problems.

Corollary 1. *(*NP-hardness of robust stability margin computation with mixed perturbations*)* μ-computation problems with one complex scalar perturbation with the remainder being nonrepeated real scalar perturbations is NP-hard.

Proof 3. Because the μ problem in the proof of Theorem 1 has real M and all perturbations are 1×1, application of Schur's determinant formulae (Zhou, Doyle & Glover, 1995) implies that any one of the real perturbations can be replaced by a complex scalar perturbation without affecting the value of the robustness margin. QED.

That the computation of robust performance margins is also a hard problem immediately follows.

Corollary 2. *(*NP-hardness of robust performance margin computation with independent real perturbations*).* Robust performance margin computation with nonrepeated real scalar perturbations is NP-hard.

Proof 4. The robust performance margin for a system with nonrepeated real scalar perturbations can be written directly in terms of a skewed-μ problem with the same number of nonrepeated real scalar perturbations plus one additional complex perturbation (Smith, 1990; Fan & Tits, 1992). The main loop theorem (Theorem 5.2 of Packard (1988)) allows the polynomial-time construction of $\tilde{M}$ such that

$$\Delta_\Delta^s(M) \geq k \Leftrightarrow \mu_\Delta(\tilde{M}) \geq k, \tag{22}$$

(Fan & Tits, 1992), thus the complexity of the skewed-μ problem is equivalent to the complexity of the μ-computation problem. The μ-computation problem for this perturbation class is NP-hard by corollary 1. QED.

5. Computational complexity of ϵ-approximate robustness margin computation

Here we show that it is hard to compute robust performance margins within a given ϵ.

The first step in our development is to show that the polynomial program with box constraints (Eq. (6)) can be represented as a skewed-μ problem.

Lemma 4 (*Polynomial program with box constraints reduces to robustness margin problems*). The polynomial program with box constraints polynomially reduces to real μ, mixed μ, real skewed-μ, and mixed skewed-μ problems.

Proof 5. The proof is trivial for $k = 0$, so consider the case where $k > 0$. Treat the constraints as uncertainty and the objective function as the performance objective of a robust performance problem. The constraint set is

$$\begin{aligned} \{x | 0 \leq x \leq 1\} &= \{x | x = \bar{x} + \Delta^x w;\ \Delta^x \\ &= \text{diag}[\delta_1^r, \ldots, \delta_n^r];\ \delta_i^r \in [-1/k, 1/k]\}, \end{aligned} \tag{23}$$

where

$$\bar{x} = (1/2)\underline{1} \tag{24}$$

and

$$w = (k/2)\underline{1} \tag{25}$$

The realization algorithm of Russell et al. (1997) computes in polynomial time a matrix M that satisfies

$$\begin{aligned} f_w(x) &:= \sum_{k=1}^{t} \left\{ \left(\prod_{i \in A_k} x_i \right) \left(\prod_{j \in B_k} (1 - x_j) \right) \right\} = F_u(M, \Delta^r) \\ &= |F_u(M, \Delta^r)|, \end{aligned} \tag{26}$$

where $M \in Q^{(m+1) \times (m+1)}$,

$$\begin{aligned} \Delta^r = \{ &\text{diag}\,[\delta_1^r, \ldots, \delta_m^r] | \delta_j^r \\ &\in R;\ \text{each}\ \delta_j^r\ \text{possibly repeated}\}, \end{aligned} \tag{27}$$

$$m = \sum_{k=1}^{t} |\mathscr{A}_k| + |\mathscr{B}_k|, \tag{28}$$

$|A_k|$ refers to the number of elements in the vector of indices A_k, and the linear fractional transformation $F_u(M, \Delta^r)$ is defined by

$$F_u(M, \Delta^r) = M_{22} + M_{21}\Delta^r(I - M_{11}\Delta^r)^{-1}M_{12}. \quad (29)$$

Then the polynomial program with box constraints can be written as

$$\max_{0 \le x \le 1} f_w(x) = \max_{\|\Delta^r\| \le 1/k} \left|F_u(M,\Delta^r)\right| \quad (30)$$

Now apply the robust performance theorem (Doyle, 1982) to give

$$\mu_\Delta(M) \ge k \Leftrightarrow \max_{0 \le x \le 1} \sum_{k=1}^{t} \left\{\left(\prod_{i \in A_k} x_i\right)\left(\prod_{j \in B_k} (1 - x_j)\right)\right\} \ge k, \quad (31)$$

where

$$\Delta = \{\text{diag}\,[\delta^r_1,\ldots,\delta^r_m,\delta^c_{m+1}] | \delta^r_j \in R;\ \delta^c_{m+1} \in C;\ \text{each } \delta^r_j \text{ possibly repeated}\}. \quad (32)$$

Similarly, apply the skewed-μ main loop theorem (Smith, 1990; Fan & Tits, 1992) to give

$$\max_{0 \le x \le 1} f_w(x) = \max_{\|\Delta^r\| \le 1/k} \left|F_u(\tilde{M},\Delta^r)\right| = \mu^s_\Delta(\tilde{M}) \quad (33)$$

where $\tilde{M}$ is equal to M but with some of its rows scaled by k. Using the fact that M is real, the Schur determinant formula can be used to show that in both cases the complex scalar perturbation δ^c can be replaced by a real scalar perturbation.

Since

$$m = \sum_{k=1}^{t} |A_k| + |B_k| \le \sum_{k=1}^{t} n = tn, \quad (34)$$

the quantity of data needed to describe the μ and skewed-μ problems is bounded by a polynomial function of the quantity of data needed to describe the polynomial program with box constraints. QED.

The main result applies Lemma 4 to show that the ϵ-approximation of robust performance margins is hard.

Theorem 2 (ϵ-Approximation of robust performance margins is hard). Consider the skewed-μ problem with: (i) all real scalar uncertainties; (ii) one complex and the rest real scalar uncertainties; or (iii) any superset that contains these uncertainty descriptions. There is a constant $\gamma > 0$ such that the following is true. If there exists a polynomial-time algorithm that can compute an ϵ-approximation for skewed-μ where $\epsilon(n) = 1 - n^{-\gamma}$, then P = NP.

Proof 6. Suppose there exists a polynomial-time ϵ-approximation algorithm for skewed-μ. Apply the algorithm to the skewed-μ problem defined in the proof of Lemma 4. The one-to-one correspondences between Δ^r and x and their objective functions (Eq. (26)) imply that the polynomial-time ϵ-approximation algorithm for skewed-μ could be used to provide a polynomial-time ϵ-approximation for the polynomial program with box constraints. From Lemma 1 this implies that P = NP. QED.

For all practical purposes, computing approximations of μ is just as hard as computing approximations of skewed-μ, since either function can be approximated by the other using bisection (Packard, 1988; Skogestad & Morari, 1988; Fan & Tits, 1992), which is a polynomial-time operation (Khachiyan & Todd, 1993; Boyd, El Ghaoui, Feron & Balakrishnan, 1994).

6. Comparison with other results

In a rather lengthy proof, Coxson and DeMarco (1992) used the results of Poljak and Rohn to prove that exact μ-computation with independent real parameter variations is NP-hard. Braatz et al. (1994) proved NP-hardness of μ for the following classes of systems:

1. systems with pure real perturbations;
2. systems with mixed perturbations in which the complex block enters nontrivially in the robustness margin computation; and
3. systems for which the robustness margin computation is a continuous function of the problem data.

The result for classes 2 and 3 implied that the NP-hardness of μ is rather generic. Theorem 1 and corollary 1 are stronger results than those of Braatz et al. (1994), because they show NP hardness with nonrepeated real perturbations. Toker and Ozbay (1995) proved that μ computation is NP-hard for systems with pure complex scalar uncertainties. Taken together, these results indicate that the is NP-hard for nearly any practical class of perturbations, that is, NP-hardness is an *inherent property of worst-case robust stability analysis*. Corollary 2 indicates that the same is true for robust performance analysis.

The NP-hardness of robustness margin computation is not a property of the problem formulation—the same result holds when the robustness margin is written in terms of k_m (Safonov, 1982; Chiang & Safonov, 1992) or other formalisms. Subclasses for which robustness margin computation is polynomial time are problems where M has rank 1 (Kharitonov, 1978; Bartlett, Hollot & Lin, 1989; Chen et al., 1991; Barmish, 1993; Braatz & Crisalle, 1998), and for processes with special types of transfer function matrices and uncertainty structures (Braatz & VanAntwerp, 1996; Hovd, Braatz & Skogestad, 1996). *Any realistic large scale chemical process will have a corresponding M matrix of much higher rank than one*. The class of systems considered by (Braatz & VanAntwerp, 1996; Hovd et al. 1996) is of industrial relevance, but the results cannot be extended to processes with general transfer function matrices.

The result that computing approximate robust performance margins is hard (Theorem 2) is truly a powerful result. The previous results concerning the computational complexity of robustness margin computation (Demmel, 1992; Nemirovskii, 1993; Poljak & Rohn, 1993; Braatz et al., 1994; Toker & Ozbay, 1995) do not imply this. This is because the requirements for relating the NP-hardness of one approximation problem to another are much stricter than for relating the NP-hardness of their exact problems (see, for example, the discussion on page 134 of Garey and Johnson (1983) for a basic discussion, or the papers (Feige, Goldwasser, Lovasz, Safra & Szegedy, 1991; Arora & Safra, 1992; Arora, Lund, Motwani, Sudan & Szegedy, 1992; Feige & Lovasz, 1992; Bellare & Rogaway, 1993; Kolaitis & Thakur, 1993; Zuckerman, 1993; Khanna, Motwani, Sudan & Vazirani, 1994; Tardos, 1994; Yannakakis, 1994; Agarwal & Condon, 1995; Arora, 1995; Bellare, Goldreich & Sudan, 1995; Feige & Verbitsky, 1996) for more details.

The exact and approximate computational complexity results for μ directly generalize to related robustness analysis and synthesis problems, using techniques described by Toker and Ozbay (1995). In particular, both the exact and approximate computation of

- the supremum of μ of a transfer function evaluated as a function of frequency, and
- μ-optimal controllers

are hard problems. Also, any such robustness problem which contains a superset of the uncertainty structures considered here are also hard. For example, exact or approximate μ-computation for systems with two complex scalar perturbations, one full complex block, and repeated real scalar perturbations are hard problems. Any easy proof of this is to define a class of matrices M which have zero rows and columns corresponding to the extra complex and repeated real perturbations, so that μ for the overall matrix M is equal to μ for a lower dimension matrix with independent real scalar perturbations and one complex perturbation. Then apply Corollary 1 and Theorem 2.

7. Conclusions

Computational complexity theory provides a powerful guide for solving the computational problems that arise when applying rigorous optimization-based approaches to large scale systems. When a problem is determined to be in the class P, then computationally-efficient polynomial-time algorithms can be developed. When a problem is determined to be NP-hard, then its computation is more difficult and practical algorithms must be based on heuristics, branch-and-bound, and/or local search.

Computational complexity theory was applied to the problems of robust stability and performance margin computation. The main results imply that it is highly unlikely that polynomial-time algorithms exist that can calculate robustness margins either exactly or within a useful desired level of accuracy for all problems. These results are especially important in light of recently-proposed techniques for robustness margin computations using Monte Carlo methods and other methods of testing only a subset of the plants in the uncertainty set (Barmish & Lagoa, 1997). Although such algorithms are of interest, the main result implies that there exist large scale uncertain systems for which such algorithms will grossly underestimate the size of the robust performance margin.

Practical algorithms for computing robustness margins should be based on rigorous upper and lower bounds that are tight for *typical* systems, although the main result implies that such bounds will not be tight for some problems. The feasibility of this approach for systems with up to a few hundred uncertain parameters has been demonstrated in computational studies (Young & Doyle, 1990; Young, Newlin & Doyle, 1991; Newlin, 1996). Current efforts are underway to develop polynomial-time model reduction algorithms that can substantially reduce the computational expense associated with robustness margin computation for typical large scale systems. Preliminary results indicate that such an approach can enable robustness margins to be computed for systems with up to a thousand uncertain parameters (Russell & Braatz, 1998b).

References

Agarwal, S., & Condon, A. (1995). On approximation algorithms for hierarchical MAX-SAT. In *Proceedings of the 10th annual structure in complexity theory conference* (pp. 214–222). Minneapolis, MN.

Amann, N., & Allgower, F. (1994). μ-Suboptimal design of a robustly performing controller for a chemical reactor. *International Journal of Control*, *59*, 665–687.

Arkun, Y., & Calvet, J.-P. (1992). Robust stabilization of input/output linearizable systems under uncertainty and disturbances. *AIChE Journal*, *38*, 1145–1156.

Arora, S. (1995). Reductions, codes, PCPs, and inapproximability. In *Proceedings of the 36th annual symposium on foundations of computer science* (pp. 404–413). Milwaukee, WI: IEEE Computer Society Press.

Arora, S., Lund, C., Motwani, R., Sudan, M., & Szegedy, M. (1992). Proof verification and hardness of approximation problems. In *Proceedings of the 33rd annual symposium on foundations of computer science* (pp. 14–23). Pittsburgh, PA: IEEE Computer Society Press.

Arora, S., & Safra, S. (1992). Probabilistic checking of proofs. In *Proceedings of the 33rd annual symposium on foundations of computer science* (pp. 2–13). Pittsburgh, PA: IEEE Computer Society Press.

Ausiello, G., D'Atri, A., & Protasi, M. (1980). Structure preserving reductions among convex optimization problems. *Journal of Computer and Systems Sciences*, *21*, 136–153.

Barland, I., Kolaitis, P. G., & Thakur, M. N. (1996). Integer programming as a framework for optimization and approximability. In *Proceedings of the eleventh annual IEEE conference on computational complexity* (pp. 249–259). Philadelphia, PA.

Barmish, B. R. (1993). *New tools for robustness of linear systems*. New York: MacMillan.

Barmish, B. R., & Lagoa, C. M. (1997). The uniform distribution: a rigorous justification of its use in robustness analysis. *Mathematics of Control, Signals, and Systems*, *10*, 203–222.

Bartlett, A. C., Hollot, C. V., & Lin, H. (1989). Root locations of an entire polytope of polynomials: it suffices to check the edges. *Mathematics of Control, Signals, and Systems*, *1*, 61–71.

Bellare, M., Goldreich, O., & Sudan, M. (1995). Free bits, PCPs, and non-approximability—towards tight results. In *Proceeding of the 36th annual symposium on foundations of computer science* (pp. 422–431). Milwaukee, WI: IEEE Computer Society Press.

Bellare, M., & Rogaway, P. (1993). The complexity of approximating a nonlinear program. In P. M. Pardalos, *Complexity in numerical optimization* (pp. 16–32). Singapore: World Scientific.

Boyd, S., El Ghaoui, L., Feron, E., & Balakrishnan, V. (1994). Linear matrix inequalities in system and control theory. In *Studies in applied mathematics*, vol. 15. Philadelphia, PA: SIAM.

Braatz, R. D., & Crisalle, O. D. (1998). Robustness analysis for systems with ellipsoidal uncertainty. *International Journal of Robust and Nonlinear Control*, *8*, 1113–1117.

Braatz, R. D., & Morari, M. (1991). μ-Sensitivities as an aid for robust identification. In *Proceedings of the American control conference* (pp. 231–236). Piscataway, NJ: IEEE Press.

Braatz, R. D., Morari, M., & Skogestad, S. (1996). Loopshaping for robust performance. *International Journal of Robust and Nonlinear Control*, *6*, 805–823.

Braatz, R. D., & VanAntwerp, J. G. (1996). Robust cross-directional control of large scale paper machines. In *Proceedings of the IEEE international conference on control applications* (pp. 155–160). Piscataway, NJ: IEEE Press.

Braatz, R. D., Young, P. M., Doyle, J. C., & Morari, M. (1994). Computational complexity of μ calculation. *IEEE Transactions on Automatic Control*, *39*, 1000–1002.

Campo, P. J., & Morari, M. (1990). Robust control of processes subject to saturation nonlinearities. *Computers and Chemical Engineering*, *14*, 343–358.

Chen, J., Fan, M. K. H., & Nett, C. N. (1991). The structured singular value and stability of uncertain polynomials: a missing link. In: ASME Annual Winter Meeting, Atlanta, GA.

Chiang, R. Y., & Safonov, M. G. (1992). *Robust control toolbox: for use with MATLAB*. Natick, MA: The Math Works.

Coxson, G., & DeMarco, C. (1992). *Computing the real structured singular value is NP-hard*. Report ECE-92-4, Department of Electrical and Computer Engineering, University of Wisconsin-Madison.

Dave, P., Willig, D. A., Kudva, G. K., Pekny, J. F., & Doyle, F. J. (1997). LP methods in MPC of large scale systems: application to paper-machine CD control. *AIChE Journal*, *43*, 1016–1031.

Demmel, J. W. (1992). The component-wise distance to the nearest singular matrix. *SIAM Journal of Matrix Analysis and Applications*, *13*, 10–19.

Doyle, J. C. (1982). Analysis of feedback systems with structured uncertainties. *IEE Proc. Pt. D.*, *129*, 242–250.

Doyle III, F. J., Packard, A. K., & Morari, M. (1989). Robust controller design for a nonlinear CSTR. *Chemical Engineering Science*, *44*, 1929–1947.

Fan, M. K. H., & Tits, A. L. (1992). A measure of worst-case H_∞ performance and of largest acceptable uncertainty. *Systems and Control Letters*, *18*, 409–421.

Fan, M. K. H., Tits, A. L., & Doyle, J. C. (1991). Robustness in the presence of mixed parametric uncertainty and unmodeled dynamics. *IEEE Transactions on Automatic Control*, *36*, 25–38.

Featherstone, A. P., & Braatz, R. D. (1997). Control-oriented modeling of sheet and film processes. *AIChE Journal*, *43*, 1989–2001.

Featherstone, A. P., & Braatz, R. D. (1998a). Input design for large scale sheet and film processes. *Industrial Engineering and Chemical Research*, *37*, 449–454.

Featherstone, A. P., & Braatz, R. D. (1998b). Integrated robust identification and control of large scale processes. *Industrial Engineering and Chemical Research*, *37*, 97–106.

Feige, U., Goldwasser, S., Lovasz, L., Safra, S., & Szegedy, M. (1991). Approximating clique is almost NP-complete. In *Proceedings 32nd annual symposium on foundations of computer science* (pp. 2–12). San Juan, Puerto Rico.

Feige, U., & Lovasz, L. (1992). Two-prover one-round proof systems: their power and their problems. In *Proceedings of the 24th annual ACM symposium on the theory of computing* (pp. 733–744). Victoria, BC, Canada.

Feige, U., & Verbitsky, O. (1996). Error reduction by parallel repetition-a negative result. In *Proceedings of the eleventh annual IEEE conference on computational complexity* (pp. 70–76). Philadelphia, PA.

Garey, M. R., & Johnson, D. S. (1983). *Computers and Intractibility: a guide to NP-completeness*. New York: W.H. Freeman.

Horn, I. G., Arulandu, J. R., Gombas, C. J., VanAntwerp, J. G., & Braatz, R. D. (1996). Improved filter design in internal model control. *Industrial Engineering and Chemical Research*, *35*, 3437–3441.

Hovd, M., Braatz, R. D., & Skogestad, S. (1996). SVD controllers for H_{2-}, $H_{\infty-}$, and μ-optimal control. *Automatica*, *33*, 433–439.

Johnson, D. S., & Papadimitriou, C. H. (1989). Computational complexity. In E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, & D. B. Shmoys, *The traveling salesman problem* (pp. 37–85). Chichester: Wiley.

Khachiyan, L. G., & Todd, M. J. (1993). On the complexity of approximating the maximal inscribed ellipsoid for a polytope. *Mathematical Programming*, *61*, 137–159.

Khanna, S., Motwani, R., Sudan, M., & Vazirani, U. (1994). On syntactic versus computational views of approximability. In *Proceedings 35th annual symposium on foundations of computer science* (pp. 819–830). Santa Fe, NM.

Kharitonov, V. L. (1978). Asymptotic stability of an equilibrium position of a family of systems of linear differential equations. *Differential'nye Uraveniya*, *14*, 1483–1485.

Kolaitis, P. G., & Thakur, M. N. (1993). Polynomial-time optimization, parallel approximation, and fixpoint logic. In *Proceedings of the 8th annual structure in complexity theory conference* (pp. 31–41). San Diego, CA.

Kozub, D. J., MacGregor, J. F., & Harris, T. J. (1989). Optimal IMC inverses: design and robustness considerations. *Chemical Engineering Science*, *44*, 2121–2136.

Laughlin, D. L., Jordan, K. G., & Morari, M. (1986). Internal model control and process uncertainty-mapping uncertainty regions for SISO controller-design. *International Journal of Control*, *44*, 1675–1698.

McCormick, G. P. (1983). *Nonlinear programming. theory, algorithms, and applications*. New York: Wiley Interscience.

Morari, M., & Zafiriou, E. (1989). *Robust process control*. Englewood Cliffs, NJ: Prentice-Hall.

Nemirovskii, A. (1993). Several NP-hard problems arising in robust stability analysis. *Mathematics of Control, Signals, Systems*, *6*, 99–105.

Nesterov, Y., & Nemirovskii, A. (1994). Interior point polynomial algorithms in convex programming. In *Studies in applied mathematics*, vol. 13. Philadelphia, PA: SIAM.

Newlin, M. P. (1996). *Model validation, computation, and control*. PhD thesis, Caltech, Pasadena.

Packard, A. K. (1988). *What's new with μ: structured uncertainty in multivariable control*. PhD thesis, University of California, Berkeley.

Palmor, Z. J., & Shinnar, R. (1981). Design of advanced process controllers. *AIChE Journal*, *27*, 793–805.

Papadimitriou, C. H., & Steiglitz, K. (1982). *Combinatorial optimization: algorithms and complexity*. Englewood Cliffs, NJ: Prentice-Hall.

Pardalos, P. M., & Vavasis, S. A. (1991). Quadratic programming with one negative eigenvalue is NP-hard. *Journal of Global Optimization*, *1*, 15–22.

Poljak, S., & Rohn, J. (1993). Checking robust non-singularity is NP-hard. *Mathematics of Control, Signals, and Systems*, *6*, 1–9.

Rao, C. V., Campbell, J. C., Rawlings, J. B., & Wright, S. J. (1997). Efficient implementation of model predictive control for sheet and film forming processes. In *Proceedings of the American control conference* (pp. 2940–2944). Piscataway, NJ: IEEE Press.

Reinelt, G. (1994). The traveling salesman problem: computational solutions for TSP applications. In *Lectures in computer science*, vol. 840. Berlin: Springer-Verlag.

Ricker, N. L., & Lee, J. H. (1995). Nonlinear model predictive control of the Tennessee–Eastman challenge process. *Computers and Chemical Engineering*, *19*, 961–981.

Rigopoulos, A., Arkun, Y., & Kayihan, F. (1997). Identification of full profile disturbance models for sheet forming processes. *AIChE Journal*, *43*, 727–740.

Russell, E. L., & Braatz, R. D. (1996). Analysis of large scale systems with model uncertainty, actuator and state constraints, and time delays. In *AIChE Annual Meeting*, Chicago, IL. Paper 45a.

Russell, E. L., & Braatz, R. D. (1998a). The average-case identifiability and controllability of large scale systems. *Journal of Process Control*, in press.

Russell, E. L., & Braatz, R. D. (1998b). Model reduction for the robustness margin computation of large scale uncertain systems. *Computers and Chemical Engineering*, *22*, 913–926.

Russell, E. L., Power, C. P. H., & Braatz, R. D. (1997). Multidimensional realizations of large scale uncertain systems for multivariable stability margin computation. *International Journal of Robust and Nonlinear Control*, *7*, 113–125.

Ryoo, H. S., & Sahinidis, N. V. (1996). A branch-and-reduce approach to global optimization. *Journal of Global Optimization*, *8*, 107–139.

Safonov, M. G. (1982). Stability margins of diagonally perturbed multivariable feedback systems. *IEE Proc. Pt. D*, *129*, 251–256.

Sahni, S., & Gonzales, T. (1976). P-complete approximation problems. *Journal of the ACM*, *23*, 555–565.

Schaper, C. D., Seborg, D. E., & Mellichamp, D. A. (1992). Probabilistic approach to robust process control. *Industrial Engineering and Chemical Research*, *31*, 1694–1704.

Skogestad, S., & Morari, M. (1988). Some new properties of the structured singular value. *IEEE Transactions on Automatic Control*, *33*, 1151–1154.

Skogestad, S., & Postlethwaite, I. (1996). *Multivariable feedback control: analysis and design*. New York: Wiley.

Smith, R. S. R. (1990). *Model validation for uncertain systems*. PhD thesis, California Institute of Technology, Pasadena.

Song, H. K., Fisher, D. G., & Shah, S. L. (1984). Experimental evaluation of a robust self-tuning PID controller. *Canadian Journal of Chemical Engineering*, *62*, 755–763.

Stryczek, K., & Brosilow, C. B. (1996). Mp tuning of multivariable uncertain processes. In *Proceedings of the fifth international conference on chemical process control (CPC-V)*. Tahoe City, CA.

Tardos, G. (1994). Multi-prover encoding schemes & three-prover proof systems. In *Proceedings of the ninth annual structure in complexity theory conference* (pp. 308–317). Amsterdam, The Netherlands.

Toker, O., & Ozbay, H. (1995). On the NP-hardness of the purely complex μ computation, analysis/synthesis, and some related problems in multidimensional systems. In *Proceedings of the American control conference* (pp. 447–451). Piscataway, NJ: IEEE Press.

VanAntwerp, J. G., & Braatz, R. D. (1998). Fast model predictive control of sheet and film processes. *IEEE Transactions on Control Systems Technology*, in press.

VanAntwerp, J. G., Braatz, R. D., & Sahinidis, N. V. (1998). Globally optimal robust process control. *Journal of Process Control*, in press.

Vavasis, S. A. (1992). On approximation algorithms for concave programming. In *Recent advances in global optimization* (pp. 3–18). Princeton, NJ: Princeton University Press.

Vavasis, S. A. (1993). Polynomial time weak approximation algorithms for quadratic programming. In *Complexity in numerical optimization* (pp. 490–500). Singapore: World Scientific.

Yannakakis, M. (1994). Some open problems in approximation. In M. Bonuccelli, P. Crescenzi, & R. Petreschi, *Algorithms and complexity*. In: *Lectures in computer science*, vol. 778 (pp. 33–53). Berlin: Springer-Verlag.

Young, P. M., & Doyle, J. C. (1990). Computation of the structured singular value with real and complex uncertainties. In *Proceedings of the IEEE conference on decision and control* (pp. 1230–1235). Piscataway, NJ: IEEE Press.

Young, P. M., Newlin, M. P., & Doyle, J. C. (1991). μ analysis with real parametric uncertainty. In *Proceedings of the IEEE conference on decision and control* (pp. 1251–1256). Piscataway, NJ: IEEE Press.

Yousefpor, M., Palazoglu, A., & Hess, R. (1988). Robust control design for SISO systems based on constrained optimization. *International Journal of Control*, *62*, 477.

Zafiriou, E., & Marchal, A. L. (1991). Stability of SISO quadratic dynamic matrix control with hard output constraints. *AIChE Journal*, *37*, 1550–1560.

Zhou, K., Doyle, J. C., & Glover, K. (1995). *Robust and optimal control*. New Jersey: Prentice-Hall.

Zuckerman, D. (1993). NP-complete problems have a version that's hard to approximate. In *Proceedings of the eight annual structure in complexity theory conference* (pp. 305–312). San Diego, CA.