J Glob Optim (2007) 38:509–526
DOI 10.1007/s10898-006-9092-2

Global optimization in stabilizing controller design

YoungJung Chang · Nikolaos V. Sahinidis

Received: 30 August 2006 / Accepted: 31 August 2006 / Published online: 7 October 2006

Abstract In this paper, we develop a global optimization methodology to solve stabilization problems. We first formulate stabilization problems as bilevel programming problems. By invoking the Hurwitz stability conditions, we reformulate these bilevel programs to equivalent single-level nonconvex optimization programs. The branch-and-reduce global optimization algorithm is finally applied to these problems. Using the proposed methodology, we report improved solutions for two feedback stabilization problems from the literature. In addition, we improve the lower bound of the stabilizability parameter of the Belgian chocolate problem from the previous best known 0.96 to 0.973974.

Keywords Stability · Simultaneous stabilization · Linear controller design · Global optimization · Belgian chocolate problem

1 Introduction

For a given process model, controller design seeks to design a set of controllers and place them so as to enforce certain control objectives on the underlying process. Because of its close relation to safety, stability is one of the objectives often considered in controller design as the minimum requirement to be met. Therefore, stabilization, i.e., designing a controller that stabilizes an unstable process, is a central aspect in controller system design.

If the control law to be applied to the system is determined beforehand, two types of stabilization problems can arise. The first one is the *stabilizability problem*, in which one must determine whether and under what conditions the process is stabilizable with the given type of controller. The second problem is the *stabilizing controller*

Y. J. Chang · N. V. Sahinidis (✉)
Department of Chemical and Biomolecular Engineering, University of Illinois
at Urbana-Champaign, 600 South Mathews Avenue, Urbana, IL 61801, USA
e-mail: nikos@uiuc.edu

design problem, in which one must find a controller accomplishing stabilization of the given process. This is the problem addressed in this paper and is known to be a difficult problem. In particular, there are many stabilizing controller design problems for which no polynomial-time algorithm exists unless $\mathcal{P} = \mathcal{NP}$, even for linear controller design problems. Static (memoryless) state feedback stabilization with constraints on gains is one such $\mathcal{NP}$-hard problem [7].

Since the model is only an approximation of the physical process and various types of uncertainties exist in a real system, stability issues are often studied along with robustness issues. In particular, controllers must be designed to *robustly stabilize* the process. In other words, controllers must be able to simultaneously stabilize a number of "closely related" processes (could be infinitely many). Such $\mathcal{NP}$-hard problems include simultaneous stabilization by static output feedback and simultaneous stabilization of three scalar systems by dynamic state feedback [5,7].

There have been theoretical works to address many stabilization problems. The French champagne problem [6] involves determining the simultaneous stabilizability of two processes by a stable controller. This problem has been answered theoretically [19]. The Belgian chocolate problem is a similar problem of finding the range of a process parameter for which the process is stabilizable by a minimum phase stable controller [5]. It has been shown in [5] that there exists a critical value which splits the stabilizable and unstabilizable parameter region. Also in [5], an upper bound slightly less than 1 has been found theoretically for this critical value. A computational and theoretical study of the Belgian chocolate problem has recently appeared in [8]. In the computational study, these authors observed that stabilizing controllers of low degree take a special structure. This motivated theoretical analysis to find the maximum of the parameter value which is stabilizable by a fixed order controller. The computations in [8] have shown that 0.96 is a lower bound on the critical value of the stabilizability parameter for the Belgian chocolate problem.

However, most efforts in the area of stabilizing controller design have been computational and relied extensively on local optimization techniques or suboptimal heuristics because of the $\mathcal{NP}$-hardness of the underlying mathematical problem [8,20]. Since local optimization algorithms do not guarantee global optimality of solutions, local approaches are successful only when they happen to identify a controller that stabilizes the process. When no stabilizing controller is found, no conclusion can be drawn regarding the stabilizability of the process.

In this work, we adopt a global optimization approach to develop a rigorous methodology for stabilization problems of linear systems. These stabilization problems are naturally formulated as bilevel programming problems. Following an approach we first introduced in the context of metabolic networks [9], in Sect. 2, we transform these bilevel programs to equivalent single-level optimization problems. In Sect. 3, we show that the necessary algebraic conditions for stability are nonconvex. In Sect. 4, we describe the branch-and-reduce global optimization algorithm [31] that we use for solution of the models addressed in this paper. The benefits of the proposed methodology are demonstrated on several problems from the literature in Sects. 5 through 8. The treatment includes well-known $\mathcal{NP}$-hard problems, including simultaneous feedback stabilization and the Belgian chocolate problem [5], for both of which the proposed methodology obtains improved solutions compared to those in the existing literature.

2 Stabilization of linear systems

Consider a multiple-input and multiple-output linear system

$$\frac{\mathrm{d}x}{\mathrm{d}t} = Ax + Bu, \tag{1}$$

$$y = Cx, \tag{2}$$

where $x \in \mathbb{R}^n$ is a *plant state* vector, $y \in \mathbb{R}^m$ is a *measured output* vector, and $u \in \mathbb{R}^l$ is a *control input* vector.

Assume that, without any control, i.e., with $u = 0$, the *open-loop* system $\dot{x} = Ax$ is unstable, which means that the matrix A has an eigenvalue with nonnegative real part. This system must be stabilized by the application of some linear feedback control action (*stabilizer*) such that the *closed-loop* system becomes stable. If the controller has a *controller state* $x_c \in \mathbb{R}^{n_c}$ in some finite-dimensional space, we refer to that controller as a linear *dynamic* output feedback controller of *order* n_c, and its control law is expressed as:

$$u = K_1 y + K_2 x_c,$$
$$\frac{\mathrm{d}x_c}{\mathrm{d}t} = K_3 y + K_4 x_c$$

If $n_c = 0$, then we have a *static* or *memoryless* linear output feedback controller.

In general, the closed-loop dynamic system can be expressed as:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = (A + BK_1C)x + BK_2x_c,$$
$$\frac{\mathrm{d}x_c}{\mathrm{d}t} = K_3Cx + K_4x_c.$$

Therefore, for the closed-loop system to be stable, we require:

$$\begin{bmatrix} A + BK_1C & BK_2 \\ K_3C & K_4 \end{bmatrix} \in \mathcal{SM},$$

where $\mathcal{SM}$ denotes the set of Hurwitz-stable matrices. For a matrix to be *Hurwitz stable*, we require all its eigenvalues to lie in the left half of the complex plane.

Instead of simply ensuring the stability of the closed-loop system, minimizing the maximum real part of its eigenvalues (the so-called *spectral abscissa*, which is often related to the *stability degree* or *real stability radius*) is in common practice because it provides the slowest decay rate of the system [2]. Now let

$$K = \begin{bmatrix} K_1 & K_2 \\ K_3 & K_4 \end{bmatrix}. \tag{3}$$

Then, the problem of maximizing the stability degree, or equivalently minimizing the spectral abscissa of the dynamic matrix, can be formulated as follows:

$$\begin{aligned} &\min\ z \\ &\text{s.t.}\ \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} B & 0 \\ 0 & I \end{bmatrix} K \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix} - zI \in \mathcal{SM}. \end{aligned}$$

This problem can be translated into a bilinear matrix inequality (BMI) problem. However, unlike linear matrix inequality problems which are often convex and solvable by semi-definite programming techniques, BMI problems are generally nonconvex. There have been some efforts to solve BMI problems to global optimality [3,12,15,34]. The computational efficiency of these methods and their ability to deal with side constraints involving more general nonlinear functions are still under investigation.

We can alternatively ensure that a matrix is Hurwitz stable by requiring all the roots of its characteristic equation $f_K(z;s) = 0$ to lie in the left half of complex plane, i.e., by requiring $f_K(z;s) \in \mathcal{SP}$, where $\mathcal{SP}$ is the set of Hurwitz stable polynomials. Therefore, the problem can be rewritten as the following bilevel programming problem:

$$
\begin{array}{ll}
\text{(B)} \ \min_{K,z} \ z & \\
\text{s.t.} & \left\{ \begin{array}{ll} \max_{\lambda,\mu} \ \lambda & \\ \quad \text{s.t.} & \Re(f_K(z;\lambda+\mu\mathbb{i})) = 0 \\ & \Im(f_K(z;\lambda+\mu\mathbb{i})) = 0 \\ & \lambda,\mu \in \mathbb{R} \end{array} \right\} < 0,
\end{array}
$$

where $\Re$ and $\Im$ denote, respectively, the real and imaginary parts of the characteristic polynomial.

The lower-level problem is nothing but the problem of finding the *abscissa* of a polynomial $f_K(z;s)$. To reformulate this problem into an equivalent single-level program without locating all the zeros of $f_K(z;s)$, we need to find equivalent algebraic conditions for Hurwitz stability of polynomials. This is addressed in the following section.

3 Hurwitz stability of polynomials

3.1 Direct application of stability criteria

Consider a polynomial $f(s) = \sum_{i=0}^{n} f_i s^i$, with $f_n > 0$. To ensure that all the zeros of $f(s)$ have negative real parts, we will rely on the *symbolic* application of well-established stability criteria to construct equivalent inequalities for stability. A well-known necessary condition on the stability of real polynomials is Descartes rule of signs, which requires all coefficients of the polynomial to be of the same sign. More sophisticated necessary and sufficient continuous stability criteria include:

- Routh/Hurwitz/Bilharz criteria [4,14,21].
- Liénard-Chipart criterion [17].
- Markov criterion [11].
- Schwarz criterion [26].
- Strelitz criterion [28].

The Routh, Hurwitz, and Bilharz criteria are all equivalent. In particular, the Hurwitz criterion requires the leading principal minors $\Delta_n^i, i = 1, \ldots, n$ of the *Hurwitz matrix*

$$H = \begin{pmatrix} f_{n-1} & f_{n-3} & f_{n-5} & f_{n-7} & \cdots \\ f_n & f_{n-2} & f_{n-4} & f_{n-6} & \cdots \\ 0 & f_{n-1} & f_{n-3} & f_{n-5} & \cdots \\ 0 & f_n & f_{n-2} & f_{n-4} & \cdots \\ 0 & 0 & f_{n-1} & f_{n-3} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

to be positive. If $f_{n-1} > 0$, then the Hurwitz criterion reduces to $\Delta_n^i > 0, i = 2, \ldots, n-1$. The Liénard–Chipart criterion is a further simplification of the Hurwitz criterion with the help of Descartes' rule of signs. The Markov criterion computes *Markov parameters* of *Laurent series* expansion from the f_i's by decomposing $f(s)$ to construct stability conditions. The Schwarz criterion makes use of a similarity transformation of a matrix into a specially structured *Schwarz matrix* to derive stability conditions. The Strelitz criterion builds an auxiliary sum-of-roots polynomial of $f(s)$ to get stability conditions.

The algebraic expressions from the above criteria are ultimately functions of the coefficients $f_i, i = 0, \ldots, n$. Because the very nature of the stability conditions is restrictions on the signs of coefficients, these criteria result in strict inequalities. Let these expressions be $h(f(s)) > 0$. With these algebraic conditions at hand, we can reformulate B into the following single-level problem:

$$\begin{aligned} \text{(S)}\ \min\ & z \\ \text{s.t.}\ & h(f_K(z; s)) > 0. \end{aligned}$$

3.2 Decomposing high-degree polynomials

It is well known that the cost for computing the determinant of an $n \times n$ square matrix by a naive Laplacian expansion by minors or cofactors, or by the explicit Leibniz formula using permutation matrices is $O(n!)$. The determinant can also be calculated in $O(n^3)$ effort by applying Gaussian elimination or LU factorization. For symbolic purposes, variants of Gaussian elimination, such as a fraction-free algorithm, have been developed in order to avoid lengthy and fractional intermediate expressions. However, their efficiency is still just comparable to the minor expansion algorithm [25]. Therefore, constructing algebraic stability constraints for high-degree characteristic polynomials is still a considerable undertaking.

In addition, when the degree of the characteristic polynomial is high, the algebraic constraints that enforce the stability conditions also become of high degree. This eventually leads to scaling issues when numerical optimization is performed over these constraints. To remedy these difficulties, we can factor the polynomial in terms of lower degree polynomials with additional algebraic equations. For example,

$$f(s) = \sum_{j=0}^{n} f_j s^j = f_n \prod_{j=1}^{n} (s - \lambda_j - \mu_j \mathbb{i})$$

with additional constraints for complex conjugacy, whenever needed, is the most obvious such factorization. However, in order to avoid dealing with complex numbers, it is preferable to factor the original polynomial into second-order polynomials and, possibly, a single first-order polynomial. If $n = 2m + \ell$, where $m = \lfloor \frac{n}{2} \rfloor$ and $\ell \in \{0, 1\}$:

$$f(s) = f_n(s+c)^{\ell} \prod_{j=1}^{m} (s^2 + a_j s + b_j).$$

The necessary and sufficient conditions for $f(s) \in \mathcal{SP}$ are $a_j, b_j > 0$, for $j = 1, \ldots, m$ and $c > 0$. Since $f(s)$ is a real polynomial, there always exists such a decomposition. To get rid of symmetry, we can further require $a_j \geq a_{j+1}, j = 1, \ldots, m-1$. While there is no guarantee for the existence of a real factorization into odd-order real polynomials, a factorization into even-order real polynomials always exists (the remainder would be some real polynomial of lower degree).

4 Branch-and-reduce global optimization algorithm

Problem S is a single-level reformulation of the bilevel problem B, which minimizes the spectral abscissa of the system. Alternatively, one could consider the optimization of an economic objective subject to the same stability conditions of problems S/B. We note that these constraints are nonlinear and contain multilinear or higher order multivariate product terms and are thus nonconvex. To solve S to global optimality, we employ the branch-and-reduce global optimization algorithm, which is a variant of the branch-and-bound algorithm. The building blocks of the branch-and-reduce algorithm are: (a) a polyhedral outer approximation of the feasible set that is based on convex under- and concave over-estimators of elemental problem functions, (b) partitioning of feasible sets that often results in finite global optimization algorithms, and (c) extensive use of optimality and feasibility arguments to achieve reduction of the search space over problem subdomains.

Many branch-and-bound algorithms for nonlinear programs make use of McCormick's bounding techniques for factorable programs [18]. While these factorable approaches lead to a completely automatable procedure for the construction of convex lower-bounding problems for nonconvex functions, these bounding problems often exhibit a large relaxation gap. From the point of view of relaxation quality, it is always advantageous to directly convexify the original problem functions and constraints to the extent possible. The theory of convex extensions [30] provides a systematic methodology for constructing the closed-form expression of convex envelopes of multidimensional, lower semi-continuous functions. This theory provides the capability to construct the convex and concave envelopes of continuous functions based on their generating sets. This theory has also shown that product disaggregation tightens certain relaxations [29]. The branch-and-reduce approach combines these convexification techniques with a sandwich algorithm to construct a *polyhedral* outer-approximation of the nonlinear functions, thus facilitating the use of fast and reliable linear programming techniques for computing lower bounds [32,33].

The quality of the relaxations obtained from the previously mentioned techniques is a strong function of the bounds of variables that participate in nonlinear relationships. Thus, tighter variable bounds imply tighter relaxations and, hence, faster convergence of branch-and-bound. The branch-and-reduce algorithm places a strong emphasis on the derivation of tight bounds for all problem variables. In each node of the branch-and-reduce tree, constraint programming techniques are utilized in a *preprocessing* step to reduce ranges of problem variables before the relaxation is constructed (feasibility-based range reduction). Once the relaxation is solved, a *post-processing* step utilizes the solution of the relaxed problem in an attempt to further reduce ranges of variables (optimality-based range reduction) [22,31]. Subsequently, branching occurs.

The branch-and-reduce algorithm uses node selection and branching rules that guarantee finite termination in certain cases [27] and ϵ-convergence in general [22]. Overall, at each node of the search tree, the branch-and-reduce algorithm performs the following steps:

- **Step 1.** Preprocessing for range reduction based on constraint propagation.
- **Step 2.** Construction of a convex relaxation problem by composing a polyhedral outer approximation of the current feasible set and a convexification of the objective function.
- **Step 3.** Solution of the relaxation problem with a linear programming solver.
- **Step 4.** Based on the solution to the relaxation problem, do one of the following:

  (1) If the relaxation is infeasible or inferior to the current upper bound, fathom the current node.

  (2) Else, perform postprocessing to further reduce the current region using dual information of the relaxation, and either resolve the current relaxation or partition the current node to generate two children nodes.

For more detailed descriptions of the branch-and-reduce algorithm, the reader is referred to [22,23,31–33]. In the sequel, we use this algorithm to tackle formulation S and some of its variants for stabilizing controller design problems.

5 Stabilization by static output feedback

For illustrative purposes, we first consider a small stabilization problem by static output feedback with constraints on gain. The problem is taken from [3] and involves the following system matrices in the defining equations (1) and (2):

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \qquad C = \begin{bmatrix} 1 \ 0 \ 2 \end{bmatrix}.$$

The gain, K, in Eq. 3 is required to be bounded: $K \in [-10, -1]$.

In [3], a bilinear matrix inequality formulation equivalent to B was solved by a decomposition technique to identify a *Lyapunov matrix* $P \succ \kappa I$, where κ is a constant defining how positive P should be. For $\kappa = 0.1$, the solution obtained after 24 iterations in [3] was $K = -9.4277$ with a stability radius of $-z = 1.0541$. The authors in [3] reported heavy increase in computational costs to compute the stability radius when the value of κ was reduced to 0.05.

To apply the proposed framework to this problem, we first need to compute the characteristic polynomial of:

$$A + BKC - zI = \begin{bmatrix} K - z & 1 & 2K \\ 1 & -1 - z & 0 \\ 1 & 0 & 1 - z \end{bmatrix}.$$

Doing this symbolically, we obtain:

$$\begin{aligned} f_K(z; s) = s^3 + (3z - K)s^2 + (3z^2 - 2Kz - 2K - 2)s \\ + z^3 - Kz^2 - 2(K + 1)z - K + 1. \end{aligned}$$

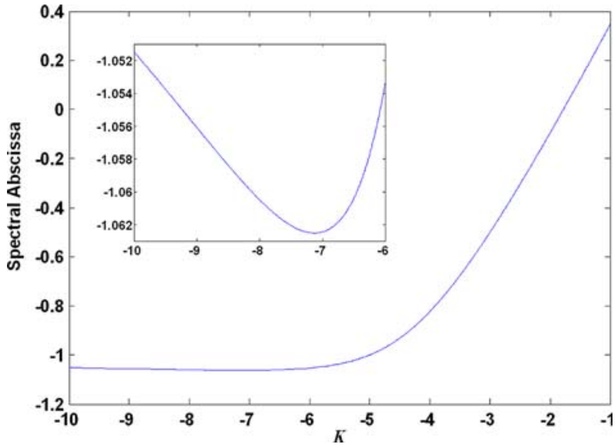

Fig. 1 Closed-loop spectral abscissa against K for the first example

Then, the Hurwitz criterion gives

$$h_2 = \Delta_3^2 = 8z^3 - 8Kz^2 + 2(K^2 - 2K - 2)z + 2K^2 + 3K - 1 \geq \epsilon$$

for some small positive number ϵ. For computations, we use $\epsilon = 10^{-5}$. Solving problem S gives the global solution of $K = -7.1281$ with stability radius $-z = 1.06249$ in 12 branch-and-reduce iterations and 0.11 CPU sec on a Dell workstation with a 3 GHz CPU and 1 GB RAM. The constraint on h_2 is binding at this solution. Further computation by setting $\epsilon = 0$ shows that the best possible stability radius is $-z = 1.0625$. Figure 1 shows the variation of spectral abscissa along K and the insert is a magnification of y-axis for $K \in [-10, -6]$. This shows that the true minimizer is at $K = -7.125$ with a stability radius of $-z = 1.0625$, which is very close to the solution obtained within ϵ tolerances and somewhat better than the solution of $-z = 1.0541$ that was earlier reported in [3].

6 Low-order controller design

The *low-order controller design problem* is the problem of finding a minimum order stabilizing controller for a given process, i.e., the problem of finding the smallest possible n_c for which the controller stabilizes the process. The advantages of low-order controllers are described in [1]. Let us consider the following flexible actuator example from [10,16], which is not stabilizable by static output feedback but stabilizable by dynamic output feedback with $n_c = 1$:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.02 \\ 0.2 & 0 & 0 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ -0.2 \\ 0 \\ 1 \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

With bounds of 10 on the magnitude of gain matrix elements, this system is stabilizable with $n_c = 1$ giving a globally optimal stability degree of 1.10334. This result was obtained by solving formulation S. This problem turned out very difficult numerically

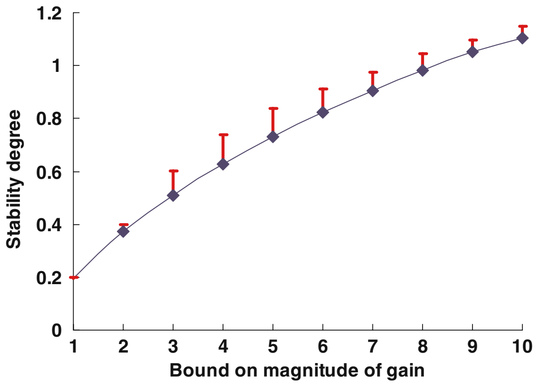

Fig. 2 Trade-off between stability degree and the magnitude of gain

requiring 5.8 CPU hours of computing time to close the relative gap between the lower and upper bounds under 4%. Varying the bounds on the magnitude of gain, we get the trade-off curve of stability degree as in Fig. 2, where each error bar shows the absolute gap between upper and lower bounds obtained after 2 CPU h (the problem for a bound of 10 on the gain matrix elements was allowed to run longer).

7 Simultaneous stabilization by static state feedback

Consider the following problem from [13] that seeks to design one controller K for stabilizing three processes with the following dynamic matrices:

$$A_1 = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & -0.5 \end{bmatrix}, \qquad A_2 = \begin{bmatrix} 1.5 & -7 & 0 \\ 7 & 1.5 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad A_3 = \begin{bmatrix} -0.5 & -3 & 0 \\ 3 & -0.5 & 0 \\ 0 & 0 & 2 \end{bmatrix},$$

$$B_1 = B_2 = B_3 = \begin{bmatrix} 0.2477 & -0.1645 \\ 0.4070 & 0.8115 \\ 0.6481 & 0.4083 \end{bmatrix}$$

and $C_1 = C_2 = C_3 = I$. We further require that $|K_{ij}| \leq 50$. If we want to further reduce the maximum gain, we can formulate the problem as:

$$\begin{aligned} \min \quad & t \\ \text{s.t.} \quad & A_k + B_k K \in \mathcal{SM} \quad k = 1,2,3 \\ & -t \leq K_{ij} \leq t \quad \forall i,j \\ & t \in [0,50]. \end{aligned}$$

We first compute characteristic polynomials $f_K^{(k)}$ for each closed-loop $A_k + B_k K$ and construct stability conditions $h(f_K^{(k)}(s)) \geq \epsilon > 0$ with their coefficients. Since

$$f_K^{(k)}(s) = \det(sI - A_k - B_k K) = s^3 + \phi_2^{(k)} s^2 + \phi_1^{(k)} s + \phi_0^{(k)}$$

the Hurwitz criterion gives

$$h\left(f_K^{(k)}(s)\right) = \phi_1^{(k)} \phi_2^{(k)} - \phi_0^{(k)} \geq \epsilon > 0,$$

together with positivity constraints on the ϕ's. This system is stabilizable with $t^* = 2.35$. This t^* value can be slightly improved (by less than 0.001) by sacrificing the stability margin to almost 0.

On the other hand, if our objective is to maximize the stability degree, the problem becomes

$$\begin{aligned} \min \quad & z \\ \text{s.t.} \quad & A_k + BK - zI \in \mathcal{SM} \quad k = 1, 2, 3, \\ & K_{ij} \in [-50, 50] \quad \forall i, j. \end{aligned}$$

Following the same procedure as above, we construct $h(f_K^{(k)}(z; s)) \geq \epsilon$ for each matrix $A_k + B_k K - zI$. Since

$$f_K^{(k)}(z; s) = \det((s + z)I - A_k - B_k K) = s^3 + \psi_2^{(k)}(z)s^2 + \psi_1^{(k)}(z)s + \psi_0^{(k)}(z)$$

the positivity conditions and the Hurwitz criterion give

$$\begin{aligned} \psi_0^{(k)}(z) \geq \epsilon, \quad \psi_1^{(k)}(z) \geq \epsilon, \quad \psi_2^{(k)}(z) \geq \epsilon, \\ h\left(f_K^{(k)}(z; s)\right) = \psi_1^{(k)}(z)\psi_0^{(k)}(z) - \psi_0^{(k)}(z) \geq \epsilon > 0. \end{aligned}$$

The achieved maximum stability degree is $-z^* = 7.2387$. This solution is shown to be within 0.01% range of best possible solution after 9 h of computation and is significantly better than the local solution 1.0510 found by Hassibi et al. [13].

8 The Belgian chocolate problem

Consider three processes $p_i(s) = \frac{b_i(s)}{a_i(s)}$, for $i = 1, 2, 3$. The problem of simultaneous stabilization of three systems (S3P) is to design a rational controller $c(s) = \frac{y(s)}{x(s)}$ which stabilizes all three processes at the same time. Mathematically, this latter condition is equivalent to requiring $a_i x + b_i y$ to be a stable polynomial for all i. The problem of simultaneous strong stabilization of two systems (SS2P) is a special case of S3P with $p_3(s) = 0/1$ (which means that the controller must be stable). Furthermore, the problem of simultaneous bistable stabilization of one system (SB1P) is a special case of SS2P with $p_2(s) = 1/0$ (which means that the controller must also be *inverstable* and thus *bistable*). There exist equivalence relations among S3P, SS2P, and SB1P showing that these three are equivalent $\mathcal{NP}$-hard problems.

The Belgian chocolate problem (BCP) is similar to SB1P with $p_1(s) = \frac{b(s)}{a(s)}$. The only difference is that, in the Belgian chocolate problem, the inverstable requirement of the controller in SB1P is weakened to a *minimum phase* requirement, i.e., a controller whose zeros are ∞ or lie in the left half of the complex plane. (If ∞ is a zero of a controller and all the other zeros have negative real parts, the controller is minimum phase but not inverstable.)

The original statement of BCP in [5] is twofold:

1. Find a stable and minimum phase controller stabilizing $\frac{(s-1)^2}{s^2-1.8s+1}$ or show that such a controller does not exist.
2. For what values of $\delta \in \mathbb{R}$ is the scalar linear system $\frac{s^2-1}{s^2-2\delta s+1}$ stabilizable by a controller that is both stable and minimum phase?

It is believed that the process in the first question should be corrected to $\frac{s^2-1}{s^2-1.8s+1}$, thus giving the special case of the second question with $\delta = 0.9$. This modified first question has been answered first by Patel et al. [20] using a random search method. For the second question, it is known that the process is stabilizable if and only if $\delta < \delta^*$ for some $\delta^* < 0.9999800002$ [5]. Furthermore, it has been shown that $\delta^* > 0.96$ by a computational local optimization approach [8].

8.1 Formulation of the Belgian chocolate problem

To find δ^*, we need to solve the following problem:

$$\begin{aligned} \max\ & \delta \\ \text{s.t.}\ & \deg(y) \leq \deg(x) \\ & x(s), y(s), (ax+by)(s) \in \mathcal{SP}. \end{aligned}$$

This problem has an infinite number of variables and constraints. To make the problem more manageable, we will fix the degree of the polynomial $x(s)$ to n giving:

$$\begin{aligned} \max\ & \delta \\ \text{s.t.}\ & x(s) \in \mathcal{MP}^n, \\ & y(s) \in \mathcal{P}^n, \\ & x(s), y(s), (ax+by)(s) \in \mathcal{SP}, \end{aligned}$$

where $\mathcal{MP}^n$ is the set of real monic polynomials of degree n and $\mathcal{P}^n$ is the set of real polynomials of degree no more than n. In this formulation, the degree requirement is enforced for free.

To formulate this problem as a mathematical program, we must convert each constraint into algebraic equations or inequalities. Since x and y are polynomials of degree n, we can write $x(s) = \sum_{i=0}^{n} x_i s^i$ and $y(s) = \sum_{i=0}^{n} y_i s^i$. From Descartes' rule of signs, we know that $x_i > 0$ for every i. To ensure the same condition on y_i, we introduce binary variables $z_i, i = 0, \ldots, n$ and write

$$\begin{aligned} \epsilon_i \leq x_i \leq \tau_i, \quad & i = 0, \ldots, n, \\ \epsilon_i z_i \leq y_i \leq \tau_i z_i, \quad & i = 0, \ldots, n \end{aligned}$$

with ϵ being the smallest positive values to be considered as nonzeros (this restriction ensures robustness in the designed parameters) and τ being sufficiently large upper bounds on the coefficients. Therefore, $z_i = 1$ for $i = 0, \ldots, \deg(y(s))$ and $z_i = 0$ for $i = \deg(y(s)) + 1, \ldots, n$. It is clear that $z_0 = 1$ since $y_0 > 0$. The BCP then becomes:

$$\begin{aligned} \max\ & \delta \\ \text{s.t.}\ & \epsilon_i \leq x_i \leq \tau_i, \quad i = 0, \ldots, n, \\ & \epsilon_i z_i \leq y_i \leq \tau_i z_i, \quad i = 0, \ldots, n, \\ & z_0 = 1, \\ & z_i \leq z_{i-1}, \quad i = 1, \ldots, n, \\ & x(s), y(s), (ax+by)(s) \in \mathcal{SP}, \\ & z_i \in \{0, 1\}, \quad i = 0, \ldots, n. \end{aligned}$$

Algebraic stability conditions for $x(s)$ and $(ax+by)(s)$ in the above formulation can be obtained easily from the stability tests discussed in Sect. 3, such as the Hurwitz stability criterion, since the degrees of these two polynomials are fixed at n and $n+2$, respectively. However, since the degree of $y(s)$ is variable, we must devise a way to enforce stability for variable degree polynomials.

8.2 Stability test for variable degree polynomials

Let $f(s) = \sum_{i=0}^{n} f_i s^i$ with $f_i \geq 0$, for all i. The basic idea is to use binary variables z_i to augment the left-hand side of each inequality $h(f(s)) > 0$. Consider the Hurwitz criterion for degree n polynomials and denote by Δ_n^k the principal minor of order k of the Hurwitz matrix. Then, $h_k = \Delta_n^k, k = 2, \ldots, n-1$ (h_1 corresponds to a suitable positivity constraint on the coefficients).

By inspecting the Hurwitz matrix H, we can easily see that every term of Δ_n^k contains at least one of f_n and f_{n-1}. Furthermore, Δ_{n-1}^{k-1} can be obtained from Δ_n^k by simply setting $f_n = 0$. However, if $f_n = f_{n-1} = 0$, then Δ_n^k becomes trivial ($\Delta_n^k = 0$), thus giving rise to the infeasible condition $0 > 0$. Therefore, we need to augment h_k so that $h_k = \Delta_n^k + (1 - z_n)(1 - z_{n-1})\Delta_{n-2}^{k-2}$ for all $k > 2$. Therefore, if $f_n \neq 0$ or $f_{n-1} \neq 0$, then $h_k = \Delta_n^k + 0$. Otherwise, $h_k = 0 + \Delta_{n-2}^{k-2}$. Similarly, we must keep augmenting h_k for degree $n-4, n-6, \ldots$ polynomials whenever needed, giving

$$h_{n-\bar{k}} = \sum_{i=0}^{\lfloor (n-\bar{k})/2 \rfloor - 1} \left(\Delta_{n-2i}^{n-2i-\bar{k}} \prod_{j=0}^{2i-1} (1 - z_{n-j}) \right) > 0, \quad \bar{k} = 1, \ldots, n-2,$$

where $\bar{k} = n - k$. These arguments prove to the following result:

Proposition 1 $h_{n-\bar{k}} > 0, \bar{k} = 1, \ldots, n-2$ *provide the stability conditions for a variable degree polynomial* $y(s)$ *with* $\deg(y) \leq n$.

Numerically, the stability condition is implemented as:

$$h_{n-\bar{k}} = \sum_{i=0}^{\lfloor (n-\bar{k})/2 \rfloor - 1} \left(\Delta_{n-2i}^{n-2i-\bar{k}} \prod_{j=0}^{2i-1} (1 - z_{n-j}) \right) \geq \epsilon z_{\bar{k}+2}, \quad \bar{k} = 1, \ldots, n-2.$$

Considering that $z_i \geq z_j$, for all $j > i$, we can replace $\prod_{j=0}^{2i-1}(1 - z_{n-j})$ by $1 - z_{n-2i+1}$.

The above results show that a mixed-integer nonlinear formulation of the BCP is possible once the degree of $x(s)$ has been fixed. One can progressively increase this degree to obtain increasingly more accurate approximations of δ^*. Computations with such an approach are discussed next.

8.3 Computational results

Computational experiments are performed in two stages. In the first stage, for each degree n of $x(s)$, upper bounds on $\sup \delta$ are computed using the global optimization solver GAMS/BARON [24]. This is done by relaxing the strict inequality $h > 0$ of the stability conditions to $h \geq 0$. The optimal δ value from solving this relaxed problem corresponds to δ unstabilizable by any controller of degree n. However, this provides a very tight upper bound on the supremum of stabilizable δ. These upper bounds up to $n = 8$ are shown in Table 1.

Also during this stage, the following assumptions and theoretical arguments in [8] are numerically verified:

Table 1 Summary of stabilizable δ

deg(x)	LB on max δ	UB on sup δ
3	0.922974	0.9239[a]
4	0.950412	0.9511[a]
5	0.951385	0.9521
6	0.962921	0.9749
7	0.962927	0.9749
8	0.966966	0.9848
9	0.967001	1[b]
10	0.973974	1[b]

[a]Confirmed by theoretical analysis in [8]
[b]Numerical upper bounds not available

- When $n \leq 4, y(s)$ takes a scalar value at the critical δ where $(ax + by)(s)$ has all its zeros at the origin. The global optima for these cases are the same as those obtained from the theoretical analysis in [8] under the assumption of 0-degree $y(s)$.
- When $n \leq 3$, only the stability conditions of $(ax + by)(s)$ are binding at the critical δ values. Stability of $x(s)$ and $y(s)$ are obtained for free. However, this situation begins to change when $n = 4$. At the critical value, stability conditions for both $x(s)$ and $(ax + by)(s)$ become binding.

In the second stage, we search for feasible controllers so as to determine a lower bound on the maximum stabilizable δ for each n. The results of the first stage suggest that improving the lower bound of δ^* beyond 0.96 can be accomplished only by controllers of degree $\geq$6. The local optimization solver GAMS/DICOPT [35] is used with carefully chosen scaling factors and bounds to search for local optima of the MINLP formulation. Similar solutions were obtained by GAMS/BARON but the search for global solutions required excessive computing times. These lower bounds are also included in Table 1. The zeros of x, y, and $ax + by$ are double-checked by symbolic computations in MATLAB to ensure stability. The stabilizing controllers that were identified through this process are listed in the appendix.

The differences between the lower and upper bounds in Table 1 are mainly due to numerical errors and increasing sensitivity of roots on coefficients; the limited precision supported by numerical optimization solvers, local as well as global that we used, is suitable for low degree polynomials only.

As n and the corresponding feasible δ value increase, a change in the structure of the roots near the critical δ value is observed. The stability condition on $x(s)$ kicks in when $n \geq 4$ (Fig. 3), and the stability condition on $y(s)$ also becomes active when $n \geq 6$ (Fig. 4). Figure 5 shows the location of roots when $n = 10$.

9 Conclusions

The subject of this work has been the application of global optimization to the design of stabilizing feedback controllers for linear systems. We have shown how to use symbolic computations and stability tests in order to obtain algebraic expressions for stability conditions and convert natural bilevel programs into single-level programs for stabilizing controller design. This framework is, in general, applicable to any

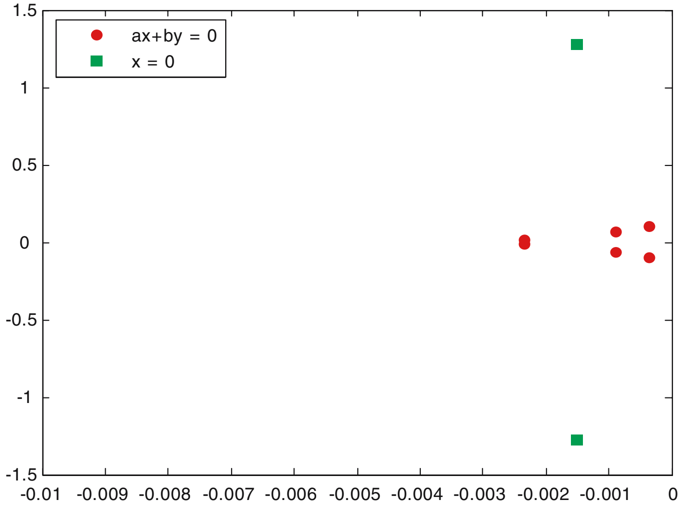

Fig. 3 Distribution of roots when $n = 4$

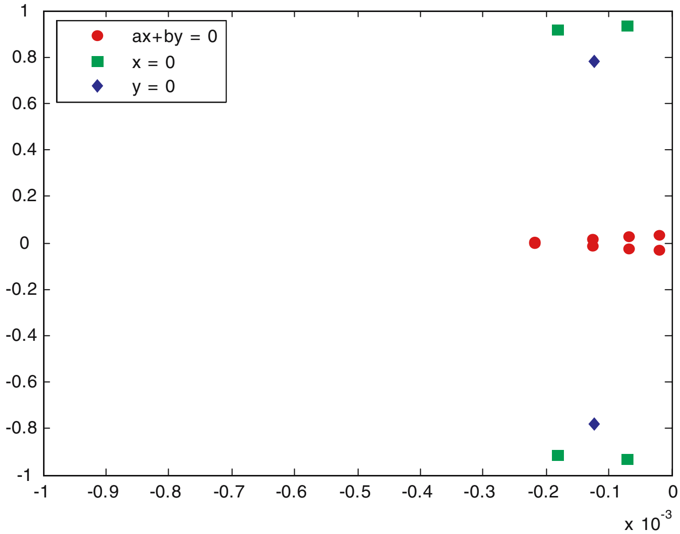

Fig. 4 Distribution of roots when $n = 6$

problem involving eigenvalue or root optimization. The proposed methodology has been demonstrated on several stabilizing controller design problems and has successfully found global solutions which were often significantly better than local solutions in the prior literature.

The proposed methodology was also used to improve the lower bound of δ^* for the Belgian chocolate problem. To address this problem, an extension of the Hurwitz criterion to variable degree polynomials was proposed. Using this criterion, global optimization was used to find valid upper bounds for increasingly more complex versions of the problem. These upper bounds guided further identification of actual

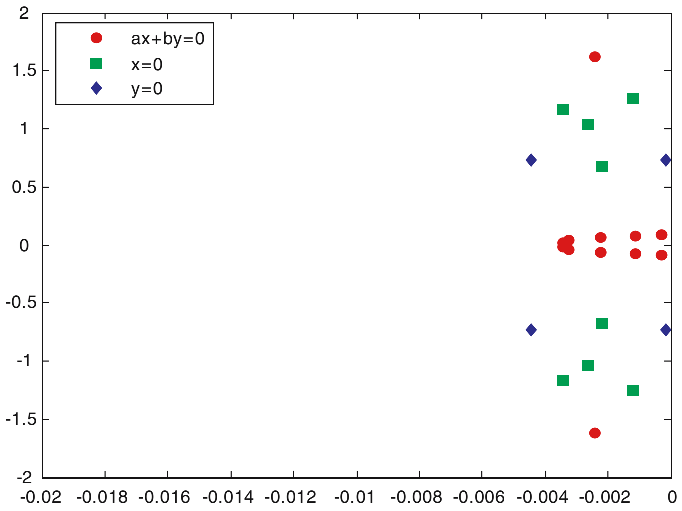

Fig. 5 Distribution of roots when $n = 10$ (large negative roots trimmed)

stabilizing controllers by local optimization. Using this approach, we improved the lower bound of δ^* from 0.96 to $\delta^* > 0.973974$. In addition, this approach also found a stabilizing controller of degree 6 for $\delta = 0.96$, which was stabilizable by a more complex controller of degree 8 in [8].

Throughout the computation, it has been observed that, as n increases, the computational effort increases and so do numerical difficulties related to scaling issues. Often times, the numerical solution (stabilizing controller) obtained by a local optimization solver does not satisfy the stability conditions and must be reconciled using extended precision arithmetic. These situations occur when the system is of high-order, thus making difficult the application of fixed point arithmetic optimization solvers to this type of problems of realistic size. We conjecture that the use of interval arithmetic (or other extended arithmetic) solvers is required to successfully solve larger versions of this problem.

Appendix: Stabilizing controllers for the Belgian chocolate problem

- $n = 3$

$$\begin{aligned}\delta &= 0.92297431,\\ x(s) &= s^3 + 1.85297039s^2 + 2.43003569s + 1.31640537,\\ y(s) &= 1.31640527.\end{aligned}$$

- $n = 4$

$$\begin{aligned}\delta &= 0.95041244930046,\\ x(s) &= s^4 + 1.90797693760092s^3\\ &\quad + 2.64067193514815s^2 + 3.11156225991536s\\ &\quad + 1.63695355,\\ y(s) &= 1.63695354.\end{aligned}$$

- $n = 5$

$$
\begin{aligned}
\delta &= 0.95138549197075,\\
x(s) &= s^5 + 1.9209227879415s^4\\
&\quad + 2.67295500988711s^3 + 2.87550963911529s^2\\
&\quad + 1.78481987973646s + 0.40523431,\\
y(s) &= 0.28987361293749s^2 + 1.0137516784761s\\
&\quad +0.405234309.
\end{aligned}
$$

- $n = 6$

$$
\begin{aligned}
\delta &= 0.96292177890276,\\
x(s) &= 3.032919s^6 + 5.84354436209896s^5\\
&\quad + 8.22680504549906s^4 + 9.9999996610297s^3\\
&\quad + 7.4139338571005s^2 + 4.27718005670024s\\
&\quad +2.22047268492766,\\
y(s) &= 3.61769956856173s^2 + 0.00089704114892s\\
&\quad + 2.22047268492765.
\end{aligned}
$$

- $n = 7$

$$
\begin{aligned}
\delta &= 0.96292783033099,\\
x(s) &= 11.06044s^7 + 26.30081098341228s^6\\
&\quad + 39.61499384240553s^5 + 49.9999957630742s^4\\
&\quad + 43.48724899045916s^3 + 27.79153257631012s^2\\
&\quad + 15.13179255749897s + 3.65437800000000,\\
y(s) &= 13.19054921281717s^3 + 5.95864022290877s^2\\
&\quad + 8.093988s + 3.65437799999999.
\end{aligned}
$$

- $n = 8$

$$
\begin{aligned}
\delta &= 0.96696634493729,\\
x(s) &= 14.6663s^8 + 28.49760772810765s^7\\
&\quad + 40.58902826861004s^6 + 50.00000141907101s^5\\
&\quad + 40.3164268091795s^4 + 27.90556720113966s^3\\
&\quad + 16.9093606305329s^2 + 4.83445335504693s\\
&\quad + 2.48675572900000,\\
y(s) &= 15.79166285880689s^4 + 0.06369042351932s^3\\
&\quad + 12.53336470723619s^2 + 0.025235159s\\
&\quad + 2.48675572899998.
\end{aligned}
$$

- $n = 9$

$$\begin{aligned}
\delta &= 0.96700163,\\
x(s) &= 42.18306595s^9 + 84.12438812709499s^8\\
&\quad + 121.042139373607s^7 + 150.0000017409253s^6\\
&\quad + 123.7858462811487s^5 + 86.74955722120957s^4\\
&\quad + 53.19559304387997s^3 + 16.69545311064127s^2\\
&\quad + 7.97777094698186s + 0.41090216000000,\\
y(s) &= 45.27489624055756s^5 + 2.65278220378291s^4\\
&\quad + 36.06738807398357s^3 + 2.08822241154759s^2\\
&\quad + 7.18308483s + 0.41090215999999.
\end{aligned}$$

- $n = 10$

$$\begin{aligned}
\delta &= 0.97397439924082,\\
x(s) &= s^{10} + 1.97351109136261s^9\\
&\quad + 5.49402092964662s^8 + 8.78344232801755s^7\\
&\quad + 11.67256448604672s^6 + 13.95449016040116s^5\\
&\quad + 11.89912895529042s^4 + 9.19112429409894s^3\\
&\quad + 5.75248874640322s^2 + 2.03055901420484s\\
&\quad + 1.03326203778346,\\
y(s) &= 0.00066128189295s^5 + 3.611364710425s^4\\
&\quad + 0.03394722108511s^3 + 3.86358782861648s^2\\
&\quad + 0.0178174691792s + 1.03326203778319.
\end{aligned}$$

References

1. Anderson, B.D.O.: Controller design: Moving from theory to practice. IEEE Control Syst. **13**(4), 16–25 (1993)
2. Balakrishnan, V., Boyd, S.: Global optimization in control system analysis and design. Control Dyn. Syst. Adv. Theory Appl. **53**, 1–56 (1992)
3. Beran, E., Vandenberghe, L., Boyd, S.: A global BMI algorithm based on the generalized Benders decomposition. In: Proceedings of the European Control Conference. Bruxelles, Belgium (1997)
4. Bilharz, H.: Bemerkung zu einem satze von (hurwitz). Zeitschrift für angewandte Mathematik und Mechanik **24**(1), 77–82 (1944)
5. Blondel, V.D.: Simultaneous Stabilization of Linear Systems, volume 191 of Lecture notes in Control and Information Sciences. Springer-Verlag, London (1994)
6. Blondel, V.D., Gevers, M.: Simultaneous stabilizability of three linear systems is rationally undecidable. Math. Control Signals Syst. **6**(2), 135–145 (1993)
7. Blondel, V.D., Tsitsiklis, J.H.: NP-hardness of some linear control design problems. SIAM J. Control Optim. **35**(6), 2118–2127 (1997)
8. Burke, J.V., Henrion, D., Lewis, A.S., Overton, M.L.: Analysis of a Belgian chocolate stabilization problem. LAAS-CNRS Research Report, 05164, Toulouse, France (2005)
9. Chang, Y., Sahinidis, N.V.: Optimization of metabolic pathways under stability considerations. Comput. Chem. Eng. **29**(3), 467–479 (2005)
10. Fares, B., Apkarian, P., Noll, D.: An augmented Lagrangian method for a class of LMI-constrained problems in robust control theory. Int. J. Control **74**(4), 348–360 (2001)
11. Gantmacher, F.R.: Applications of the Theory of Matrices. Interscience, New York (1959). Translation and revision by J. L. Brenner

12. Goh, K.-C., Safonov, M.G., Papavassilopoulos, G.P.: Global optimization for the biaffine matrix inequality problem. J. Glob. Optim. **7**(4), 365–380 (1995)
13. Hassibi, A., How, J., Boyd, S.: A path-following method for solving BMI problems in control. In: Proceedings of American Control Conference. San Diego, California, USA (1999)
14. Hurwitz, A.: Ueber die bedingungen, unter welchen eine gleichung nur wurzeln mit negativen reellen theilen besitzt. Math. Annalen **46**(2), 273–284 (1895)
15. Kawanishi, M., Sugie, T., Kanki, H.: BMI global optimization based on branch and bound method taking account of the property of local minima. In: Proceedings of the 36th IEEE Conference on Decision and Control, vol. 1, pp. 781–786 (1997)
16. Leibfritz, F.: COMPl$_e$ib: Constraint matrix-optimization problem library–A collection of test examples for nonlinear semidefinite programs, control system design and related problems. Technical report, University of Trier (2003)
17. Liénard, A., Chipart, M.H.: Sur la signe de la partie réelle des racines d'une équation algébrique. J. Math. Pures Appl. **10**(4), 291–346 (1914)
18. McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: Part I – Convex underestimating problems. Math. Program. **10**(2), 147–175 (1976)
19. Patel, V.V.: Solution to the "Champagne problem" on the simultaneous stabilisation of three plants. Syst. Control Lett. **37**(3), 173–175 (1999)
20. Patel, V.V., Deodhare, G., Viswanath, T.: Some applications of randomized algorithms for control system design. Automatica **38**(12), 2085–2092 (2002)
21. Routh, E.J. (ed.): A Treatise on the Stability of a Given State of Motion. Macmillan and Co., London (1877)
22. Ryoo, H.S., Sahinidis, N.V.: A branch-and-reduce approach to global optimization. J. Glob. Optim. **8**, 107–139 (1996)
23. Sahinidis, N.V.: BARON: a general purpose global optimization software package. J. Glob. Optim. **8**, 201–205 (1996)
24. Sahinidis, N.V., Tawarmalani, M.: BARON 7.5: Global Optimization of Mixed-Integer Nonlinear Programs. User's Manual, Champaign, Illinois, USA (2005)
25. Sasaki, T., Murao, H.: Efficient Gaussian elimination method for symbolic determinants and linear systems. ACM Trans. Math. Software **8**(3), 277–289 (1982)
26. Schwarz, H.R.: Ein verfahren zur stabilitätsfrage bei matrizen-eigenwerte-probleme. Zeitschrift für angewandte Mathematik und Physik **7**(5), 473–500 (1956)
27. Shectman, J.P., Sahinidis, N.V.: A finite algorithm for global minimization of separable concave programs. J. Glob. Optimi. **12**, 1–36 (1998)
28. Strelitz, Sh.: On the Routh-Hurwitz problem. Am. Math. Monthly **84**(7), 542–544 (1977)
29. Tawarmalani, M., Ahmed, S., Sahinidis, N.V.: Product disaggregation and relaxations of mixed-integer rational programs. Optim. Eng. **3**(3), 281–303 (2002)
30. Tawarmalani, M., Sahinidis, N.V.: Convex extensions and envelopes of lower semi-continuous functions. Math. Program. **93**(2), 247–263 (2002)
31. Tawarmalani, M., Sahinidis, N.V.: Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications, volume 65 of Nonconvex Optimization and Its Applications. Kluwer Academic Publishers, Dordrecht (2002)
32. Tawarmalani, M., Sahinidis, N.V.: Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. Math. Program. **99**(3), 563–591 (2004)
33. Tawarmalani, M., Sahinidis, N.V.: A polyhedral branch-and-cut approach to global optimization. Math. Program. **103**, 225–249 (2005)
34. Tuan, H.D., Apkarian, P., Nakashima, Y.: A new Lagrangian dual global optimization algorithm for solving bilinear matrix inequalities. Int. J. Robust Nonlinear Control **10**(7), 561–578 (2000)
35. Viswanathan, J., Grossmann, I.E.: A combined penalty function and outer-approximation method for MINLP optimization. Comput. Chem. Eng. **14**(7), 769–782 (1990)