16th European Symposium on Computer Aided Process Engineering
and 9th International Symposium on Process Systems Engineering
W. Marquardt, C. Pantelides (Editors)
© 2006 Published by Elsevier B.V.

Branch and Bound Methods for Control Structure Design

Vinay Kariwala* and Sigurd Skogestad†

Department of Chemical Engineering, Norwegian University of Science and Technology,
N-7491 Trondheim, Norway

We demonstrate the potential of branch and bound methods for handling the combinatorial problems arising in control structure design. The method efficiently solves large-scale benchmark problems for selection of controlled variables using minimum singular value rule and variable selection for stabilization. We also provide an overview of some other interesting problems that can be solved in the same framework using non-standard bounding strategies.

Keywords: Branch and bound, Control structure design, Controlled variables, Decentralized control, Stabilization.

1. Introduction

Control structure design (CSD) deals with the selection of controlled and manipulated variables, and the links interconnecting them [1]. Many of the available methods for CSD require an enumeration over the entire set of alternatives and thus suffer from computational intractability. In this paper, we show that for many problems arising in CSD, the combinatorial growth in the number of alternatives can be handled efficiently using branch and bound (B&B) method.

Effective B&B methods require tight lower and upper bounds on the objective function value for all feasible solutions that can be generated by expanding upon the partial solution (node). It works by eliminating partial solutions, which cannot give a better solution than the best known lower bound (often taken as the objective function value for best known integer solution). As an example, consider that we want to select m out of n variables by maximizing Φ and $\Phi^{'}$ is the best known lower bound. Let Ω be the set of feasible solutions, which can be generated by expanding upon the partial solution. Then, Ω can be eliminated (pruned), if an upper bound on $\max_{\Omega} \Phi < \Phi^{'}$.

The use of B&B methods for CSD problems has earlier been explored in [2] and also in [3], where the minimization of RGA-number [1] was shown to be a mixed integer linear program. Here, we apply B&B method using different bounding strategies to: (a) controlled variable selection using minimum singular value (MSV) rule [1] and; (b) variable selection for stabilization. The latter problem has also been considered in [4], but the bounding strategy used here is computationally more efficient. Later in this paper, we also give an overview of some other problems that can be cast in the B&B framework.

*Present Address: Division of Chemical & Biomolecular Engg., Nanyang Technological University, Singapore.
†Corresponding Author: Tel: +47-735-94154; Fax: +47-735-94080; E-mail:skoge@chemeng.ntnu.no

2. Controlled variable selection using MSV rule

The MSV rule [1] is a simple method for selecting the set of primary controlled variables (CVs). Let J denote the scalar economic objective function and, G and G_d represent the linearized (unscaled) process and disturbance gain matrices, respectively. For finding the locally optimal set of CVs, the rule requires that $\underline{\sigma}(G')$ be maximized where $\underline{\sigma}$ is the MSV and G' is the scaled gain matrix. The output and input scaling matrices are $J_{uu}^{0.5}$ and $\mathrm{diag}(|G\,J_{uu}^{-1}\,J_{ud} - G_d| + e_n)$, respectively. Here, $J_{xy} = \partial J / \partial x\, \partial y$ and e_n is the implementation error (see [1] for details).

A quick observation reveals that $(G\,J_{uu}^{-1}\,J_{ud}{-}G_d)$ can be calculated beforehand, as the scaling procedure is independent of the choice of CVs. Thus the MSV rule is equivalent to squaring down a matrix such that the MSV of the squared matrix is maximum. Let G_{all} be the scaled gain matrix for all candidate CVs and $H = \mathrm{diag}(h_i)$, $h_i = \{0, 1\}$, $\sum_i h_i = m$ be the decision matrix such that j^{th} CV is selected, if $h_j = 1$. The MSV rule requires solving

$$\max_{H} \underline{\sigma}(H\,G_{\text{all}}) \tag{1}$$

2.1. Bounding strategy

When the constraint $h_i = \{0,\, 1\}$ is relaxed as $0 \le h_i \le 1$, (1) can be posed as a linear matrix inequality (LMI). The solution to this relaxed convex program provides the desired upper bound for pruning branches. A computationally cheaper approach, used in this paper, is based on Weyl's inequalities. Let σ_i be the singular values, where $\sigma_{i+1} \le \sigma_i$. For $A \in \mathbb{R}^{p \times q}$, $p < q$, [5, Th. 4.3.6],

$$\sigma_{(p+1)} \begin{pmatrix} A \\ b \end{pmatrix} \le \sigma_p(A) \tag{2}$$

In words, when rows are added to a fat matrix, the least non-zero singular value is non-increasing. Thus, the least non-zero singular value of a partially completed solution provides the required upper bound on the MSV. The lower bound is obtained sequentially by adding one variable to the partially completed solution at every step such that the least non-zero singular value is maximum, until the resulting matrix is square. To illustrate these concepts, we consider an example where we want to select 3 of 4 CVs

$$G_{\text{all}} = \begin{bmatrix} 10 & 10 & 10 \\ 10 & 9 & 1 \\ 2 & 1 & 3 \\ 2 & 1 & 0.1 \end{bmatrix}$$

With only y_1 selected $\sigma_1 = 17.321$, where as σ_2 for sets $\{y_1,\, y_2\}$, $\{y_1,\, y_3\}$ and $\{y_1,\, y_4\}$ is 5.699, 1.386 and 1.337, respectively. Thus, 17.321 is the upper bound for node $\{y_1\}$ as it bounds σ_2 (and σ_3) for all solutions that are generated by expanding it. The sequential approach first adds y_2 to the set $\{y_1\}$ and then y_3 to $\{y_1,\, y_2\}$. The set $\{y_1,\, y_2,\, y_3\}$ has $\sigma_3 = 0.78$, which is a lower bound for $\{y_1\}$. Similarly, for $\{y_1,\, y_2\}$, 5.699 and 0.78 are the upper and lower bounds, respectively.

2.2. Application

For this problem, we use a breadth-first B&B method. The initial feasible solution is obtained using the same sequential approach as used for obtaining the lower bound. Numerical results suggest that the sequential approach usually performs better than other alternatives for generating an initial solution, *e.g.* non-square relative gain array (RGA) [1].

The B&B method is applied to the Hydrodealkylation of Toluene (HDA) process, which has 8 degrees of freedom and 129 candidate CVs giving 1.52×10^{12} alternatives. The MSVs of the solutions obtained using sequential and RGA methods are 13.63 and 2, respectively, which emphasizes the superiority of the sequential approach. The solution obtained using B&B method has an MSV of 14.73, which requires evaluation of 8.1×10^5 alternatives ($5.3 \times 10^{-5}\%$ of total alternatives). Motivated by controllability reasons, Cao *et al.* [2] proposed using an asymmetrical tree based B&B method for solving (1). In comparison, their method requires evaluation of 2.5×10^6 alternatives. Though the improvement in MSV is only minor as compared to the sequential approach for this process, in general, the relative difference can be as large as 70%, as confirmed using randomly generated matrices.

3. Variable selection for stabilization

For complex unstable systems, often the system is first stabilized using a subset of CVs and manipulated variables (MVs) and then another controller is designed to satisfy the performance requirements. The question remains: which CVs and MVs should be used for stabilization? These variables can be selected such that the input usage for stabilization is minimized as it reduces the likelihood of otherwise destabilizing valve saturation [1].

The $\mathcal{H}_\infty$-optimal achievable input performance is given as $\min_{K(s)} \|KS(s)\| = \underline{\sigma}_H^{-1}(\mathcal{U}(G(s))^*)$, where $\mathcal{U}(G(s))^*$ is the mirror image of anti-stable part and $\underline{\sigma}_H$ is the minimum Hankel singular value (HSV) [6]. Then, the optimal subset of variables is chosen by maximizing $\underline{\sigma}_H(\mathcal{U}(G(s))^*)$. For systems with single unstable pole, the optimal solution can be obtained using pole vectors [1]. This approach, however, cannot provide the optimal solution for systems with multiple unstable poles and a method for obtaining the same is discussed next.

3.1. Bounding Strategy

Without loss of generality, we assume that $G(s)$ has only unstable poles. Let the triplet (A, B, C) denote the minimal state-space realization of $G(s)$. Then, $\underline{\sigma}_H(G(s)^*) \geq \gamma$ if and only if $P, Q \geq 0$ solve [7]

$$\begin{aligned} -A^* P - P A + C^* C &\geq 0 \\ -Q A^* - A Q + B B^* &\geq 0 \\ \begin{bmatrix} P & \gamma I \\ \gamma I & Q \end{bmatrix} &\geq 0 \end{aligned} \tag{3}$$

When the system has distinct unstable poles, P, Q can be removed by noting [6]

$$P = \sum_{i=1}^{n_y} ((C_{*i})^* C_{*i}) \circ M \tag{4}$$

$$Q = \sum_{j=1}^{n_u} (B_{i*} B_{i*}^*) \circ M \tag{5}$$

where $M_{ij} = 1/(p_i + p_j^*)$, p_i are unstable poles, $\circ$ is the Hadamard product and B_{j*}, C_{*i} denote the j^{th} row and i^{th} column of B and C, respectively. Note that the products depend only on the

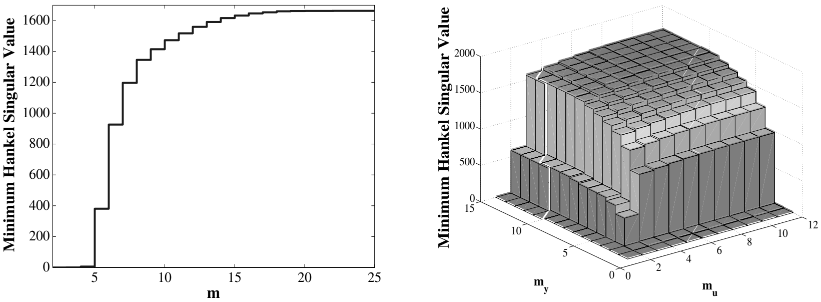
i^{th} CV and j^{th} MV, respectively. Then the variable selection problem requires solving

$$\begin{bmatrix} \sum_{i=1}^{n_y} \alpha_i \, ((C_{*i})^* \, C_{*i}) \circ M & \gamma I \\ \gamma I & \sum_{j=1}^{n_u} \beta_j \, (B_{i*} \, B_{i*}^*) \circ M \end{bmatrix} \; \geq \; 0 \tag{6}$$

where $\alpha_i, \, \beta_j \; = \; \{0, \, 1\}$. The i^{th} CV is selected if $\alpha_i \; = \; 1$ and *vice versa*. For selecting m variables out of $(n_y + n_u)$ CVs and MVs, we need the additional constraint $\sum_i \alpha_i + \sum_j \beta_j = m$. An upper bound is obtained by relaxing the decision variables as $0 \leq \alpha_i, \, \beta_j \leq 1$ and a lower bound is obtained by rounding-off $\alpha, \, \beta$ that optimally solves the convex program. Note that the assumption of distinct unstable poles can be satisfied by numerically perturbing the poles.

3.2. Application

There are many software packages available for solving integer semi-definite programs. In this paper, we use the B&B algorithm available with Yalmip [8], where the package Sedumi 1.05 [9] is used for solving the LMI problems resulting on relaxation of binary variables.



(a) Total number of CVs and MVs fixed (b) Number of CVs and MVs fixed individually

Figure 1. Selection of variables for stabilization of Tennessee Eastman process.

The algorithm is applied to Tennessee Eastman process with six unstable poles. The model with 22 CVs and 12 MVs is scaled as outlined in [4]. The variation of maximum HSV with the total number of CVs and MVs fixed is shown in Figure 1(a). Cao and Saha [4] suggest using $m = 6$, for which maximum HSV is 926.65. The bounding strategy used in [4] is similar to the strategy used for bounding MSV. Out of 1.72×10^{10} alternatives, the proposed method evaluates only 1441 alternatives cumulatively for all m, which is an order of magnitude less than reported in [4]. It is also possible to fix the number of CVs and MVs individually through the constraints $\sum_i \alpha_i \; = \; m_y$, $\sum_j \beta_j \; = \; m_u$ and the results are shown in Figure 1(b). For $m_y \; = \; m_u \; = \; 3$, the optimal solution is same as for $m = 6$, as expected.

4. Other problems

Similar to the problems discussed earlier, the combinatorial nature of many other problems arising in CSD can be handled using B&B methods. Usually, the challenging task is to find a tight upper bound on the objective function. Next, we discuss some problems requiring novel manipulations of the partial solution to get monotonic relationships useful as upper bounds.

The first problem consists of selecting pairings such that $\mu(E)$ is minimized (or $-\mu(E)$ is maximized), where μ is the structured singular value [1] and $E = (G - G_{\text{diag}})G_{\text{diag}}^{-1}$ with G_{diag}

consisting of diagonal elements of G. Selection of pairings with minimal $\mu(E)$ (equivalent to finding optimal permutation of G) puts least restrictions on decentralized controller synthesis using independent designs [1]. When there exists a pairing with $\mu(E) < 1$, iterative-RGA provides the optimal solution explicitly [1], otherwise a B&B method is required and the bounding strategy is discussed next. For any given partial solution (node), a lower bound on $-\mu(E)$ can be obtained using the sequential approach discussed in § 2. Now, assume that G is permuted such that the $(1, 1)$-block of permuted G correspond to gain of partially selected pairings. By partitioning E similar to G, we then have $-\mu(E_{11}) \geq -\mu(E)$ [1], which serves as the upper bound on all the alternatives that can be generated by expanding on this node. This happens as any expansions on this node do not affect E_{11}.

Another problem consists of selecting pairings such that the achievable decentralized output performance (J_{decen}) is minimal. Though exact characterization of J_{decen} is an open problem, sub-optimal methods providing upper bounds on J_{decen} are available. Then, at any node, the lower bound on $-J_{\text{decen}}$ is obtained by completing the partial solution using any reasonable heuristic and the sub-optimal methods. For a given pairing, an upper bound on $-J_{\text{decen}}$ was recently presented in [10]. This method can be adapted for partially completed solutions by using a block decentralized controller, where the remaining (unpaired) CVs and MVs are controlled using a full multivariable controller. As only bounds on J_{decen} can be computed, B&B method may end with a number of alternatives, for all of which the upper bound on $-J_{\text{decen}}$ is larger than the best available lower bound. Nevertheless, B&B method is useful for eliminating inferior pairing alternatives.

5. Conclusions

B&B methods can efficiently handle the combinatorial problems arising in CSD. In comparison with standard mixed integer non-linear programs, CSD problems require novel bounding and search strategies. We demontrated the use of monotonic relationships and integer semi-definite programs for obtaining tight upper bounds on partial solutions. The algorithms scale well with problem dimensions and are shown to provide optimal solutions for large-scale benchmark problems. The future work will focus on extending these ideas to multi-objective optimization problems.

REFERENCES

1. S. Skogestad and I. Postlethwaite. John Wiley & Sons Ltd., Chichester, UK, 2005.
2. Y. Cao, D. Rossiter and D. H. Owens, Proceedings of DYCOPS 5, Korfu, Greece, 1998.
3. I. K. Kookos and J. D. Perkins. Ind. Eng. Chem. Res., 40, 2001.
4. Y. Cao and P. Saha, Chem. Engg. Sci., 60, 2005.
5. R. Horn and C. Johnson. Cambridge University Press, New York, NY, USA, 1985.
6. V. Kariwala, S. Skogestad, J. F. Forbes and E. S. Meadows. Intl. J. Control, 78, 2005.
7. S. Boyd., L. Ghaoui, E. Feron and V. Balakrishnan. SIAM, Philadelphia, PA, USA, 1994.
8. J. Löfberg. Proceedings of CACSD, Taipei, Taiwan, 2004. (Available from `http://control.ee.ethz.ch/~joloef/yalmip.php`).
9. J. F. Sturm, Optimization Methods and Software, 11-12, 1999. (Available from `http://sedumi.mcmaster.ca/`).
10. V. Kariwala, J. F. Forbes and E. S. Meadows. Proceedings of ACC, Boston, MA, 2005.

6. Appendix: Average performance

We further evaluate the performance of B&B methods for maximization of MSV and HSV using randomly generated examples. For maximization of MSV, the method is applied to 1000 randomly generated matrices of different sizes and the results are shown in Table 1.

Table 1
Performance of B&B method for maximization of MSV for randomly generated matrices

n	m	Alternatives	% of alternatives evaluated				Mean CPU Time
			Mean	Median	Max	Min	(sec)
10	3	120	44.1	40.8	132.5	11.7	0.0042
20	3	1140	11.1	8.8	66.5	2.1	0.0067
20	5	15504	13.7	10.9	75.2	0.6	0.1188
30	5	142510	3.8	2.8	37.1	0.1	0.3409

For maximization of HSV, the efficiency of this algorithm is tested using 100 random square transfer matrices of different dimensions generated using Matlab command *rss* and the results are shown in Table 2. For the last two cases, the maximum number of alternatives evaluated are not reported, as they were restricted to 5000 and 20000, respectively, and there exists examples that exceed these bounds.

Table 2
Performance of B&B method for maximization of HSV for randomly generated systems

n_p	n	m	Alternatives	% of alternatives evaluated				Mean CPU Time
				Mean	Median	Max	Min	(sec)
3	5	3	120	43.3	32.5	155.8	0.8	4.0657
5	5	3	120	84.5	87.5	142.5	11.7	10.858
5	10	3	1140	63.3	70.5	134.6	0.6	75.631
5	10	5	15504	4.17	1.5	-	0.0	56.057
5	15	5	142510	1.7	0.2	-	0.0	298.67

Some of the salient observations are:

1. The efficacy of the method, measured in terms of the number of the alternatives evaluated, increases with problem dimensions. This is reasonable, as many more branches can be pruned, when less variables are selected.
2. There exist cases that are worse than complete enumeration. As most combinatorial problems are $\mathcal{NP}$-hard, the worst-case performance is likely to increase non-polynomially with problem dimensions, when *any* ingenious B&B strategies are used.
3. The average performance is much worse than the large-scale benchmark problems considered previously. This happens as real problems often show more structure than randomly generated problems. For example, for the HDA process, the scaled gain for 23 out of 129 candidate CVs is nearly zero, which are eliminated in the first step of the B&B method resulting in substantially reduced search space.
4. As we need to solve an LMI problem iteratively for finding an upper bound for the HSV maximization problem, the mean CPU time is much larger than the MSV maximization problem. An advantage of this formulation is that there exist cases, for which the optimal integer solution is obtained through evaluations of only one alternative.