

Branch and Bound Method for for Multiobjective Pairing Selection ¹

Vinay Kariwala ^a Yi Cao ^b

^aDivision of Chemical & Biomolecular Engineering, Nanyang Technological University, Singapore 637459

^bSchool of Engineering, Cranfield University, Cranfield, Bedford MK43 0AL, UK

Abstract

Most of the available methods for selection of input-output pairings for decentralized control require evaluation of all alternatives to find the optimal pairings. As the number of alternatives grows rapidly with process dimensions, pairing selection through an exhaustive search can be computationally forbidding for large-scale processes. Furthermore, the different criteria can be conflicting necessitating pairing selection in a multiobjective optimization framework. In this paper, an efficient branch and bound (BAB) method for multiobjective pairing selection is proposed. The proposed BAB method is illustrated through a biobjective pairing problem using selection criteria involving the relative gain array and the μ -interaction measure. The computational efficiency of the proposed method is demonstrated by using randomly generated matrices and the large-scale case study of cross-direction control.

Key words: Computer-aided control system design, Decentralized control, Global optimization, Large-scale systems, Multiobjective optimizations, Relative gain array, Structured singular value.

1 Introduction

Decentralized controllers are widely used in process industries. The selection of input-output pairings is a key step in the design of decentralized controllers. In some cases, pairings can be selected based on process knowledge and heuristics. For interacting multivariable processes often encountered in industries [1], however, systematic tools are needed to complement the engineering insights. During the past few decades, a number of useful tools have been developed for selection of appropriate pairings [31,33]. Most of these tools require evaluation of all alternatives to find the optimal pairings. The large number of pairing alternatives ($n!$ for an $n \times n$ process) makes pairing selection through an exhaustive search computationally forbidding for large-scale processes.

For efficiently selecting pairings based on a single criterion, genetic algorithms [13] and mixed integer linear programs (MILP) [18] have been used. Different criteria, however, address different properties of the closed-

loop system and are often conflicting in nature [5]. This conflict necessitates pairing selection in a multiobjective optimization framework such that the practicing engineer can select the appropriate pairings from the Pareto-optimal set by trading-off different criteria.

Multiobjective pairing selection has not been considered in the literature earlier. Closely related multiobjective problems like controllability analysis [9], combined variable and pairing selection [23], and integrated design and control [26] have been addressed using weighted sum and ϵ -constraint approaches. A drawback of these approaches is that the choice of weights and constraint limits is non-trivial [27]. Furthermore, the Pareto-optimal set obtained using the weighted sum approach is not necessarily complete [10]. Homotopy techniques [29] can overcome the former limitation, but are only applicable to biobjective problems. Recently, evolutionary algorithms (EAs) [10] have been applied to solve various multiobjective control system design problems, see *e.g.* [12,30]. Although useful for obtaining practical solutions, EAs do not guarantee global optimality.

Branch and bound (BAB) methods can provide globally optimal solutions for combinatorial problems [11]. Recently, the usefulness of BAB has been demonstrated for solving various single-objective problems related to

¹ A preliminary version of this work was presented at 17th IFAC world congress held in Seoul, Korea [19].

² Corresponding Author: Tel: +65-6316-8746; Fax: +65-6794-7553 ; E-mail: vinay@ntu.edu.sg

the selection [2,6,8,19–21] and analysis [17,34] of control structures. In the area of multiobjective combinatorial optimization, BAB methods have been used for solving MILPs [28] and subset selection problems [32]. Multiobjective permutation problems have only been solved using BAB method by converting the problem to a single-objective problem, *e.g.* using weighted sum method [4,25]. The main contribution of this paper is the development of a BAB framework to directly handle the multiobjective nature of the permutation problem, for which the pairing problem is a special case.

The proposed BAB framework is general and can handle most of the available pairing selection criteria simultaneously. In this paper, we consider a biobjective problem with selection criteria involving the relative gain array (RGA) [3] and the μ -interaction measure (μ -IM) [14] to illustrate the usefulness of the proposed approach. For pruning, we derive a number of lower bounds on these criteria. Furthermore, efficient pruning conditions are proposed for the μ -IM to avoid the evaluation of the expensive lower bound. This novel feature significantly improves the computational efficiency of the BAB approach. We use random matrices as well as the large-scale case study of cross-direction control [24] to demonstrate the computational efficiency of the proposed method.

Notation: The unordered and ordered sets consisting of elements a and b are denoted as $\{a, b\}$ and (a, b) , respectively. We define N_n as the set of first n natural numbers, *i.e.* $N_n = \{1, 2, \dots, n\}$, where the subscript n denotes the size of the set. $\mathbb{P}(X_n)$ represents the ensemble of all possible permutations of the elements of X_n . For $A \in \mathbb{C}^{n \times n}$ and $P_n \in \mathbb{P}(N_n)$, A_{P_n} denotes the permuted matrix obtained from A by indexing the columns with P_n . $G(s)$ denotes the transfer function matrix relating the inputs u and outputs y of the process. $G(s)$ evaluated at the frequency ω and steady-state are represented as $G(j\omega)$ and G , respectively. For given P_n , the pairings are selected on the diagonal elements of G_{P_n} .

2 BAB for Multiobjective Permutation Problems

Multiobjective permutation problems like pairing selection, flowshop scheduling [25], data seriation [4] and travelling salesman problem [15] involve solving

$$\begin{aligned} \min_{P_n \in \mathbb{P}(N_n)} & \left[J_1(P_n), J_2(P_n), \dots, J_m(P_n) \right]^T & (1) \\ \text{s.t.} & L_i(P_n) \leq 0; \quad i = 1, 2, \dots, \ell & (2) \end{aligned}$$

where J_1, J_2, \dots, J_m are the m selection criteria and L_1, L_2, \dots, L_ℓ denote the ℓ constraints. The non-dominated solutions of (1)-(2) construct the Pareto-optimal set \mathcal{P} , where every pair of $P_n^i, P_n^j \in \mathcal{P}$ satisfy

$$\exists s, t \in N_m : J_s(P_n^i) < J_s(P_n^j), J_t(P_n^i) > J_t(P_n^j) \quad (3)$$

To solve the optimization problem in (1)-(2), BAB *branches* the problem into several non-overlapping sub-problems. A sub-problem is *pruned*, if all solutions of the sub-problem are dominated by a member of \mathcal{P} , else the sub-problem is branched further. Whenever an n -element solution is reached, which is also Pareto-optimal, \mathcal{P} is *updated* using (3). The pruning of non-optimal subproblems allows BAB to gain efficiency in comparison with exhaustive search. Next, we provide further details of branching and pruning.

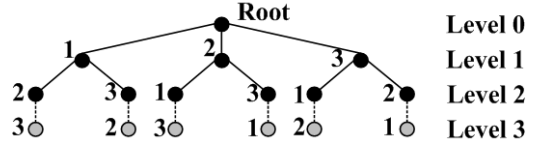


Fig. 1. Solution tree for $n = 3$

Branching. The implementation of BAB schemes requires a solution tree containing all possible alternatives. A node in the solution tree is denoted as a 2-tuple, *i.e.* $S = (F_f, C_c)$. Here, the fixed set F_f is an ordered set with f elements selected from N_n and denotes a partial solution. The candidate set $C_c = N_n \setminus F_f$ is an unordered set, whose elements can be freely chosen to append F_f . The solution tree is branched as follows:

Definition 1 For a node $S = (F_f, C_c)$ with $C_c = \{c_i\}$, the fixed and candidate sets of the i^{th} sub-node $S^i = (F_{f+1}^i, C_{c-1}^i)$, $i = 1, 2, \dots, c$ are defined as $F_{f+1}^i = (F_f, c_i)$ and $C_{c-1}^i = C_c \setminus c_i$.

The solution tree branched based on Definition 1 has $(n+1)$ levels. The label of a node denotes the element being moved from C_c to F_f . The $n!$ terminal nodes (marked by grey circles in Figure 1) represent different solutions $P_n \in \mathbb{P}(N_n)$.

Proposition 2 Every terminal node of the solution tree branched based on Definition 1 belongs to one and only one branch.

Proposition 2 implies that the solution tree is complete and non-redundant. The completeness of the solution tree ensures global optimality of the solution, whilst non-redundancy is important for efficiency of BAB.

Pruning. Let the ensemble of all n -element ordered sets, obtained by expanding F_f of node $S = (F_f, C_c)$, be

$$\mathcal{S} = \{(F_f, P_{n-f}) | P_{n-f} \in \mathbb{P}(C_c)\} \quad (4)$$

Let $\underline{J}_i(\mathcal{S})$ be a lower bound of J_i , calculated over all the elements of \mathcal{S} , *i.e.* $\underline{J}_i(\mathcal{S}) \leq J_i(P_n)$ for all $P_n \in \mathcal{S}$. If $\exists P_n^j \in \mathcal{P}$, where \mathcal{P} is the current Pareto-optimal set, such that $\underline{J}_i(\mathcal{S}) \geq J_i(P_n^j)$ with strict inequality occurring for at least one $i \in N_m$, then \mathcal{S} cannot contain

any Pareto solutions and hence can be pruned without any evaluations. Pruning can be performed similarly if $\underline{L}_k(\mathcal{S}) > 0$ for $k \in N_\ell$, where $\underline{L}_k(\mathcal{S}) < L_k(P_n) \forall P_n \in \mathcal{S}$.

3 A Biobjective Pairing Selection Problem

The proposed BAB framework can be applied to solve the multiobjective pairing selection problem. In this case, level i in the solution tree shown in Figure 1 represents output y_i and the label of the node corresponds to the input with which y_i is paired. To illustrate the proposed BAB method, we consider a biobjective pairing problem with selection criteria involving the RGA and the μ -IM. The BAB method can be easily extended to handle other selection criteria; see [19] for details.

3.1 Relative Gain Array

For $G(j\omega) \in \mathbb{C}^{n \times n}$, the RGA is defined as [3,31]

$$\Lambda(j\omega) = G(j\omega) \circ G^{-H}(j\omega) \quad (5)$$

where \circ represents the Hadamard product. For pairing selection, the following rules are often used [31]:

- (1) Avoid pairings on the negative elements of $\Lambda(0)$.
- (2) Prefer pairings such that $\Lambda_{P_n}(j\omega_B)$ is close to the identity matrix, where ω_B is the (expected) bandwidth frequency.

The first rule is a necessary condition for integrity of the closed-loop system against loop failure [22]. The second rule is based on interpreting the RGA as an interaction measure. Pairings can be selected according to the second rule by minimizing the RGA-number defined as [31]

$$\text{RGA-number}(\omega) = \|\Lambda_{P_n}(j\omega) - I\|_{\text{sum}} \quad (6)$$

where $\|\cdot\|_{\text{sum}}$ is the sum-norm [31].

3.2 μ -Interaction Measure

The decentralized controller $K(s)$ is often designed using the independent design method [31]. In this method, controller design can be viewed as being based on $\tilde{G}(s)$, which consists of the diagonal elements of $G_{P_n}(s)$. When the μ -IM condition is satisfied, *i.e.* [14]

$$\bar{\sigma}(\tilde{T}(j\omega)) < \mu_\Delta^{-1}(E(j\omega)) \quad \forall \omega \in \mathbb{R} \quad (7)$$

any $K(s)$ stabilizing $\tilde{G}(s)$ also stabilizes $G_{P_n}(s)$. In (7), μ denotes the structured singular value [31] computed with a diagonal Δ , $\tilde{T}(s) = \tilde{G}K(s)(I + \tilde{G}K(s))^{-1}$ and

$$E(s) = (G_{P_n}(s) - \tilde{G}(s))\tilde{G}^{-1}(s) \quad (8)$$

It follows from (7) that if pairings are chosen such that $\mu_\Delta(E(j\omega))$ is small at all ω , the restrictions on designing $K(s)$ using independent design method is minimum. $\mu_\Delta(E(j\omega))$ is also a measure of generalized diagonal dominance, where $G_{P_n}(j\omega)$ is said to be generalized diagonally dominant if $\mu_\Delta(E(j\omega)) < 1$ [31]. A pairing, for which $\mu_\Delta(E(j\omega)) < 1$, can be easily found using the iterative RGA [31]. In absence of existence of such a pairing, a BAB method can be used to find the pairing for which $G_{P_n}(j\omega)$ is closest to being diagonally dominant in the sense that $\mu_\Delta(E(j\omega))$ is as close to 1 as possible. As the exact computation of μ is computationally intractable, we instead minimize the upper bound on μ (denoted as $\bar{\mu}$) obtained through D -scaling method [31].

3.3 Lower Bounds on selection criteria

Without loss of generality, we consider pairing selection by minimizing the RGA-number and $\bar{\mu}_\Delta(E)$ evaluated at steady-state, which requires solving

$$\begin{aligned} \min_{P_n \in \mathbb{P}(N_n)} & \left[\|\Lambda_{P_n} - I\|_{\text{sum}}, \bar{\mu}_\Delta(G_{P_n}(I \circ G_{P_n})^{-1} - I) \right]^T \quad (9) \\ \text{s.t.} & \quad \Lambda_{P_n}]_{ii} > 0; \quad i = 1, 2, \dots, n \quad (10) \end{aligned}$$

For pruning, BAB method requires lower bounds on the RGA-number and the μ -IM, calculated over \mathcal{S} in (4), which are presented next.

Proposition 3 For a node $S = (F_f, C_c)$, let

$$M = |\Lambda - 1_{nn}| - |\Lambda| = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \quad (11)$$

where 1_{nn} is an $n \times n$ matrix of 1's and, $M_{11} \in \mathbb{R}^{f \times f}$ and $M_{22} \in \mathbb{R}^{c \times c}$, respectively. Then,

$$\begin{aligned} \min_{P_n \in \mathcal{S}} \|\Lambda_{P_n} - I\|_{\text{sum}} & \geq \|\Lambda\|_{\text{sum}} + \text{trace}(M_{11}) \\ & + \max \left(\sum_{i=1}^c \min_j [M_{22}]_{ij}, \sum_{j=1}^c \min_i [M_{22}]_{ij} \right) \quad (12) \end{aligned}$$

The bound in (12) is tight and becomes exact, if $\min_j [M_{22}]_{ij}$ occurs for different j for every $i \in N_c$ or $\min_i [M_{22}]_{ij}$ occurs for different i for every $j \in N_c$.

Proposition 4 Consider a node $S = (F_f, C_c)$. Let

$$E = G_{P_n}(I \circ G_{P_n})^{-1} - I = \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix} \quad (13)$$

where $E_{11} \in \mathbb{R}^{f \times f}$ and $E_{22} \in \mathbb{R}^{c \times c}$. Then

$$\min_{P_n \in \mathcal{S}} \bar{\mu}_\Delta(E) \geq \bar{\mu}_{\Delta_1}(E_{11}) \geq \max_{D \in \mathcal{D}} \rho(E_{11}D) \geq \rho(E_{11}) \quad (14)$$

where $\Delta = \text{diag}(\Delta_1, \Delta_2)$, ρ is the spectral radius and \mathcal{D} is the set of all diagonal matrices with elements ± 1 .

Although $\bar{\mu}_{\Delta_1}(E_{11})$ is tighter than other lower bounds presented in (14), its computation is costly. As a BAB method spends most of its time in evaluating non-optimal nodes, the use of computationally cheaper albeit weaker lower bound $\rho(E_{11})$ provides a more efficient BAB method. We further note that for pruning purposes, computing lower bound is not necessary and it suffices to establish whether expansion of a node can lead to the optimal solution. With this insight, we derive computationally cheaper pruning conditions next.

Lemma 5 For a matrix $A \in \mathbb{R}^{n \times n}$ and scalar $B > 0$,

$$\det(BI - A) \leq 0 \Rightarrow \rho(A) \geq B \quad (15)$$

$$\det(BI + A) \leq 0 \Rightarrow \rho(A) \geq B \quad (16)$$

Proposition 6 Let S^i , $i = 1, 2, \dots, c$, be the sub-nodes of the node $S = (F_f, C_c)$ and

$$E_{11}^i = \begin{bmatrix} E_{11} & E_{12}^i \\ E_{21} & 0 \end{bmatrix} \quad (17)$$

where $E_{11} \in \mathbb{R}^{f \times f}$, $E_{12}^i \in \mathbb{R}^{f \times 1}$ and $E_{21} \in \mathbb{R}^{1 \times f}$ are defined based on the definition of E in (8). For $B > 0$,

$$|E_{21}(BI - E_{11})^{-1}E_{12}^i| \geq B \Rightarrow \bar{\mu}_{\Delta_1}(E_{11}^i) \geq B \quad (18)$$

$$|E_{21}(BI + E_{11})^{-1}E_{12}^i| \geq B \Rightarrow \bar{\mu}_{\Delta_1}(E_{11}^i) \geq B \quad (19)$$

The main computational load in using (18) and (19) is evaluation of $(BI - E_{11})^{-1}$ and $(BI + E_{11})^{-1}$, respectively, which needs to be carried out only once for all c sub-nodes. Thus, in comparison with computation of $\rho(E_{11})$, the use of these conditions is more efficient, especially when c is large. These conditions, however, are derived considering the μ -IM as the only selection criterion. The next proposition allows embedding of these pruning conditions into the biobjective BAB algorithm.

Proposition 7 Let the vectors B_Λ and B_μ contain the RGA-number and μ -IM values of the current Pareto-set \mathcal{P} , respectively. For the node $S = (F_f, C_c)$, let b_Λ be the lower bound of the RGA-number calculated using (12). For the subset of \mathcal{P} defined as $\tilde{\mathcal{P}} = \{P_n \in \mathcal{P} | B_\Lambda(P_n) \leq b_\Lambda\}$, let $B = \max_{P_n \in \tilde{\mathcal{P}}} B_\mu(P_n)$. Then, if $\bar{\mu}_{\Delta}(E_{11}) \geq B$, all n -element solutions of S are dominated by at least one member of \mathcal{P} and hence can be pruned.

Proposition 7 implies that pruning can be carried out using (18) and (19) with $B = \max_{P_n \in \tilde{\mathcal{P}}} B_\mu(P_n)$. The constraints in (10) provide additional conditions for pruning nodes. These constraints only need to be checked for the element being moved from C_c to F_f during node expansion using the branching rule given in Definition 1.

4 Numerical Tests

The efficiency of the BAB method is demonstrated through numerical studies, which are carried out on a notebook with Intel® Core™ Duo Processor T2400 (1.83 GHz, 2MB RAM) using MATLAB® 2007b [7].

4.1 Random matrices

The efficiency of the BAB approach is first examined using 1000 $n \times n$ random matrices for every n between 3 and 15. Elements of these matrices are normally distributed with zero mean and unit variance. The average computation times and number of node evaluations required by BAB algorithm are shown in Figure 2. For comparison, the computational time of a brute force search is estimated by multiplying $n!$ with the time required for evaluating the RGA-number and μ -IM of an $n \times n$ matrix. Figure 2 shows that the BAB approach can easily handle matrix sizes as large as $n = 15$ within 3 minutes, whilst the brute-force search can only deal with n up to 8 within the same time. For $n = 15$, the average number of nodes evaluated and average solution time required by the BAB approach are 7 orders of magnitude lower than brute-force search showing computational efficiency.

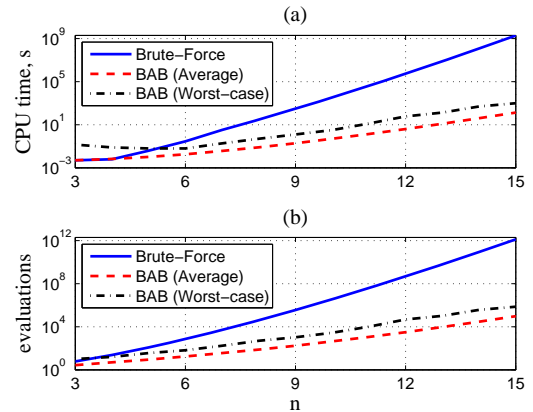


Fig. 2. Random tests for pairing $n \times n$ systems: (a) computation time and; (b) number of nodes evaluated

For $3 \leq n \leq 15$, the following empirical relationship estimates the variation of average computation time (t) of the BAB method reasonably well

$$\ln(t) = -6.5 + 0.2n + 0.037n^2 \quad (20)$$

The exponential complexity indicates that the BAB algorithm is not able to overcome the NP-hard nature of the problem. For the randomly generated matrices, however, the worst-case computation times and number of node evaluations differ only by factors of 5 – 15 from their average values; see Figure 2. This observation highlights that the proposed BAB method can be used for efficiently solving most problems of practical interest. Furthermore, due to the sparsity of the models of industrial

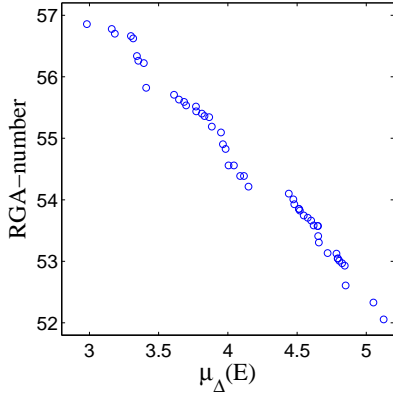


Fig. 3. Pareto-optimal set for CD control

processes, the efficiency attained by the BAB method is significantly better than seen for random matrices, as demonstrated next.

4.2 Cross-direction control problem

CD control is important in paper manufacturing to maintain paper quality, and to reduce raw material and energy consumptions [24,35]. The large dimension and presence of strong interactions makes the selection of the best pairings difficult for decentralized CD control.

The gain matrix of the CD response model is a sparse “banded symmetric” matrix parameterized by p_1, p_2, \dots, p_r , where r is the number of interaction levels [24]. Among several models summarized in [24], the case with $r = 7$ and $n = 20$ is considered in this work, where p_1, p_2, \dots, p_7 are 1, 1.3, 0.8, -0.6 , -0.3 , 0 and -0.1 , respectively. As $\bar{\mu}_\Delta(E)$ is infinite, when pairings are selected on a zero element of G , this process has $6^{19} \approx 6.09 \times 10^{14}$ valid pairing alternatives. Evaluation of the RGA-number and the μ -IM for a 20×20 system takes about 2 ms on the aforementioned notebook. Thus, brute force search would require approximately 4.3×10^4 years to solve this problem. In comparison, the BAB algorithm finds the Pareto-optimal set containing 55 pairing alternatives, shown in Figure 3, in about 5.5 minutes through evaluation of 2.83×10^5 nodes. Note that due to the non-convexity, not all Pareto-optimal pairings can be found by a weighted sum approach.

5 Conclusions

Closed-loop performance of decentralized controllers greatly relies on the selected pairings. Efficient methods are needed to address the combinatorial difficulty and conflicting nature of different selection criteria. It is the first time that a general multiobjective formulation of the pairing problem is proposed. This paper also appears to be the first to present a BAB framework to

directly handle the multiobjective nature of permutation problems such as pairing selection. The proposed BAB method can handle most of the available pairing selection criteria simultaneously. For the biobjective problem with the RGA-number and the μ -IM as the selection criteria, a number of lower bounds and efficient pruning conditions have been derived. Numerical tests show that a reduction of several orders of magnitude in solution time over brute-force search is achieved by the proposed BAB algorithm. The case study of CD control indicates that the BAB approach is practically applicable to large-scale industrial problems. Future work will focus on derivation of tighter lower bounds and alternate pruning strategies in order to improve the efficiency further. In addition, the development of an evolutionary algorithm and comparison of its efficiency with the proposed BAB algorithm will be pursued.

Acknowledgements

This work was partially supported by Nanyang Technological University, Singapore through grants nos. M581200012 and RG42/06, and partially by the Royal Society, UK through grant no. IV0871568.

Appendix: Proofs

Proof of Proposition 2. For $S = (F_f, C_c)$, let (F_f, c_i) and (F_f, c_j) be the fixed sets of S^i and S^j , respectively. Since $c_i \neq c_j$, $\mathcal{S}(S^i) \cap \mathcal{S}(S^j) = \emptyset$ for \mathcal{S} in (4). Thus, every terminal node belongs to only one branch. Now, the number of terminal nodes of S is the same as the sum of the number of terminal nodes of all its sub-nodes. As all terminal nodes are distinct, every terminal node must belong to at least one branch. \square

Proof of Proposition 3. Note that $\|\Lambda - I\|_{\text{sum}} = \|\Lambda\|_{\text{sum}} + \text{trace}(M_{11}) + \text{trace}(M_{22})$, where $\|\Lambda\|_{\text{sum}}$ and $\text{trace}(M_{11})$ are the same $\forall P_n \in \mathcal{S}$. The result follows as $\forall P_n \in \mathcal{S}$, $\text{trace}(M_{22}) \geq \sum_{i=1}^c \min_j [M_{22}]_{ij}$, and $\text{trace}(M_{22}) \geq \sum_{j=1}^c \min_i [M_{22}]_{ij}$. \square

Proof of Proposition 4. For any $P_n \in \mathcal{S}$, $\bar{\mu}_\Delta(E) \geq \bar{\mu}_{\Delta_1}(E_{11})$ and $\bar{\mu}_{\Delta_1}(E_{11}) = \max_{U \in \mathcal{U}} \rho(E_{11}U)$, where \mathcal{U} is the set of all diagonal unitary matrices [31]. Now, the result follows as $\mathcal{D} \subset \mathcal{U}$ and $I \in \mathcal{D}$. \square

Proof of Lemma 5. Note that $\det(BI - A) = \prod_{i=1}^n \lambda_i(BI - A)$ and $\lambda_i(BI - A) = B - \lambda_i(A)$, where $\lambda_i(\cdot)$ denote the eigenvalue. If $\det(BI - A) \leq 0$, $\lambda_i(BI - A) < 0$ and thus $\lambda_i(A) > B$ for some $i \in N_n$ implying (15). Similarly, when $\det(BI + A) \leq 0$, for some $i \in N_n$, $\lambda_i(BI + A) = B + \lambda_i(A) < 0$ and thus $-\lambda_i(A) > B$ or $|\lambda_i(A)| > B$ implying (16). \square

Proof of Proposition 6. Based on (14), $\bar{\mu}_{\Delta_1}(E_{11}^i) \geq B$, if $\exists D \in \mathcal{D}$ such that $\rho(E_{11}^i D) > B$. Let $D =$

$\text{diag}(I_f, d_j)$, where $d_j = \pm 1$. Now, $\det(BI - E_{11}^i D) = \det(BI - E_{11})(B - \beta_i d_j)$ [16], where $\beta_i = E_{21}(BI - E_{11})^{-1} E_{12}^i$. Therefore, when $\max_{d_j} \beta_i d_j = |\beta_i| \geq B$ either $\det(BI - E_{11}^i) \leq 0$ or $\det(BI - E_{11}) \leq 0$. Based on Lemma 5, in either case, $\bar{\mu}_{\Delta_1}(E_{11}) \geq B$, which proves (18). The proof of (19) is similar and is omitted. \square

Proof of Proposition 7. For $P_n^* \in \mathcal{P}$, let $B = B_\mu(P_n^*)$. Since $b_\Lambda \geq B_\Lambda(P_n^*)$, if $\mu_\Delta(E_{11}) \geq B$, S and its sub-nodes are dominated by P_n^* , and hence can be pruned. \square

References

- [1] R. Agustriyanto and J. Zhang. Control structure selection for the ALSTOM gasifier benchmark process using GRDG analysis. *Intl. J. of Modelling, Identification and Control*, 6:126–135, 2009.
- [2] R. D. Braatz. *Robust loopshaping for Process Control*. Ph.D. thesis, California School of Technology, Pasadena, CA, 1993.
- [3] E. H. Bristol. On a new measure of interaction for multivariable process control. *IEEE Trans. Automat. Contr.*, 11, 1966.
- [4] M.J. Brusco and S. Stahl. *Branch-and-bound applications in combinatorial data analysis*. Springer Verlag, 2005.
- [5] Y. Cai and T. E. Marlin. Control structure design to achieve multiple performance criteria. In *Proc. 7th Intl. Symposium on DYCOPS*, Boston, MA, 2004.
- [6] Y. Cao and V. Kariwala. Bidirectional branch and bound for controlled variable selection Part I. Principles and minimum singular value criterion. *Comput. Chem. Eng.*, 32(10):2306–2319, 2008.
- [7] Y. Cao and V. Kariwala. B3muRGA. MATLAB File Exchange, February 2010. Available at <http://www.mathworks.com/matlabcentral/fileexchange/26548>.
- [8] Y. Cao and P. Saha. Improved branch and bound method for control structure screening. *Chem. Engg. Sci.*, 60(6):1555–1564, 2005.
- [9] Y. Cao and Z. Yang. Multiobjective process controllability analysis. *Comput. Chem. Engng.*, 28:93–90, 2004.
- [10] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, 2001.
- [11] M. Ehrgott and X. Gandibleux. A survey and annotated bibliography of multiobjective combinatorial optimization. *OR Spectrum*, 22(4):425–460, 2000.
- [12] P. J. Fleming and R. C. Purshouse. Evolutionary algorithms in control systems engineering: A survey. *Control Engineering Practice*, 10(11):1223–1241, 2002.
- [13] I. G. French, C. S. Cox, and C. K. S. Ho. Genetic algorithms in model structure and controller structure identification. *Proc. Instn Mech Engrs, Part I*, 211:333–343, 1997.
- [14] P. Grosdidier and M. Morari. Interaction measures for systems under decentralized control. *Automatica*, 22(3):309–319, 1986.
- [15] G. Gutin and A. P. Punnen. *The Traveling Salesman Problem and Its Variations*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.
- [16] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, UK, 1985.
- [17] M. Hovd, D.L. Ma, and R.D. Braatz. On the Computation of Disturbance Rejection Measures. *Ind. Eng. Chem. Res.*, 42(10):2183–2188, 2003.
- [18] J. B. Jørgensen and S. B. Jørgensen. Towards automatic decentralized control structure selection. *Comput. Chem. Engng.*, 24:841–846, 2000.
- [19] V. Kariwala and Y. Cao. Efficient branch and bound methods for pairing selection. In *Proc. 17th IFAC World Congress*, Seoul, Korea, 2008.
- [20] V. Kariwala and Y. Cao. Bidirectional branch and bound for controlled variable selection: Part II. Exact local method for self-optimizing control. *Comput. Chem. Eng.*, 33(8):1402–1412, 2009.
- [21] V. Kariwala and Y. Cao. Bidirectional branch and bound for controlled variable selection: Part III. Local average loss minimization. *IEEE Trans. Ind. Informat.*, In Press, 2010.
- [22] V. Kariwala, J. F. Forbes, and E. S. Meadows. Integrity of systems under decentralized integral control. *Automatica*, 41(9):1575–1581, 2005.
- [23] I. K. Kookos and J. D. Perkins. Heuristic-based mathematical programming framework for control structure selection. *Ind. Eng. Chem. Res.*, 40:2079–2088, 2001.
- [24] D. L. Laughlin, M. Morari, and R. D. Braatz. Robust performance of cross-directional basis-weight control in paper machines. *Automatica*, 29:1395–1410, 1993.
- [25] D. Lei. Multi-objective production scheduling: A survey. *Int. J. Adv. Manuf. Technol.*, 43:926–938, 2009.
- [26] M. L. Luyben and C. A. Floudas. Analyzing the interaction of design and control - 1. A multiobjective framework and application to binary distillation column. *Comput. Chem. Engng.*, 18(10):933–969, 1994.
- [27] R. T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, 2004.
- [28] G. Mavrotas and D. Diakoulaki. A branch and bound algorithm for mixed zero-one multiple objective linear programming. *Eur. J. Oper. Res.*, 107:530–541, 1998.
- [29] J. Rakowska, R.T. Haftka, and L.T. Watson. Tracing the efficient curve for multi-objective control-structure optimization. Technical Report TR-91-02, Virginia Polytechnic Institute & State University, Blacksburg, VA, USA, 1994.
- [30] M. P. Sanchez and J. A. Almansa. A real application example of a control structure selection by means of a multiobjective genetic algorithm. In J. Mira and J. R. Alvarez, editors, *Artificial Neural Nets Problem Solving Methods*, pages 369–376. Springer-Verlag, Heidelberg, Germany, 2003.
- [31] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control: Analysis and Design*. John Wiley & sons, Chichester, UK, 2nd edition, 2005.
- [32] F. Sourd, O. Spanjaard, and P. Perny. Multi-objective branch-and-bound: Application to the bi-objective spanning tree problem. In *Proc. 7th Int. Conf. MOPGP*, Loire Valley, France, 2006.
- [33] M. Van de wal. Control structure design for dynamic systems: A review. Technical Report WFW-94-084, Eindhoven University of Technology, Eindhoven, Netherlands, 1994.
- [34] J. G. VanAntwerp and R. D. Braatz. A tutorial on linear and bilinear matrix inequalities. *Journal of Process Control*, 10(4):363–385, 2000.
- [35] J. G. VanAntwerp, A. P. Featherstone, R. D. Braatz, and B.A. Ogunnaike. Cross-directional control of sheet and film processes. *Automatica*, 43(2):191–211, 2007.