Computers & Chemical Engineering, Volume 33, Issue 8, 2009, Pages 1402-1412

Bidirectional Branch and Bound for Controlled Variable Selection Part II. Exact Local Method for Self-optimizing Control

Vinay Kariwala[†] and Yi Cao[‡*]

[†] Division of Chemical & Biomolecular Engineering,
Nanyang Technological University, Singapore 637459
[‡]School of Engineering, Cranfield University, Cranfield, Bedford MK43 0AL, UK

This version: January 28, 2009

Abstract

The selection of controlled variables (CVs) from available measurements through enumeration of all possible alternatives is computationally forbidding for large-dimensional problems. In Part I of this work [5], we proposed a bidirectional branch and bound (BAB) approach for subset selection problems and demonstrated its efficiency using the minimum singular value criterion. In this paper, the BAB approach is extended for CV selection using the exact local method for self-optimizing control. By redefining the loss expression, we show that the CV selection criterion for exact local method is bidirectionally monotonic. A number of novel determinant based criteria are proposed for fast pruning and branching purposes resulting in a computationally inexpensive BAB approach. We also establish a link between the problems of selecting a subset and combinations of measurements as CVs and present a partially bidirectional BAB method for selection of measurements, whose combinations can be used as CVs. Numerical tests using randomly generated matrices and binary distillation column case study demonstrate the computational efficiency of the proposed methods.

Keywords: Branch and bound, Control structure design, Controlled variables, Combinatorial optimization, Self-optimizing control.

*Corresponding Author: Tel: +44-1234-750111; Fax: +44-1234-754685; E-mail:y.cao@cranfield.ac.uk

Nomenclature

$\mathbf{1}_{p,q}$	$p \times q$ matrix of ones
$\mathbf{a}$	column vector (lower case bold face letter)
$\mathbf{A}$	matrix (upper case bold face letter)
B	best available lower bound on selection criterion
C	candidate set of a node
$\mathcal{C}_m^n$	binomial coefficient of m choose n
F	fixed set of a node
$\tilde{\mathbf{G}}$	Defined as $\tilde{\mathbf{G}} = \mathbf{G}^y \mathbf{J}_{uu}^{-1}$; see equation (22)
$\tilde{\mathbf{G}}_X$	sub-matrix of $\tilde{\mathbf{G}}$ consisting of rows with indices in set X
$\mathbf{H}$	measurement selection or combination matrix
$\mathbf{I}_p$	$p \times p$ Identity matrix
J	objective functional related to steady-state economics of process
L_1	local loss when individual measurements are used as CVs
L_2	local loss when measurement combinations are used as CVs
$\underline{L}_2(X)$	lower bound on L_2 for all n-element supersets of X
$\mathbf{M}(X_p)$	Defined as $\mathbf{M}(X_p) = \mathbf{R}^{-T} \tilde{\mathbf{G}}_{X_p} \tilde{\mathbf{G}}_{X_p}^T \mathbf{R}^{-1}$; see Equation 31
n	number of measurements to be selected, whose combinations are used as CVs
n_u	number of degrees of freedom or inputs
n_y	number of available measurements
$\mathbf{N}(X_p)$	Defined as $\mathbf{N}(X_p) = \tilde{\mathbf{G}}_{X_p}^T (\mathbf{Y}_{X_p} \mathbf{Y}_{X_p}^T)^{-1} \tilde{\mathbf{G}}_{X_p}$; see Equation 32
$\mathbf{R}$	Cholesky factor
S	union of the sets F and C, *i.e.* $S = F \cup C$
$\mathcal{S}$	a two-tuple, $\mathcal{S} = (F, C)$ represents a node in the search tree
T	selection criterion (to be minimized)
$\underline{T}_n(X)$	lower bound on T for all n-element subsets or supersets of X
X_t	subscript t represents the size of the set X
X^i	superscript i represents the index of the sub or super set obtained from X
$\mathbf{Y}$	Defined as $\mathbf{Y} = \begin{bmatrix} (\mathbf{G}^y \mathbf{J}_{uu}^{-1} \mathbf{J}_{ud} - \mathbf{G}_d^y) \mathbf{W}_d & \mathbf{W}_e \end{bmatrix}$; see equation (21)
α	downwards pruning index
β	upwards pruning index

λ_i	i^{th} largest eigenvalue of a square matrix
$\bar{\lambda}$	maximum eigenvalue of a square matrix
$\underline{\lambda}$	least non-zero eigenvalue of a square matrix
$\bar{\sigma}$	maximum singular value of a matrix

1 Introduction

A plant usually has many measurements available for monitoring and control purposes. Self-optimizing control involves selection of a subset or combinations of available measurements as controlled variables (CVs) such that when the selected CVs are maintained at constant setpoints using feedback controllers, the overall plant operation is nearly optimal even in the presence of various disturbances [16]. Thus, the concept of self-optimizing control provides a simple operational strategy, where the loss incurred due to the use of suboptimal feedback based strategy in comparison with the use of an online optimizer is minimal.

The loss incurred by the feedback based strategy depends on the selected CVs. For appropriate selection of CVs using the concept of self-optimizing control, various criteria have been proposed including the minimum singular value (MSV) rule [17] and exact local methods with worst-case [9, 12] and average loss minimization [3, 13]. Like other control structure selection problems, CV selection is a combinatorial optimization problem; see *e.g.* [19]. To find the optimal CVs, the selection criteria need to be evaluated for each possible alternative resulting in huge computational requirements, especially when the number of available measurements and the number of CVs to be selected are large. For such large scale problems, some heuristic rules may have to be applied to reduce the size of the search space. With the use of heuristic rules, however, the global optimality of selected CVs cannot be guaranteed.

The combinatorial difficulty associated with the CV selection problem was recently addressed in Part I of this work [5], where a novel bidirectional branch and bound (BAB) approach was proposed and its efficiency for CV selection was demonstrated using the MSV rule. The MSV rule, however, is approximate and can sometimes lead to non-optimal set of CVs [10]. In general, it is more appropriate to select CVs using the exact local methods [3, 9, 13]. The objective of this paper is to extend the BAB approach for CV selection using the exact local method with worst-case loss minimization.

The selection of CVs from available measurements can be seen as a subset selection problem, where the number of CVs to be selected is the same as the number of available inputs or degrees of freedom. For such problems, a bidirectional BAB method gains its efficiency by pruning both supersets and subsets

(measurement sets with the number of elements greater than and smaller than the number of inputs, respectively), which cannot lead to the optimal solution. A difficulty in the use of BAB method for the exact local method is that the loss expression for this method is restrictively defined for square systems, *i.e.* where the number of selected measurements is equal to the number of inputs. On the other hand, a BAB method requires evaluation of the selection criterion, when the number of selected variables differs from the target subset size. In this paper, we re-define the loss expression for exact local method such that it holds for non-square configurations. We subsequently show that the re-defined loss expression is bidirectionally monotonic and thus is amenable to the use of bidirectional BAB approach.

In comparison with the traditional unidirectional BAB approaches, the use of bidirectional pruning, *i.e.* simultaneous pruning of both supersets and subsets, provides significant improvement in computational efficiency. The evaluation of (re-defined) loss expression for exact local method, however, is computationally expensive. We note that a BAB method spends most of its time in evaluation of non-optimal nodes. Therefore, we develop several efficient determinant based conditions to replace the computationally demanding calculation of exact local loss so as to quickly decide upon whether expansion of a node can lead to the optimal solution. With these improvements, the proposed BAB method achieves similar computational efficiencies as the BAB approach for CV selection using MSV rule [5].

A related problem involves selection of combinations of available measurements as CVs, which provides lower losses than the use of a subset of available measurements as CVs [2, 9, 12, 13]. Halvorsen et al. [9] proposed the use of nonlinear optimization based approach to design the combination matrix, which is computationally expensive and may converge to local optima. Alstad and Skogestad [2] proposed the use of computationally more efficient null space method to find measurement combinations, but this method ignores implementation error and thus can only provide a suboptimal solution. Recently, explicit solutions to the problem of finding locally optimal measurement combinations have been proposed [3, 12, 13], which significantly simplify the design procedure. It is noted in [1, 10, 12, 13] that the use of combinations of a few measurements as CVs often provides similar loss as compared to the case where combinations of all available measurements are used. Though the former approach results in control structures with lower complexity, it gives rise to another combinatorial optimization problem involving the identification of the set of measurements, whose combinations can be used as CVs.

In this paper, we extend the BAB method to find a subset of available measurements, whose combinations can be used CVs. Unlike the selection of a subset of measurements as CVs, however, the selection criterion for this problem is only downwards monotonic (gradually decreasing subset size). We show that the advantages of bidirectional BAB method can still be realized to some extent, as a lower bound of the

selection criterion satisfies upwards monotonicity, when the number of selected measurements is greater than a certain number. We propose partially bidirectional BAB method for this problem and demonstrate the efficiency using the case study of a binary distillation column [15]. In addition to the extension of the bidirectional BAB for CV selection using exact local method, a contribution of this work is the demonstration of the fact that BAB methods can still be used when the selection criterion does not satisfy monotonicity requirements.

The rest of the paper is organized as follows: Section 2 provides a tutorial overview of unidirectional and bidirectional BAB methods for subset selection problems. The problems of selecting a subset or combinations of available measurements as CVs using the concept of self-optimizing control are formulated in Section 3. For these problems, efficient bidirectional BAB algorithms are developed in Section 4. The developed algorithms are tested with several numerical examples in Section 5 and the work is concluded in Section 6.

2 Branch and bound methods for subset selection

This section gives a brief overview of the principles of unidirectional (upward or downward) and bidirectional BAB approaches for subset selection problems; see [5] for further details. The bidirectional BAB approach is adapted for CV selection using the exact local method for self-optimizing control later in the paper.

2.1 Subset selection problem

Assume that X_m is an m-element set of all the available elements. The subset selection problem involves finding an n-element subset $X_n \subset X_m$ such that the selection criterion T is minimized among all possible $X_n \subset X_m$, *i.e.*

$$T(X_n^{\text{opt}}) = \min T(X_n) \quad \forall X_n \subset X_m \tag{1}$$

which is a combinatorial optimization problem. For small m and n, the globally optimal subset X_n^{opt} can be obtained through an exhaustive search. For large m and n, however, the number of available alternatives $\mathcal{C}_m^n = m!/(m-n)!n!$ can be too large to carry out a brute-force search. BAB is one of the efficient approaches, which are able to find the globally optimal subset without exhaustive evaluation.

2.2 Branch and bound approaches

The first BAB approach for subset selection problems was proposed by Narendra and Fukunaga [14], which was further improved in [8, 18, 20]. The BAB method used in this paper differs from these approaches, as it uses the concepts of fixed and candidate sets introduced in [6, 7] to facilitate the implementation of bidirectional pruning and branching. The basic principle, however, remains the same, as discussed next.

Principle. The basic principle of BAB approach is to divide the original selection problem into smaller sub-problems (branching). Then, if an estimated lower bound of T of a sub-problem is larger than an upper bound of $T(X_n^{\rm opt})$, then the sub-problem under consideration cannot lead to the optimal solution and hence can be discarded without further evaluation (pruning). If a sub-problem cannot be discarded, it is further divided into smaller sub-problems. This procedure is repeated until there are no more sub-problems left to solve.

Fixed and candidate sets. To standardize notation, consider a sub-problem $\mathcal{S} = (F_f, C_c)$ with an f-element fixed set F_f and a c-element candidate set C_c, where $f \leq n$ and $n \leq f + c \leq m$. Here, the elements of F_f are included in all n-element subsets that can be obtained by solving $\mathcal{S}$, while elements of C_c can be freely chosen to append F_f. Then, a subset X_n belonging to $\mathcal{S}$ must satisfy the following two relationships:

$$\text{upwards relationship:} \quad F_f \subset X_n \tag{2}$$

$$\text{downwards relationship:} \quad X_n \subset (F_f \cup C_c) \tag{3}$$

Furthermore, $\mathcal{S} = (F_f, C_c)$ can be divided into $2c$ subproblems either by moving $x_j \in C_c$ to F_f or by discarding $x_k \in C_c$, where $j, k = 1, 2, \cdots, c$. Each of the sub-problems $\mathcal{S}^i = (F_{f_i}^i, C_{c_i}^i)$, $i = 1, 2, \cdots, 2c$, satisfy

$$\text{upwards fixed-set relationship:} \quad F_f \subseteq F_{f_i}^i \tag{4}$$

$$\text{downwards candidate-set relationship:} \quad C_c \supseteq C_{c_i}^i \tag{5}$$

$$\text{downwards union relationship:} \quad (F_f \cup C_c) \supseteq (F_{f_i}^i \cup C_{c_i}^i) \tag{6}$$

Bidirectional pruning using monotonicity. Let $\underline{T}(\mathcal{S})$ be a lower bound of T over all n-element subsets that can be reached from $\mathcal{S}$, *i.e.*

$$\underline{T}(\mathcal{S}) \leq \min_{\substack{X_n \supset F_f \\ X_n \subset (F_f \cup C_c)}} T(X_n) \tag{7}$$

Further, let B be an upper bound of $T(X_n^{\text{opt}})$, *i.e.* $B \geq T(X_n^{\text{opt}})$. Then, $\mathcal{S}$ can be discarded, if $\underline{T}(\mathcal{S}) > B$.

The computation of $\underline{T}(\mathcal{S})$ can be considerably simplified, if the selection criterion is monotonic. Here, the selection criterion T is said to be upwards monotonic, when

$$T(X_s) \geq T(X_t) \quad \text{if}\, X_s \supset X_t;\, t < s < n \tag{8}$$

Similarly, T is said to be downwards monotonic, when

$$T(X_s) \geq T(X_t) \quad \text{if}\, X_s \subset X_t;\, t > s > n \tag{9}$$

For upwards monotonic T, the lower bound of T on $\mathcal{S} = (F_f, C_c)$ can be estimated as

$$\underline{T}(\mathcal{S}) = T(F_f) \tag{10}$$

In this case, an upward pruning operation to discard $\mathcal{S}$ can be conducted if $T(F_f) > B$. Similarly, for downwards monotonic T, the lower bound of T on $\mathcal{S} = (F_f, C_c)$ can be estimated as

$$\underline{T}(\mathcal{S}) = T(F_f \cup C_c) \tag{11}$$

A downward pruning operation to discard $\mathcal{S}$ can be carried out if $T(F_f \cup C_c) > B$. Furthermore, it is shown in [5] that if T satisfies both upward (for subset size less than n) and downward (for subset size larger than n) monotonicity, then pruning can be carried out bidirectionally so that efficiency can be significantly improved.

In bidirectional BAB approach [5], pruning is carried out on the $2c$ sub-problems of $\mathcal{S}$, instead of on $\mathcal{S}$ directly. Assume that $T(F_f) < B$ and $T(F_f \cup C_c) < B$. For $x_i \in C_c$, upward pruning is conducted by discarding x_i from C_c, if $T(F_f \cup x_i) > B$. Similarly, if $T(F_f \cup (C_c \backslash x_i)) > B$, then downward pruning is performed by moving x_i from the candidate set to the fixed set. Finally, if both conditions are satisfied, then bidirectional pruning discards all the $2c$ sub-problems and thus entire $\mathcal{S}$. Here, an advantage of performing pruning on sub-problems is that the bounds $T(F_f \cup x_i)$ and $T(F_f \cup (C_c \backslash x_i))$ can be computed from $T(F_f)$ and $T(F_f \cup C_c)$, respectively, for all $x_i \in C_c$ together resulting in computational efficiency.

2.3 Bidirectional branching

A BAB approach also gains its efficiency by an effective branching rule, *i.e.* the way in which a problem is divided into several subproblems. The aim of an effective branching rule is to facilitate pruning of as many non-optimal subproblems as possible. Based on the bidirectional BAB principle, a bidirectional branching rule has been proposed in [5]. In this approach, when branching is required for $\mathcal{S} = (F_f, C_c)$, instead of branching all $2c$ subproblems of $\mathcal{S}$, only two branches are produced based on a decision element, $x_k \in C_c$; see Figure 3 for an example. Here, an upward branch corresponds to moving x_k from the candidate set C_c to the fixed set F_f and a downward branch corresponds to discarding x_k from the candidate set C_c. Between these two branches, the branch with fewer n-element subsets (terminal nodes) is evaluated first so that the branch with more alternatives might be discarded at a later stage.

For a given problem $\mathcal{S} = (F_f, C_c)$, the upward and downward branches have $\mathcal{C}_{c-1}^{n-f-1}$ and $\mathcal{C}_{c-1}^{n-f}$ terminal nodes, respectively. Thus, upward-first branching is conducted if $\mathcal{C}_{c-1}^{n-f-1} \leq \mathcal{C}_{c-1}^{n-f}$ or $2(n-f) \leq c-1$ and downward-first branching otherwise. The decision element itself is chosen on a best-first basis. More specifically, for upwards-first branching, the decision element x_k is chosen to provide the minimum $T(F_f \cup x_k)$ (best upward branch evaluated first) or the maximum $T(F_f \cup (C_c \backslash x_k))$ (worst downward branch kept for future pruning) among all $x_k \in C_c$, whilst for downwards-first branching, x_k is selected to give the maximum $T(F_f \cup x_k)$ (best downward branch evaluated first) or the minimum $T(F_f \cup (C_c \backslash x_k))$ (worst upward branch kept for future pruning) among all $x_k \in C_c$.

3 Exact Local Method for Self-optimizing Control

In this section, we introduce the exact local method for self-optimizing control. We also represent the problems of selecting measurements, which can be either directly used or combined as CVs, as subset selection problems.

3.1 Self-optimizing Control

The economically optimal operation of a process requires the use of an online optimizer to update the operating point according to the changes in disturbances $\mathbf{d} \in \mathbb{R}^{n_d}$. A simpler strategy is to update the degrees of freedom or inputs $\mathbf{u} \in \mathbb{R}^{n_u}$ indirectly using feedback controllers such that some CVs are held constant. The use of this simpler strategy is clearly sub-optimal. Self-optimizing control is said to occur,

when an acceptable loss is achieved by the feedback based operational strategy without the need to re-optimize when disturbances occur [16]. Based on this concept, the appropriate CVs can be selected by comparing the losses for different alternatives.

3.2 Local method

CV selection based on the general non-linear formulation of self-optimizing control can be time-consuming and local methods are often used for pre-screening alternatives. To present the local methods, let the economics of the plant be characterized by the scalar objective function $J(\mathbf{u}, \mathbf{d})$ and $\mathbf{u}^{\text{opt}}(\mathbf{d}^*)$ be the optimal value of inputs minimizing J for the nominal disturbance $\mathbf{d}^*$. Around the nominally optimal operating point $(\mathbf{u}^{\text{opt}}(\mathbf{d}^*), \mathbf{d}^*)$, let the linearized model of the process be

$$\mathbf{y} = \mathbf{G}^y \, \mathbf{u} + \mathbf{G}_d^y \, \mathbf{W}_d \, \mathbf{d} + \mathbf{W}_e \, \mathbf{e} \tag{12}$$

where $\mathbf{y} \in \mathbb{R}^{n_y}$ denotes the process measurements and $\mathbf{e} \in \mathbb{R}^{n_y}$ denotes the implementation error, which results due to measurement and control error. Here, the diagonal matrices $\mathbf{W}_d$ and $\mathbf{W}_e$ contain the magnitudes of expected disturbances and implementation errors associated with the individual measurements, respectively. The CVs $\mathbf{c} \in \mathbb{R}^{n_u}$ are given as

$$\mathbf{c} = \mathbf{H} \, \mathbf{y} = \mathbf{G} \, \mathbf{u} + \mathbf{G}_d \, \mathbf{W}_d \, \mathbf{d} + \mathbf{H} \, \mathbf{W}_e \, \mathbf{e} \tag{13}$$

where

$$\mathbf{G} = \mathbf{H} \, \mathbf{G}^y \quad \text{and} \quad \mathbf{G}_d = \mathbf{H} \, \mathbf{G}_d^y \tag{14}$$

It is assumed that $\mathbf{G} \in \mathbb{R}^{n_u \times n_u}$ is invertible. This assumption is necessary for integral control.

Assume that the feedback controller maintains $\mathbf{c}$ at $\mathbf{c}^*$ and let $\mathbf{u}^{\text{opt}}(\mathbf{d})$ denote the optimal value of $\mathbf{u}$ for any allowable disturbance. Then for given $\mathbf{d}$ and $\mathbf{e}$, the loss incurred due to controlling CVs at constant set-point is defined as

$$L(\mathbf{H}, \mathbf{d}, \mathbf{e}) = J(\mathbf{u}, \mathbf{d}, \mathbf{e})|_{\mathbf{c}=\mathbf{c}^*} - J(\mathbf{u}^{\text{opt}}(\mathbf{d}), \mathbf{d}) \tag{15}$$

When $\mathbf{d}$ and $\mathbf{e}$ are constrained to satisfy

$$\left\| \begin{bmatrix} \mathbf{d}^T & \mathbf{e}^T \end{bmatrix} \right\|_2^T \leq 1 \tag{16}$$

Halvorsen et al. [9] have shown that the worst-case loss over the set (16) is given as

$$L_1(\mathbf{H}) = \frac{1}{2}\bar{\sigma}^2 \left(\begin{bmatrix} \mathbf{M}_d & \mathbf{M}_e \end{bmatrix} \right) \tag{17}$$

where

$$\mathbf{M}_d = \mathbf{J}_{uu}^{1/2} \left(\mathbf{J}_{uu}^{-1} \mathbf{J}_{ud} - \mathbf{G}^{-1} \mathbf{G}_d \right) \mathbf{W}_d \tag{18}$$

$$\mathbf{M}_e = \mathbf{J}_{uu}^{1/2} \mathbf{G}^{-1} \mathbf{H} \mathbf{W}_e \tag{19}$$

Here, $\mathbf{J}_{uu}$ and $\mathbf{J}_{ud}$ represent $\frac{\partial^2 J}{\partial u^2}$ and $\frac{\partial^2 J}{\partial u \partial d}$, evaluated at the nominally optimal operating point, respectively.

3.3 Selection of controlled variables

Individual measurements. The loss in (17) depends on $\mathbf{H}$ and CVs are selected by minimizing loss with respect to $\mathbf{H}$. When individual measurements are selected as CVs, the elements of $\mathbf{H}$ are restricted to be 0 or 1 and

$$\mathbf{H}\mathbf{H}^T = \mathbf{I} \tag{20}$$

In words, selection of a subset of available measurements as CVs involves selecting n_u among n_y measurements, where the number of available alternatives is $\mathcal{C}_{n_y}^{n_u}$. We note that, however, the expression for L_1 in (17) requires inversion of $\mathbf{G}$ and thus only holds, when $\mathbf{G}$ is a square matrix. On the other hand, BAB methods require evaluation of loss, when the number of selected measurements differs from n_u. Motivated by this drawback, we present an alternate representation of L_1 in the following discussion. For notational simplicity, we define

$$\mathbf{Y} = \left[\left(\mathbf{G}^y \mathbf{J}_{uu}^{-1} \mathbf{J}_{ud} - \mathbf{G}_d^y \right) \mathbf{W}_d \quad \mathbf{W}_e \right] \tag{21}$$

$$\tilde{\mathbf{G}} = \mathbf{G}^y \mathbf{J}_{uu}^{-1/2} \tag{22}$$

Now, L_1 can be represented as

$$L_1(\mathbf{H}) = \frac{1}{2}\bar{\sigma}^2 \left((\mathbf{H}\tilde{\mathbf{G}})^{-1} \mathbf{H}\mathbf{Y} \right) \tag{23}$$

$$= \frac{1}{2}\bar{\lambda} \left((\mathbf{H}\tilde{\mathbf{G}})^{-1} \mathbf{H}\mathbf{Y}\mathbf{Y}^T\mathbf{H}^T (\mathbf{H}\tilde{\mathbf{G}})^{-T} \right) \tag{24}$$

$$= \frac{1}{2}\underline{\lambda}^{-1} \left((\mathbf{H}\tilde{\mathbf{G}})^T (\mathbf{H}\mathbf{Y}\mathbf{Y}^T\mathbf{H}^T)^{-1} \mathbf{H}\tilde{\mathbf{G}} \right) \tag{25}$$

where $\underline{\lambda}(\cdot)$ denotes the least non-zero eigenvalue. We note that in practice, every measurement has non-zero implementation error associated with it. Thus, based on (21), $[\mathbf{W}_e]_{ii} \neq 0$ and $\mathbf{Y}$ has full row rank.

These observations imply that the inverse of $\mathbf{H}\mathbf{Y}\mathbf{Y}^T\mathbf{H}^T$ is well defined for all practical problems and the expression for L_1 in (25) holds for any number of measurements.

To represent L_1 in (25) using index notation, let X_p be an p-element index set, $p \leq n_y$, consisting of the indices of selected measurements and, $\tilde{\mathbf{G}}_{X_p}$ and $\mathbf{Y}_{X_p}$ consist of rows of $\tilde{\mathbf{G}}$ and $\mathbf{Y}$ with indices in X_p, respectively. Then,

$$L_1(X_p) = \frac{1}{2}\underline{\lambda}^{-1}\left(\tilde{\mathbf{G}}_{X_p}^T(\mathbf{Y}_{X_p}\mathbf{Y}_{X_p}^T)^{-1}\tilde{\mathbf{G}}_{X_p}\right) \tag{26}$$

$$= \frac{1}{2}\underline{\lambda}^{-1}\left(\mathbf{R}^{-T}\tilde{\mathbf{G}}_{X_p}\tilde{\mathbf{G}}_{X_p}^T\mathbf{R}^{-1}\right) \tag{27}$$

where $\mathbf{R}^T\mathbf{R} = \mathbf{Y}_{X_p}\mathbf{Y}_{X_p}^T$ (Cholesky factorization). As the expressions for L_1 in (17) and (27) are same for $p = n_u$, the optimal set of CVs can be found by minimizing L_1 in (27).

Measurement combinations. When individual measurements are used as CVs, the information contained in only n_u out of n_y measurements is used for updating the inputs. Clearly, better self-optimizing properties or lower loss can be obtained by using the information contained in other measurements as well. This can be achieved by using combinations of all the available measurements as CVs. For example, for the binary distillation column case study discussed in Section 5.2, the lowest achievable loss is 0.2809 with the use of individual measurements as CVs. The loss for this process, however, decreases approximately 5 times to 0.0517, when combinations of all available measurements as CVs.

When measurement combinations are used CVs, the integer constraint on $\mathbf{H} \in \mathbb{R}^{n_u \times n_y}$ is relaxed, but the condition $\text{rank}(\mathbf{H}) = n_u$ is still imposed to ensure invertibility of $\mathbf{H}\,\mathbf{G}^y$. The minimal worst-case loss over the set (16) using measurements combinations as CVs is given as [12, 13]

$$L_2 = \min_{\mathbf{H}} L_1 = \frac{1}{2}\lambda_{n_u}^{-1}\left(\tilde{\mathbf{G}}^T\left(\mathbf{Y}\,\mathbf{Y}^T\right)^{-1}\tilde{\mathbf{G}}\right) \tag{28}$$

Equation (28) can be used to calculate the minimum loss provided by the optimal combination of a given set of measurements. It is noted in [1, 10, 12, 13], however, that use of all measurements is often unnecessary and equivalent losses can be obtained by combining only a few of the available measurements. Then, the combinatorial optimization problem involves finding the set of n among n_y measurements ($n_u \leq n \leq n_y$) that can provide minimal loss, where n is specified.

In index notation, for a given p-element index set X_p, L_2 is denoted as

$$L_2(X_p) = \frac{1}{2}\underline{\lambda}^{-1}\left(\tilde{\mathbf{G}}_{X_p}^T\left(\mathbf{Y}_{X_p}\,\mathbf{Y}_{X_p}^T\right)^{-1}\tilde{\mathbf{G}}_{X_p}\right) \tag{29}$$

$$= \frac{1}{2}\underline{\lambda}^{-1}\left(\mathbf{R}^{-T}\tilde{\mathbf{G}}_{X_p}\tilde{\mathbf{G}}_{X_p}^T\mathbf{R}^{-1}\right) \tag{30}$$

where $n_u \leq p \leq n_y$ and $\mathbf{R}^T\mathbf{R} = \mathbf{Y}_{X_p}\mathbf{Y}_{X_p}^T$ (Cholesky factorization). Now, the set of best n measurements, whose combination can be used as CVs, can be selected by minimizing L_2 in (30). Note that $L_2(X_p) = L_1(X_p)$ for $n_u \leq p \leq n_y$. This observation noted in this work is useful for development of efficient BAB approaches, as discussed in the next section.

4 Bidirectional controlled variable selection

As shown in Section 3, the selection of CVs using exact local method can be seen as subset selection problems. In this section, we present BAB methods for solving these problems efficiently. For simplicity of notation, we define the $p \times p$ matrix $\mathbf{M}(X_p)$ as

$$\mathbf{M}(X_p) = \mathbf{R}^{-T}\tilde{\mathbf{G}}_{X_p}\tilde{\mathbf{G}}_{X_p}^T\mathbf{R}^{-1} \tag{31}$$

where $\mathbf{R}$ is the Cholesky factor of $\mathbf{Y}_{X_p}\mathbf{Y}_{X_p}^T$. Moreover, we denote the $n_u \times n_u$ matrix $\mathbf{N}(X_p)$ as

$$\mathbf{N}(X_p) = \tilde{\mathbf{G}}_{X_p}^T(\mathbf{Y}_{X_p}\mathbf{Y}_{X_p}^T)^{-1}\tilde{\mathbf{G}}_{X_p} \tag{32}$$

Note that $\underline{\lambda}(\mathbf{M}(X_p)) = \underline{\lambda}(\mathbf{N}(X_p))$.

4.1 Monotonicity

In this section, we show that the loss expressions in (27) and (30) or their lower bounds satisfy the monotonicity requirement and thus are amenable to the application of BAB approach discussed in Section 2.

Individual measurements. To prove monotonicity for L_1 in (27), we use the following property of matrices:

Lemma 1 Let the matrix $\hat{\mathbf{A}}$ be defined as

$$\hat{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{b}^T & a \end{bmatrix} \tag{33}$$

where $\mathbf{A} \in \mathbb{R}^{p\times p}$ is a Hermitian matrix, $\mathbf{b} \in \mathbb{R}^{p\times 1}$ and $a \in \mathbb{R}$. Let the eigenvalues of $\mathbf{A}$ and $\hat{\mathbf{A}}$ be arranged in descending order. Then [11, Th. 4.3.8]

$$\lambda_{p+1}(\hat{\mathbf{A}}) \leq \lambda_p(\mathbf{A}) \leq \lambda_p(\hat{\mathbf{A}}) \leq \lambda_{p-1}(\mathbf{A}) \leq \cdots \leq \lambda_1(\mathbf{A}) \leq \lambda_1(\hat{\mathbf{A}}) \tag{34}$$

Proposition 1 Consider a node $\mathcal{S} = (F_f, C_c)$ and index $i \in C_c$. For L_1 defined in (27),

$$L_1(F_f) \leq L_1(F_f \cup i); \quad f < n_u \tag{35}$$

$$L_1((F_f \cup C_c) \setminus i) \geq L_1(F_f \cup C_c); \quad f + c > n_u \tag{36}$$

Proof: Let $\tilde{\mathbf{G}}_{F_f \cup i} = \begin{bmatrix} \tilde{\mathbf{G}}_{F_f}^T & \tilde{\mathbf{G}}_i^T \end{bmatrix}^T$ and $\mathbf{Y}_{F_f \cup i} = \begin{bmatrix} \mathbf{Y}_{F_f}^T & \mathbf{Y}_i^T \end{bmatrix}^T$. Further, let $\mathbf{R}^T\mathbf{R} = \mathbf{Y}_{F_f}\mathbf{Y}_{F_f}^T$ and $\tilde{\mathbf{R}}^T\tilde{\mathbf{R}} = \mathbf{Y}_{F_f \cup i}\mathbf{Y}_{F_f \cup i}^T$ (Cholesky factorization). Then, it follows that $\mathbf{R}$ and $\mathbf{M}(F_f)$ are principal submatrices of $\tilde{\mathbf{R}}$ and $\mathbf{M}(F_f \cup i)$, respectively, obtained by deleting the last row and column of the corresponding matrices. Using (34), we have

$$\lambda_f^{-1}(\mathbf{M}(F_f)) \leq \lambda_{f+1}^{-1}(\mathbf{M}(F_f \cup i)); \quad f < n_u \tag{37}$$

which implies (35).

To prove (36), let $\mathbf{R}^T\mathbf{R} = \mathbf{Y}_{(F_f \cup C_c)\setminus i}\mathbf{Y}_{(F_f \cup C_c)\setminus i}^T$ and $\tilde{\mathbf{R}}^T\tilde{\mathbf{R}} = \mathbf{Y}_{F_f \cup C_c}\mathbf{Y}_{F_f \cup C_c}^T$ (Cholesky factorization). As before, it can be shown that $\mathbf{R}$ and $\mathbf{M}((F_f \cup C_c) \setminus i)$ are principal submatrices of $\tilde{\mathbf{R}}$ and $\mathbf{M}(F_f \cup C_c)$, respectively. Based on (34), we have

$$\lambda_{n_u}^{-1}(\mathbf{M}((F_f \cup C_c) \setminus i)) \geq \lambda_{n_u}^{-1}(\mathbf{M}(F_f \cup C_c)); \quad f + c > n_u \tag{38}$$

Now the result follows by noting that $\lambda_j(\mathbf{M}((F_f \cup C_c) \setminus i)) = \lambda_j(\mathbf{M}(F_f \cup C_c)) = 0$ for $j > n_u$ and thus $\lambda_{n_u}(\mathbf{M}((F_f \cup C_c) \setminus i))$ and $\lambda_{n_u}(\mathbf{M}(F_f \cup C_c))$ represent the least non-zero eigenvalues of $\mathbf{M}((F_f \cup C_c) \setminus i)$ and $\mathbf{M}(F_f \cup C_c)$, respectively. ■

Based on Proposition 1, it follows that $L_1(F_f)$ and $L_1(F_f \cup C_c)$ represent lower bounds on the loss seen using any n_u measurements as CVs, which can be obtained by appending measurement indices to F_f or removing measurement indices from $F_f \cup C_c$, respectively. Let B represent the best available upper bound on $L_1(X_{n_u}^{\text{opt}})$. Then repeated application of (35) implies that, if $L_1(F_f) > B$, the optimal solution cannot be a superset of F_f and hence all supersets of F_f need not be evaluated. Similarly, if $L_1(F_f \cup C_c) > B$, repeated application of (36) implies that the optimal solution cannot be a subset of $F_f \cup C_c$ and hence all subsets of $F_f \cup C_c$ need not be evaluated. Thus, upwards and downwards pruning can be conduced using (35) and (36), respectively, and the optimal solution can be found without complete enumeration.

Measurements combinations. We note that the expression for L_2 in (30) is the same as the expression for L_1 in (27). Thus, based on Proposition 1

$$L_2((F_f \cup C_c) \setminus i) \geq L_2(F_f \cup C_c); \quad f + c > n \tag{39}$$

For selecting a subset of measurements, whose linear combinations can be used as CVs, the result in (39) can be used for downwards pruning. Equation (36), however, also implies that when $n_u \leq f < n$, $L_2(F_f)$ decreases as the size of the fixed set increases. Thus, unlike L_1, L_2 does not posses upwards monotonicity. In the following proposition, we present a lower bound on L_2, which shows upwards monotonicity, whenever $n - n_u < f < n$.

Proposition 2 For the node $\mathcal{S} = (F_f, C_c)$, let

$$\underline{L}_2(F_f) = 0.5\lambda^{-1}_{f+n_u-n}\left(\mathbf{M}(F_f)\right); \quad f > n - n_u \tag{40}$$

Then, $\underline{L}_2(F_f)$ represents a lower bound on the loss corresponding to combinations of any n measurements obtained by appending indices to F_f, *i.e.*

$$\underline{L}_2(F_f) \quad \leq \quad \min_{\substack{X_n \supset F_f \\ X_n \subset (F_f \cup C_c)}} L_2(X_n) \tag{41}$$

where L_2 is defined in (30). Furthermore, $\underline{L}_2(F_f)$ satisfies upwards monotonicity, *i.e.* for any $i \in C_c$

$$\underline{L}_2(F_f) \quad \leq \quad \underline{L}_2(F_f \cup i); \quad f < n \tag{42}$$

Proof: Consider the index set $X_n \subset (F_f \cup C_c)$. For $j \in X_n$ with $j \notin F_f$, similar to the proof of Proposition 1, it can be shown that $\mathbf{M}(X_n \setminus j)$ is a principal submatrix of $\mathbf{M}(X_n)$. Based on the interlacing property of eigenvalues in (34), we have that

$$\lambda^{-1}_{n_u-1}(\mathbf{M}(X_n \setminus j)) \quad \leq \quad \lambda^{-1}_{n_u}(\mathbf{M}(X_n)) \tag{43}$$

Through repeated application of (34), for $i \in X_n$, $i \notin F_f$ and $i \neq j$

$$\lambda^{-1}_{n_u-(n-f)}(\mathbf{M}(F_f)) \leq \lambda^{-1}_{n_u-(n-f-1)}(\mathbf{M}(F_f \cup i)) \leq \cdots \leq \lambda^{-1}_{n_u-1}(\mathbf{M}(X_n \setminus j)) \leq \lambda^{-1}_{n_u}(\mathbf{M}(X_n)) \tag{44}$$

which implies (41) and (42). ■

Proposition 2 implies that the lower bound of L_2 defined in (40) posses upwards monotonicity and thus can be used for upwards pruning. In this case, upwards pruning can only be applied when the size of fixed set of the node under consideration is greater than $n - n_u$. Thus, the BAB algorithm based on $\underline{L}_2$ in (40) is referred to as partial bidirectional BAB (PB^3) algorithm. Development of fully bidirectional BAB algorithm for selection of measurements, which can be combined to yield CVs, is an open problem.

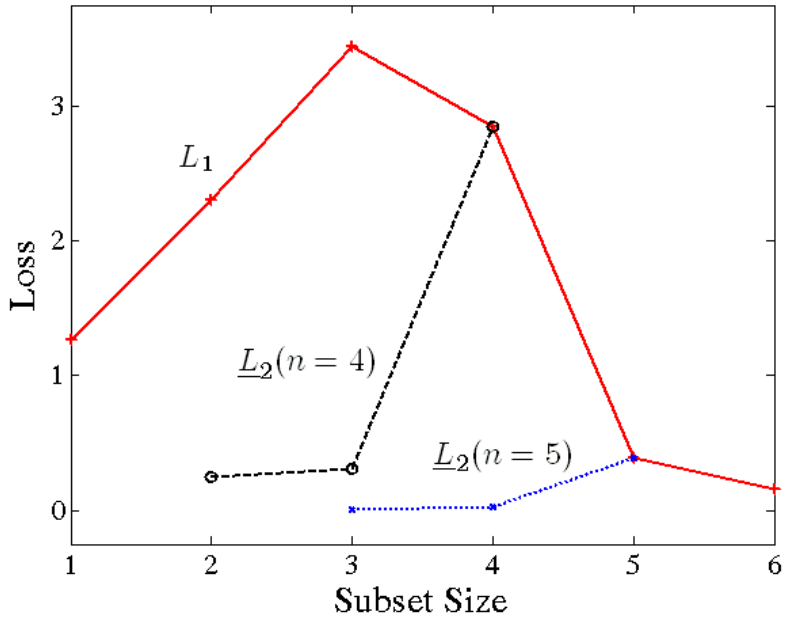

Figure 1: Monotonicity of local loss functions for CV selection; for subset size i, the loss is calculated for the measurement set $\{1, 2, \cdots, i\}$

Example 1. To illustrate the findings of this section, we use a simple toy example, where

$$\mathbf{G}^y = \begin{bmatrix} 15 & 4 & -4 \\ 10 & -1 & 6 \\ 3 & 7 & 6 \\ -8 & -18 & 10 \\ -5 & 12 & 9 \\ 9 & -1 & 12 \end{bmatrix}, \quad \mathbf{G}_d^y = \begin{bmatrix} -2 & -4 \\ -7 & 3 \\ -6 & 8 \\ -3 & -10 \\ 12 & -1 \\ -15 & 1 \end{bmatrix},$$

$\mathbf{J_{uu}} = \mathbf{I}_3$, $\mathbf{J_{ud}} = \mathbf{1}_{3,2}$ (matrix of ones), $\mathbf{W_d} = \mathbf{I}_2$ and $\mathbf{W_n} = \mathbf{I}_6$. Figure 1 shows the variation of local loss for different subset sizes. For selection of individual measurements as CVs, the local loss L_1 (solid line in Figure 1) monotonically increases with subset size i, when $i \leq 3$ (n_u) and monotonically decreases with subset size i, when $i \geq 3$, hence showing bidirectional monotonicity. The case with $i > 3$ corresponds to the use of combinations of selected measurements as CVs and thus a lower loss is expected than seen using 3 individual measurements as CVs. The local loss decreases, even when the subset size $i < 3$. This happens as the control of every CV can be seen as an equality constraint for the optimization problem describing the optimal operation of the process. Thus, the increase in the number of such equality constraints results in increased loss because it leads to more sub-optimal variation of the inputs as compared to the truly optimal operation, which has no equality constraints.

For selection of measurements, whose combinations can be used as CVs, the local loss L_2 only satisfies downwards monotonicity. For example, when $n = 4$, $L_2 = L_1$ increases when the subset size is decreased from 5 to 4. In comparison the lower bound on local loss $\underline{L}_2$ (dashed line in Figure 1) satisfies upwards monotonicity and can be used for application of bidirectional BAB method. Note that, however, the lower bound $\underline{L}_2$ can be used only when the subset size is greater than $n - n_u$, *i.e.* $4 - 3 = 1$ for $n = 4$ and $5 - 3 = 2$ for $n = 5$.

4.2 Fast pruning algorithms

As the criterion for selection of CVs using the exact local method satisfy bidirectional monotonicity, the non-optimal nodes can be pruned quickly. Thus, the optimal solution can be found with evaluation of fewer nodes, but the solution time can still be large, as direct evaluation of L_1 in (27) and $\underline{L}_2$ in (40) is computationally expensive. We note that for pruning purposes, it suffices to know whether a sub- or super-node obtained from the node under consideration can provide a better bound than the best available bound B. With this observation, we present computationally efficient determinant-based pruning algorithms such that the evaluation of L_1 and $\underline{L}_2$ is avoided at non-terminal nodes.

Individual measurements. We first present fast pruning algorithms for selection of a subset of available measurements as CVs through minimization of L_1 in (26). The case, where combinations of available measurements are used as CVs through minimization of L_2 in (30) is dealt with later in this section.

Proposition 3 (Upwards pruning for L_1) Consider a node $\mathcal{S} = (F_f, C_c)$ and index $i \in C_c$. For a given positive scalar B, if $L_1(F_f) < B$,

$$\beta_i = \mathbf{d}_i^T \mathbf{d}_i - \mathbf{d}_i^T \mathbf{D}^T \left(\mathbf{M}(F_f) - (0.5/B)\mathbf{I}_f\right)^{-1} \mathbf{D}\mathbf{d}_i < (0.5/B) \Leftrightarrow L_1(F_f \cup i) > B \tag{45}$$

where $\mathbf{D} = \mathbf{R}^{-T}\tilde{\mathbf{G}}_{F_f}$ with $\mathbf{R}$ being the Cholesky factor of $\mathbf{Y}_{F_f}\mathbf{Y}_{F_f}^T$, and $\mathbf{d}_i^T = (\tilde{\mathbf{G}}_i - \mathbf{p}_i^T\mathbf{D})/\delta_i$ with $\mathbf{p}_i = \mathbf{R}^{-T}\mathbf{Y}_{F_f}\mathbf{Y}_i^T$ and $\delta_i = \sqrt{\mathbf{Y}_i\mathbf{Y}_i^T - \mathbf{p}_i^T\mathbf{p}_i}$.

Proof: Let $\mathbf{Q}$ be the Cholesky factor of $\mathbf{Y}_{F_f\cup i}\mathbf{Y}_{F_f\cup i}^T$, *i.e.* $\mathbf{Q}^T\mathbf{Q} = \mathbf{Y}_{F_f\cup i}\mathbf{Y}_{F_f\cup i}^T$. Through simple algebraic manipulations, it can be shown that

$$\mathbf{Q} = \begin{bmatrix} \mathbf{R} & \mathbf{p}_i \\ 0 & \delta_i \end{bmatrix}; \qquad \mathbf{Q}^{-T}\tilde{\mathbf{G}}_{F_f\cup i} = \begin{bmatrix} \mathbf{D} \\ \mathbf{d}_i^T \end{bmatrix} \tag{46}$$

Since, $L_1(F_f) < B$, $\underline{\lambda}(\mathbf{M}(F_f)) > 0.5/B$, which implies that $\det(\mathbf{M}(F_f) - (0.5/B)\mathbf{I}_f) > 0$ [5]. Using Schur complement Lemma [11],

$$\det(\mathbf{M}(F_f \cup i) - (0.5/B)\mathbf{I}_{f+1}) = (\beta_i - 0.5/B)\det(\mathbf{M}(F_f) - (0.5/B)\mathbf{I}_f) \tag{47}$$

Thus, $\beta_i < 0.5/B \Leftrightarrow \det(\mathbf{M}(F_f \cup i) - (0.5/B)\mathbf{I}_f) < 0$, as $\det(\mathbf{M}(F_f) - (0.5/B)\mathbf{I}_f) > 0$. Now,

$$\det(\mathbf{M}(F_f \cup i) - (0.5/B)\mathbf{I}_f) = (\underline{\lambda}(\mathbf{M}(F_f \cup i)) - 0.5/B)\prod_{i=1}^{f}(\lambda_i(\mathbf{M}(F_f \cup i)) - 0.5/B) \tag{48}$$

Since $\underline{\lambda}(\mathbf{M}(F_f)) > 0.5/B$, $(\lambda_k(\mathbf{M}(F_f \cup i)) - 0.5/B) > 0$ for $k = 1, 2, \cdots f$ due to interlacing property of eigenvalues. Finally, we have $\beta_i < 0.5/B \Leftrightarrow \underline{\lambda}(\mathbf{M}(F_f \cup i)) < 0.5/B \Leftrightarrow L_1(F_f \cup i) > B$. ■

For a node $\mathcal{S} = (F_f, C_c)$, if the condition (45) is satisfied, then candidate i can be discarded or the super-node $\mathcal{S}^i = (F_f \cup i, C_c \setminus i)$ can be pruned. Therefore, this condition is referred to as upwards pruning condition. Furthermore, the main computation load in condition (45) is the Cholesky factorization and the matrix inversion, which needs to be calculated only once for all $i \in C$. Hence, this test is more efficient than direct calculation of L_1.

Proposition 4 (Downward pruning for L_1) For a node $\mathcal{S} = (F_f, C_c)$, let $S_s = F_f \cup C_c$, where $s = f + c$. For a given positive scalar B, if $L_1(S_s) < B$,

$$\alpha_i = 1 - \mathbf{x}_i^T(\mathbf{N}(S_s) - (0.5/B)\mathbf{I}_{n_u})^{-1}\mathbf{x}_i/\eta_i < 0 \Leftrightarrow L_1(S_s \setminus i) > B \tag{49}$$

where $\mathbf{x}_i = \tilde{\mathbf{G}}_{S_s\setminus i}^T(\mathbf{Y}_{S_s\setminus i}\mathbf{Y}_{S_s\setminus i}^T)^{-1}\mathbf{Y}_{S_s\setminus i}\mathbf{Y}_i^T - \tilde{\mathbf{G}}_i^T$ and $\eta_i = \mathbf{Y}_i(\mathbf{I} - \mathbf{Y}_{S_s\setminus i}^T(\mathbf{Y}_{S_s\setminus i}\mathbf{Y}_{S_s\setminus i}^T)^{-1}\mathbf{Y}_{S_s\setminus i})\mathbf{Y}_i^T$.

Proof: For simplicity of notation, define $\mathbf{Q} = \mathbf{Y}_{S_s\setminus i}\mathbf{Y}_{S_s\setminus i}^T$. Then,

$$(\mathbf{Y}_{S_s}\mathbf{Y}_{S_s}^T)^{-1} = \begin{bmatrix} \mathbf{Q} & \mathbf{Y}_{S_s\setminus i}\mathbf{Y}_i^T \\ \mathbf{Y}_i\mathbf{Y}_{S_s\setminus i}^T & \mathbf{Y}_i\mathbf{Y}_i^T \end{bmatrix}^{-1} \tag{50}$$

$$= \begin{bmatrix} \mathbf{Q}^{-1} + \mathbf{Q}^{-1}\mathbf{Y}_{S_s\setminus i}\mathbf{Y}_i^T\mathbf{Y}_i\mathbf{Y}_{S_s\setminus i}^T\mathbf{Q}^{-1}/\eta_i & -\mathbf{Q}^{-1}\mathbf{Y}_{S_s\setminus i}\mathbf{Y}_i^T/\eta_i \\ -\mathbf{Y}_i\mathbf{Y}_{S_s\setminus i}^T\mathbf{Q}^{-1}/\eta_i & 1/\eta_i \end{bmatrix} \tag{51}$$

$$= \begin{bmatrix} \mathbf{Q}^{-1} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} + 1/\eta_i \begin{bmatrix} \mathbf{Q}^{-1}\mathbf{Y}_{S_s\setminus i}\mathbf{Y}_i^T \\ -1 \end{bmatrix} \begin{bmatrix} \mathbf{Y}_i\mathbf{Y}_{S_s\setminus i}^T\mathbf{Q}^{-1} & -1 \end{bmatrix} \tag{52}$$

where (51) is obtained using the matrix inversion formula for partitioned matrices [11]. Since $\tilde{\mathbf{G}}_{S_s}^T = \begin{bmatrix} \tilde{\mathbf{G}}_{S_s\setminus i}^T & \tilde{\mathbf{G}}_i^T \end{bmatrix}$, we have

$$\mathbf{N}(S_s) = \tilde{\mathbf{G}}_{S_s}^T(\mathbf{Y}_{S_s}\mathbf{Y}_{S_s}^T)^{-1}\tilde{\mathbf{G}}_{S_s} = \tilde{\mathbf{G}}_{S_s\setminus i}^T\mathbf{Q}^{-1}\tilde{\mathbf{G}}_{S_s\setminus i} + \mathbf{x}_i\mathbf{x}_i^T/\eta_i = \mathbf{N}(S_s \setminus i) + \mathbf{x}_i\mathbf{x}_i^T/\eta_i \tag{53}$$

which implies that

$$\det(\mathbf{N}(S_s \setminus i) - (0.5/B)\mathbf{I}_{n_u}) = \det(\mathbf{N}(S_s) - (0.5/B)\mathbf{I}_{n_u} - \mathbf{x}_i\mathbf{x}_i^T/\eta_i) \quad (54)$$

$$= \det(\mathbf{N}(S_s) - (0.5/B)\mathbf{I}_{n_u})\alpha_i \quad (55)$$

As $L_1(S_s) < B$, $\underline{\lambda}(\mathbf{N}(S_s)) > 0.5/B$, which implies that $\det(\mathbf{N}(S_s) - (0.5/B)\mathbf{I}_{n_u}) > 0$ [5]. Thus $\alpha_i < 0 \Leftrightarrow \det(\mathbf{N}(S_s \setminus i) - (0.5/B)\mathbf{I}_{n_u}) < 0$. Using similar arguments about the interlacing property of eigenvalues, as used in the proof of Proposition 3, we have that $\alpha_i < 0 \Leftrightarrow \underline{\lambda}(\mathbf{N}(S_s \setminus i) < 0.5/B \Leftrightarrow L_1(S_s \setminus i) > B$. ■

For a node $\mathcal{S} = (F_f, C_c)$, if the condition (49) is satisfied, candidate i can be fixed or the sub-node $\mathcal{S}^i = (F_f, C_c \setminus i)$ can be pruned. Therefore, this condition is referred to as downwards pruning condition. To evaluate the computational efficiency of the condition (49), let the index set $S_s = F_f \cup C_c$ be permuted such that the index i is the last element of S_s. Now, based on (51), we note that $1/\eta_i$ is the (n_u, n_u)th element of $(\mathbf{Y}_{S_s}\mathbf{Y}_{S_s}^T)^{-1}$ and $\mathbf{x}_i^T/\eta_i$ is the last row of the matrix $-(\mathbf{Y}_{S_s}\mathbf{Y}_{S_s}^T)^{-1}\tilde{\mathbf{G}}_{S_s}$. Therefore, the use of condition (49) requires inversion of two matrices, $(\mathbf{Y}_{S_s}\mathbf{Y}_{S_s}^T)^{-1}$ and $(\mathbf{N}(S_s) - (0.5/B)\mathbf{I}_{n_u})^{-1}$, which need to be calculated only once for all $i \in C_c$. Hence, this test is more efficient than direct calculation of L_1.

As L_1 in (26) satisfies bidirectional monotonicity, both upwards and downwards pruning conditions in Propositions 3 and 4, respectively, can be applied simultaneously reducing the solution time enormously.

Measurement combinations. As before, the downwards pruning condition presented in Proposition 4 can also be applied for selection of CVs as combinations of available measurements. In the next proposition, we present algorithms for fast upwards pruning for measurement selection through minimization of L_2.

Proposition 5 (Partially upwards pruning rules for L_2) Consider a node $\mathcal{S} = (F_f, C_c)$ and index $i \in C_c$. For a positive scalar B and $f > n - n_u$, if $\underline{L}_2(F_f) < B$ or $\lambda_{f+n_u-n}(\mathbf{M}(F_f)) > 0.5/B$ and $\lambda_{f+n_u-n+1}(\mathbf{M}(F_f)) < 0.5/B$,

$$\beta_i < (0.5/B) \Leftrightarrow \underline{L}_2(F_f \cup i) > B \quad (56)$$

where β_i is defined in (45).

Proof: Based on (47), we have

$$\prod_{j=1}^{f+1} \lambda_j(\mathbf{M}(F_f \cup i) - (0.5/B)) = (\beta_i - 0.5/B) \prod_{j=1}^{f} \lambda_j(\mathbf{M}(F_f) - (0.5/B)) \quad (57)$$

As $\mathbf{M}(F_f)$ is a principal submatrix of $\mathbf{M}(F_f \cup i)$, the interlacing property of eigenvalues implies that

$$\lambda_{f+n_u-n+1}(\mathbf{M}(F_f)) \leq \lambda_{f+n_u-n+1}(\mathbf{M}(F_f \cup i)) \leq \lambda_{f+n_u-n}(\mathbf{M}(F_f)) \leq \lambda_{f+n_u-n}(\mathbf{M}(F_f \cup i)) \tag{58}$$

Since $\lambda_{f+n_u-n}(\mathbf{M}(F_f)) > 0.5/B$, (58) implies that $\lambda_j(\mathbf{M}(F_f)) > 0.5/B$ and $\lambda_j(\mathbf{M}(F_f \cup i)) > 0.5/B$ for $j = 1, 2, \cdots, f+n_u-n$. Similarly, since $\lambda_{f+n_u-n+1}(\mathbf{M}(F_f)) < 0.5/B$, (58) implies that $\lambda_k(\mathbf{M}(F_f)) < 0.5/B$ and $\lambda_{k+1}(\mathbf{M}(F_f \cup i)) < 0.5/B$ for $k = f+n_u-n+1, \cdots, f$. Since equal number of eigenvalues of $\mathbf{M}(F_f)$ and $\mathbf{M}(F_f \cup i)$ are greater than and less than $0.5/B$, based on (57), the signs of $(\lambda_{f+n_u-n+1}(\mathbf{M}(F_f \cup i)) - 0.5/B)$ and $(\beta_i - 0.5/B)$ are the same and the result follows. ■

The reader should note the similarities between Propositions 3 and 5, when $n = n_u$. Note that Proposition 5 requires checking whether $\lambda_{f+n_u-n+1}(\mathbf{M}(F_f)) < 0.5/B$. When this condition is not satisfied, due to the interlacing property of eigenvalues, $\lambda_{f+n_u-n+1}(\mathbf{M}(F_f \cup i)) > 0.5/B$ or $\underline{L}_2(F_f \cup i) < B$ for all $i \in C_c$. Thus, any super-node of the node under consideration cannot be pruned.

4.3 Fast branching algorithms

The availability of fast pruning algorithms avoids the calculation of loss at non-terminal nodes for pruning purposes. The efficiency of the bidirectional BAB method can be further improved using bidirectional branching. As mentioned in Section 2.3, bidirectional branching involves selecting a decision element so that the upward and downward branches can be formulated. Here, the decision element itself is chosen on a best-first basis, *i.e.* the element that leads to the lowest loss among the members of the candidate set is taken as the decision element. Thus, the loss still needs to be calculated at non-terminal nodes for the selection of the decision element. In this section, we establish relationships between the pruning indices (α and β) calculated for different nodes and the expected loss upon expansion of these nodes. These results can be used to avoid the loss computation at non-terminal nodes entirely, hence greatly enhancing the computational efficiency.

Proposition 6 (Loss bounds for fast branching) For a node $\mathcal{S} = (F_f, C_c)$, let $S_s = F_f \cup C_c$, where $s = f + c$. For a given positive scalar B and $i \in C_c$

$$L_1(F_f \cup i) \geq 0.5/\beta_i \tag{59}$$

$$\frac{L_1^{-1}(S_s \setminus i) - 1/B}{L_1^{-1}(S_s) - 1/B} \leq \alpha_i \tag{60}$$

where β_i and α_i are given by (45) and (49), respectively.

Proof: To show that (59) holds, based on (47), we note that

$$\frac{\prod_{i=1}^{f+1}\left(\lambda_i(\mathbf{M}(F_f\cup i))-0.5/B\right)}{\prod_{i=1}^{f}\left(\lambda_i(\mathbf{M}(F_f))-0.5/B\right)} = \beta_i-0.5/B \tag{61}$$

$$\prod_{i=1}^{f}\frac{\lambda_i(\mathbf{M}(F_f\cup i))-0.5/B}{\lambda_i(\mathbf{M}(F_f))-0.5/B}\left(\underline{\lambda}(\mathbf{M}(F_f\cup i))-0.5/B\right) = \beta_i-0.5/B \tag{62}$$

Since $\mathbf{M}(F_f)$ is a principal submatrix of $\mathbf{M}(F_f\cup i)$, the interlacing property of eigenvalue implies that $\lambda_i(\mathbf{M}(F_f\cup i))\geq\lambda_i(\mathbf{M}(F_f))$, $i=1,2,\cdots,f$. Thus,

$$\prod_{i=1}^{f}\frac{\lambda_i(\mathbf{M}(F_f\cup i))-0.5/B}{\lambda_i(\mathbf{M}(F_f))-0.5/B}\geq 1 \tag{63}$$

and we have $\underline{\lambda}(\mathbf{M}(F_f\cup i))\leq\beta_i$, which implies (59). That (60) holds can be shown to be true similarly using (55), where

$$\prod_{i=1}^{n_u-1}\frac{\lambda_i(\mathbf{N}(S_s\setminus i))-0.5/B}{\lambda_i(\mathbf{N}(S_s))-0.5/B}\frac{\underline{\lambda}(\mathbf{N}(S_s\setminus i))-0.5/B}{\underline{\lambda}(\mathbf{N}(S_s))-0.5/B}=\alpha_i \tag{64}$$

Due to the interlacing properties of eigenvalues,

$$\prod_{i=1}^{n_u-1}\frac{\lambda_i(\mathbf{N}(S_s\setminus i))-0.5/B}{\lambda_i(\mathbf{N}(S_s))-0.5/B}\leq 1 \tag{65}$$

Thus, we have

$$\frac{\underline{\lambda}(\mathbf{N}(S_s\setminus i))-0.5/B}{\underline{\lambda}(\mathbf{N}(S_s))-0.5/B}\geq\alpha_i \tag{66}$$

which implies (60). ■

According to Proposition 6, both α and β can be used to select the decision element for bidirectional branching. More specifically, consider a selection problem $\mathcal{S}(F_f,C_c)$. Based on the discussion in Section 2.3, upward branch is evaluated first, if $2(n_u-f)\leq c-1$, and downward branch otherwise. For upward-first branching, the decision element is determined as the element with largest β_i or smallest α_i among all $i\in C_c$. Similarly, for downward-first branching, the decision element is selected as the element with smallest β_i or largest α_i among all $i\in C_c$.

We point out that the selection of decision element based on the loss relationships in (59) and (60) does not necessary result in a sub or super-node with smallest loss among the different alternatives. Although the sub-optimal choice of the decision element does not affect the optimality of the solution, it may lead to evaluation of more nodes for finding the optimal solution. Bidirectional branching based on α and β is still useful, as the computational load for calculating exact local loss at every node far outweighs the computational cost for evaluating a few additional nodes. A flowchart for recursive implementation of the

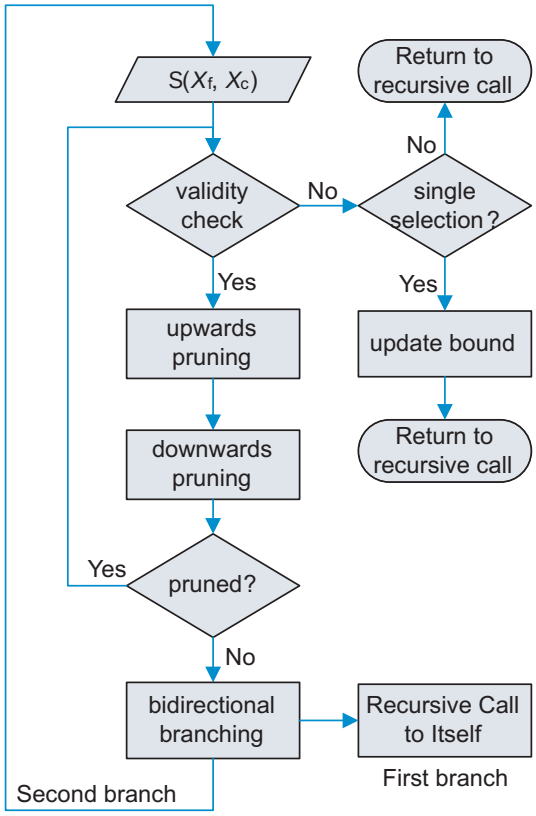

Figure 2: Flow chart of bidirectional branch and bound algorithm

bidirectional BAB (B^3) algorithm, based on the principles of bidirectional pruning and branching using the determinant based criteria developed in this paper, is shown in Figure 2.

For selection of n measurements, whose combinations can be used as CVs, the downward pruning index α can be used for selecting the decision element as before. It is, however, difficult to establish a relationship between the upwards pruning index β and expected loss L_2, when the node under consideration is expanded. For this reason, only the downward pruning index α is used to select the decision element for both upward-first and downward-first branching in the partially bidirectional BAB (PB^3) algorithm. The PB^3 algorithm can also be implemented using the flowchart shown in Figure 2, except that the upwards pruning condition only needs to be checked when $f > n - n_u$.

Example 1 continued. To illustrate the application of B^3 algorithm, Example 1 is revisited. The objective is to select 3 out of 6 measurements, which can be used as CVs. The bidirectional solution tree for this example is shown in Figure 3. The algorithm is initialized with $F = \emptyset$, $C = \{1, 2, 3, 4, 5, 6\}$ and $B = \infty$. As the current bound is infinite, no pruning is possible. For branching, only the upwards pruning indices are calculated as $\beta^{(0)} = \begin{bmatrix} 0.3948 & 0.2178 & 0.1712 & 2.3689 & 0.8170 & 0.1424 \end{bmatrix}$. Since $2(n_u - f) > (c-1)$ (*i.e.*

$6 > 5$), downward-first branching is desired. Hence, the decision element is chosen as the smallest element of $\beta^{(0)}$, *i.e.* measurement 6. Two sub-problems are generated by removing element 6 from the candidate set (sub-problem $\mathcal{S}_1$) and by moving element 6 from candidate to fixed set (sub-problem $\mathcal{S}_2$). As $\mathcal{S}_1$ is the upward branch, it is evaluated first.

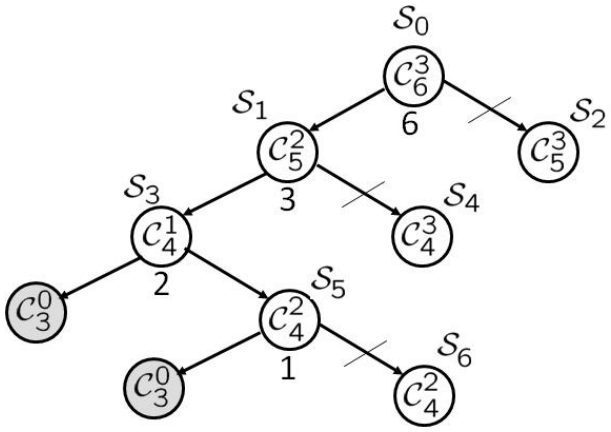

Figure 3: Bidirectional solution tree for the toy example

As the bound is still infinite, no pruning is possible and the calculation of downwards pruning index is not required. Moreover, $\beta^{(1)} = \beta^{(0)}$ (unchanged), as the fixed set F or the bound B has not changed as compared to the previous iteration. For sub-problem $\mathcal{S}_1$, $c = 5$ and $f = 0$. Since, $2(n_u - f) > c - 1$ (*i.e.* $6 > 4$), downward-first branching is conducted. Among the first 5 elements of $\beta^{(1)}$ (members of candidate set C), element 3 has the smallest value and is thus taken as the decision element for branching purposes. Again, two sub-problems are generated by removing element 3 from the candidate set (sub-problem $\mathcal{S}_3$) and by moving element 3 from candidate to fixed set (sub-problem $\mathcal{S}_4$), where sub-problem $\mathcal{S}_3$ is evaluated first.

For sub-problem $\mathcal{S}_3$, $c = 4$, *i.e.* only one element needs to be discarded. In this case, the use of downward pruning index is better than the use of upward pruning index for selecting the decision element. Therefore, it is calculated as $\alpha^{(3)} = \begin{bmatrix} 0.0084 & 0.0086 & * & 0.0018 & 0.0030 & * \end{bmatrix}$. A terminal node is obtained by removing element 2, which has the highest value of $\alpha^{(3)}$. The corresponding loss for this terminal node is $L_1 = 3.9537$ and the bound B is updated to be 3.9537. The other sub-problem $\mathcal{S}_5$ is obtained by moving element 2 from the candidate to the fixed set. As the bound is updated, β and α are calculated as $\beta^{(5)} = \begin{bmatrix} 1.9711 & * & * & 191.6600 & 1.1601 & * \end{bmatrix}$ and $\alpha^{(5)} = \begin{bmatrix} 0.0076 & * & * & 0.0014 & 0.0020 & * \end{bmatrix}$. Since every element of $\beta^{(5)}$ is greater than $0.5/B$ and every element of $\alpha^{(5)}$ is greater than zero, pruning is not possible. A terminal node is obtained by removing the element with highest value of $\alpha^{(5)}$, *i.e.* element 1. The loss for this terminal node with elements $\{2, 4, 5\}$ is $L_1 = 2.7704$. As the loss for this node is less than best

available bound, B is updated to 2.7704. This gives $0.5/B = 0.1805$, which is larger than both $\beta_3^0 = 0.1712$ and $\beta_6^0 = 0.1424$. Therefore, both elements 3 and 6 should be removed, *i.e.* both sub-problems $\mathcal{S}_2$, which has $F = \{6\}$, and $\mathcal{S}_4$, which has $F = \{3\}$, can be pruned without further evaluation.

As there are no problems left for evaluation, the algorithm terminates. The optimal subset is $\{2, 4, 5\}$, which provides a loss of 2.7704. The B^3 algorithm finds the optimal solution by evaluating 6 nodes, where as complete enumeration requires evaluation of 20 nodes.

5 Numerical examples

To examine the efficiency of the proposed BAB algorithms, numerical tests are conducted using randomly generated matrices and a binary distillation column case study. Programs used for loss minimization are listed in Table 1 [4]. All tests are conducted on a Windows XP SP2 notebook with an Intel® Core™ Duo Processor T2500 (2.0 GHz, 2MB L2 Cache, 667 MHz FSB) using MATLAB® R2008a.

Table 1: Branch and bound programs for loss minimization

program	description
UP	upwards pruning using determinant condition (45)
DOWN	downwards pruning using determinant condition (49)
B^3	bidirectional branch and bound algorithm by combining (45) and (49)
PB^3	partially bidirectional branch and bound algorithm by combining (49) and (56)

5.1 Random tests

Four sets of random tests are conducted to evaluate the efficiency of the B^3 algorithm for selection of a subset of available measurements as CVs through minimization of L_1. For each test, six random matrices are generated: three full matrices, $\mathbf{G}^y \in \mathbb{R}^{n_y \times n_u}$, $\mathbf{G}_d^y \in \mathbb{R}^{n_y \times n_d}$ and $\mathbf{J}_{ud} \in \mathbb{R}^{n_u \times n_d}$, and three diagonal matrices, $\mathbf{W}_e \in \mathbb{R}^{n_y \times n_y}$, $\mathbf{W}_d \in \mathbb{R}^{n_d \times n_d}$ and $\mathbf{J}_{uu} \in \mathbb{R}^{n_u \times n_u}$. The elements of these matrices are normally distributed with zero mean and unit variance. For all tests, we use $n_d = 5$, while n_u and n_y are varied.

The first and second tests are designed to select $n_u = 5$ and $n_u = n_y - 5$ out of n_y measurements, respectively. Each selection problem is tested for 100 sets of randomly generated matrices and the average computation time and average number of nodes evaluated are summarized in Figure 4. It is seen that upwards pruning based algorithm (UP) is more suitable for problems involving selection of a few variables

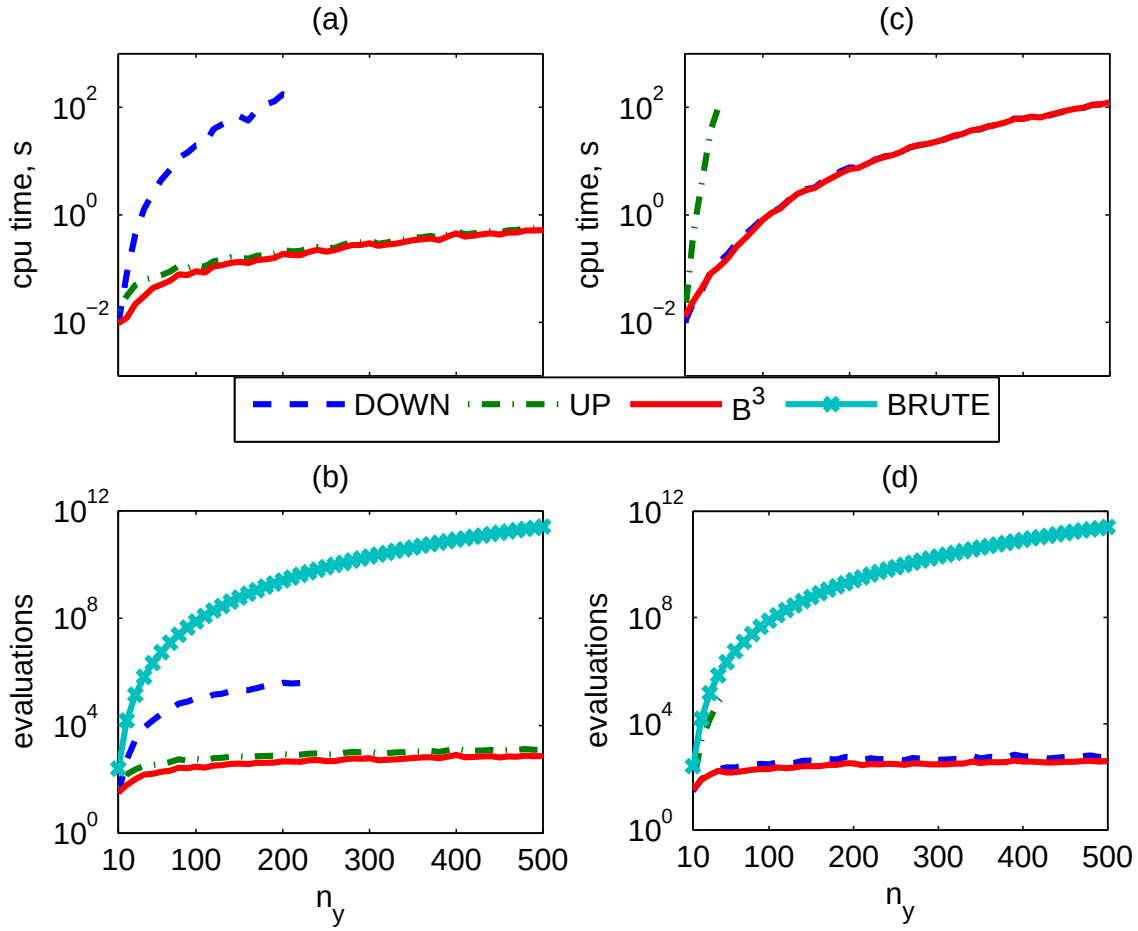

Figure 4: Random test 1: selection of 5 out of n_y measurements, (a) computation time against n_y and (b) number of nodes evaluated against n_y; Random test 2: selection of $n_y - 5$ out of n_y measurements, (c) computation time against n_y and (d) number of nodes evaluated against n_y.

from a large candidate set, whilst downwards pruning based algorithm (DOWN) is more efficient for problems, where a few among many candidate variables need to be discarded to find the optimal solution. The solution times for UP and DOWN algorithms increase only modestly with problem size, when $n_u << n_y$ and $n_u \approx n_y$, respectively. The solution times for the B^3 algorithm is similar to the better of UP and DOWN algorithms, however, its efficiency is insensitive to the kind of selection problem.

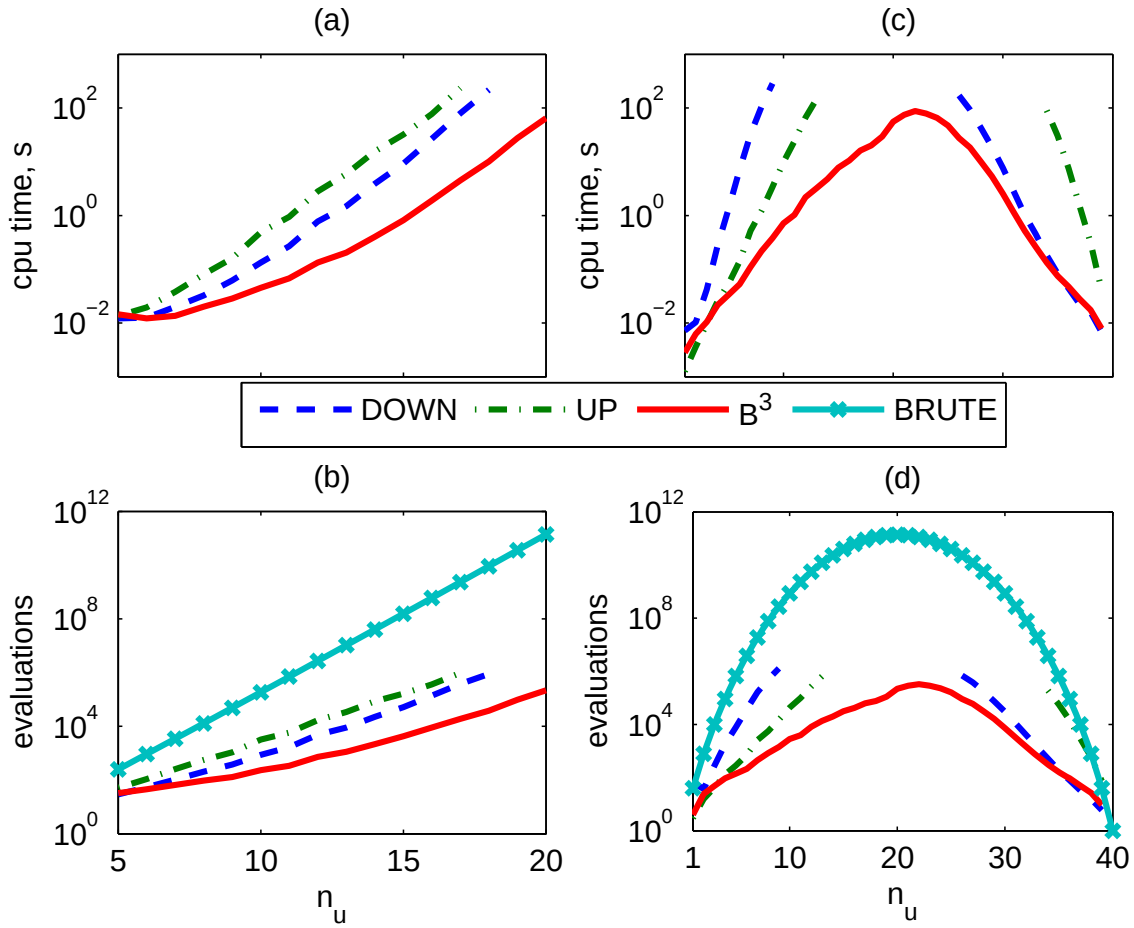

Figure 5: Random test 3: selection of n_u out of $n_y = 2n_u$ measurements, (a) computation time against n_u and (b) number of nodes evaluated against n_u; Random test 4: selection of n_u out of 40 measurements, (c) computation time against n_u and (d) number of nodes evaluated against n_u.

The third test consists of selecting n_u out of $n_y = 2n_u$ measurements with n_u increasing from 5 to 20, while the fourth test involves selecting n_u out of $n_y = 40$ variables with n_u ranging from 1 to 39. For each n_u, 100 sets of random matrices are generated and the average computation time and average number of nodes evaluated are summarized in Figure 5. While the UP and DOWN problems show reasonable performance for small n_u, their performances degrade rapidly for the fourth test, when n_u approaches $n_y/2$. Within 300 seconds, both UP and DOWN algorithms can only handle problems with $n_u < 18$. For all cases, however,

the B^3 algorithm exhibits superior efficiency by combining upward and downward pruning and is able to solve problems up to $n_u = 20$ within 100 seconds.

In summary, for selection of individual measurements as CVs by minimizing L_1, all the developed algorithms (UP, DOWN and B^3) show much superior performance than the currently used brute force method. In comparison with the UP and DOWN algorithms, the B^3 algorithm shows superior performance and similar efficiency for different problem dimensions including problems with $n_u << n_y$, $n_u \approx n_y$ and $n_u \approx n_y/2$.

5.2 Distillation column case study

To demonstrate the efficiency of the developed PB^3 algorithm, we consider self-optimizing control of a binary distillation column [15]. The objective is to minimize the deviation of the distillate and bottoms composition from their nominal steady-state values in presence of disturbances in feed flow rate, feed composition and vapor fraction of feed. Two degrees of freedom (reflux and vapor boilup rates) are available and thus two CVs are required for implementation of self-optimizing control strategy. It is considered that the temperatures on 41 trays are measured with an accuracy of $\pm 0.5^o$ C. The combinatorial optimization problem involves selection of n out of 41 candidate measurements, whose combinations can be used as CVs. The reader is referred to [10] for further details of this case study.

The PB^3 algorithm is used to select the 10 best measurement combinations for every n, where n ranges from 2 to 41. The trade-off between the losses corresponding to the 10 best selections and n is shown in Figure 6(a). It can be seen that when combinations of more than 14 measurements are used as CVs, the loss is less than 0.075, which is close to the minimum loss (0.0517) seen using combinations of all 41 measurements. Furthermore, the reduction in loss is negligible, when combinations of more than 25 measurements are used. Figure 6(a) also shows that the 10 best selections have similar self-optimizing capabilities particularly when combinations of more than 5 measurements are used. Then, the designer can choose the subset of measurements among these 10 best alternatives based on some other important criteria, such as dynamic controllability.

Figure 6(b) and (c) show the computation time and number of node evaluations for PB^3 and DOWN algorithms. To facilitate the comparison further, the ratios of number of node evaluations and computation times are also shown in Figure 6(d). The PB^3 algorithm is able to reduce the number of node evaluations and hence computation time up to a factor of 20 for selection problems involving selection of a few measurements from a large candidate set. It is expected that a fully upwards pruning rule would improve

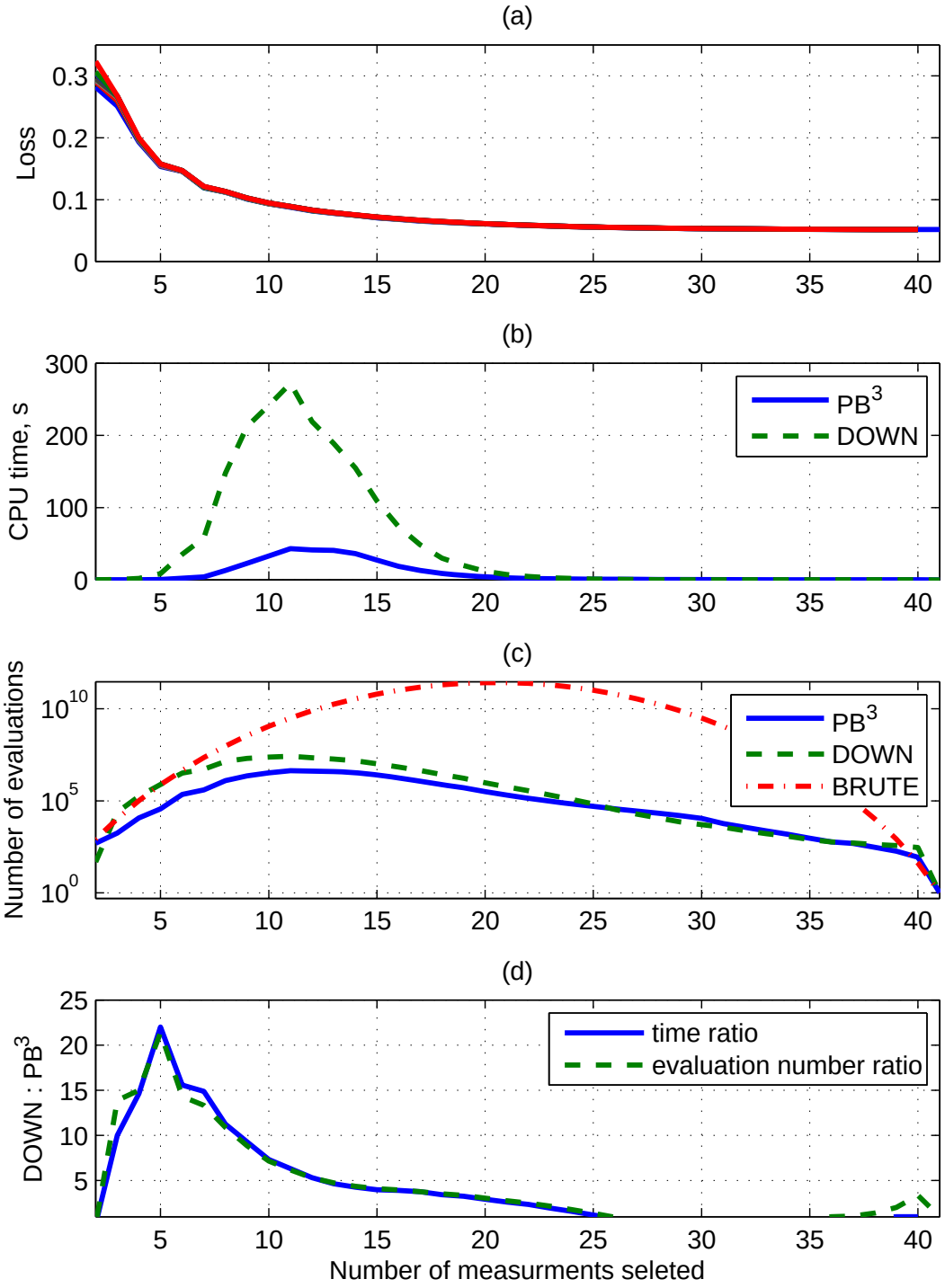

Figure 6: (a) Losses of 10-best measurement combinations against the number of measurements, (b) Comparison of computation time between PB^3 and DOWN algorithms, (c) Comparison of number of node evaluations between PB^3 and DOWN algorithms, and (d) Ratios of computation time and number of node evaluations required by PB^3 and DOWN algorithms

the efficiency even further, but the derivation of such a rule is currently an open problem.

Overall, both algorithms are very efficient and are able to reduce the number of node evaluations by 5 to 6 orders of magnitude, as compared to the brute force search method. For example, to select 20 measurements from 41 candidates, evaluation of a single alternative requires about 0.15 ms on the specified notebook computer. Thus, a brute force search methods would take more than one year to evaluate all possible alternatives. However, the proposed PB^3 and DOWN algorithms are able to solve this problem within 4 and 12 seconds, respectively. Therefore, the generation of the trade-off curve shown in Figure 6(a) would be practically impossible without the algorithms developed in the work.

6 Conclusions

In this paper, the concept of bidirectional branch and bound (BAB) proposed in Part I of this work [5] has been further developed for selection of controlled variables (CVs) using the exact local method for self-optimizing control. The numerical tests using randomly generated matrices and binary distillation column case study show that the number of evaluations for proposed algorithms is 5 to 6 orders of magnitude lower than the current practice of CV selection using brute force search.

The computationally efficiency of the algorithms developed in this paper based on bidirectional pruning and branching principles using novel determinant based criteria is similar to the BAB approach for CV selection based on minimum singular value (MSV) rule [5]. Despite the availability of the exact local criterion, one of the apparent reasons for continued use of the approximate MSV rule is its computational efficiency. This work makes CV selection using the exact local criterion computationally tractable so that it can be adopted as a standard tool for selection of CVs based on the concept of self-optimizing control.

While the algorithm for selection of individual measurements as CVs is fully bidirectional, the algorithm for selection of subset of measurements, whose combinations can be used as CVs, is only partially bidirectional. It is expected that the development of a fully bidirectional BAB algorithm for the latter problem would improve the computational efficiency further. This challenging problem is currently open and is an issue for future research. Furthermore, an extension of the bidirectional BAB algorithm to select CVs based on the minimization of local average loss for self-optimizing control [13] is currently under consideration.

Acknowledgements

The first author gratefully acknowledges the financial support from Office of Finance, Nanyang Technological University, Singapore through grant no. R42/06.

References

[1] V. Alstad. *Studies on Selection of Controlled Variables*. PhD thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2005. Available at `http://www.nt.ntnu.no/users/skoge/publications/thesis/2005_alstad/`.

[2] V. Alstad and S. Skogestad. Null space method for selecting optimal measurement combinations as controlled variables. *Ind. Eng. Chem. Res.*, 46(3):846–853, 2007.

[3] V. Alstad, S. Skogestad, and E. S. Hori. Optimal measurement combinations as controlled variables. *J. Proc. Control*, 19(1):138–148, 2009.

[4] Y. Cao and V. Kariwala. B3WC. MATLAB File Exchange, January 2009. Available at `http://www.mathworks.com/matlabcentral/fileexchange/22632`.

[5] Y. Cao and V. Kariwala. Bidirectional branch and bound for controlled variable selection: Part I. Principles and minimum singular value criterion. *Comput. Chem. Engng.*, 32(10):2306–2319, 2008.

[6] Y. Cao and P. Saha. Improved branch and bound method for control structure screening. *Chem. Engg. Sci.*, 60(6):1555–1564, 2005.

[7] Y. Cao, D. Rossiter, and D. H. Owens. Globally optimal control structure selection using branch and bound method. In *Proc. 5th International Symposium on DYCOPS*, pages 183–188, Corfu, Greece, 1998.

[8] X.-W. Chen. An improved branch and bound algorithm for feature selection. *Pattern Recognition Letters*, 24:1925–1933, 2003.

[9] I. J. Halvorsen, S. Skogestad, J. C. Morud, and V. Alstad. Optimal selection of controlled variables. *Ind. Eng. Chem. Res.*, 42(14):3273–3284, 2003.

[10] E. S. Hori and S. Skogestad. Selection of controlled variables: Maximum gain rule and combination of measurements. *Ind. Eng. Chem. Res.*, 47(23):9465–9471, 2008.

[11] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, UK, 1985.

[12] V. Kariwala. Optimal measurement combination for local self-optimizing control. *Ind. Eng. Chem. Res.*, 46(11):3629–3634, 2007.

[13] V. Kariwala, Y. Cao, and S. Janardhanan. Local self-optimizing control with average loss minimization. *Ind. Eng. Chem. Res.*, 47(4):1150–1158, 2008.

[14] P. Narendra and K. Fukunaga. A branch and bound algorithm for feature subset selection. *IEEE Trans. Comput.*, C-26:917–922, 1977.

[15] S. Skogestad. Dynamics and control of distillation columns - A tutorial introduction. *Trans. IChemE Part A*, 75:539–562, 1997.

[16] S. Skogestad. Plantwide control: The search for the self-optimizing control structure. *J. Proc. Control*, 10(5):487–507, 2000.

[17] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control: Analysis and Design*. John Wiley & sons, Chichester, UK, 1st edition, 1996.

[18] P. Somol, P. Pudil, F. Ferri, and J. Kittler. Fast branch and bound algorithm in feature selection. In B. Sanchez, J. Pineda, J. Wolfmann, Z. Bellahsense, and F. Ferri, editors, *Proceedings of World Multiconference on Systemics, Cybernetics and Informatics*, volume VII, pages 1646–651, Orlando, Florida, USA, 2000.

[19] M. Van de Wal and B. de Jager. A review of methods for input/output selection. *Automatica*, 37(4): 487–510, 2001.

[20] B. Yu and B. Yuan. A more efficient branch and bound algorithm for feature selection. *Pattern Recognition*, 26:883–889, 1993.