Bidirectional Branch and Bound Method for Selecting Controlled Variables

Vinay Kariwala [*] **Yi Cao** [**]

[*] *Division of Chemical & Biomolecular Engineering, Nanyang Technological University, Singapore 637459 (e-mail: vinay@ntu.edu.sg)*
[**] *School of Engineering, Cranfield University, Bedford, UK (e-mail: y.cao@cranfield.ac.uk)*

Abstract: Controlled variable (CV) selection from available measurements through exhaustive search is computationally forbidding for large-scale problems. We have recently proposed novel bidirectional branch and bound (B^3) approaches for CV selection using the minimum singular value (MSV) rule and the local worst-case loss criterion in the framework of self-optimizing control. However, the MSV rule is approximate and worst-case scenario may not occur frequently in practice. In this work, the B^3 approach is extended to CV selection based on the recently developed local average loss metric, which represents the expected loss incurred over the long-term operation of the plant. Lower bounds on local average loss and fast pruning algorithms are derived for the efficient B^3 algorithm. Numerical tests and binary distillation column case study are used to demonstrate the computational efficiency of the proposed method.

Keywords: Branch and bound, Combinatorial optimization, Controlled variable, Self-optimizing control.

1. INTRODUCTION

The selection of controlled variables (CVs) from available measurements is an important task during the design of control systems for complex processes. Traditionally, CVs have been selected based on intuition and process knowledge. To systematically select CVs, Skogestad (2000) introduced the concept of self-optimizing control. In this approach, CVs are selected such that in presence of disturbances, the loss incurred in implementing the operational policy by holding the selected CVs at constant setpoints is minimal, as compared to the use of an online optimizer.

The choice of CVs based on the general non-linear formulation of self-optimizing control requires solving large-dimensional non-convex optimization problems (Skogestad, 2000). To quickly pre-screen alternatives, local methods have been proposed including the minimum singular value (MSV) rule (Skogestad and Postlethwaite, 1996) and exact local methods with worst-case (Halvorsen et al., 2003) and average loss minimization (Kariwala et al., 2008). Though the local methods simplify loss evaluation for a single alternative, every feasible alternative still needs to be evaluated to find the optimal solution. As the number of alternatives grows rapidly with process dimensions, such an exhaustive search is computationally intractable for large-scale processes. Thus, an efficient method is needed to find a subset of available measurements, which can be used as CVs (Problem 1).

Instead of selecting CVs as a subset of available measurements, it is possible to obtain lower losses using combinations of available measurements as CVs (Halvorsen et al., 2003). Recently, explicit solutions to the problem of finding locally optimal measurement combinations have been proposed (Kariwala, 2007; Kariwala et al., 2008; Alstad et al., 2009). It is possible, however, that the use of combinations of a few measurements as CVs may provide similar loss as the case where combinations of all available measurements are used (Kariwala, 2007; Kariwala et al., 2008; Alstad et al., 2009). 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 (Problem 2).

Both Problems 1 and 2 can be seen as subset selection problems, for which only exhaustive search and branch and bound (BAB) method guarantee globally optimal solution. For minimization problems, a BAB approach divides the problem into several sub-problems (nodes) and calculates a lower bound of the selection criterion over all possible solutions of a node. If the lower bound is greater than an upper bound of the optimal solution, then the corresponding node is pruned (eliminated without further evaluation). In this way, the BAB method gains its efficiency in comparison with exhaustive search. The traditional BAB methods for subset selection use downwards approach, where pruning is performed on nodes with gradually decreasing subset size (Narendra and Fukunaga, 1977). Recently, a novel bidirectional BAB (B^3) approach (Cao and Kariwala, 2008) has been proposed for CV selection, where non-optimal nodes are pruned in downwards as well as upwards (gradually increasing subset size) directions simultaneously, which significantly reduces the solution time.

The bidirectional BAB (B^3) approach has been applied to solve Problem 1 with MSV rule (Cao and Kariwala, 2008) and local worst-case loss (Kariwala and Cao, 2009) as selection criteria. A partially bidirectional BAB (PB^3) method has also been proposed to solve Problem 2 through minimization of local worst-case loss (Kariwala and Cao, 2009). The MSV rule, however, is approximate and can lead to non-optimal set of CVs (Hori and Skogestad, 2008). Selection of CVs based on local worst-case loss minimization can also be conservative, as the worst-case may not occur frequently in practice (Kariwala et al., 2008). Thus, CV selection through minimization of local average loss, which represents the expected loss incurred over the long-term operation of the plant, can be deemed as most reliable.

In this paper, lower bounds on local average loss and fast pruning algorithms are derived to develop an efficient B^3 method for CV selection using the exact local method with average loss minimization. A PB^3 method is also developed to find a subset of available measurements, whose combinations can be used as CVs to minimize local average loss. Numerical tests and binary distillation column case study are used to demonstrate the computational efficiency of the proposed method.

2. BAB METHODS FOR SUBSET SELECTION

Let $X_m = \{x_1, x_2, \cdots, x_m\}$ be an m-element set. The subset selection problem with selection criterion T involves finding an n-element subset $X_n \subset X_m$ such that

$$T(X_n^*) = \min_{X_n \subset X_m} T(X_n) \qquad (1)$$

For a subset selection problem, the total number of candidates grows very quickly as m and n increase, which renders exhaustive search unviable. BAB approach can find the globally optimal subset without exhaustive search.

2.1 Unidirectional BAB approaches

Downwards. BAB search is traditionally conducted downwards (gradually decreasing subset size). A downwards solution tree for selecting 2 out of 6 elements is shown in Figure 1(a), where the root node is the same as X_m. Other nodes represent subsets obtained by eliminating one element from their parent sets. Labels at nodes denote the elements discarded there. To describe the pruning principle, let B be an upper bound of the globally optimal criterion, *i.e.* $B \geq T(X_n^*)$ and $\underline{T}_n(X_s)$ be a downwards lower bound over all n-element subsets of X_s, *i.e.* $\underline{T}_n(X_s) \leq T(X_n)\ \forall X_n \subseteq X_s$. Then,

$$T(X_n) > T(X_n^*)\ \forall X_n \subseteq X_s, \text{if } \ \underline{T}_n(X_s) > B \qquad (2)$$

Hence, any n-element subset of X_s cannot be optimal and can be pruned without further evaluation, if $\underline{T}_n(X_s) > B$.

Upwards. Subset selection can also be performed upwards (gradually increasing subset size). An upwards solution tree for selecting 2 out of 6 elements is shown in Figure 1(b), where the root node is an empty set. Other nodes represent supersets obtained by adding one element to their parent sets. Labels at nodes denote the elements added there. To introduce the pruning principle, let the upwards lower bound of the selection criterion be defined as $\underline{T}_n(X_t) \leq T(X_n)\ \forall X_n \supseteq X_t$. Then,

$$T(X_n) > T(X_n^*)\ \forall X_n \supseteq X_t, \text{if } \ \underline{T}_n(X_t) > B \qquad (3)$$

As downwards BAB, if $\underline{T}_n(X_t) > B$, any n-element superset of X_t cannot be optimal and hence can be pruned without further evaluation.

2.2 Bidirectional BAB approach

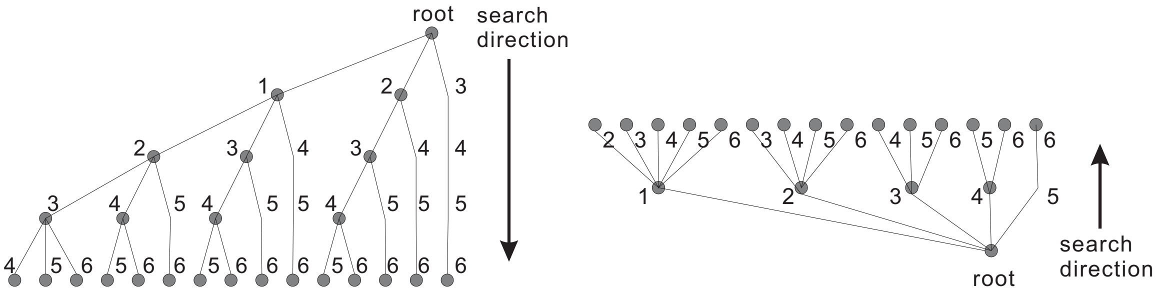
The upwards and downwards BAB approaches can be combined to form a more efficient bidirectional BAB (B^3) approach. This approach is applicable to any subset selection problem, for which both upwards and downwards lower bounds on the selection criterion are available (Cao and Kariwala, 2008).

Bidirectional pruning. In a B^3 approach, the whole subset selection problem is divided into several subproblems. A sub-problem is represented as the 2-tuple $\mathcal{S} = (F_f, C_c)$, where F_f is an f-element fixed set and C_c is a c-element candidate set. Here, $f \leq n$ and $n \leq f + c \leq m$. 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. In terms of fixed and candidate sets, downwards and upwards pruning can be performed if $\underline{T}_n(F_f \cup C_c) > B$ and $\underline{T}_n(F_f) > B$, respectively. In B^3 approach, these pruning conditions are used together (bidirectional pruning), where the subproblem $\mathcal{S}$ is pruned, if either downwards or upwards pruning condition is met.

The use of bidirectional pruning significantly improves the efficiency as non-optimal subproblems can be pruned at an early stage of the search. Further gain in efficiency is achieved by carrying out pruning on the sub-problems of $\mathcal{S}$, instead of on $\mathcal{S}$ directly. For $x_i \in C_c$, upward pruning is conducted by discarding x_i from C_c, if $\underline{T}_n(F_f \cup x_i) > B$. Similarly, if $\underline{T}(F_f \cup (C_c \backslash x_i)) > B$, then downward pruning is performed by moving x_i from C_c to F_f. Here, an advantage of performing pruning on sub-problems is that the bounds $\underline{T}_n(F_f \cup x_i)$ and $\underline{T}_n(F_f \cup (C_c \backslash x_i))$ can be computed from $\underline{T}_n(F_f)$ and $\underline{T}_n(F_f \cup C_c)$, respectively, for all $x_i \in C_c$ together, resulting in computational efficiency.

Bidirectional branching. In downwards and upwards BAB methods, branching is performed by removing elements from C_c and moving elements from C_c to F_f, respectively. These two branching approaches can be combined into an efficient bidirectional approach by selecting a decision element and deciding upon whether the decision element be eliminated from C_c or moved to F_f. In the B^3 algorithm, the decision element is selected as the one with the largest upwards or downwards upper bound for upward or downward search (best-first search), respectively.

The branching direction (upwards or downwards) is selected by comparing the number of terminal nodes (n-element subsets) of the resulting subproblems with alternate approaches such that the simpler branch is evaluated first, whilst the other branch is kept for possible pruning in future. For downwards branching, removing an element from C_c results in a subproblem with $\mathcal{C}_{c-1}^{n-f}$ terminal nodes, whilst for upwards branching, moving an element from C_c to F_f gives a subproblem with $\mathcal{C}_{c-1}^{n-f-1}$ terminal nodes. Therefore, if $2(n-f) > c$, downwards branching is performed, otherwise upwards branching is selected.



(a) downward search

(b) upward search

Fig. 1. Solution trees for selecting 2 out of 6 elements.

3. SELF-OPTIMIZING CONTROL

To present the local method for self-optimizing control, consider that the economics of the plant is characterized by the scalar objective functional $J(\mathbf{u}, \mathbf{d})$, where $\mathbf{u} \in \mathbb{R}^{n_u}$ and $\mathbf{d} \in \mathbb{R}^{n_d}$ denote the degrees of freedom or inputs and disturbances, respectively. The linearized model of the process around the nominally optimal operating point is

$$\mathbf{y} = \mathbf{G}^y\,\mathbf{u} + \mathbf{G}_d^y\,\mathbf{W}_d\,\mathbf{d} + \mathbf{W}_e\,\mathbf{e} \tag{4}$$

where $\mathbf{y} \in \mathbb{R}^{n_y}$ denotes the process measurements and $\mathbf{e} \in \mathbb{R}^{n_y}$ represents the implementation error including measurement and control errors. 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

$$\mathbf{c} = \mathbf{H}\,\mathbf{y} = \mathbf{G}\,\mathbf{u} + \mathbf{G}_d\,\mathbf{W}_d\,\mathbf{d} + \mathbf{H}\,\mathbf{W}_e\,\mathbf{e} \tag{5}$$

where $\mathbf{G}_d = \mathbf{H}\,\mathbf{G}_d^y$ and $\mathbf{G} = \mathbf{H}\,\mathbf{G}^y \in \mathbb{R}^{n_u \times n_u}$ is invertible, a necessary condition for integral control.

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

$$\left\| \left[\, \mathbf{d}^T \;\; \mathbf{e}^T \,\right] \right\|_2^T \leq 1 \tag{6}$$

Kariwala et al. (2008) have shown that the average loss over the set (6) is given as

$$L_{\text{average}}(\mathbf{H}) = \frac{1}{6(n_y + n_d)} \left\| (\mathbf{H}\tilde{\mathbf{G}})^{-1} \mathbf{H}\mathbf{Y} \right\|_F^2 \tag{7}$$

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

$$\mathbf{Y} = \left[\, (\mathbf{G}^y\,\mathbf{J}_{uu}^{-1}\,\mathbf{J}_{ud} - \mathbf{G}_d^y)\,\mathbf{W}_d \quad \mathbf{W}_e \,\right] \tag{8}$$

When individual measurements are selected as CVs, the elements of $\mathbf{H}$ are restricted to be 0 or 1 and $\mathbf{H}\mathbf{H}^T = \mathbf{I}$. Using index notation, this problem can be stated as

$$\min_{X_{n_u} \subset X_{n_y}} L_1(X_{n_u}) = \left\| \tilde{\mathbf{G}}_{X_{n_u}}^{-1}\,\mathbf{Y}_{X_{n_u}} \right\|_F^2 \tag{9}$$

Note that the scalar constant $1/(6(n_y+n_d))$ is neglected in (9), as it does not depend on the selected CVs. Instead of 2-norm, as used in (6), if a different norm is used to define the allowable set of $\mathbf{d}$ and $\mathbf{e}$, the resulting expressions for average losses only differ by scalar constants (Kariwala et al., 2008). Thus, the formulation of optimization problem in (9) is independent of the norm used to define the allowable set of $\mathbf{d}$ and $\mathbf{e}$.

Instead of using individual measurements, it is possible to use combinations of measurements as CVs. In this case, the integer constraint on $\mathbf{H} \in \mathbb{R}^{n_u \times n_y}$ is relaxed, but the condition $\operatorname{rank}(\mathbf{H}) = n_u$ is still imposed to ensure invertibility of $\mathbf{H}\,\mathbf{G}^y$. The minimal average loss over the set (6) using measurements combinations as CVs is given as (Kariwala et al., 2008)

$$\min_{\mathbf{H}} L_{\text{average}} = \frac{1}{6\,(n_y + n_d)} \sum_{i=1}^{n_u} \lambda_i^{-1} \left(\tilde{\mathbf{G}}^T \left(\mathbf{Y}\,\mathbf{Y}^T\right)^{-1} \tilde{\mathbf{G}} \right) \tag{10}$$

Equation (10) can be used to calculate the minimum loss provided by the optimal combination of a given set of measurements. However, the use of all measurements is often unnecessary and similar losses may 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 the minimal loss for specified n. In index notation, the n measurements are selected by minimizing

$$\min_{X_n \subset X_{n_y}} L_2(X_n) = \sum_{i=1}^{n_u} \lambda_i^{-1} \left(\tilde{\mathbf{G}}_{X_n}^T (\mathbf{Y}_{X_n} \mathbf{Y}_{X_n}^T)^{-1} \tilde{\mathbf{G}}_{X_n} \right) \tag{11}$$

where the scalar constant has been omitted as (9).

4. BAB METHOD FOR CV SELECTION

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

$$\mathbf{M}(X_p) = \mathbf{R}^{-T} \tilde{\mathbf{G}}_{X_p} \tilde{\mathbf{G}}_{X_p}^T \mathbf{R}^{-1} \tag{12}$$

$$\mathbf{N}(X_p) = \tilde{\mathbf{G}}_{X_p}^T (\mathbf{Y}_{X_p} \mathbf{Y}_{X_p}^T)^{-1} \tilde{\mathbf{G}}_{X_p} \tag{13}$$

where $\mathbf{R}$ is the Cholesky factor of $\mathbf{Y}_{X_p} \mathbf{Y}_{X_p}^T$.

4.1 Lower bounds

Individual measurements. L_1 in (9) requires inversion of $\mathbf{G}_{X_{n_u}}$ and thus $L_1(X_p)$ is well-defined only when $\mathbf{G}_{X_p}$ is a square matrix, *i.e.* $p = n_u$. On the other hand, BAB methods require evaluation of loss, when the number of selected measurements differs from n_u. Motivated by this drawback, two alternate representations of L_1 are derived as follows:

$$L_1(X_p) = \sum_{i=1}^{r} \lambda_i^{-1}\left(\mathbf{N}(X_p)\right) = \sum_{i=1}^{r} \lambda_i^{-1}\left(\mathbf{M}(X_p)\right) \quad (14)$$

where $r = \operatorname{rank}(\tilde{\mathbf{G}}_{X_p})$. It is clear that for $r = p = n_u$, (14) is equivalent to (9). However, (14) generally holds for any number of measurements since $\mathbf{Y}_{X_p}\mathbf{Y}_{X_p}^T$ is invertible under the reasonable assumption that every measurement has a non-zero implementation error. Using the generalized expression for L_1 and interlacing properties of eigenvalues (Horn and Johnson, 1985), the downwards and upwards lower bounds required for the application of B^3 algorithm are derived as follows.

Proposition 1. (Lower bounds for L_1). Consider a node $\mathcal{S} = (F_f, C_c)$. For L_1 defined in (14),

$$L_1(F_f) \leq \min_{X_{n_u} \supset F_f} L_1(X_{n_u});\ f < n_u \quad (15)$$

$$L_1(F_f \cup C_c) \leq \min_{X_{n_u} \subset (F_f \cup C_c)} L_1(X_{n_u});\ f + c > n_u \quad (16)$$

To illustrate the implications of Proposition 1, let B represent the best available upper bound on $L_1(X_{n_u}^*)$. Then (15) 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 $L_1(F_f \cup C_c) > B$, (16) 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 (15) and (16) and the optimal solution can be found without complete enumeration.

Measurements combinations. The expression for L_2 in (11) is the same as the expression for L_1 in (14). Thus, similar to Proposition 1, it can be shown that

$$L_2(F_f \cup C_c) \leq \min_{X_n \subset (F_f \cup C_c)} L_2(X_n); \quad f + c > n \quad (17)$$

For selecting measurements, whose combinations can be used as CVs, the result in (17) is useful for downwards pruning. Equation (16), however, also implies that when $n_u \leq f < n$, $L_2(F_f)$ decreases as the subset size increases. Thus, unlike L_1, the expression for L_2 cannot be directly used for upwards pruning. In the following proposition, a lower bound on L_2 is derived, which can instead be used for upwards pruning, whenever $n - n_u < f < n$.

Proposition 2. (Upwards lower bound for L_2). For the node $\mathcal{S} = (F_f, C_c)$, let

$$\underline{L}_2(F_f) = \sum_{i=1}^{f+n_u-n} \lambda_i^{-1}\left(\mathbf{N}(F_f)\right) \quad (18)$$

where $f > n - n_u$. 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.*

$$\underline{L}_2(F_f) \leq \min_{\substack{X_n \supset F_f \\ X_n \subset (F_f \cup C_c)}} L_2(X_n) \quad (19)$$

Proposition 2 implies that the lower bound of L_2 defined in (18) 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 (18) is referred to as partially bidirectional BAB (PB^3) algorithm. Development of fully bidirectional BAB algorithm for selection of measurement combination as CVs is an open problem.

4.2 Fast pruning and branching

Propositions 1 and 2 can be used to prune the non-optimal nodes 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 (14) and L_2 in (11) requires eigenvalue decomposition, which is computationally expensive.

Individual measurements. When $f < n_u$, $\mathbf{M}(F_f)$ in (12) is invertible. Similarly when $s = f + c > n_u$, $\mathbf{N}(S_s)$ in (13) for $S_s = F_f \cup C_c$ is invertible. Thus,

$$L_1(F_f) = \sum_{i=1}^{r} \lambda_i^{-1}(\mathbf{M}(F_f)) = \operatorname{trace}(\mathbf{M}^{-1}(F_f)) \quad (20)$$

$$L_1(S_s) = \sum_{i=1}^{r} \lambda_i^{-1}(\mathbf{N}(S_s)) = \operatorname{trace}(\mathbf{N}^{-1}(S_s)) \quad (21)$$

The use of (20) and (21) for evaluation of lower bounds on L_1 avoids computation of eigenvalues. The next two propositions relate the bounds of a node with the bounds of sub-nodes allowing pruning on sub-nodes directly and thus improving efficiency of the B^3 algorithm further.

Proposition 3. (Upwards pruning for L_1). Consider a node $\mathcal{S} = (F_f, C_c)$ and index $i \in C_c$. Then

$$L_1(F_f \cup i) = L_1(F_f) + \frac{\|\mathbf{z}_i^T \mathbf{Y}_{F_f} - \mathbf{Y}_i\|_2^2}{\eta_i} \quad (22)$$

where $\mathbf{z}_i = (\tilde{\mathbf{G}}_{F_f}\tilde{\mathbf{G}}_{F_f}^T)^{-1}\tilde{\mathbf{G}}_{F_f}\tilde{\mathbf{G}}_i^T$ and $\eta_i = \tilde{\mathbf{G}}_i(\mathbf{I} - \mathbf{G}_{F_f}^T(\tilde{\mathbf{G}}_{F_f}\tilde{\mathbf{G}}_{F_f}^T)^{-1}\tilde{\mathbf{G}}_{F_f})\tilde{\mathbf{G}}_i^T$.

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 $i \in C_c$,

$$L_1(S_s \setminus i) = L_1(S_s) + \frac{\|\mathbf{x}_i \mathbf{N}^{-1}(S_s)\|_2^2}{\zeta_i - \mathbf{x}_i \mathbf{N}^{-1}(S_s)\mathbf{x}_i^T} \quad (23)$$

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

In comparison with the direct calculation of L_1, the use of (22) and (23) is computationally less demanding. This happens as in (22), the inverse $(\tilde{\mathbf{G}}_{F_f}\tilde{\mathbf{G}}_{F_f}^T)^{-1}$ needs to be evaluated only once for all c sub-nodes, whilst in (23), two inverses $(\tilde{\mathbf{Y}}_{S_s\setminus i}\tilde{\mathbf{Y}}_{S_s\setminus i}^T)^{-1}$ and $N^{-1}(S_s)$ are evaluated only once for all c sub-nodes.

Measurements combinations. As the downwards pruning criteria for minimization of L_1 and L_2 are the same, Proposition 4 can be used for fast downwards pruning for selection of a subset of measurements, whose combinations can be used as CVs. The fast upwards pruning criteria for minimization of L_2 is presented in the next proposition.

Proposition 5. (Upwards pruning for L_2). Consider a node $\mathcal{S} = (F_f, C_c)$ and index $i \in C_c$. Then

$$\underline{L}_2(F_f \cup i) \geq \sum_{j=1}^{f+n_u-n+1} \frac{1}{\lambda_j(\mathbf{N}(F_f)) + t_j} \tag{24}$$

where $t = [t_1 \; \cdots \; t_{f+n_u-n+1}]^T$ is determined by solving the following linear equations:

$$t_j - t_{j+1} = \lambda_{j+1} - \lambda_j, \; j = 1, 2, \cdots, f + n_u - n \tag{25}$$

$$\sum_{j=1}^{f+n_u-n+1} t_j = \|\mathbf{s}_i\|_2^2 / \beta_i \tag{26}$$

with $\mathbf{s}_i = \mathbf{Y}_i \mathbf{Y}_{F_f}^T (\mathbf{Y}_{F_f} \mathbf{Y}_{F_f}^T)^{-1} \mathbf{G}_{F_f} - \mathbf{G}_i^T$ and $\beta_i = \mathbf{Y}_i(\mathbf{I} - \mathbf{Y}_{F_f}^T (\mathbf{Y}_{F_f} \mathbf{Y}_{F_f}^T)^{-1} \mathbf{Y}_{F_f}) \mathbf{Y}_i^T$.

Note that the relationship in (24) is an inequality, which can be conservative. As a BAB method spends most of its time in evaluating nodes that cannot lead to the optimal solution, we use the computationally cheaper albeit weaker pruning criteria in this paper.

5. NUMERICAL EXAMPLES

To examine the efficiency of the proposed BAB algorithms developed in this work and listed in Table 1, numerical tests are conducted using randomly generated matrices and binary distillation column case study. 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. BAB programs for comparison

program	description
UP	upwards pruning (22)
DOWN	downwards pruning (23)
B^3	bidirectional BAB by combining (22) and (23)
PB^3	partially B^3 by combining (23) and (24)

5.1 *Random tests*

To evaluate the efficiency of the different BAB algorithms developed in this work, we consider selection of n_u out of $n_y = 36$ variables, where n_u varies between 1 and 35 with $n_d = 5$. 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 average computation time and number of nodes evaluated over the 100 random cases are summarized in Figure 2.

From Figure 2, it can be seen that all the developed algorithms (UP, DOWN and B^3) show much superior performance than the currently used brute force method. As one may expect, upwards pruning based algorithm (UP) shows better efficiency for problems involving selection of a few variables 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 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. Within 1000 seconds, both UP and DOWN algorithms can only handle problems with $n_u < 9$ or $n_y - n_u < 9$. For all cases, however, the B^3 algorithm exhibits superior efficiency and is able to solve the problem with $n_u = 18$ within 200 seconds.

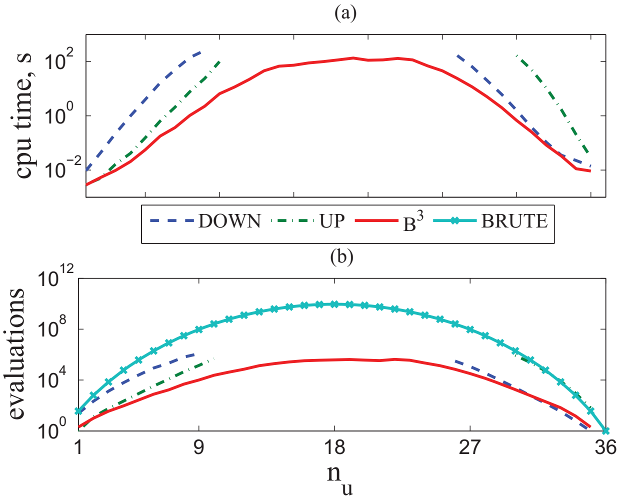


Fig. 2. Random test: (a) computation time and (b) number of nodes evaluated against n_u.

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 (Skogestad, 1997). 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 Hori and Skogestad (2008) for further details of this case study.

The PB^3 algorithm is used to select the 10 best measurement combinations for $2 \leq n \leq 41$. The trade-off between the losses of the 10 best selections and n is shown in Figure 3(a). It can be seen that when $n \geq 14$, the loss is less than 0.075, which is close to the minimum loss (0.052) by using a combination of all 41 measurements. Furthermore, the reduction in loss is negligible, when combinations of more than 20 measurements are used.

Figures 3(b) and (c) show the computation time and number of node evaluations for PB^3 and DOWN algorithms. 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, both PB^3 and DOWN algorithms are able to solve this problem within 100 seconds. Hence, without algorithms developed here, it

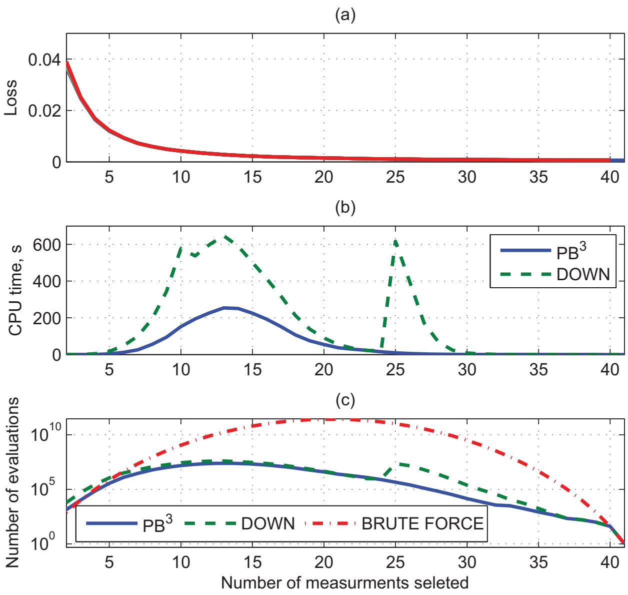


Fig. 3. (a) Average losses of 10-best measurement combinations against the number of measurements, (b) Comparison of computation time, and (c) Comparison of number of node evaluations

would be practically impossible to generate of the trade-off curve shown in Figure 3(a).

Due to the conservativeness of the pruning condition (24), the PB^3 algorithm is only able to reduce the number of node evaluations and hence computation time up to a factor of 2 for selection problems involving selection of a few measurements from a large candidate set. It is expected that a less conservative or fully upwards pruning rule would improve the efficiency, but the derivation of such a rule is currently an open problem.

6. CONCLUSIONS

In this paper, the concept of bidirectional branch and bound (BAB) proposed in Cao and Kariwala (2008) has been further developed for selection of controlled variables (CVs) using the local average loss minimization criterion for self-optimizing control (Kariwala et al., 2008). The numerical tests using randomly generated matrices and binary distillation column case study show that the number of evaluations for proposed algorithms is 4 to 5 orders of magnitude lower than the current practice of CV selection using brute force search.

The computational efficiency of the algorithms developed in this paper based on bidirectional pruning and branching principles and fast pruning algorithms is compatible to the BAB approach for CV selection based on minimum singular value (MSV) rule (Cao and Kariwala, 2008) and the local worst-case criterion (Kariwala and Cao, 2009). Despite the availability of the exact local criteria (the worst case and average loss), one of the apparent reasons for continued use of the approximate MSV rule is its computational efficiency. This work makes CV selection using the local average loss criterion computationally tractable so that it can be adopted as a standard tool for CV selection in the self-optimizing control framework. 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. Furthermore, the combination matrix H that minimizes average loss also minimizes worst-case loss (Kariwala et al., 2008). This super-optimality, however, only holds for a given subset of measurements and in general, different measurement subsets can be optimal for these two criteria. An extension of the bidirectional BAB algorithm to select CVs based on the bi-objective minimization of local worst-case and average losses for self-optimizing control is currently under consideration.

ACKNOWLEDGEMENTS

The first author gratefully acknowledges the financial support from Office of Finance, Nanyang Technological University, Singapore through grant no. RG42/06.

REFERENCES

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

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

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

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

Horn, R.A. and Johnson, C.R. (1985). *Matrix Analysis*. Cambridge University Press, Cambridge, UK.

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

Kariwala, V. and Cao, Y. (2009). Bidirectional branch and bound for controlled variable selection: Part II. Exact local method for self-optimizing control. *Comput. Chem. Eng.*, Accepted for publication.

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

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

Skogestad, S. (1997). Dynamics and control of distillation columns - A tutorial introduction. *Trans. IChemE Part A*, 75, 539–562.

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

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