



Approximate multi-parametric programming based B&B algorithm for MINLPs

Taoufiq Gueddar, Vivek Dua*

Centre for Process Systems Engineering, Department of Chemical Engineering, University College London, Torrington Place, London WC1E 7JE, United Kingdom

ARTICLE INFO

Article history:

Received 5 October 2011

Received in revised form 5 March 2012

Accepted 8 March 2012

Available online 22 March 2012

Keywords:

Parametric programming

Branch & Bound

MINLP

ABSTRACT

In this work an improved B&B algorithm for MINLPs is proposed. The basic idea of the proposed algorithm is to treat binary variables as parameters and obtain the solution of the resulting multi-parametric NLP (mp-NLP) as a function of the binary variables, relaxed as continuous variables, at the root node of the search tree. It is recognized that solving the mp-NLP at the root node can be more computationally expensive than exhaustively enumerating all the terminal nodes of the tree. Therefore, only a local approximate parametric solution, and not a complete map of the parametric solution, is obtained and it is then used to guide the search in the tree.

© 2012 Elsevier Ltd. All rights reserved.

1. Background and problem formulation

The process synthesis area remains a very important subject within chemical process design and optimization research field. The need for optimizing plant configurations and flowsheet structures is even more critical now with constantly tightening market conditions and commercial specifications for the products. There is also more concern for environmental issues and for the ability and the flexibility of producing a wide range of products. Special classes of process synthesis problems include: heat recovery systems, heat and mass exchangers network synthesis, utilities systems and separation processes. The overall process synthesis problem aims to find the best configuration (selection of units and interaction between the different blocks of the process flow sheet) that allows one to transform the raw materials into the desired products whilst meeting the specified performances criteria of maximum profit, minimum operating cost, energy efficiency and good operability with respect to flexibility, controllability, reliability, safety and environmental regulations. Process synthesis problems can be modeled as a Mixed Integer Non-linear Programming (MINLP) problem (Grossmann, 1996; Grossmann & Daichendt, 1996). The nonlinear terms in the MINLP are usually due to the non-linear nature of the chemical processes and energy networks, where the integer variables can represent existence or not of a process unit or a heat exchanger, existence of trays in a distillation tower, routing possibilities of a by-product to a final blend, etc. These problems

can also be formulated as MINLP problems (Kallrath, 2000). Stability analysis of nonlinear model predictive control problems (Dua, 2006) and optimal configuration of artificial neural network problems (Dua, 2010) can also be formulated as MINLPs.

In this work an improved Branch & Bound (B&B) algorithm for MINLPs is presented and its performance is analyzed. The algorithm is based upon the fundamentals of parametric programming. Considering the integer variables as parameters and solving at the root node, the objective function, Lagrangian function, and the continuous variables are approximated as a function of the integer variables. Using the multi-parametric programming framework, the approximate objective and Lagrangian functions are then evaluated at the terminal nodes and used to guide the search in the tree. The proposed approach improves the computational time compared to standard traditional Branch & Bound algorithm through reduction of number of nodes explored in the search tree. The details of the algorithm are discussed next.

1.1. Mixed-Integer Nonlinear Programming

Consider the following Mixed-Integer Nonlinear Program (MINLP), **problem P1**:

$$\begin{aligned} z_1 &= \min_{\mathbf{x}, \mathbf{y}} f(\mathbf{x}, \mathbf{y}) \\ \text{subject to: } & \mathbf{h}(\mathbf{x}, \mathbf{y}) = 0 \\ & \mathbf{g}(\mathbf{x}, \mathbf{y}) \leq 0 \\ & \mathbf{x} \in \mathbb{R}^{n_x} \\ & \mathbf{y} \in \{0, 1\}^{n_y} \end{aligned}$$

In this formulation \mathbf{x} is a vector of continuous variables, \mathbf{y} is a vector of binary variables, \mathbf{h} is a vector of equality constraints, \mathbf{g} is a

* Corresponding author. Tel.: +44 020 7679 0002; fax: +44 020 7383 2348.
E-mail address: v.dua@ucl.ac.uk (V. Dua).

Table 1
mp-LP solution of the MILP.

i	CR ⁱ	Optimal solution given by $z(y_1, y_2)$ and $\mathbf{x}(y_1, y_2)$
1	$-827.59 y_1 + 2103.45 y_2 \leq 896.55$ $0 \leq y_1 \leq 1$ $0 \leq y_2$	$z(y_1, y_2) = 27931.04 y_1 + 43758.63 y_2 + 286758.6$ $x_1(y_1, y_2) = 10344.83 y_1 - 3793.10 y_2 + 26206.897$ $x_2(y_1, y_2) = -5172.41 y_1 + 6896.55 y_2 + 6896.552$
2	$-827.59 y_1 + 2103.45 y_2 \geq 896.55$ $0 \leq y_1 \leq 1$ $y_2 \leq 1$	$z(y_1, y_2) = 45147.55 y_1 + 305409.86$ $x_1(y_1, y_2) = 8852.459 y_1 + 24590.164$ $x_2(y_1, y_2) = -2459.016 y_1 + 9836.065$

vector of inequality constraints and f is the scalar objective function. Typically, the mass and energy balances are given by \mathbf{h} and process specification by \mathbf{g} .

The deterministic algorithms for MINLPs can be broadly classified as those based upon decomposition principles and Branch & Bound (B&B) techniques (Bonami et al., 2008; Floudas, 1995). Two most commonly used decomposition algorithms are based upon Generalized Benders Decomposition (GBD) (Geoffrion, 1972) and Outer Approximation (OA) (Duran & Grossmann, 1986). In the GBD and the OA algorithms a sequence of iterating primal and master sub-problems is constructed that converges in a finite number of iterations. The sequence of primal sub-problems represents non-increasing upper bounds and the sequence of master sub-problems represents non-decreasing lower bounds. The primal sub-problem is formulated by fixing the binary variables resulting in a nonlinear program (NLP). The main difference between the GBD and the OA algorithm is in the formulation of the master sub-problem. In the GBD algorithm the master sub-problem is based upon duality theory whereas in the OA algorithm the master sub-problem is obtained by linearizing the constraints and the objective function. In the GBD and the OA algorithms the master sub-problems is formulated as a mixed-integer linear program (MILP). It has been shown that the lower bound generated by the OA algorithm is greater than or equal to that generated by the GBD algorithm. The OA algorithm therefore takes fewer iterations than the GBD to converge and has been successfully applied to several process and product design problems. Decomposition algorithm based upon simplicial approximation (Goyal & Ierapetritou, 2004) and cutting plane methods requiring repetitive solution of MILPs have also been presented (Westerlund & Pettersson, 1995).

B&B algorithms are based upon a systematic tree search methodology (Borchers & Mitchell, 1994; Gupta & Ravindran, 1985). At the root node of the tree all the binary variables are relaxed as continuous variables, at the terminal nodes all the binary variables are fixed and at the intermediate nodes some of the binary variables are fixed and the remaining ones are relaxed. The problem at each node of the tree corresponds to an NLP; the solution at the root node represents a lower bound and at a terminal node represents an upper bound on the solution. The efficiency of the B&B algorithms depends upon enumerating as few nodes as possible. The decision of whether to enumerate or fathom a node depends upon the solution obtained at its predecessor node and the best upper bound that is available.

1.2. Multi-parametric programming

Consider the following multi-parametric Nonlinear Programming (mp-NLP) problem (Pistikopoulos, 2009; Pistikopoulos, Georgiadis, & Dua, 2007a, 2007b), **problem P2**:

$$z_2(\theta) = \min_{\mathbf{x}} f(\mathbf{x}, \theta)$$

subject to : $\mathbf{h}(\mathbf{x}, \theta) = 0$
 $\mathbf{g}(\mathbf{x}, \theta) \leq 0$
 $\mathbf{x} \in \mathbb{R}^{n_x}$
 $\theta \in \mathbb{R}^{n_\theta}$

Parametric programming provides \mathbf{x}^* , the optimal value of \mathbf{x} , as a set of explicit functions of θ without exhaustively enumerating the entire space of θ , the regions where these explicit functions are valid are known as critical regions (CRs). For the solution of the mp-NLPs, the nonlinear terms are outer-approximated and multi-parametric linear program (mp-LP) is formulated and solved. The points in the space of θ where the difference between the solution of the NLP and the mp-LP is maximum are identified and at those points mp-LPs are formulated and solved. This procedure is repeated until this difference is within a certain tolerance (Dua & Pistikopoulos, 1999).

In the next section a new B&B algorithm for solving **P1** based upon parametric programming is presented, in Section 3 illustrative examples are presented and concluding remarks are presented in Section 4.

2. Multi-parametric programming based B&B Algorithm for MINLPs

In this work the MINLP (problem P1) is reformulated as an mp-NLP (problem P2) by relaxing the binary variables, \mathbf{y} , as continuous variables bounded between 0 and 1 and treating \mathbf{y} as parameters, **problem P3**:

$$z_3(\mathbf{y}) = \min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y})$$

subject to : $\mathbf{h}(\mathbf{x}, \mathbf{y}) = 0$
 $\mathbf{g}(\mathbf{x}, \mathbf{y}) \leq 0$
 $\mathbf{x} \in \mathbb{R}^{n_x}$
 $\mathbf{y} \in [0, 1]^{n_y}$

The solution of problem P3 provides the objective function, z_3 , and the continuous variables, \mathbf{x} , as a function of \mathbf{y} given by $z_3(\mathbf{y})$ and $\mathbf{x}(\mathbf{y})$ respectively. The optimal solution can then be obtained by fixing all the possible combinations of \mathbf{y} and evaluating $z_3(\mathbf{y})$ through simple function evaluations at those fixed values and then selecting the best solution.

2.1. Motivating example

Consider the following MILP, formulated as an mp-LP:

$$z(y_1, y_2) = \max_{\mathbf{x}} 8.1x_1 + 10.8x_2$$

subject to : $0.80x_1 + 0.44x_2 \leq 24000 + 6000y_1$
 $0.05x_1 + 0.10x_2 \leq 2000 + 500y_2$
 $0.10x_1 + 0.36x_2 \leq 6000$
 $0 \leq y_1, y_2 \leq 1$ (1)

The solution of this problem is given in Table 1. Evaluating $z(y_1, y_2)$ by fixing y_1 and y_2 at the binary values gives $z(0,0) = 2.87E5$, $z(0,1) = 3.05E5$, $z(1,0) = 3.15E5$, $z(1,1) = 3.51E5$. The optimal solution of the MILP is therefore given by $z = 3.51E5$, $y_1 = 1$, $y_2 = 1$, $x_1 = 3.34E4$ and $x_2 = 7.38E3$.

This approach in general can be more computationally expensive than exhaustively solving the MILP or MINLP for all the possible fixed values of \mathbf{y} . In the next section an algorithm based upon an approximate parametric programming, which estimates the

solution at the terminal nodes of the B&B tree from the solution at the root node and intermediate nodes is presented.

2.2. Approximate parametric programming

In this work a complete parametric solution profile of problem P3 is not obtained and instead an approximate parametric solution of problem P3 is obtained. These approximations are obtained by solving the NLP in P3 for a certain value of \mathbf{y} , at the root node of the B&B tree, and are given by explicit functions of \mathbf{y} . Evaluating these approximate solutions at the terminal nodes of the B&B tree provides an estimate of the solution of the original MINLP for fixed binary values of \mathbf{y} . These estimated solutions are then ranked, based upon whether the solution is feasible as well as the values of the estimates. This ranking is then used to guide the search in the B&B tree – so as to make decisions which nodes to fathom and which ones to branch on.

The NLP problem for fixed integer variable $\mathbf{y} = \mathbf{y}^k$ is formulated as **problem P4**:

$$\begin{aligned} z_{UB}^k &= \min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}^k) \\ \text{subject to: } & \mathbf{h}(\mathbf{x}, \mathbf{y}^k) = 0 \\ & \mathbf{g}(\mathbf{x}, \mathbf{y}^k) \leq 0 \\ & \mathbf{x} \in \mathcal{R}^{n_x} \end{aligned}$$

The solution of this problem provides an upper bound on the solution of P1. Assuming that f , \mathbf{g} and \mathbf{h} are twice continuously differentiable in \mathbf{x} the first-order KKT conditions for problem P4 are given as follows:

$$\begin{aligned} L(\mathbf{x}, \mathbf{y}^k) &= \nabla f(\mathbf{x}, \mathbf{y}^k) + \sum_{i=1}^p \lambda_i \nabla g_i(\mathbf{x}, \mathbf{y}^k) + \sum_{j=1}^q \mu_j \nabla h_j(\mathbf{x}, \mathbf{y}^k) \\ \lambda_i \nabla g_i(\mathbf{x}, \mathbf{y}^k) &= 0, \lambda_i \geq 0, \forall i = 1, \dots, p \\ h_j(\mathbf{x}, \mathbf{y}^k) &= 0, \forall j = 1, \dots, q \\ \nabla L(\mathbf{x}, \mathbf{y}^k) &= 0 \end{aligned} \tag{2}$$

In (2) L is the Lagrangian function, λ_i the Lagrangian multipliers for the inequality constraints and μ_j the Lagrangian multipliers for the equality constraints.

Theorem 1. Basic Sensitivity Theorem (Fiacco, 1976; Jackson & McCormick, 1988). Let y_0 be a vector of binary variables considered as parameter values and $(\mathbf{x}_0, \boldsymbol{\lambda}_0, \boldsymbol{\mu}_0)$ a KKT triple where $\boldsymbol{\lambda}_0$ is non-negative and \mathbf{x}_0 is feasible. Also assume that: (i) strict complementary slackness (SCS) holds; (ii) the binding constraint gradients are linearly independent (LICQ: Linear Independence Constraint Qualification); and (iii) the second-order sufficiency conditions (SOSC) hold. Then, in neighbourhood of y_0 , there exists a unique, once continuously differentiable function $[\mathbf{x}(\mathbf{y}), \boldsymbol{\lambda}(\mathbf{y}), \boldsymbol{\mu}(\mathbf{y})]$ satisfying the SOSC of problem P4 with $[\mathbf{x}(\mathbf{y}_0), \boldsymbol{\lambda}(\mathbf{y}_0), \boldsymbol{\mu}(\mathbf{y}_0)] = (\mathbf{x}_0, \boldsymbol{\lambda}_0, \boldsymbol{\mu}_0)$, where $\mathbf{x}(\mathbf{y})$ is a unique isolated minimizer for P4 and:

$$\mathbf{U} = -(\mathbf{M}_0)^{-1} \mathbf{N}_0 \tag{3}$$

$$\mathbf{U} = \begin{pmatrix} \frac{d\mathbf{x}(\mathbf{y}_0)}{d\mathbf{y}} \\ \frac{d\boldsymbol{\lambda}(\mathbf{y}_0)}{d\mathbf{y}} \\ \frac{d\boldsymbol{\mu}(\mathbf{y}_0)}{d\mathbf{y}} \end{pmatrix} \tag{4}$$

$$\mathbf{M}_0 = \begin{pmatrix} \nabla^2 L & \nabla g_1 & \dots & \nabla g_p & \nabla h_1 & \dots & \nabla h_q \\ -\lambda_1 \nabla^T g_1 & -g_1 & & & & & \\ \vdots & & \ddots & & & & \\ -\lambda_p \nabla^T g_p & & & -g_p & & & \\ \nabla^T h_1 & & & & & & \\ \vdots & & & & & & \\ \nabla^T h_q & & & & & & \end{pmatrix} \tag{5}$$

where

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \mathbf{y}) = f(\mathbf{x}, \mathbf{y}) + \sum_{i=1}^p \lambda_i g_i(\mathbf{x}, \mathbf{y}) + \sum_{j=1}^q \mu_j h_j(\mathbf{x}, \mathbf{y}) \tag{6}$$

$$\mathbf{N}_0 = (\nabla_{y_1}^2 L \cdot -\lambda_1 \nabla_y^T g_1, \dots, -\lambda_p \nabla_y^T g_p, \nabla_y^T h_1, \dots, \nabla_y^T h_q)^T$$

Under the assumption of Theorem 1, the first order Taylor–Lagrange expansion of $[\mathbf{x}(\mathbf{y}), \boldsymbol{\lambda}(\mathbf{y}), \boldsymbol{\mu}(\mathbf{y})]^T$ in a neighborhood of y_0 can be stated as:

$$\begin{bmatrix} \mathbf{x}(\mathbf{y}) \\ \boldsymbol{\lambda}(\mathbf{y}) \\ \boldsymbol{\mu}(\mathbf{y}) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_0 \\ \boldsymbol{\lambda}_0 \\ \boldsymbol{\mu}_0 \end{bmatrix} - (\mathbf{M}_0)^{-1} \cdot \mathbf{N}_0 \cdot (\mathbf{y} - y_0) + o(\|\mathbf{y}\|) \tag{7}$$

2.3. Improved B&B algorithm for MINLPs

Based upon the theory described in the previous section, the steps of the proposed algorithm are presented as follows:

Step 1: Initialize the upper and lower bounds of the MINLP problem ($LB = -\infty, UB = +\infty$), then relax the integer variables and solve the corresponding NLP problem (P3). The integer vector will be considered as a vector of continuous parameters belonging to the [0,1] interval. If the NLP relaxation subproblem is infeasible then the problem is infeasible as well. If the NLP relaxation solution is integer the algorithm will terminate and the solution of this subproblem will be the solution of the initial MINLP problem (P1). If the solution is fractional (either all or some of the relaxed integer variables are in the [0,1] interval) then further branching will take place. The lower bound is updated $Z_{LB} = Z^*$, where Z^* is the optimal objective function of the relaxed MINLP problem at the root node.

Step 2: Generate the terminal nodes matrix $\boldsymbol{\Omega}(i,j)$, the dimensions of this matrix are $\mathbf{n} \times 2^n$, \mathbf{n} being the dimension of the integer vector and 2^n the number of integer enumerations of the tree. Each column will represent a potential integer solution vector of the MINLP problem (P1). When an NLP subproblem relaxation leads to a fractional solution inside the tree then the dimensions of the terminal nodes matrix below this node will be $\mathbf{n} - k$ and $2^{\mathbf{n} - k}$, where k is the level where the node is located in the tree. For instance $k = 0$ corresponds to the root node and $k = n$ to the terminal nodes.

For a given MINLP problem with 3 integer variables the terminal nodes matrix at the root node can be presented as:

$$\boldsymbol{\Omega}_{\mathbf{n} \times 2^n} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Step 3: The integer vector is considered now as a parameter as stated in problem P3. Obtain $[\mathbf{x}(\mathbf{y}), \boldsymbol{\lambda}(\mathbf{y}), \boldsymbol{\mu}(\mathbf{y})]$ by using Eq. (7).

The multiparametric approximations will be performed every time when no fathoming actions can be taken in the node. The linear sensitivity system’s dimensions depend on the number of fractional integer variables. In other words the $\mathbf{U}(\mathbf{y})$ vector will have as many columns as fractional integer variables after solving a relaxed NLP subproblem (sensitivities for the integer variables that are already integer are not required)

Step 4: Using the estimated values of the variables from Eq. (7), the values of the inequality and equality constraints and therefore

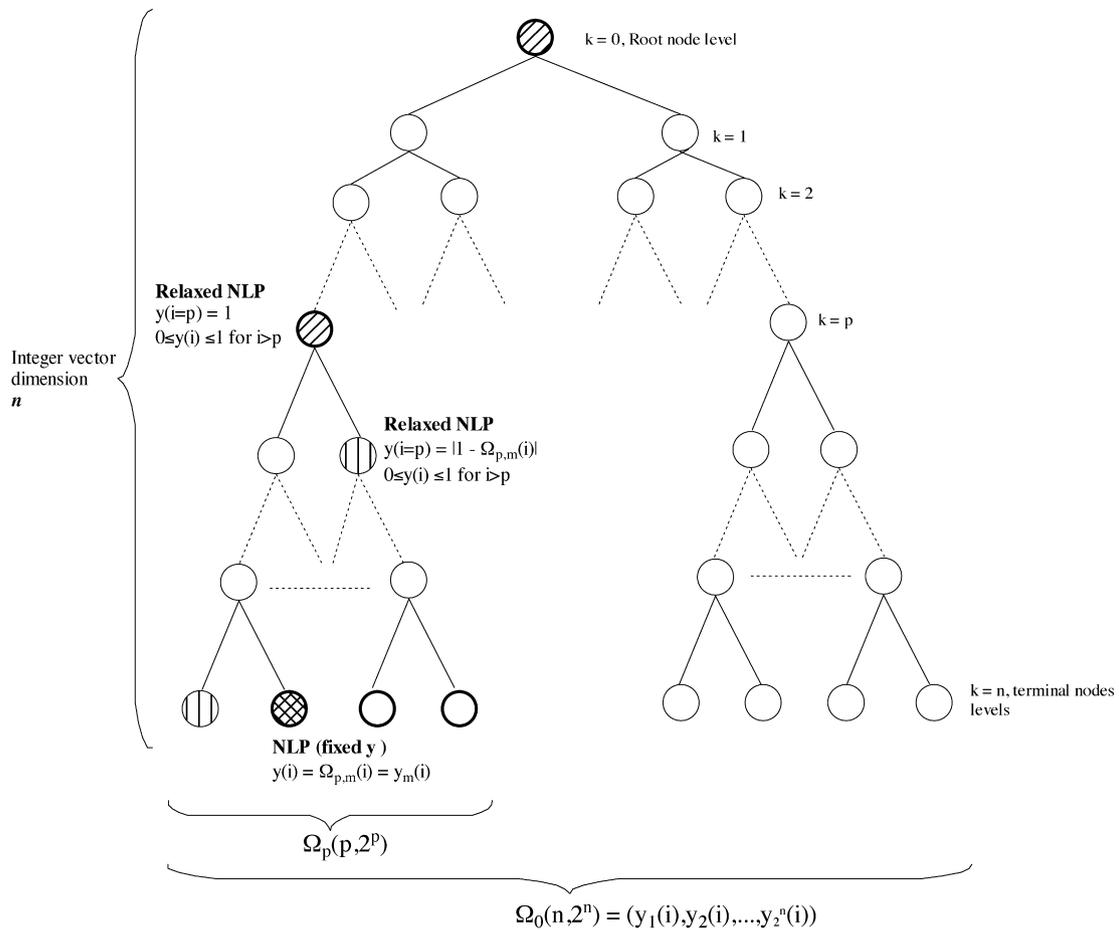


Fig. 1. Structure of the tree.

the Lagrangian function values for the different terminal nodes are estimated as follows:

$$L(\mathbf{x}(\mathbf{y}), \boldsymbol{\lambda}(\mathbf{y}), \boldsymbol{\mu}(\mathbf{y}), \mathbf{y}) = f(\mathbf{x}(\mathbf{y}), \mathbf{y}) + \sum_{i=1}^p \lambda_i(\mathbf{y})g_i(\mathbf{x}(\mathbf{y}), \mathbf{y}) + \sum_{j=1}^q \mu_j(\mathbf{y})h_j(\mathbf{x}(\mathbf{y}), \mathbf{y}) \quad (8)$$

Step 5: The idea is to transform the constrained problem P3 into an unconstrained problem (Fiacco, 1983; Floudas, 1995). Such transformation involves the use of the Lagrangian function:

$$L(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}, \mathbf{y}) + \sum_{i=1}^p \lambda_i g_i(\mathbf{x}, \mathbf{y}) + \sum_{j=1}^q \mu_j h_j(\mathbf{x}, \mathbf{y}) \quad (9)$$

The transformed unconstrained problem then becomes finding the stationary points of the Lagrange function.

$$\min_{\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}, \boldsymbol{\mu} \geq 0} L(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}, \mathbf{y}) + \boldsymbol{\lambda}^T \mathbf{g}(\mathbf{x}, \mathbf{y}) + \boldsymbol{\mu}^T \mathbf{h}(\mathbf{x}, \mathbf{y}) \quad (10)$$

Therefore, after calculation of the Lagrangians function values at the terminal nodes, they are ranked (from minimum to maximum value), the best candidate is selected corresponding to the minimum Lagrangian function value. The integer variable values corresponding to the selected candidate are fixed and the NLP subproblem is then solved (P4). If the NLP is feasible it will provide an upper bound to the tree, if not, the next candidates will be selected

successively until a feasible solution is reached. This node will be used to direct the branching in the tree.

Step 6: In general, when the relaxed NLP subproblem is solved at tree level k and at the 2^{n-k} terminal nodes the Lagrangian functions are estimated and if $\Omega_{k,m}(i) = y_m(i)$ is the best candidate integer vector it is solved as an NLP (as in Step 5) then the next step is to check its optimality. This is achieved through exploring the complementary sections of the tree as shown in Fig. 1. In this figure at level $k = p$ a relaxed NLP problem is solved and the Lagrange function estimations of the terminal nodes are derived from the relaxed node. To check the optimality of the Lagrangian candidate the branching step takes place. Explicit enumeration of all tree nodes is practically impossible due to the exponentially increasing number of possible solutions. The use of bounds for the objective function to be minimized combined with the value of the current best solution enables the algorithm to search sections of the solution space only implicitly. For instance from the original relaxed NLP a constraint ($y_{(i=p)} = |1 - \Omega_{p,m}(i)|$) is added to perform the branching.

The branched nodes fall into a class of problems that can be formulated as Problems $P_k(p)$ (p is a tree level between k and n):

$$\begin{aligned} z_p(\mathbf{y}) &= \min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}) \\ \text{subject to : } & \mathbf{h}(\mathbf{x}, \mathbf{y}) = 0 \\ & \mathbf{g}(\mathbf{x}, \mathbf{y}) \leq 0 \\ & \mathbf{x} \in \mathbb{N}^{n_x} \\ & \mathbf{y}(i) \in [0, 1]^{n_y} \text{ for } i > p \\ & \mathbf{y}(i) = |1 - \Omega_{k,m}(i)| \text{ for } i = p \\ & \mathbf{y}(i) = \Omega_{k,m}(i) \text{ for } i < p \end{aligned} \quad (11)$$

We can clearly verify that the sub-problems $\{P_k(p), k \leq p \leq n\}$ are a subdivision of the initial solution space of problem P1 as they satisfy the following conditions:

- A feasible solution of any of the sub-problems $\{P_k(p), k \leq p \leq n\}$ is a feasible solution of (P1)
- Every feasible solution of (P1) is a feasible solution of exactly one of the sub-problems.

This implies that the sub problems $P_k(p)$ are disjoint, hence the same feasible solution appearing in different subspaces of the search tree is avoided.

Step 7: when all the tree nodes are explored the algorithm terminates. Figs. 1 and 2 provide more in-depth information to help understand the algorithm better.

The detailed steps of the proposed improved B&B algorithm outlined in this section can be summarized in the box below:

Algorithm statement

Step 1: Solve (P3) by treating y as free variable in the [0,1] interval

Step 2: Generate terminal nodes matrix. Each column will represent a potential integer solution vector of the MINLP problem

Step 3: Obtain $[\mathbf{x}(y), \lambda(y), \mu(y)]$ from Eq. (7)

Step 4: Evaluate $L(\mathbf{x}(y), \lambda, \mu)$ at the terminal nodes by using Eq. (8)

Step 5: Rank the Lagrangian functions, select the best candidate integer solution and for this candidate solve the NLP (P4). If the selected candidate is infeasible then go to the next candidate.

Step 6: Solve subset nodes for the relaxed integer vector and fixing the branching variables (11) and apply fathoming when criteria below apply.

Criteria 1: The optimal value of the subproblem is $\geq Z^*$ (Z^* is the best integer solution found)

Criteria 2: The NLP subproblem is infeasible

Criteria 3: The optimal solution of the NLP subproblem is integer

If a node is fathomed then select new node. If a node cannot be fathomed yet go to step 3 and repeat step 3–5.

Step 7: program stops when the whole tree space is explored

Examples from the literature where this approximate parametric programming approach has been applied are presented next to demonstrate the main ideas of the approach.

3. Illustrative examples

3.1. Example 1: linear binary variable

This example is relatively simple but helps to demonstrate the basic concepts of the proposed algorithm. The problem has one binary variable and one continuous variable and is described by the formulation below:

$$z = \min_x -2.7y + x^2$$

subject to :

$$1) -\ln(1+x) + y \leq 0$$

$$2) -\ln(x-0.57) + y - 1.1 \leq 0$$

$$3) x \leq 2$$

$$4) -x \leq 0$$

$$y \in \{0, 1\}$$

The NLP relaxation is solved to get $z_0 = -0.479$, $x_0 = 1.35$, $y_0 = 0.856$, $\lambda^*_1 = 0.869$, $\lambda^*_2 = 1.831$, $\lambda^*_3 = 0.000$ and $\lambda^*_4 = 0.000$.

The integer variable is fixed to the relaxed value $y = y_0$ and solved to obtain the sensitivities. Theorem 1 is used to obtain the parametric sensitivities of the continuous variables and the Lagrangian multipliers

$$x(y + \Delta y) = x^* + \frac{\partial x}{\partial y} \Delta y \quad \text{and} \quad \lambda(y + \Delta y) = \lambda^* + \frac{\partial \lambda}{\partial y} \Delta y$$

$$\frac{\partial x}{\partial y} = 2.353, \quad \frac{\partial \lambda_1}{\partial y} = 18.449, \quad \frac{\partial \lambda_2}{\partial y} = 3.336, \quad \frac{\partial \lambda_3}{\partial y} = 0, \quad \frac{\partial \lambda_4}{\partial y} = 0$$

$$x(y) = x_0 + 2.353(y - y_0)$$

$$\lambda_1 = \lambda_{1,0} + 18.449(y - y_0)$$

$$\lambda_2 = \lambda_{2,0} + 3.336(y - y_0)$$

$$\lambda_3 = \lambda_{3,0} = 0$$

$$\lambda_4 = \lambda_{4,0} = 0$$

The Lagrangian function is constructed using the approximate parametric functions for x and λ

$$\begin{aligned} L(y) = & -0.479 - 0.58(y - 0.856) + (0.869 + 18.449(y - 0.856)) \\ & \times (\ln(2.35 + 2.353(y - 0.856)) + y) \\ & + (1.83 + 3.336(y - 0.856))(-\ln(0.78 + 2.353(y - 0.856)) \\ & + y - 1.1) \end{aligned}$$

The Lagrangians are estimated for the integer solutions candidates (no optimization problem is solved, it is a function evaluation only). $L(y = 1) = -1.028$ where $L(y = 0)$ is undefined because of unacceptable values for the log function $L(y = 1) = -1.056$, $L(y = 0) = \text{UNDEF}$. The best Lagrangian was obtained for the candidate ($y = 1$), an NLP problem is then solved by fixing $y = 1$ and the optimal solution is $Z^* = 0.2525$, $x^* = 1.72$, $y^* = 1$.

The branching procedure is relatively simple in this case and consists of solving the problem for $y = 0$ in order to cover the whole feasibility space. For $y = 0$, $Z^* = 0.815 > 0.2525$. Therefore this solution can be pruned.

Finally the solution to this problem is $Z^* = 0.2525$, $x^* = 1.72$, $y^* = 1$.

3.2. Example 2: quadratic binary variable

$$z = \min_x -2.7y^2 + x^2$$

subject to :

$$-\ln(1+x) + y^2 \leq 0$$

$$\ln(x-0.57) + y^2 - 1.5y - 1.1 \leq 0$$

$$x \leq 2$$

$$-x \leq 0$$

$$y \in \{0, 1\}$$

The NLP relaxation is solved to get $z_0 = -0.553$, $x_0 = 0.760$, $y_0 = 0.752$, $\lambda^*_1 = 2.000$, $\lambda^*_2 = 0.073$, $\lambda^*_3 = 0.000$ and $\lambda^*_4 = 0.000$.

The integer variable is fixed to the relaxed value $y = y_0$ and solved to obtain the sensitivities. Theorem 1 is used in order to obtain the parametric sensitivities of the continuous variables and the Lagrangian multipliers

$$x(y + \Delta y) = x^* + \frac{\partial x}{\partial y} \Delta y \quad \text{and} \quad \lambda(y + \Delta y) = \lambda^* + \frac{\partial \lambda}{\partial y} \Delta y$$

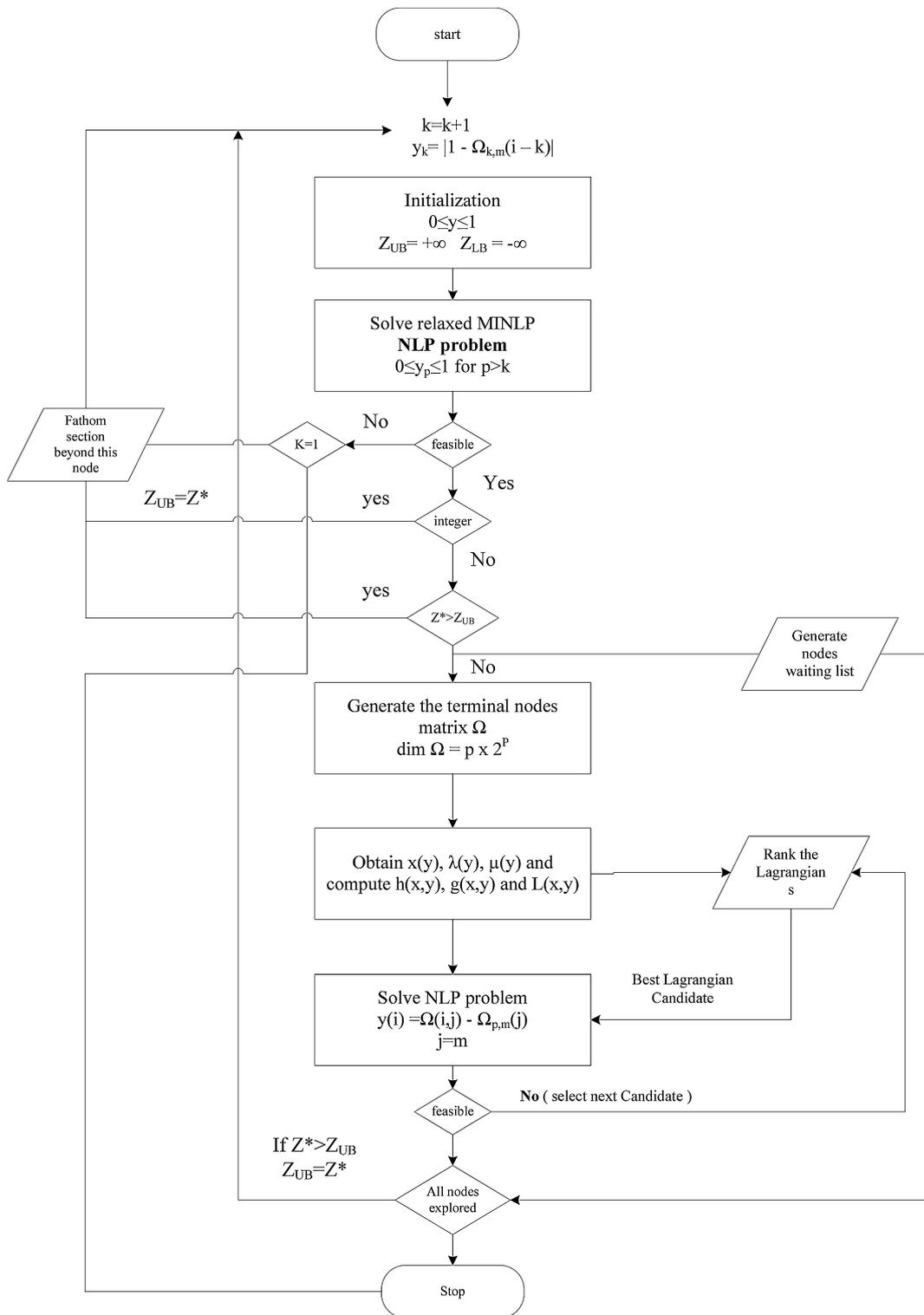


Fig. 2. Flowchart of the improved B&B algorithm.

$$\frac{\partial x}{\partial y} = -2.646, \quad \frac{\partial \lambda_1}{\partial y} = 11.634, \quad \frac{\partial \lambda_2}{\partial y} = -0.921, \quad \frac{\partial \lambda_3}{\partial y} = 0, \quad \frac{\partial \lambda_4}{\partial y} = 0$$

$$\begin{aligned} x(y) &= x_0 - 2.646(y - y_0) \\ \lambda_1 &= \lambda_{1,0} + 11.634(y - y_0) \\ \lambda_2 &= \lambda_{2,0} - 0.921(y - y_0) \\ \lambda_3 &= \lambda_{3,0} = 0 \\ \lambda_4 &= \lambda_{4,0} = 0 \end{aligned}$$

The Lagrangian function is constructed using the approximate parametric functions for x and λ

$$\begin{aligned} L(y) &= -0.553 + 3.006(y - 0.752) + (2.000 + 11.634(y - 0.752)) \\ &\quad \times (-\ln(1.76 - 2.646(y - 0.752)) + y^2) \\ &\quad + (0.073 - 0.921(y - 0.752))(-\ln(0.19 - 2.646(y - 0.752)) \\ &\quad + y^2 - 1.5y - 1.1) \end{aligned}$$

The Lagrangians are estimated for the integer solutions candidates (no optimization problem is solved, it is a function evaluation only). $L(y=1)$ is undefined because of unacceptable values for the log function, $L(y=0)=9.189$. The best Lagrangian was obtained for the candidate $(y=0)$, an NLP problem is then solved by fixing $y=0$: $Z^*=0.815$, $x^*=0.903$, $y^*=0$. Similarly to Example 1 the branching procedure is simple in this case and consists of solving the problem for $y=1$ in order to cover the whole feasibility space. For $y=1$, $Z^*=0.952 > 0.815$. Therefore this solution can be pruned. Finally the solution to this problem is $Z^*=0.815$, $x^*=0.903$, $y^*=0$.

3.3. Example 3: Process synthesis optimization I (Duran & Grossmann, 1986)

$$z = \min 5y_1 + 8y_3 + 6y_2 + 10x_1 - 7x_3 - 18 \ln(x_2 + 1) - 19.2 \times \ln(x_1 - x_2 + 1) + 10$$

$$\text{Subject to: } -0.8 \ln(x_2 + 1) + 0.96 \ln(x_1 - x_2 + 1) + 0.8x_3 \leq 0$$

$$\ln(x_2 + 1) - 1.2 \ln(x_1 - x_2 + 1) + x_3 + 2y_3 - 2 \leq 0$$

$$x_2 - x_1 \leq 0$$

$$x_2 - 2y_1 \leq 0$$

$$x_1 - x_2 - 2y_2 \leq 0$$

$$y_1 + y_2 - 1 \leq 0$$

$$y \in \{0, 1\}^3, a \leq X \leq b, X = \{x_j, j = 1, 2, 3\}, a^T = (0, 0, 0),$$

$$b^T = (2, 2, 1)$$

Step 1: Relaxed NLP

$$\begin{bmatrix} x_1^* \\ x_2^* \\ x_3^* \end{bmatrix} = \begin{bmatrix} 1.147 \\ 0.547 \\ 1.000 \end{bmatrix}, \begin{bmatrix} y_1^* \\ y_2^* \\ y_3^* \end{bmatrix} = \begin{bmatrix} 0.273 \\ 0.300 \\ 0.000 \end{bmatrix}, z^* = 0.759$$

Step2: Generate the terminal nodes matrix

$$\Omega_{n,2^n} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Number of rows = dimension of the integer vector n ; number of columns = 2^n

The root node is solved with $z=0.759$.

Eqs. (7) and (8) are used to estimate Lagrangians for nodes #8–15 (they are only estimated and are not solved) – see Fig. 3. The best estimate is for terminal node #13. This node is solved and gives $Z=6.010$ providing an upper bound to the problem.

The branching step starts by solving node #2 by adding the constraint below as stated in (11):

$$y_1 = |1 - \Omega_{1,13}(2)| = |1 - 0| = 1.$$

$$0 \leq y_2 \leq 1$$

$$0 \leq y_3 \leq 1$$

The objective function is greater than the best upper bound so far it is then fathomed. Similarly the next node explored is node #7 by adding the constraints:

$$y_1 = \Omega_{1,13}(1) = 0.$$

$$y_2 = |1 - \Omega_{2,13}(2)| = |1 - 1| = 0.$$

$$0 \leq y_3 \leq 1$$

which gives greater objective function than the upper bound. It is then fathomed. The next one to be explored is node #12 which yields an objective function $Z=14.010$ gives also greater objective function than the current upper bound and is then fathomed.

The solution is finally node #13 $y=(0,1,0)$.

3.4. Example 4: Process synthesis optimization II (Duran & Grossmann, 1986)

Consider the following MINLP example problem (Duran & Grossmann, 1986):

$$z = \min 5y_1 + 6y_3 + 8y_2 + 10y_4 + 6y_5 - 10x_1 - 15x_2 - 15x_5 + 15x_3 + 5x_4 - 20x_6 + \exp(x_1) + \exp(x_2/1.2) - 60$$

$$\ln(x_3 + x_4 + 1) + 140$$

$$\text{Subject to: } -\ln(x_3 + x_4 + 1) \leq 0$$

$$-x_1 - x_2 - 2x_5 + x_3 + 2x_6 \leq 0$$

$$-x_1 - x_2 - 0.75x_5 + x_3 + 2x_6 \leq 0$$

$$x_5 - x_6 \leq 0$$

$$2x_5 - x_3 - 2x_6 \leq 0$$

$$-0.5x_3 + x_4 \leq 0$$

$$0.2x_3 - x_4 \leq 0$$

$$\exp(x_1) - 10y_1 \leq 1$$

$$\exp(x_2/1.2) - 10y_2 \leq 1$$

$$1.25x_5 - 10y_3 \leq 0$$

$$x_3 + x_4 - 10y_4 \leq 0$$

$$-2x_5 + 2x_6 - 10y_5 \leq 0$$

$$y_4 + y_5 \leq 1$$

$$y_1 + y_2 = 1$$

$$y \in \{0, 1\}^5, a \leq X \leq b, X = \{x_j, j = 1, 2, 3, 4, 5, 6\},$$

$$a^T = (0, 0, 0, 0, 0, 0), b^T = (2, 2, -, -, 2, 3)$$

A B&B tree depicting the path taken in the tree is shown in Fig. 4. The dark coloured nodes are the nodes that were explored and light coloured nodes were not explored. The nodes are numbered for reference, the root node is numbered as 1 and the terminal nodes are numbered from 32 to 63.

The approximate parametric solution of the NLP at node 1 is used to estimate the solution at nodes 32 to 63. Estimates at nodes 32–45 give infeasible solution.

The root node is solved and gives $z=-0.554$ and $y=(0.571, 0.429, 0.250, 0.210, 0)$.

The solution at this node is used to estimate Lagrangians for terminal nodes (#32–63), these nodes are not solved. The best Lagrangian based candidate is node #47 $y=(0,1,1,1,1)$, this node is then solved but is infeasible, therefore the next best Lagrangian based candidate is selected. The second best Lagrangian based candidate is node #46 $y=\Omega_{1,46}=(0,1,1,1,0)$, this node is then solved to provide an upper bound for the problem ($Z=73.03$).

Node #46 is then a starting point for the branching step in order to explore the whole feasibility space. To be able to check the optimality, or not, of the best lagrangian candidate, the complementary sections of the tree will have to be explored.

Node #3 is solved as an NLP relaxation by adding the following constraints:

$$y_1 = |1 - \Omega_{1,46}(1)| = |1 - 0| = 1.$$

$$0 \leq y_2 \leq 1, 0 \leq y_3 \leq 1, 0 \leq y_4 \leq 1, 0 \leq y_5 \leq 1$$

It leads to a fractional solution with an objective function of $z=67.9$ which mean no fathoming criteria can be applied yet. Further branching should take place from this node. The same concept is then applied to this part of the tree to estimate solutions at the terminal nodes 48–63.

The best candidate (node #54) is solved with an objective function of $z=82.13$ and $y=\Omega_{2,54}=(1,0,1,1,0)$. This node is in its turn a starting point for sub-branching in the section of tree bounded by node #3 and nodes #63–48.

The node that is first explored is node #7 through the addition of the following constraints:

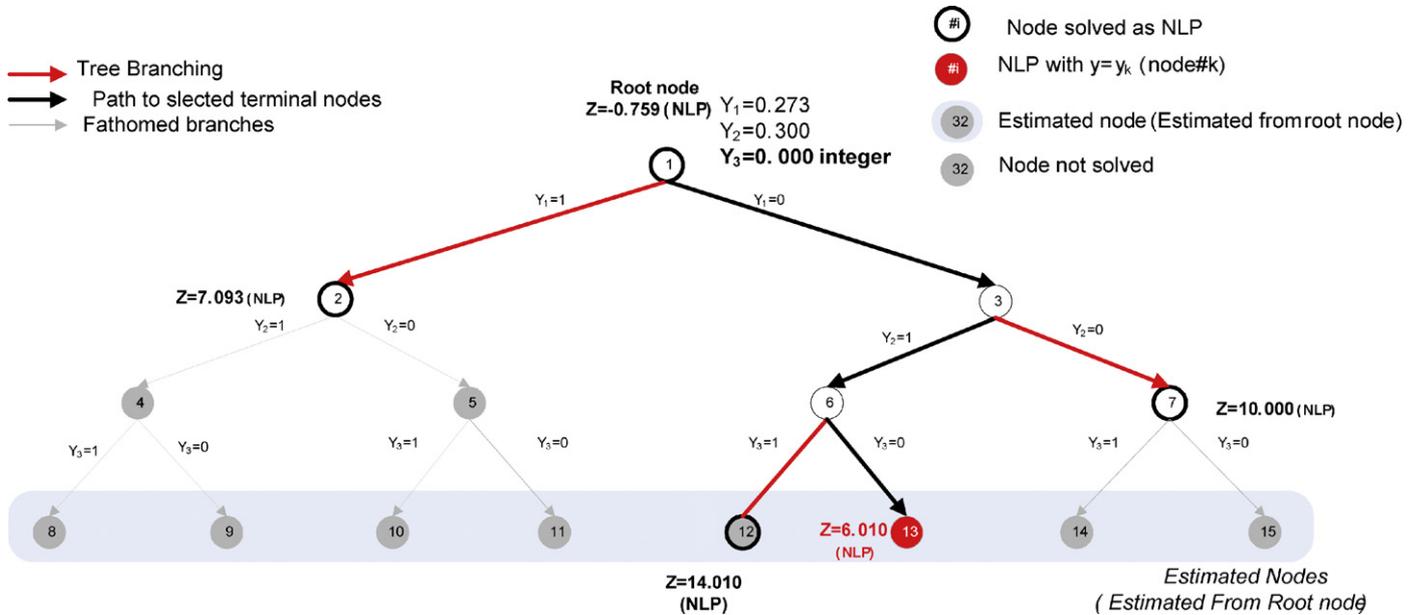


Fig. 3. Approximate parametric programming based B&B tree for Example 3.

y_1 same as for node #3

$$y_2 = |1 - \Omega_{2,50}(3)| = |1 - 0| = 1.$$

$$0 \leq y_3 \leq 1, 0 \leq y_4 \leq 1, 0 \leq y_5 \leq 1$$

This node yields an infeasible solution it will then be fathomed. The next node to be explored is node #12 through the addition of the following constraints:

y_1 same as for node #3

$$y_2 = \Omega_{2,54}(2)$$

$$Y_3 = |1 - \Omega_{2,54}(3)| = |1 - 1| = 0.$$

$$0 \leq y_4 \leq 1, 0 \leq y_5 \leq 1$$

This node yields an objective function ($Z = 86.45$) that is greater than the current best bound (73.03) it will then be fathomed. The next one to be explored is node #26:

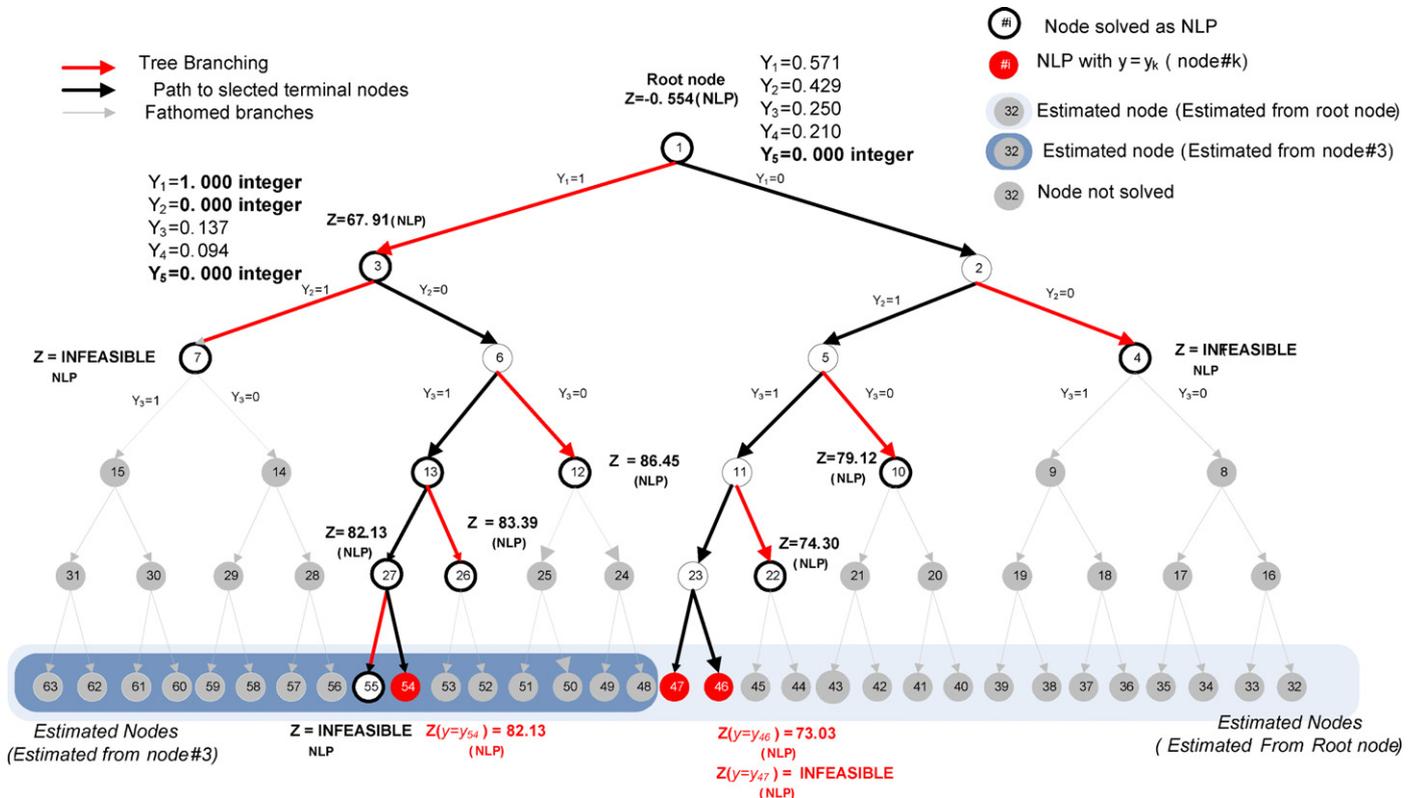


Fig. 4. Approximate parametric programming based B&B tree for Example 4.

Table 2
Computational results for example 2.

Algorithm	Traditional Branch & Bound	OA	Improved B&B
Number of MIP calls	–	3 MIP	–
Number of NLP calls	4 NLP	3 NLP	3 NLP
Solution found in	2nd node	–	2nd node

Table 3
Computational results for example 3.

Algorithm	Traditional Branch & Bound	OA	Improved B&B
Number of MIP calls	–	3 MIP	–
Number of NLP calls	6 NLP	4 NLP	5 NLP
Solution found in	2nd node	–	2nd node
CPU time (s)	0.064	0.613	0.051

Table 4
Computational results for example 4.

Algorithm	Traditional Branch & Bound	OA	Improved B&B
Number of MIP calls	–	7 MIP	–
Number of NLP calls	15 NLP	7 NLP	12 NLP
Solution found in	12th node	6th major iteration	1st node
CPU time (s)	0.141	1.224	0.080

Table 5
Computational results for example 4.

B&B algorithm	Depth first search (DFS)	Best bound (BB)	Best estimate (BE)	DFS/BB mix	DFS/BE mix	DFS/BB/BE mix	Automatic	Improved B&B
Number of NLP calls	22	8	8	15	12	12	15	12
Solution found in	20th node	5th node	6th node	12th node	7th node	7th node	12th node	1st node
Obj	73.03	73.03	73.03	74.29	73.03	73.03	74.29	73.03
CPU time (s)	0.157	0.074	0.081	0.128	0.105	0.110	0.128	0.080

y_1 same as for node #3

$$\begin{aligned} y_2 &= \Omega_{2,54}(2) \\ y_3 &= \Omega_{2,54}(3) \\ y_4 &= |1 - \Omega_{2,54}(4)| = |1 - 1| = 0. \\ 0 &\leq y_5 \leq 1 \end{aligned}$$

This node yields an objective function ($Z=83.39$) that is greater than the current best bound (73.03) it will also then be fathomed. The next one to be explored is node #55:

y_1 same as for node #3

$$\begin{aligned} y_2 &= \Omega_{2,50}(2) \\ y_3 &= \Omega_{2,50}(3) \\ y_4 &= \Omega_{2,50}(3) \\ y_5 &= |1 - \Omega_{2,50}(5)| = |1 - 0| = 1. \end{aligned}$$

This node yields an infeasible solution it will then be fathomed. Hence, the section of tree bounded by node #3 and nodes #63–48 will be fathomed. The initial branching procedure which started from the terminal node #46 can now carry on. Therefore, now that the section below node 3 is cleared, the next node to explore is now node #4.

$$\begin{aligned} y_1 &= \Omega_{1,46}(1) = 0 \\ y_2 &= |1 - \Omega_{1,46}(2)| = |1 - 1| = 0. \\ 0 &\leq y_3 \leq 1, 0 \leq y_4 \leq 1, 0 \leq y_5 \leq 1 \end{aligned}$$

This NLP problem is infeasible hence fathomed. The next one is node #10

$$\begin{aligned} y_1 &= \Omega_{1,46}(1) = 0 \\ y_2 &= \Omega_{1,46}(2) = 1 \\ y_3 &= |1 - \Omega_{1,46}(3)| = |1 - 0| = 1. \\ 0 &\leq y_4 \leq 1, 0 \leq y_5 \leq 1 \end{aligned}$$

This node yields an objective function ($Z=79.12$) that is greater than the current best bound (73.03) it will be fathomed.

The last node to be explored is node #22

$$\begin{aligned} y_1 &= \Omega_{1,46}(1) = 0 \\ y_2 &= \Omega_{1,46}(1) = 1 \\ y_3 &= \Omega_{1,46}(1) = 1 \\ y_4 &= |1 - \Omega_{1,46}(4)| = |1 - 1| = 0. \\ 0 &\leq y_5 \leq 1 \end{aligned}$$

Similarly node #22 ($z=74.30$) is also fathomed for the same reason. The node #47 was solved as a potential Lagrangian based candidate and has been found infeasible, therefore, no need to solve this node again, it is then fathomed also. The solution is finally node #46 $y=(0,1,1,1,0)$.

4. Algorithm computational results

In this section, a comparison between the proposed improved B&B algorithm, the traditional Branch & Bound (GAMS/SBB) and Outer Approximation (GAMS/DICOPT) algorithms is presented for examples 2 (Table 2), 3 (Table 3) and 4 (Table 4).

A more detailed comparison between (GAMS/SBB) and the proposed improved B&B algorithm for Example 4 is also presented in Table 5.

The computational requirements for the proposed algorithm are comparable to or better than for 5 out of 7 traditional B&B algorithms (Table 5). A more detailed computational comparison will be presented in the future publications; the main objective of this work was to present and demonstrate applicability of the basic concepts of the proposed algorithm. Different tree search techniques (e.g. Depth First Search (DFS)) will also be explored in conjunction with the proposed algorithm. The finite convergence and guaranteed optimality properties, for convex functions for the traditional B&B algorithm, are retained in the proposed algorithm.

5. Concluding remarks

An approximate parametric programming solution at the root node and other fractional nodes of the B&B tree are obtained and used to estimate the solution at the terminal nodes in different sections of the tree. These estimates are then used to guide the search in the B&B tree, resulting in fewer nodes being evaluated and reduction in the computational effort. Preliminary computational results are encouraging, future work will involve testing the proposed algorithm on larger scale problems and comparing with other algorithms reported in the literature.

Acknowledgment

Financial support from EPSRC (EP/G059195/1) is gratefully acknowledged.

References

- Bonami, P., Biegler, L. T., Conn, A. R., Cornuejols, G., Grossmann, I. E., Laird, C. D., et al. (2008). An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5, 186–204.
- Borchers, B., & Mitchell, J. E. (1994). An improved branch and bound algorithm for mixed-integer nonlinear programs. *Computers and Operations Research*, 21, 359–367.
- Dua, V. (2006). Stability analysis of nonlinear model predictive control: An optimization based approach. In *16th European symposium on computer aided process engineering and 9th international symposium on process systems engineering*. Amsterdam: Elsevier., pp. 1287–1292.
- Dua, V. (2010). A mixed-integer programming approach for optimal configuration of artificial neural networks. *Chemical Engineering Research and Design*, 88, 55–60.
- Dua, V., & Pistikopoulos, E. N. (1999). Algorithms for the solution of multiparametric mixed-integer nonlinear optimization problems. *Industrial & Engineering Chemistry Research*, 38, 3976–3987.
- Duran, M. A., & Grossmann, I. E. (1986). An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36, 307–339.
- Fiacco, A. V. (1976). Sensitivity analysis for nonlinear programming using penalty methods. *Mathematical Programming*, 10, 287–311.
- Fiacco, A. V. (1983). *Introduction to sensitivity and stability analysis in nonlinear programming*. New York: Academic Press.
- Floudas, C. A. (1995). *Nonlinear and mixed-integer optimization: Fundamentals and applications*. Oxford University Press.
- Geoffrion, A. M. (1972). Generalized Benders Decomposition. *Journal of Optimization Theory and Applications*, 10, 237–260.
- Goyal, V., & Ierapetritou, M. G. (2004). Computational studies using a novel simplicial-approximation based algorithm for MINLP optimization. *Computers and Chemical Engineering*, 28, 1771–1780.
- Grossmann, I. E. (1996). *Mixed-integer optimization techniques for algorithmic process synthesis*. *Advances in chemical engineering: Process synthesis* Elsevier., pp. 171–246.
- Grossmann, I. E., & Daichendt, M. M. (1996). New trends in optimization based approaches for process synthesis. *Computers and Chemical Engineering*, 20, 665–683.
- Gupta, O. K., & Ravindran, A. (1985). Branch and bound experiments in convex nonlinear integer programming. *Management Science*, 31, 1533–1546.
- Jackson, R. H. F., & McCormick, G. P. (1988). Second-order sensitivity analysis in factorable programming: Theory and applications. *Mathematical Programming*, 41, 1–27.
- Kallrath, J. (2000). Mixed integer optimization in the chemical process industry: Experience, potential and future. *Trans. I. Chem E. Part A*, 78, 809–822.
- Pistikopoulos, E. N. (2009). Perspectives in multi-parametric programming and explicit model predictive control. *AIChE Journal*, 55, 1918–1925.
- Pistikopoulos, E. N., Georgiadis, M. C., & Dua, V. (2007a). *Multi-parametric programming* Wiley-VCH.
- Pistikopoulos, E. N., Georgiadis, M. C., & Dua, V. (2007b). *Multi-parametric model based control* Wiley-VCH.
- Westerlund, T., & Pettersson, F. (1995). An extended cutting plane method for solving convex MINLP problems. *Computers and Chemical Engineering*, 19, S131–S136.