



# Mixed integer polynomial programming

Vivek Dua\*

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

## ARTICLE INFO

### Article history:

Received 16 January 2014  
Received in revised form 15 July 2014  
Accepted 20 July 2014  
Available online 29 July 2014

### Keywords:

Mixed integer programming  
Polynomial programming  
Nonlinear inversion  
Multi-parametric programming

## ABSTRACT

The mixed integer polynomial programming problem is reformulated as a multi-parametric programming problem by relaxing integer variables as continuous variables and then treating them as parameters. The optimality conditions for the resulting parametric programming problem are given by a set of simultaneous parametric polynomial equations which are solved analytically to give the parametric optimal solution as a function of the relaxed integer variables. Evaluation of the parametric optimal solution for integer variables fixed at their integer values followed by screening of the evaluated solutions gives the optimal solutions.

© 2014 The Author. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/3.0/>).

## 1. Introduction

Mathematical modelling and model-based optimization of chemical process systems have great potential for providing answers on how to optimally design and operate these systems. One key difficulty in not being able to fully exploit this potential is the presence of nonlinear terms in the mathematical models. This issue is further exacerbated when the model also includes integer variables, for incorporating structural choices. A generic formulation of such mathematical programming problems is given by mixed integer nonlinear programmes (MINLP) (Grossmann, 2002):

Problem P1 :

$$z_1 = \min_{\mathbf{x}, \mathbf{y}} f(\mathbf{x}, \mathbf{y})$$

subject to :  $\mathbf{h}(\mathbf{x}, \mathbf{y}) = 0$

$$\mathbf{g}(\mathbf{x}, \mathbf{y}) \leq 0$$

$$\mathbf{x} \in \mathbb{R}^{n_x}$$

$$\mathbf{y} \in \{0, 1\}^{n_y}$$

where  $\mathbf{x}$  is a vector of continuous variables,  $\mathbf{y}$  is a vector of binary variables,  $\mathbf{h}$  is an  $n_h$  dimensional vector of equality constraints,  $\mathbf{g}$  is an  $n_g$  dimensional vector of inequality constraints and  $f$  is the scalar objective function. Synthesis of chemical process flowsheets and design of materials are two typical problems demonstrating the

application of mathematical programmes simultaneously involving nonlinearities and integer variables (Dua and Pistikopoulos, 1998). For process synthesis problems,  $\mathbf{x}$  represents continuously varying quantities such as temperature, pressure, flowrates etc.,  $\mathbf{y}$  is used to model structural decisions such as selection of appropriate processing units and inter-connections between the units etc.,  $f$  represents an objective function such as cost or environmental impact to be optimized,  $\mathbf{h}$  represents conservation equations i.e. mass and energy balances and  $\mathbf{g}$  represents constraints on quantities such as lowest acceptable purity and highest allowable safe operating temperatures and pressures. For material design problems  $\mathbf{x}$  represents material properties,  $\mathbf{y}$  models selection of constituent molecular groups,  $f$  represents deviation from desired property values,  $\mathbf{h}$  represents property prediction correlations and  $\mathbf{g}$  represents lower and upper bounds on values of the material properties. Note that this approach for material design problems is based upon matching property targets but other formulations for such problems also exist in the literature.

Solving P1 is NP-hard and has created huge interest for developing computationally efficient algorithms for obtaining solution of P1. New theoretical developments for solving P1 have pushed the boundaries of application of P1 to many areas in engineering and science. Several software tools are available to solve these problems, DICOPT (Viswanathan and Grossmann, 1990), MINOPT (MINOPT, 1998), BARON (Sahinidis, 1996), GloMIQO (Misener and Floudas, 2013), Alpha-ECP (Westerlund and Lundqvist, 2005) – to name a few. The reader is referred to recent survey papers by Belotti et al. (2013) and D'Ambrosio and Lodi (2011) presenting an overview of advances for solving P1 and the books (Biegler et al.,

\* Tel.: +44 020 7679 0002; fax: +44 020 7383 2348.  
E-mail address: [v.dua@ucl.ac.uk](mailto:v.dua@ucl.ac.uk)

1997; Floudas, 1995; Nemhauser and Wolsey, 1988) for an introduction to the topic.

Gueddar and Dua (2012) reformulated P1 as a multi-parametric nonlinear programme (mp-NLP) by first relaxing  $\mathbf{y}$  as continuous variables and then treating  $\mathbf{y}$  as parameters. An approximate solution of the mp-NLP is obtained and that solution is then used to estimate the solution at the terminal nodes of the Branch and Bound (B&B) tree and guide the search in the tree; integer variables are the branching variables in the tree. For details of multi-parametric programming the reader is referred to Dua and Pistikopoulos (1999), Pistikopoulos et al. (2007a,b), Pistikopoulos (2009) and Wittmann-Hohlbein and Pistikopoulos (2014).

In this work, the case when  $f$ ,  $\mathbf{h}$  and  $\mathbf{g}$  are polynomial functions is considered, problem P1 therefore becomes a Mixed Integer Polynomial Optimization (MIPOPT) problem. Patil et al. (2012) presented a Bernstein polynomial approach for solving such problems. Teles et al. (2013) proposed a discretization approach using bilinear terms as the building block. In this work, the integer variables,  $\mathbf{y}$ , are relaxed as continuous variables and then treated as parameters resulting in a multi-parametric polynomial programme (mp3). An exact solution of the resulting mp3 can be obtained by exact multi-parametric nonlinear inversion of the optimality conditions, see for example Fotiou et al. (2007). The proposed approach hence does not require approximate solution of the mp-NLP followed by a tree search as in Gueddar and Dua (2012).

In the next section polynomial programming is introduced and an example for exact solution of polynomial programmes is presented. An algorithm for solving MIPOPT based upon mp3 reformulation is proposed in Section 3 and in Section 4 illustrative examples are presented. A discussion of results and concluding remarks are provided in Section 5.

## 2. Polynomial programming

Consider the following nonlinear programming (NLP) problem:

Problem P2 :

$$\begin{aligned} z_2 &= \min_{\mathbf{x}} f(\mathbf{x}) \\ \text{subject to : } & \mathbf{h}(\mathbf{x}) = 0 \\ & \mathbf{g}(\mathbf{x}) \leq 0 \\ & \mathbf{x} \in \mathbb{R}^{n_x} \end{aligned}$$

Descent or similar algorithms for computing solution of P2 are based upon an iterative strategy where the solution obtained at an iteration verifies Fritz–John (FJ) or Karush–Kuhn–Tucker (KKT) conditions (Bazaraa et al., 1993), in this work the KKT conditions are considered, as follows.

KKT conditions :

(a) Equality constraints :

$$\nabla_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda}) = 0$$

$$h(\mathbf{x}) = 0$$

$$\lambda_j g_j(\mathbf{x}) = 0, j = 1, \dots, n_g$$

(b) Inequality constraints :

$$\mathbf{g}(\mathbf{x}) \leq 0$$

$$\lambda_j \geq 0, j = 1, \dots, n_g$$

where

$$L(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_{i=1}^{n_h} \mu_i h_i(\mathbf{x}) + \sum_{j=1}^{n_g} \lambda_j g_j(\mathbf{x})$$

is the Lagrangian function.

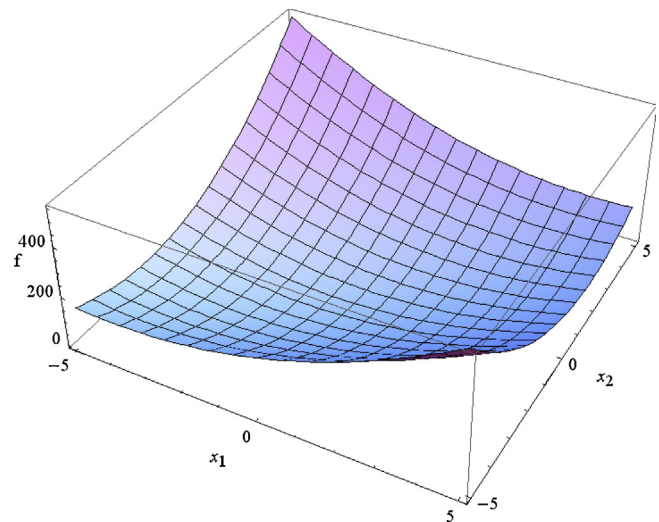


Fig. 1. Example 1, plot of the objective function,  $f$ , as a function of  $x_1$  and  $x_2$ .

The equality constraints in the KKT conditions are  $n_x + n_h + n_g$  dimensional and the vector of variables,  $[\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda}]$ , is also  $n_x + n_h + n_g$  dimensional. For generic nonlinear functions solution of equality constraints is usually obtained by employing a numerical technique, such as Newton's method. The solution of the equality constraints obtained is verified by checking whether it satisfies the inequality constraints in the KKT conditions. Considering a special case when  $f$ ,  $\mathbf{g}$  and  $\mathbf{h}$  in the NLP are polynomial, a set of equations polynomial in  $[\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda}]$  is obtained. These polynomial equations can be solved analytically, at least in theory, to obtain a closed form solution which includes all the solutions (Hägglöf et al., 1995). This can be achieved by using the theory of Gröbner Bases where the Buchberger algorithm can be used to transform the set of polynomial equations into a triangular system of equations (Buchberger and Winkler, 1998). The triangular system is the nonlinear polynomial equivalent of the triangular system obtained by Gaussian elimination for a linear system of equations. The computational complexity of this method grows exponentially with the number of variables, but it is an active area of research with various developments including parallel computing to improve computational speed. There are softwares for symbolic manipulations such as Mathematica (Wolfram Research, 2013) that can analytically solve systems of polynomial equations, which in our case is given by the equality constraints in the KKT conditions. The set of solutions thus obtained can then be checked to see if they satisfy the inequality constraints in the KKT conditions. Further screening tests are also carried out, i.e., whether complementary slackness (CS) and constraints qualification (CQ) conditions are met is checked. In this paper linear independence constraint qualification (LICQ) was used for checking the CQ condition. Consider the following illustrative example for demonstrating the basic idea.

Example 1 : Polynomial programming

$$\min_{\mathbf{x}} f(\mathbf{x}) = 5x_1^2 + 9x_2^2 - 8x_1x_2$$

subject to :

$$g_1(\mathbf{x}) = 2 - x_1x_2 \leq 0$$

$$g_2(\mathbf{x}) = -3 + x_1x_2 \leq 0$$

$$g_3(\mathbf{x}) = -5 + x_1 \leq 0$$

$$g_4(\mathbf{x}) = -5 - x_1 \leq 0$$

The objective function is plotted as a function of  $x_1$  and  $x_2$  in Fig. 1 and the feasible region is given by the shaded area in Fig. 2. The two optimal solutions are also shown in Fig. 2.

Download English Version:

<https://daneshyari.com/en/article/6595499>

Download Persian Version:

<https://daneshyari.com/article/6595499>

[Daneshyari.com](https://daneshyari.com)