Operations Research Letters 38 (2010) 39–46

Contents lists available at ScienceDirect

Operations Research Letters

journal homepage: www.elsevier.com/locate/orl

Piecewise linear approximation of functions of two variables in MILP models

Claudia D'Ambrosio, Andrea Lodi*, Silvano Martello

DEIS, University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy

ARTICLE INFO

Article history:
Received 1 February 2009
Accepted 24 July 2009
Available online 6 October 2009

Keywords:
Mixed integer linear programming
Non-linear functions of two variables
Piecewise linear approximation

ABSTRACT

We consider three easy-to-implement methods for the piecewise linear approximation of functions of two variables. We experimentally evaluate their approximation quality, and give a detailed description of how the methods can be embedded in a MILP model. The advantages and drawbacks of the three methods are discussed on numerical examples.

1. Introduction

In recent years, the increased efficiency of mixed integer linear programming (MILP) software tools has encouraged their use also in the solution of non-linear problems, bringing to the need for efficient techniques to linearize non-linear functions of one or more variables. The standard methodologies consist in the piecewise linear approximation of such functions.

For functions of a single variable, say, $f(x)$, the piecewise linear approximation is obtained by introducing a number n of sampling coordinates $x_1, \ldots, x_n$ on the x axis (*breakpoints*) on which the function is evaluated, with x_1 and x_n coinciding with the left and right extremes of the x domain (see Fig. 1). The function is then approximated by the linear segments $[(x_i, f(x_i)), (x_{i+1}, f(x_{i+1}))]$ $(i = 1, \ldots, n-1)$. More precisely, for any given x value, say, $\bar{x}$, with $x_i \le \bar{x} \le x_{i+1}$, the function value is approximated by convex combination of $f(x_i)$ and $f(x_{i+1})$. Let λ be the (unique) value in $[0, 1]$ such that

$$\bar{x} = \lambda x_i + (1-\lambda)x_{i+1}. \tag{1}$$

Then the approximated value is

$$f^a(\bar{x}) = \lambda f(x_i) + (1-\lambda) f(x_{i+1}). \tag{2}$$

This methodology can alternatively be described through the slope $(f(x_{i+1}) - f(x_i))/(x_{i+1} - x_i)$ of the interpolating function, namely

$$f^a(\bar{x}) = f(x_i) + (\bar{x} - x_i)\frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}. \tag{3}$$

From (3), one has $\lambda = (x_{i+1} - \bar{x})/(x_{i+1} - x_i)$.

In order to use the above technique in a MILP solver, it is necessary to include in the model variables and constraints that force any x value to be associated with the proper pair of consecutive breakpoints (or with a single one, in case $x \in \{x_1, \ldots, x_n\}$). Let us introduce a continuous variable α_i for each breakpoint i, such that $\alpha_i \in [0, 1]$ $(i = 1, \ldots, n)$. Let h_i be a binary variable associated with the ith interval $[x_i, x_{i+1}]$ $(i = 1, \ldots, n-1)$, with dummy values $h_0 = h_n = 0$ at the extremes. The approximate value f^a can then be obtained by imposing the following constraints:

$$\sum_{i=1}^{n-1} h_i = 1 \tag{4}$$

$$\alpha_i \le h_{i-1} + h_i \quad (i = 1, \ldots, n) \tag{5}$$

$$\sum_{i=1}^{n} \alpha_i = 1 \tag{6}$$

$$x = \sum_{i=1}^{n} \alpha_i x_i \tag{7}$$

$$f^a = \sum_{i=1}^{n} \alpha_i f(x_i). \tag{8}$$

Constraint (4) imposes that only one h_i, say, $h_{\bar{i}}$, takes the value 1. Hence, constraints (5) impose that the only α_i values different from 0 can be $\alpha_{\bar{i}}$ and $\alpha_{\bar{i}+1}$. It follows from (6) and (7) that $\alpha_{\bar{i}} = \lambda$ and $\alpha_{\bar{i}+1} = 1 - \lambda$ (see (1)). Constraint (8) ensures then the correct computation of the approximate value according to (2).

In contexts of this type, the MILP constraints can be simplified by the so-called special ordered sets, introduced by Beale and Tomlin [1], and extensively studied in [2–4]. By defining a set of variables to be a *Special Ordered Set of type k* (SOS*k*), one imposes that at most k such variables can take a non-zero value, and that they must be adjacent. Most modern MILP solvers are capable of

* Corresponding author.
E-mail addresses: c.dambrosio@unibo.it (C. D'Ambrosio), andrea.lodi@unibo.it (A. Lodi), silvano.martello@unibo.it (S. Martello).

0167-6377/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.orl.2009.09.005

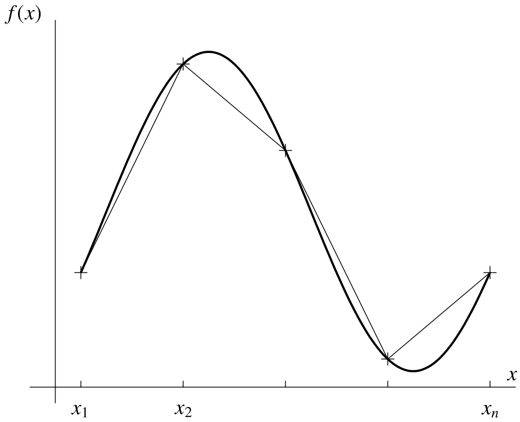

Fig. 1. Piecewise linear approximation of a univariate function.

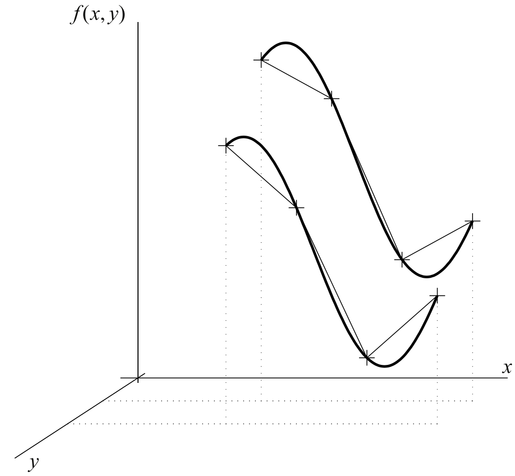

Fig. 2. One-dimensional method.

automatically handling special ordered sets of type 1 and 2. In our case, by defining the α variables to be a SOS2, one does not need to explicitly state h variables, so constraints (6)–(8) are enough to produce the correct computation. The additional advantage of this technique is that the enumerative phase may be enhanced by the internal use of special purpose branching rules.

This article concentrates on the piecewise linear approximation of functions $f(x, y)$ of two variables, with special emphasis on their practical use within MILP models. When a function is approximated one can require that the approximating function has favorable theoretical properties such as continuity, differentiability, and so on. Depending on how they are viewed, some of the approximations discussed here do not have strong theoretical properties (for example, they can be discontinuous). However, their properties are favorable for practical use: they are simple to model, we can minimize them, and they produce small-size MILPs. For theoretical treatments of standard numerical approximation methods, the reader is referred, e.g., to [5,6] and in particular to Chapter III of [7].

The article is organized as follows. In Section 2, we present three approaches and give a detailed description of how they can be embedded in a MILP model. The simplest method (Section 2.1) consists of using the one-variable technique above for a discretized set of y values. A more complex and classical approach (Section 2.2) is based on the definition of triangles in the three-dimensional space, and can be seen as the extension of the one-variable technique (see [8,9,2,4,10]). In Section 2.3, we give full description of a third approach, recently used within an applied context (see [11]), which appears particularly suitable for MILP modeling. In Section 3, we discuss the computational issues associated with the three approaches. In Section 3.1 we examine the average and maximum error of the approximation on a large set of randomly generated instances. In Section 3.2, we examine the advantages and drawbacks of the associated MILP models by discussing theoretical properties, and by analyzing the outcome of a number of computational experiments on a real-world application in electric power generation.

2. Methods

In this section, we describe three techniques for the piecewise linear approximation of functions of two variables.

2.1. One-dimensional method

An immediate adaptation of the one-variable technique to the case of functions of two variables is as follows. Let us introduce a number m of coordinates on the y axis, $y_1, \ldots, y_m$ (y_1 and y_m being the left and right extremes of the y domain). For the jth interval $[y_j, y_{j+1})$, let $\widetilde{y}_j$ be the associated sampling coordinate, leading to $m-1$ univariate functions $f(x, \widetilde{y}_j)$ $(j = 1, \ldots, m-1)$. For any given y value, say, $\bar{y} \in [y_j, y_{j+1})$, the approximated function values $f^a(x, \bar{y})$ are then given by the piecewise linear approximation of $f(x, \widetilde{y}_j)$ with breakpoints $x_1, \ldots, x_n$ (see Fig. 2). In the following, we assume that the sampling coordinate is the left extreme of the interval, i.e., $\widetilde{y}_j = y_j$. In this way, the approximating function agrees with the given function at the breakpoints. In practical applications, it can often be preferable to use the central point of the interval as the sampling coordinate, thus loosing such property.

Let $\beta_1, \ldots, \beta_{m-1}$ be binary variables, defined as an SOS1, with β_j taking the value 1 if and only if the given value $\bar{y}$ belongs to $[y_j, y_{j+1})$. The approximate value f^a is then obtained through (6) and (7), and

$$y \le \sum_{j=1}^{m-1} \beta_j y_{j+1} \tag{9}$$

$$y \ge \sum_{j=1}^{m-1} \beta_j y_j \tag{10}$$

$$\sum_{j=1}^{m-1} \beta_j = 1 \tag{11}$$

$$f^a \le \sum_{i=1}^{n} \alpha_i f(x_i, \widetilde{y}_j) + M(1-\beta_j) \quad (j = 1, \ldots, m-1) \tag{12}$$

$$f^a \ge \sum_{i=1}^{n} \alpha_i f(x_i, \widetilde{y}_j) - M(1-\beta_j) \quad (j = 1, \ldots, m-1) \tag{13}$$

where α is the SOS2 introduced in the previous section and M is a very large value ("big-M"). Constraints (9)–(11) impose $\beta_{\bar{j}} = 1$ and $\beta_j = 0$ for $j \ne \bar{j}$, $\bar{j}$ being the interval which contains y. Constraints (12) and (13) are inactive if $\beta_j = 0$, hence providing $f^a = \sum_{i=1}^{n} \alpha_i f(x_i, \widetilde{y}_{\bar{j}})$ for the correct interval $\bar{j}$.

Note that in model (9)–(13), for a given x value, f^a can take two values for $y = y_{\bar{j}}$, as either $\beta_{\bar{j}-1}$ or $\beta_{\bar{j}}$ can equivalently take the value 1. Although this can be inessential in practice, such theoretical drawback can be corrected by replacing (9) with $y \le \sum_{j=1}^{m-1} \beta_j y_{j+1} - 2\theta$, where θ is the feasible tolerance of linear constraints in the specific MILP solver.

2.2. Triangle method

A more complex method can be obtained by extending the one-variable technique to the two-variable case. Consider again n sampling coordinates $x_1, \ldots, x_n$ on the x axis and m sampling