RAIRO-Oper. Res. 49 (2015) 845–864
DOI: 10.1051/ro/2015031

RAIRO Operations Research
www.rairo-ro.org

A JOINTLY CONSTRAINED BILINEAR PROGRAMMING METHOD FOR SOLVING GENERALIZED COURNOT–PARETO MODELS *

Nguyen Van Quy[1]

Abstract. We propose a vector optimization approach to linear Cournot oligopolistic market equilibrium models where the strategy sets depend on each other. We use scalarization technique to find a Pareto efficient solution to the model by using a jointly constrained bilinear programming formulation. We then propose a decomposition branch-and-bound algorithm for globally solving the resulting bilinear problem. The subdivision takes place in one-dimensional intervals that enables solving the problem with relatively large sizes. Numerical experiments and results on randomly generated data show the efficiency of the proposed algorithm.

Keywords. Generalized Cournot model, bilinear programming, branch-and-bound, Pareto solution.

Mathematics Subject Classification. 47J20, 49J40.

1. Introduction

The Cournot oligopolistic market model is one of fundamental models in economics. In the classic models that has been considered in a lot of papers (see *e.g.* [5, 14, 16, 25] and the references therein), it is assumed that there are n firms producing a common homogeneous commodity. Each firm i has an independent strategy set $U_i \subset R_+$ and a profit function. Let $x_i \in U_i$ denote a corresponding production level of firm i. Actually, each firm seeks to maximize its profit by

Received February 1, 2015. Accepted August 8, 2015.

* *This work was completed during the stay of the author at the University of Pittsburgh, USA.*

[1] Academy of Finance, Hanoi, Vietnam. quynv2002@yahoo.com

choosing a corresponding production level under the presumption that the production of the other firms are parametric input. A commonly used approach to this model is based upon the famous Nash equilibrium concept. In the linear Cournot model, the profit function of firm i is given by

$$f_i(x) = \left(\alpha_i - \beta_i \sum_{i=1}^{n} x_i\right) x_i - h_i(x_i) \ (i = 1, \dots, n),$$

where $\alpha_i, \beta_i > 0$ and the cost function h_i is an affine function that depends only on the quantity x_i of firm i. In this linear case, it has been shown that (see *e.g.* [14]) the model has a unique Nash equilibrium point which is the unique solution of a strongly convex quadratic program. In the case, when h_i $(i = 1, \dots, n)$ are convex, the problem of finding a Nash equilibrium point can be formulated as a variational inequality. When some of functions h_i $(i = 1, \dots, n)$ are piecewise linear concave, the model can be formulated as a (nonconvex) mixed variational inequality and a solution algorithm is developed in [19] to solve the resulting mixed variational inequality.

In generalized Nash−Cournot model, where the strategy set of each firm depends on the strategy sets of other firms, the model then can be formulated as a quasivariational inequality. Recently, Fukushima [8] and Kubota *et al.* [15] proposed a class of gap functions for generalized Nash equilibrium problems in order to convert the problem into an optimization problem. However, this optimization problem being nonconvex, supplementary conditions must be imposed to ensure that any stationary point is a solution of the generalized variational inequality. An axiomatic gap function approach is proposed by Aussel *et al.* in [2] for quasivariational inequalities and generalized Nash equilibrium problems. A smooth dual gap function and smoothness of regularized gap functions for quasivariational inequalities are considered in [10–12]. In [7] Facchinei *et al.* used the KKT conditions for solving quasivariational inequalities. Very recently, Strodiot *et al.* [26] presented new extragradient algorithms for finding a solution of quasiequilibrium problems which contain the generalized Nash equilibrium problem as a special case. The convergence of these algorithms are proved under some additional conditions that, unfortunately, are not satisfied for the generalized linear Nash−Cournot equilibrium models. A comprehensive survey for generalized Nash equilibrium problems can be found in [6].

In this paper we proposed a vector optimization approach rather than the equilibrium approach to generalized Cournot market models with linear cost functions. Our proposal is motivated by the following facts. First, to our knowledge, without supplementary conditions, generalized Nash equilibrium problems, including generalized linear Nash−Cournot models cannot be solved by the existing methods. Second, in the Cournot model described above, the total profit $\sum_{j=1}^{n} f_j(x)$ where x is a Pareto solution of the vector function $f(x) := (f_1(x), \dots, f_n(x))^T$ on the strategy set, in general, is greater than that when x is a Nash equilibrium point. Moreover, in some cases of practical models, the total amount of the commodity

$\sum_{j=1}^{n} x_j$ must be equal to a given quota $m_0 > 0$. Third, by using the Pareto solution, the model can be handled by a jointly constrained bilinear programming efficient algorithm employing specific properties of the model under consideration. It is worth mentioning (see Example 2.5 in Sect. 3) that a Pareto solution may or may not be an equilibrium point.

To be specific, we suppose that the strategy set of the model is a polyhedral convex set given by

$$D := \begin{cases} Ax \le b, \\ x_i \in U_i \ (i = 1, \dots, n), \end{cases}$$

where A is a $m \times n$ matrix and $b \in \mathbb{R}^m$. In this case we formulate the problem of finding a Pareto solution of the model as a jointly constrained bilinear mathematical program. There are some available methods for globally solving the latter problem, however all of them can solve the problem with very limited size [1, 13]. Fortunately, for our problem, thanks to its specific structure, we can propose a branch-and-bound algorithm which solves the problem efficiently. In fact, the adaptive branching operation in the proposed algorithm takes place in one dimensional intervals, which makes the algorithm efficient for the models with relatively large sizes. We have tested the algorithm with a number of models on randomly generated data. The obtained computational results show the efficiency of the algorithm.

2. Preliminaries on the equilibria approach to the Nash−Cournot model

First we briefly present the classical oligopolistic market model where it is assumed that there are n firms producing a commonly homogeneous commodity. The commodity's quantity of firm i is assigned by $x_i \in U_i$ $(i = 1, \dots, n)$, and $x^T = (x_1, \dots, x_n)$ is the vector commodity's quantity of all n firms. In this paper, we assume that the price p_i of firm i depends on the total quantity $\sigma = \sum_{i=1}^{n} x_i$ of the commodity, and the cost function h_i of firm i depends only on the quantity x_i $(i = 1, \dots, n)$.

The price function of firm i is defined by:

$$p_i(x) := \alpha_i - \beta_i \sum_{i=1}^{n} x_i,$$

where β_i $(i = 1, \dots, n)$ is the price-reduced coefficient of firm i. Naturally, the profit function of firm i has the form:

$$f_i(x) = \left(\alpha_i - \beta_i \sum_{i=1}^{n} x_i \right) x_i - h_i(x_i).$$

A commonly used approach to this model is based upon the Nash equilibrium concept. This concept is used when each firm seeks to maximize its profit by choosing the corresponding production level under the presumption that the production of the other firms are parametric input. In this context, a Nash equilibrium is a production pattern in which no firm can increase its profit by changing its controlled variables.

Mathematically, a point $x^* = (x_1^*, \dots, x_n^*) \in U := U_1 \times U_2 \times \dots \times U_n$ is said to be a (global) Nash-equilibrium if

$$f_i(x_1^*, \dots, x_{i-1}^*, y_i, x_{i+1}^*, \dots, x_n^*) \le f_i(x_1^*, .., x_n^*), \ \forall y_i \in U_i, \ \forall i = 1, \dots, n. \tag{2.1}$$

When $h_i(.)$ and $p(.)$ are affine, this Nash$-$Cournot model can be formulated as a special Nash equilibrium problem in the n-person noncooperative game theory, which in turn is a strongly monotone variational inequality.

In fact, let

$$\Psi(x, y) := -\sum_{i=1}^{n} f_i(x_1, \dots, x_{i-1}, y_i, x_{i+1}, \dots, x_n) \tag{2.2}$$

and

$$\Phi(x, y) := \Psi(x, y) - \Psi(x, x., \tag{2.3}$$

Using this well-known Nikaido$-$Isoda bifunction [23] it has been proved that (see *e.g.* [14, 19]) the problem of finding an equilibrium point of this model can be formulated as the following equilibrium problem [3, 4]

$$\begin{cases} \text{find} \ \ x^* \in U \text{ such that} \\ \Phi(x^*, y) \ge 0, \ \forall y \in U. \end{cases} \tag{EP}$$

In the classic linear model, the cost and the price functions for each firm are assumed to be affine with the forms

$$\begin{aligned} h_i(x_i) &:= \mu_i x_i + \xi_i, \ \mu_i \ge 0, \ \xi_i \ge 0, \\ p_i(\sigma) \ &:= p_i\left(\textstyle\sum_{j=1}^{n} x_j\right) = \alpha_i - \beta_i \textstyle\sum_{j=1}^{n} x_j, \ \alpha_i \ge 0, \ \beta_i > 0, \\ &\quad \text{with } \sigma = \textstyle\sum_{i=1}^{n} x_i, \ (i = 1, \dots, n). \end{aligned} \tag{2.4}$$

In this case, Problem (EP) becomes the variational inequality

$$\text{Find } x^* \in U : \ \langle \tilde{Q}x^* + \mu - \bar{\alpha}, y - x^* \rangle \ge 0 \ \forall y \in U \tag{VI}$$

where

$$\tilde{Q} = \begin{pmatrix} 2\beta_1 & \beta_1 & \beta_1 & \dots & \beta_1 \\ \beta_2 & 2\beta_2 & \beta_2 & \dots & \beta_2 \\ \dots & \dots & \dots & \dots & \dots \\ \beta_n & \beta_n & \beta_n & \dots & 2\beta_n \end{pmatrix}.$$

Since $\beta_i > 0$ for all $i = 1, 2, \dots, n$ and the strategy set U is a box, this affine variational inequality (VI) is equivalent to the strongly convex quadratic program (see *e.g.* [19])

$$\min_{x \in U} \left\{ \frac{1}{2} x^T \hat{Q} x + c^T x \right\} \tag{QP}$$

where

$$\hat{Q} = \begin{pmatrix} 2 & 1 & 1 & \dots & 1 \\ 1 & 2 & 1 & \dots & 1 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 2 \end{pmatrix}$$

is symmetric positive definite and $c^T = (c_1, \dots, c_n)$ with $c_i = (\mu_i - \alpha_i)/\beta_i \quad (i = 1, \dots, n)$. (see *e.g.* [14, 19]).

Mathematical convex programming and monotone variational inequality approaches for determining an equilibrium point to equilibrium Cournot oligopolistic market models with convex cost functions can be found in [5, 16] respectively. A model with piecewise-concave cost functions is considered in [19]. In this case, the model can be formulated as a mixed variational inequality

$$\text{Find } x \in U : \langle \tilde{B}x - \alpha, y - x \rangle + \varphi(y) - \varphi(x) \geq 0, \forall y \in U,$$

with

$$\varphi(y) := y^T By + \sum_{i=1}^{n} h_i(y_i);\ \varphi(x) := x^T Bx + \sum_{i=1}^{n} h_i(x_i),$$

where $\tilde{B}$ is the matrix whose entries $\tilde{B}_{i,j} = \beta_i$ if $i \neq j$, $\tilde{B}_{i,i} = 0$ for all i, j, and B is the diagonal matrix whose ith diagonal entry is β_i $(i, j = 1, 2, \dots, n)$. In contrast to the convex cost function case, in this case, since φ is not convex, this mixed variational inequality is not convex and therefore a local equilibrium point may not be a global one. Algorithms for finding a local equilibrium point are proposed in [24].

Now we consider the model where the strategy set is a polyhedral convex set given by

$$D := \begin{cases} x \in U := U_1 \times U_2 \times \dots \times U_n, \\ Ax \leq b, \end{cases}$$

where

$$U_i := \begin{cases} x_i \in \mathbb{R} : 0 \leq x_i \leq d_i > 0 \ (i = 1, \dots, p \leq n); \\ x_i \in \mathbb{R} : 0 \leq x_i \ (i = p+1, \dots, n), \end{cases} \tag{2.5}$$

d_i $(i = 1, \dots, p)$ are scalars in $\mathbb{R}$.

For each $x \in D$, let

$$D_i(x) := \begin{cases} y_i \in U_i, \\ Ax^{y_i} \leq b, \end{cases}$$

where $x^{y_i} := (x_1, \dots, x_{i-1}, y_i, x_{i+1}, \dots, x_n)^T \in \mathbb{R}^n$, and

$$D(x) := D_1(x) \times D_2(x) \times \dots \times D_n(x).$$

Since $D(x)$ may not be contained in D for some $x \in D$, we take $C(x) := D(x) \cap D$ to ensure that C is a mapping from D to D. Moreover $x \in C(x)$ for every $x \in D$.

Similarly as in the classic model where the strategy set is a constant set, we call $x^* = (x_1^*, \dots, x_n^*)^T \in D$ a Nash-equilibrium point if

$$f_i(x_1^*, \dots, x_{i-1}^*, y_i, x_{i+1}^*, \dots, x_n^*) \le f_i(x_1^*, .., x_n^*),\ \forall y_i \in D_i(x^*),\ \forall i = 1, \dots, n. \tag{2.6}$$

In this case, by (2.2) and (2.3) the problem of finding an equilibrium point of the model can be formulated as the following quasi-equilibrium problem of the form (see *e.g.* [12])

$$\begin{cases} \text{find } \ x^* \in C(x^*) \text{ such that} \\ \varPhi(x^*, y) \ge 0,\ \forall y \in C(x^*). \end{cases} \tag{QEP}$$

3. A vector optimization model with nonindependent strategy sets

In this section we consider linear Cournot oligopolistic market models with nonindependent strategy sets by using a vector optimization approach rather than considering equilibrium Problem (QEP). In practice, this approach can be used when we are interested in the total profit with weights of all firms rather than the profit of each firm as in Nash equilibrium approach. Actually, in this case, there are often constraints for the productions $x_i,\ i = 1, 2, \dots, n$ of the firms which relate to subjects such as markets, investment of the Govermment and/or technical factors. In this paper, we assume that the constraints are described by a system of linear inequalities. The following small example shows that a point that gives the maximal total profit may or may not be an equilibrium point.

Example 3.1. Let the Cournot model with two firms is given by taking the strategy

$$D := \begin{cases} 0 \le x_1 \le 20;\ 0 \le x_2 \le 30, \\ 2x_1 + x_2 \le 50, \end{cases}$$

the linear cost functions

$$h_1(x_1) := 10x_1;\ h_2(x_2) := 12x_2$$

with $\beta^T = (\beta_1, \beta_2)$. Then profit functions are

$$f_1(x_1, x_2) = \big(12 - \beta_1\big(x_1 + x_2\big)\big)x_1 = -\beta_1 x_1^2 - \beta_1 x_1 x_2 + 2x_1$$

and

$$f_2(x_1, x_2) = \big(15 - \beta_2\big(x_1 + x_2\big)\big)x_2 = -\beta_2 x_2^2 - \beta_2 x_1 x_2 + 3x_2.$$

Let the weight vector $\lambda^T = (1, 1)$, then the total profit of the two firms is the quadratic form

$$G(x) := f_1(x) + f_2(x) = -\beta_1 x_1^2 - \beta_2 x_2^2 - (\beta_1 + \beta_2)x_1 x_2 + 2x_1 + 3x_2.$$

Then by definition, the Nikaido−Isoda bifunction is

$$\varPhi(x, y) = \langle \tilde{B}x + \mu - \alpha, y - x\rangle + y^T By - x^T Bx,$$

where

$$B = \begin{pmatrix} \beta_1 & 0 \\ 0 & \beta_2 \end{pmatrix}; \ \tilde{B} = \begin{pmatrix} 0 & \beta_1 \\ \beta_2 & 0 \end{pmatrix}.$$

If we choose $\beta_1 = 0.02$; $\beta_2 = 0.03$, the total profit function $G(x)$ attains its maximum on D at $x^* = (10, 30)^T$. Then by a simple calculation we have

$$C(x^*) = [0, 10] \times [0, 30]$$

and

$$g(x^*) := \min_{y \in C(x^*)} \Phi(x^*, y) = 0.$$

Thus

$$\Phi(x^*, y) \geq 0, \ \forall y \in C(x^*)$$

which shows that $x^* = (10, 30)^T$ is an equilibrium point of the model.

On the other hand, if we choose $\beta_1 = 0.02, \beta_2 = 0.04$, the total profit function $G(x)$ attains its maximum at $\bar{x} = (5, 30)^T$. In this case, by the same simple calculation as before we get

$$C(\bar{x}) = [0, 10] \times [0, 30]$$

Since

$$g(\bar{x}) := \min_{y \in C(\bar{x})} \Phi(\bar{x}, y) = -5.5 < 0,$$

$\bar{x}$ is not an equilibrium.

3.1. A jointly constrained bilinear programming formulation

As usual, for two vectors $x^T = (x_1, \ldots, x_n), y^T = (y_1, \ldots, y_n) \in \mathbb{R}^n$ we write $x \geq y$ if and only if $x_i \geq y_i$ for all $i = 1, \ldots, n$ and $x_j \neq y_j$ for some j.

In the Cournot market model under consideration in this section, we suppose that the price and the cost functions of all firms are affine and given respectively by

$$p_i(\sigma) := p_i\left(\sum_{j=1}^{n} x_j\right) = \alpha_i - \beta_i \sum_{j=1}^{n} x_j, \ \alpha_i \geq 0, \ \beta_i \geq 0 \ (i = 1, \ldots, n)$$

and by

$$h_i(x_i) = \mu_i x_i + \xi_i, \ \mu_i \geq 0, \ \xi_i \geq 0 \ (i = 1, \ldots, n).$$

In this case, the profit function of firm i $(i = 1, \ldots, n)$ can be rewritten as

$$f_i(x) = x_i p_i(x) - h_i(x_i) = -x^T C^i x + (\alpha_i - \mu_i) x_i - \xi_i, \tag{3.1}$$

where

$$C^i := \begin{pmatrix} 0 & 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \beta_i & \beta_i & \beta_i & \ldots & \beta_i \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 0 \end{pmatrix}.$$

Let

$$f(x) := (f_1(x), f_2(x), \dots, f_n(x)).$$

then $f(.)$ is a mapping from $\mathbb{R}^n$ to $\mathbb{R}^n$. Thus the vector optimization problem to this Cournot model can be written as

$$\max_{x\in D}\{f(x) = (f_1(x), f_2(x), \dots, f_n(x)\}. \tag{VP}$$

Recall [20, 27] that a point $x^* \in D$ is a Pareto efficient solution to (VP) (or Pareto efficient point of f on D) if whenever $x \in D$ and $f_i(x) \geq f_i(x^*)$ for every $i = 1, \dots, n$, then $f_i(x) = f_i(x^*)$ for every i. In what follows we are interested in finding an efficient Pareto solution to this vector optimization problem.

One efficient and commonly used technique for finding Pareto efficient solutions to vector optimization problem (VP) is scalarization (see *e.g.* [21]). It defines the scalarization function

$$G(\lambda, x) := \sum_{i=1}^{n} \lambda_i f_i(x) \tag{3.2}$$

where $\lambda_i > 0$ $(i = 1, \dots, n)$ are the weights. It is easy to see that if $\lambda_i > 0$ $(i = 1, \dots, n)$, then any optimal solution of the scalarized problem

$$\max\{G(\lambda, x) : x \in D\} \tag{$VP1$}$$

is a Pareto efficient point of f over D. In particular, if $\lambda_i = 1,\ \forall i = 1, \dots, n$, then $G(\lambda, x) := \sum_{i=1}^{n} f_i(x)$. Thus, an optimal solution of the function $\sum_{i=1}^{n} f_i(x)$ over D is just the maximum of the total profit of the model.

To easy exposition, in the sequel, we take

$$F(\lambda, x) := -G(\lambda, x).$$

Then, instead of problem (VP1), we can consider the problem

$$\min\{F(\lambda, x) : x \in D\} \tag{SVP}$$

Since the solution sets of these two problems coincide, from (3.1) and (3.2), we have

$$F(\lambda, x) = x^T Q x + (\mu(\lambda) - \alpha(\lambda))^T x + \xi,$$

where

$$\alpha(\lambda)^T = (\lambda_1\alpha_1, \dots, \lambda_n\alpha_n), \mu(\lambda)^T = (\lambda_1\mu_1, \dots, \lambda_n\mu_n), \xi = \sum_{i=1}^{n} \lambda_i\xi_i;$$

$$Q := \begin{pmatrix} \beta_1\lambda_1 & \beta_1\lambda_1 & \dots & \beta_1\lambda_1 \\ \beta_2\lambda_2 & \beta_2\lambda_2 & \dots & \beta_2\lambda_2 \\ \dots & \dots & \dots & \dots \\ \beta_n\lambda_n & \beta_n\lambda_n & \dots & \beta_n\lambda_n \end{pmatrix}.$$

Thus problem (SVP) is a quadratic program. Note that, since the matrix Q may not be positive semidefinite, problem (SVP) is not a convex program, and therefore finding a global optimal solution to the scalarized problem is a difficult task when the number of the variables is somewhat large (see [13, 17]). However, thanks to the specific structure of the matrix Q, we can globally solve (SVP) by formulating the problem as a jointly constrained bilinear program. For this purpose, let $t := \sum_{i=1}^n x_i$, then

$$f_i(x) = (\alpha_i - \beta_i t)x_i - \mu_i x_i - \xi_i,$$

and the scalarization function now becomes

$$F(\lambda, t, x) := \sum_{i=1}^n \lambda_i(\beta_i t + \mu_i - \alpha_i)x_i + \xi.$$

Thus the scalarized problem (SVP) can be rewritten as the following:

$$\min\left\{F(\lambda, t, x) := \sum_{i=1}^n \lambda_i(\beta_i t + \mu_i - \alpha_i)x_i\right\} \qquad (LSVP)$$

$$\text{subject to} \begin{cases} x_1 + x_2 + \ldots + x_n = t; \\ Ax \le b; \\ 0 \le x_j \le d_j, j = 1, \ldots, p; \\ x_j \ge 0, j = p+1, \ldots, n; \\ t \ge 0. \end{cases}$$

In this problem, (t, x) are variables and $\lambda_i > 0$ $(i = 1, \ldots, n)$ are fixed. It is easy to see that, if (t^*, x^*) is an optimal solution of problem (LSVP) then x^* is an optimal solution of problem (SVP).

For each fixed $t \ge 0$, problem (LSVP) is a linear program in the variable x which can be rewritten as

$$\min\left\{F(\lambda, t, x) := \sum_{i=1}^n \lambda_i(\beta_i t + \mu_i - \alpha_i)x_i\right\} \qquad (LSVP(t))$$

$$\text{subject to} \begin{cases} x_1 + x_2 + \ldots + x_n = t; \\ Ax \le b; \\ 0 \le x_j \le d_j, j = 1, \ldots, p; \\ x_j \ge 0, j = p+1, \ldots, n. \end{cases}$$

Theorem 3.2. *If either $\beta_i > 0$ or U_i is bounded for all $i = p+1, \ldots, n$, and the problem (LSVP) has a feasible solution, then it has an optimal solution, where U_i is given by* (2.5).

Proof. Denote by $\tilde{D}$ the set of feasible solutions to problem (LSVP). Assume that $\tilde{D}$ is a nonempty polyhedral convex set in $\mathbb{R}^{n+1}$. Suppose in contradiction that problem (LSVP) has no optimal solution, then there exists a sequence $\{(t_k, x^k)\} \subset \tilde{D}$ such that

$$\lim_{k\to\infty} F(\lambda, t_k, x^k) = -\infty.$$

By definition of F, there exists an index i_0 such that

$$p + 1 \leq i_0 \leq n, \lim_{k\to\infty} \lambda_{i_0}(\beta_{i_0} t_k + \mu_{i_0} - \alpha_{i0})x^k_{i_0} = -\infty.$$

Since $\lambda_{i_0} > 0$, by the assumption, there exists $k_0 \in N$ such that

$$(\beta_{i_0} t_k + \mu_{i_0} - \alpha_{i0}) < 0, \ \forall k > k_0$$

and

$$\lim_{k\to\infty} x^k_{i_0} = +\infty.$$

This is a contradiction, since if $\lim_{k\to\infty} x^k_{i_0} = +\infty$ then $\beta_{i_0} > 0$ and $\lim_{k\to\infty} t_k = +\infty$ implies

$$\lim_{k\to\infty} (\beta_{i_0} t_k + \mu_{i_0} - \alpha_{i0}) = +\infty. \qquad \square$$

Remark 3.3. The assumption that the production set U_i of firm i is bounded, when its reduced price coefficient $\beta_i = 0$ $(i = p + 1, \dots, n)$, is quite appropriate in practice. In fact, if it is not the case, the firm can produce an arbitrarily large production to obtain an arbitrarily large profit.

Corollary 3.4. *If either $\beta_i > 0$ or U_i is bounded for all $i = p+1, \dots, n$ and the feasible set D is not empty then the vector optimization problem (VP) has a Pareto efficient solution.*

3.2. Finding a Pareto efficient solution to the model

We suppose that the strategy set of the model is bounded and given by

$$D := \begin{cases} x \in U := U_1 \times U_2 \times \ldots \times U_n, \\ Ax \leq b, \end{cases}$$

where

$$U_i = \{x_i : 0 \leq x_i \leq d_i, \ i = 1, 2, \dots, n\}. \tag{3.3}$$

Let

$$T_{\max} := \max_{x\in D}\{x_1 + x_2 + \ldots + x_n\} \tag{3.4}$$

and

$$T_{\min} := \min_{x\in D}\{x_1 + x_2 + \ldots + x_n\}. \tag{3.5}$$

Obviously, by (3.3), if D is not empty, then $0 \leq T_{\min} \leq T_{\max} < +\infty$ and the scalar Problem (LSVP) can be rewritten as

$$\min \left\{ F(\lambda, t, x) := \sum_{i=1}^{n} \lambda_i(\beta_i t + \mu_i - \alpha_i)x_i \right\} \tag{$LSVP_1$}$$

$$\text{subject to} \begin{cases} x_1 + x_2 + \ldots + x_n = t; \\ Ax \leq b; \\ 0 \leq x_j \leq d_j, j = 1, \ldots, n; \\ T_{\min} \leq t \leq T_{\max}. \end{cases}$$

Obviously this problem always has an optimal solution whenever the strategy set $D \neq \emptyset$.

For an interval $I := [t_I^0, t_I^1] \subseteq [T_{\min}, T_{\max}]$, we denote by $\alpha(I)$ the optimal value of the problem

$$\alpha(I) := \min \left\{ F(\lambda, t, x) := \sum_{i=1}^{n} \lambda_i(\beta_i t + \mu_i - \alpha_i)x_i \right\} \tag{$LSVP_1(I)$}$$

$$\text{subject to} \begin{cases} x_1 + x_2 + \ldots + x_n = t; \\ Ax \leq b; \\ 0 \leq x_j \leq d_j, j = 1, \ldots, n; \\ t_I^0 \leq t \leq t_I^1. \end{cases}$$

We consider the relaxed problem

$$\beta(I) := \min_{(t,\tau,x)} \left\{ F(\lambda, \tau, x) := \sum_{i=1}^{n} \lambda_i(\beta_i \tau + \mu_i - \alpha_i)x_i \right\} \tag{$LSVP_2(I)$}$$

$$\text{subject to} \begin{cases} x_1 + x_2 + \ldots + x_n = t; \\ Ax \leq b; \\ 0 \leq x_j \leq d_j, j = 1, \ldots, n; \\ t_I^0 \leq t \leq t_I^1; t_I^0 \leq \tau \leq t_I^1. \end{cases}$$

Theorem 3.5.

(i) *For each fixed λ then problem $(LSVP_1(I))$ and $(LSVP_2(I))$ always have optimal solutions.*
(ii) *Suppose that (t^*, τ^*, x^*) is an optimal solution to problem $(LSVP_2(I))$. If $t^* = \tau^*$ then (t^*, x^*) also is an optimal solution to problem $(LSVP_1(I))$.*
(iii) *The optimal solution of problem $(LSVP_2(I))$ attains at a feasible solution $(\bar{t}, \bar{\tau}, \bar{x})$ with $\bar{\tau} = t_I^0$.*

Proof.

(i) is obvious, from the compactness of the feasible domains and continuity of F.
(ii) is obvious by the definition.

(iii) For each fixed $\tau \in I$, let

$$l(\tau) := \min_{(t,x)} \left\{ F(\lambda, \tau, x) := \sum_{i=1}^{n} \lambda_i(\beta_i\tau + \mu_i - \alpha_i)x_i \right\} \qquad (LSVP_\tau(I))$$

$$\text{subject to} \begin{cases} x_1 + x_2 + \ldots + x_n = t; \\ Ax \le b; \\ 0 \le x_j \le d_j, j = 1, \ldots, n; \\ t_I^0 \le t \le t_I^1. \end{cases}$$

From the $\beta_i \ge 0, x_i \ge 0, \forall i = 1, \ldots, n$, it follows that for every fixed $(t, x) \in D$, if $\tau_2 \ge \tau_1$ then $F(\lambda, \tau_2, x) \ge F(\lambda, \tau_1, x)$. Thus, $l(\tau)$ is nondecrease on I.

Let $\beta(I) := l(t_I^0)$, then $\beta(I)$ is a lower bound of $\alpha(I)$. □

Bisection Rule 1. Let $a < b$ be fixed scalars.

Step 0: Set $I_0 := [a, b],\ u_0 := a,\ v_0 := b,\ \mathit{\Gamma}_0 := \{I_0\},\ k \leftarrow 0$ and go to Step 1.
Step 1: Choose I_k any interval in $\mathit{\Gamma}_k$ and let $u_k < \xi_k \le v_k$ be an arbitrary point. Denote by t_k the midpoint of interval $[u_k, \xi_k]$.
Step 2: Set

$$I_k^- := [u_k, t_k],\ I_k^+ := [t_k, v_k],\ \mathit{\Gamma}_{k+1} := (\mathit{\Gamma}_k \setminus I_k) \cup \{I_k^-, I_k^+\}.$$

Let $k \leftarrow k + 1$ and go back to Step 1.

Lemma 3.6.

(i) *The bisection rule 1 generates an infinite nested subintervals $\{I_{k_j}\}$ of the interval $[a, b]$ such that $I_{k_{j+1}}$ is obtained from I_{k_j} by the above bisection rule.*
(ii) *The three corresponding subsequences $\{u_{k_j}\}$, $\{t_{k_j}\}$, $\{\xi_{k_j}\}$ convergence to the same limit point t_*.*

Proof.

(i) The first assertion is obvious from the bisection rule 1. For the sake of simple notation, we denote also the subsequence of subintervals by $\{I_k\}$.
(ii) Since $u_k < t_k < \xi_k$ for every iteration $k = 0, 1, 2, \ldots$, and $t_k = \frac{1}{2}(u_k + \xi_k)$, it is easy to see that if the conclusion of lemma is not true then

$$u_k \to u_*, t_k \to t_*, \xi_k \to \xi_* \text{ as } k \to +\infty$$

and $u_* < t_* < \xi_*$, which implies that there exists k_0 such that

$$u_k < t_{k_0} < \xi_k, \forall k > k_0.$$

At $k := k_0 + 1$, according to the bisection rule 1, if the interval I_k is obtained from I_{k_0} then $I_k = I_{k_0}^-$ or $I_k = I_{k_0}^+$.

First consider the case when $I_k = I_{k_0}^-$. By the algorithm $\xi_k = \xi_{k_0+1} < t_{k_0}$, which is a contradiction.

If $I_k = I_{k_0}^+$ then $u_k = u_{k_0+1} = t_{k_0}$ is again a contradiction. □

Using the above bisection rule we can describe the main algorithm as follows.

Algorithm 3.1.
Initialization: Compute $T_{\max}, T_{\min}$ by solving problems (3.4) and (3.5). Set $I_0 := [T_{\min}, T_{\max}]$ and compute $\beta_0 := \beta(I_0)$. Denote by (t_0, x^0) the obtained optimal solution of problem $(LSVP_2(I_0))$. Let $\alpha_0 := F(\lambda, t_0, x^0)$.

Set $\varGamma_0 := \{I_0\}, k \leftarrow 0$ and go to Step 1.

Iteration k with $k = 0.1\ldots$

Step 1:

1a) If $\alpha_k \leq \beta_k$, then terminate; x^k is an optimal solution.
1b) If $\alpha_k > \beta_k$, then define

$$I_k^- := [t^0_{I_k}, \xi_k]; \quad I_k^+ := [\xi_k, t^1_{I_k}]$$

where ξ_k is the midpoint of the interval I_k.

Step 2: Compute $\beta(I_k^+)$ and $\beta(I_k^-)$. Let $(\bar{t}_{k+1}, \bar{x}^{k+1})$ be the currently best feasible point obtained as computation of $\beta(I_k^+)$ and $\beta(I_k^-)$, and let

$$\begin{aligned} \alpha_{k+1} &:= \min\{\alpha_k, F(\lambda, \bar{t}_{k+1}, \bar{x}^{k+1})\} = F(\lambda, t_{k+1}, x^{k+1}), \\ \varDelta_k &:= (\varGamma_k \setminus I_k) \cup \{I_k^+, I_k^-\}, \\ \varGamma_{k+1} &:= \{I \in \varDelta_k : \beta(I) < \alpha_{k+1}\}. \end{aligned}$$

Step 3: Set $I_{k+1} \in \varGamma_{k+1}$ such that

$$\beta(I_{k+1}) = \min\{\beta(I) : I \in \varGamma_{k+1}\}.$$

Let $\beta_{k+1} := \beta(I_{k+1}),\ k \leftarrow k+1$ and go back to Step 1.

This completes the description of the algorithm.

Remark 3.7.

(1) For each interval $I = [t_I^0, t_I^1]$, to compute $\beta(I)$ we have to solve one linear program

$$\min\left\{F(\lambda, t_I^0, x) := \sum_{i=1}^n \lambda_i(\beta_i t_I^0 + \mu_i - \alpha_i)x_i\right\}$$

$$\text{subject to} \begin{cases} x_1 + x_2 + \ldots + x_n = t; \\ Ax \leq b; \\ 0 \leq x_j \leq d_j, j = 1, \ldots, n; \\ t_I^0 \leq t \leq t_I^1. \end{cases}$$

(2) The midpoint bisection used in the above algorithm does not take into account the iteration point obtained through the bounding operation. Here we give a branching bisection which uses the iteration point as well as the objective function. Suppose that we are in case 1b) of Step 1 with $I_k := [t^0_{I_k}, t^1_{I_k}]$. Let $(\tilde{t}_k, \tilde{x}^k)$ be the optimal solution for the problem of computing $\beta(I_k)$.
Define
$$I_k^- := [t^0_{I_k}, \xi_k], \quad I_k^+ := [\xi_k, t^1_{I_k}]$$
where ξ_k is the midpoint of interval $[t^0_{I_k}, \tilde{t}_k]$. We will refer to this branching as an adaptive bisection.

(3) If we are interested only in an ϵ-solution, then at each iteration k, an interval I can be deleted from Γ_k if
$$|\alpha_k - \beta(I)| \le \epsilon \max\{|\alpha_k|, 1\}.$$
In particular, if $|\alpha_k - \beta_k| \le \epsilon \max\{|\alpha_k|, 1\}$, then we call (λ, t_k, x^k) an ϵ-solution, and α_k is an ϵ-optimal value of problem (LSVP).

We are now turning to the convergence of the above algorithm. Let F_* denote the optimal value of the problem (LSVP). From the definition of β_k, α_k and x^k it follows that $\beta_k \le \beta_{k+1} \le F_* \le \alpha_{k+1} \le \alpha_k$ for all k. Hence if the algorithm terminates at some iteration k, *i.e.* $\alpha_k \le \beta_k$, then $\beta_k = F_* = \alpha_k = F(\lambda, t_k, x^k)$. Thus x^k is an optimal solution. If the algorithm does not terminate, then we have the following convergence result:

Convergence of the algorithm. $\alpha_k \searrow F_*$, $\beta_k \nearrow F_*$ and any cluster point of $\{x^k\}$ is an optimal solution of problem (LSVP).

Proof. Since $\{\beta_k\}$ is nondecreasing and $\{\alpha_k\}$ is non-increasing, $\beta_* := \lim \beta_k$, $\alpha_* := \lim \alpha_k$ exist.

From Lemma 3.6 one has
$$t^0_k \to t_*, \ \xi_k \to t_*, \ \tilde{t}_k \to t_* \text{ as } k \to +\infty.$$
Let
$$\bar{\alpha_k} := F(\lambda, \tilde{t}_k, \tilde{x}^k),$$
where $(\tilde{t}_k, \tilde{x}^k)$ is the optimal solution of the problem for computing $\beta(I_k)$. Obviously, for all k, $\alpha_k \le \bar{\alpha_k}$ and
$$\bar{\alpha_k} - \beta_k = (\tilde{t}_k - t^0_{I_k}) \sum_{i=1}^n \lambda_i \beta_i \tilde{x}^k \le T\bar{\beta}(\tilde{t}_k - t^0_{I_k}),$$
where $\bar{\beta} := \max\{\beta_i : i = 1, \ldots, n\}$. This implies that $\bar{\alpha_k}$ converges to β_*. Note that for all k, one has
$$\beta_k \le \alpha_k \le \bar{\alpha_k}.$$
Thus, α_k converges to $\alpha_* = \beta_* = F_*$.

Now let x^* be any cluster point of $\{x^k\}$. For simplicity of notation, we may suppose that $x^k \to x^*$. By continuity of F, we have

$$\alpha_k = F(\lambda, t_k, x^k) \to F(\lambda, t_*, x^*) = F_* \text{ as } k \to +\infty. \qquad \square$$

An illustrative example. To illustrate the algorithm we consider a Cournot model with three firms ($n = 3$). The price, cost functions, strategy set for each firm and the constraint matrix A are given as

$$\begin{cases} p_1(x) := 14.5 - 0.02(x_1 + x_2 + x_3), & h_1(x_1) := 8.2x_1, \quad U_1 := [0, 30]; \\ p_2(x) := 16.4 - 0.04(x_1 + x_2 + x_3), & h_2(x_2) := 10.7x_2, \quad U_2 := [0, 40]; \\ p_3(x) := 17.2 - 0.01(x_1 + x_2 + x_3), & h_3(x_3) := 9.4x_3, \quad U_3 := [0, 50]; \\ A := \begin{pmatrix} 2 & 1 & 1 \\ 3 & -1 & 1 \\ -1 & -1 & 0 \end{pmatrix}, \; b := \begin{pmatrix} 90 \\ 60 \\ -20 \end{pmatrix}. & \end{cases}$$

In this example

$$\alpha = (14.5, 16.4, 17.2); \; \beta = (0.02, 0.04, 0.01); \; \mu = (8.2, 10.7, 9.4).$$

Let $\lambda := (3, 2, 5)$ be the weight vector. In this case, problem (LSVP(t)) takes the form

$$\min\{F(\lambda, t, x) = (0.06t - 18.9)x_1 + (0.08t - 11.4)x_2 + (0.05t - 39)x_3\}$$

$$\text{subject to} \begin{cases} x_1 + x_2 + x_3 = t; \\ 2x_1 + x_2 + x_3 \le 90; \\ 3x_1 - x_2 + x_3 \le 60; \\ -x_1 - x_2 \le -20; \\ 0 \le x_1 \le 30, 0 \le x_2 \le 40, 0 \le x_3 \le 50, 20 \le t \le 90. \end{cases}$$

Choose $\epsilon = 0.005$. Now, we use the algorithm to find an ϵ-solution to this problem.

Initialization: Compute $T_{\min}$ by solving the linear program

$$\min\{x_1 + x_2 + x_3\}$$

$$\text{subject to} \begin{cases} 2x_1 + x_2 + x_3 \le 90; \\ 3x_1 - x_2 + x_3 \le 60; \\ -x_1 - x_2 \quad \le -20; \\ 0 \le x_1 \le 30, 0 \le x_2 \le 40, 0 \le x_3 \le 50. \end{cases}$$

we obtain $T_{\min} = 20$.

Compute $T_{\max}$ by solving the linear program

$$\max\{x_1 + x_2 + x_3\}$$

$$\text{subject to} \begin{cases} 2x_1 + x_2 + x_3 \le 90; \\ 3x_1 - x_2 + x_3 \le 60; \\ -x_1 - x_2 \quad \le -20, \\ 0 \le x_1 \le 30, 0 \le x_2 \le 40, 0 \le x_3 \le 50 \end{cases}$$

we have $T_{\max} = 90$. Let $I_0 := [20, 90], \mathit{\Gamma}_0 := \{I_0\}$. Compute

$$\beta_0 = \beta(I_0) = -2292,\ \alpha_0 = \alpha(I_0) = -1893,\ \mathit{\Gamma}_0 := \{I_0\}.$$

Iteration 0:
Step 1:
1b) Bisect the interval $I_0 = [20, 90]$ to obtain

$$I_0^- := [20, 55],\ I_0^+ := [55, 90].$$

Step 2: Compute:

$$\beta(I_0^-) = -1614.9,\ \alpha(I_0^-) = -1505.5$$

$$\beta(I_0^+) = -2108.5,\ \alpha(I_0^+) = -1991.$$

Let

$$\alpha_1 := \alpha(I_0^+) = -1991,\ x^1 = (10, 20, 50),\ \mathit{\Delta}_0 = \{I_0^-, I_0^+\},\ \mathit{\Gamma}_1 = \{I_0^+\}.$$

Step 3: Set

$$I_1 := I_0^+ = [55, 90],\ \beta_1 := \beta(I_0^+) = -2108.5$$

Iteration 1:
Step 1:
1b) Bisect the interval $[55, 90]$ to get

$$I_1^- := [55, 72.5],\ I_1^+ := [72.5, 90].$$

Step 2: Compute:

$$\beta(I_1^-) = -2039.9,\ \alpha(I_1^-) = -1967.5$$

$$\beta(I_1^+) = -2026.2,\ \alpha(I_1^+) = -1991.$$

Let

$$\alpha_2 := \alpha(I_1^+) = -1991, x^2 = (10, 20, 50),\ \mathit{\Delta}_1 = \{I_1^-, I_1^+\},\ \mathit{\Gamma}_2 = \{I_1^-, I_1^+\}.$$

Step 3: Set

$$I_2 := I_1^- = [55, 72.5],\ \beta_2 := \beta(I_1^-) = -2039.9,$$

In iteration 2, at Step 1 we further bisect the interval $[55, 72.5]$ to get

$$I_2^- := [55, 63.75],\ I_2^+ := [63.75, 72.5],$$

and the algorithm proceeds similarly. At iteration 7 we have

$$\alpha_8 = -1991,\ x^8 = (10, 20, 50)^T, \beta_8 = -1998.6.$$

Since $|\alpha_8 - \beta_8| < \epsilon \max\{|\alpha_8|, 1\}$, we terminate the algorithm at iteration 8 to obtain $x^* := x^8 = (10, 20, 50)$, an ϵ- optimal solution to the problem (SVP), which is an efficient point of the model. At this efficient solution the total profit of three firms is 447.

TABLE 1.

Size		Algorithm 3.1		Interior-point		
n	m	Average time	Average iter.	Average time	Average iter	Glob
50	10	0.38	222.0	0.06	51.4	8
100	20	6.53	126.5	0.19	102.2	9
150	30	9.37	125.7	0.50	153.9	10
200	30	6.68	82.3	0.99	203.2	10
250	50	15.37	86.2	1.86	254.9	9
300	50	31.69	150.5	3.42	306.4	8
400	30	16.84	121.3	8.22	406.6	10
500	30	21.52	132.5	17.07	506.3	7
500	100	61.90	42.2	17.25	506.1	10
500	200	98.54	20.5	17.55	509.0	9
600	30	20.85	97.0	43.03	607.6	10
700	30	22.15	101.6	50.63	707.8	8
850	20	19.22	111.9	99.35	856.4	10
1000	50	175.95	317.9	177.89	1013.2	10
1200	20	32.57	131.5	342.49	1205.7	9

4. COMPUTATIONAL RESULTS AND EXPERIMENTS

For evaluation, the algorithm was written on Matlab and tested on different groups of problems. Each group contains 10 problems of the same size (n, m), but having input data generated randomly. For each problem we suppose that the upper bound for every x_i is randomly generated in the interval $[100, 500]$ and the data input are the followings:

- λ: $1 \times n$-vector whose elements are positive real numbers that sum to 10;
- α: $1 \times n$-vector of integer numbers in the interval [20, 30];
- β: $1 \times n$-vector of real numbers in the interval [0.01, 0.05];
- μ: $1 \times n$-vector of integer numbers in the interval [10, 20];
- A: $m \times n$-matrix of integer numbers in the interval [0, 20];
- B: $m \times 1$-vector of integer numbers in the interval [500, 5000].

The program runs using Matlab version 7.11 installed on a Windows 7 machine, RAM 6Gb, CPU Intel Core2 Duo 2.26Ghz. To evaluate the proposed algorithm we compared with the interior-point algorithm for quadratic programming using the quadprog implementation of Matlab. Both algorithms run on the same data set which was randomly generated. For each size $m \times n$ of the matrix A, the program runs with 10-matrices whose entries are randomly generated. For Algorithm 3.1, we terminate the program at iteration k, whenever the difference between the currently best upper and lower bounds satisfies the condition $\alpha_k - \beta_k \le \epsilon \max\{1, |\alpha_k|\}$ with $\epsilon = 10^{-4}$

The obtained results are reported in Tables 1 and 2 where we use the following headings:

- n: the number of firms;
- m: the number of the rows of matrix A;

Table 2.

Size	Algorithm 3.1			Interior-point	
m	Average time	Average iter.	Store	Average time	Average iter.
10	8.44	54	53	81.69	817
30	22.71	76	32	81.14	817
50	26.17	43	14	124.92	820
70	45.54	42	12	140.52	821
100	56.11	30	14	82.37	824

- *Average time*: the average time (in second) needed to solve one problem;
- *Average iter*: the average numbers of iterations for one problem.
- *Glob*: the number of problems for which a global optimal solution was obtained by the interior algorithm.
- *Store*: the maximum number of intervals to be stored.

From Table 1 one can conclude that, for the running time, Algorithm 3.1 works well when m is small or median (less than 50), but n may be much larger. In contrast, for the interior point algorithm, m may be large, but the running time increases quickly as n- gets larger. Furthermore, we emphasize that for the proposed branch-and-bound algorithm the obtained solutions are always global ones, while for the interior algorithms, the obtained solutions may not be global ones. In fact, for 150-tested problems, by using the obtained local solution as the starting point to run again the program, about 90 percentage of the obtained solutions are global. Following a suggestion of one referee we have tested Algorithm 3.1 with ten problems for each $m = 10, 30, 50, 70, 100$ with the same $n = 800$. The obtained results presented in Table 2 below show that Algorithm 3.1 really is sensitive with m as the referee remarked.

5. Conclusion

In this paper we have used a vector-optimization approach to Cournot oligopolistic market equilibrium models having jointly constrained strategy set. By introducing an extra variable, the problem of finding an efficient Pareto point for generalized Cournot oligopolistic market models has been converted into a jointly constrained bilinear program. We have proposed a branch-and-bound algorithm for globally solving the latter problem. By employing the specific structure of the model, the branching operation can perform adaptively in one- dimensional intervals that makes the algorithm efficient. We have compared the proposed algorithm with the interior algorithm using the quadprog implementation of Matlab. Computational results on a lot of randomly generated problems show that the proposed algorithm works well when the number of the joint constraints is median, the number of the variables may be much more larger.

Acknowledgements. This work was supported in part by the National Foundation for Science and Technology Development (NAFOSTED). The author would like to thank the referees for their useful comments, remarks and suggestions which improved very much the paper.

References

[1] F.A. Al-Khayyal, Jointly constrained bilinear programs and related problems: An overview. *Comput. Math. Appl.* **19** (1990) 53–62.

[2] D. Aussel, R. Correa and M. Marechal, Gap functions for quasivariational inequalities and generalized Nash equilibrium problems. *J. Optim. Theory Optim.* **151** (2011) 474–488.

[3] G. Bigi, M. Castellani, M. Pappalardo and M. Passacantando, Existence and solution methods for equilibria. *Eur. J. Oper. Res.* **227** (2013) 1–11.

[4] E. Blum and W. Oettli, From optimization and variational inequality to equilibrium problems. Princeton University Press, 1963. *Math. Stud.* **63** (1994) 127–149.

[5] F. Facchinei and J.S. Pang, Finite-Dimensional Variational Inequalities and Complementarity Problems. Springer, Berlin (2003).

[6] F. Facchinei and C. Kanzow, Generalized Nash equilibrium problems. *Ann. Oper. Res.* **175** (2010) 177–211.

[7] F. Facchinei, C. Kanzow and S. Sagratella, Solving quasi-variational inequalities via their KKT conditions. *Math. Program.* **144** Ser. A (2014) 369–412.

[8] M. Fukushima, A class of gap functions for quasi-variational inequlity problems. *J. Ind. Manag. Optim.* **3** (2007) 165–174.

[9] M. Fukushima and J.S. Pang, Quasi-variational inequality, generalized Nash equilibria, and multi-leader-folower games. *Comput. Manag. Sci.* **2** (2005) 21–26.

[10] R. Gupta and A. Mehra, Gap functions and error bounds for quasi-variational inequalities. *J. Global Optim.* **53** (2012) 737–748.

[11] N. Harms, T. Hoheisel and C. Kanzow, On a smooth dual gap function for a class of quasi-variational inequalities. *J. Optim. Theory Appl.* **163** (2014) 413–438.

[12] N. Harms, C. Kanzow and O. Stein, Smoothless properties of a regularized gap function for quasi-variational inequalities. *Optim. Methods Softw* **29** (2014) 720–750.

[13] R. Horst and H. Tuy, Global Optimization, Deterministic Approach. Springer, Berlin (1990).

[14] I. Konnov, Combined Relaxation Methods for Variational Inequalities. Springer, Berlin (2001).

[15] K. Kubota and M. Fukushima Gap function approach to the generalized Nash equilibrium problem. *J. Optim. Theory Appl.* **144** (2010) 511–531.

[16] H.F. Murphy, H.D. Sherali and A.L. Soyster, A mathematical programming approach for determining oligopolistic market equilibrium. *Math. Program.* **24** (1982) 92–106.

[17] L.D. Muu and W. Oettli, A method for minimizing a convex-concave function over a convex set. *J. Optim. Theory Appl.* **70** (1990) 377–384.

[18] L.D. Muu and N.V. Quy, A global optimization method for solving convex quadratic bilevel programming problems. *J. Global Optim.* **26** (2003) 199–219.

[19] L.D. Muu, V.H. Nguyen and N.V. Quy, On Nash−Cournot oligopolistic market models with concave cost functions. *J. Global Optim.* **41** (2007) 351–364

[20] D.T. Luc, Theory of Vector Optimization. Lecture Notes in Economics and Mathematical Systems. Springer-Verlag, New York Berlin, Heidelberg (1989).

[21] D.T. Luc, T.Q. Phong and M. Volle, Scalarizing Functions for Generating the Weakly Efficient Solution Set in Convex Multiobjective Problems. *SIAM J. Optim.* **15** (2005) 987–1001.

[22] A. Nagurney, Network Economics: A Variational Inequality Approach. Kluwer Academic Publishers (1993).

[23] H. Nikaido and K. Isoda, Note on non-cooperative convex games. *Pacific J. Math.* **5** (1955) 807–815.

[24] T.D. Quoc and L.D. Muu, A splitting proximal point method for Nash−Cournot equilibrium models involving nonconvex cost functions. *J. Nonlin. Convex Anal.* **12** (2011) 519–534.

[25] N.V. Quy, A vector optimization approarch to Cournot oligopolistic market Models. *Int. J. Optim.: Theory, Methods Appl.* **1** (2009) 341–360.

[26] J.J. Strodiot, T.T. V. Nguyen and V.H. Nguyen, A new class of hybrid extragradient algorithms for solving quasi- equilibrium problems. *J. Global Optim.* **56** (2013) 373–397.

[27] P.L. Yu, Multiple − Criteria Decision Making: Concepts, Techniques and Extensions. Plenum Press, New York, London (1985).