Optim Lett (2017) 11:1357–1370
DOI 10.1007/s11590-016-1081-x

ORIGINAL PAPER

On a linearization technique for solving the quadratic set covering problem and variations

Pooja Pandey[1] · **Abraham P. Punnen**[1]

Received: 7 October 2015 / Accepted: 9 September 2016 / Published online: 15 September 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract In this paper we identify various inaccuracies in the paper by Saxena and Arora (Optimization 39:33–42, 1997). In particular, we observe that their algorithm does not guarantee optimality, contrary to what is claimed. Experimental analysis has been carried out to assess the value of this algorithm as a heuristic. The results disclose that for some classes of problems the Saxena–Arora algorithm is effective in achieving good quality solutions while for some other classes of problems, its performance is poor. We also discuss similar inaccuracies in another related paper.

Keywords 0-1 programming · Quadratic programming · Agorithms · Set covering problem · Heuristic

1 Introduction

The set covering problem is well studied in the operations research literature [2,3,5,8,13]. Most of the works on the problem reported in the literature have a linear objective function. Bazaraa and Goode [3] introduced the quadratic set covering problem (QSP) and proposed a cutting plane algorithm to solve it. Adams [1] and Liberti [14] proposed linearization techniques for binary quadratic programs. Since QSP is a binary quadratic programming problem, these linearization techniques can be used to formulate QSP

✉ Pooja Pandey
poojap@sfu.ca

Abraham P. Punnen
apunnen@sfu.ca

1 Department of Mathematics, Simon Fraser University, 250-13450 102nd Avenue, Surrey, BC V3T 0A3, Canada

as a 0–1 integer linear program. QSP is known to be NP-hard and polynomial time approximation algorithms are also available [7] to solve special classes of this problem.

Saxena and Arora [17] studied QSP and discussed various structural properties of the problem along with a linearization algorithm which is claimed to produce an optimal solution. The notion of linearization used in [17] is different from the concept of "linearization" used by Adams [1] and Liberti [14] and also different from what is discussed in [6,12,16].

In this paper, we show that the properties of QSP established in [17] are incorrect and that the algorithm they proposed need not produce an optimal solution. Gupta and Saxena [10] extended the results of [17] to the quadratic set packing and partitioning problems. These extensions also suffer from the same drawbacks as that of [17] and the algorithm in [10] could also produce a non-optimal solution, contrary to what is claimed. Since the algorithm of [17] is not guaranteed to produce an optimal solution, it will be interesting to examine its value as a heuristic. Our experimental analysis discloses that the algorithm of [17] produces good solutions for some classes of problems while it produces very poor solutions for other classes.

2 The quadratic set covering problem

Let $I = \{1, 2, \dots, m\}$ be a finite set and $P = \{P_1, P_2, \dots, P_n\}$ be a family of subsets of I. The index set for the elements of P is denoted by $J = \{1, 2, \dots, n\}$. For each element $j \in J$, a cost c_j is prescribed and for each element $(i, j) \in J \times J$, a cost d_{ij} is also prescribed. We refer to c_j the *linear cost* of the set P_j and $\mathbf{c} = (c_1, \dots, c_n)$ the *linear cost vector*. Similarly d_{ij} is referred to as the *quadratic cost* corresponding to the ordered pair (P_i, P_j) and the matrix $\mathbf{D} = (d_{ij})_{n\times n}$ is referred to as the *quadratic cost matrix*.

A subset V of J is said to be a *cover* of I, if $\cup_{j\in V} P_j = I$. Then the *linear set covering problem* (LSP) is to find a cover $L = \{\pi(1), \dots, \pi(l)\}$ such that $\sum_{i=1}^{l} c_{\pi(i)}$ is minimized. Likewise the *quadratic set covering problem* (QSP) is to select a cover $L = \{\sigma(1), \dots, \sigma(l)\}$ such that $\sum_{i=1}^{l} c_{\sigma(i)} + \sum_{i=1}^{l}\sum_{j=1}^{l} d_{\sigma(i)\sigma(j)}$ is minimized.

For each $i \in I$, consider the vector $\mathbf{a}_i = (a_{i1}, a_{i2}, \dots, a_{in})$ where

$$a_{ij} = \begin{cases} 1 & \text{if } i \in P_j \\ 0 & \text{otherwise.} \end{cases}$$

and $\mathbf{A} = (a_{ij})_{m\times n}$ be an $m \times n$ matrix. Also, consider the decision variables $x_1, x_2, \dots, x_n$ where

$$x_j = \begin{cases} 1 & \text{if set } P_j \text{ is selected} \\ 0 & \text{otherwise.} \end{cases}$$

The vector of decision variables is represented by $\boldsymbol{x} = (x_1, \dots, x_n)^T$ and $\mathbf{1}$ is a column vector of size m where all entries are equal to 1. Then the LSP and QSP can be formulated respectively as 0–1 integer programs

$$\begin{aligned} \text{LSP:} \quad & \text{Maximize } \mathbf{cx} \\ & \text{Subject to } \mathbf{Ax} \geq \mathbf{1} && (1) \\ & \mathbf{x} \in \{0, 1\}^n && (2) \end{aligned}$$

and

$$\begin{aligned} QSP: \quad & \text{Minimize } \mathbf{c}\boldsymbol{x} + \mathbf{x}^T \mathbf{D}\boldsymbol{x} \\ & \text{Subject to } \mathbf{A}\boldsymbol{x} \geq \mathbf{1} && (3) \\ & \boldsymbol{x} \in \{0, 1\}^n && (4) \end{aligned}$$

As indicated in [17] the continuous relaxation of QSP, denoted by $QSP^{'}$, is obtained by replacing the constraint $\boldsymbol{x} \in \{0, 1\}^n$ by $\boldsymbol{x} \geq \mathbf{0}$, where $\mathbf{0}$ is the zero vector of size n. The family of feasible solutions of both LSP and QSP is denoted by $\bar{S} = \{\boldsymbol{x} | A\boldsymbol{x} \geq \mathbf{1}, \boldsymbol{x} \in \{0, 1\}^n\}$.

The following definitions are taken directly from [17]. Any $\boldsymbol{x} \in \bar{S}$ is called a *cover solution* and an optimal solution to the underlying problem (LSP or QSP) is called an *optimal cover solution*. Note that each cover solution corresponds to a cover and vice versa. A *cover* V is said to be redundant if $V - \{j\}$ for $j \in V$ is also a *cover*. A cover which is not *redundant* is called a *prime cover*. The incidence vector $\boldsymbol{x}$ that corresponds to a *prime cover* is called a *prime cover solution*.

Garfinkel and Nemhauser [8] proved that if the objective function in LSP has a finite optimal value then there exists a prime cover solution for which this value is attained whenever $\mathbf{c} \geq \mathbf{0}$.

Saxena and Arora claimed an extension of this result to $QSP^{'}$, assuming $\mathbf{c} \geq \mathbf{0}$ and $\mathbf{D}$ is symmetric and positive semi-definite. More precisely, they claimed:

Theorem 1 (Theorem 3 of [17]) If the objective function in $QSP^{'}$ has finite optimal value then there exists a prime cover solution where this value is attained.

This result however is not true as indicated by the following example. Let

$$\mathbf{c} = (0, 0, 0), \ \mathbf{A} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \text{ and } \mathbf{D} = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}$$

Note that $\mathbf{D}$ is a symmetric and positive semi-definite matrix. For the QSP and $QSP^{'}$ with $\mathbf{A}$, $\mathbf{D}$ and $\mathbf{c}$ defined as above, it can be verified that $\boldsymbol{x}^* = (1, 1, 1)^T$ is an optimal solution with the objective function value zero for both problems. The optimal *cover* corresponding to $\boldsymbol{x}^*$ is $V^* = \{1, 2, 3\}$ which is a *redundant cover* since $V^* - \{2\} = \{1, 3\}$ is also a cover. All other cover solutions and their respective objective function values are listed below:

$$\boldsymbol{x}^1 = (1, 0, 1)^T \text{ redundant cover solution} \qquad f(\boldsymbol{x}^1) = 1$$
$$\boldsymbol{x}^2 = (1, 1, 0)^T \text{ redundant cover solution} \qquad f(\boldsymbol{x}^2) = 1$$

$$\boldsymbol{x}^3 = (0, 1, 1)^T \text{ prime cover solution} \qquad f(\boldsymbol{x}^3) = 2$$
$$\boldsymbol{x}^4 = (1, 0, 0)^T \text{ prime cover solution} \qquad f(\boldsymbol{x}^4) = 2$$

None of these corresponds to an optimal solution for QSP or $QSP^{'}$. In particular, no prime cover solution is optimal for the instances of QSP and $QSP^{'}$ constructed above, contradicting Theorem 1. This example also shows that Theorem 1 cannot be corrected by replacing $QSP^{'}$ with QSP in the theorem.

We now show that a variation of Theorem 1 is true, which relaxes the requirement of **D** being positive semi-definite while sign restrictions are imposed on its elements. This is summarized in our next theorem.

Theorem 2 *There always exists a prime cover optimal solution for QSP if* **c** *and* **D** *are non-negative.*

Proof Let $\boldsymbol{x}^0 \in \bar{S}$ be an optimal solution of QSP. Then the corresponding optimal objective function value is

$$f(\boldsymbol{x}^0) = \mathbf{c}\boldsymbol{x}^0 + \boldsymbol{x}^{0^T}\mathbf{D}\boldsymbol{x}^0$$

Let J_o be the cover corresponding to the solution $\boldsymbol{x}^0$. If J_o is a prime cover then statement of the theorem is correct. Otherwise we can construct a prime cover, let say J_1, from J_o by dropping the redundant columns. Let $\boldsymbol{x}^1$ be the solution of QSP with respect to the prime cover J_1 and

$$f(\boldsymbol{x}^1) = \mathbf{c}\boldsymbol{x}^1 + \boldsymbol{x}^{1^T}\mathbf{D}\boldsymbol{x}^1.$$

Since **c** and **D** are non-negative and $J_1 \subset J_0$,

$$f(\boldsymbol{x}^0) \geq f(\boldsymbol{x}^1)$$

Since $\boldsymbol{x}^0$ is an optimal solution to QSP, $f(\boldsymbol{x}^0) = f(\boldsymbol{x}^1)$ and the and the result follows. □

The family of feasible solutions for continuous relaxations of LSP and QSP is represented by $S = \{\boldsymbol{x} | A\boldsymbol{x} \geq \mathbf{1}, \boldsymbol{x} \geq \mathbf{0}\}$. The continuous relaxation of LSP is denoted by $LSP^{'}$.

Saxena and Arora [17] also proposed an algorithm to solve QSP and claimed that it will produce an optimal solution. Their algorithm is re-stated here.

The Saxena–Arora algorithm for QSP

Step 1: From the QSP, construct the corresponding $QSP^{'}$
Step 2: Choose a feasible solution $\mathbf{x}^0 \in S$ such that $\nabla f(\boldsymbol{x}^0) \neq \mathbf{0}$ and form the corresponding linear programming problem $LSP^{'}$ as

$$LSP^{'} \quad \text{Minimize}_{\boldsymbol{x}\in S} \nabla f(\boldsymbol{x}^0)^T\boldsymbol{x}. \tag{5}$$

On solving $(LSP^{'})$, let $\boldsymbol{x}^1$, be its optimal solution. Let $S^1 = \{\boldsymbol{x}^1\}$.

Step 3: Starting with the point $\boldsymbol{x}^1$, form the corresponding $LSP^{'}$, and let its optimal solution be $\boldsymbol{x}^2 \neq \boldsymbol{x}^1$. Update S^1 i.e. $S^1 = \{\boldsymbol{x}^1, \boldsymbol{x}^2\}$.
Step 4: Repeat Step 3 for the point $\boldsymbol{x}^2$, and suppose at the ith stage $S^1 = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^i\}$. Stop, if at the $(i+1)^{th}$ stage $\boldsymbol{x}^{i+1} \in S^1$, then $\boldsymbol{x}^{i+1}$, is the optimal solution of $QSP^{'}$.
Step 5: If $\boldsymbol{x}^{i+1}$ is an optimal solution of the form 0 or 1 then it is a solution of QSP otherwise, go to **Step 6**.
Step 6: Apply Gomory cuts to find a solution of the 0 or 1 form and the corresponding prime cover.

The algorithm discussed above suffers from various drawbacks as listed below.

1. Even if $\mathbf{c} \geq \mathbf{0}$, and $\mathbf{D}$ is symmetric and positive semi-definite, the $LSP^{'}$ in Step 2 could be unbounded and hence it need not have an optimal solution for all instances.
2. Suppose that we apply the algorithm only for instances where $LSP^{'}$ in Step 2 is bounded in all iterations. Even then, the solution produced in Step 4 could be non-optimal to $QSP^{'}$.
3. If the algorithm terminates in Step 5 the resulting solution could be non-optimal to QSP.
4. If the algorithm successfully moves to Step 6, then also the solution produced could be non-optimal.

We now illustrate each of the drawbacks discussed above using counterexamples.

1. Since $\mathbf{D}$ is a positive semi-definite matrix and $\mathbf{c} \geq \mathbf{0}$, the objective function value of $QSP^{'}$ is bounded below by zero. However, $LSP^{'}$ in Step 2 or in Step 3 need not be bounded below. Let

$$\mathbf{c} = (0, 0, 0, 0),\ \mathbf{A} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \text{ and } \mathbf{D} = \begin{pmatrix} 10 & -3 & -4 & -4 \\ -3 & 2 & 1 & 1 \\ -4 & 1 & 3 & 1 \\ -4 & 1 & 1 & 3 \end{pmatrix}$$

Note that $\mathbf{D}$ is symmetric and positive semi-definite. Consider the instance of QSP with the above values for $\mathbf{c}$, $\mathbf{D}$, and $\mathbf{A}$. Starting with the feasible solution $\boldsymbol{x}^0 = (1, 0, 0, 0)^T \in S$ of $QSP^{'}$, we get the $LSP^{'}$ in Step 2 as

$$\begin{aligned} \text{Minimize} \quad & \nabla f(\boldsymbol{x}^0)^T \boldsymbol{x} = 20x_1 - 6x_2 - 8x_3 - 8x_4 \\ \text{Subject to:} \quad & x_1 + x_2 \geq 1 \\ & x_1 + x_3 \geq 1 \\ & x_1 + x_4 \geq 1 \\ & x_j \geq 0 \quad \text{for } j = 1, 2, 3, 4. \end{aligned}$$

This problem is unbounded. Thus the algorithm can not be applied in this case. The immediate conclusion is that the Saxena–Arora [17] algorithm is potentially applicable only to those QSP instances where the resulting $LSP^{'}$ is bounded in every step.

2. The algorithm can fail in Step 4. The Saxena–Arora algorithm claims to produce an optimal solution of QSP' in Step 4 but this may not be true always. Consider the data

$$\mathbf{c} = (0,0,0,0),\ \mathbf{A} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \text{ and } \mathbf{D} = \begin{pmatrix} 10 & 2 & 2 & 2 \\ 2 & 3 & 1 & 1 \\ 2 & 1 & 3 & 1 \\ 2 & 1 & 1 & 4 \end{pmatrix}$$

Note that **D** is symmetric and positive semi-definite. Consider the instance of QSP with the above values for **c**, **D**, and **A**, we get the QSP as

$$\begin{aligned} \text{Min } f(\boldsymbol{x}) &= 10x_1^2 + 3x_2^2 + 3x_3^2 + 4x_4^2 + 4x_1x_2 + 4x_1x_3 + 4x_1x_4 + 2x_2x_3 + 2x_2x_4 + 2x_3x_4 \\ \text{st: } & x_1 + x_2 \geq 1 \\ & x_1 + x_3 \geq 1 \\ & x_1 + x_4 \geq 1 \\ & x_j \in \{0,1\} \text{ for } j = 1,2,3,4. \end{aligned}$$

and

$$\begin{aligned} \nabla f(\boldsymbol{x}) = & (20x_1 + 4x_2 + 4x_3 + 4x_4, 6x_2 + 4x_1 + 2x_3 + 2x_4, \\ & 6x_3 + 4x_1 + 2x_2 + 2x_4, 8x_4 + 4x_1 + 2x_2 + 2x_3)^T \end{aligned}$$

Select the feasible solution $\boldsymbol{x}^0 = (1,0,0,0)^T \in S$ of QSP'. Construct the LSP' with respect to $\boldsymbol{x}^0$, the objective function of LSP' is

$$\nabla f(\boldsymbol{x}^0)^T \boldsymbol{x} = 20x_1 + 4x_2 + 4x_3 + 4x_4$$

Note that $\boldsymbol{x}^1 = (0,1,1,1)^T$ is an optimal solution to this LSP'. Thus, we set $S^1 = \{\boldsymbol{x}^1\}$. Now, using $\boldsymbol{x}^1$ construct the new LSP', and the optimal solution to this LSP' is $\boldsymbol{x}^2 = (1,0,0,0)^T$. Since $\boldsymbol{x}^2 \notin S^1$, construct the new LSP' , the optimal solution to this LSP' is $\boldsymbol{x}^3 = (0,1,1,1)^T$. Since $\boldsymbol{x}^3 \in S^1$, in Step 4 the algorithm concludes that $\boldsymbol{x}^3$ is an optimal solution of QSP' with objective function value 16. However, $\boldsymbol{x}^* = (0.714286, 0.285714, 0.285714, 0.285714)^T$ which is a better solution for QSP', contradicting the optimality of x^3. Thus the algorithm could fail in Step 4.

3. As per the Saxena–Arora algorithm, Step 5 produces an optimal solution to QSP' and if this optimal solution is binary, they claim this solution to be an optimal solution of QSP. We now show that a binary solution produced in Step 5 need not be an optimal solution to QSP. For example.
Consider the data

$$\mathbf{c} = (0,0,0,0),\ \mathbf{A} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \text{ and } \mathbf{D} = \begin{pmatrix} 4 & 1 & 1 & 1 \\ 1 & 2 & 0 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 0 & 0 & 2 \end{pmatrix}$$

Note that $\mathbf{D}$ is a symmetric, positive semi-definite and non-negative. As noted in Theorem 2, a prime cover optimal solution exists for this QSP. But still the Saxena–Arora algorithm fails to produce an optimal solution for QSP. Consider the instance of QSP with the above values for $\mathbf{c}$, $\mathbf{D}$, and $\mathbf{A}$.

Select the feasible solution $\boldsymbol{x}^0 = \left(\frac{3}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)^T \in S$ which is also an optimal solution of QSP'. Construct the LSP' with respect to $\boldsymbol{x}^0$ and the objective function is $\nabla f(\boldsymbol{x}^0)^T\boldsymbol{x} = \frac{15}{2}x_1 + \frac{5}{2}x_2 + \frac{5}{2}x_3 + \frac{5}{2}x_4$.

$\boldsymbol{x}^1 = (0, 1, 1, 1)^T$ is an optimal solution to this LSP'. Thus, we set $S^1 = \{\boldsymbol{x}^1\}$. Now, using $\boldsymbol{x}^1$, construct the new LSP' with the objective function as $\nabla f(\boldsymbol{x}^1)^T\boldsymbol{x} = 6x_1 + 4x_2 + 4x_3 + 4x_4$. An optimal solution to this LSP' is $\boldsymbol{x}^2 = (1, 0, 0, 0)^T$. Since $\boldsymbol{x}^2 \notin S^1$, we update $S^1 = \{\boldsymbol{x}^1, \boldsymbol{x}^2\}$. Starting with $\boldsymbol{x}^2$, construct the LSP' with the objective function $\nabla f(\boldsymbol{x}^2)^T\boldsymbol{x} = 8x_1 + 2x_2 + 2x_3 + 2x_4$. An optimal solution to this LSP' is $\boldsymbol{x}^3 = (0, 1, 1, 1)^T$. Since $\boldsymbol{x}^3 \in S^1$, the algorithm concludes that $\boldsymbol{x}^3$ is an optimal solution of QSP'. Since $\boldsymbol{x}^3$ contains 0 and 1 entries only, as per the algorithm, it is an optimal solution to QSP and the corresponding objective function value is 6.

However $\boldsymbol{x}^* = (1, 0, 0, 0)^T$ is a better solution to the QSP with objective function value $f(\boldsymbol{x}^*) = 4$. Thus, the solution produced by the the Saxena–Arora algorithm for the above instance of QSP is not optimal.

In the previous example if $\boldsymbol{x}^0 = (0, 1, 1, 1)^T$ is selected instead of $\left(\frac{3}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)^T$, the algorithm produces $x_1 = (1, 0, 0, 0)$, $x_2 = (0, 1, 1, 1)$, and $x_3 = (1, 0, 0, 0)$, leading to an accurate optimal solution $x_3 = (1, 0, 0, 0)$ to QSP. Note that $x_0 = (0, 1, 1, 1)^T$ and $(1, 0, 0, 0)^T$ are alternate optimal solutions of LSP' with the objective function $\nabla f(\boldsymbol{x}^0)^T\boldsymbol{x} = \frac{15}{2}x_1 + \frac{5}{2}x_2 + \frac{5}{2}x_3 + \frac{5}{2}x_4$. It is easy to show that trouble of the Saxena–Arora algorithm is not because of the presence of alternate optimal solutions, leading to a choice in selection. This can be demonstrated with the same example but by selecting a different starting point as given below.

Select the feasible solution $\boldsymbol{x}^0 = \left(1, \frac{1}{2}, 0, 0\right)^T \in S$ of QSP'. Construct the LSP' with respect to $\boldsymbol{x}^0$ and the objective function is $\nabla f(\boldsymbol{x}^0)^T\boldsymbol{x} = 9x_1 + 4x_2 + 2x_3 + 2x_4$. $\boldsymbol{x}^1 = (0, 1, 1, 1)^T$ is the unique optimal solution to this LSP' (easily verifiable by enumerating the basic feasible solutions). Thus, we set $S^1 = \{\boldsymbol{x}^1\}$. Now, using $\boldsymbol{x}^1$, construct the new LSP' with the objective function as $\nabla f(\boldsymbol{x}^1)^T\boldsymbol{x} = 6x_1 + 4x_2 + 4x_3 + 4x_4$. The unique optimal solution to this LSP' is $\boldsymbol{x}^2 = (1, 0, 0, 0)^T$. Since $\boldsymbol{x}^2 \notin S^1$, we update $S^1 = \{\boldsymbol{x}^1, \boldsymbol{x}^2\}$. Starting with $\boldsymbol{x}^2$, construct the LSP' with the objective function $\nabla f(\boldsymbol{x}^2)^T\boldsymbol{x} = 8x_1 + 2x_2 + 2x_3 + 2x_4$. The unique optimal solution to this LSP' is $\boldsymbol{x}^3 = (0, 1, 1, 1)^T$. Since $\boldsymbol{x}^3 \in S^1$, the algorithm concludes that $\boldsymbol{x}^3$ is an optimal solution of QSP'. Since $\boldsymbol{x}^3$ contains 0 and 1 entries only, as per the algorithm, it is an optimal solution to QSP and the corresponding objective function value is 6. The solution produced by the the Saxena–Arora algorithm for the above instance of QSP is not optimal.

4. As per the Saxena–Arora algorithm, Step 5 produces an optimal solution to QSP' and if this optimal solution is not binary, the algorithm proceeds to Step 6 where Gomory cuts are applied to find a solution which they claim to be an optimal

solution to QSP. We now show that Step 6 need not produce an optimal solution to QSP even if the solution produced in Step 4 is optimal for $QSP^{'}$.

In point 3 we gave a counterexample where the solution is a basic feasible solution (BFS) to $LSP^{'}$ which is binary but not optimal to $QSP^{'}$. Note that $QSP^{'}$ is a continuous quadratic problem and an optimal solution need not correspond to an extreme point. We now illustrate that if the $LSP^{'}$ solver works with any solution (such as interior point methods) and not necessarily with BFS (as in simplex method) it may be possible to get an optimal solution to $QSP^{'}$ in Step 4. For example

Consider the instance of QSP from the previous case. $\boldsymbol{x}^0 = \left(\frac{3}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)^T \in S$ is an optimal solution of $QSP^{'}$. Construct the $LSP^{'}$ with respect to $\boldsymbol{x}^0$ and the resulting objective function is $\nabla f(\boldsymbol{x}^0)^T\boldsymbol{x} = \frac{15}{2}x_1+\frac{5}{2}x_2+\frac{5}{2}x_3+\frac{5}{2}x_4$. The algorithm produces $x_1 = \left(\frac{3}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)^T$, $x_2 = \left(\frac{3}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)^T$, leading to an accurate optimal solution $x_2 = (\frac{3}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ to $QSP^{'}$ and the algorithm successfully completes Step 4.

To apply Gomory cut,first reduce the non-basic feasible solution (non-BFS) to a basic feasible solution (BFS). From the previous example, the optimal non-BFS $\left(\frac{3}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)$ of $LSP^{'}$ to an optimal BFS $\boldsymbol{x}^1 = (0, 1, 1, 1)^T$ of $LSP^{'}$. Since this is binary , no cutting plane will be added and a Gomory cut phase terminates with the non-optimal solution $\boldsymbol{x}^1 = (0, 1, 1, 1)^T$ of QSP.

Alternatively if we do not reduce the non-BFS to a BFS to apply Gomory cuts, but use any Integer programming (IP) solver to compute an optimal integer solution to $LSP^{'}$ we could still get non-optimal solution. For example: Solving the $LSP^{'}$ at $\left(\frac{3}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)$ for 0-1 optimal solution we could get $\boldsymbol{x}^1 = (0, 1, 1, 1)^T$ as the optimal 0-1 solution of $LSP^{'}$. This is not an optimal solution to QSP. (We note that the paper [17] does not say anything about the use of general IP solver; but we mentioned it here for the clarity and completeness).

3 The quadratic set packing and partitioning problems

A subset H of J is said to be a *pack* of I if $\bigcup_{j\in H} P_j = I$, and for $j, k \in H$, $j \neq k$, implies $P_j \bigcap P_k = \emptyset$. Then the *linear set packing problem* (LSPP) is to select a *pack* $V = \{\pi(1), \ldots, \pi(v)\}$ such that $\sum_{i=1}^{v} c_{\pi(i)}$ is maximized. Likewise, the *quadratic set packing problem* (QSPP) is to select a *pack* $L = \{\sigma(1), \ldots, \sigma(l)\}$ such that $\sum_{i=1}^{l} c_{\sigma(i)} + \sum_{i=1}^{l}\sum_{j=1}^{l} d_{\sigma(i)\sigma(j)}$ is maximized.

Let $\mathbf{A} = (a_{ij})_{m\times n}$ be as defined in Sect. 2. Also, consider the decision variables $x_1, x_2, \ldots, x_n$ where

$$x_j = \begin{cases} 1 & \text{if } j \text{ is in the pack} \\ 0 & \text{otherwise.} \end{cases}$$

The vector of decision variables is represented as $\boldsymbol{x} = (x_1, \ldots, x_n)^T$. Then the LSPP and QSPP can be formulated respectively as 0–1 integer programs

$$\text{LSPP:}\quad \text{Maximize } \mathbf{c}\boldsymbol{x}$$

$$\text{Subject to } \mathbf{A}\boldsymbol{x} \leq \mathbf{1} \tag{6}$$

$$\boldsymbol{x} \in \{0, 1\}^n \tag{7}$$

and

$$\text{QSPP:}\quad \text{Maximize } \mathbf{c}\boldsymbol{x} + \boldsymbol{x}^T\mathbf{D}\mathbf{x}$$

$$\text{Subject to } \mathbf{A}\boldsymbol{x} \leq \mathbf{1} \tag{8}$$

$$\boldsymbol{x} \in \{0, 1\}^n \tag{9}$$

The continuous relaxations of LSPP and QSPP, denoted respectively by LSPP(C) and QSPP(C), are obtained by replacing the constraint $\boldsymbol{x} \in \{0, 1\}^n$ by $\boldsymbol{x} \geq \mathbf{0}$, respectively in LSPP and QSPP.

The family of feasible solutions of both LSPP and QSPP is denoted by $S = \{\boldsymbol{x}|A\boldsymbol{x} \leq \mathbf{1}, \boldsymbol{x} \in \{0, 1\}^n\}$ and the family of feasible solutions for their continuous relaxations is denoted by $\bar{S} = \{\boldsymbol{x}|A\boldsymbol{x} \leq \mathbf{1}, \boldsymbol{x} \geq \mathbf{0}\}$.

Following are some definitions given in [10]. A solution $\boldsymbol{x} \in S$ which satisfies (8) and (9) is said to be a *pack solution*. For any *pack* V, a column of $\mathbf{A}$ corresponding to $j^* \in V$ is said to be redundant if $V - \{j^*\}$ is also a *pack*. If a *pack* corresponds to one or more redundant columns, it is called a *redundant pack*. A *pack* V^* is said to be a *prime pack*, if none of the columns corresponding to $j^* \in V^*$ is *redundant*. A solution corresponding to the *prime pack* is called a *prime packing solution*.

From the definition of a redundant column given above (as in [10]), zero vector is the only prime packing solution for the set packing problem. Thus the results of [10] are incorrect with respect to their definitions. We believe the "$-$" sign in the above definition of redundant column discussed in [10] is a typo and it is probably supposed to be "$\cup''$ which is consistent with the definitions given in [11] by the same authors. Hereafter, we use this modified definition.

Thus, for any *pack* V, a column of $\mathbf{A}$ corresponding to $j \in J$ is said to be redundant if $V \cup \{j\}$ is also a pack. If a *pack* contains one or more redundant columns, it is called a *redundant pack*. A *pack* V^* is said to be a *prime pack*, if none of the columns corresponding to $j \in J$ is *redundant*. A solution corresponding to the *prime pack* is called a *prime packing solution*.

Gupta and Saxena [10] assumed $\mathbf{D}$ to be a negative semi-definite matrix and extended most of the results for QSP in [17] to QSPP. In particular, they claimed that:

Theorem 3 (Theorem 2 of [10]) If the objective function in QSPP has finite value then there exists a prime packing solution where this value is attained.

Because of the definition of the prime pack solution given by Gupta and Saxena [10], a prime pack is always a zero vector hence the theorem is given incorrect. The theorem is still incorrect even if we use the modified definition [11] which is indicated above.

For example, consider an instance of QSPP with

$$\mathbf{c} = (0, 0, 0),\ A = \begin{bmatrix} 1\ 1\ 0 \\ 1\ 0\ 1 \end{bmatrix},\ D = \begin{bmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -2 \end{bmatrix}$$

Note that **D** is symmetric and negative semi-definite.

$\boldsymbol{x}^* = (0, 0, 0)^T$ is $\boldsymbol{x}$QSPP with the objective function value zero. We list below all prime pack solutions with the objective function values.

$$\begin{aligned} \boldsymbol{x}^1 &= \{1, 0, 0\} \text{ prime pack solution} \quad f(\boldsymbol{x}^1) = -2 \\ \boldsymbol{x}^2 &= \{0, 1, 1\} \text{ prime pack solution} \quad f(\boldsymbol{x}^2) = -2 \end{aligned}$$

Note that none of these solutions are optimal.

We now show that a variation of Theorem 3 is true and this is summarized in our next theorem.

Theorem 4 *There always exists a prime pack optimal solution for QSPP if all elements of* **c** *and* **D** *are non-negative.*

Proof Let $\boldsymbol{x}^0 \in \bar{S}$ be an optimal solution of QSPP. Then the corresponding optimal objective function value is

$$f(\boldsymbol{x}^0) = \mathbf{c}\boldsymbol{x}^0 + \boldsymbol{x}^{0^T}\mathbf{D}\boldsymbol{x}^0$$

Let J_o be the pack corresponding to the solution $\boldsymbol{x}^0$. If J_o is a prime pack then we are done. Otherwise we can construct a prime pack, let say J_1, from J_o by adding the redundant columns. Let $\boldsymbol{x}^1$ be the solution of QSP with respect to the prime pack J_1 and

$$f(\boldsymbol{x}^1) = \mathbf{c}\boldsymbol{x}^1 + \boldsymbol{x}^{1^T}\mathbf{D}\boldsymbol{x}^1.$$

Since J_1 obtained by adding redundant columns to J_0, therefore, $J_0 \subset J_1$. When all elements of **c** and **D** are non-negative, or

$$f(\boldsymbol{x}^0) \leq f(\boldsymbol{x}^1)$$

Since $\boldsymbol{x}^0$ is an optimal solution to QSPP, $f(\boldsymbol{x}^0) = f(\boldsymbol{x}^1)$ and the proof follows. □

Along the same lines as in [17], the authors of [10], provide a solution algorithm for QSPP. Following the insight generated in our counter examples in Sect. 2, and by the above observation, it is not difficult to construct counter examples to show that the algorithm of [10] need not provide an optimal solution for QSPP.

If in Eq. (8) we replace constraints $\mathbf{A}\boldsymbol{x} \leq \mathbf{1}$ with $\mathbf{A}\boldsymbol{x} = \mathbf{1}$, then QSPP changes into quadratic set partitioning problem. Gupta and Saxena [10] proposed a similar algorithm for the quadratic set partitioning problem, which has similar issues as in the quadratic set packing problem. We omit the discussion about the quadratic set partitioning problem.

4 Computational results

Since the algorithm of [17] is not guaranteed to be optimal, it would be interesting to examine its value as a heuristic to solve QSP. We have conducted some preliminary experimental analysis to assess the value of the Saxena–Arora algorithm as a heuristic using different classes of test problems.

The test data was taken from standard benchmark problems for the set covering problem [2,4,9], and the vertex covering problem [18], with appropriate amendments to incorporate quadratic objective. In this class, we took only small size instances since the quadratic problem is much more difficult and time consuming to solve compared to their linear counterparts. We have also generated some quadratic vertex cover instances on random graphs taken from [15]. We divided computational experiments into two different categories, with each $\mathbf{c} \geq 0$, while in category 1: $\mathbf{D}$ is a positive semi-definite matrix and in category 2: $\mathbf{D}$ is non-negative and positive semi-definite.

Each element of the linear cost vector $\mathbf{c}$ is a random integer from the interval [3,5]. Since the quadratic cost matrix $\mathbf{D}$ is positive semi-definite, there exists a square matrix $\mathbf{B}$ such that $\mathbf{D} = \mathbf{B}\mathbf{B}^T$. This $\mathbf{D}$ is generated by a random square matrix $\mathbf{B}$ where each element of $\mathbf{B}$ is a random integer between -10 and 10. When $\mathbf{D}$ is non-negative and positive semi-definite, each element of $\mathbf{B}$ is selected as a random integer between 0 and 20.

The Saxena–Arora algorithm was coded in C++ and tested on a PC with windows 7 operating system, Intel 3770 i7 3.40 GHz processor and with 16 GB of RAM. We also used CPLEX 0-1 integer quadratic solver (version 12.5) to compute exact (heuristic) solutions. For each instance that we tested, we set CPLEX time limit to be the same as the time taken by Saxena–Arora algorithm and also run CPLEX by doubling this running time. These two implementations provide heuristic solutions and were compared with the solution produced by the Saxena–Arora algorithm.

In the tables, t_1 is the cpu time taken by Saxena–Arora algorithm. The column "CPLEX Sol (t_1)" represents the heuristic solution obtained by CPLEX by fixing its running time to t_1 and the column "CPLEX Sol ($2t_1$)" represents CPLEX run with $2t_1$ upper bound on the execution time. The column "negative entries in Q" provides percentage of negative entries in the matrix $\mathbf{D}$. The column "Sol" refers the objective function values. CPLEX quadratic solver takes more time to solve QSP to optimality when $\mathbf{D}$ is positive semi-definite compare to the instances when $\mathbf{D}$ is non-negative and positive semi-definite. Therefore, Table 1 reports lower bound value and Table 2 reports optimal solution value.

When $\mathbf{D}$ is a random positive semi-definite matrix, Table 1 shows that the Saxena–Arora algorithm does not return a good quality solution for QSP. Note that a general purpose solver like CPLEX obtained much better solutions within the same time limit for the test problems used. But when $\mathbf{D}$ is non-negative and positive semi-definite, Table 2 shows that the Saxena–Arora algorithm produced solutions as good as those

Table 1 Benchmark instances, **D** is positive semi-definite

Problem	Size		Lower bound on opt	Saxena Algo.		CPLEX Sol (t_1)	CPLEX Sol ($2t_1$)	Negative entries in D (%)
	m	n		Cpu time t_1 (s)	Sol			
Qscpcyc06	192	240	48	156	241,189	71	70	43.59
Qscpcyc07	448	672	112	125	1,486,218	192	187	46.00
Qscp41	1000	200	432.27	78	7,743,176	455	455	44.48
Qscpe3	500	50	3.307	63	1,777,128	10	10	46.46
Qscpe4	500	50	3.455	62	1,638,640	10	10	45.69
Qscpe5	500	50	3.393	47	1,918,480	28	15	45.69
Qgraph50-01	612	50	111.75	156	144	122	122	41.04
Qgraph50-02	490	50	93.6684	172	641	124	124	42
Qgraph50-03	735	50	101.25	156	153	152	152	40.23
Qgraph50-04	612	50	87.9275	156	1057	149	149	45.6
Qgraph50-05	490	50	108.25	156	127	124	124	45.6
Qgraph50-06	857	50	89.71486	187	261	154	154	41.6
Qgraph50-07	735	50	85.1275	156	1738	119	119	48
Qgraph50-08	612	50	89.1389	156	1030	100	100	44
Qgraph50-09	980	50	89.6389	156	600	146	146	47.52
Qgraph50-10	612	50	92.5273	156	2363	142	142	48.72
Qfrb30-15-1	17,827	450	810.306	11,185	5106	1494	1494	42.56
Qfrb30-15-2	17,874	450	799.139	7909	18,906	1479	1479	44.59
Qfrb30-15-3	17,809	450	799.139	6100	11,497	1478	1477	44.59
Qfrb30-15-4	17,831	450	787.806	5274	21,930	1475	1475	44.46
Qfrb30-15-5	17,794	450	799.139	5616	14,032	1478	1475	44.59

Table 2 Benchmark instances, **D** is non-negative and positive semi-definite

Problem	Size		Optimal sol	Saxena Algo.		CPLEX Sol (t_1)	CPLEX Sol ($2t_1$)
	m	n		Cpu time t_1 s	Sol		
Qscpcyc06	192	240	147,523	483	147,523	147,523	147,523
Qscpcyc07	448	672	726,070	780	726,070	726,070	726,070
Qscp41	1000	200	7271	343	7271	7271	7271
Qscpe3	500	50	7	390	3369	7	7
Qscpe4	500	50	8	515	3726	8	8
Qscpe5	500	50	7	359	1854	7	7
Qgraph50-01	612	50	5149	187	5149	5149	5149
Qgraph50-02	490	50	5108	156	5108	5108	5108
Qgraph50-03	735	50	5163	188	5163	5163	5163
Qgraph50-04	612	50	9272	156	9272	9272	9272
Qgraph50-05	490	50	4168	125	4168	4168	4168
Qgraph50-06	857	50	8042	124	8042	8042	8042
Qgraph50-07	735	50	7088	124	7088	7088	7088
Qgraph50-08	612	50	4670	109	4670	4670	4670
Qgraph50-09	980	50	8872	141	8872	8872	8872
Qgraph50-10	612	50	6614	125	6614	6614	6614

produced by CPLEX for many instances. For vertex cover instances the Saxena–Arora algorithm produced an optimal solution. For the set cover instances CPLEX produces better solutions than the Saxena–Arora algorithm. Thus, for **D** is non-negative and positive semi-definite, the Saxena–Arora algorithm could be used as a heuristic to solve QSP. As our counter example indicates, even for this class the Saxena–Arora algorithm need not produce an optimal solution.

Acknowledgements This work was supported by an NSERC discovery Grant awarded to Abraham Punnen.

References

1. Adams, W.P., Sherali, H.D.: A tight linearization and an algorithm for zero-one quadratic programming problems. Manag. Sci. **32**, 1274–1290 (1986)
2. Balas, E., Ho, A.: Set covering algorithms using cutting planes, heuristics, and sub-gradient optimization: a computational study. Math. Progr. **12**, 37–60 (1980)
3. Bazaraa, M.S., Goode, J.J.: A cutting-plane algorithm for the quadratic set-covering problem. Oper. Res. **23**, 150–158 (1975)
4. Beasley, J.E.: A lagrangian heuristic for set-covering problems. Naval Res. Logist. **37**, 151–164 (1990)
5. Bector C.R., Bhatt S.K.: A linearization technique for solving integral linear fractional program: Proc. fifth Manitoba Conference on Numerical Mathematics, pp. 221–229 (1975)
6. Custic A., Punnen A.P.: A characterization of linearizable instances of the quadratic minimum spanning tree problem. arXiv:1510.02197 (2015)
7. Escoffier, B., Hammer, P.L.: Approximation of the quadratic set covering problem. Discrete Optim. **4**, 378–386 (2007)
8. Garfinkel, R.S., Nemhauser, G.L.: Integer Programming. Wiley, New York (1972)
9. Grossman, T., Wool, A.: Computational experience with approximation algorithms for the set covering problem. Eur. J. Oper. Res. **101**, 81–92 (1997)
10. Gupta, R., Saxena, R.R.: Linearization technique for solving quadratic set packing and partitioning problems. Int. J. Math. Comput. Appl. Res. **4**, 9–20 (2014)
11. Gupta, R., Saxena, R.R.: Set packing problem with linear fractional objective function. Int. J. Math. Comput. Appl. Res. **4**, 9–18 (2014)
12. Kabadi, S.N., Punnen, A.P.: An $O(n^4)$ algorithm for the QAP linearization problem. Math. Oper. Res. **36**, 754–761 (2011)
13. Lemke, C.E., Salkin, H.M., Spielberg, K.: Set covering by single branch enumeration with linear programming sub-problem. Oper. Res. **19**, 998–1022 (1971)
14. Liberti L.: Compact linearization for binary quadratic problems. 4OR 5:231–245 (2007)
15. Periannan, M.: An ant-based algotithm for the minimum vertex cover problem: Thesis. The Pennsylvania State University, Master of Science (2007)
16. Punnen, A.P., Kabadi, S.N.: A linear time algorithm for the Koopmans-Beckman QAP linearization and related problems. Discrete Optim. **10**, 200–209 (2013)
17. Saxena, R.R., Arora, S.R.: A linearization technique for solving the quadratic set covering problem. Optimization **39**, 33–42 (1997)
18. Xu, K.: http://www.nlsde.buaa.edu.cn/~kexu/benchmarks/graph-benchmarks.htm