A Computational Study of Solution Techniques For 0-1 Quadratic Programs

Noah Hunt-Isaak

July 17, 2018

1 Introduction

In this study, we will computationally determine the most efficient linearization techniques for solving various classes of 0-1 quadratic programs with the general form:

$$\text{QP: Max } \left\{ \sum_{i=1}^{n} c_i x_i + \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} C_{ij} x_i x_j : x \in \boldsymbol{X}, x \text{ binary} \right\},$$

where n is the number of variables x_i, c and C are coefficient matrices, and $\boldsymbol{X}$ is a set of linear constraints. There are a number of problem classes that fit this structure, which we will outline in the subsequent sections.

1.1 QKP

The most well known QP problem class is the quadratic knapsack problem (QKP), where $\boldsymbol{X}$ represents a capacity constraint given by:

$$\text{QKP: } \boldsymbol{X} \equiv \left\{ \boldsymbol{x} \in \mathbb{R}^n : \sum_{i=1}^{n} a_i x_i \leq b \right\}.$$

The QKP was first introduced in 1980 by Gallo et al., and can be applied naturally to a number of real world scenarios in operations research, statistics and combinatorics [1]. When generating QKP problem instances for our computational study, the objective coefficient values c_i and C_{ij} were taken from a uniform distribution over [1,100]. The coefficients a_i were taken from a uniform distrubtion over [1,50] and the capacity constraint b from [50, $\sum_{i=1}^{n} a_i$]. These are all standard ranges used by nearly every computation study of the QKP. We can assume without loss of generality that $C_{ii} = 0 \, \forall i$

1.2 KQKP

A subset of the QKP is the exact k-item quadratic knapsack problem (KQKP),which has an additional constraint explicitly restricting the number of items that can be taken as follows:

$$\text{KQKP: } \boldsymbol{X} \equiv \left\{ \boldsymbol{x} \in \mathbb{R}^n : \sum_{i=1}^{n} a_i x_i \leq b, \text{ AND } \sum_{i=1}^{n} x_i = k \right\}.$$

The KQKP along with a heuristic for solving it was first considered by Letocart et al. in 2014 and then again with exact solution techniques in 2016 [2, 3]. These problems were generated in the same way as the QKP, except the range used for b was $[50, 30k]$ with k from the range $[2, n/4]$.

1.3 UQP

There is also a version of the QP called the unconstrained quadratic problem where $\boldsymbol{X} = \emptyset$. Note that this would be trivial to solve if all of the c, C values were the same sign and thus we only consider instances containing mixed sign. We generate our UQP problem instances in the manner of the boolean least squares application [4]. The UQP also has applications in statistical mechanics, clustering, project selection and graph theory [5].

1.4 HSP

The heaviest k-subgraph problem (HSP) is concerned with determining a block of k nodes of a weighted graph such that the total edge width within the subgraph induced by the block is maximized [6]. The HSP can be formulated as

$$\text{HSP: Max } \left\{ \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} C_{ij} x_i x_j : \sum_{i=1}^{n} x_i = k, x \text{ binary} \right\}.$$

The number of items to be taken, k, is set to be either $n/4, n/2$ or $3n/4$, and all coefficients C_{ij} are either 0 or 1 to indicate whether a node should be included or not.

1.5 QSAP

Another QP type is the Quadratic Semi-Assignment Problem which is formulated as follows:

$$\text{QP: Min } \left\{ \sum_{i=1}^{m} \sum_{k=1}^{n} e_{ik} x_{ik} + \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \sum_{k=1}^{n} \sum_{l=1}^{n} C_{ikjl} x_{ik} x_{jl} \right.$$
$$\left. : \sum_{k=1}^{n} x_{ik} = 1 (i = 1, ..., m), x \text{ binary} \right\}.$$

The coefficients e_{ik} and C_{ikjl} are taken from a distribution across [-50,50]. e_{ik} represents the execution cost of task T_i when assigned to processor P_k [7].

2 Linearizations

2.1 Standard Linearization

Directly solving any of these problem types is difficult because of the nonlinear terms. Thus, a common technique is to reformulate the QP as a linear program (LP) using auxillary variables and constraints, which can then easily be solved with commercial solvers such as CPLEX and Gurobi. The simplest way to do this is with the standard linearization from Glover and Woolsey [8]. Their strategy is to replace all of the quadratic terms $x_i x_j$ with a single continuous variable w_{ij} to obtain the new objective function:

$$\text{STD: Max } \sum_{i=1}^{n} c_i x_i + \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} C_{ij} w_{ij}$$

The following set of auxillary constraints must be added in order to ensure that $w_{ij} = x_i x_j$:

$$w_{ij} \leq x_i \qquad \forall\,(i,j) \tag{1}$$
$$w_{ij} \leq x_j \qquad \forall\,(i,j) \tag{2}$$
$$w_{ij} \geq x_i + x_j - 1 \qquad \forall\,(i,j) \tag{3}$$
$$w_{ij} \geq 0 \qquad \forall\,(i,j) \tag{4}$$

Note that (1), (2) and (4) will force w_{ij} to be zero when x_i or x_j is zero, while (3) will force W_{ij} to be one when both x_i and x_j are one. It is worth noting that if $C_{ij} \geq 0\ \forall (i,j)$ then constraints (3) and (4) are not necessary and we will test whether or not it is better to remove them from the LP. The formulation is identical for all of our problem types except the QSAP, the formulation for which is as follows:

$$\text{STD (QSAP): Max } \sum_{i=1}^{m}\sum_{k=1}^{n} e_{ik}x_{ik} + \sum_{i=1}^{m-1}\sum_{j=i+1}^{m}\sum_{k=1}^{n}\sum_{l=1}^{n} C_{ikjl}w_{ikjl}$$

Subject to

$$w_{ikjl} \leq x_{ik} \qquad \forall\,(i,k,j,l) \tag{5}$$
$$w_{ikjl} \leq x_{jl} \qquad \forall\,(i,k,j,l) \tag{6}$$
$$w_{ikjl} \geq x_{ik} + x_{jl} - 1 \qquad \forall\,(i,k,j,l) \tag{7}$$
$$w_{ikjl} \geq 0 \qquad \forall\,(i,k,j,l) \tag{8}$$

2.2 Glovers Linearization

A year later, in 1975, Glover proposed a second, more concise reformulation technique known as Glover's Linearization [9]. Consider our original QP objective function, and note that the double summation $\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} C_{ij}x_ix_j$ can be rewritten as $\sum_{i=1}^{n}(x_i\sum_{j=1,j\neq i}^{n} C_{ij}x_j)$ by simply factoring the x_i out of the inner summation. Glovers technique is then to replace each term $x_i\sum_{j=1,j\neq i}^{n} C_{ij}x_j$ with a continuous variable z_i to obtain the new objective function:

$$\text{G1: Max } \sum_{i=1}^{n} c_ix_i + \sum_{i=1}^{n} z_i$$

To ensure the equality $z_i = x_i\sum_{j=1,j\neq i}^{n} C_{ij}x_j$ holds, the following auxillary constraints are added:

$$z_i \leq U_i^1 x_i \qquad \forall\, i \tag{9}$$
$$z_i \geq L_i^1 x_i \qquad \forall\, i \tag{10}$$
$$z_i \leq \sum_{j=1}^{n} C_{ij}x_j - L_i^0(1-x_i) \qquad \forall\, i \tag{11}$$
$$z_i \geq \sum_{j=1}^{n} C_{ij}x_j - U_i^0(1-x_i) \qquad \forall\, i \tag{12}$$

where U_i and L_i are upper and lower bounds respectively for $\sum_{j=1}^{n} C_{ij}x_j$. Note that when $x_i = 1$, constraints (9) and (10) are redundant while constraints (11) and (12) will force $z_i = \sum_{j=1}^{n} C_{ij}x_j$. If $x_i = 0$, then constraints (11) and (12) are redundant while constraints (9) and (10) will force $z_i = 0$.

Since we are maximizing, constraints 10 and 12 will not be tight at optimality, and thus can be left out of the LP moving forward (Adams & Forrester, 2005). The G1 formulation for the QSAP is given by:

$$\text{G1 (QSAP): Max } \sum_{i=1}^{m}\sum_{k=1}^{n} e_{ik}x_{ik} + \sum_{i=1}^{m-1}\sum_{k=1}^{n} C_{ikjl}z_{ik}$$

Subject to

$$z_{ik} \leq U_{ik}^{1}x_{ik} \qquad \forall\, i,k \tag{13}$$

$$z_{ik} \leq \sum_{j=i+1}^{m}\sum_{l=1}^{n} C_{ikjl}x_{jl} - L_{ik}^{0}(1-x_{ik}) \qquad \forall\, i,k \tag{14}$$

2.2.1 Glover Bounds

There are a number of ways to compute U and L, with varying degrees of computational difficulty and tightness of the bound. Tighter bounds likely would lead to faster solve times but since finding them requires more work there is a tradeoff here to be examined.

Simple Bounds The simplest way to compute U and L would be to simply include all positive terms for U, and all negative terms for L. That is, we would compute $U_i = \sum_{j=1,C_{ij}>0}^{n} C_{ij}$ and $L_i = \sum_{j=1,C_{ij}<0}^{n} C_{ij}$. This method would require little computational work, but the bounds would likely not be very tight since we are ignoring any knapsack constraints or k-item constraints entirely.

Tight Bounds We can account for these constraints and improve our bounds by setting up and solving a simple linear optimization problem. This can be done by letting $U_i = \text{Max } \{\sum_{j=1}^{n} C_{ij}x_j : x \in \boldsymbol{X}\}$ and $L_i = \text{Min } \{\sum_{j=1}^{n} C_{ij}x_j : x \in \boldsymbol{X}\}$, where x_i is a continuous variable constrained to be between 0 and 1.

Tighter Bounds The bounds can be made even tighter by using the same formulation as above but instead of letting x_i be continuous, make them binary. This will result in the tightest possible bounds, but will likely require more time as solving integer programs is slower than solving a continuous program.

2.2.2 Constraint Substitution

We can further reduce the number of constraints in LP2 by defining and substituting a slack variable s_i such that one of our constraints becomes a simple non negativity constraint, which is already handled implicitly by the Simplex algorithm. For example, if we define $s_i = U_i x_i - z_i$ (representing the slack for constraint 9), and then substitute we get the simplified LP:

$$\text{G2: Max } \sum_{i=1}^{n} c_i x_i + \sum_{i=1}^{n} U_i x_i - s_i$$

Subject to

$$s_i \geq 0 \qquad \forall\, i \tag{15}$$

$$s_i \geq U_i x_i - \sum_{j=1}^{n} C_{ij} x_j + L_i(1 - x_i) \qquad \forall\, i \tag{16}$$

Similarly, we can define s_i to be the slack variable for constraint 11 from LP2 so that $s_i = \sum_{j=1}^{n} C_{ij} x_j - L_i(1 - x_i) - z_i$. Then after substitution we have:

$$\text{G3: Max } \sum_{i=1}^{n} c_i x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} x_j - L_i(1 - x_i) - s_i$$

Subject to

$$s_i \geq 0 \qquad \forall\, i \tag{17}$$

$$s_i \geq -U_i x_i + \sum_{j=1}^{n} C_{ij} x_j - L_i(1 - x_i) \qquad \forall\, i \tag{18}$$

G2 and G3 can also be applied for the QSAP to get the resulting formulations:

$$\text{G2 (QSAP): Max } \sum_{i=1}^{m} \sum_{k=1}^{n} e_{ik} x_{ik} + \sum_{i=1}^{m-1} \sum_{k=1}^{n} U_{ik}^{1} x_{ik} - s_{ik}$$

Subject to

$$s_{ik} \geq 0 \qquad \forall\, i, k \tag{19}$$

$$s_{ik} \geq U_{ik}^{1} x_{ik} + L_{ik}^{0}(1 - x_{ik}) - \sum_{j=1}^{n} \sum_{l=1}^{n} C_{ikjl} x_{jl} \qquad \forall\, i, k \tag{20}$$

$$\text{G3 (QSAP): Max } \sum_{i=1}^{m}\sum_{k=1}^{n} e_{ik}x_{ik} + \sum_{i=1}^{m-1}\sum_{k=1}^{n} -s_{ik} - L_{ik}^{0}(1-x_{ik}) + \sum_{l=1}^{n}\sum_{j=i+1}^{m} C_{ikjl}x_{jl}$$

Subject to

$$s_{ik} \geq 0 \qquad \forall\, i \qquad (21)$$

$$s_{ik} \geq -L_{ik}^{0}(1-x_{ik}) - x_{ik}U_{ik}^{1} + \sum_{l=1}^{n}\sum_{j=i+1}^{m} C_{ikjl}x_{jl} \qquad \forall\, i \qquad (22)$$

2.3 Sherali-Smith Linear Formulation

Another linearization technique introduced in [10] is as follows:

$$\text{SS: Max } \sum_{i=1}^{n} s_i + \sum_{i=1}^{n}(c_i + L_i)x_i$$

Subject to

$$y_i = \sum_{j=1}^{n} C_{ij}x_j - s_i - L_i \qquad \forall\, i \qquad (23)$$

$$y_i \leq (U_i - L_i)(1 - x_i) \qquad \forall\, i \qquad (24)$$

$$s_i \leq (U_i - L_i)x_i \qquad \forall\, i \qquad (25)$$

$$y_i, s_i \geq 0 \qquad \forall\, i \qquad (26)$$

$$x \in X \qquad (27)$$

The SS formulation adds $2n$ variables and $3n$ constraints. Here is how the formulation looks for the QSAP:

$$\text{SS (QSAP): Max } \sum_{i=1}^{m}\sum_{k=1}^{n} e_{ik}x_{ik} + \sum_{i=1}^{m-1}\sum_{k=1}^{n} s_{ik} + x_{ik}L_{ik}$$

Subject to

$$y_{ik} = \sum_{l=1}^{n}\sum_{j=i+1}^{m} C_{ikjl}x_{jl} - s_{ik} - L_{ik} \qquad \forall\, i,k \qquad (28)$$

$$y_{ik} \leq (U_{ik} - L_{ik})(1 - x_{ik}) \qquad \forall\, i,k \qquad (29)$$

$$s_{ik} \leq (U_{ik} - L_{ik})x_{ik} \qquad \forall\, i,k \qquad (30)$$

$$y_{ik}, s_{ik} \geq 0 \qquad \forall\, i,k \qquad (31)$$

$$x \in X \qquad (32)$$

2.4 Extended Linear Formulation

Another linearization technique introduced in [11] is as follows:

$$\text{ELF: Max } \sum_{i=1}^{n} c_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} C_{ij} - C_{ij}(z_{ij}^i + z_{ij}^j)$$

Subject to

$$z_{ij}^i + z_{ij}^j \leq 1 \qquad \forall\, i, j, (i < j) \tag{33}$$
$$x_i + z_{ij}^i \leq 1 \qquad \forall\, i, j, (i < j) \tag{34}$$
$$x_j + z_{ij}^j \leq 1 \qquad \forall\, i, j, (i < j) \tag{35}$$
$$x_i + z_{ij}^i + z_{ij}^j \geq 1 \qquad \forall\, i, j, (i < j) \tag{36}$$
$$x_j + z_{ij}^i + z_{ij}^j \geq 1 \qquad \forall\, i, j, (i < j) \tag{37}$$
$$x \in X \tag{38}$$

The ELF formulation adds $2n(n-1)$ variables and $5n(n-1)/2$ constraints. Here is how the formulation looks for the QSAP:

$$\text{ELF (QSAP): Min } \sum_{i=1}^{m} \sum_{k=1}^{n} e_{ik} x_{ik} + \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \sum_{k=1}^{n} \sum_{l=1}^{n} C_{ikjl} - C_{ikjl}(z_{ikjl}^{ik} + z_{ikjl}^{jl})$$

Subject to

$$z_{ikjl}^{ik} + z_{ikjl}^{jl} \leq 1 \qquad \forall\, i, j, k, l, (i < j) \tag{39}$$
$$x_{ik} + z_{ikjl}^{ik} \leq 1 \qquad \forall\, i, j, k, l, (i < j) \tag{40}$$
$$x_{jl} + z_{ikjl}^{jl} \leq 1 \qquad \forall\, i, j, k, l, (i < j) \tag{41}$$
$$x_{ik} + z_{ikjl}^{ik} + z_{ikjl}^{jl} \geq 1 \qquad \forall\, i, j, k, l, (i < j) \tag{42}$$
$$x_{jl} + z_{ikjl}^{ik} + z_{ikjl}^{jl} \geq 1 \qquad \forall\, i, j, k, l, (i < j) \tag{43}$$
$$x \in X \tag{44}$$

References

[1] Giorgio Gallo, Peter L Hammer, and Bruno Simeone. Quadratic knapsack problems. In *Combinatorial optimization*, pages 132–149. Springer, 1980.

[2] Lucas Létocart, Marie-Christine Plateau, and Gérard Plateau. An efficient hybrid heuristic method for the 0-1 exact k-item quadratic knapsack problem. *Pesquisa Operacional*, 34(1):49–72, 2014.

[3] Lucas Létocart and Angelika Wiegele. Exact solution methods for the k-item quadratic knapsack problem. In *International Symposium on Combinatorial Optimization*, pages 166–176. Springer, 2016.

[4] Ray Porn, Otto Nissfolk, Anders Skjal, and Tapio Westerlund. Solving 0–1 quadratic programs by reformulation techniques. *Industrial & Engineering Chemistry Research*, 56(45):13444–13453, 2017.

[5] Serigne Gueye and Philippe Michelon. A linearization framework for unconstrained quadratic (0-1) problems. *Discrete Applied Mathematics*, 157(6):1255–1266, 2009.

[6] Alain Billionnet. Different formulations for solving the heaviest k-subgraph problem. *INFOR: Information Systems and Operational Research*, 43(3):171–186, 2005.

[7] Alain Billionnet, Sourour Elloumi, and Marie-Christine Plateau. Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation: The qcr method. *Discrete Applied Mathematics*, 157(6):1185–1197, 2009.

[8] Fred Glover and Eugene Woolsey. Converting the 0-1 polynomial programming problem to a 0-1 linear program. *Operations research*, 22(1):180–182, 1974.

[9] Fred Glover. Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, 22(4):455–460, 1975.

[10] Hanif D Sherali and J Cole Smith. An improved linearization strategy for zero-one quadratic programming problems. *Optimization Letters*, 1(1):33–47, 2007.

[11] Fabio Furini and Emiliano Traversi. Extended linear formulation for binary quadratic problems. *Optim Online Google Scholar*, 2013.