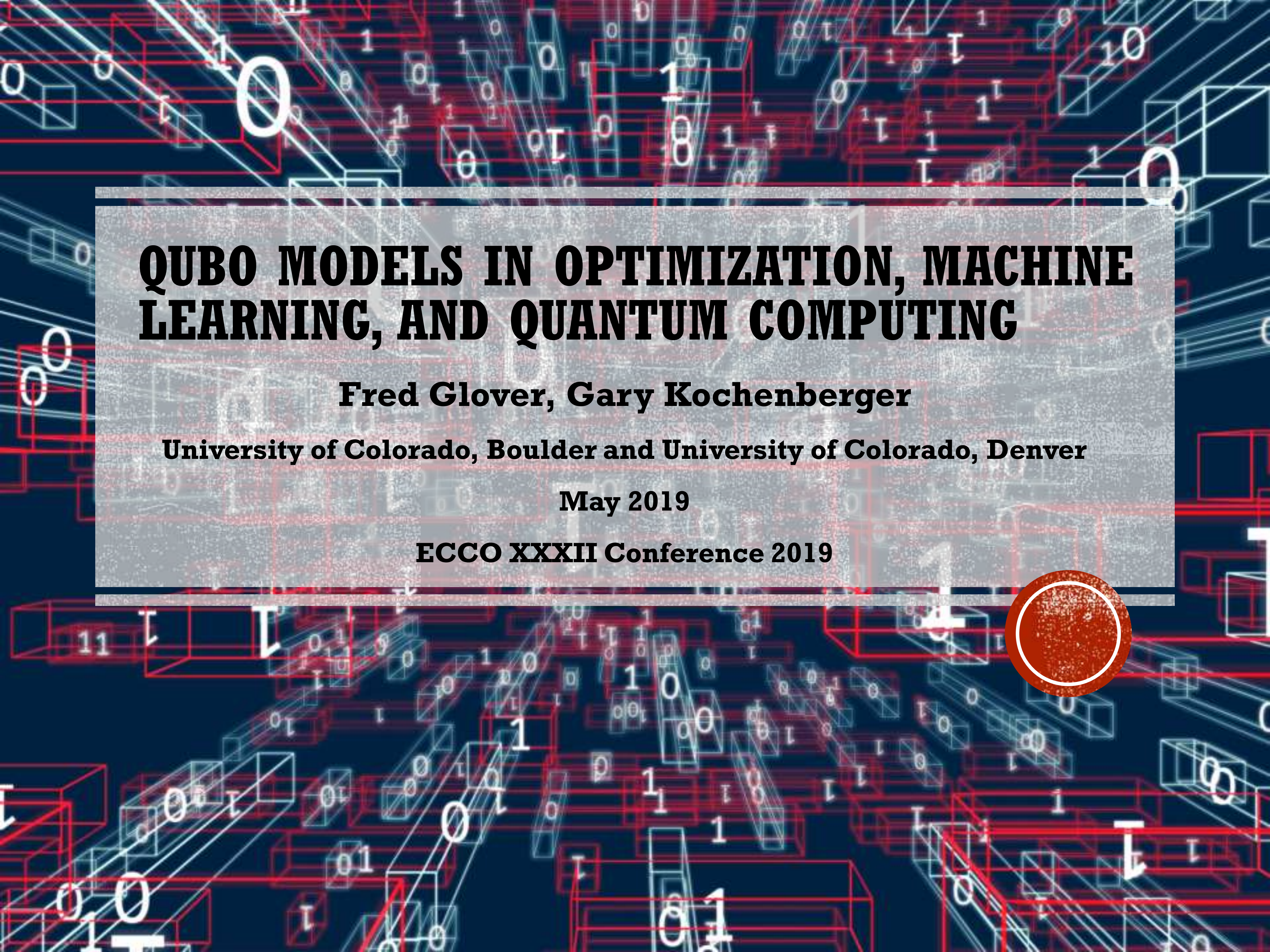


QUBO MODELS IN OPTIMIZATION, MACHINE LEARNING, AND QUANTUM COMPUTING

Fred Glover, Gary Kochenberger

University of Colorado, Boulder and University of Colorado, Denver

May 2019

ECCO XXXII Conference 2019

CONTENTS

University of Colorado
Boulder | Colorado Springs | Denver | Anschutz Medical Campus

- QUBO model – Introduction
  - QUBO Definition
  - QUBO model encompasses
  - Practical Relevance of QUBO
  - Problem Solving with QUBO
- Creating QUBO Models
  - Natural QUBO Formulations
  - Using Known Penalties
  - Using a General Purpose Approach
- Connections with Quantum Computing and Machine Learning
  - Quantum Computing QUBO Developments
  - Meta-analytics and Alpha-QUBO
  - Unsupervised Machine Learning
  - Supervised Machine Learning
  - Machine learning to improve QUBO solution processes
- Conclusions

QUBO Model - Introduction

QUBO DEFINITION:

- The Unconstrained Quadratic Binary Optimization problem (QUBO) is:

$$QUBO: opt\ \ x^t Q x$$

  - where
  - X is an n-vector of binary variables
  - Q is an n-by-n symmetric matrix of constants

THE QUADRATIC UNCONSTRAINED BINARY OPTIMIZATION (QUBO) MODEL – KEY FEATURES

- QUBO unifies a rich variety of combinatorial optimization problems.
- QUBO has significant applications in Machine Learning
- QUBO is important in the quantum computing area:
  - **D-Wave Systems quantum annealing computers**
  - **IBM neuromorphic computers**
  - **QAOA computers**
  - **Fujitsu Digital Annealer**

MOTIVATION:

- The QUBO model has become a unifying framework for combinatorial optimization.
- Many important optimization problems can be **re-cast** as a QUBO model and then solved with appropriate software.

- Options:
  - Many models, many solution techniques
  - One model (QUBO), one solution technique

QUADRATIC UNCONSTRAINED BINARY OPTIMIZATION (QUBO) MODEL

Organizations and research groups actively engaged in applications

- Google
- Amazon
- IBM
- Lockheed Martin
- Los Alamos National Laboratory
- Oak Ridge National Laboratory
- Lawrence Livermore National Laboratory
- NASA Ames Research Center
- Fujitsu
- D-Wave
- Many others …

THE QUBO MODEL ENCOMPASSES

- Quadratic Assignment Problems
- Capital Budgeting Problems
- Multiple Knapsack Problems
- Task Allocation Problems (distributed computer systems)
- Maximum Diversity Problems
- P-Median Problems
- Asymmetric and Symmetric Assignment Problems
- Spin Glass Problems

THE QUBO MODEL ENCOMPASSES (CONTINUED)

- General Linear 0/1 Problems
- Quadratic Knapsack Problems
- Constraint Satisfaction Problems (CSPs)
- Portfolio Analysis Problems
- Set Partitioning Problems
- Set Packing Problems
- Warehouse Location Problems
- Maximum Clique Problems

THE QUBO MODEL ENCOMPASSES (STILL CONTINUED)

- Maximum Independent Set Problems
- Maximum Cut Problems
- Graph Coloring Problems
- Number Partitioning Problems
- Linear Ordering Problems
- Clique Partitioning Problems
- SAT and Max Sat Problems
- Clustering Problems
    - Modularity Maximization
    - Correlation Clustering
    - Other

ADDED PRACTICAL RELEVANCE OF QUBO

- QUBO problems are NP-hard.
- Exact solvers designed to find “optimal” solutions (CPLEX and Gurobi solvers) often can solve only very small problem instances.
- Realistic sized problems can run for days and even weeks with CPLEX and Gurobi –and still fail to provide high quality solutions.
- By contrast, modern metaheuristic methods – based on Tabu search and path relinking – can find high quality solutions in only seconds to minutes.

Creating QUBO Models

The tutorial provided in the following sections that will illustrate the process of reformulating important optimization problems as QUBO models through a series of explicit examples.

https://arxiv.org/abs/1811.11538

BASIC QUBO PROBLEM FORMULATION

Minimize/Maximize xQx: x binary

For a symmetric matrix $Q = (q_{ij}: i, j \in N = \{1, \ldots, n\})$

where:

$$xQx = \sum(q_{ij}x_ix_j: i, j \in N)$$

In linear + quadratic form

$$= \sum(q_{ii}x_i: i \in N) + \sum(q_{ij}x_ix_j: i, j \in N: i \neq j)$$

Since binary variables satisfy $x_i = {x_i}^2$

EXAMPLE PROBLEM IN BINARY X VARIABLES:

Minimize: $y = -5x_1 - 3x_2 - 8x_3 - 6x_4 + 4x_1x_2 + 8x_1x_3 + 2x_2x_3 + 10x_3x_4$

Linear part: $-5x_1 - 3x_2 - 8x_3 - 6x_4$

Quadratic part: $4x_1x_2 + 8x_1x_3 + 2x_2x_3 + 10x_3x_4$

$$\text{Matrix Form: Minimize } y = (x_1\ x_2\ x_3\ x_4)\begin{bmatrix} -5 & 2 & 4 & 0 \\ 2 & -3 & 1 & 0 \\ 4 & 1 & -8 & 5 \\ 0 & 0 & 5 & -6 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = x^T Q x$$

Optimal Solution: $y = -11,\ x_1 = x_4 = 1, x_2 = x_3 = 0.$

CREATING QUBO MODELS

In the next few slides, I'll highlight some of the computational experience we've produced on casting into QUBO formulation

- Natural Formulation
- Known penalties
- Constructing penalties via Transformation # 1
- Employing a change of variable
- Using Special Penalties (e.g. Linear Ordering Problem)

EXAMPLE: THE NUMBER PARTITIONING PROBLEM

Partition a set of numbers into two subsets such that the subset sums are as close to each other as possible. We model this problem as a QUBO instance as follows:

Consider a set of number S= $\{s_1\ s_2, s_3, ..s_m\}$.

Let $x_j\ = 1$ if s_j is assigned to subset 1; 0 otherwise. Then the sum for subset 1 is given by, $sum_1\ = \sum_{j=1}^{m}(s_j\ x_j)$ and the sum for subset 2 is given by $sum_2 = \ \sum_{j=1}^{m}(s_j\ )$ $-\ \sum_{j=1}^{m}(s_j\ x_j)$. The difference in the sums is then

$$\text{diff} = \sum_{j=1}^{m}(s_j\ )\ - 2\sum_{j=1}^{m}(s_j\ x_j) = \text{c} - 2\ \sum_{j=1}^{m}(s_j\ x_j)$$

EXAMPLE: THE NUMBER PARTITIONING PROBLEM

We approach the goal of minimizing the difference by minimizing

$$diff^2 = \left\{ c - 2\sum_{j=1}^{m} s_j x_j \right\}^2 = c^2 + 4x^t Qx$$

Dropping the additive and multiplicative constants, our QUBO optimization problem becomes:

$$\text{QUBO: } \min y = x^T \text{ Qx}$$

Numerical Example: Consider the set of eight numbers

S = { 25, 7, 13, 31, 42, 17, 21, 10}

- By the development above, we have c^2 = 27,556 and the equivalent QUBO problem is min y = x^T Qx with

$$Q = \begin{bmatrix} -3525 & 175 & 325 & 775 & 1050 & 425 & 525 & 250 \\ 175 & -1113 & 91 & 217 & 294 & 119 & 147 & 70 \\ 325 & 91 & -1989 & 403 & 546 & 221 & 273 & 130 \\ 775 & 217 & 403 & -4185 & 1302 & 527 & 651 & 310 \\ 1050 & 294 & 546 & 1302 & -5208 & 714 & 882 & 420 \\ 425 & 119 & 221 & 527 & 714 & -2533 & 357 & 170 \\ 525 & 147 & 273 & 651 & 882 & 357 & -3045 & 210 \\ 250 & 70 & 130 & 310 & 420 & 170 & 210 & -1560 \end{bmatrix}$$

- Solving QUBO gives x = (0,0,0,1,1,0,0,1), yielding perfectly matched sums which equal 83.
- The development employed here can be expanded to address other forms of the number partitioning problems as discussed in Alidaee, et.al. (2005)

EXAMPLE: THE MAX CUT PROBLEM

Given an undirected graph G(V, E), the Max Cut problem seeks to partition V into two sets such that the number of edges between the two sets (the cut), is as large as possible.

We can model this problem by introducing binary variables $x_j = 1$ if vertex j is in one set and $x_j = 0$ if it is in the other set. Viewing a cut as severing edges joining two sets, the quantity $x_i + x_j - 2x_ix_j$ identifies whether the edge (i, j) is in the cut.

Thus, the problem of maximizing the number of edges in the cut can be formulated as

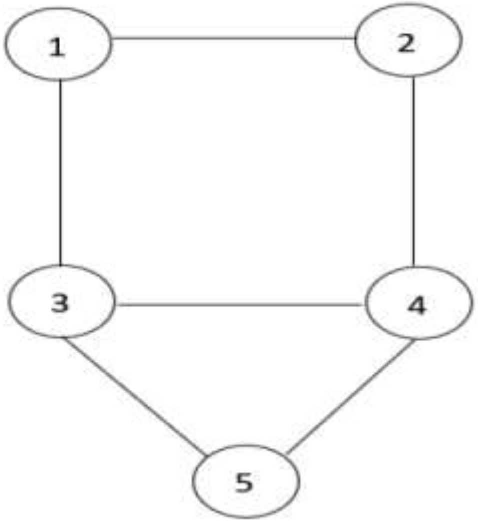
$$\text{Maximize} \qquad y = \sum_{(i,j)\in E} \left(x_i + x_j - 2x_ix_j\right)$$

Which is an instance of

$$QUBO : \max\, y = x^tQx$$

THE MAX CUT PROBLEM

To illustrate the Max Cut problem, consider the undirected graph with 5 vertices and 6 edges.



Explicitly taking into account all edges in the graph gives the following formulation:

$$\begin{aligned} \textit{Maximize} \quad y = (x_1 + x_2 - 2x_1x_2) + (x_1 + x_3 - 2x_1x_3) + (x_2 + x_4 - 2x_2x_4) \\ + (x_3 + x_4 - 2x_3x_4) + (x_3 + x_5 - 2x_3x_5) + (x_4 + x_5 - 2x_4x_5) \end{aligned}$$

THE MAX CUT PROBLEM

This takes the desired form QUBO = max x^tQx by writing the symmetric Q matrix as:

$$Q=\begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 2 & 0 & -1 & 0 \\ -1 & 0 & 3 & -1 & -1 \\ 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

Solving this QUBO model gives x =(0, 1, 1, ,0, 0). Hence vertices 2 and 3 are in one set and vertices 1, 4, and 5 are in the other, with a maximum cut value of 5

CREATING QUBO USING KNOWN PENALTIES

- A penalty function is said to be a ***valid infeasible penalty (VIP)*** if it is zero for feasible solutions and otherwise positive.

- Including quadratic VIPs in the objective function for each constraint in the original model yields a transformed model in the form of QUBO. VIPs for several commonly encountered constraints are given below

QUBO MODELS FOR CONSTRAINED PROBLEMS

Most problems of interest include additional constraints. Many of these models can be re-formulated as a QUBO model by introducing quadratic penalties with a positive scalar P:

Classical Constraint	Equivalent Penalty
x + y <=1	P(xy)
x + y >=1	P(1-x-y+xy)
x + y =1	P(1-x-y+2xy)
x<=y	P(x-xy)
$x_1 + x_2 + x_3 \le 1$	$P(x_1 x_2 + x_1 x_3 + x_2 x_3)$
x = y	P(x+y-2xy)

Simple examples: Known constraint/penalty pairs

RE-CASTING INTO THE UNIFIED FRAMEWORK:

- For certain types of constraints, equivalent quadratic penalty representations are known in advance
- For instance, let x_i and x_j be binary variables and consider the constraint

$$x_i + x_j \leq 1 \qquad (1)$$

- A quadratic infeasibility penalty that imposes the same condition on x_i and x_j is:

$$P x_i x_j \qquad (2)$$

where P is a large positive scalar.

RE-CASTING INTO THE UNIFIED FRAMEWORK:

- This penalty function is positive when both variables are set to one (i.e., when (1) is violated), and otherwise the function is equal to zero.

- For a minimization problem then, adding the penalty function to the objective function is an alternative equivalent to imposing the constraint of (1) in the traditional manner.

- Due to their importance and frequency of use, we refer to this special case as ***Transformation #2.***

EXAMPLE: THE MINIMUM VERTEX COVER PROBLEM

- A vertex cover is a subset of the vertices (nodes) such that each edge in the graph is incident to at least one vertex in the subset. The Minimum Vertex Cover problem seeks to find a cover with a minimum number of vertices in the subset.
- MVC can be formulated as follows. Let $x_j = 1$ if vertex j is in the cover (i.e., in the subset) and $x_j = 0$ otherwise. Then this standard constrained, linear 0/1 optimization model is:

EXAMPLE: THE MINIMUM VERTEX COVER PROBLEM

$$\text{Minimize} \sum_{j \in V} x_j$$

$$x_i + x_j \geq 1 \quad \text{for all } (i, j) \in E$$

Note the constraints ensure that at least one of the endpoints of each edge will be in the cover and the objective function seeks to find the cover using the least number of vertices .

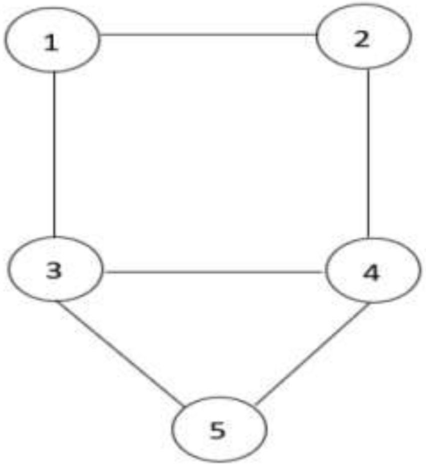
EXAMPLE: THE MINIMUM VERTEX COVER PROBLEM

The constraints in the standard MVC model can be represented by a penalty of the form P(1-x-y+xy). Thus, an unconstrained alternative to the constrained model for MVC is

$$\text{Minimize } y = \sum_{j \in V} x_j + P\Big(\sum_{(i,j) \in E} \big(1 - x_i - x_j + x_i x_j\big)\Big)$$

where P again represents a positive scalar penalty. In turn, we can write this as minimize x^tQx plus a constant term.

Numerical Example: Consider the graph to determine a minimum vertex cover. For this graph with n = 6 edges and m = 5 nodes, the model becomes:



$$\begin{aligned} Minimize \quad y = {} & x_1 + x_2 + x_3 + x_4 + x_5 + \\ & P(1 - x_1 - x_2 + x_1 x_2) + \\ & P(1 - x_1 - x_3 + x_1 x_3) + \\ & P(1 - x_2 - x_4 + x_2 x_4) + \\ & P(1 - x_3 - x_4 + x_3 x_4) + \\ & P(1 - x_3 - x_5 + x_3 x_5) + \\ & P(1 - x_4 - x_5 + x_4 x_5) \end{aligned}$$

which can be written as

$$\begin{aligned} Minimize \quad y = {} & (1-2P)x_1 + (1-2P)x_2 + (1-3P)x_3 + (1-3P)x_4 + (1-2P)x_5 \\ & + Px_1x_2 + Px_1x_3 + Px_2x_4 + Px_3x_4 + Px_3x_5 + Px_4x_5 + 6P \end{aligned}$$

Numerical Example:

Arbitrarily choosing P to be equal to 8 and dropping the additive constant (6P = 48) gives our QUBO model with the Q matrix given by

$$\begin{bmatrix} -15 & 4 & 4 & 0 & 0 \\ 4 & -15 & 0 & 4 & 0 \\ 4 & 0 & -23 & 4 & 4 \\ 0 & 4 & 4 & -23 & 4 \\ 0 & 0 & 4 & 4 & -15 \end{bmatrix}$$

Solving this QUBO model gives: at for which $x^TQx = -45$
$x = (0,1,1,0,1)$, $y = 3$.

EXAMPLE: THE SET PACKING PROBLEM

- In general, this class of problems is given by

$$\max \sum_{j=1}^{n} w_j x_j$$

$$st$$

$$\sum_{j=1}^{n} a_{ij} x_j \leq 1 \quad for\ i = 1, \ldots m$$

Maximize the weighted total number of subsets such that the selected sets have to be pairwise disjoint.

- **Numerical Example:**

$$\text{Max } x_1 + x_2 + x_3 + x_4$$

$$\text{s.t. } x_1 \quad + x_3 + x_4 \leq 1$$

$$\text{s.t. } x_1 + x_2 \qquad \leq 1$$

- Re-casting as QUBO via the penalties of previous Table.

$$\text{Max } y = x_1 + x_2 + x_3 + x_4 - Px_1x_3 - Px_1x_4 - Px_3x_4 - Px_1x_2$$

- The equivalent QUBO model depends only on the number of original variables, being independent of the number of constraints in the original problem.

- This has our customary QUBO form

$$QUBO : \max x^t Q x$$

- where the Q matrix , with P arbitrarily chosen to be 6, is given by

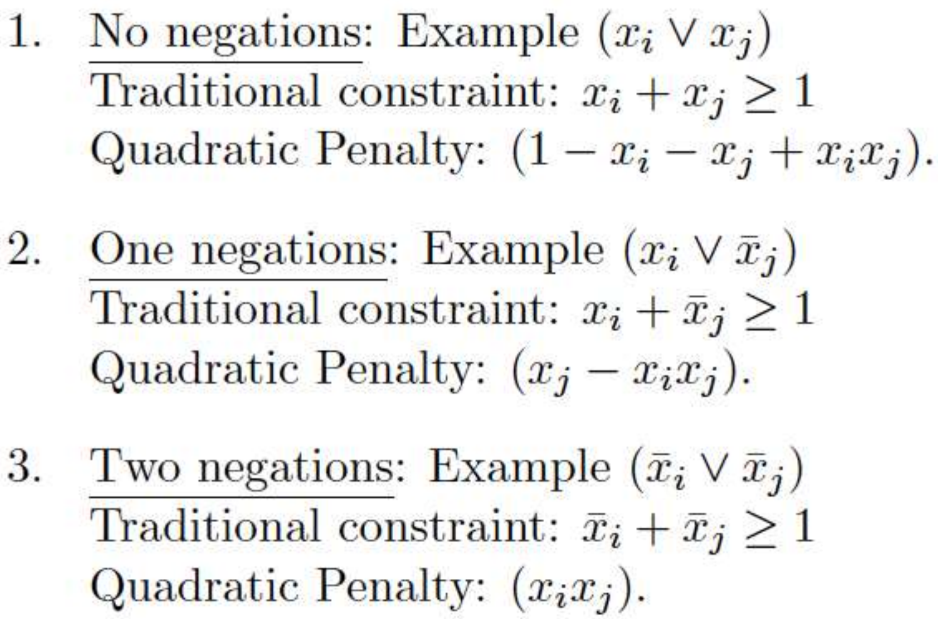
$$\begin{bmatrix} 1 & -3 & -3 & -3 \\ -3 & 1 & 0 & 0 \\ -3 & 0 & 1 & -3 \\ -3 & 0 & -3 & 1 \end{bmatrix}$$

- Solving the QUBO model gives y = 2, and x = (0,1,1,0). Note that at this solution, all four penalty terms are equal to zero.

- Set packing problems with thousands of variables and constraints have been efficiently reformulated and solved in Alidaee, et. al. (2008).

EXAMPLE: THE MAX 2-SAT PROBLEM

- For Max 2-Sat, each clause consists of two literals and a clause is satisfied if either or both literals are true.



1. No negations: Example $(x_i \vee x_j)$
   Traditional constraint: $x_i + x_j \geq 1$
   Quadratic Penalty: $(1 - x_i - x_j + x_ix_j)$.

2. One negations: Example $(x_i \vee \bar{x}_j)$
   Traditional constraint: $x_i + \bar{x}_j \geq 1$
   Quadratic Penalty: $(x_j - x_ix_j)$.

3. Two negations: Example $(\bar{x}_i \vee \bar{x}_j)$
   Traditional constraint: $\bar{x}_i + \bar{x}_j \geq 1$
   Quadratic Penalty: (x_ix_j).

- The QUBO approach illustrated above has been successfully used in Kochenberger, et. al. (2005) to solve Max 2-sat problems with hundreds of variables and thousands of clauses.

CREATING QUBO MODELS: A GENERAL APPROACH

- For general constraints, however, VIPs are not known in advance and need to be "discovered."
- Consider the general constrained problem:
- $$\min x_0 = xQx \qquad (3)$$

st

$$Ax = b, \quad x\ binary$$

For a positive scalar P:

$$\begin{aligned} x_0 &= x^tQx + P\left(Ax - b\right)^t\left(Ax - b\right) \qquad (4) \\ &= x^tQx + x^tDx + c \\ &= x^t\hat{Q}x + c \end{aligned}$$

CREATING QUBO MODELS: A GENERAL APPROACH

Dropping the additive constant, the equivalent unconstrained version of the constrained problem becomes

$$QUBO : \min x\hat{Q}x,\ x\ binary \qquad (5)$$

Transformation #1: The preceding steps that transform (3) and (4) into (5)

EXAMPLE: THE SET PARTITIONING PROBLEM

- The set partitioning problem can be formulated as

$$\min \sum_{j=1}^{n} c_j x_j$$

$$st$$

$$\sum_{j=1}^{n} a_{ij} x_j = 1 \quad for\ i = 1, \ldots m$$

Partitioning a set of items into subsets so that each item appears in exactly one subset and the cost of the subsets chosen is minimized.

- Applying Transformation 1 the set partitioning problem becomes a QUBO problem without introducing new variables.
- The QUBO approach to solving set partitioning problems has been successfully applied in Lewis, et. al. (2008) to solve large instances with thousands of variables and hundreds of constraints.

EXAMPLE: THE GRAPH COLORING PROBLEM

Vertex coloring problems seek to assign colors to nodes of a graph in such a way that adjacent nodes receive different colors. These problems can be modeled as satisfiability problems as follows:

Let x_{ij} = 1 if node I is assigned color j, and 0 otherwise.

Since each node must be colored. we have the constraints

$$\sum_{j=1}^{K} x_{ij} = 1 \quad i = 1,\dots,n$$

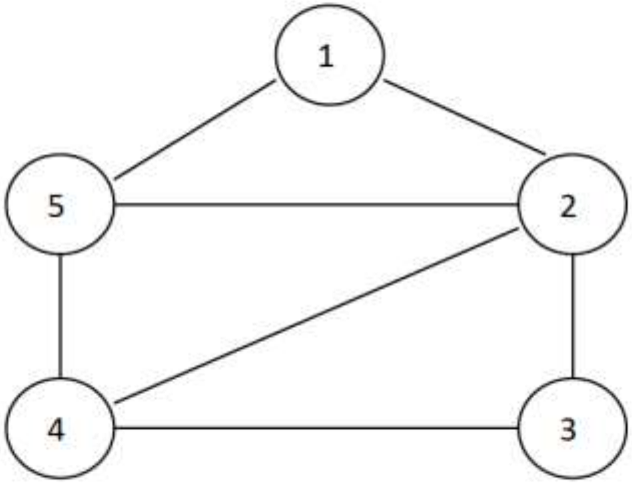
for all adjacent nodes (i, j) in the graph.

A feasible coloring, in which adjacent nodes are assigned different colors, is assured by imposing the constraints

$$x_{ip} + x_{jp} \leq 1 \quad p = 1,\dots,K$$

This problem, then, can be re-cast in the form of a QUBO model by using Transformation # 1 on the node assignment constraints and using Transformation # 2 on the adjacency constraints.

Numerical Example: Consider the problem of finding a feasible coloring of the graph using K= 3 colors. Given the discussion, we see that the goal is to find a solution to the svstem:



$$x_{i1} + x_{i2} + x_{i3} = 1 \quad i = 1,\ldots,5$$

$$x_{ip} + x_{jp} \le 1 \quad p = 1,\ldots,3$$

(for all adjacent nodes i and j)

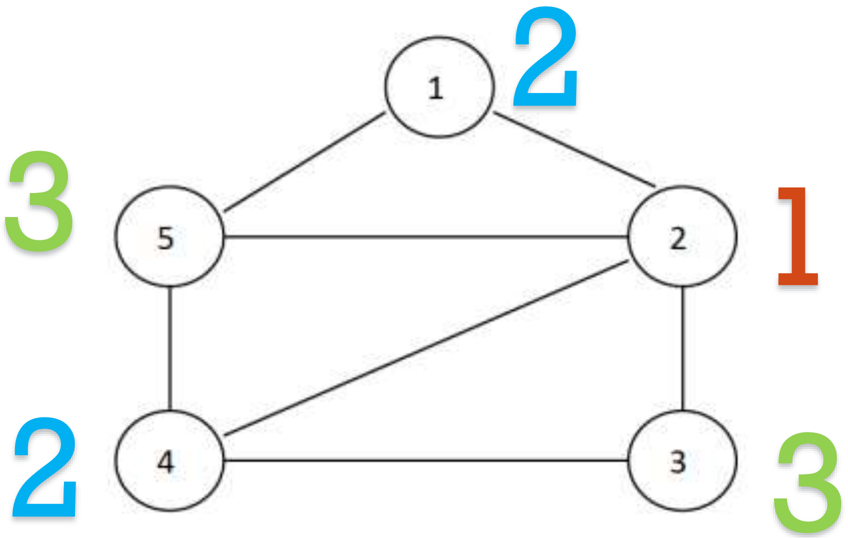
$$QUBO: \min x\hat{Q}x,\ x\ binary$$

$$Q = \begin{array}{ccccccccccccccc}
11 & 12 & 13 & 21 & 22 & 23 & 31 & 32 & 33 & 41 & 42 & 43 & 51 & 52 & 53 \\
-4 & 4 & 4 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\
4 & -4 & 4 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\
4 & 4 & -4 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\
2 & 0 & 0 & -4 & 4 & 4 & 2 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 \\
0 & 2 & 0 & 4 & -4 & 4 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & 2 & 0 \\
0 & 0 & 2 & 4 & 4 & -4 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & 2 \\
0 & 0 & 0 & 2 & 0 & 0 & -4 & 4 & 4 & 2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & 0 & 4 & -4 & 4 & 0 & 2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 2 & 4 & 4 & -4 & 0 & 0 & 2 & 0 & 0 & 0 \\
0 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & -4 & 4 & 4 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 4 & -4 & 4 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 2 & 4 & 4 & -4 & 0 & 0 & 2 \\
2 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & -4 & 4 & 4 \\
0 & 2 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 4 & -4 & 4 \\
0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 2 & 4 & 4 & -4
\end{array}$$

- Solving this model yields the feasible coloring:

$$x_2 = x_4 = x_9 = x_{11} = x_{15} = 1$$

- with all other variables equal to zero.
- Switching back to our original variables, this solution means that nodes 1 and 4 get color #2, node 2 gets color # 1, and nodes 3 and 5 get color # 3.
- This approach to graph coloring problems has proven to be very effective for a wide variety of coloring instances with hundreds of nodes, as demonstrated in Kochenberger, et. al. (2005) and Hao, et al. (2010).



EXAMPLE: THE GENERAL 0/1 LINEAR MODEL

Many important problems in industry and government can be modeled as 0/1 linear programs with a mixture of constraint types. The general problem of this nature can be represented in matrix form by

$$\begin{aligned} &\max cx \\ &st \\ &Ax = b \\ &x \; binary \end{aligned}$$

where slack variables are introduced as needed to convert inequality constraints into equalities. Given a problem in this form, Transformation # 1 can be used to re-cast the problem into the QUBO form

$$\begin{aligned} &\max x_0 = x^t Q x \\ &st \;\; x \; binary \end{aligned}$$

Numerical Example: Consider the general 0/1 problem

$$\begin{aligned} &\max\ 6x_1 + 4x_2 + 8x_3 + 5x_4 + 5x_5 \\ &st \\ &\quad 2x_1 + 2x_2 + 4x_3 + 3x_4 + 2x_5 \leq 7 \\ &\quad 1x_1 + 2x_2 + 2x_3 + 1x_4 + 2x_5 = 4 \\ &\quad 3x_1 + 3x_2 + 2x_3 + 4x_4 + 4x_5 \geq 5 \\ &\quad\ x \in \{0,1\} \end{aligned}$$

Introducing slack variables

$$\begin{aligned} &0 \leq s_1 \leq 3 \quad \Rightarrow s_1 = 1x_6 + 2x_7 \\ &0 \leq s_3 \leq 6 \quad \Rightarrow s_3 = 1x_8 + 2x_9 + 4x_{10} \end{aligned}$$

We can now use Transformation # 1 to reformulate our problem as a QUBO instance.

$$\begin{aligned} \max y = {} & 6x_1 + 4x_2 + 8x_3 + 5x_4 + 5x_5 \\ & - P(2x_1 + 2x_2 + 4x_3 + 3x_4 + 2x_5 + 1x_6 + 2x_7 - 7)^2 \\ & - P(1x_1 + 2x_2 + 2x_3 + 1x_4 + 2x_5 - 4)^2 \\ & - P(3x_1 + 3x_2 + 2x_3 + 4x_4 + 4x_5 - 1x_8 - 2x_9 - 4x_{10} - 5)^2 \end{aligned}$$

Taking P = 10 and re-writing this in the QUBO format with an additive constant of - 900 and a Q matrix gives

$$\begin{bmatrix} 526 & -150 & -160 & -190 & -180 & -20 & -40 & 30 & 60 & 120 \\ -150 & 574 & -180 & -200 & -200 & -20 & -40 & 30 & 60 & 120 \\ -160 & -180 & 688 & -220 & -200 & -40 & -80 & 20 & 40 & 80 \\ -190 & -200 & -220 & 645 & -240 & -30 & -60 & 40 & 80 & 160 \\ -180 & -200 & -200 & -240 & 605 & -20 & -40 & 40 & 80 & 160 \\ -20 & -20 & -40 & -30 & -20 & 130 & -20 & 0 & 0 & 0 \\ -40 & -40 & -80 & -60 & -40 & -20 & 240 & 0 & 0 & 0 \\ 30 & 30 & 20 & 40 & 40 & 0 & 0 & -110 & -20 & -40 \\ 60 & 60 & 40 & 80 & 80 & 0 & 0 & -20 & -240 & -80 \\ 120 & 120 & 80 & 160 & 160 & 0 & 0 & -40 & -80 & -560 \end{bmatrix}$$

Solving gives the non-zero values $x_1 = x_4 = x_5 = x_9 = x_{10} = 1$ for which y =916.

Note that the third constraint is loose. Adjusting for the additive constant, gives an objective function value of 16.

EXAMPLE: THE QUADRATIC ASSIGNMENT PROBLEM

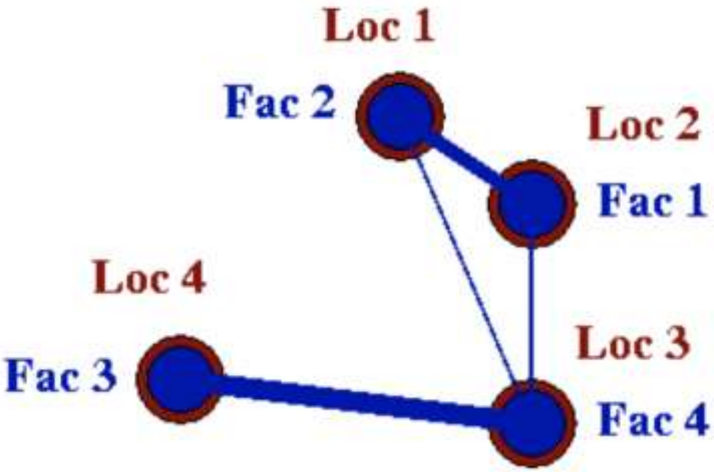
- We are given n facilities and n locations along with a flow matrix denoting the flow of material between facilities i and j. A distance matrix specifies the distance between sites i and j. The optimization problem is to find an assignment of facilities to locations to minimize the weighted flow across the system.
- The classic QAP model can be stated as:

$$\text{Minimize} \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} f_{ij} d_{kl} x_{ik} x_{jl}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad j = 1,\ldots,n$$

$$\sum_{j=1}^{n} x_{ij} = 1 \quad i = 1,\ldots,n$$

$$x_{ij} \in \{0,1\}, \quad i,j = 1,\ldots,n$$



- Transformation # 1 can be used to convert any QAP problem into a QUBO instance.
- A QUBO approach to solving QAP problems, as illustrated above, has been successfully applied to problems with more than 30 facilities and locations in Wang, et. al. (2016).

Numerical Example:

Consider a small example with n = 3 facilities and 3 locations with flow and distance matrices respectively given as follows:

$$\begin{bmatrix} 0 & 5 & 2 \\ 5 & 0 & 3 \\ 2 & 3 & 0 \end{bmatrix} \text{ and } \begin{bmatrix} 0 & 8 & 15 \\ 8 & 0 & 13 \\ 15 & 13 & 0 \end{bmatrix}.$$

Given the flow and distance matrices our QAP model becomes minimizing the following:

$$\begin{aligned} x_0 = {} & 80x_1x_5 + 150x_1x_6 + 32x_1x_8 + 60x_1x_9 + 80x_2x_4 + 130x_2x_6 + 60x_2x_7 + 52x_2x_9 \\ & + 150x_3x_4 + 130x_3x_5 + 60x_3x_7 + 52x_3x_8 + 48x_4x_8 + 90x_4x_9 + 78x_5x_9 + 78x_6x_8 \end{aligned}$$

s.t.

$$\begin{aligned} x_1 + x_2 + x_3 &= 1 \\ x_4 + x_5 + x_6 &= 1 \\ x_7 + x_8 + x_9 &= 1 \\ x_1 + x_4 + x_7 &= 1 \\ x_2 + x_5 + x_8 &= 1 \\ x_3 + x_6 + x_9 &= 1 \end{aligned}$$

Converting the constraints into quadratic penalty terms

$$\begin{aligned}\min y = {} & 80x_1x_5 + 150x_1x_6 + 32x_1x_8 + 60x_1x_9 + 80x_2x_4 + 130x_2x_6 + 60x_2x_7 + 52x_2x_9 \\ & + 150x_3x_4 + 130x_3x_5 + 60x_3x_7 + 52x_3x_8 + 48x_4x_8 + 90x_4x_9 + 78x_5x_9 + 78x_6x_8 \\ & + P(x_1 + x_2 + x_3 - 1)^2 + P(x_4 + x_5 + x_6 - 1)^2 + P(x_7 + x_8 + x_9 - 1)^2 \\ & + P(x_1 + x_4 + x_7 - 1)^2 + P(x_2 + x_5 + x_8 - 1)^2 + P(x_3 + x_6 + x_9 - 1)^2\end{aligned}$$

Choosing a penalty value of P = 200, this becomes the standard QUBO problem with an additive constant of 1200 and the following 9-by-9 Q matrix:

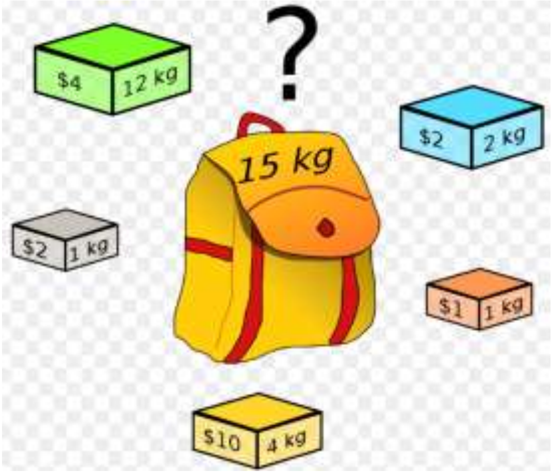
$$\begin{bmatrix} -400 & 200 & 200 & 200 & 40 & 75 & 200 & 16 & 30 \\ 200 & -400 & 200 & 40 & 200 & 65 & 16 & 200 & 26 \\ 200 & 200 & -400 & 75 & 65 & 200 & 30 & 26 & 200 \\ 200 & 40 & 75 & -400 & 200 & 200 & 200 & 24 & 45 \\ 40 & 200 & 65 & 200 & -400 & 200 & 24 & 200 & 39 \\ 75 & 65 & 200 & 200 & 200 & -400 & 45 & 39 & 200 \\ 200 & 16 & 30 & 200 & 24 & 45 & -400 & 200 & 200 \\ 16 & 200 & 26 & 24 & 200 & 39 & 200 & -400 & 200 \\ 30 & 26 & 200 & 45 & 39 & 200 & 200 & 200 & -400 \end{bmatrix}$$

Solving QUBO gives y = - 982 at $x_1 = x_5 = x_9 = 1$ and all other variables = 0. Adjusting the constant to get the original objective equals 218.

EXAMPLE: THE QUADRATIC KNAPSACK PROBLEM

For the general case with n projects, the Quadratic Knapsack Problem (QKP) is commonly modeled as

$$\max \sum_{i=1}^{n-1} \sum_{j=i}^{n} v_{ij} x_i x_j$$



subject to the budget constraint

$$\sum_{j=1}^{n} a_j x_j \le b$$

where $x_j = 1$ if project j is chosen, else it is 0.

We re-cast this into the form of a QUBO model by first converting the constraint into an equation and then using the ideas embedded in Transformation # 1.

Consider the QKP model with four projects:

$$\max 2x_1 + 5x_2 + 2x_3 + 4x_4 + 8x_1x_2 + 6x_1x_3 + 10x_1x_4 + 2x_2x_3 + 6x_2x_4 + 4x_3x_4$$

subject to the knapsack constraint:

$$8x_1 + 6x_2 + 5x_3 + 3x_4 \leq 16$$

Introducing a slack variable:

$$8x_1 + 6x_2 + 5x_3 + 3x_4 + 1x_5 + 2x_6 = 16$$

Including the penalty term in the objective function

$$\begin{aligned} \max y = {} & 2x_1 + 5x_2 + 2x_3 + 4x_4 + 8x_1x_2 + 6x_1x_3 \\ & + 10x_1x_4 + 2x_2x_3 + 6x_2x_4 + 4x_3x_4 \\ & - P(8x_1 + 6x_2 + 5x_3 + 3x_4 + 1x_5 + 2x_6 - 16)^2 \end{aligned}$$

- Choosing a penalty P = 10, and cleaning up the algebra gives the QUBO model with an additive constant of -2560 and the Q matrix

$$\begin{bmatrix} 1922 & -476 & -397 & -235 & -80 & -160 \\ -476 & 1565 & -299 & -177 & -60 & -120 \\ -397 & -299 & 1352 & -148 & -50 & -100 \\ -235 & -177 & -148 & 874 & -30 & -60 \\ -80 & -60 & -50 & -30 & 310 & -20 \\ -160 & -120 & -100 & -60 & -20 & 600 \end{bmatrix}$$

- Solving QUBO gives y = 2588 at x = (1, 0, 1, 1, 0, 0).
- Adjusting for the additive constant, gives the value 28 for the original objective function.
- The QUBO approach to QKP has proven to be successful on problems with several hundred variables as shown in Glover, et. al. (2002).

Connections with Quantum Computing and Machine Learning

QUANTUM COMPUTING WITH QUBO:

Quantum Computing QUBO Developments:

- Significant fact: QUBO is equivalent to the famous Ising problem in physics. Physics approaches try to solve Ising problems with annealing.
- The D-Wave quantum computer - based on quantum annealing - nevertheless attempts to incorporate tabu search ideas to enhance its effectiveness.
- Another approach, called quantum gate (or quantum circuit) systems, is actively debated for its potential superiority over quantum annealing (adiabatic) systems.

QUANTUM-BRIDGE ANALYTICS :

- *Quantum-Bridge Analytics has emerged,* a field which devoted to bridging the gap between classical and quantum computational methods and technologies.
- National Academies of Sciences, Engineering and Medicine has released the Consensus Study Report titled *Quantum Computing: Progress and Prospects in 2019:*
  - "Formulating an R&D program with the aim of developing commercial applications for near-term quantum computing is critical to the health of the field ".
  - "Such a program will rest on developing "hybrid classical-quantum techniques," which is the focus of Quantum-Bridge Analytics "
  - Studies devoted to the use of Alpha-QUBO are currently underway to investigate the possibilities for achieving such speedup

MAJOR QUBO INITIATIVE: UNITE QUANTUM COMPUTING AND CLASSICAL COMPUTING

- Meta-Analytics represents the unification of metaheuristics and analytics, two fields of the foremost interest and practical importance for applications ranging from biotechnology to energy to logistics and financial planning.
- An important development Meta-Analytics has come about with the emergence of *Quantum-Bridge Analytics.*
- Exploit specific advantages unique to each computing paradigm
- The branch of Meta-Analytics is being actively pursued with the creation of the Alpha-QUBO solver, whose forerunner has been embodied in a hybrid classical-quantum method called *qbsolv*, which has been applied in a wide range of commercial and academic research settings.
- Goal: provide more effective solutions to QUBO and QUBO-related problems.

MACHINE LEARNING WITH QUBO:

Unsupervised Machine Learning with QUBO

- Salient form of unsupervised machine learning represented by clustering.
- The QUBO clique partitioning model provides a very natural form of clustering.
- CPP(clique partitioning problem) is popular in the area of machine learning as it offers a general model for the correlation clustering (CC) and the modularity maximization (MM), Charikar et al.(2008) and Caeri, et al.(2013).
- Application of QUBO to unsupervised machine learning in Glover et al. (2018) can be employed either together with quantum computing or independently.
- Recent use of clustering with QUBO models in Samorani et al. (2018) gives a foundation for studying additional uses of clustering.

MACHINE LEARNING WITH QUBO:

Supervised Machine Learning with QUBO:

- *Supervised Machine Learning with QUBO*: -- From the physics perspective, Schneidman, Berry, Segev and Bialek (2006) argue that the Ising model (which is equivalent to the QUBO model) is useful for neural network analysis.
- Consequently, the QUBO model has a natural role in statistical neural models of supervised machine learning.

MACHINE LEARNING TO IMPROVE QUBO SOLUTION PROCESSES

Glover, Lewis and Kochenberger (2017) introduce logical tests to learn relationships among variables in QUBO applications

- Achieved a 45% reduction in size for about half of the problems tested
- Succeeded in fixing all the variables in 10 cases, exactly solving these problems.
- Also identified implied relationships between pairs of variables to facilitate solving these problems.

Conclusions

REFERENCE

- **A Tutorial on Formulating and Using QUBO Models**
- https://arxiv.org/abs/1811.11538

- **Alpha QUBO**
- http://meta-analytics.net/Home/AlphaQUBO

ACKNOWLEDGEMENTS:

- Special thanks to Dr. Yu Du who assisted in putting the slides together.
- This tutorial was influenced by our collaborations on many papers over recent years with several colleagues to whom we owe a major debt of gratitude. These key co-workers, listed in alphabetical order, are: Bahram Alidaee, Dick Barr, Andy Badgett, Rajesh Chawla, Yu Du, Jin-Kao Hao, Mark Lewis, Karen Lewis, Zhipeng Lu, Abraham Punnen, Cesar Rego, Yang Wang, Haibo Wang and Qinghua Wu, etc.
- Other collaborators may be found listed as our coauthors on our home pages.