

# A QUBO Formulation of the $k$ -Medoids Problem

C. Bauckhage<sup>1,3,4</sup>, N. Piatkowski<sup>2</sup>, R. Sifa<sup>1,3</sup>, D. Hecker<sup>1,3</sup>, and S. Wrobel<sup>1,3,4</sup>

<sup>1</sup>Fraunhofer Center for Machine Learning, Sankt Augustin, Germany

<sup>2</sup>AI Group, TU Dortmund, Dortmund, Germany

<sup>3</sup>Fraunhofer IAIS, Sankt Augustin, Germany

<sup>4</sup>B-IT, University of Bonn, Bonn, Germany

**Abstract.** We are concerned with  $k$ -medoids clustering and propose a quadratic unconstrained binary optimization (QUBO) formulation of the problem of identifying  $k$  medoids among  $n$  data points without having to cluster the data. Given our QUBO formulation of this NP-hard problem, it should be possible to solve it on adiabatic quantum computers.

## 1 Introduction

Quadratic unconstrained binary optimization problems (QUBOs) are concerned with finding an  $n$ -dimensional binary vector that minimizes a quadratic objective

$$z^* = \operatorname{argmin}_{z \in \{0,1\}^n} z^T Q z + z^T q. \quad (1)$$

They thus pose combinatorial optimization problems which generally prove to be NP-hard. Indeed, difficult problems such as capital budgeting or task allocation in operations research, constraint satisfaction in AI, or maximum diversity, maximum clique, or graph partitioning in data mining and machine learning are but a few examples of where QUBOs occur in practice [1].

Owing to their practical importance, there exists a venerable literature on QUBOs and efficient solution strategies (for an extensive survey, see [2] and the references therein). More recently, however, interest in QUBOs has also arisen in a different context. Since adiabatic quantum computers such as produced by D-Wave [3,4] are designed to solve them, research on QUBO reformulations of various problems has noticeably intensified. Indeed, Boolean satisfiability [5], graph cuts [6,7,8], graph isomorphisms [9], binary clustering [6,10], or classifier training [11,12] have lately been solved via adiabatic quantum computing.

Following this line of research, we propose a QUBO formulation of the problem of identifying  $k$  medoids among  $n$  data points which, to our knowledge, has neither been attempted nor reported before. While we do not consider a quantum computing implementation itself, our result indicates that prototype extraction can be accomplished on quantum computers.

---

Copyright ©2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

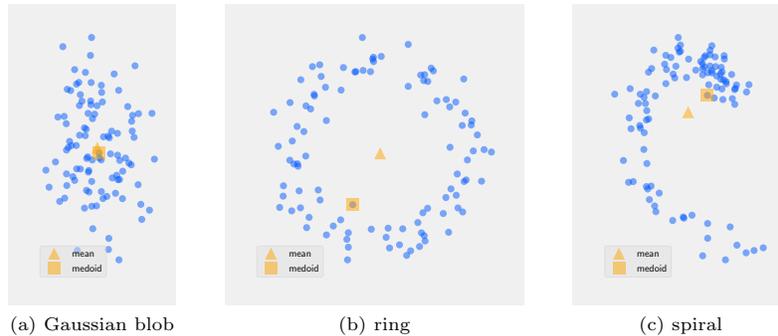


Fig. 1: Three didactic data samples together with their means and medoids.

Next, we briefly summarize the notion of a medoid and the ideas behind  $k$ -medoids clustering. We then present our QUBO formulation of the  $k$ -medoids problem and discuss simple examples that illustrate the practical performance of our approach. Finally, we review related work and summarize our contributions.

## 2 $k$ -Medoids Clustering

Consider a sample  $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  of  $n$  Euclidean data points  $\mathbf{x}_i \in \mathbb{R}^m$ . The well established *sample mean*

$$\boldsymbol{\mu} = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^m} \sum_{\mathbf{x}_i \in \mathcal{X}} \|\mathbf{x}_i - \mathbf{x}\|^2 = \frac{1}{n} \sum_{\mathbf{x}_i \in \mathcal{X}} \mathbf{x}_i \quad (2)$$

is a frequently used summary statistic for such data. The so called *sample medoid*

$$\mathbf{m} = \operatorname{argmin}_{\mathbf{x}_j \in \mathcal{X}} \sum_{\mathbf{x}_i \in \mathcal{X}} \|\mathbf{x}_i - \mathbf{x}_j\|^2 \quad (3)$$

on the other hand, is arguably less widely known but has interesting applications as well.

Figure 1 illustrates characteristics of means and medoids. Both minimize sums of distances to available sample points. However, while the mean is not necessarily contained in the sample, the medoid always coincides with an element of the sample and, for squared Euclidean distances as in (3), it is easy to prove that this element is the sample point closest to the mean [13].

An interesting feature of medoids is that they result from evaluating squared distances between given data only.<sup>1</sup> As these can be precomputed and stored in an  $n \times n$  distance matrix  $\mathbf{D}$  where  $D_{ij} = d^2(\mathbf{x}_i, \mathbf{x}_j)$ , medoids can be computed

<sup>1</sup> Note once again that  $\mathbf{x}$  in equation (2) may not be contained in  $\mathcal{X}$  whereas  $\mathbf{x}_i$  and  $\mathbf{x}_j$  in equation (3) always are.

---

**Algorithm 1**  $k$ -medoids clustering via Lloyd’s algorithm

---

**Require:** index set  $\mathcal{I} = \{1, \dots, n\}$ , distance matrix  $\mathbf{D} \in \mathbb{R}^{n \times n}$ , and parameter  $k \in \mathbb{N}$   
initialize set of  $k$  cluster medoid indices

$$\mathcal{M} = \{j_1, j_2, \dots, j_k\} \subset \mathcal{I}$$

**repeat**

**for**  $l = 1, \dots, k$  **do**  
determine cluster

$$\mathcal{C}_l = \{i \in \mathcal{I} \mid D_{ij_l} \leq D_{ij_q}\} \quad (4)$$

**for**  $l = 1, \dots, k$  **do**  
update medoid index

$$j_l = \operatorname{argmin}_{j \in \mathcal{C}_l} \sum_{i \in \mathcal{C}_l} D_{ij} \quad (5)$$

**until** clusters stabilize

---

from relational data. Contrary to means, medoids may thus also be estimated on sets of strings or graphs or other non-numeric data as long as there is an appropriate distance measure  $d(\cdot, \cdot)$  [13].

Moreover, since medoids coincide with actual data points, analysts often find them easy to interpret. This is why  $k$ -medoids clustering is an increasingly popular tool whenever explainable or physically plausible prototypes need to be determined [14,15,16,17,18].

Conceptually,  $k$ -medoids clustering is almost identical to  $k$ -means clustering. To accomplish it, we may, for instance, simply adapt Lloyd’s algorithm [19]. In other words, given randomly initialized medoids, we may determine clusters by assigning data points to their closest medoid, update the medoid of each cluster, and repeat these steps until clusters stabilize.

Note, however, that (local) medoids, too, are relational objects that can solely be determined from analyzing a distance matrix. Given such a distance matrix for a data set  $\mathcal{X}$ ,  $k$ -medoids clustering can thus be performed on index sets only. We demonstrate this in Algorithm 1 where  $\mathcal{I} = \{1, 2, \dots, n\}$  indexes the data in  $\mathcal{X}$ . Having initialized a set  $\mathcal{M} = \{j_1, j_2, \dots, j_k\} \subset \mathcal{I}$  of medoid indices, each cluster  $\mathcal{C}_l$  is an index set containing the indices of those data points  $\mathbf{x}_i$  that are closest to medoid  $\mathbf{m}_{j_l}$  (equation (4)). Given the clusters, each medoid index  $j_l$  is then set to the index of the medoid of the points indexed by  $\mathcal{C}_l$  (equation (5)).

Similar to  $k$ -means clustering,  $k$ -medoids clustering subdivides the input space into convex cells each of which is defined w.r.t. a prototype. Just as in  $k$ -means clustering via MacQueen’s algorithm [20],  $k$ -medoids clustering could therefore in principle be performed in two steps where medoids are determined first and data are assigned to them second. However, contrary to  $k$ -means clustering, the combinatorial nature of  $k$ -medoids clustering, which selects prototypes

from a discrete set of data points, prevents the use of continuous MacQueen-type updates of prototypes.

Indeed, we are not aware of prior work on how to find local medoids without having to compute clusters. The quadratic unconstrained binary optimization formulation of the problem which we present in the next section, however, can accomplish this.

### 3 A QUBO Formulation for $k$ -Medoids Estimation

Our idea for how to select  $k$  local medoids among  $n$  data points is based on an observation regarding classical  $k$ -means clustering.

Recall that, if a set  $\mathcal{X}$  of  $n$  data points is partitioned into  $k$  clusters  $\mathcal{C}_l$  of  $n_l$  elements whose mean is  $\boldsymbol{\mu}_l$ , adding the within cluster scatter

$$S_W = \sum_{l=1}^k \sum_{\mathbf{x} \in \mathcal{C}_l} \|\mathbf{x} - \boldsymbol{\mu}_l\|^2 \quad (6)$$

and the between cluster scatter

$$S_B = \frac{1}{2n} \sum_{i=1}^k \sum_{j=1}^k n_i n_j \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|^2 \quad (7)$$

yields a constant; that is, we have  $S_W + S_B = c$ .

This follows from Fisher’s analysis of variance [21,22] and is to say that the two scatter values are inverse under addition: either  $S_W$  is large and  $S_B$  is small, or  $S_W$  is small and  $S_B$  is large. Since a small within cluster scatter is usually taken to be the objective of  $k$ -means clustering, we therefore observe that *“good” cluster means are close to the data points within their cluster and, at the same time, far apart from each other.*

Consequently, we posit that the same should hold true for the medoids that are supposed to result from  $k$ -medoids clustering.

Note, however, that (6) involves variances about local centroids. Since medoids are not necessarily central, we henceforth work with a more robust measure of similarity. In particular, we consider Welsch’s  $M$ -estimator

$$\Delta_{ij} = 1 - \exp\left(-\frac{1}{2} D_{ij}\right) \quad (8)$$

which is known from robust regression [23] and also referred to as the correntropy loss [24,25].

#### 3.1 A QUBO to identify far apart data points

The problem of selecting  $k$  mutually far apart objects among a total of  $n$  objects is known as the max-sum dispersion problem [26] and can be formalized as

follows: given an  $n \times n$  similarity matrix  $\Delta$  over  $n$  objects which are indexed by  $\mathcal{I} = \{1, 2, \dots, n\}$ , determine a subset  $\mathcal{M}^* \subset \mathcal{I}$ ,  $|\mathcal{M}^*| = k$  such that

$$\begin{aligned} \mathcal{M}^* &= \operatorname{argmax}_{\mathcal{M} \subset \mathcal{I}} \frac{1}{2} \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{M}} \Delta_{ij} \\ \text{s.t.} \quad &|\mathcal{M}| = k. \end{aligned} \quad (9)$$

Looking at this problem, we note that, upon introducing binary indicator vectors  $\mathbf{z} \in \{0, 1\}^n$  whose entries are given by

$$z_i = \begin{cases} 1, & \text{if } i \in \mathcal{M} \\ 0, & \text{otherwise,} \end{cases} \quad (10)$$

it can also be written in terms of a constrained binary optimization problem

$$\begin{aligned} \mathbf{z}^* &= \operatorname{argmax}_{\mathbf{z} \in \{0,1\}^n} \frac{1}{2} \mathbf{z}^\top \Delta \mathbf{z} \\ \text{s.t.} \quad &\mathbf{z}^\top \mathbf{1} = k \end{aligned} \quad (11)$$

where  $\mathbf{1} \in \mathbb{R}^n$  denotes the vector of all ones.

If we further note that  $\mathbf{z}^\top \mathbf{1} = k \Leftrightarrow (\mathbf{z}^\top \mathbf{1} - k)^2 = 0$ , we find that (11) can equivalently be expressed as a quadratic unconstrained binary optimization problem, namely

$$\mathbf{z}^* = \operatorname{argmax}_{\mathbf{z} \in \{0,1\}^n} \frac{1}{2} \mathbf{z}^\top \Delta \mathbf{z} - \gamma (\mathbf{z}^\top \mathbf{1} - k)^2 \quad (12)$$

where  $\gamma \in \mathbb{R}$  is a Lagrange multiplier. Treating this multiplier as a constant and expanding the expression on the right hand side, we obtain

$$\mathbf{z}^* = \operatorname{argmax}_{\mathbf{z} \in \{0,1\}^n} \frac{1}{2} \mathbf{z}^\top \Delta \mathbf{z} - \gamma \mathbf{z}^\top \mathbf{1} \mathbf{1}^\top \mathbf{z} + 2\gamma k \mathbf{z}^\top \mathbf{1} - \gamma k^2 \quad (13)$$

$$= \operatorname{argmax}_{\mathbf{z} \in \{0,1\}^n} \mathbf{z}^\top \left( \frac{1}{2} \Delta - \gamma \mathbf{1} \mathbf{1}^\top \right) \mathbf{z} + 2\gamma k \mathbf{z}^\top \mathbf{1} - \text{const} \quad (14)$$

$$= \operatorname{argmin}_{\mathbf{z} \in \{0,1\}^n} \mathbf{z}^\top \left( \gamma \mathbf{1} \mathbf{1}^\top - \frac{1}{2} \Delta \right) \mathbf{z} - 2\gamma k \mathbf{z}^\top \mathbf{1}. \quad (15)$$

### 3.2 A QUBO to identify central data points

Using the above notation, the problem of selecting  $k$  most central objects among a total of  $n$  objects can be formalized as follows

$$\begin{aligned} \mathcal{M}^* &= \operatorname{argmin}_{\mathcal{M} \subset \mathcal{I}} \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{I}} \Delta_{ij} \\ \text{s.t.} \quad &|\mathcal{M}| = k. \end{aligned} \quad (16)$$

Introducing binary indicator vectors and a Lagrange multiplier, reasoning as above then leads to the following quadratic unconstrained binary optimization

problem

$$\mathbf{z}^* = \operatorname{argmin}_{\mathbf{z} \in \{0,1\}^n} \mathbf{z}^\top \mathbf{\Delta} \mathbf{1} + \gamma (\mathbf{z}^\top \mathbf{1} - k)^2 \quad (17)$$

$$= \operatorname{argmin}_{\mathbf{z} \in \{0,1\}^n} \mathbf{z}^\top (\gamma \mathbf{1} \mathbf{1}^\top) \mathbf{z} + \mathbf{z}^\top (\mathbf{\Delta} \mathbf{1} - 2\gamma k \mathbf{1}). \quad (18)$$

### 3.3 Putting both QUBOs together

In order to combine the QUBO that identifies far apart points (15) and the QUBO that identifies central points (18) into a single model that identifies rather central points that are rather far apart, we adhere to common practice [27]. That is, we introduce two additional tradeoff parameters  $\alpha, \beta \in \mathbb{R}$  to weigh the contributions of either model. This way, we obtain

$$\mathbf{z}^* = \operatorname{argmin}_{\mathbf{z} \in \{0,1\}^n} \mathbf{z}^\top (\gamma \mathbf{1} \mathbf{1}^\top - \alpha \frac{1}{2} \mathbf{\Delta}) \mathbf{z} + \mathbf{z}^\top (\beta \mathbf{\Delta} \mathbf{1} - 2\gamma k \mathbf{1}). \quad (19)$$

### 3.4 Choosing $\alpha$ , $\beta$ , and $\gamma$

Working with Welsch's function in (8) has the additional benefit that it maps squared distances such as  $\|\mathbf{x}_i - \mathbf{x}_j\|^2$  into the interval  $[0, 1]$ .

For the weighted, model specific contributions that occur in (19), we therefore have the following two upper bounds

$$\alpha \frac{1}{2} \mathbf{z}^\top \mathbf{\Delta} \mathbf{z} \leq \alpha \frac{1}{2} k^2 \cdot 1 \quad (20)$$

$$\beta \mathbf{z}^\top \mathbf{\Delta} \mathbf{1} \leq \beta n k \cdot 1. \quad (21)$$

This suggests to set the weighting parameters to  $\alpha = 1/k$  and  $\beta = 1/n$  so as to normalize the two contribution to about the same range. For the Lagrange multiplier which enforces solutions  $\mathbf{z}^*$  to have  $k$  entries equal to 1, we choose  $\gamma = 2$  so as to prioritize this constraint.

## 4 Practical Examples

Next, we present two admittedly simple experiments that illustrate the behavior of the QUBOs in (15), (18), and (19). Note again that our primary goal in this paper is to investigate whether it is possible to identifying  $k$  medoids among  $n$  data points without having to cluster the data; efficiency and scalability are not our main concerns. Correspondingly, we consider two samples of 2D data points that are small enough to allow for brute force estimation of the minimizers of the above QUBOs. In other words, in both experiments, we compute similarity matrices  $\mathbf{\Delta}$  over  $n \in \{12, 16\}$  data points, chose  $\alpha$ ,  $\beta$ , and  $\gamma$  as above and then evaluate each of the  $2^n$  possible solutions to the QUBOs in (15), (18), and (19) in order to determine the respective best one.

The data in our first experiment were deliberately chosen such that extremal, central, as well as locally medoidal data points can be easily identified by visual

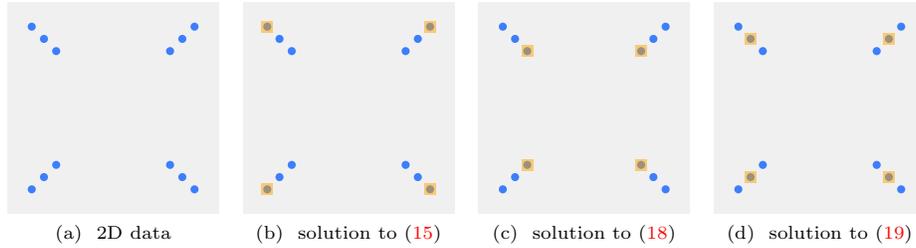


Fig. 2: Given a didactic sample of 2D data points forming 4 clusters, we solve the QUBOs in (15), (18), and (19) for  $k = 4$ . While the solution to (15) identifies the 4 most extremal data points, the solution to (18) identifies the 4 most central ones. Solving (19) trades off both these results and identifies 4 local medoids.

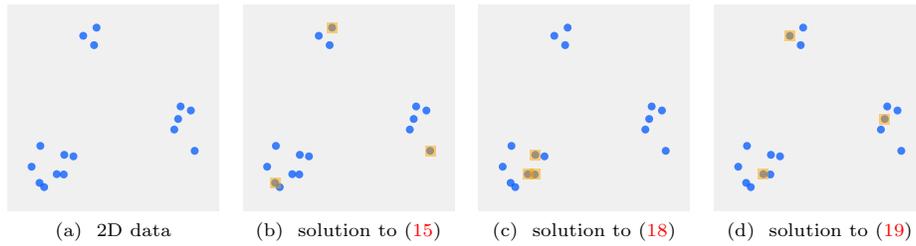


Fig. 3: Given 16 data points sampled from 3 bivariate Gaussian distributions, we solve the QUBOs in (15), (18), and (19) for  $k = 3$ . Again, the solution to (15) identifies the 3 rather extremal data points and the solution to (18) corresponds to 3 rather central ones. The solution to (19) identifies 3 local medoids.

inspection. They can be seen in Fig. 2a which shows  $n = 12$  two-dimensional data points which form 4 apparent clusters. Setting  $k = 4$  and solving the QUBOs in (15), (18), and (19) produces indicator vectors that identify the data points highlighted in Fig. 2b–2d. Looking at these results suggests that they are well in line with what human analysts would deem reasonable.

In our second experiment, we generated 100 sets of  $n = 16$  data points sampled from 3 bivariate Gaussian distributions; Fig. 3a shows an example. Setting  $k = 3$ , we then solved the QUBOs in (15), (18), and (19). Exemplary results can be seen in Fig. 3b–3d. The three extremal data points in Fig. 3b make intuitive sense. The fact that the three rather central data points in Fig. 3c are all situated in the cluster to the bottom left is due to the fact that this is the largest of the three clusters; here, our use of Welsch’s  $M$ -estimator has the effect that points which are central w.r.t. several nearby points are being favored. The local medoids highlighted in Fig. 3d are again intuitive.

For each of the 100 data sets considered in this experiment, we also compared the  $k = 3$  medoids obtained from solving (19) to those produced by Algorithm 1.

In each of our 100 tests, we found them to be identical. While such a perfect agreement between both methods is likely an artifact of the small sample sizes we worked with, it nevertheless corroborates the utility of our QUBO based approach to the  $k$ -medoids problem.

## 5 Related Work

As of this writing, there are three major algorithmic approaches to  $k$ -means clustering, namely those due to Lloyd [19], MacQueen [20], and Hartigan and Wong [28]. Lloyd- and Hartigan-type approaches alternately update means and clusters and have been applied to the combinatorial problem of  $k$ -medoids clustering, too. Examples for the former can be found in [13] and [29]; examples of the latter include the algorithms PAM and CLARA as well as variations thereof [30,31,32].

However, to the best of our knowledge, MacQueen type procedures, which determine local means without having to compute clusters, have not yet been reported for the  $k$ -medoids setting. The QUBO presented in equation (19) fills this gap. While it is not an iterative method such as MacQueen’s procedure, it nevertheless suggests that medoids, too, can be estimated without clustering.

Our derivation of a QUBO for the  $k$ -medoids problem was motivated by the quest for quantum computing solutions for relational or combinatorial clustering. Here, most prior work we are aware of has focused on solutions for  $k = 2$  [6,7,8]. Work on solutions for  $k > 2$  can be found in [33].

However, the methods proposed there are similar in spirit to kernel  $k$ -means clustering in that they operate on binary cluster membership indicator matrices  $\mathbf{Z} \in \{0, 1\}^{k \times n}$ . This is to say that they perform quantum clustering in a manner that produces clusters but does not identify cluster prototypes. Our approach in this paper, however, only requires binary indicator vectors  $\mathbf{z} \in \{0, 1\}^n$  and is, once again, able to identify  $1 \leq k \leq n$  prototypes without having to compute clusters.

## 6 Summary

In this paper, we have proposed a quadratic unconstrained binary optimization (QUBO) formulation of the problem of identifying  $k$  local medoids within a sample of  $n$  data points. The basic idea is to trade off measures of central and extremal tendencies of individual data points. Just as conventional approaches to  $k$ -medoids clustering, our solution works with relational data and therefore applies to a wide range of practical settings. However, to the best of our knowledge, our solution is the first that is capable of extracting  $k$  local medoids without having to compute clusters.

This capability comes at a price. While conventional  $k$ -medoids clustering as well as our QUBO formulation constitute NP-hard combinatorial problems, the former can be solved (approximately) by means of greedy heuristics. Yet, for the latter, such heuristics are hard to conceive.

However, an aspect of our model that may soon turn into a viable advantage is that it can be easily solved on quantum computers. While this paper did not consider corresponding quantum computing implementations, they are the topic of ongoing work and results will be reported once available.

## References

1. Kochenberger, G., Glover, F.: A Unified Framework for Modeling and Solving Combinatorial Optimization Problems: A Tutorial. In Hager, W., Huang, S.J., Pardalos, P., Prokopyev, O., eds.: *Multiscale Optimization Methods and Applications*. Volume 82 of NOIA. Springer (2006)
2. Kochenberger, G., Hao, J.K., Glover, F., Lewis, M., Lü, Z., Wang, H., Wang, Y.: The Unconstrained Binary Quadratic Programming Problem: A Survey. *J. of Combinatorial Optimization* **28**(1) (2014)
3. Johnson, M., et al.: Quantum Annealing with Manufactured Spins. *Nature* **473**(7346) (2011)
4. D-Wave press release: D-Wave announces D-Wave 2000Q quantum computer and first system order (Jan 2017)
5. Farhi, E., Goldstone, J., Gutmann, S., Sipser, M.: Quantum Computation by Adiabatic Evolution. arXiv:quant-ph/0001106 (2000)
6. Bauckhage, C., Brito, E., Cvejovski, K., Ojeda, C., Sifa, R., Wrobel, S.: Ising Models for Binary Clustering via Adiabatic Quantum Computing. In: *Proc. EMMCVPR*. Volume 10746 of LNCS., Springer (2017)
7. Ushijima-Mwesigwa, H., Negre, C., Mniszewski, S.: Graph Partitioning Using Quantum Annealing on the D-Wave System. In: *Proc. Int. Workshop on Post Moores Era Supercomputing*, ACM (2017)
8. Jünger, M., Lobe, E., Mutzel, P., Reinelt, G., Rendl, F., Rinaldi, G., Stollenwerk, T.: Performance of a Quantum Annealer for Ising Ground State Computations on Chimera Graphs. arXiv:1904.11965 [cs.DS] (2019)
9. Calude, C., Dinneen, M., Hua, R.: QUBO Formulations for the Graph Isomorphism Problem and Related Problems. *Theoretical Computer Science* **701** (2017)
10. Bauckhage, C., Ojeda, C., Sifa, R., Wrobel, S.: Adiabatic Quantum Computing for Kernel  $k=2$  Means Clustering. In: *Proc. KDML-LWDA*. (2018)
11. Pudenz, K., Lidar, D.: Quantum Adiabatic Machine Learning. *Quantum Information Processing* **12**(5) (2013)
12. Adachi, S., Henderson, M.: Application of Quantum Annealing to Training of Deep Neural Networks. arXiv:1510.06356 [quant-ph] (2015)
13. Bauckhage, C.: NumPy / SciPy Recipes for Data Science: k-Medoids Clustering. researchgate.net (Feb 2015)
14. Drachen, A., Sifa, R., Thureau, C.: The Name in the Game: Patterns in Character Names and Game Tags. *Entertainment Computing* **5**(1) (2014)
15. Bauckhage, C., Drachen, A., Sifa, R.: Clustering Game Behavior Data. *IEEE Trans. on Computational Intelligence and AI in Games* **7**(3) (2015)
16. Caro, M., Aarva, A., Deringer, V., Csanyi, G., Laurila, T.: Reactivity of Amorphous Carbon Surfaces: Rationalizing the Role of Structural Motifs in Functionalization Using Machine Learning. *Chemistry of Materials* **30**(21) (2018)
17. Molina, A., Vergari, A., Di Mauro, N., Natarajan, S., Esposito, F., Kersting, K.: Mixed Sum-Product Networks: A Deep Architecture for Hybrid Domains. In: *Proc. AAAI*. (2018)

18. Johnson, M., et al.: A Universal Probe Set for Targeted Sequencing of 353 Nuclear Genes from Any Flowering Plant Designed Using k-Medoids Clustering. *Systematic Biology* (12 2018)
19. Lloyd, S.: Least Squares Quantization in PCM. *IEEE Trans. Information Theory* **28**(2) (1982)
20. MacQueen, J.: Some Methods for Classification and Analysis of Multivariate Observations. In: *Proc. Berkeley Symp. on Mathematical Statistics and Probability*. (1967)
21. Fisher, R.: On the Probable Error of a Coefficient Correlation Deduced from a Small Sample. *Metron* **1** (1921)
22. Bauckhage, C.: *k*-Means and Fisher's Analysis of Variance. [researchgate.net](https://www.researchgate.net) (May 2018)
23. Dennis, J., Welsch, R.: Techniques for Nonlinear Least Squares and Robust Regression. *Communications in Statistics – Simulation and Computation* **7**(4) (1978)
24. Liu, W., Pokharel, P., Principe, J.: Correntropy: Properties and Applications in Non-Gaussian Signal Processing. *IEEE Trans. on Signal Processing* **55**(11) (2007)
25. Feng, Y., Huang, X., Shi, L., Yang, Y., Suykens, J.: Learning with the Maximum Correntropy Criterion Induced Losses for Regression. *J. of Machine Learning Research* **16** (2015)
26. Ravi, S., Rosenkrantz, D., Tayi, G.: Heuristic and Special Case Algorithms for Dispersion Problems. *Operations Research* **42**(2) (1994)
27. Lucas, A.: Ising Formulations of Many NP Problems. *Frontiers in Physics* **2**(5) (2014)
28. Hartigan, J., Wong, M.: Algorithm AS 136: A *k*-Means Clustering Algorithm. *J. of the Royal Statistical Society C* **28**(1) (1979)
29. Park, H.S., Jun, C.H.: A Simple and Fast Algorithm for K-Medoids Clustering. *Expert Systems with Applications* **36**(2) (2009)
30. Kaufman, L., Rousseeuw, P.: Partitioning Around Medoids (Program PAM). In: *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons (1990)
31. Kaufman, L., Rousseeuw, P.: Clustering Large Applications (Program CLARA). In: *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons (1990)
32. Schubert, E., Rousseeuw, P.: Faster k-Medoids Clustering: Improving the PAM, CLARA, and CLARANS Algorithms. [arXiv:1810.05691 \[cs.LG\]](https://arxiv.org/abs/1810.05691) (2018)
33. Kumar, V., Bass, G., Tomlin, C., Dulny III, J.: Quantum Annealing for Combinatorial Clustering. *Quantum Information Processing* **17**(2) (2018)