An Evaluation of a Hierarchical Clustering Method Using Quantum Annealing

Introduction – Quantum Annealing and Clustering –

- ✓ **Digital computers** (e.g. CPU, Vector Processer Unit (VPU), GPU)
  - ○ Multiplicity of various problems
  - × Performance limited by the Moore's law
  - × Huge amounts of power consumption
- ✓ **Quantum annealing (QA)** (on Quantum Processer Unit (QPU))
  - ○ Power efficient
  - ○ Specialized to **Combinatorial Optimization Problems**
  - × Limited multiplicity of some problems
  - → **Hybrid computing with digital computers and QA**
- ✓ **Hierarchical clustering using QA**
  - ○ Possibility of accelerating clustering
  - × Small problem size due to the limitation of qubits in QA

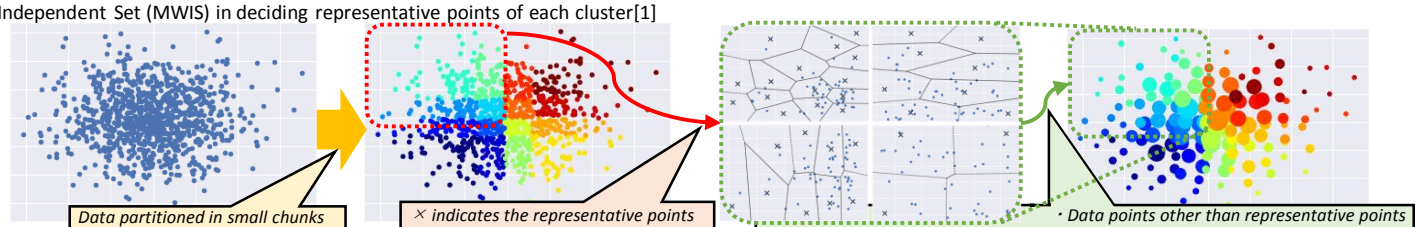
- ✓ **Non-hierarchical clustering** (like K-means)
  - ○ Low computational complexity
  - × Need to know the number of clusters in advance
- ✓ **Hierarchical clustering**
  - × High computational complexity
  - ○ No need to know the number of clusters in advance
  - → **Need to use the appropriate clustering method depending on the situation**

Objective

Need to clarify features of each processor and each clustering method by comparing the execution time and quality of the clustering method

A Combinatorial Optimization Problem in Clustering

Hierarchical agglomerative clustering (HAC), one of the hierarchical clustering methods, using the combinatorial optimization problem of Maximum Weighted Independent Set (MWIS) in deciding representative points of each cluster[1]



✓***How to choose representative points by MWIS***

- *A chunk having $\{x_1,\ldots,x_n\}$ and each weight of w_i*
- *ε defined as degree of similarity*
- *Binary variables S obtained as solutions of MWIS*

① *Similarity matrix* $N_{ij}^{(\epsilon)} = \begin{cases} 1 & \text{if } distance(x_i, x_j) < \varepsilon \\ 0 & otherwise \end{cases}$ ② *Quadratically Constrained Quadratic Program (QCQP) given by* ③ $\underset{s\in(0,1)^n}{maximize} \sum_{i=1}^{n} s_i w_i \quad subject\ to \sum_{i=1}^{n}\sum_{i<j} s_i N_{ij}^{(\epsilon)} s_j = 0$

1. *QCQP is being solved by* ***Greedy algorithm***
2. *QCQP is transformed into QUBO[2] to solve by* ***SA*** *or* ***QA***

Performance Evaluation

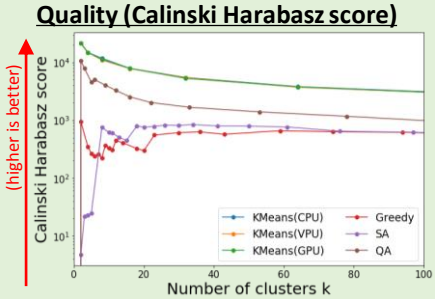
Data set : MoCap Hand Postures

- 5 types of hand postures from 14 users in a motion capture environment
- Number of data is 78095 and number of features used for experiments in 9 of 36 features

Environments

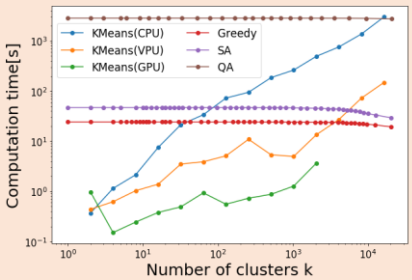
Method		Hardware	Software
K-means	CPU	Intel Xeon Gold 6126	Scikit-learn v0.22.0
	VPU	NEC Vector Engine Type 10B	Frovedis v0.9.5
	GPU	NVIDIA Tesla V100	Rapidsai v0.12.0a
SA		Intel Xeon Gold 6126	OpenJij v0.0.9
Greedy			-
QA		D-Wave 2000Q	Ocean SDK v1.4.0

Quality (Calinski Harabasz score)



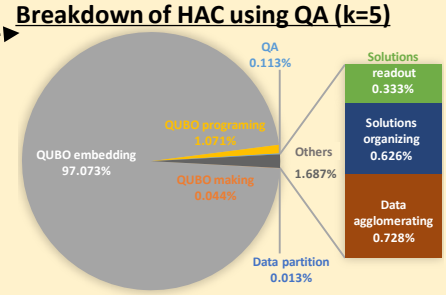
- QA achieves the best quality in HAC
  - The quality of solutions of QCQP is important for HAC
    - QA can search the whole search space by quantum fluctuations
    - SA cannot improve the quality by increasing the annealing time to explore the search space
    - Greedy falls a local optimum and cannot find the best solution
- K-means is always superior to HAC
  - K-means does not agglomerate data and can avoid loss of information of the original data

Execution time



- Quite low performance of QA
  - QA still needs to be accelerated
- Steady execution times of HAC (Greedy, SA, and QA)
  - It is because the number of processed data decreases by agglomerating data for each hierarchy
- The execution times of K-means increase as the number of clusters increases
  - The execution time of K-means of CPU, VPU, and GPU is long in the cases that the number of cluster is large
- HAC is effective when the number of clusters is large, even if the number of clusters is known

Breakdown of HAC using QA (k=5)



- The time of QA is short
- QUBO embedding executed on a CPU dominates the whole execution time
  - QUBO embedding duplicates data to make up for missing connections because of not all qubits connected in QPU

Discussion : towards the high performance clustering

- HAC ; The number of clusters is unknown
- K-means ; After the number of clusters is decided by HAC
  - Combination of the hierarchical clustering on QA and non-hierarchical clustering on digital computers can be the promising

Conclusions and Future Work

- By using QA in HAC, the quality of the clustering results becomes higher than those of SA and Greedy.
- HAC is effective not only when the number of clusters is unknown but also when the number of clusters is known and large.
- As future work, we will combine hierarchical clustering and non-hierarchical clustering for the large-scale clustering.

References

[1] Tim, J. et al. :A Quantum Annealing-Based Approach to Extreme Clustering, FICC 2020, 2020 , DOI: 10.1007/978-3-030-39442-4_15.

[2] Boros, E. et al. :Local search heuristics for quadratic unconstrained binary optimization (QUBO), *Journal of Heuristics*, 2007, 13.2: 99-132.