Quantum Optimization for Location Assignment Problem in ASSR

Kuniaki Satori[1], and Nobuyuki Yoshikawa[2]

[1,2] *Mitsubishi Electric Information Reaserch and Development Laboratory, Kamakura, Kanagawa, 247-8501, Japan*
Satori.Kuniaki@bc.MitsubishiElectric.co.jp

Abstract

In an Automated Storage and Retrieval System (AS/RS), a location assignment of products is important to improve the picking efficiency. In this paper, the optimization of shelf location assignment with a quantum annealing is investigated. Product pairs are considered in order of picking frequency and are assigned to empty shelves in order of distance from an outlet. Then swapping the position of product in the pair is considered as the decision variable. This reduces the number of required qubits and guarantees the feasibility of solution. The efficiency of quantum algorithm is evaluated by comparing with mixed integer programming (MIP).

1. Introduction

AS/RS (Automated Storage and Retrieval System) is an automated storage and transportation system that enables efficient warehouse management by automating the accommodation and storage of products. Automation has advantages such as reduction of labor costs and working hours, improvement of work quality, and accurate management of inventory differences (Roodbergen & Vis, 2009). However, picking efficiency from automated warehouses is affected by the location of products placed on shelves, and if efficiency is poor, it becomes a bottleneck in the entire shipping process. In order to improve picking efficiency, products with high retrieve frequency should be placed near the retrieve port. Or, if you're using a two-fork crane, it is necessary to arrange the shelves close to each other for products that are likely to be retrieved at the same time. Such problems are generally treated as placement optimization problems. The facility layout problem plays a crucial role in various real-world domains. It involves optimizing the arrangement of multiple elements or objects under given resources and constraints. Examples include facility placement(de Vries, van de Klundert, & Wagelmans, 2020), delivery route optimization(Aljohani, 2023) and factory layout design(Li, Wang, Fan, Yu, & Chu, 2021). Solving facility layout problems offers several benefits, such as optimal resource utilization, efficient process design, cost reduction, and time savings. Proper layout configurations contribute to effective resource management and improved service, enhancing the competitiveness of businesses and organizations. Prior research has proposed a method of formulating and optimizing product placement with mixed interger programming (MIP) so as to minimize shipping time. This allows us to find the optimal solution for the ideal combination of products and shelves(Chen, Huang, Danielczuk, Ichnowski, & Goldberg, 2022; Kovács, 2011). While MIP exhibits high accuracy, it is known that execution time increases exponentially with problem size.

First Author et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

In this paper, the optimization of shelf position assignment will be investigated using quantum annealing, a computational technique that is expected to optimize combinatorial problems quickly and accurately. However, considering all shelves and products increases the number of constraints and qu bits, so it is not suitable for large problems. Therefore, a method is proposed to perform global optimization by formulating the problem as a swap problem between two shelves and performing local optimization multiple times. The proposed method can approach global optimization in a short time compared to conventional methods, and improves computational feasibility for large-sized problems.

2. Preliminaries

2.1. Assignment Optimization in AS/SR

This paper considers product location allocation to improve picking efficiency. As a warehouse layout that shortens retrieve time, it is conceivable to locate products that are frequently shipped near the retrieve port, and to locate closely related products close to each other. In this Location Assignment problem in AS/RS, retrieving assumes a crane with two forks, so up to two products can be retrieved at the same time. When retrieving a product from a shelf, there is a time T_p for the crane to pull the product out of the shelf using a fork in addition to the travel time. Normally, picking two products from a shelf takes $2T_p$ time, but picking two adjacent products can be done in T_p time.

In this paper, use the expected retrieve time to evaluate the combination of products placed on the shelf. The expected retrieve time is calculated by multiplying the product retrieval frequency P and the retrieval time according to the shelf position. The retrieve time based on the shelf position is the sum of the travel time $T_{ss}(s_E, s)$ from the exit s_E to s and the product picking time T_p. The following formula is shown as a formulation of the expected retrieve time of this warehouse.

Variables

$$\begin{aligned}
b &= \text{product index},\\
s &= \text{shelf position index},\\
s_E &= \text{retrieve exit port},\\
T_p &= \text{time to pick product from shelf},\\
N_L &= \text{shelf row size},\\
P(b_0, b_1) &= \text{joint retrieve probability of } b_0 \text{ and } b_1,\\
T_{ss}(s_0, s_1) &= \text{travel time from } s_0 \text{ to } s_1.
\end{aligned}$$

Formulas

$$E_s(s,b) = P(b,\emptyset)(2T_{ss}(s_E,s)+T_p), \tag{1a}$$

$$\begin{aligned}E_{ds}(s_0,s_1,b_0,b_1) = P(b_0,b_1)(T_{ss}(s_E,s_0)\\ + T_{ss}(s_0,s_1)+T_{ss}(s_1,s_E)+2T_p),\end{aligned} \tag{1b}$$

$$\begin{aligned}E_{dd}(s_0,s_1,b_0,b_1) = P(b_0,b_1)(2\min(T_{ss}(s_E,s_0)\\ ,T_{ss}(s_E,s_1))+T_p),\end{aligned} \tag{1c}$$

$$E_d(s_0,s_1,b_0,b_1) = \begin{cases} E_{dd}(s_0,s_1,b_0,b_1) & \text{if } |s_0-s_1|-N_L=0,\\ E_{ds}(s_0,s_1,b_0,b_1) & \text{otherwise.}\end{cases} \tag{1d}$$

(1a) expected time to retrieve one product, (1b) expected time to pick and retrieve two products separately, (1c) expected time to pick and retrieve two products at the same time, (1d) function that uses E_{dd} if the absolute value of the difference between product shelf positions s_0 and s_1 matches the number of rows N_L on the shelf, and E_{ds} otherwise.

2.1.1. All-Layout MIP

The Location Assignment problem in AS/RS is formulated in many studies. and one is formulated as a mixed integer programming (MIP). MIP is a method used in mathematical optimization where some or all of the variables are required to be integers. This technique is particularly useful when dealing with discrete decision variables, such as the number of units to produce, which cannot be expressed as real numbers. MIP allows for the formulation of complex constraints and objective functions, making it a versatile tool for a wide range of optimization problems. MIP searches for the optimal combination among the formulated patterns. Applying MIP to this problem structure is described as follows. This MIP formulation assumes that all products can be placed on all shelves. Henceforth, this formulation is called All-Layout formulation.

Variables

$$\begin{aligned}
x_{s,b} &= 1 \rightarrow \text{product b is set into shelf s},\\
y_{s_0,s_1,b_0,b_1} &= 1 \rightarrow x_{s_0,s_0} = 1 \text{ and } x_{s_1,s_1} = 1.
\end{aligned}$$

Minimize

$$\begin{aligned}&\sum_{b\in B}\sum_{s\in S} E_s(s,b)x_{s,b}\\ &\quad + \sum_{b_0<b_1\in B}\sum_{s_0<s_1\in S} E_d(s_0,s_1,b_0,b_1)y_{s_0,s_1,b_0,b_1}.\end{aligned} \tag{2a}$$

subject to

$$\sum_{b\in B} x_{bs} \le 1, \quad \forall s \in S, \tag{2b}$$

$$\sum_{s\in S} x_{bs} = 1, \quad \forall b \in B, \tag{2c}$$

$$\begin{aligned}y_{s_0,s_1,b_0,b_1} \ge x_{m,b_0} + x_{m',b_1} - 1, \quad &\forall b_0 < b_1 \in B,\\ &\forall s_0 < s_1 \in S,\\ &m \ne m' \in \{s_0, s_1\}.\end{aligned} \tag{2d}$$

(2a) the objective is to minimize a average expected retrieve time for all buckets are place on the shelf. x is a two-dimensional binary variable that specifies the shelf position and bucket type. (2b) each bucket $b \in B$ should be assigned only one to a shelf. (2c) each shelf $s \in S$ should be assigned only one to a bucket. (2d) the flag variable y is designed so that the variables x have an AND relationship.

2.1.2. Swap MIP

In the All-layout MIP mentioned earlier, calculation is difficult because the number of variables increases enormously as the number of shelves increases. In this paper, a problem formulation for swapping of shelf positions between two products. Henceforth, this formulation is called Swap formulation. This makes it possible to significantly reduce the number of variables. This Swap formulation can only perform local optimizations at one time. However, multiple iterations of optimization bring it closer to global optimization.

Variables

$$\begin{aligned}
x_a &= 1 \rightarrow \text{swap positions within pair a},\\
y_{a_0,a_1,i,j} &= 1 \rightarrow x_{s_0,s_0} = i \text{ and } x_{s_1,s_1} = j,\\
I &= [i, \bar{i}],\\
J &= [j, \bar{j}].
\end{aligned}$$

Minimize

$$\sum_{a \in A}((E_s(s_{a_0}, b_{a_0}) + E_s(s_{a_1}, b_{a_1}))(1 - x_a) + (E_s(s_{a_0}, b_{a_1}) + E_s(s_{a_1}, b_{a_0}))x_a) + \sum_{a_0 < a_1 \in A} \sum_{i=(0,1)} \sum_{j=(0,1)} \sum_{k=(0,1)} \sum_{l=(0,1)} E_d(s_{a_0,i}, s_{a_1,j}, b_{a_0,k}, b_{a_1,l}) y_{i,j,I_k,J_l}. \tag{3a}$$

subject to

$$y_{a_0,a_1,0,0} \geq (1 - x_{a_0}) + (1 - x_{a_1}) - 1, \tag{3b}$$
$$y_{a_0,a_1,0,1} \geq (1 - x_{a_0}) + x_{a_1} - 1, \tag{3c}$$
$$y_{a_0,a_1,1,0} \geq x_{a_0} + (1 - x_{a_1}) - 1, \tag{3d}$$
$$y_{a_0,a_1,1,1} \geq x_{a_0} + x_{a_1} - 1. \tag{3e}$$

(3a) the objective to swap the shelf positions within pairs $a \in A$ of two products to minimize the expected retrieve time. x is a one-dimensional binary variable that specifies the pair. A pair consists of two products and their corresponding shelf positions, where the two products in the pair a are $(b_{a,0}, b_{a,1})$, the two shelf positions are $(s_{a,0}, s_{a,1})$. (3b, 3c, 3d, 3e) the flag variable y is designed so that the variables x have a certain relationship.

2.2. Quantum Annealing

Quantum annealing is an optimization technique based on the principles of quantum mechanics. In this approach, the problem is encoded using quantum bits (or qubits), which represent the physical system, and the optimal solution is sought through the evolution of the quantum state(Kadowaki & Nishimori, 1998; Ohzeki et al., 2018). Specifically, in quantum annealing, the cost function of the problem is defined as the energy function of the qubits. Initially, the cost function is high, and the quantum state gradually evolves, exploring the quantum state corresponding to the desired minimum cost (optimal solution). By decreasing a parameter controlling the rate of change, the system can leverage quantum tunneling and thermal fluctuations to increase the likelihood of converging to the optimal solution. A prominent architecture for quantum annealing is the quantum annealer provided by D-Wave Systems(Johnson et al., 2011). This architecture implements qubits using superconducting circuits and performs annealing operations. D-Wave Systems offer a combination of hardware and software solutions to apply quantum annealing to various optimization problems.

In the quantum annealing the optimization problem is required to be described as a the quadratic unconstrained binary optimization (QUBO) form as the following:

$$f_q(x) = x^T Q x. \tag{4}$$

where $f_q() : \mathbb{B}^n \to \mathbb{R}$ is the objective function to be minimzed, $x \in \mathbb{B}^n$ is a binary decision variables, and $Q \in \mathbb{R}^{n \times n}$ is a coefficient matrix to represent the problem. This coefficient matrix is mentioned as QUBO matrix.

3. Quantum Assignment Optimization

In this paper, we have formulated QUBO to solve the warehouse problem in quantum annealing.

3.1. All Layout Formulation

This section changes All-Layout formulation to QUBO formulation.

Variables

$$x_{s,b} = 1 \to \text{product b is set into shelf s},$$

Minimize

$$H(x) = \frac{H_c(x)}{\sum_x H_c(x)} + \lambda_b H_b(x) + \lambda_s H_s(x), \tag{5a}$$

$$H_c(x) = \sum_{s \in S} \sum_{b \in B} E_s(s, b) x_{s,b} + \sum_{s_0 \neq s_1 \in S} \sum_{b_0 < b_1 \in B} E_d(s_0, s_1, b_0, b_1) x_{s_0,b_0} x_{s_1,b_1}. \tag{5b}$$

subject to

$$H_b(x) = \sum_{b \in B} (\sum_{s \in S} x_{s,b} - 1)^2, \tag{5c}$$

$$H_s(x) = \sum_{b_0 < b_1 \in B} \sum_{s \in S} x_{s,b_0} x_{s,b_1}. \tag{5d}$$

(5b) the objective is to minimize a average expected retrieve time for all buckets are place on the shelf. It consists of an expectated retrieve time formula and two penalty formulas. (5c) each bucket $b \in B$ should be assigned no more than 1 to a shelf. (5d) each shelf $s \in S$ should be assigned only one to a bucket.

3.2. Swap Formulation

This section changes Swap formulation to QUBO formulation.

Variables

$$x_a = 1 \to \text{swap positions within pair a}.$$

Minimize

$$H(x) = H_c(x), \tag{6a}$$

$$\begin{aligned}H_c(x) = \sum_{a\in A} &((E_s(s_{a_0}, b_{a_0}) + E_s(s_{a_1}, b_{a_1}))(1 - x_a) \\ &+ (E_s(s_{a_0}, b_{a_1}) + E_s(s_{a_1}, b_{a_0}))x_a) \\ + &\sum_{a_0<a_1\in A} (\\ &(1 - x_{a_0})(1 - x_{a_1}) \sum_{i=(0,1)} \sum_{j=(0,1)} E_d(s_{0,i}, s_{1,j}, b_{0,i}, b_{0,j}) \\ &+ x_{a_0}(1 - x_{a_1}) \sum_{i=(0,1)} \sum_{j=(0,1)} E_d(s_{0,i}, s_{1,j}, b_{0,\bar{i}}, b_{0,j}) \\ &+ (1 - x_{a_0})x_{a_1} \sum_{i=(0,1)} \sum_{j=(0,1)} E_d(s_{0,i}, s_{1,j}, b_{0,i}, b_{0,\bar{j}}) \\ &+ x_{a_0}x_{a_1} \sum_{i=(0,1)} \sum_{j=(0,1)} E_d(s_{0,i}, s_{1,j}, b_{0,\bar{i}}, b_{0,\bar{j}}).\end{aligned} \tag{6b}$$

(6b) the objective is to minimize a average expected retrieve time for all buckets are place on the shelf. Consists only of the expected retrieve time expression and has no constraints.

4. Numerical Experiments

In this chapter, some experiments will be conducted. Experiments use the python library pymip for MIP formulation and CBC(COIN-OR Brand-and-Cut) as the solver. Henceforth, when using the solver CBC, it is written as MIP (PythonMIP-team, 2023). QUBO formulation uses python library pyqubo, and SA(simulated annealing), QA, and HybridQA are used as solvers (Zaman, Tanahashi, & Tanaka, 2021). The matrix building and MIP solving is performed on the Intel core i7 2.8GHz 4 cores cpu, 32BG memory. In addition, QA uses ”D-wave’s DW_2000Q_6” solver and HybridQA uses D-wave’s ”LeapHybridSampler” solver in the QPU calculations that appear in the following experiments. HybridQA is a sampler that combines quantum and traditional computing resources. Table 1 summarizes the features of each solver.

4.1. Validation of Expected Retrieve Time Indicator

The formulation of this paper uses the expected retrieve time as the objective function. In this section, the validity of this objective function is shown by comparing the expected retrieve time with the actual simulated retrieve time. In the experiment, the expected retrieve time and the simulated retrieve time were measured for randomly generated shelf arrangements. The experiment was performed 100 times and the results are averaged. In order to measure the departure time of the simulation, 100 retrieve were performed and the results were averaged. For the shelf size of the experiment, 14 size [(2, 2), (3, 2), (4, 2), (6, 3), (8, 4), (12, 6), (16, 8), (20, 10), (24, 12), (28, 14), (32, 16), (36, 18), (40, 20), (44, 22)] consisting of (row, column) was used. The method of creating a pair of swap formulation is to randomly select two products from the shelf.

Figure 1 shows the evaluation for each shelf size with expected retrieve time and simulation retrieve time. There is no big difference between the two ratings. This result shows that minimizing the expected retrieve time leads to minimizing the actual retrieve time, and this objective function is valid.

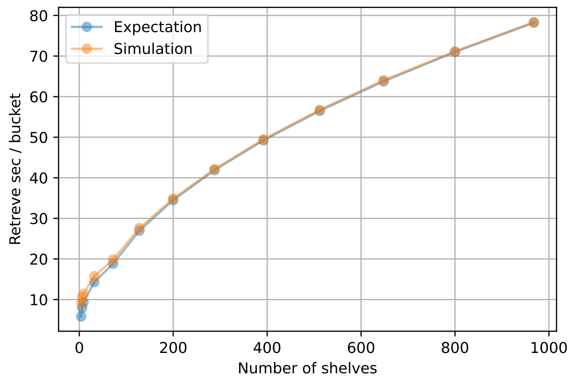
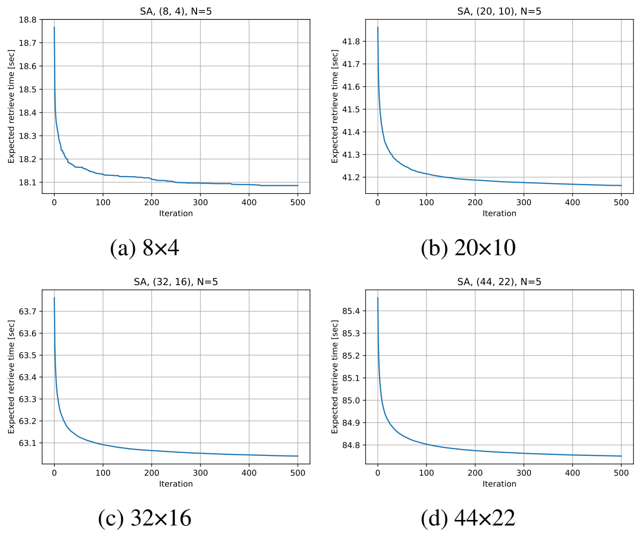


Figure 1. Comparison of expectation and simulation

4.2. Iterative Swap Optimization

The swap problem formulation defined in the previous chapter can only be optimized locally in a single optimization. However, iterative optimization of the swap problem can yield intermittent solution improvements. In this section it is shown that iterative optimization of the Swap problem improves the solution. The experiment was performed 5 times and the results are averaged.

Figure 2, 500 iterative optimizations using SA for each shelf size (8×4), (20×10), (32×16), (44×22) gone. The expected retrieve time decreases as the number of iterations increases. This result shows that the Swap formulation improves the evaluation value by iteratively optimizing. There is a rapid improvement in ratings for the first 100 iterations of any problem size. From this result, in the subsequent experiments, iterative swap formulations are conducted up to 100 iterations.



(a) 8×4 (b) 20×10 (c) 32×16 (d) 44×22

Figure 2. Iterative Swap Optimization for each problem size

	Strong point	Weak point
MIP	- High solution quality.	- Computation time increases as the size of the problem increases - The number of decision variables explodes due to constraints
SA	- Fast approximate solution. - Flexible.	- No guarantee of convergence to optimal solution - A good solution requires a large number of trials
QA	- Fast search even for complex problems. - Less likely to fall into local optima.	- There are limits on the number of qubits and embedding - Can only be used on the internet - Complex problems reduce the quality of solutions
HybridQA	- Similar to QA - Better solution quality compared to QA.	- Similar to QA - Expensive to use computational resources compared to QA

Table 1. Characteristics of each method

4.3. Performance Verification

In this section, a performance comparison between the All-Layout formulation and the Swap formulation is presented. The solvers used in each formulation are MIP, SA, QA, and HybridQA. The experiment was performed 5 times and the results are averaged. Since there is a performance difference depending on the number of iterations in the Swap formulation, the graphs are plotted for 1, 10, and 100 iterations, respectively. For the shelf size of the experiment, 14 size [(2, 2), (3, 2), (4, 2), (6, 3), (8, 4), (12, 6), (16, 8), (20, 10), (24, 12), (28, 14), (32, 16), (36, 18), (40, 20), (44, 22)] consisting of (row, column) was used. Some data are not plotted on the graph because the calculation time was too long or the calculation could not be performed due to embedding error on the Dwave side. Constraint weights of All-Layout QUBO formulation are set to $\lambda_b = 1$, $\lambda_s = 1$.

Figure 3 shows the delta improved by the optimization compared to the expected retrieve time for a random initial placement. Swap MIP and Swap QA could not be calculated after (16, 8), Swap SA and Swap HybridQA could be calculated up to (44, 22). There is no difference in solution quality results between the Swap formulations, and the overlapping lines in the figure indicate that all methods yield similar solutions. The result obtained by iterative optimization of the Swap formulation is close to the result of All-Layout MIP, which could be calculated up to (4,2). This shows that iterative optimization of the Swap formulation yields high quality solutions. Comparing the All-Layout formulation and the Swap formulation increases the computable problem size by a factor of nearly 100. Also, compared to MIP, HybridQA and SA increase the computable size by nearly 100 times. However, in (44, 22), SA takes nearly 10 times longer than HybridQA. From this, it can be seen that using the HybridQA solver in the Swap formulation can expand the computable problem size and reduce the computation time. Figure 4 shows the total computational time required for optimization of each method. Regarding QA and HybridQA, only QPU (Quantum Processing Unit) time required for calculation on the quantum computer does not include communication time with D-wave or embedding time. In the All-Layout MIP, the calculation time increases greatly as the number of shelves increases.

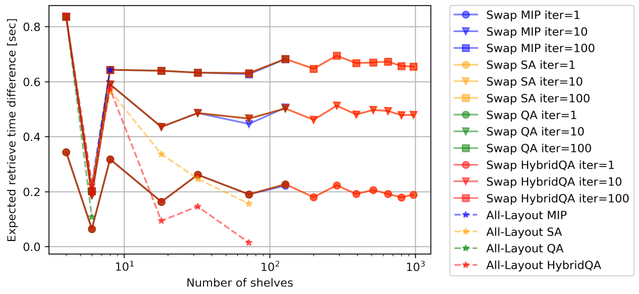


Figure 3. Comparison of expected retrieve times

On the other hand, the swap formulation does not increase the calculation time even if the number of shelves increases. Both QA and HybridQA formulations show small increase in computation time with increasing problem size. However, the All-Layout formulation became incomputable for large problems. Figure 5 shows the total optimization calculation

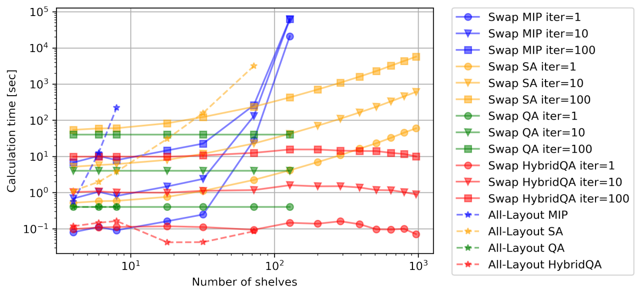


Figure 4. Comparison of pure calc times

time including communication time with each D-wave and embedding time. There is no change other than swap QA and swap HybridQA. Consideration is required when actually using QA in optimization problems. Even if communication time and embedding time are included, the rate of increase in swap HybridQA computation time is not large. On the other hand, swap QA has a large increase in computation time due to an increase in embedding time. From these results, swap formulation realizes reduction of computation time and expansion of computability by simplifying the

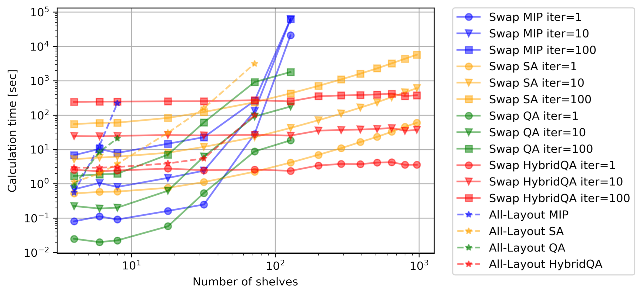


Figure 5. Comparison of actual calc times

problem compared to All-Layout formulation. Furthermore, when comparing swap SA and swap HybridQA, SA and HybridQA have similar performance, but the computation time of HybridQA does not increase exponentially. From this, it was shown that the computation time can be greatly reduced by using quantum Computation.

4.4. Examination of Pairing Method

In this section, verification is performed on the pair creation method for iterative optimization of the swap formulation. There are simply two ways to pair two products. The first method is to randomly select two products and pair them. The second method is to pair products that are adjacent to each other in the order of delivery time on the shelf. Random pairs and adjacent pairs are considered to correspond to search and knowledge use, respectively. Therefore, the performance is evaluated by the ratio ϵ of the two methods. The experiment was performed 10 times and the results are averaged.

Figure 6, evaluation values are compared at each ϵ value, with ϵ creating adjacent pairs and $(1-\epsilon)$ creating random pairs. $\epsilon = 0.1$ has the highest performance. From this result, it is expected that setting ϵ to an appropriate value depending on the task will improve the performance.

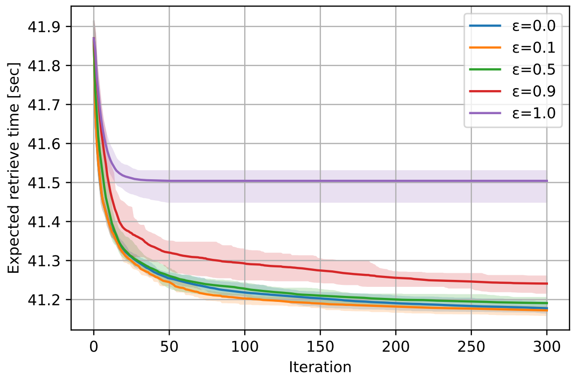


Figure 6. Comparison of Expected Retrieve Time by ϵ

5. Conclusion

In this paper, the Swap fomulation was proposed as a problem formulation to solve the Location Assignment Problem in AS/SR. A Swap QUBO formulation was also proposed to solve it with QA. Experiments show that the swap formulation makes it possible to calculate a problem size that cannot be calculated and reduces the computation time compared to the conventional All-Layout formulation. In addition, it has been shown that the use of quantum computing can significantly reduce computation time. In the future, we will aim at efficient search by improving the pair creation method when iterating the Swap formulation.

References

Aljohani, K. (2023). Optimizing the distribution network of a bakery facility: A reduced travelled distance and food-waste minimization perspective. *Sustainability*, *15*(4), 1-26.

Chen, L. Y., Huang, H., Danielczuk, M., Ichnowski, J., & Goldberg, K. (2022). Optimal shelf arrangement to minimize robot retrieval time. *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, 993-1000.

de Vries, H., van de Klundert, J., & Wagelmans, A. P. (2020). The roadside healthcare facility location problem a managerial network design challenge. *Production and Operations Management*, *29*(5), 1165-1187. doi: https://doi.org/10.1111/poms.13152

Johnson, M., Amin, M., Gildert, S., Lanting, T., Hamze, F., Dickson, N., … Rose, G. (2011, 05). Quantum annealing with manufactured spins. *Nature*, *473*, 194-8. doi: 10.1038/nature10012

Kadowaki, T., & Nishimori, H. (1998, Nov). Quantum annealing in the transverse ising model. *Phys. Rev. E*, *58*, 5355–5363. doi: 10.1103/PhysRevE.58.5355

Kovács, A. (2011). Optimizing the storage assignment in a warehouse served by milkrun logistics. *International Journal of Production Economics*, *133*, 312-318.

Li, H., Wang, Y., Fan, F., Yu, H., & Chu, J. (2021). Sustainable plant layout design for end of life vehicle recycling and disassembly industry based on slp method, a typical case in china. *IEEE Access*, *9*, 81913-81925. doi: 10.1109/ACCESS.2021.3086402

Ohzeki, M., Takahashi, C., Okada, S., Terabe, M., Taguchi, S., & Tanaka, K. (2018). Quantum annealing: next-generation computation and how to implement it when information is missing. *Nonlinear Theory and Its Applications, IEICE*, *9*(4), 392-405. doi: 10.1587/nolta.9.392

PythonMIP-team. (2023). *coin-or/python-mip*. `https://github.com/coin-or/python-mip`. GitHub.

Roodbergen, K. J., & Vis, I. F. (2009). A survey of literature on automated storage and retrieval systems. *European Journal of Operational Research*, *194*(2), 343-362. doi: https://doi.org/10.1016/j.ejor.2008.01.038

Zaman, M., Tanahashi, K., & Tanaka, S. (2021). *Pyqubo: Python library for mapping combinatorial optimization problems to qubo form.*