

Petri net-based QUBO Model Formulation for Multi-Resource Flow-Shop Scheduling Problems

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

4th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

5th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

Abstract—This paper introduces a Petri net-based approach to addressing multi-resource flow-shop scheduling problems within multi-objective quantum optimization. The multi-resource flow-shop problem, which holds both theoretical and practical significance, represents a real-world application scenario. The practical nature of the problem necessitates incorporating multiple objective functions, which adds complexity to its formulation, particularly in representing the energy function within the QUBO (Quadratic Unconstrained Binary Optimization) model. Our approach mitigates the complexity of formulating the QUBO model by employing Petri net theory, providing a more efficient solution for these inherently complex problems. The paper further demonstrates the effectiveness of this method through computational examples utilizing a CPU-based QUBO optimization platform.

I. INTRODUCTION

A combinatorial optimization problem involves identifying an optimal solution that either maximizes or minimizes an objective function subject to constraints. Typical instances of such problems include the traveling salesman problem, the knapsack problem, and various scheduling problems. However, many combinatorial optimization problems fall under the category of NP-hard. Consequently, as the number of potential solutions increases exponentially, it becomes challenging to find the optimal solution within a feasible timeframe consistently. To mitigate this challenge, researchers have developed exact algorithms that minimize computational complexity by efficiently narrowing the solution space and heuristic algorithms that efficiently identify feasible solutions that approximate the optimal solution.

Quantum annealing and quantum approximate optimization algorithms (QAOA) represent optimization algorithms designed for quantum computers [1], [2]. Both approaches formulate the optimization problem as an energy function, represented either in the QUBO (Quadratic Unconstrained Binary Optimization) model or the Ising model, and are anticipated to generate solutions near the optimal solution efficiently. QAOA, being a quantum gate-based optimization algorithm, requires

further advancements in quantum hardware to enable large-scale optimization computations. Conversely, quantum annealing has already seen practical applications, with D-Wave's commercial quantum annealing machine capable of processing tens of thousands of qubits. Additionally, quantum-inspired systems, which perform optimization computations digitally based on the QUBO or Ising model, and systems that emulate annealing processes using GPUs have been proposed. These emerging platforms are gaining attention as novel solutions for combinatorial optimization.

In our previous study, we developed an algorithm to formulate QUBO and Ising models based on Petri net modeling. Our approach can drastically reduce the formulation's hardness because we can model target optimization problems with Petri nets, and it requires only the domain knowledge of the problem. This paper examines multi-resource flow shop scheduling problems to minimize the resource costs associated with each task and the waiting time between tasks. Furthermore, the study investigates the extent to which the system can handle an increase in the number of jobs and whether appropriate hyperparameters can be determined under such conditions.

II. PRERIMINARIES

A. Petri net

A Petri net $PN = (N, M_0)$ is represented by a directed bipartite graph consisting of two types of nodes, places, and transitions, denoted as $N = (P, T, Pre, Post)$, along with an initial marking M_0 [3], [4].

$P = \{p_1, p_2, \dots, p_n\}$ represents the set of places, and $T = \{t_1, t_2, \dots, t_n\}$ represents the set of transitions. The state of the current system can be represented by the placement of tokens in the places. $Pre(p, t)$ denotes the arc weight that connects the input place p to the transition t . $Post(p, t)$ denotes the arc weight that connects the output place p from the transition t . A transition t is said to be enabled when the number of tokens in its input places meets or exceeds the

weight of the corresponding arcs. The firing of a transition represents the occurrence of an event, which leads to a change in the distribution of tokens.

B. Multi-Resource Flow-Shop Scheduling Problem

The multi-resource flow-shop scheduling problem (MRFSSP) is a type of scheduling problem that involves processing multiple jobs, each composed of several operations. Each job is comprised of multiple operations, and each operation requires the use of designated shared resources. A single processing operation within a job is referred to as a task [5].

The MRFSSP addressed in this paper is summarized as follows:

- 1) The number of jobs is N , and each job consists of M tasks (corresponding to operations).
- 2) The operating cost per unit time is predetermined for each machine.
- 3) The processing time per unit operation length is predetermined for each machine.
- 4) Each job must process M tasks in a specified order, which is the same for all jobs.
- 5) Each task is processed using a machine from a predetermined set of machine resources. Once a task has started, it must be completed without interruption.
- 6) Each task cannot begin processing until the preceding task of the same job is completed (precedence constraint).
- 7) Every task of every job is assigned to a machine (completion constraint).
- 8) There is no resource contention between any two tasks (conflict constraint).

Additionally, the following two objectives are considered as the objective functions:

- 9) Minimize the total waiting time for resources across all tasks
- 10) Minimize the total resource costs

III. PETRI NET-BASED QUBO FORMULATION FOR MRFSSP

A. Petri Net Model

The Petri net model for the multi-resource flow shop scheduling problem can be represented as shown in Fig. 1. Fig. 1 simplifies the representation by showing just three jobs.

Each job is depicted as a path consisting of an alternating sequence of places and transitions. For example, the path corresponding to Job 1, $P_{1,1}, T_{1,1}, P_{1,2}, \dots, P_{1,4}$, is composed of three types of tasks (operations) $T_{1,1}, T_{1,2}, T_{1,3}$, each requiring machine resources R_0, R_1 , and R_2 , respectively.

The three types of machine resources are represented as colored tokens, strings containing information such as machine ID, processing speed, and machine cost. Due to space constraints, detailed explanations are omitted here.

As demonstrated by this Petri net model, given sufficient domain knowledge, one can construct a Petri net model based on a few simple Petri net description rules.

B. QUBO Formulation

In our previous research, we proposed a basic approach to QUBO formulation. This paper applies our approach to formulate the MRFSSP from the Petri net model.

The variables used in these equations are summarized in TABLE I.

TABLE I
SIMBOLS IN ENERGY FUNCTIONS

Symbol	Explanation
rc^r	Cost of using machine resource r per unit time
fd^r	Processing time when using machine resource r to process a task
t_i^j	The i -th task of job j
$x_k^r(t_i^j)$	Binary variable indicating $x_k^r(t_i^j) = 1$ if t_i^j uses machine resource r at time k , otherwise 0

To comply with the firing rules of the Petri net, it is necessary to minimize the following energy functions to zero. Specifically, for both E_{c1} and E_{c2} , a transition can only fire if all tokens are present in its input places. Additionally, firing multiple transitions based on the same token in a single input place is prohibited. In the case of competing transitions, the token must be utilized by only one transition. Based on these firing rules, the following energy functions can be generated:

$$E_{c1} = \sum_{k_1, k_2} \sum_r \sum_{(j_1, j_2)} \sum_i x_{k_1}^r(t_{i_1}^{j_1}) \cdot x_{k_2}^r(t_{i_2}^{j_2}) \quad (1)$$

$$E_{c2} = \sum_{k_1, k_2} \sum_{r_1, r_2} \sum_j \sum_i x_{k_1}^{r_1}(t_i^j) \cdot x_{k_2}^{r_2}(t_{i+1}^j) \quad (2)$$

On the other hand, since the scheduling problem aims to create an execution plan for tasks until all jobs are completed, each transition must fire exactly once. Therefore, the following energy function can be generated:

$$E_{c3} = \left(1 - \sum_k \sum_r \sum_j \sum_i x_k^r(t_i^j) \right)^2 \quad (3)$$

As the objective function is to minimize the total resource cost and the waiting time for task processing, the following energy functions can be derived from the behavior model of the Petri net:

$$E_{o1} = \sum_k \sum_r \sum_j \sum_i rc^r \cdot fd^r \cdot x_k^r(t_i^j) \quad (4)$$

$$E_{o2} = \sum_i \sum_{k_1, k_2} \sum_{r_1, r_2} \sum_j \sum_i (k_1 - fd^{r_2}) \cdot x_{k_2}^{r_2}(t_{i+1}^j) \cdot x_{k_1}^{r_1}(t_i^j) \quad (5)$$

By assigning weights A, B, C, D , and E to the energy functions in equations (1), (2), (3), (4), and (5), the overall energy function can be expressed as follows:

$$E = A \cdot E_{c1} + B \cdot E_{c2} + C \cdot E_{c3} + D \cdot E_{o1} + E \cdot E_{o2} \quad (6)$$

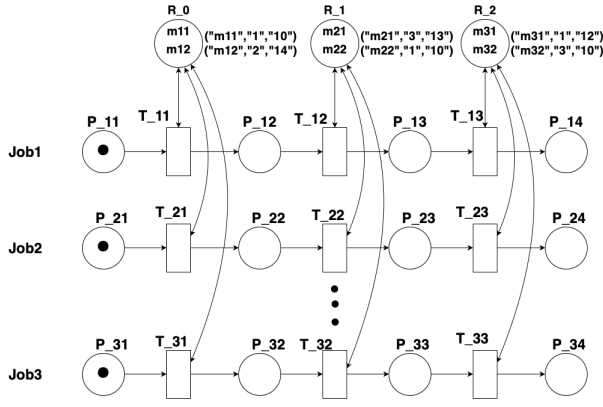


Fig. 1. Petri net model of a flow-shop system

IV. COMPUTATIONAL EVALUATION

Optimization calculations were conducted using OpenJij. The experimental environment is configured as follows:

Hardware Environments

- CPU: Apple M1 3.2 GHz
- Memory: 8 GB RAM

OS and Libraries:

- OS: macOS Sonoma 14.3.1
- Programming Language and Libraries: Python 3.10.14, openjij 0.9.2, pyqubo 1.4.0

The weight parameters A , B , and C for the constraints (1), (2), and (3) were fixed at 600, 150, and 250, respectively, based on prior adjustments. Evaluation experiments were conducted by varying only the weight parameters D and E of the two objective functions (4) and (5). For each of the 30 different combinations of D and E , optimization calculations were performed 20 times, and the average data was obtained using only the feasible solutions. Additionally, the probability of obtaining a feasible solution was calculated from the 20 runs.

The energy diagram of the objective functions when fixing D to one of the values in 1, 5, 8, 10, 13 and varying the parameter E is shown in Fig. 2.

Average Resource Cost and Waiting Time by Pattern and Iteration=100

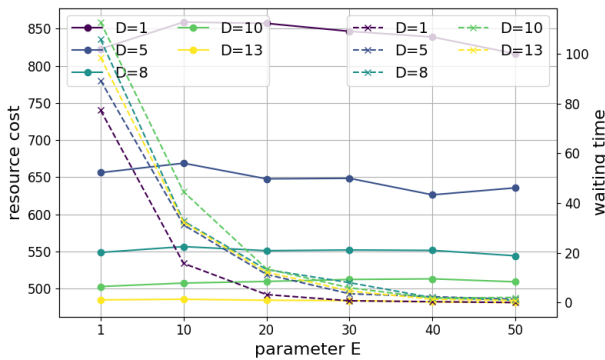


Fig. 2. Energy Value for Resource Cost and Waiting Time

TABLE II
PROBABILITY OF OBTAINING FEASIBLE SOLUTIONS

(D, E)	success[%]	(D, E)	success[%]
(1,1)	100	(8,30)	70
(1,10)	95	(8,40)	90
(1,20)	100	(8,50)	80
(1,30)	100	(10,1)	60
(1,40)	100	(10,10)	65
(1,50)	100	(10,20)	100
(5,1)	65	(10,30)	90
(5,10)	45	(10,40)	80
(5,20)	65	(10,50)	90
(5,30)	75	(13,1)	100
(5,40)	70	(13,10)	85
(5,50)	90	(13,20)	90
(8,1)	55	(13,30)	100
(8,10)	60	(13,40)	100
(8,20)	75	(13,50)	100

From Fig. 2, it can be confirmed that as parameter E increases, the dashed line corresponding to E_{o2} decreases. Additionally, when the value of parameter D is large, the solid line corresponding to E_{o1} shows lower energy, indicating that more flexible resource allocation is possible.

As observed from TABLE II, there is a tendency for the success rate to increase as the value of parameter E increases.

V. CONCLUSION

In this study, we formulated a QUBO model based on Petri nets to address the flow shop scheduling problem. We fixed the number of jobs at 15. We evaluated not only the probability of obtaining feasible solutions under different parameter settings but also the quality of the solutions to assess the effectiveness of the parameter settings.

The results indicated that the quality of the solutions and the probability of obtaining feasible solutions varied depending on the settings of parameters D and E . This suggests that, within the context of the flow shop scheduling problem, the parameter settings of the objective functions can contribute to the efficiency of scheduling, particularly when the constraint parameters are fixed. Consequently, the importance of adjusting parameters to balance solution quality and computational efficiency was demonstrated.

REFERENCES

- [1] Bingzhi Zhang, Akira Sone, and Quntao Zhuang. Quantum computational phase transition in combinatorial problems. *npj Quantum Information*, Vol. 8, No. 1, p. 87, 2022.
- [2] Tadashi Kadowaki and Hidetoshi Nishimori. Quantum annealing in the transverse ising model. *Phys. Rev. E*, Vol. 58, pp. 5355–5363, Nov 1998.
- [3] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, Vol. 77, No. 4, pp. 541–580, 1989.
- [4] Kurt Jensen, Lars Michael Kristensen, and Lisa Wells. Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. *International Journal on Software Tools for Technology Transfer*, Vol. 9, No. 3, pp. 213–254, 2007.
- [5] Housseem Eddine Nouri, Olfa Belkahla Driss, and Khaled Ghédira. Solving the flexible job shop problem by hybrid metaheuristics-based multiagent model. *Journal of Industrial Engineering International*, Vol. 14, No. 1, pp. 1–14, 2018.