2021 IEEE Congress on Evolutionary Computation (CEC) | 978-1-7281-8393-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/CEC45853.2021.9504701

Two Stage Quantum Optimization for the School Timetabling Problem

Otto Menegasso Pires
Informática e Estatística, Universidade Federal de Santa Catarina
Florianópolis, Brasil
otto28mai@gmail.com

Rafael de Santiago
Informática e Estatística, Universidade Federal de Santa Catarina
Florianópolis, Brasil
r.santiago@ufsc.br

Jerusa Marchi
Informática e Estatística, Universidade Federal de Santa Catarina
Florianópolis, Brasil
jerusa.marchi@ufsc.br

***Abstract*— Timetabling is an NP-Hard problem that searches for periodic scheduling of events that must meet a set of hard and soft constraints. Because of the difficulty in finding an exact solution, the use of heuristics to address the problem is a common practice. When only the hard constraints are considered, the timetabling problem can be reduced to graph vertex coloring. This similarity between both problems has motivated the use of graph coloring heuristics as a means to solve the timetabling problem. We propose the use of the Quantum Approximate Optimization Algorithm as a heuristic to solve the school timetabling problem. The QAOA is a hybrid quantum-classical algorithm that can be used to address combinatorial optimization problems. We simulated QAOA in a minimal example with 42 qubits using the Ket Quantum Programming Language and our results showed that it is possible to apply QAOA to the school timetabling problem.**

***Index Terms*—qaoa, quantum computing, school timetabling, timetabling problem**

I. Introduction

Timetables are an important factor of many organizations. Public transportation, schools and universities regularly use timetables to organize time, to such an extent that it is difficult to imagine an organized and modern society that does not use them. Despite its widespread use, it is unknown a polynomial-time exact method to construct timetables. The problem of constructing a timetable is considered hard in computation [1]. Factors such as the limitation of resources, e.g. people or time, or the aspects that differentiate the quality of a timetable increase its difficult. Many works addressed this problem, such as [2]–[10] and the need for better timetables solutions has still motivated the research of current and better ways to make them [11]–[13].

Timetabling is a combinatorial optimization problem. [14] These problems are about finding the optimal solution to an assignment of discrete values to variables according to some criteria, called hard and soft constraints. The hard constraints define the feasibility of the solution and the soft constraints define the quality of it. For the school timetabling problem, the optimal solution is defined as a feasible timetable that satisfies all soft constraints. Finding an exact solution to the timetabling problem is considered NP-Hard [15]. Because of that, metaheuristics have been widely employed to solve timetabling problem, instead of using exact methods [16].

Considering only the hard constraints, the Timetabling problem can be represented as a graph vertex coloring problem. Graph vertex coloring serves as a general model for conflict resolution. When applied to the school timetabling problem, a conflict graph is created where each edge represents a time conflict of a student or a teacher. Coloring the conflict graph is equivalent to find the length of time periods required to schedule the lectures without time conflict. The similarity between Timetabling and Graph Coloring has motivated the use of graph coloring heuristics to solve the timetabling problem [3], [17].

Our work proposes the use of Quantum Approximate Optimization Algorithm (QAOA) [18] as a heuristic to solve the timetabling problem. The QAOA algorithm is an hybrid quantum-classical algorithm that uses a parameterized quantum circuit with a classical optimization process over the quantum gates parameters. It is expected QAOA to run in near term quantum devices since its complexity grows linearly according to a parameter p, that regulates the number of applications of the quantum circuit. QAOA was used before to address combinatorial optimization problems, such as Vehicle Routing Problem (VRP) [19] and Tail Assignment Problem [20].

It remains an open question how the performance of QAOA compares with existing classical algorithms for most problems. In [21], Bravyi et al. showed that QAOA for MaxCut with a constant depth performs worse than the best known classical algorithm on some classes of instances. This result opened up a new approach to QAOA, where, instead of an approximation algorithm, it can be used as a heuristic optimizer. Moussa et al [22] expand on this idea and developed a method to find graph instances where QAOA could provide advantage over the classical algorithms for the MaxCut problem. We follow the same way, using QAOA to obtain good heuristics to solve timetabling problem when represented as a coloring graph problem.

This work is partly supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

978-1-7281-8393-0/21/$31.00 ©2021 IEEE

This work is divided as follows. In Section II we introduce the School Timetabling Problem and the QAOA Algorithm Section III contextualize the school timetabling problem for our case. Section IV explains our application of QAOA on the timetabling problem. Section V shows the results of our experiments and our analysis and Section VI concludes the paper, bringing some final considerations and perspectives of future works.

II. Background

A. School Timetabling

Since 1963, automated timetabling problems have been addressed. They can be classified into three main problems [5]:

- School timetabling: periodic scheduling for classes and teachers of a school, avoiding time conflicts;
- Course timetabling: periodic scheduling for lectures from a set of university courses, minimizing the overlaps of lectures of courses having the same students;
- Examining timetabling: scheduling of exams of university courses, avoiding overlaps of exams having common students, locally spreading students as much as possible.

The work related in this paper is about school timetabling problem.

In [3], Werra described a basic class-teacher formulation. Assume that $C = \{c_1, c_2, \ldots, c_m\}$ is the set of classes; $T = \{t_1, t_2, \ldots, t_n\}$ is the set of teachers; $R \in \mathbb{Z}^{n \times m}$ is a matrix in which r_{ij} is the number of lectures involving teacher t_i and class c_j. In addition, consider there are p periods of time. Given the set of variables $x_{ijk} \in \{0, 1\}$ for all $i = 1, 2, \ldots n$, $j = 1, 2, \ldots m$, and $k = 1, 2, \ldots p$, the basic class-teacher formulation can be solved by finding an assignment to x variables such that the number of lectures mapped in R should be addressed:

$$\sum_{k=1}^{p} x_{ijk} = r_{ij} \qquad \forall i = 1, \ldots, n; j = 1, \ldots, m \tag{1}$$

each teacher cannot have more than one lecture for a class by time

$$\sum_{i=1}^{n} x_{ijk} \leq 1 \qquad \forall j = 1, \ldots, m; k = 1, \ldots, p \tag{2}$$

each class cannot have more than one lecture given by a teacher by time

$$\sum_{j=1}^{m} x_{ijk} \leq 1 \qquad \forall i = 1, \ldots, n; k = 1, \ldots, p. \tag{3}$$

Although this formulation can be solved in polynomial-time for two teachers, the formulation with more than two teachers is NP-Complete. Other models with a small increment lead to NP-Complete problems [1].

With basic class-teacher formulation, it is possible to associate a bipartite multigraph $G = (C, T, R')$ in which teachers and classes are nodes, and R' is the collection of edges between teachers and classes. There are r_{ij} parallel edges $\{t_i, c_j\} \in R'$. So, the problem to find a feasible assignment to x can be addressed as a coloring problem: finding an assignment of one among p colors to each edge such that no two adjacent edges have the same color. Each x_{ijk} has value 1 iff some edge $\{t_i, c_j\}$ gets the color k [3].

Werra [3] suggests new problems by changing the constraints of basic class-teacher formulation like: (*i*) including a maximum number of lectures in which a teacher or a class can be assigned; and (*ii*) if there is a solution for any p.

Hilton [2] described a school timetable as a matrix M in which $m_{ij} = \{w_1, w_2, \ldots\}$ for a given teacher $i \in T$ and a class $j \in C$. Each $w_k \in m_{ij}$ is a period of the week. Considering the cell m_{ij}, no other cell in the same row or column can have a symbol $w_k \in m_{ij}$. These constraints ensure that no teacher or class should be assigned twice a time.

There are problems in which is almost impossible to meet all constraints. So, the timetabling can be considered as a search problem, in which it is desirable that the maximum number of constraints be satisfied. It is also possible to address timetabling problems as an optimization problem, in which is required to find a timetable that satisfies all the hard constraints and minimizes (or maximizes) a given objective function composed of the soft constraints [5].

B. Quantum Approximate Optimization Algorithm

The Quantum Approximate Optimization Algorithm (QAOA) is a hybrid quantum-classical algorithm developed by Farhi et al. [23]. The algorithm was developed to give approximate optimization solutions for combinatorial optimization problems. Later, the algorithm was expanded to deal with more general families of operators [18]. This expansion enabled QAOA to deal with optimization problems, such as exact optimization and sampling problems. It is particularly useful for optimization problems with hard and soft constraints.

An instance of an optimization problem is a pair (F, f) where F is the domain and $f : F \to \mathbb{R}$ is the objective function to be optimized. The domain F is the feasible subset of a configuration space, for implementation of a QAOA circuit, the configuration space is encoded into a subspace of Hilbert space of a multiqubit system [18] [24]. Given a string $x \in F$, we define the phase hamiltonian

$$C = \sum_{k=1}^{m} C_k(x), \tag{4}$$

where C_k represents each constraint to be satisfied by string x. The operator acts on a quantum state $|x\rangle$, that encodes the string x, as

$$C |x\rangle = \sum_{k=1}^{m} C_k |x\rangle = f(x) |x\rangle . \tag{5}$$

The QAOA quantum circuit, as shown in Figure 1, is described by p alternating applications of two parameterized families of operators, the phase separator $U_P(\gamma)$ and the

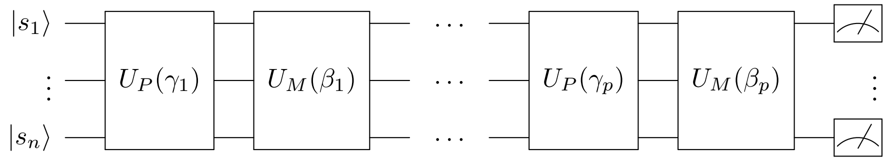

Fig. 1. QAOA circuit

mixer $U_M(\beta)$, where γ and β are real parameters. The phase separator can be obtained from the exponentiation of the phase hamiltonian, $U_P(\gamma) = e^{-i\gamma C}$, and the mixer may be any unitary operator that preserves feasible space and provides transitions between all pairs of states corresponding to feasible points [18]. Both operators are unitary and serve specific roles, the phase separator adds a relative phase to a state for each constraint that it satisfies, and the mixer transfers probability amplitude between different basis states encoding problem solutions.

Through the application of such a circuit to a suitably simple initial state $|s\rangle$ an ansatz $|\gamma, \beta\rangle$ is generated.

$$|\gamma, \beta\rangle = U_M(\beta_p)U_P(\gamma_p) \cdots U_M(\beta_1)U_P(\gamma_1)\,|s\rangle \,. \tag{6}$$

The ansatz is then measured on the computational basis, which will make the system collapse to a state $|z\rangle$ where z is a solution to the objective function f. This process is repeated enough times to obtain a distribution of states $|z\rangle$, where the average of the distribution will be the expected value $\langle\gamma, \beta|\, C\, |\gamma, \beta\rangle$ of the phase hamiltonian C. The largest value obtained $C(z')$ and its eigenvector $|z'\rangle$ are then selected to be the output of the function.

The classical part of QAOA is an optimization process over the parameters γ and β that maximize or minimize the expected value $\langle\gamma, \beta|\, C\, |\gamma, \beta\rangle$. Since the expected value is direct related to the average solution, this improves the solution quality when the system is measured. This process can be considered as a $2p$-parameter function optimization problem where the function evaluation involves a quantum system.

III. Problem Representation

We address the following school timetabling search problem. Assume that $T = \{t_1, t_2, \ldots, t_i\}$ is a set of teachers, $C = \{c_1, c_2, \ldots, c_j\}$ is a set of classes, and $R = \{r_1, r_2, \ldots, r_k\}$ is a set of rooms available in the school. Each room has a list of features from the set $F = \{f_1, f_2, \ldots, f_l\}$ that represents the available resources in the room, e.g. blackboards, available computers, etc. Each teacher also has a list of the needed resources for his/her subject. We define a lecture as a tuple composed by a teacher, a class and a room. A lecture can only be made if the room has the set of features needed by the teacher to present his/her subject.

Consider there are k time periods. Usually the time periods are evenly distributed among 5 days, composing a working week. Our scheduling must follow the given restrictions for each time period:

1) Two concurrent lectures cannot be allocated at the same room;
2) A teacher cannot give two lectures at the same time;
3) A class cannot attend two different lectures at the same time.

These restrictions are the hard-constraints of our problem, because if one of them is not met then our scheduling is flawed. Furthermore, our scheduling has also to meet the following constraints:

1) Students should not have lectures at the last period of the day;
2) Students should not have only one lecture in a day;
3) Students should not have more than two consecutive lectures a day.

These restrictions are called soft-constraints, because they can improve our solution but if any of them is not met the scheduling is still valid.

IV. Method

We propose to solve the school timetabling problem as formalised in the previous Section applying a Two-stage optimization using the QAOA quantum circuit (presented in Section II-B). A Two-stage optimization algorithm attempts to maximize the soft-constraints only after a feasible solution, i.e. a solution that follows all hard-constraints, has been reached. This method differs from an One-stage optimization algorithm, where both hard and soft constraints are considered simultaneously during the optimization process [25].

Our Two-stage optimization uses the QAOA circuit described by Hadfield [24] for solving the Minimum Graph Coloring problem to address the hard constraints and later addressing the soft constraints of the timetabling problem using the classic optimization process of QAOA. We detail these stages in the following.

A. First Stage - Hard Constraints

We first create the lectures to be scheduled by grouping a teacher, a room, and a class. The teachers are assigned to the least busy room that matches with their subject restrictions. We hope it will reduce scheduling conflicts later in the process. The classes are then paired with the teachers according to the necessities defined by the curriculum, as presented in Algorithm 1 (line 1).

After that, we create the conflict graph $G = (N, E)$ where N is the set of all lectures and $|N| = n$, $E = \{(i, j) \mid i, j \in N, \quad i \text{ and } j \text{ share the same room or teacher or class}\}$ and $|E| = m$ (see Alg. 1 line 2). The graph summarizes all the hard-constraints of the timetabling problem.

When the graph is ready we start to prepare the QAOA algorithm (lines 4-5). There are different strategies used to

Algorithm 1: QAOA for School Timetabling Problem

Input : Set of teachers T, Set of rooms R, Set of classes C, Number of time periods k, QAOA parameter p

1 $lectures \leftarrow assignLectures(T, R, C)$
2 $G \leftarrow createConflictGraph(lectures)$
`// QAOA algorithm parameters`
3 $initialState \leftarrow colorGraph(G, k)$
4 $\beta[1..p] \leftarrow random(0, \pi)$
5 $\gamma[1..p] \leftarrow random(0, 2\pi)$
`// Classic Optimization Process over the` β `and` γ `parameter vectors`
6 **while** *finalResult not converge* **do**
7     $finalResult \leftarrow minimize(QAOACircuitEvaluation(\beta, \gamma, initialState))$
8 **return** $finalResult$

initialize the parameter vectors γ and β such as multistart optimization [26] or the use of heuristics [27]. We decided to use the random initialization in our solution to simplify our algorithm description.

We also prepare the initial state for QAOA (line 3). We color the conflict graph with at most k colors, where k is the number of available time periods for the lectures to be assigned. The initial state must be a valid coloring, otherwise the algorithm will fail. The QAOA mixer guarantees that the evolution of the answer stays in a feasible subspace so if the coloring of the initial state preserves all hard-constraints then the final solution will still preserve the constraints.

The quantum circuit as shown in Figure 1 is then prepared (line 7, calling the Algorithm 2). The QAOA circuit can be created with $O(p(k^2m+nk))$ basic quantum gates, where p is the QAOA parameter. Our implementation uses $nk+n$ qubits, this value differs from [24] because we changed the number of ancillas qubits in order to reduce the circuit depth.

Inside Algorithm 2, after the measurement (line 4), the final state will collapse to a vector $|z\rangle$ where z is a valid graph coloring. Since we guarantee that the initial state meets all hard-constraints of our problem, the result z obtained also meets all hard-constraints.

Algorithm 2: QAOA Quantum Circuit Evaluation

Input : parameter vector β, parameter vector γ, graph coloring $initialState$

1 $results \leftarrow []$
2 $QAOACircuit \leftarrow constructQuantumCircuit(\beta, \gamma, initialState)$
3 **for** $j \leftarrow 1$ ***to*** 10000 **do**
4     $measurements \leftarrow (measure(QAOACircuit))$
    `// ObjectiveFunction can be seen in Equation (7)`
5     $results.append(ObjectiveFunction(measurement))$
6 $expectedValue \leftarrow results.average()$
7 **return** $expectedValue$

B. Second Stage - Soft Constraints

Our solution addresses the soft constraints using the classic optimization loop of the QAOA algorithm, shown in the Algorithm 1 (line 6-7). We use the Nelder-Mead algorithm [28] as the optimization routine to work with QAOA. The most common approach for optimizing parameters in hybrid quantum-classical algorithms is currently gradient-free black-box methods such as Nelder-Mead [29] and Bayesian methods [30]. Furthermore, in some cases analytic gradients for quantum circuits cannot be computed, and approximating gradients can be computationally expensive [26].

Usually, the phase separator encodes the function to be optimized, but we changed the objective function so that it represents our soft constraints while keeping the same phase-separator. This alteration changes significantly the inner working of QAOA. The quantum circuit normally looks for an answer that already optimizes the objective function, which in the case of Minimum Graph Coloring, is the function that counts how many colors were used in the coloring.

Even though the objective function is not encoded by the phase separation operator, the solutions found by the quantum circuit are still valid solutions. When the first stage ends, the QAOA circuit returns to the classical optimization process a graph coloring, since the timetabling problem can be reduced to the graph coloring problem the solution returned represents a valid timetable schedule.

Therefore, while the QAOA circuit searches for a solution that uses the least amount of colors, keeping the solution within feasible space, the classical optimization loop guides the parameters so that the solution fails the least amount of soft constraints. To compensate for this change, we expect that longer optimization runs will be needed.

Our new objective function that counts the number of soft constraints violations receives as input a valid coloring and is described as:

$$\sum_{c \in C}^{|C|} f_1(c) + f_2(c) + f_3(c) \tag{7}$$

where C is the set of all classes and

- $f_1(c)$ is the number of times that a class c has a lecture at the last time of the day
- $f_2(c)$ is the number of times that a class c has only one lecture in a day
- $f_3(c)$ is the number of times a class c has more than two consecutive lectures, each time weighted by the number of lectures exceeding two in sequence.

The function evaluation for our $2p$-parameter function optimization is described by Algorithm 2. After the measurement of the QAOA quantum circuit (line 4), the result of the measurement is then passed to the objective function (line 5). This process must be repeated in order to create a state distribution and the expected value to be measured (line 6).

The expected value is then returned to the classical optimization process (line 7). We seek to optimize the expected value instead of the final result because of the probabilistic nature of quantum computation [31, Chapter 1]. The expected value represents the average result returned by QAOA.

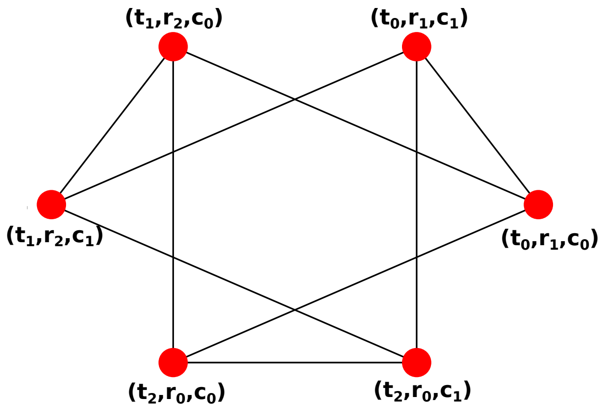

Fig. 2. Conflict graph for the school timetabling problem which summarizes its hard-constraints. Each node represents a lectures (t, r, c), where t is a teacher, r is a room and c is a class. The edges connect the lectures that cannot be assigned to the same time period.

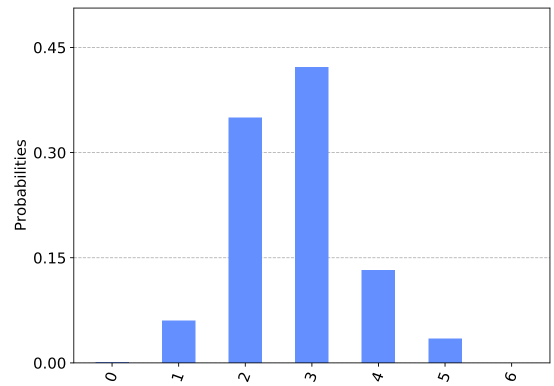

Fig. 3. Objective function distribution of the best run.

V. Results

We simulated the quantum circuits with the Ket-Bitwise-Simulator from Ket Quantum Programming Language [32]. We used an Intel Core i7-8550U with 8Gb of RAM to run our simulations. We required a total amount of 42 qubits to simulate our algorithm. The amount of qubits required exceed the limit of most quantum computers available, which makes the use of simulation necessary.

We tested our method with a minimal example. First, we defined $T = \{t_0, t_1, t_2\}$ as the set of teachers, $C = \{c_0, c_1\}$ as the set of classes, and $R = \{r_0, r_1, r_2\}$ as the set of rooms. The features available for the rooms were defined by the set $F = \{f_0, f_1\}$. We set $k = 6$ for the time periods, representing a single day of the week divided in 6 timeslots. We created a restriction that only room r_0 had the feature f_1 needed by teacher t_2.

We had a total of 6 lectures. Each teacher was assigned to an individual room and each class had to take all the subjects. The resulting conflict graph with each lecture is shown in Figure 2. The edges connect the lectures that share either the class or the pair teacher-room, since they cannot be allocated to the same time period.

We made a total of 10 runs of our algorithm for $p = 1$. Each run used the same initial state but started with a random initialization of both parameters β and γ. The independent runs served to avoid local minima in our optimization process.

For the initial state of QAOA, we assigned each node of the graph to a different color, using all k time periods. Our initial state scored a 5 in the objective function, Equation 7, as we deliberately chose it to break as much soft constraints as possible. This choice was made so we could better ascertain the effects of our method.

The number of iterations until convergence of each run varied between 2 and 39 and its results are summarized in Table I. On average, the expected value was 2.7727, with the worst value found diverging only 7% from the best, 2.92017 and 2.72969 respectively. This means that the average result obtained by QAOA scored less than 3 in the objective function, which shows an improvement from the initial 5 of our initial state.

TABLE I
Results obtained for each optimization run of QAOA.

Expected Value	Iterations
2.74826	27
2.74051	20
2.73894	8
2.74905	14
2.92017	13
2.74886	39
2.84589	2
2.75339	13
2.72969	31
2.75289	12

We calculated the distribution of the objective function of our best result, 2.72969, and plotted it in Figure 3. The histogram shows that the most common results scored 3, with 42% of probability, and 2, with 35%, in our objective function. Only few results, 3%, showed no improvement from the initial value.

From the distribution obtained in Figure 3, the result with the highest probability is listed in Table II. The final score of the result was 3, and it used a total of 4 time periods. As expected, the result respects all hard-constraints and the schedule is valid. In relation to soft constraints, both $f_1(c)$ and $f_2(c)$ were satisfied. As previously said, we consider solutions which break soft constraints, as occurred with $f_3(c)$. Our solution allows the scheduling of all lectures at class c_1 sequentially.

VI. Conclusion

This work proposes the use of the Quantum Approximate Optimization Algorithm as a heuristic to solve the School Timetabling Problem. We developed a Two-stage optimization process, where on the first stage the QAOA quantum circuit

TABLE II
FINAL TIMETABLING SCHEDULE

Time Period	Lectures allocated
0	(t_0, r_1, c_1)
1	$(t_1, r_2, c_0)\|(t_2, r_0, c_1)$
2	$(t_1, r_2, c_1)\|(t_2, r_0, c_0)$
3	–
4	(t_0, r_1, c_0)
5	–

addressed the hard constraints of the problem and on the second stage the classical optimization loop of QAOA addressed the soft constraints. This work is a first step in using QAOA as a heuristic for the timetabling problem.

Despite of the simple instance used, our results demonstrated that QAOA can be applied to the school timetabling problem. The choice of QAOA as an heuristic may become a matter of choosing the instances where quantum computation shows an advantage over classical strategies [22]. The number of qubits available on real quantum computers and the presence of noise is still a limiting factor in the widespread use of quantum computation, but once this technology improves its capacity, hybrid algorithms approaches like presented in this work will be also feasible.

As an alternative to the graph-coloring method, the QAOA could be used to solve the timetabling problem modeled as a constraint-satisfaction problem. This alternative could be easier to simulate because a n-bit string that composes the configuration space of this problem could be mapped to n qubits, while for the graph coloring the number of qubits used scale faster, since it depends on the product of the number of events and the number of time periods.

For future works, we seek to model the school timetable problem direct into the QAOAs phase-separator, so the two-stage optimization problem becomes only one-stage. This change may lead to better results and shorter optimization time.

ACKNOWLEDGMENT

This study was financed in part by the the Federal University of Santa Catarina and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

REFERENCES

[1] S. Even, A. Itai, and A. Shamir, "On the Complexity of Timetable and Multicommodity Flow Problems," *SIAM Journal on Computing*, vol. 5, no. 4, pp. 691–703, 1976.

[2] A. Hilton, "School Timetables," in *North-Holland Mathematics Studies*, nov 1981, vol. 53, no. 9, pp. 177–188.

[3] D. de Werra, "An introduction to timetabling," *European Journal of Operational Research*, vol. 19, no. 2, pp. 151–162, 1985.

[4] C. J. Colbourn and P. C. van Oorschot, "Applications of combinatorial designs in computer science," *ACM Computing Surveys (CSUR)*, vol. 21, no. 2, pp. 223–250, 1989.

[5] A. Schaerf, "A Survey of Automated Timetabling," *Artificial Intelligence Review*, vol. 13, pp. 87–127, 1999.

[6] G. N. Beligiannis, C. N. Moschopoulos, G. P. Kaperonis, and S. D. Likothanassis, "Applying evolutionary computation to the school timetabling problem: The Greek case," *Computers and Operations Research*, vol. 35, no. 4, pp. 1265–1280, 2008.

[7] R. Raghavjee and N. Pillay, "Evolving solutions to the school timetabling problem," *2009 World Congress on Nature and Biologically Inspired Computing, NABIC 2009 - Proceedings*, pp. 1524–1527, 2009.

[8] N. Pillay, "A comparative study of hyper-heuristics for solving the school timetabling problem," *ACM International Conference Proceeding Series*, pp. 278–285, 2013.

[9] S. Ribic, R. Turcinhozic, and A. Muratovic-Ribic, "Modelling constraints in school timetabling using integer linear programming," *2015 25th International Conference on Information, Communication and Automation Technologies, ICAT 2015 - Proceedings*, pp. 2–7, 2015.

[10] M. Veenstra and I. F. Vis, "School timetabling problem under disturbances," *Computers and Industrial Engineering*, vol. 95, pp. 175–186, 2016.

[11] R. Turcinhodzic, S. Ribic, and E. Zdilar, "School Timetable - suggestions for a more efficient," *ICAT 2019 - 27th International Conference on Information, Communication and Automation Technologies, Proceedings*, 2019.

[12] L. Saviniec, M. O. Santos, A. M. Costa, and L. M. Santos, "Pattern-based models and a cooperative parallel metaheuristic for high school timetabling problems," *European Journal of Operational Research*, vol. 280, no. 3, pp. 1064–1081, 2020.

[13] J. S. Tan, S. L. Goh, G. Kendall, and N. R. Sabar, "A survey of the state-of-the-art of optimisation methodologies in school timetabling problems," *Expert Systems with Applications*, vol. 165, no. September 2020, p. 113943, 2021.

[14] D. de Werra and D. Kobler, *Graph Coloring Problems*. John Wiley & Sons, Ltd, 2014, ch. 10, pp. 265–310. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119005353.ch10

[15] T. B. Cooper and J. H. Kingston, *The complexity of timetable construction problems*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1996, vol. 1153, pp. 281–295. [Online]. Available: http://dx.doi.org/10.1007/3-540-61794-9_66

[16] G. H. Fonseca, H. G. Santos, and E. G. Carrano, "Integrating matheuristics and metaheuristics for timetabling," *Computers and Operations Research*, vol. 74, pp. 108–117, 2016.

[17] T. A. Budiono and Kok Wai Wong, "A pure graph coloring constructive heuristic in timetabling," in *2012 International Conference on Computer Information Science (ICCIS)*, vol. 1, 2012, pp. 307–312.

[18] S. Hadfield, Z. Wang, B. O'Gorman, E. Rieffel, D. Venturelli, and R. Biswas, "From the quantum approximate optimization algorithm to a quantum alternating operator ansatz," *Algorithms*, vol. 12, 09 2017.

[19] Utkarsh, B. K. Behera, and P. K. Panigrahi, "Solving Vehicle Routing Problem Using Quantum Approximate Optimization Algorithm," *arXiv e-prints*, p. arXiv:2002.01351, Feb. 2020.

[20] P. Vikstâl, M. Grönkvist, M. Svensson, M. Andersson, G. Johansson, and G. Ferrini, "Applying the Quantum Approximate Optimization Algorithm to the Tail-Assignment Problem," *Physical Review Applied*, vol. 14, no. 3, p. 034009, Sep. 2020.

[21] S. Bravyi, A. Kliesch, R. Koenig, and E. Tang, "Obstacles to variational quantum optimization from symmetry protection." *Physical review letters*, vol. 125 26, p. 260505, 2020.

[22] C. Moussa, H. Calandra, and V. Dunjko, "To quantum or not to quantum: towards algorithm selection in near-term quantum optimization," *Quantum Science and Technology*, vol. 5, no. 4, p. 044009, oct 2020. [Online]. Available: https://doi.org/10.1088/2058-9565/abb8e5

[23] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," 11 2014.

[24] S. Hadfield, "Quantum algorithms for scientific computing and approximate optimization," *arXiv: Quantum Physics*, 2018.

[25] R. M. R. Lewis, "Metaheuristics for university course timetabling." Ph.D. dissertation, school: sch_comp Department: School of Computing, Faculty of Engineering, Computing and Creative Industries. [Online]. Available: http://researchrepository.napier.ac.uk/id/eprint/2392

[26] R. Shaydulin, I. Safro, and J. Larson, "Multistart methods for quantum approximate optimization," in *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, 2019, pp. 1–8.

[27] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, "Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices," *Phys.*

Rev. X, vol. 10, p. 021067, Jun 2020. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.10.021067

[28] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 01 1965.

[29] G. G. Guerreschi and A. Y. Matsuura, "Qaoa for max-cut requires hundreds of qubits for quantum speed-up," *Scientific Reports*, vol. 9, 05 2019.

[30] D. Zhu, N. M. Linke, M. Benedetti, K. A. Landsman, N. H. Nguyen, C. H. Alderete, A. Perdomo-Ortiz, N. Korda, A. Garfoot, C. Brecque, L. Egan, O. Perdomo, and C. Monroe, "Training of quantum circuits on a hybrid quantum computer," vol. 5, no. 10, 2019.

[31] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge: Cambridge University Press, 2010.

[32] E. R. R. Chagas, "Ket quantum programming language," https://gitlab.com/quantum-ket/ket, 2020.