

Constrained Quantum Optimization for Resource Distribution Management

Sara El Gaily¹, Sándor Imre²

Department of Networked Systems and Services
Budapest University of Technology and Economics, Budapest, Hungary

Abstract—The cloud computing field suffers from the heavy processing caused by the exponentially increasing data traffic. Therefore, optimizing the network performance and achieving a better quality of service (QoS) became a central goal. In cloud computing, the problem of energy consumption of resource distribution management system (RDMS) is presented as an optimization problem. Most of the existing classical optimization approaches, such as heuristic and metaheuristic have high computational complexity. In this work, we proposed a quantum optimization strategy that executes the tasks exponentially faster and with high accuracy named constrained quantum optimization algorithm (CQOA). We exploit the CQOA in RDMS as a toy example for pointing out the efficiency of the proposed quantum strategy in reducing energy consumption and computational complexity. Following that, we investigate the CQOA's implementation, setup, and computational complexity. Finally, we create a simulation environment to evaluate the efficiency of the suggested implemented constrained quantum strategy.

Keywords—Quantum computing; constrained quantum optimization algorithm; quantum extreme values searching algorithm; resource distribution management; cloud computing

I. INTRODUCTION

Parameter optimization and the choice of the best solution candidate play a crucial factor in gaining optimal performance in many application types in a wide range of disciplines. It is useful to mention that most of the computational problems arising from the practical optimization world are frequently mapped to searching the minimum or maximum of a goal function (or a constraint goal function).

The cloud computing industry is suffering from intensive processing due to rapidly rising data traffic. As a result, improving the network speed and attaining a higher quality of service (QoS) became a top priority. In the last decades, several well-known quantum strategies have been proposed in quantum computing [1,2] such as quantum phase estimation (QPE) which is exponentially faster than the classical ones, it computes the eigenvalue of a unitary operator, it has many useful applications, here we list some examples, such as, quantum counting algorithm for computing the number of occurrences of a query (searched item) in a certain database, Shor's algorithm for integer factorization [3], or the HHL algorithm [4] for solving a linear system of equations. Another interesting well-known searching algorithm, the so-called Grover's algorithm (quantum solution for searching an item in an unsorted database) which enables a dramatic reduction in computational complexity. The optimal classical solution takes

$O(N)$ iterations to carry out the search while Grover's strategy requires only $O(\sqrt{N})$ step [5-6].

In this work, we exploited the constrained quantum optimization algorithm (CQOA) [17]-[20] for optimizing the energy consumption of a proposed resource distribution management system (RDMS). It is important to mention that one of the motivations behind exploiting the CQOA in RDMS is that most of the constrained classical searching strategies suffer from high computational complexity for one of the following reasons,

- Most of the databases are Unsorted (unordered).
- The database may contain many local extremes (minimum or maximum) gratifying the constraint.
- The database structure is not always continuous.

The outline of this paper is as follows: Section II presents the literature review. Section III is devoted to introducing the CQOA and its computational complexity as well as showing a comparison between the CQOA and its original quantum algorithm version, the so-called quantum extreme value searching algorithm (QEVSA). Section IV deals with describing the resource distribution management model and the problem statement. Section V is concerned with implementing and configuring the proposed constrained quantum optimization approach in the desired resource distribution management. Section VI is devoted to discussing the computational complexity of the applied CQOA. Section VII presents the simulation results, and Section VIII concludes the paper.

II. RELATED WORK

It is not easy to build a new efficient quantum algorithm that outperforms a classical one. For this sake, there are few discovered quantum algorithms. Concerning the constrained extreme value searching, the most leading quantum heuristic candidate is introduced in [7], the so-called quantum approximate optimization algorithm (QAOA) which approximates hard optimization problems by converting the classical objective function into a Hamiltonian problem. Later, the QAOA was exploited for solving a constrained optimization problem, this alternative solution investigates the ground state of the Hamiltonian cost function instead of minimizing the original cost function [8]. Another extended version of the QAOA, the so-called quantum alternating operator ansatz is designed for finding the approximate solutions to optimization problems with hard constraints [9].

The only drawback is that the calculation of the derivatives of the goal function can be computationally expensive. On the other hand, several well-known classical approaches for handling the constrained optimization search have been proposed in the literature, the most applied ones namely deterministic optimization algorithms and metaheuristic ones, these strategies are often failing to find the global optimum result when the goal function is non-monotonic, non-continuous, or the database is unsorted and very large [10]-[16].

It is worth showing the motivation behind applying the CQOA in RDMS. As it is known, new technologies continue to merge in cloud computing to meet the challenges imposed by the exponentially increasing data traffic [21]-[55]. Recent works [56-58], of the cloud radio access network, have shown significant performance gains by centralizing the management of radio and processing resources in the cost of computational complexity.

Due to the significance of the issues that the cloud environment addresses, many optimization methods have been proposed lately. The majority of them strive for a high quality of service.

The authors in [21] employed a hybrid prediction model that merges both statistical and machine learning approaches to generate higher-quality prediction outcomes for cloud computing. The proposed approach was able to predict with high accuracy the necessary workload. However, these techniques suffer from high computational complexity due to complex and highly nonlinear data. The model has been exploited to predict both seasonality and random workload patterns.

The authors in [22] proposed a hybrid technique with a shuffled leapfrog algorithm and ubiquitous binary search (SLFA-UBS) to resolve these issues with an optimal assignment and better resource distribution. The method performed better in terms of optimum dynamic resource provisioning with QoS and cheap cost.

Several metaheuristic approaches have been applied to improve the quality of service of cloud computing. For example, the authors in [26] suggested a multi-objective hybrid Ant Colony Optimization (ACO) with Bacterial Foraging (ACOBF) behavior to maximize resource utilization and also minimize the Makespan. While, the authors in [36] proposed another multi-objective hybrid method that combines the two well-known strategies, particle swarm optimization (PSO), and grey wolf optimization (GWO). The experimental outcomes proved that the newly developed method reduces the total execution time and cost compared to PSO, heterogeneous earliest time first (HEFT), ant colony optimization (ACO), and round-robin (RR) algorithms.

The author in [59] considered a cloud radio access network and aimed to minimize the energy consumption of the overall system while satisfying constraint demands. Moreover, the work in [60] executed sequentially two heuristic approaches for minimizing the power consumption of task processing. In addition, the author in [61] used the Matroid algorithm to optimally solve the constrained resource allocation problem.

Another classical technique that is often used is the so-called knapsack optimization strategy which is widely investigated in the processing resource assignment in a cloud radio access network system [62-64]. However, the proposed optimization algorithms for task allocation require high computational complexity.

III. CONSTRAINED QUANTUM OPTIMIZATION ALGORITHM

Before introducing the CQOA, let's present first the QEVSA. In [65]-[66] the author built a new quantum algorithm named quantum existence testing (QET) which is a special case of quantum counting algorithm that determines the number of occurrences M of a certain item in a database consisting of N entries [67]. While the QET tests whether a given entry exists or not in a certain database, in other words, it checks the value of M , if it equals zero or not. Next, the author developed a new quantum method called the quantum extreme value searching algorithm (QEVSA) by combining the well-known classical binary searching algorithm [68] and the QET. More details are presented in [65]-[66]. The QEVSA finds the extreme (minimum or maximum) of an unconstrained goal function or unsorted database.

The computational complexity (CC) of the QEVSA depends on,

- The CC of the binary searching algorithm embedded in the QEVSA which equals $O(\log_2(T))$, where T is the maximum number of steps needed to run the logarithmic search.
- And, the CC of the QET which equals $O(\log_2^3(\sqrt{N}))$, where $N = 2^a$ is the entry size of the database, where a is the total number of the required qubits with respect to the size N of the database.

The overall number of bits n_E used in the physical implementation of the QET is strongly influenced by the quantum uncertainty and the classical accuracy of the application. In case the quantum uncertainty demand is neglected i.e. it corresponds to the idealistic phase estimation with no error, this implies that the value of n_E can be written as.

$$n_E = \underbrace{\frac{a}{2} - 1}_{c_E} \quad (1)$$

where c_E is the optimum number of qubits required for classical accuracy in order to represent the phase, in this case, the CC of the QET is $O(\log_2^3(\sqrt{N}))$. On the other hand, if the upper bound of the error probability is denoted by \check{P}_E of the quantum uncertainty _originated from the QPE_ is taken into consideration by the application, then the value of n_E is expressed as.

$$n_E = \underbrace{\frac{a}{2} - 1}_{c_E} + \underbrace{\left[\log_2(2\pi) + \log_2\left(\frac{1}{8\check{P}_E}\right) \right]}_{p_E} \quad (2)$$

where p_E is the optimum number of qubits needed to handle the quantum uncertainty problem originating from the error probability of converting the phase to a probability

amplitude. In this respect, the CC of the QET is $O\left(\log_2^3\left(2^{\frac{p_E}{2}}\sqrt{N}\right)\right)$. We assume that the CC of the QEVSA is only influenced by the classical certainty parameter. For this sake, the CC can be written as $O\left(\log_2(T)\log_2^3(\sqrt{N})\right)$.

In compliance with what has been discussed, we see that the QET plays a fundamental role in the search efficiency of the QEVSA.

In [17]-[18], we developed a new extended version of the QEVSA, the so called CQOA, where we extended the functionalities of the QET(*ref*) to a new quantum function that answers whether there exists an item in a certain region of the database at all, and satisfies the engineering constraint *C* and index relation *R* (The value of *R* may refer to minimization or maximization of the constraint goal function) is needed. This new extended version is named the constrained quantum relation testing *CQRT(ref, R, C)* where the parameter *ref* refers to the updated value which divides the database into two vertical parts, the index *R* refers to the used relation, and the constraint *C* can be equality or inequality constraint.

Also, we proved that implementing the constraint *C* and the relation *R* in the QET does not change the evaluation of QPE. For this sake, the estimated optimum number of qubits required for the classical accuracy denoted by c_R which corresponds to the CQRT function equals the c_E belonging to QET function, one obtains $n_R = n_E$. To this end, the computational complexity of the QEVSA equals the computational complexity of CQOA which equals $O\left(\log_2(T)\log_2^3(\sqrt{N})\right)$.

Note that the notation of *ref* value in the algorithm has been changed to $F_{med\ S}$. The CQOA is expressed as follows,

- 1) We start with $S = 0 : F_{min\ 1} = F_{min\ 0}, F_{max\ 1} = F_{max\ 0}$, and $\Delta F = F_{max\ 0} - F_{min\ 0}$
- 2) $S = S + 1$
- 3) $F_{med\ S} = F_{min\ S} + \left\lfloor \frac{F_{max\ S} - F_{min\ S}}{2} \right\rfloor$
- 4) $flag = CQET(F_{med\ S}, R, C)$:
 - if $flag = Yes$, then $F_{max\ S+1} = F_{med\ S}$
 - Else $F_{max\ S+1} = F_{max\ S}, F_{min\ S+1} = F_{med\ S}$
- 5) If $S < \log_2(T)$, then go to 2, else stop and $y_{opt} = F_{med\ S}$

It is important to mention that the QEVSA performs only the search in the continuous database structure because the QET cannot adjust the classical logarithmic search algorithm (binary searching algorithm) so that it is suitable for non-continuous databases.

The CQRT allows adapting the binary searching algorithm so that it is suitable for continuous database structures to non-continuous ones. To this end, the CQOA handles the search in a continuous or non-continuous database structure. The main similarities and differences between the QEVSA and CQOA are summarized in Table I.

TABLE I. THE MAIN SIMILARITIES AND DIFFERENCES BETWEEN THE QEVSA AND CQOA

	QEVSA	CQOA
Type of the goal function	Unconstraint goal function	Constraint goal function
Database type (Continuous/Non-Continuous)	Continuous	Continuous/Non-Continuous
The classical logarithmic search algorithm is combined with	QET	CQRT
Computational complexity	$O\left(\log_2(T)\log_2^3(\sqrt{N})\right)$	$O\left(\log_2(T)\log_2^3(\sqrt{N})\right)$
Database Structure (ordered/unordered)	Unordered database	Unordered database

IV. MODEL

This paper is an extension of the previous work published in [69-73], the of the current paper is reducing the energy consumption of processing resources by taking into consideration the delay constraint of the tasks. To model the general RDMS, we divided its functionalities into three main blocks. Fig. 1 represents the architecture of the proposed RDMS.

A. Multiple Task Generators

We consider multiple task generators, where each generator releases a task type to be served by computing units. We assume that the number of generators/task-types is denoted by *G*. Each generated task is composed of several subtasks selected from a set of subtask types, where the total number of different subtask types is *V*.

Each generator always produces identical tasks i.e. the same number of the total subtasks and the same number of subtask types. Moreover, each generator releases tasks according to an arrival time distribution (exponential intensity distribution, uniform intensity distribution, etc.). Note that, the memory needed to allocate the subtask type *v* is denoted by Δ_v . Furthermore, each task type has to be served within a specific delay constraint denoted by τ_g i.e. the subtasks belonging to task type *g* have the same *time constraint* τ_g i.e. all the subtasks of the task type *g* have to be served within this time constraint τ_g .

1) *Computing units*: The incoming tasks are served by computing units. The total number of computing units is denoted by *K*. Each computing unit has a maximum capacity c_k . Assuming that N_{vk} refers to the number of subtask type *v* under process on the k^{th} computing unit.

2) *Decision maker*: It controls the deployment of the incoming subtasks among computing units. The CQOA introduced in Section II will be implemented as a computational infrastructure core of the decision-maker.

B. Problem Statement

The decision-maker deploys the subtasks of the incoming task to different computing units. Assuming that the subtasks of the incoming tasks are processed sequentially. Fig. 2

Now, let's investigate the calculation of the total energy consumed by the RDMS. We consider that the initial power needed to turn on the computing unit k is denoted by P_k^{init} .

A subtask is composed of a specified number of identical memory pieces called *memory units*. The processing rate of the k^{th} computing unit is denoted by β_k and computed as follows,

$$\beta_k = \frac{\text{number of memory units}}{\text{second}} \quad (3)$$

On the other hand, the time needed to process a subtask type v on the computing unit k is Δ_v/β_k . Furthermore, the processing delay of the actual load on the k^{th} computing unit can be calculated as,

$$\tau_k^{act} = \frac{\sum_{v=1}^V N_{kv}\Delta_v}{\beta_k} \quad (4)$$

Assuming that $\tau_k^{act} \leq \tau_g$ and the processing of the subtasks is performed sequentially as illustrated in Fig. 2, the energy required to process the subtasks on computing unit k is given by formula (5), where ε_k is the energy consumption of one memory unit on computing unit k .

$$\begin{aligned} E_k^{act} &= \varepsilon_k \sum_{v=1}^V N_{kv}\Delta_v + P_k^{init}\tau_k^{act} \\ &= \left(\varepsilon_k + \frac{P_k^{init}}{\beta_k} \right) \sum_{v=1}^V N_{kv}\Delta_v, \end{aligned} \quad (5)$$

- $\varepsilon_k \sum_{v=1}^V N_{kv}\Delta_v$: The energy needed to serve the subtasks under process on the computing unit k without considering the energy required to turn on the computing unit k .
- $P_k^{init}\tau_k^{act}$: The energy needed to turn on the computing unit k in a period that is equal to τ_k^{act} (the energy of tasks is not considered).

It is straightforward to verify that the overall energy consumption of the system can be written as,

$$\begin{aligned} E^{act} &= \sum_{k=1}^K E_k^{act} \\ &= \sum_{k=1}^K \left(\varepsilon_k + \frac{P_k^{init}}{\beta_k} \Big|_{k \in H^{on}} \right) \sum_{v=1}^V N_{kv}\Delta_v. \end{aligned} \quad (6)$$

where the term $\frac{P_k^{init}}{\beta_k}$ is considered if and only if the k^{th} computing unit is switched on (The set H^{ON} of the switched-ON computing units). Our goal is to minimize the overall energy consumption E^{act} , one obtains.

$$E_{min}^{act} = \min_{S_i \in S} E^{act}(S_i) \text{ and } \forall k \in S_i: \tau_k^{act} \leq \tau_g \quad (7)$$

with S denotes the sets of different possible distributions of the subtasks of the incoming task among all computing units and S_i refers to the i^{th} distribution/deployment scenario of S , i.e. the i^{th} specific set of computing units $S_i = \{k\}$.

V. IMPLEMENTATION AND CONFIGURATION OF THE CQOA

To select the best and the optimum deployment scenario, we apply the CQOA as a minimum constrained searching algorithm (MCSA). To this end, the function F is substituted by E^{act} , while the constraint C corresponds to the delay constraint of the incoming task type i.e. τ_g , and the implemented relation R is a "minimization".

The maximum number of steps needed to run the logarithm search T in (8) depends on two parameters: the variation of the energy consumption of the system denoted by $\Delta E = E_{max}^{act} - E_{min}^{act}$ and the step size α which is the smallest distance between the energies of two different scenarios among all the possible scenarios in the database. Fig. 3 shows these parameters.

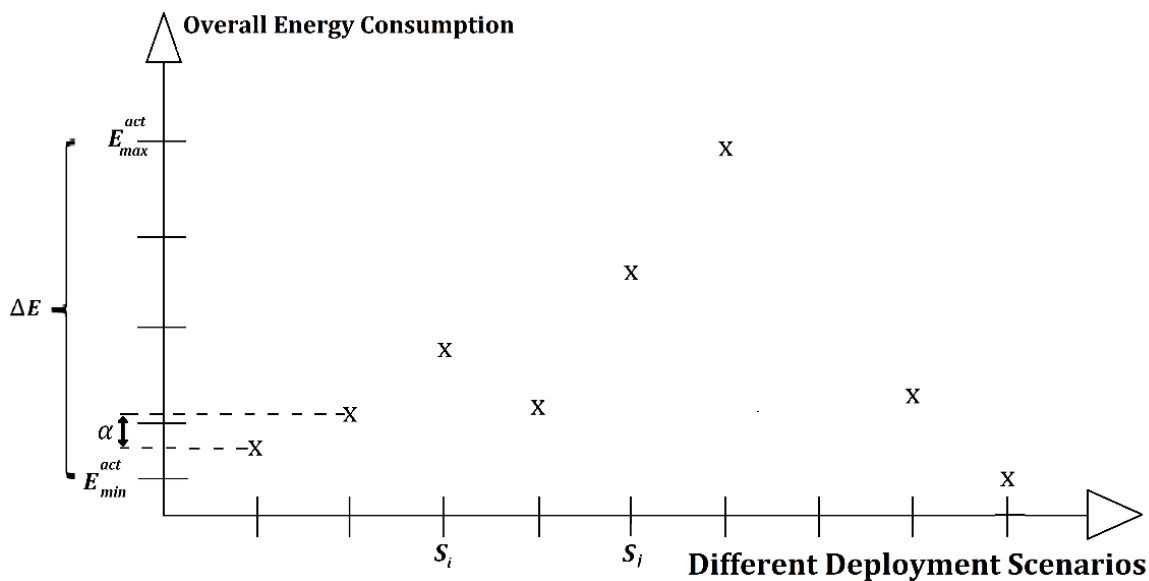


Fig. 3. The Horizontal Axis Presents all the Possible Deployment Scenarios, While the Vertical Axis Presents the Borders of the Total Energy Consumption Function (different Results), each Possible Scenario Corresponds to a Total Energy Consumption. Computing the Value of α Requires Selecting the Minimum Distance between Total Energy Consumption Functions of Two Deployment Scenarios S_i and S_j .

Since the searching region is obviously $\Delta E = E_{max}^{act} - E_{min}^{act}$, the stochastic parameter T can be expressed as follows.

$$T = \frac{E_{max}^{act} - E_{min}^{act}}{\alpha} \quad (8)$$

where E_{min}^{act} can be replaced by the energy consumption of the system without the new incoming task. We denote this energy value by \check{E}^{act} and one can observe that it does not depend on the deployment of the incoming task, this implies that.

$$T = \frac{E_{max}^{act} - \check{E}^{act}}{\alpha} \quad (9)$$

An appropriate worst-case estimation for E_{max}^{act} can be considered as.

$$E_{max}^{act} = \check{E}^{act} + \tau_g \sum_{k \in H^{OFF}} P_k^{init} + E_{max}^{inc} \quad (10)$$

where,

- E_{max}^{inc} denotes the energy consumption if the subtasks of the incoming task are deployed onto that computing units having the largest ε_k , i.e., the least energy-efficient unit.

$\tau_g \sum_{k \in H^{OFF}} P_k^{init}$ is the total energy consumed by the set H^{OFF} of the switched-OFF computing units because the switched-ON computing units are already considered in \check{E}^{act} .

On the other hand, the value of α can be written as,

$$\alpha = \min_{i \neq j} |E^{act}(S_i) - E^{act}(S_j)| \quad (11)$$

where $E^{act}(S_i)$ is the sum of the energy consumption of the system without the new task \check{E}^{act} and the energy consumption of the incoming task with assuming that it was distributed according to Z_i ,

$$E^{act}(S_i) = \check{E}^{act} + E^{inc}(Z_i) \quad (12)$$

where Z_i refers to the i^{th} set of those computing units which receive one or more subtasks of the new incoming task. It is straightforward to verify if one substitute (12) into (11), one obtains:

$$\alpha = \min_{i \neq j} |E^{inc}(Z_i) - E^{inc}(Z_j)| \quad (13)$$

Let M_{kvi} be the number of subtasks of the incoming task from subtask type v deployed onto computing unit k in case of the i^{th} deployment scenario. The formula of $E^{inc}(Z_i)$ can be expressed now by means of (14) as:

$$E^{inc}(Z_i) = \sum_{k \in Z_i} \left(\varepsilon_k + \frac{P_k^{init}}{\beta_k} \Big|_{k \in H^{on}} \right) \sum_{v=1}^V M_{kvi} \Delta_v \quad (14)$$

where this term $\frac{P_k^{init}}{\beta_k}$ is considered if and only if the k^{th} computing unit is switched-ON. To set up properly the stochastic parameter α , it is enough to investigate the non-zero solutions of $(E^{inc}(Z_i) - E^{inc}(Z_j))^2 = 0$, the solutions are located on a hyper-plane, this result is already discussed in [73].

VI. COMPUTATIONAL COMPLEXITY ANALYSIS

As previously presented, in order to minimize the constrained overall energy consumption, we have exploited the CMSA as a computational infrastructure for the RDMS. Furthermore, we have proven that the computational complexity of the implemented CQOA is $O(\log_2(T) \log_2^3(\sqrt{N}))$, it depends on the computational complexity of the CQRT function $\log_2^3(\sqrt{N})$ and the logarithmic search of the quantum algorithm $\log_2(T)$, where N refers to the total number of possible deployment scenarios.

The computational complexity analysis is divided into two main parts,

- The value of T roughly depends on the value of the stochastic parameter α . As already investigated in [73], computing repeatedly the value of α poses a real challenge. To this end, we proposed an alternative solution that consists of setting up the T value in advance before starting the assignment operation, in this case, the computational complexity of determining the parameter α of the logarithm search will be $O(1)$. More details on the computational complexity of T are presented in [73].
- The size of the search space N . This section will be devoted to estimating the value of N which refers to the set of all possible assignment scenarios for each incoming task.

Let us assume that the new task under decision has arrived from generator g . This task contains M_{gv} identical subtasks from type v and we need to select M_{gv} pieces of computing units from the overall K where repetition is allowed, i.e. a certain computing unit can be chosen more than once. One can verify that the number of such possible different sets can be written as:

$$\binom{K + M_{gv} - 1}{M_{gv}} \quad (15)$$

Considering all the subtask types, one gets.

$$N = \prod_{v=1}^V \binom{K + M_{gv} - 1}{M_{gv}} \quad (16)$$

Now, we are ready to investigate the computational complexity of the lower and upper bounds of the size of search space N . It is easy to show that

$$\begin{aligned} \binom{K + M_{gv} - 1}{M_{gv}} &= \frac{(K + M_{gv} - 1)!}{(K - 1)! M_{gv}!} \\ &= \frac{(K - 1 + M_{gv})!}{(K - 1)! M_{gv}!} \end{aligned} \quad (17)$$

Assuming that $K \gg M_{gv}$, it is interesting to note that by using formula (17), one can verify that it can be expressed in the following manner:

$$K \frac{(K+1) \cdot (K+2) \cdot \dots \cdot (K-1+M_{gv})}{1 \cdot 2 \cdot \dots \cdot M_{gv}} = \frac{K}{M_{gv}} \prod_{i=1}^{M_{gv}-1} \left(\frac{K}{i} + 1 \right) \quad (18)$$

Now we are in a position to give the upper bound for (16). Using (18), one can confirm that

$$\binom{K + M_{gv} - 1}{M_{gv}} \leq \frac{K^{M_{gv}}}{\min_v M_{gv}} \quad (19)$$

Using (19) and (16), the upper bound of the size of the search space N can be expressed as:

$$N = \prod_{v=1}^V \binom{K + M_{gv} - 1}{M_{gv}} \leq \left(\frac{K^{M_{gv}}}{\min_v M_{gv}} \right)^V \quad (20)$$

Next, we are interested in a close lower bound for (16). It is easy to verify that

$$N \geq \left(\frac{K^{M_{gv}}}{\max_v M_{gv}!} \right)^V = \left(\frac{1}{\max_v M_{gv}!} \right)^V K^{M_g} \quad (21)$$

where, $M_g = \sum_{v=1}^W M_{gv}$.

The lower bound expressed in (21) shows that the computational complexity of finding the optimal solution within the database is polynomial in terms of the numbers of computing unit K but exponential in terms of M_{gv} , i.e. if the number of subtask type V becomes large, the computational complexity rises dramatically. For this sake, performing a constrained classical computation method to find the optimum result will be time-consuming and hard to solve. So, our proposed quantum strategy is the best candidate to handle such a task assignment optimization problem.

VII. SIMULATION

To demonstrate the efficiency of the proposed CQOA, a simulation environment has been constructed, in which a constrained randomized method was considered as a reference for comparison with the proposed constrained optimization method. In the best case, the computational complexity of the constrained randomized method is $O(const)$, and in the worst case, it is $O(N)$.

This simulation aims to compare the performance of both methods in terms of computational complexity and the overall energy consumption with respect to the delay constraint.

The simulation of the RDMS hosts three computing units, the characteristics of computing units are presented in Table II, we considered three computing units where they have an identical processing rate of 40. From a practical point of view, the RDMS contains significantly more computing units, however, to observe the trends and effects it is worthwhile investigating a small-scale model.

Also, we considered two task type generators such that tasks are generated exponentially, where one of the task types has a high-intensity distribution (the mean value of intensity distribution is smaller) compared to the other one. Furthermore, we considered two subtask type, their memory requirements are respectively $\Delta_1 = 2$ and $\Delta_2 = 4$. Table III presents the number of subtask types, total memory required, and the delay constraint of each task type. While Table IV shows the total number of tasks released for each intensity distribution. It is important to mention that the timeslot of task generation is 20 seconds.

TABLE II. THE CHARACTERISTICS OF THE COMPUTING UNITS OF RESOURCE DISTRIBUTION MANAGEMENT MODEL

	Computing Unit 1	Computing Unit 2	Computing Unit 3
c_k	40	80	120
p_k^{init}	20	25	30
ϵ_k	1	2	3

TABLE III. THE CHARACTERISTICS OF THE GENERATED TASK TYPES

	Number of Subtask type 1	Number of Subtask type 2	memory	Delay constraint
Task type 1	1	3	14	1.4
Task type 2	3	1	10	1.2

TABLE IV. THE NUMBER OF TASKS RELEASED FOR EACH INTENSITY DISTRIBUTION

	Mean = 0.4	Mean = 0.3	Mean = 0.2	Mean = 0.1
Number of tasks	39	57	91	203

It is interesting to note that the decision-maker checks first the capacity constraint i.e. if there is free space in the system for allocating the new task. Then, it checks the delay constraint. Also, it is very important to mention that we did not deal with the case where the system is overloaded, i.e. we did not investigate the queueing behavior of the system.

For each two task types that have different intensity distributions, we repeat the simulation 10 times, then we calculate the average overall energy consumed for each algorithm.

Fig. 4 compares the total energy consumed by the randomized and optimized methods for different exponential intensity distributions. It can be seen that for every experiment, the constrained optimized strategy consumes less energy than the constrained randomized one. Additionally, one can notice that when the intensity distribution of arrival tasks increases i.e. when the mean value of exponential distribution becomes smaller, the optimized strategy keeps consuming lower than the randomized one.

Fig. 5 shows the percentages of the overall energy consumption reduction of the three aforementioned experiments. For example, in the first experiment where the two task type generators have respectively 0.4 and 0.3 as means (the mean value of exponential distribution), the energy consumption of the optimized method is less than the randomized one by approximately 43.85 %.

As it is shown, in the worst case the constrained randomized algorithm uses N steps, but it cannot find the optimum deployment scenarios which correspond to the minimum total energy consumption of the RDMS. While the computational complexity of the constrained optimized strategy is $O(\log_2(T) \log_2^3(\sqrt{N}))$.

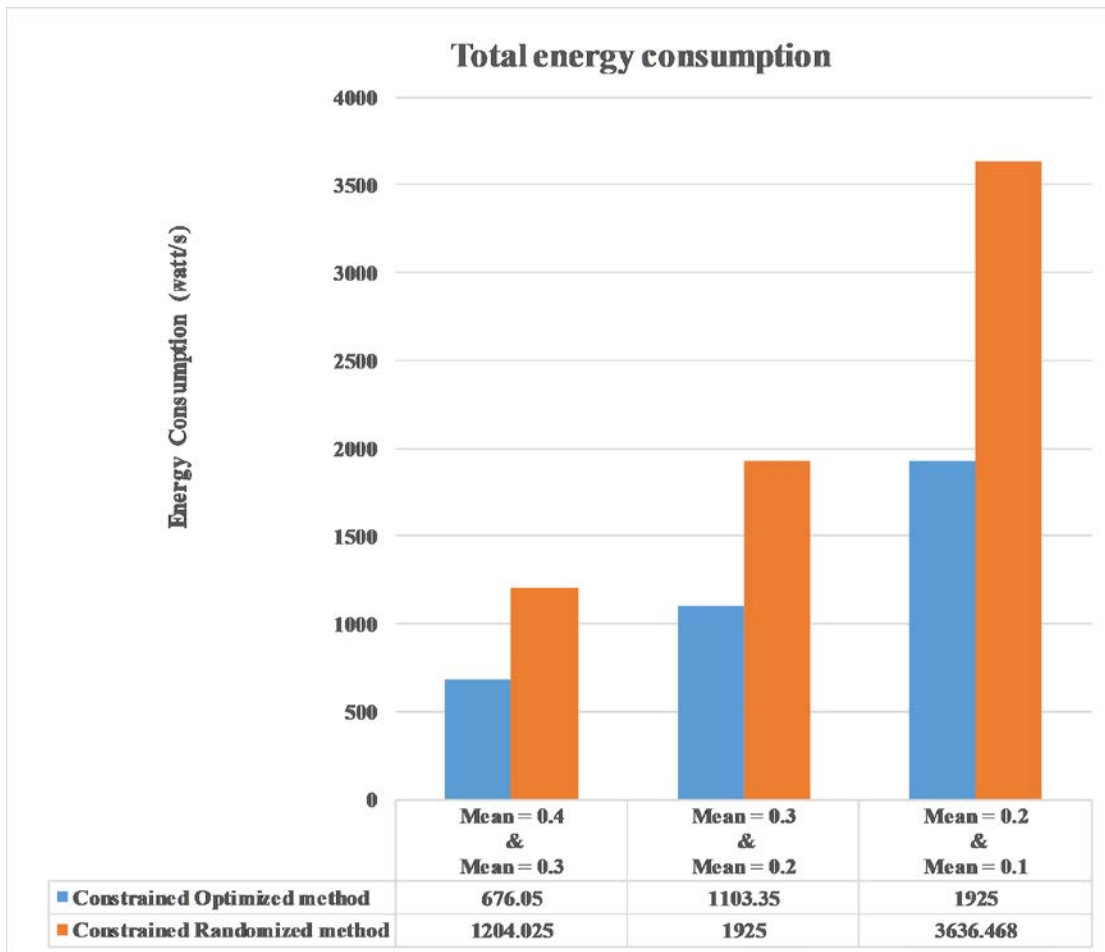


Fig. 4. Energy Consumption of the Optimized (Blue Bars) and the Randomized (Red Bars) Strategies according to different Intensity Distributions.

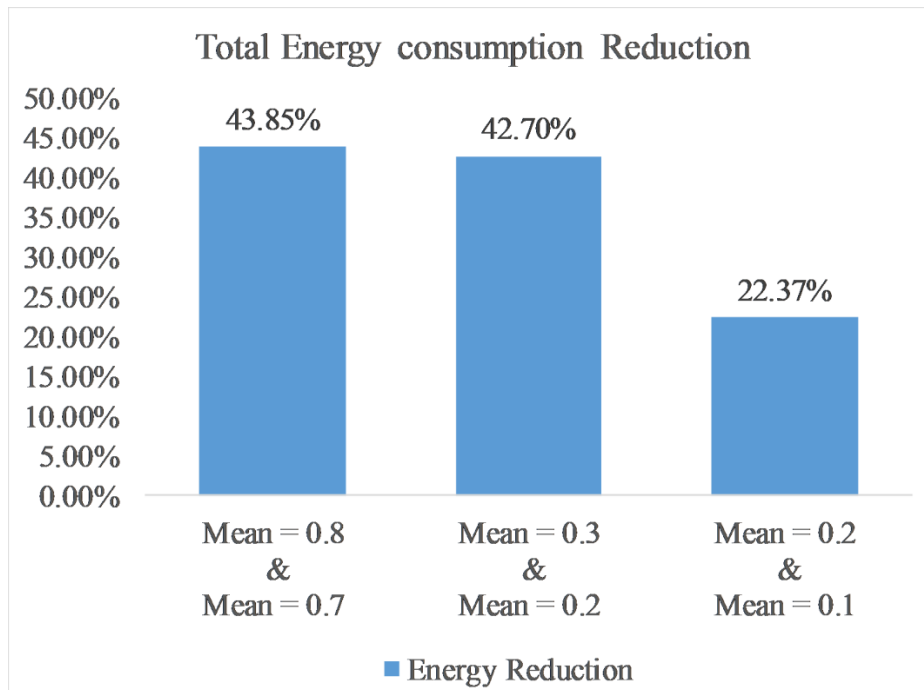


Fig. 5. Energy Consumption Reduction.

To sum up, we see that whatever the distribution intensity is, the constrained optimized approach offers a significant reduction in terms of energy consumption and computational complexity.

VIII. CONCLUSION

The CQOA finds the extreme optimum value for a constraint goal function or unsorted database with respect to certain constraints. It reduces significantly the costs connected to the application such as computational complexity and time, as well as, provides high accuracy and speed. We exploited the CQOA to minimize the constraint goal function (The total energy consumption) of the RDMS. We derived a simplified form of the constraint goal function, and we investigated the implementation and the configuration of the proposed constrained quantum strategy. Next, we proved that the computational complexity of finding the optimal solution within the database is polynomial in terms of the numbers of the computing units but exponential in terms of the number of subtasks. Fortunately, the proposed CQOA can handle such kind of optimization problem exponentially faster. In the end, we demonstrated by a simulation environment the effectiveness of the CQOA in terms of energy consumption and computational complexity by making a comparison between the constrained randomized strategy and the constrained quantum one. In future work, we will exploit the CQOA in resource distribution management by considering queueing aspect.

ACKNOWLEDGMENT

The research was partially supported by the National Research Development and Innovation Office of Hungary (Project No.2017-1.2.1-NKP-2017-00001). The research reported in this paper was supported by the BME NC TKP2020 grant of NKFIH Hungary.

REFERENCES

- [1] S. Imre S and F. Balázs. Quantum computing and communications: An engineering approach Wiley, 2005.
- [2] M. A. Nielsen and I. L. Chuang. Quantum computation and quantum information, Cambridge University Press, 2000.
- [3] P. W. Shor. Polynomial-Time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Journal on Computing*, vol 26, 1995.
- [4] A. Harrow, A. Hassidim, and S. Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* vol 103, 2009.
- [5] L. K. Grover. Quantum Computers Can Search Arbitrarily Large Databases by a Single Query, *PhysRevLett*, vol 79, 1997.
- [6] L. K. Grover. Fast quantum mechanical algorithm for database search, *STOC '96 Proc. of the Twenty-eighth annual ACM symposium on Theory of Computing*, 1996.
- [7] E. Farhi and J. Goldstone. A Quantum approximate optimization algorithm, 2014.
- [8] S. J. Suzette, R. A. Ciaran, B. William, and D. P. Ojas. Quantum approximate optimization algorithm (QAOA) on constrained optimization problems, *21st Annual Southwest Quantum Information and Technology Workshop*, 2019.
- [9] S. Hadfield, Z. Wang, B. O'Gorman, et Al. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz, *Algorithms*, vol 12, 2019.
- [10] M. H. Lin, J. F. Tsai, and C S Yu. A Review of Deterministic Optimization Methods in Engineering and Management, *Mathematical Problems in Engineering*, 2012.
- [11] C. K. Teok, A Wibowo, and M. S. Ngadiman. Review of state of the art for metaheuristic techniques in Academic Scheduling Problems. *Artificial Intelligence Review* vol 44, 2013.
- [12] R. Li and G. H. Yang. Consensus control of a class of uncertain nonlinear multiagent systems via gradient-based algorithms, *IEEE Transactions on Cybernetics*, 2019.
- [13] F. Ding, J. Pan, A. Alsaedi, and Al. Gradient-Based iterative parameter estimation algorithms for dynamical systems from observation data, *Mathematics*, vol 7, 2019.
- [14] D. Asatryan. Gradient-based technique for image structural analysis and applications *Computer Optics*, vol 43, 2019.
- [15] Y. Wang, J. Zhou, and J. Tang. Existence of generalized augmented lagrange multipliers for constrained optimization problems, *Mathematical and Computational Applications*, vol 25, 2020.
- [16] G. Audrito, F. Damiani, M. Viroli, et Al. Optimal single-path information propagation ingredient-based algorithms, Elsevier, vol 166, 2018.
- [17] S. El Gaily and S. Imre: "Constrained Quantum Optimization Algorithm", 20th International Symposium INFOTECH-JAHORINA, 17-19 March 2021.
- [18] S. El Gaily: "Extending the functionalities of the quantum extreme value searching algorithm to a constrained quantum searching algorithm", The Sixth International Conference for Young Quantum Information Scientists (YQIS 6 or YQIS 2021), 12-16 April 2021.
- [19] S. El Gaily and S Imre: "Constrained Quantum Optimization Strategy", The Bristol Quantum Information Technologies Workshop. 26-28 April 2021.
- [20] S. El Gaily: "Quantum Computing based Efficient Optimization", CTP Quantum Information Days, 22-24 February 2021.
- [21] K. C. Anupama, B. R. Shivakumar, and R. Nagaraja. Resource Utilization Prediction in Cloud Computing using Hybrid Model, *Int. J. Adv. Comput. Sci. Appl.*, vol. 12 no 4, 2021.
- [22] M. I. Hussain et Al, Hybrid SFLA-UBS Algorithm for Optimal Resource Provisioning with Cost Management in Multi-cloud Computing, *Int. J. Adv. Comput. Sci. Appl.*, vol. 12 no 4, 2021.
- [23] A. S. Gowri, P. Shanthi Bala, and I. Z. Ramdinthara, Comprehensive Analysis of Resource Allocation and Service Placement in Fog and Cloud Computing, *Int. J. Adv. Comput. Sci. Appl.*, vol. 12 no 3, 2021.
- [24] T. Chanyour and M. O. Cherkaoui Malki, Deployment and Migration of Virtualized Services with Joint Optimization of Backhaul Bandwidth and Load Balancing in Mobile Edge-Cloud Environments, *Int. J. Adv. Comput. Sci. Appl.*, vol. 12 no 3, 2021.
- [25] A. Alghamdi, A. Alzahrani, and V. Thayanathan, Fog Network Area Management Model for Managing Fog-cloud Resources in IoT Environment, *Int. J. Adv. Comput. Sci. Appl.*, vol. 12 no 3, 2021.
- [26] F. U. Zambuk et Al. Efficient Task Scheduling in Cloud Computing using Multi-objective Hybrid Ant Colony Optimization Algorithm for Energy Efficiency, *Int. J. Adv. Comput. Sci. Appl.*, vol. 12 no 3, 2021.
- [27] Á. Marosits, Z. Tabi, Z. Kallus, P. Vaderna, I. Gódor and Z. Zimborás, "Exploring Embeddings for MIMO Channel Decoding on Quantum Annealers", *Infocommunications Journal*, Vol. 13, No 1, pp. 11-17, March 2021.
- [28] S. Almutairi, N. Alghanmi, and M. M. Monowar, Survey of Centralized and Decentralized Access Control Models in Cloud Computing, *Int. J. Adv. Comput. Sci. Appl.*, vol. 12 no 2, 2021.
- [29] V. Pradeep Kumar and K. B. Prakash, Optimize the Cost of Resources in Federated Cloud by Collaborated Resource Provisioning and Most Cost-effective Collated Providers Resource First Algorithm, *Int. J. Adv. Comput. Sci. Appl.*, vol. 12 no 1, 2021.
- [30] D. Sugumar and C. R. Bharathi, AMBA: Adaptive Monarch Butterfly Algorithm based Information of Transfer Scheduling in Cloud for Big Information Application, *Int. J. Adv. Comput. Sci. Appl.*, vol. 12 no 1, 2021.
- [31] Vishal, B. Kaur, and S. Jangra. Cuckoo-Neural Approach for Secure Execution and Energy Management in Mobile Cloud Computing, *Int. J. Adv. Comput. Sci. Appl.*, vol. 12 no 1, 2021.
- [32] P. Venuthurumilli and S. Mandapati. Hybrid Invasive Weed Optimization with Tabu Search Algorithm for an Energy and Deadline

- Aware Scheduling in Cloud Computing, *Int. J. Adv. Comput. Sci. Appl.*, vol. 11 no 12, 2020.
- [33] K. Arai, S. Kapoor, et al., Intelligent Computing: Proceedings of the 2020 Computing Conference, Volume 1 (Advances in Intelligent Systems and Computing Book 1228).
- [34] N. Razzaq, et al., Cloud of Things (CoT) based Parking Prediction, *Int. J. Adv. Comput. Sci. Appl.*, vol. 11 no 8, 2020.
- [35] A. Z. Abualkishik, A A Alwan, and Y Gulzar. Disaster Recovery in Cloud Computing Systems: An Overview, *Int. J. Adv. Comput. Sci. Appl.*, vol. 11 no 9, 2020.
- [36] N. Arora and R. K. Banyal. HPSOGWO: A Hybrid Algorithm for Scientific Workflow Scheduling in Cloud Computing, *Int. J. Adv. Comput. Sci. Appl.*, vol. 11 no 10, 2020.
- [37] M. N. Adnan, et al., Comprehensive Interaction Model for Cloud Management, *Int. J. Adv. Comput. Sci. Appl.*, vol. 11 no 8, 2020.
- [38] H. Abusaimeh, Virtual Machine Escape in Cloud Computing Services, *Int. J. Adv. Comput. Sci. Appl.*, vol. 11 no 7, 2020.
- [39] H. Abusaimeh. Distributed Denial of Service Attacks in Cloud Computing, *Int. J. Adv. Comput. Sci. Appl.*, vol. 11 no 6, 2020.
- [40] K. Qazi. Modeling Real-World Load Patterns for Benchmarking in Clouds and Clusters, *Int. J. Adv. Comput. Sci. Appl.*, vol. 11 no 6, 2020.
- [41] A. Sheikh, M. Munro, and D Budgen. A Service Scheduling Security Model for a Cloud Environment, *Int. J. Adv. Comput. Sci. Appl.*, vol. 11 no 5, 2020.
- [42] M. Humayun. Role of Emerging IoT Big Data and Cloud Computing for Real Time Application, *Int. J. Adv. Comput. Sci. Appl.*, vol. 11 no 4, 2020.
- [43] H. Shah et al. Enhancing the Quality of Service of Cloud Computing in Big Data using Virtual Private Network and Firewall in Dense Mode, *Int. J. Adv. Comput. Sci. Appl.*, vol. 11 no 3, 2020.
- [44] K. Arai, Adjacency Effects of Layered Clouds by Means of Monte Carlo Ray Tracing, *Int. J. Adv. Comput. Sci. Appl.*, vol. 11 no 1, 2020.
- [45] M. Kadhum, S. Manaseer, and A. L. A. Dalhoum. Cloud-Edge Network Data Processing based on User Requirements using Modify MapReduce Algorithm and Machine Learning Techniques, *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no 12, 2019.
- [46] A. Sheikh, M. Munro, and D. Budgen. Evaluating a Cloud Service using Scheduling Security Model (SSM), *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no 10, 2019.
- [47] B. Nayak and S. K. Padhi. Mapping of Independent Tasks in the Cloud Computing Environment, *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no 8, 2019.
- [48] M. R. Belgaum et al. A Behavioral Study of Task Scheduling Algorithms in Cloud Computing, *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no 7, 2019.
- [49] B. H. Malik et al. Efficient Energy Utilization in Cloud Fog Environment, *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no 4, 2019.
- [50] S. A. Mjlae, Z. Mohamad, and W. Suryani. Impact Factors of IT Flexibility within Cloud Technology on Various Aspects of IT Effectiveness, *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no 4, 2019.
- [51] Y. H. Prasanna Raju and N. Devarakonda. Cluster based Hybrid Approach to Task Scheduling in Cloud Environment, *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no 4, 2019.
- [52] N. S. Naz et al. Efficient Load Balancing in Cloud Computing using Multi-Layered Mamdani Fuzzy Inference Expert System, *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no 3, 2019.
- [53] A. Sheikh, M. Munro, and D. Budgen. Systematic Literature Review (SLR) of Resource Scheduling and Security in Cloud Computing, *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no 4, 2019.
- [54] M. Shorfuzzaman and M. Masud. Leveraging A Multi-Objective Approach to Data Replication in Cloud Computing Environment to Support Big Data Applications, *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no 3, 2019.
- [55] I. M. Al-Jabri, M. I. Eid, and M. S. Sohail. A Group Decision-Making Method for Selecting Cloud Computing Service Model, *Int. J. Adv. Comput. Sci. Appl.*, vol. 9 no 1, 2018.
- [56] R. T. Rodoshi, T. Kim and W. Choi. Resource Management in Cloud Radio Access Network: Conventional and New Approaches, *Sensors* vol 20 no 9, 2020.
- [57] Y. Sun and M. Peng. Recent advances of heterogenous radio access networks *Journal of Mobile Multimedia Journal* vol 14 no 4, 2018.
- [58] M. F. Hossain, A. U. Mahin, T. Debnath, et Al. Recent research in cloud radio access network (C-RAN) for 5G cellular systems *Journal of Network and Computer Applications* vol 139, 2019.
- [59] Y. Liao, L. Song, Y. Li, and Y. A. Zhang. Radio resource management for Cloud-RAN networks with computing capability constraints. *Proceedings of the 2016 IEEE International Conference on Communications (ICC)* 2016.
- [60] K. Boulos, M. El Helou, K. Khawam, et Al. RRH clustering in cloud radio access networks with re-association consideration. *Proceedings of the 2018 IEEE Wireless Communications and Networking Conference (WCNC)* 2018.
- [61] N. Mharsi and M. Hadji . Joint Optimization of Communication Latency and Resource Allocation in Cloud Radio. *Access Networks 2018 International Conference on Smart Communications in Network Technologies (SaCoNeT)* 2018.
- [62] A. Li, Y. Sun, X. Xu, and C. Yuan. An energy-effective network deployment scheme for 5G Cloud Radio Access Networks. *IEEE Conference on Computer Communications Workshops*, 2016.
- [63] X. Xu, J. Liu, W Chen, Y Hou, et Al. Storage and computing resource enabled joint virtual resource allocation with QoS guarantee in mobile networks. *Science China Information Sciences*, vol 60 no 4, 2017.
- [64] L. Pu, L. Jiao, X. Chen, et Al. Online Resource Allocation, Content Placement and Request Routing for Cost-Efficient Edge Caching in Cloud Radio Access Networks, *IEEE Journal on Selected Areas in Communications* vol 36 no 8, 2018.
- [65] S. Imre. Extreme value searching in unsorted databases based on quantum computing, *Int. J. Quantum Inf*, vol 3 no 1. 2005.
- [66] S. Imre. Quantum Existence Testing and Its Application for Finding Extreme Values in Unsorted Databases. *IEEE Transactions on Computers*, 2007.
- [67] G. Brassard, P. Høyer, and A. Tapp. Quantum counting 25th Intl. Colloquium on Automata, Languages, and Programming (ICALP) 1998.
- [68] D. E. Knuth. Sorting and searching in the art of computer programming, 1998.
- [69] S. El Gaily and S. Imre. Evaluation of Resource Optimization based on Quantum Search. *Hungarian Journal of industry and chemistry*. 2019.
- [70] S. El Gaily and S. Imre: Quantum Optimization in Large Resource Management Systems - IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 2-5 July 2019, Cannes, France.
- [71] S. El Gaily and S. Imre: Derivation of Parameters of Quantum optimization in Resource Distribution Management - 42nd International Conference on Telecommunications and Signal Processing (TSP), 1-3 July 2019.
- [72] S. El Gaily and S. Imre: "Quantum Resource Distribution Management in Multi-task environment", 14th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS).
- [73] S. El Gaily and S. Imre. Quantum Optimization of Resource Distribution Management for Multi-Task, Multi-Subtasks. *Infocommunications Journal*, vol 11, no 4, 2019.