BILINGUAL PUBLISHING CO.
Pioneer of Global Academics Since 1984

Artificial Intelligence Advances
https://ojs.bilpublishing.com/index.php/aia

ARTICLE

Quantum Fast Algorithm Computational Intelligence PT I: SW / HW Smart Toolkit

Ulyanov S.V.*

State University "Dubna", Universitetskaya Str.19, Dubna, Moscow Region, 141980, Russia

ARTICLE INFO

Article history
Received: 12 March 2019
Accepted: 18 April 2019
Published Online: 30 April 2019

Keywords:
Quantum algorithm gate
Superposition
Entanglement
Interference
Quantum simulator

ABSTRACT

A new approach to a circuit implementation design of quantum algorithm gates for quantum massive parallel fast computing implementation is presented. The main attention is focused on the development of design method of fast quantum algorithm operators as superposition, entanglement and interference which are in general time-consuming operations due to the number of products that have to be performed. SW & HW support sophisticated smart toolkit of supercomputing accelerator of quantum algorithm simulation is described. The method for performing Grover's interference without product operations as Benchmark introduced. The background of developed information technology is the "Quantum / Soft Computing Optimizer" (QSCOptKBTM) software based on soft and quantum computational intelligence toolkit. Quantum genetic and quantum fuzzy inference algorithm gate design considered. The quantum information technology of imperfect knowledge base self-organization design of fuzzy robust controllers for the guaranteed achievement of intelligent autonomous robot the control goal in unpredicted control situations is described.

1. Introduction: Role of Quantum Synergetic Effects in AI and Intelligent Control Models

R. Feynman and Yu. Manin, independently, suggested and correctly shown that quantum computing can be effectively applied for simulation and searching of solutions of classically intractable quantum systems problems using quantum programmable computer (as physical devices). Recent research shows successful engineering application of end-to-end quantum computing information technologies (as quantum sophisticated algorithms and quantum programming) in searching of solutions of algorithmic unsolved problems in classical dynamic intelligent control systems, artificial intelligence, intelligent cognitive robotics etc.

Concrete developments are the cognitive "man-robot" interactions in collective multi-agent systems, "brain-computer-device" interface of autism children supporting with robots for service use, and so on. These applications are examples successful result applications of efficient classical simulation of quantum control algorithms in the algorithmic unsolved problems of classical control systems robustness in unpredicted control situations.

Related works. Many interesting results are published as fundamentals and applications of quantum / classical hybrid approach to design of different smart classical or quantum dynamic systems. For example, an error mitigation technique and classical post-processing can be con-

**Corresponding Author:*
Ulyanov S.V.,
State University "Dubna", Universitetskaya Str.19, Dubna, Moscow Region, 141980, Russia;
Email: ulyanovsv@mail.ru

DOI: https://doi.org/10.30564/aia.v1i1.619

veniently applied, thus offering a hybrid quantum-classical algorithm for currently available noisy quantum processors [1] or Quantum Triple Annealing Minimization (QTAM) algorithm utilizes the framework of simulated annealing, which is a stochastic point-to-point search method: The quantum gates that act on the quantum states formulate a quantum circuit with a given circuit height and depth [2]. A new local fixed-point iteration plus global sequence acceleration optimization algorithm for general variational quantum circuit algorithms in [3] is described. The basic requirements for universal quantum computing have all been demonstrated with ions and quantum algorithms using few-ion-qubit systems have been implemented [4]. Quantum computing is finding a vital application in providing speed-ups for machine learning problems, critical in "big data" world. Machine learning already permeates many cutting-edge technologies, and may become instrumental in advanced quantum technologies. Aside from quantum speed-up in data analysis, or classical machine learning optimization used in quantum experiments, quantum enhancements have also been (theoretically) demonstrated for interactive learning tasks, highlighting the potential of quantum-enhanced learning agents [5]. In [6] the system PennyLane as a Python 3 software framework for optimization and machine learning of quantum and hybrid quantum / classical computations is introduced. A plugin system makes the framework compatible with any gate-based quantum simulator or hardware and provided plugins for Strawberry Fields, Rigetti Forest, Qiskit, and ProjectQ, allowing PennyLane optimizations to be run on publicly accessible quantum devices provided by Rigetti and IBM Q. On the classical front, PennyLane interfaces with accelerated machine learning libraries such as TensorFlow, PyTorch, and auto grad. PennyLane can be used for the optimization of variational quantum eigensolvers, quantum approximate optimization, quantum machine learning models, and many other applications. The first industry-based and societal relevant applications will be as a quantum accelerator. It is based on the idea that any end-application contains multiple parts and the properties of these parts are better executed by a particular accelerator which can be either an FPGA, a GPU or a TPU. The quantum accelerator added as an additional coprocessor. The formal definition of an accelerator is indeed a co-processor linked to the central processor and that executes much faster certain parts of the overall application [7]. Limited quantum memory is one of the most important constraints for near-term quantum devices. Understanding whether a small quantum computer can simulate a larger quantum system, or execute an algorithm requiring more qubits than available, is both of theoretical and practical importance and in [8] is discussed. One prominent platform for constructing a multi-qubit quantum processor involves superconducting qubits, in which information is stored in the quantum degrees of freedom of nanofabricated, anharmonic oscillators constructed from superconducting circuit elements. The requirements imposed by larger quantum processors have shifted of mindset within the community, from solely scientific discovery to the development of new, foundational engineering abstractions associated with the design, control, and readout of multi-qubit quantum systems. The result is the emergence of a new discipline termed *quantum engineering*, which serves to bridge the basic sciences, mathematics, and computer science with fields generally associated with traditional engineering [9, 10].

Moreover, new synergetic effects defined and extracted from the measurement of quantum information (that hidden in classical control states of traditional controllers with time-dependent coefficient gain schedule) are the information resource for the increasing of the control system robustness and guarantee the achievement of control goal in hazard situations. The background of this synergetic effect is the creation of new knowledge from experimental response signals of imperfect knowledge bases on unpredicted situations using quantum algorithm of knowledge self-organization as quantum fuzzy inference. The background of developed information technology is the "Quantum / Soft Computing Optimizer" (QSCOptKB TM) software based on soft and quantum computational intelligence toolkit.

Algorithmic constraints on mathematical models of data processing in classical form of computing (based on Church-Turing thesis and using background of classical physics laws) are dramatically differs from physical constraints on resources limitation in data information processing models that based on quantum mechanical models such as information transmission, information bounds on the extraction of knowledge, amount of quantum accessible experimental information, quantum Kolmogorov's complexity, speed-up quantum limit of data processing, quantum channel capacity etc. Meaning exploring of the Landauer's thesis as "*Information is physical*" has prepared as result the background for changing, clarification and expanding the Church-Turing thesis, and introduce the R&D idea of quantum computing exploring and quantum computer development for successful solving many classically algorithmic unsolved (intractable in classical mean) problems.

The classification of quantum algorithms is demonstrated on Fig. 1.

 DOI: https://doi.org/10.30564/aia.v1i1.619

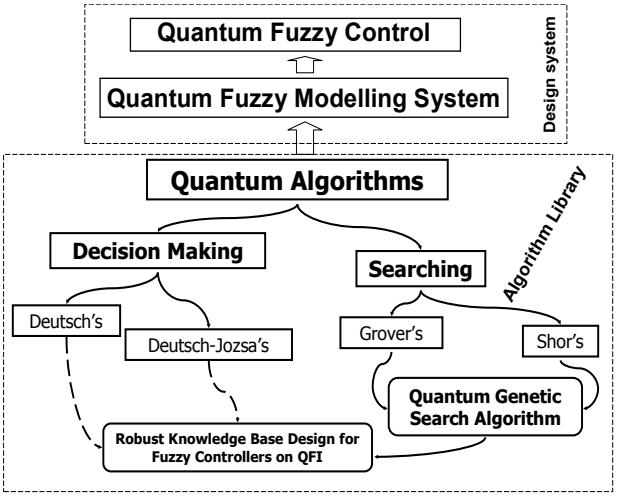


Figure 1. Classification of Quantum Algorithms and Interrelations with Quantum Fuzzy Control

Quantum algorithms are in general random: decision making quantum algorithms of Deutch-Jozsa and quantum search algorithms (QSA) of Shor and Grover are examples of successful applications of quantum effects and constraints from introduction new classes computational basis quantum operators as superposition, entanglement and interference that are absent in classical computational models. These effects given the possibility to introduce new types of computation as quantum parallel massive computing using superposition operator, operator of entanglement (super-correlation or quantum oracle) created the possibility of "good" (in general unknown) solution search and operator of quantum interference help extract searching "good" solutions with maximal amplitude probability . All of these operators are reversible, classical irreversible operator of measurement (as example, coin) extract the result of quantum algorithm computing. Note, that quantum effects that described above absent in classical models of computation and demonstrated the effectiveness of quantum constraints in classical models of computations.

Figure 2 demonstrate the computing analogy between soft and quantum algorithms and its operators that are used in quantum soft computing information technology.

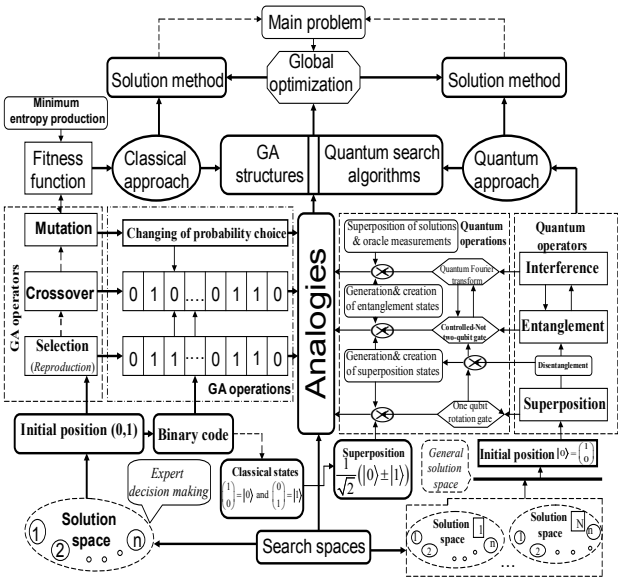


Figure 2. Interrelations between Soft and Quantum Operators in Genetic and Quantum Algorithms

From quantum programming a quantum computer point view there no exist currently the general methodology of quantum computing and simulation of dynamic systems but it was developed many proposals of quantum simulators (see, for example, the large list of quantum simulators available on [https://quantiki.org/wiki/list-qc-simulators]).

Remark. The purpose of this article is concerned with the problem of discovering new QAs. Same as D-Wave, processor supercomputing processes in a quantum computer can be described as a synergetic union of hybrid quantum / classical HW, and quantum SW with quantum soft support of quantum programming.

Remark. To understand more clearly the fundamental capabilities and limitations of quantum computation we are to discover efficient QAs for interesting engineering problems as intelligent cognitive control systems.

One the most important open problem in computer science is to estimate the possibility of quantum speed-up for the search of computational problems solution.

Oracular, or black-box, problems are the first examples of problems that can be solved faster with a quantum computer than with a classical computer. The computer in the black box model is given access to oracle (or a black box) that can be queried to acquire information about the problem. To find the solution to the problem using as few queries to the oracle as possible is the computation goal [11-13].

1.1 Goal and Problem Solving

This article consider the design possibility a family of quantum decision-making and search algorithms (QA's)

DOI: https://doi.org/10.30564/aia.v1i1.619

(see, Fig. 1) that it is the background of quantum computational intelligence for solving the problems of Big & Mining data, deep quantum machine learning (based on quantum neural network), global optimization in intelligent quantum control (using quantum genetic algorithms) etc. (see, in details Pt II).

1.2 Method of Solution and Smart Toolkit

The presented method and relative hardware implements matrix and algorithmic forms of quantum operators that are used in a QA (entanglement or oracle operators, and interference operator as in second and third steps of QA implementation) that increasing computational speed-up with respect to the corresponding SW realization of a traditional and a new QSA. A high level structure of a generic entanglement block that uses logic gates as analogy elements is described. Method for performing Grover interference without products is introduced [14, 15]. QUANTUM ALGORITHM ACCELATOR COMPUTING: SW / HW SUPPORT

A. General Structure of Quantum Algorithm

The problem solved by a QA can be stated in the symbolic form:

Input	*A function f:* $\{0,1\}^n \rightarrow \{0,1\}^m$
Problem	*Find a certain property of function* f

A given function f is the map of one logical state into another and QA estimate qualitative properties of function f.

General description of QA on Fig. 3 is demonstrated (physically the type of operator U_F describes the qualitative properties of the function f).

Figure 4 shows the steps of QA that includes almost of described qualitative peculiarities of function f and physical interpretation of applied quantum operators.

In the scheme diagram of Fig. 5 the structure of a QA is outlined.

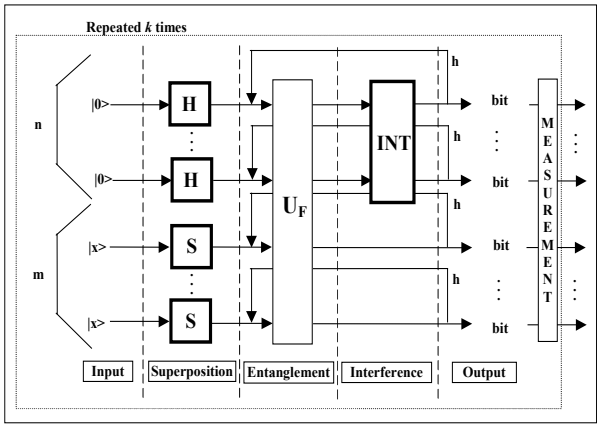


Figure 3. General Description of QAG

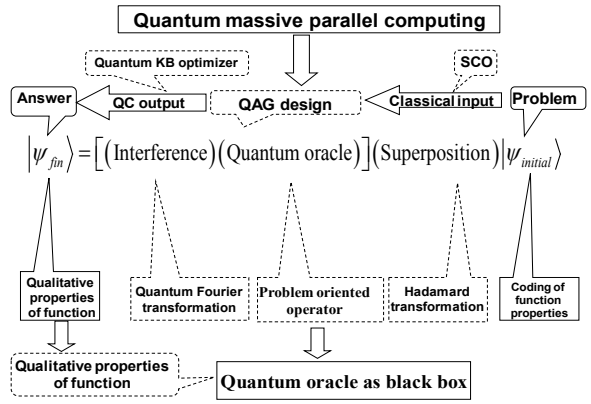


Figure 4. General Structure of QA

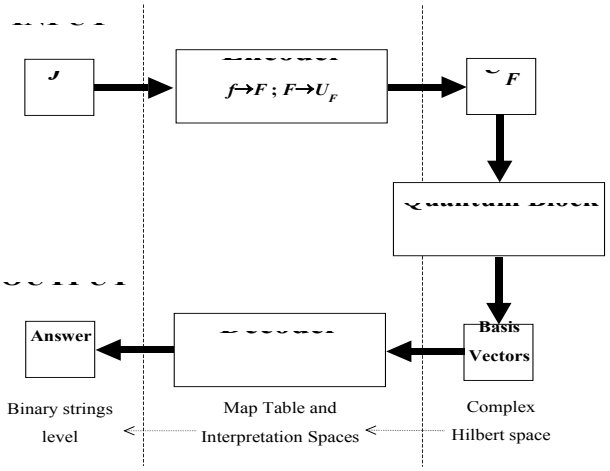


Figure 5. Scheme Diagram of QA - structure

As above mentioned QA estimates (without numerical computing) the qualitative properties of the function f. Thus with QAs we can study qualitative properties of function f without quantitative estimation of function values.

For example, Fig. 6 represents the general approach to Grover' QAG design.

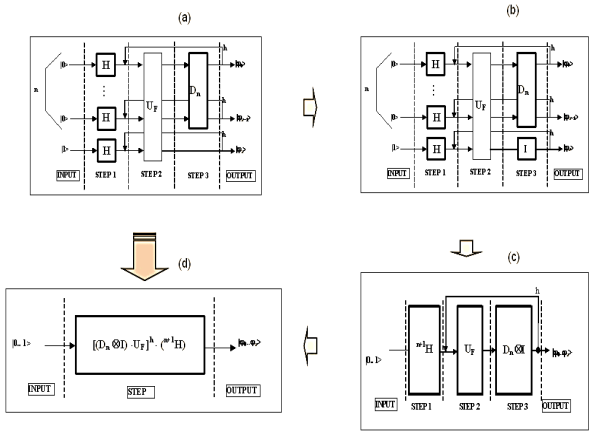


Figure 6. Circuit and Quantum Gate Representation of Grover's QSA

As a termination condition criterion minimum-entropy based method is adopted [13].

The structure of a QAG in Fig. 3 in general form defined as following:

$$QAG = \left[\left(Int \otimes {}^{n}I \right) \cdot U_F \right]^{h+1} \cdot \left[{}^{n}H \otimes {}^{m}S \right] \quad (1)$$

Where I is the identity operator; S is equal to I or H and dependent on the problem description.

Fast algorithms design to simulate most of known QAs on classical computers [15-17] and computational intelligence toolkit is following: 1) *Matrix based approach*; 2) *Model representations* of quantum operators in fast QAs; 3) *Algorithmic based approach*, when matrix elements are calculated on "demand"; 4) *Problem-oriented approach*, where we succeeded to run Grover's algorithm with up to 64 and more qubits with Shannon entropy calculation (up to 1024 without termination condition); 5) *Quantum algorithms with reduced number of operators* (entanglement-free QA, and so on).

Remark. In this article we describe briefly main blocks [13-17] in Fig. 6: i) unified operators; ii) problem-oriented operators; iii) Benchmarks of QA simulation on classical computers; and iv) quantum control algorithms based on quantum fuzzy inference (QFI) and quantum genetic algorithm (QGA) as new types of QSA (see, more in details Part II of this article).

Let us consider *matrix based* and *problem-oriented approaches* to simulate most of known QAs on classical computers and small quantum computer.

I. Quantum operator's description: SW&HW smart toolkit support

We consider from simulation viewpoint the structure of quantum operators as superposition, entanglement and interference [14,16,18,19,23-26] in matrix based approach.

Superposition operators of QA's.

The superposition operator consists in general form of the combination of the tensor products Hadamard H operators with identity operator I:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The superposition operator of most QAs can be expressed (see Fig. 3 and Eq. (1)) as:

$$Sp = \left(\bigotimes_{i=1}^{n} H \right) \otimes \left(\bigotimes_{i=1}^{m} S \right),$$

Where n and m are the numbers of inputs and of outputs respectively. Numbers of outputs m as well as structures of corresponding superposition and interference operators in [12, 13] for different QAs presented.

Elements of the Walsh-Hadamard operator could be obtained as following:

$$\left[{}^{n}H \right]_{i,j} = \frac{(-1)^{i*j}}{2^{n/2}} = \frac{1}{2^{n/2}} \begin{cases} 1, & \text{if } i*j \text{ is even} \\ -1, & \text{if } i*j \text{ is odd} \end{cases} \quad (2)$$

Where $i = 0,1,...,2^n$, $j = 0,1,...,2^n$. Its elements could be obtained by the simple replication according to the rule presented in Eq. (2).

Interference operators of main QA's

Interference operators for Grover's algorithm [18, 19] written as a block matrix:

$$\left[Int^{Grover} \right]_{i,j} = D_n \otimes I = \left(\frac{1}{2^{n/2}} - {}^{n}I \right) \otimes I =$$
$$\left(-1 + \frac{1}{2^{n/2}} \right) \otimes I \Big|_{i=j}, \left(\frac{1}{2^{n/2}} \right) \otimes I \Big|_{i \neq j} = \frac{1}{2^{n/2}} \begin{cases} -I, i = j \\ I, i \neq j \end{cases}, \quad (3)$$

where $i = 0,...,2^n - 1, j = 0,...,2^n - 1$, D_n refers to diffusion operator: $\left[D_n \right]_{i,j} = \frac{(-1)^{1AND(i=j)}}{2^{n/2}}$ [4,8]. Note that with bigger number of qubits, gain coefficient will become smaller.

Entanglement operators of main QA's

Operators of entanglement in general form are the part of QA and the information about the function (being analyzed) is coded as "input-output" relation. In the general approach for coding binary functions into corresponding entanglement gates arbitrary binary function considered as: $f : \{0,1\}^n \to \{0,1\}^m$, such that $f(x_0,...,x_{n-1}) = (y_0,...,y_{m-1})$. Firstly irreversible function f transfer into reversible function F, as following: $F : \{0,1\}^{m+n} \to \{0,1\}^{m+n}$, and

$$F\left(x_0,...,x_{n-1},y_0,...,y_{m-1}\right) = (x_0,...,x_{n-1}, f(x_0,...,x_{n-1}) \oplus (y_0,...,y_{m-1})),$$

where $\oplus$ denotes addition modulo 2. This transformation create unitary quantum operator and performs the similar transformation. With reversible function F it is possible design an entanglement operator matrix according to the following rule:

$$\left[U_F \right]_{i^B, j^B} = 1 \; iff \; F(j^B) = i^B, \quad i, j \in \left[\underbrace{0,..,0}_{n+m}; \underbrace{1,..,1}_{n+m}; \right]$$

B denotes binary coding.

A diagonal block matrix of the form: $U_F = \begin{pmatrix} M_0 & & 0 \\ & \ddots & \\ 0 & & M_{2^n-1} \end{pmatrix}$ is actually resulted entanglement operator.

Each block $M_i, i = 0,...,2^n - 1$, can be obtained as following

$$M_i = \bigotimes_{k=0}^{m-1} \begin{cases} I, & \text{iff } F(i,k) = 0 \\ C, & \text{iff } F(i,k) = 1 \end{cases} \quad (4)$$

DOI: https://doi.org/10.30564/aia.v1i1.619

And consists of m tensor products of I or of C operators, where C stays for NOT operator.

Note that entanglement operator (4) is a *sparse* matrix and according to this property, the simulation of entanglement operation accelerated.

II. QA computing accelerator: SW&HW support

Figure 7 shows the structure of intelligent quantum computing accelerator.

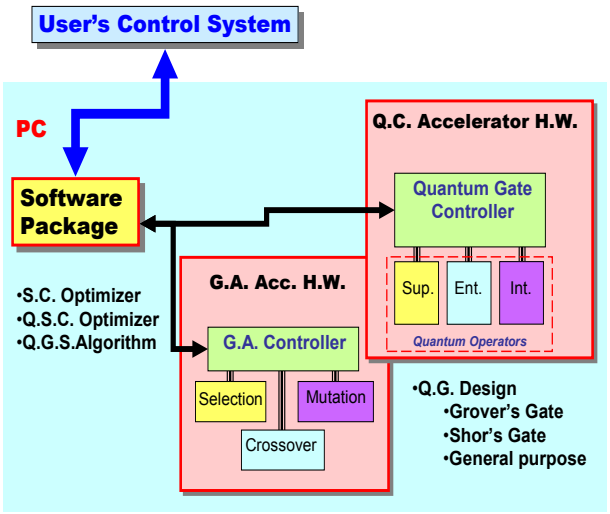


Figure 7. Intelligent Quantum Soft Computing Accelerator Structure

HW of quantum computing accelerator is based on standard silicon element background.

QA structure implementation for HW and MatLab is on Fig. 8 demonstrated (see, Fig. 23).

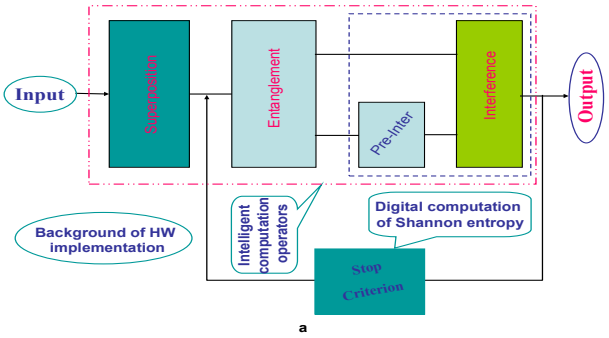


Figure 8. QA Structure Presentation for HW (a) and MatLab (b) Implementations

Different structures of QA can be realized as shown in Table 1 below.

Table 1. Quantum Gate Types for QA's Structure Design

Title	Type of Algorithm	Symbolic Form of QAG: $\left[\underbrace{(Int\otimes {}^{m}I)}_{Interference}\cdot\underbrace{U_{E}}_{Entanglement}\right]^{k+1}\cdot\left(\underbrace{{}^{n}H\otimes {}^{m}S}_{Superposition}\right)$
Deutsch-Jozsa (D. – J.)	*m=1, S=H(x=1) Int=^{n}H k=1 h=0*	$\left({}^{n}H\otimes I\right)\cdot U_{F}^{D.-J.}\cdot\left({}^{n+1}H\right)$
Simon (Sim)	*m=n,S=I (x=0)Int=^{n}H k=O(n) h=0*	$\left({}^{n}H\otimes {}^{n}I\right)\cdot U_{F}^{Sim}\cdot\left({}^{n}H\otimes {}^{n}I\right)$
Shor (Shr)	*m=n, S=I (x=0)Int=QFT_{n} k=O(Poly(n)) h=0*	$\left(QFT_{n}\otimes {}^{n}I\right)\cdot U_{F}^{Shr}\cdot\left({}^{n}H\otimes {}^{n}I\right)$
Grover (Gr)	*m=1, S=H(x=1) Int=Dn k=1, h=O($2^{n/2}$)*	$\left(D_{n}\otimes I\right)\cdot U_{F}^{Gr}\cdot\left({}^{n+1}H\right)$

1.3 Information Analysis of QA and Criterion for Solution of the QSA-termination Problem

The communication capacity gives an index of efficiency of a quantum computation [19]. The measure of Shannon information entropy is used for optimization of the termination problem of Grover's QSA. Information analysis of Grover's QSA based on of Eq. (5), gives a lower bound on necessary amount of entanglement for searching of success result and of computational time: any QSA that uses the quantum oracle calls $\{O_s\}$ as $I-2|s\rangle\langle s|$ must call the oracle at least $T\geq\left(\frac{1-P_e}{2\pi}+\frac{1}{\pi\log N}\right)\sqrt{N}$ times to achieve a probability of error P_e [20].

The information intelligent measure of QA as $\Im_T(|\psi\rangle)$ of the state $|\psi\rangle$ is [12, 21]:

$$\Im_T\left(|\psi\rangle\right)=1-\frac{S_T^{Sh}\left(|\psi\rangle\right)-S_T^{VN}\left(|\psi\rangle\right)}{|T|}. \quad (6)$$

With respect to the qubits in T and to the basis $B=\left\{|i_1\rangle\otimes\cdots\otimes|i_n\rangle\right\}$

The measure (6) is minimal (i.e., 0) when $S_T^{Sh}\left(|\psi\rangle\right)=|T|$ and $S_T^{VN}\left(|\psi\rangle\right)=0$, it is maximal (i.e., 1) when $S_T^{Sh}\left(|\psi\rangle\right)=S_T^{VN}\left(|\psi\rangle\right)$. Thus the intelligence of the QA state is maximal if the gap between the Shannon and the von Neumann entropy for the chosen result qubit is minimal.

Information QA-intelligent measure (6) and interrelations between information measures in Table 1 are used together with the step-by-step natural *majorization* principle for solution of QA-termination problem and interrelations between information measures $S_T^{Sh}(|\psi\rangle)\geq S_T^{VN}(|\psi\rangle)$ are used together with entropic relations of the step-by-step natural majorization principle for solution of QA-termi-

 DOI: https://doi.org/10.30564/aia.v1i1.619

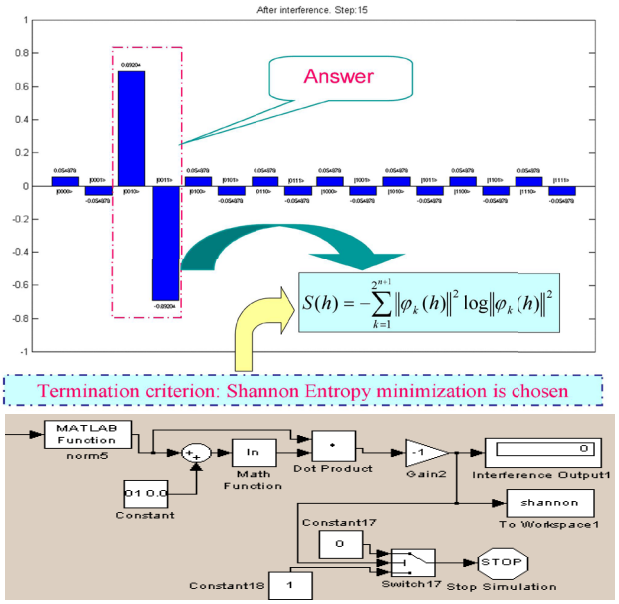
nation problem [12]. From Eq. (6) we can see that (for pure states)

$$\max \Im_T\left(|\psi\rangle\right) \mapsto 1-\min\left(\frac{S_T^{Sh}\left(|\psi\rangle\right)-S_T^{VN}\left(|\psi\rangle\right)}{|T|}\right)$$

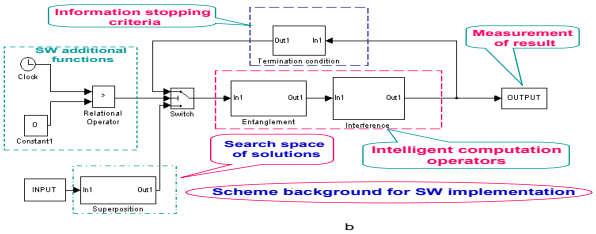
$$\mapsto \min S_T^{Sh}\left(|\psi\rangle\right), \quad S_T^{VN}\left(|\psi\rangle\right)=0, \qquad (7)$$

i.e. from Eq. (6) the principle of Shannon entropy minimum is as follows.

Figure 9 shows digital block of Shannon entropy minimum calculation and the main idea of the termination criterion based on this minimum of entropy [13, 14].



(a)



(b)

Figure 9. Digital Block of Shannon Entropy Minimum Calculation (a) and MatLab (b) Implementations

Number of iterations of QA defined during the calculation process of minimum entropy search.

The structure of HW implementation of main quantum operators.

Figure 10 shows the structure of superposition and interference operator simulation.

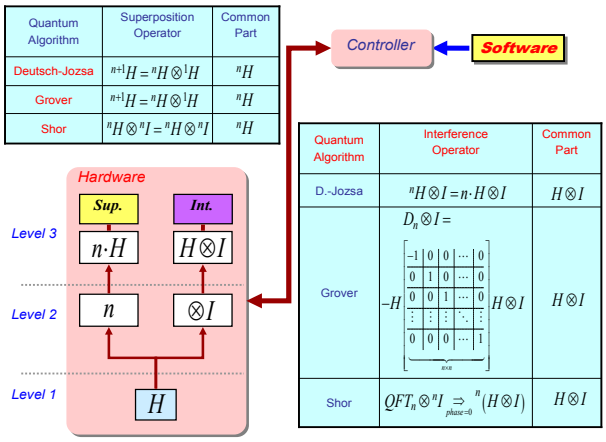


Figure 10. Computation of Superposition and Interference Operators

The superposition state is created by application of Hadamard matrix to column vector as $[1\ \ -1]^T=\begin{bmatrix}1\\-1\end{bmatrix}=\begin{bmatrix}1\\0\end{bmatrix}+\begin{bmatrix}0\\-1\end{bmatrix}=\left(|0\rangle-|1\rangle\right)$. According to this rule of quantum computing the superposition modeling circuit is developed [16].

Figure 11 shows the superposition modeling circuit.

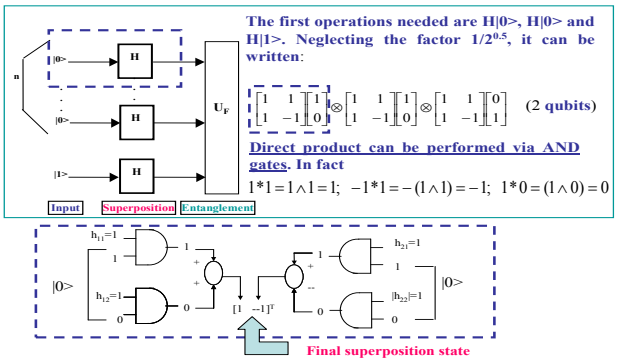


Figure 11. Superposition (Qubit) Modeling Circuit

Qubits simulation circuits with tensor product on Fig. 12 is shown.

$$\begin{bmatrix}1\\1\end{bmatrix}\otimes A=\begin{bmatrix}A\\A\end{bmatrix} \quad \text{or} \quad \begin{bmatrix}1\\0\end{bmatrix}\otimes A=\begin{bmatrix}A\\0\end{bmatrix}$$

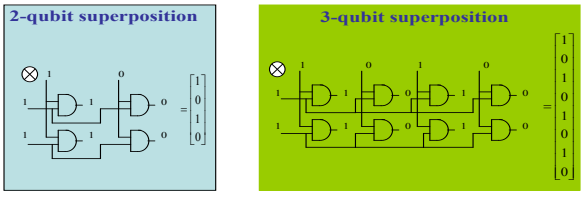


Figure 12. Qubits Simulation Circuits with Tensor Product

DOI: https://doi.org/10.30564/aia.v1i1.619

Figure 13 shows the computation of entanglement operators.

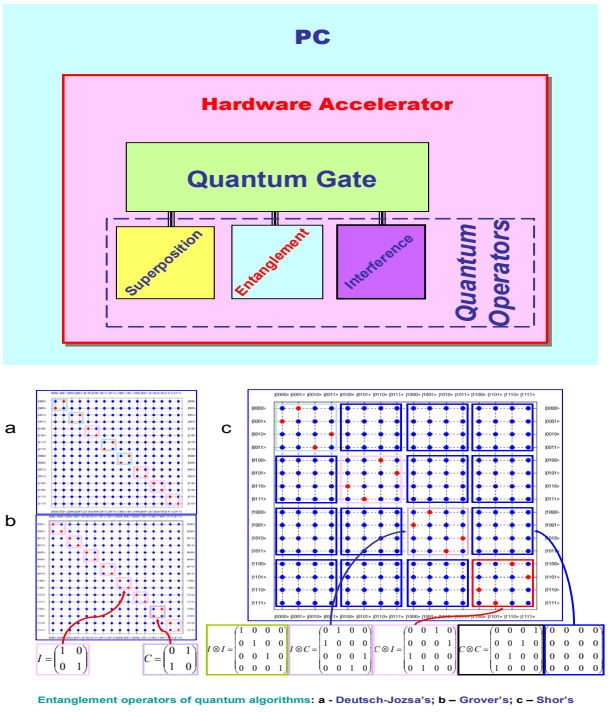


Figure 13. The Computation of Entanglement Operators

Figure 14 shows the entanglement creation circuit.

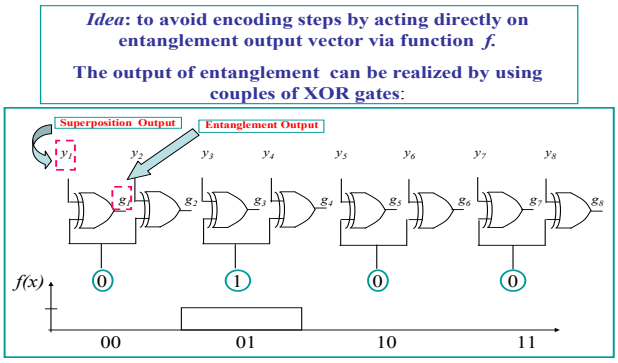
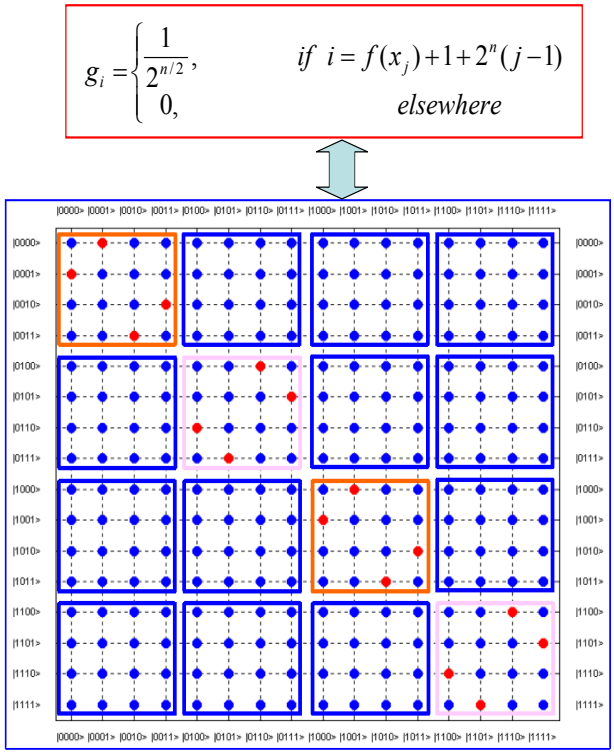


Figure 14. The Entanglement Creation Circuit

Thus it is possible to obtain output of entanglement $G=U_F \times Y$ without calculate matrix product and have only knowledge of corresponding row of diagonal U_F matrix (see, Fig. 13).

Finally output vector G can write as following (Fig. 15):



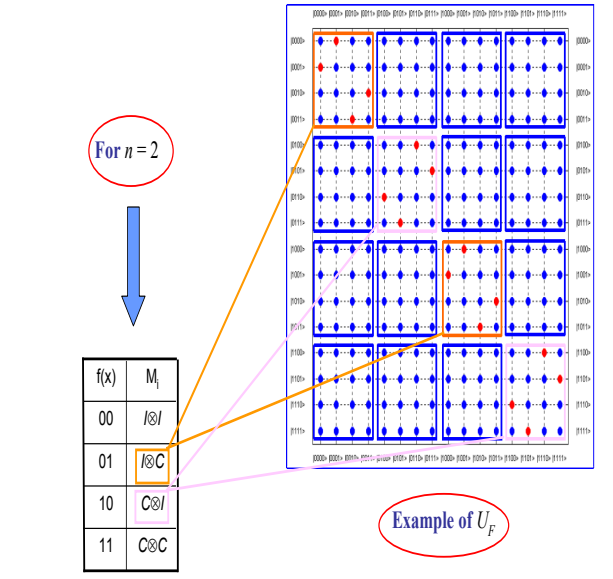


Figure 15: Equivalent form of Output Vector G

Figure 16 shows the entanglement circuit realization.

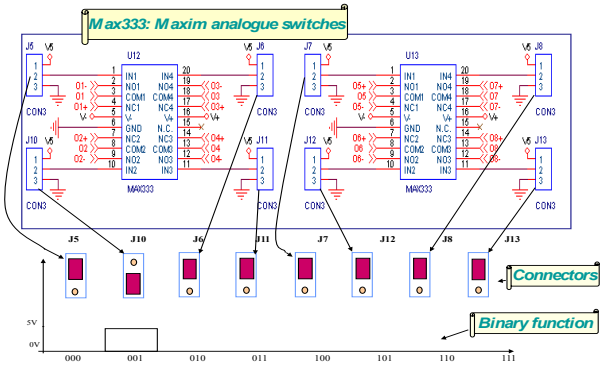


Figure 16. Entanglement Circuit Realization

Figure 17 shows the circuit realization of interference operator according to the scheme in Fig. 10.

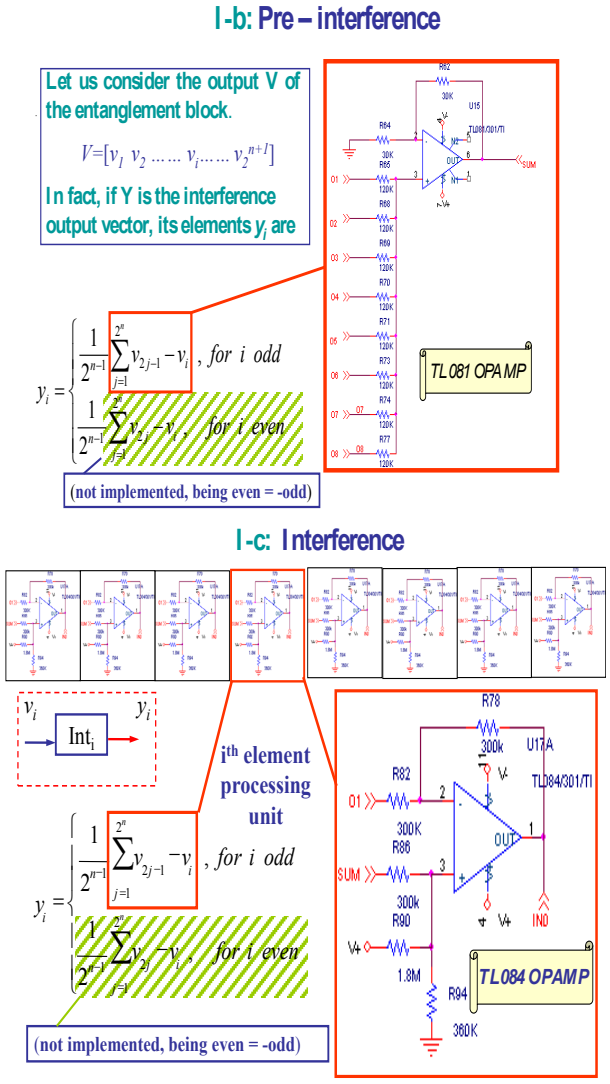


Figure 17. Interference Circuit Realization

Let us consider briefly applications of QAG design approach in highly structured QSA; and in AI, informatics, computer sciences and intelligent control problems (see Part II).

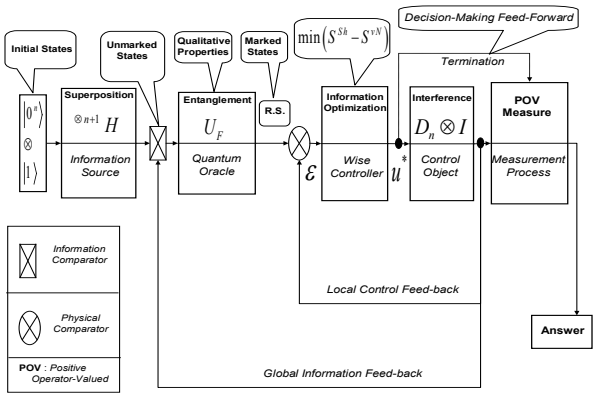
SIMULATION OF QA - COMPUTING ON CLASSICAL COMPUTER

We discuss the general outline of the Grover's QAs using the quantum gate (QAG) as

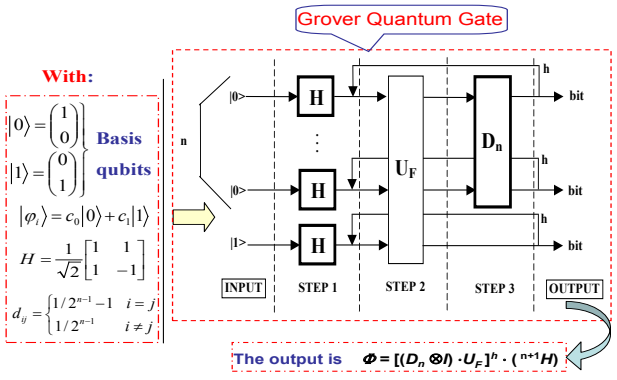
$$QAG^{Gr} = \left[\left(D_n \otimes I\right)\cdot U_F\right]^h \cdot \left(^{\otimes n+1}H\right) \qquad (7)$$

General method design of QAGs in [13, 14] is developed and is briefly described.

Figure 18a represents QAG of Grover's algorithm (7) as control system, and Fig. 18b describe a general structure scheme of Grover's QSA (see, Fig. 1 and Table 1) [13].



(a)



(b)

Figure 18. General Structure Scheme of Grover QSA

The Hadamard gates (Step 1) are the basic components for the superposition operation, the operator U_F (Step 2) performs entanglement operation and D_n (Step 3) is the diffusion matrix related to the interference operation. Our purpose is to realize some classical circuits (i.e. circuits

DOI: https://doi.org/10.30564/aia.v1i1.619

composed of classical gates *AND, NAND, XOR* etc.) that simulate the quantum operations of Grover QSA. To this aim all quantum operators must be expressed in terms of functions easily and efficiently described by classical components. When we try to make the HW components that perform this basic operations according to the classical scheme we encounter two main difficulties.

High-level gate design of Grover's QSA (Model based approach)

In this section we present a new model based HW implementing the functional steps of Grover's QSA from a high-level gate design point of view. According to the high-level scheme in Eq. (7) introduced in Fig. 4, the proposed circuit can be divided into two main parts.

Part I: (*Analogue*) Step-by-step *calculation of output values*. This part is divided into the following subparts:

I-a: *Superposition*;	**I-c**: *Pre-Interference* (for vector's approach);
I-b: *Entanglement*;	**I-d**: *Interference*

Part II: (*Digital*) Entropy evaluation, vector storing for iterations and output visualization. This part also provides initial superposition of basis vectors $|0\rangle$ and $|1\rangle$.

Figure 19 shows a general structure scheme of the HW realization for the Grover's QSA-circuits and itself can be considered as a classical prototype of intelligent control quantum system.

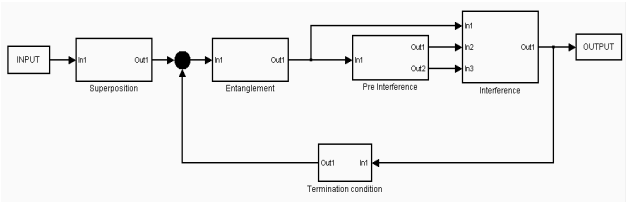


Figure 19. A General HW-scheme of the Grover's QSA

Example. The most interesting novelty involves the structure of interference: in fact the generic element v_i (interference output) can be written in function of g_i (entanglement output) as the following

$$v_i = \begin{cases} \frac{1}{2^{n-1}} \sum_{j=1}^{2^n} g_{2j-1} - g_i \quad , \quad for \quad i \quad odd \\ \frac{1}{2^{n-1}} \sum_{j=1}^{2^n} g_{2j} - g_i \quad , \quad for \quad i \quad even \end{cases} \tag{8}$$

Figures 20a and 20b show the Simulink schematic design and circuit realization of superposition, entanglement and interference operator's blocks of the Grover's QAG.

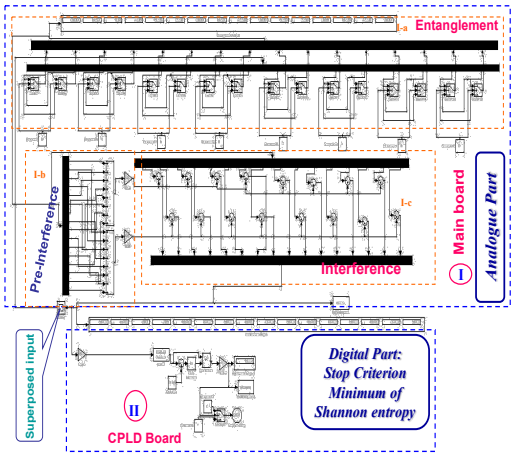


Figure 20a. Simulink Scheme of 3-qubits Grover Search System

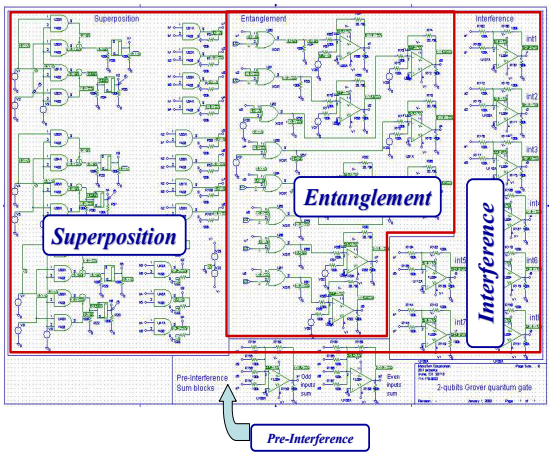


Figure 20b. Pre Prototype Scheme Circuit of Grover's QAG

Referring to Fig. 19, pre-interference operation evaluates a weighted sum of odd (even) output elements of entanglement, while interference itself uses this contribution in order to provide (by means of difference with g_i) the respective v_i. This simple (but powerful) result in Eq. (8) has several consequences.

Figure 21 shows experimental HW evolution of Grover's quantum search algorithm for three qubits.

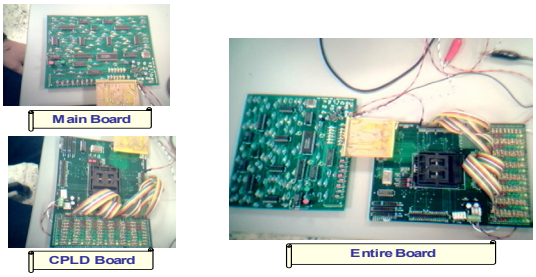


Figure 21. HW Realization of Grover QSA

 DOI: https://doi.org/10.30564/aia.v1i1.619

Remark. Regarding to speed-up of computation, a great improvement has been provided due to the smaller number of products (only one for each element of the output vector) and more precisely 2^{n+1} against 4^{n+1} of the classical approach. Also additions are less than $2^n\left(2^n+1\right)$ instead of 4^{n+1}). But the most important fact is that all these operation can be easily implemented in HW with few operational amplifiers $\left(2^n+2\right)$.

Example. Figure 22a - d shows the experimental probability evolution of finding each of the database's elements (from Iteration #1 - to Iteration #4). At this step (Iteration #2) the probabilities of finding one of the 8 elements of the database are comparable. In the following

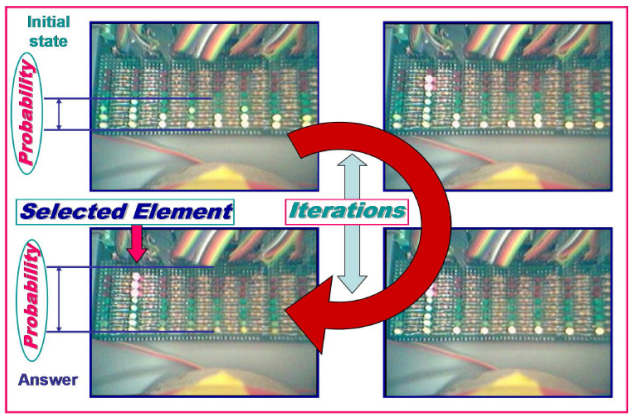


Figure 22. Experimental Results of 3-qubits HW-implementation of Grover's QSA

Figure 23 shows the result of entropy analysis for Grover's QSA according to Eq. (6), case $n=7, f\left(x_0\right)=1$.

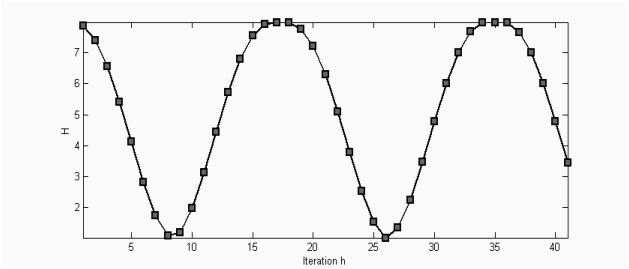


Figure 23. Shannon Entropy Simulation of QSA with 7-inputs

In Fig. 22c the probability of finding the second element of the database begins to increase with respect to the probabilities of finding the others elements. After some other iterations of the algorithm, the difference between the probability of finding the second element and the probabilities to find the others is increased. Finally the probability of extracting the second element of the database is greater than the probabilities of finding any other elements. Figures 22b, 22c and 22d show the evolution of quantum searching using Grover's QAG. It is a clear demonstration of how we can perform Grover's algorithm by a classical computer. Similar approach can be used for the realization of quantum fuzzy computing [27].

Application of Grover's QAG is classical efficient simulation process for realization of quantum search computation on classical computer (see in details [17]).

QUANTUM ALGORITHM ACCELATOR COMPUTING: SW SUPPORT: EXAMPLES (matrix approach)

The software system into two general sections is divided (see, Fig. 24).

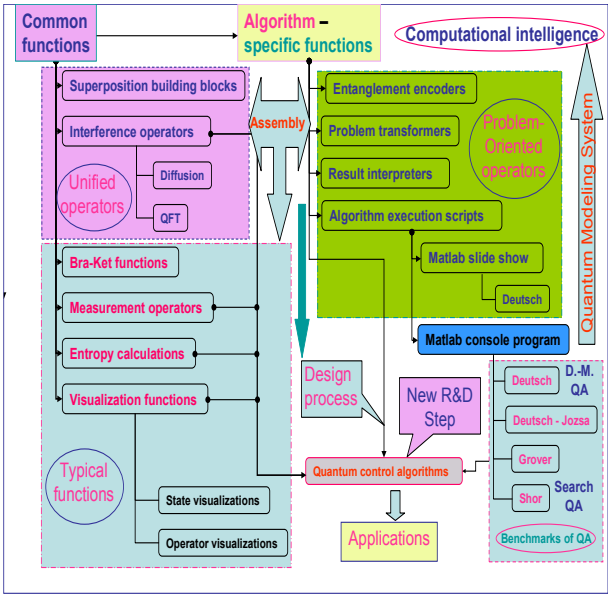


Figure 24. Structure of QFMS and SW Toolkit

The first section involves common functions. The second section involves algorithm-specific functions for realizing the concrete algorithms.

Figure 25 shows of quantum mechanical representation in SW of (*bra – ket*) vectors and calculation of quantum states as density matrices.

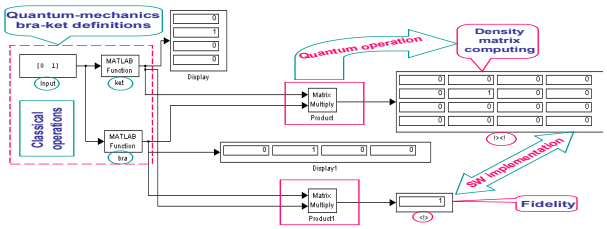


Figure 25. SW Representation of Density Matrix and Fidelity Calculation

Example: *Quantum Shor's Algorithm* (*Quantum factorization promise*). Figure 26 shows the factorization problem. Figure 27 shows the quantum Shor algorithm and its describing circuit (see Table 1). We can observe U_F block that is a diagonal matrix of $2^{2n}\times 2^{2n}$ dimension. Finally output of entanglement is processed by interference

DOI: https://doi.org/10.30564/aia.v1i1.619

block composed of Quantum Fourier Transform (*QFT*) and identity matrix *I*. The output of entire algorithm is therefore the vector obtained after application of operator $QFT_n \otimes {}^n I$.

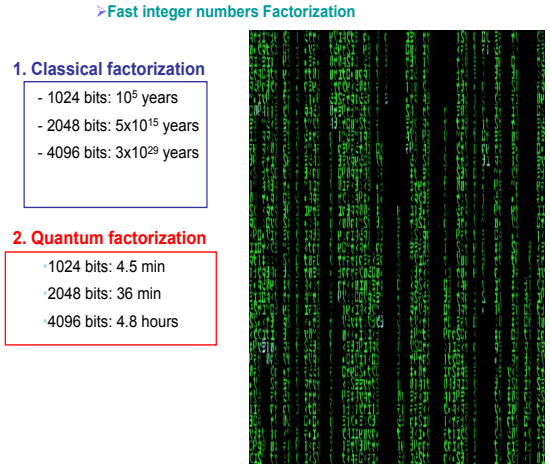


Figure 26. Fast Factorization Problem and Its Solutions

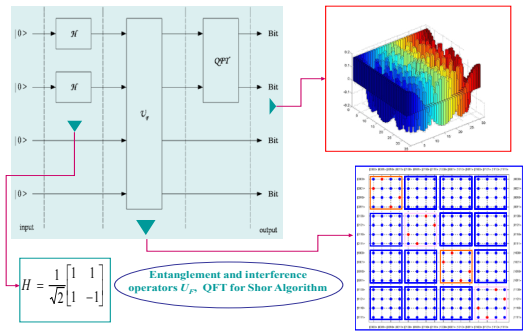


Figure 27. Quantum Shor's Algorithm Circuit and Main Quantum Operators

Factorization time using matrix and vector approach are here reported (see, Fig. 28).

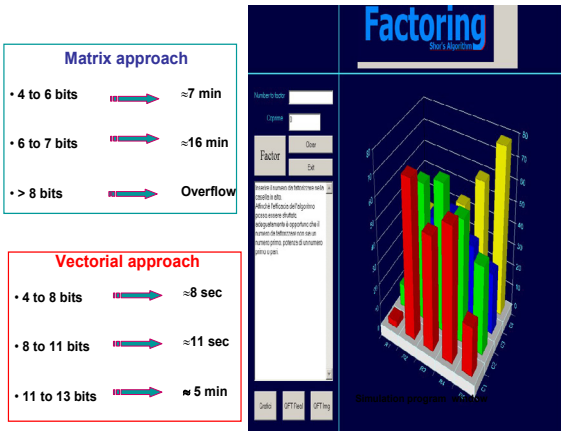


Figure 28. SW Simulation of Shor's Quantum Factorization Algorithm

Example: *Command line simulation of the Grover's quantum search algorithm* The example of the Grover's algorithm script is presented in Figs 29 and 30.

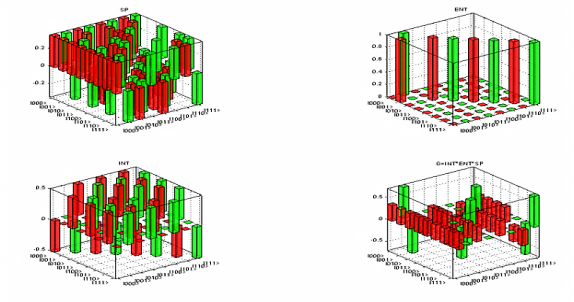


Figure 29. Example of Grover Algorithm Simulation Script (Visualization of the Quantum Operators *Sp*, *Ent*, *Int* and $G = (Int)(Ent)(Sp)$)

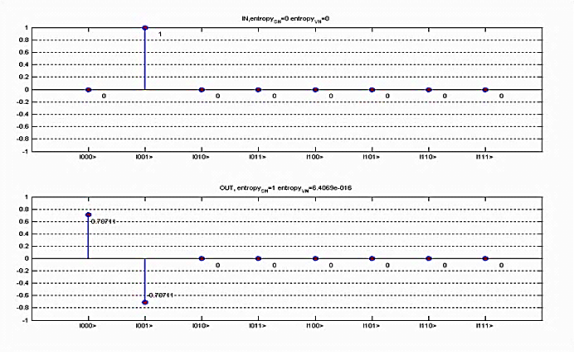


Figure 30. Example of Grover Algorithm Simulation Script (Visualization of the Input and of the Output Quantum States)

In Fig. 29, the algorithm-related script is presented. It prepares the superposition (SP), entanglement (ENT) and interference (INT) operators of the Grover's algorithm with 3 q-bits (including the measurement q-bit). Then it assembles operators into the quantum gate *G*.

Then the script creates an input state $|in\rangle = |001\rangle$ and calculates the output state $|out\rangle = G \cdot |in\rangle$. The result of this algorithm in Matlab is an allocation of the operator matrices and of the state vectors in the memory. Code displays the operator matrices in Fig. 29 in 3D visualization. In this case the vertical axis corresponds to the amplitudes of the corresponding matrix elements. Indexes of the elements are marked with the ket notation. Input |in> and the output |out> states are demonstrated in Fig. 25. In this case, the vertical axis corresponds to the probability amplitudes of the state vector components. The horizontal axis corresponds to the index of the state vector component, marked using the ket notation.

The title of the Fig. 30 contains the values of the Shannon and of the von Neumann entropies of the corresponding visualized states.

 DOI: https://doi.org/10.30564/aia.v1i1.619

Other known QA can be formulated and executed using similar scripts, and by using the corresponding equations taken from the previous section.

Simulation of QAs as dynamic control system

In order to simulate behavior of the dynamic systems with quantum effects, it is possible to represent the QA as a dynamic system in the form of a block diagram and then simulate its behavior in time. Figure 31 is an example of a Simulink diagram of the quantum circuit for calculation of the fidelity $\langle a|a\rangle$ of the quantum state and for the calculation of the density matrix $|a\rangle\langle a|$ of the quantum state. Bra and ket functions are taken from the common library. This example demonstrates the usage of the common functions for the simulation of the QA dynamics.

In Fig. 31, input is provided to the ket function. The output of the ket function is provided to the first input of the matrix multiplier and as a second input of the matrix multiplier. Input is also provided to the bra function. The output of the bra function is provided to the second input of the matrix multiplier and as a first input of the matrix multiplier. Output of the multiplier is a density matrix of the input state. Output of the multiplier is the fidelity of the input state.

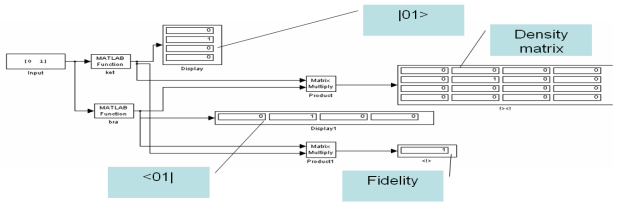


Figure 31. Simulink Diagram for the Simulation of the Arbitrary Quantum Algorithm

Figure 32 shows Simulink structure of an arbitrary QA. Such a structure can be used to simulate a number of quantum algorithms in Matlab / Simulink environment.

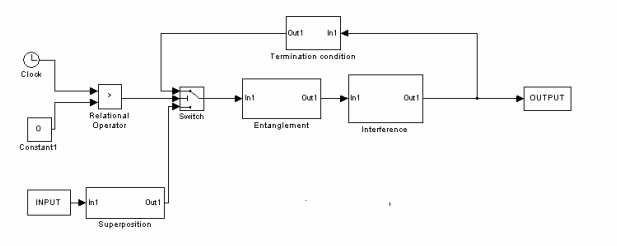


Figure 32. Simulink Diagram for the Simulation of the Arbitrary QA

Simulation result of Grover's QSA on Fig. 33 is shown.

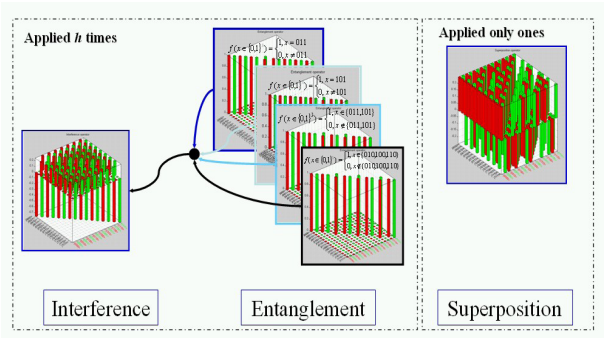


Figure 33. Evolution of Grover's Quantum Search Algorithm: Quantum Simulator on Classical Computer (Matrix Approach)

Dynamic evolution of successful results of algorithm execution for the first iteration of Grover's QAG for initial qubits state $|0001\rangle$ and different answer search is shown in Fig. 34.

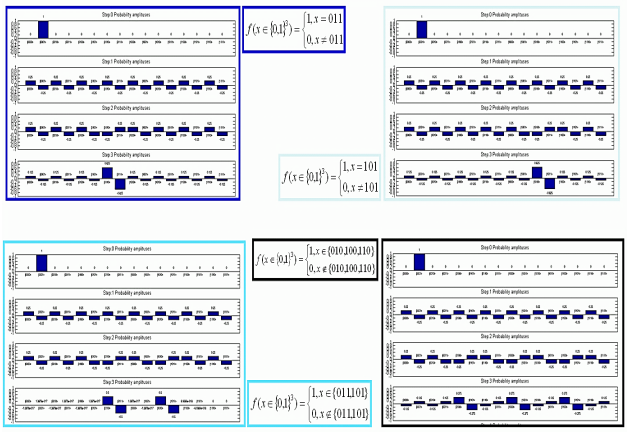


Figure 34. Grover's QSA: Algorithm Execution [First Iteration]

Figure 35 shows algorithm execution results for Grover's QSA with different number of iterations for successful results with different searching answer number.

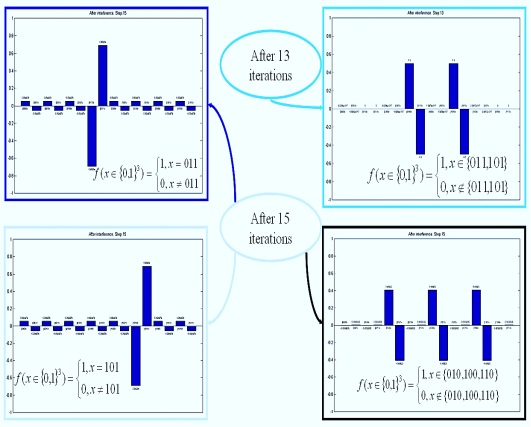


Figure 35. Grover's QA: Step 2. Algorithm Execution Results

DOI: https://doi.org/10.30564/aia.v1i1.619

Figure 36 is a 3D dynamic representation of Grover's QAG probabilities evolution (step 2 of Fig. 33) for different cases of answer search.

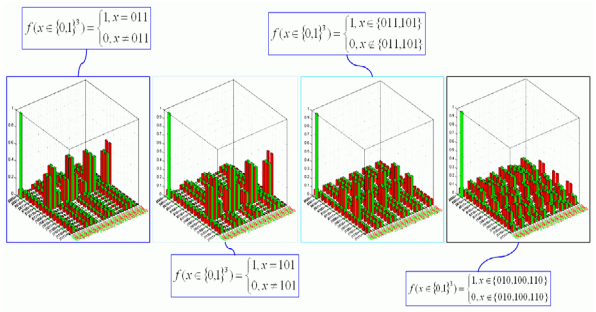


Figure 36. Grover's QA: Step 2 [Algorithm Execution 3D Dynamics: Probabilities]

Algorithm execution results of Grover's QAG (step 2 of Fig. 33) with different stopping iteration for searching answers are shown in Fig. 35.

Example: Interpretation of measurement results in simulation of Grover's QSA-QAG.

In the case of Grover's *QSA* this task is achieved (according to the results of this section) by preparing the ancillary qubit of the oracle of the transformation: $U_f : |x,a\rangle \mapsto |x, f(x) \oplus a\rangle$ in the state $|a_0\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. The operator $I_{|x_0\rangle}$ is computationally equivalent to U_f:

$U_f\left[\lvert x\rangle \otimes \frac{1}{\sqrt{2}}(\lvert 0\rangle - \lvert 1\rangle)\right]$	=	$\left[I_{\lvert x_0\rangle}(\lvert x\rangle)\right] \otimes \frac{1}{\sqrt{2}}(\lvert 0\rangle - \lvert 1\rangle)$
	=	$\underbrace{\frac{1}{\sqrt{2}}\left[I_{\lvert x_0\rangle}(\lvert x\rangle)\right]}_{\text{Computation Result}} \otimes \underbrace{\lvert 0\rangle}_{\text{Measurement}} - \underbrace{\frac{1}{\sqrt{2}}\left[I_{\lvert x_0\rangle}(\lvert x\rangle)\right]}_{\text{Computation Result}} \otimes \underbrace{\lvert 1\rangle}_{\text{Measurement}}$

and the operator U_f is constructed from a controlled $I_{|x_0\rangle}$ and two one qubit Hadamard transformations. Figure 37 shows the interpretation of results of the Grover QAG.

If measured basis vector:	$\underbrace{\lvert x_0^1 \cdots x_0^n x_0^{n+1}\rangle}_{n+1 \text{ qubits}}$
Consists of:	$\underbrace{\lvert x_0^1 \cdots x_0^n\rangle}_{n \text{ qubits of computational basis}} \otimes \underbrace{\lvert x_0^{n+1}\rangle}_{1 \text{ qubit of measurement basis}}$
Then searched argument was:	$x_0 = \underbrace{x_0^1 \cdots x_0^n}_{n \text{ bits}} \Rightarrow$ Answer of Quantum Searching

Figure 37. Grover's QA: Step 2 [Result Interpretation]

Measured basis vector is computed from the tensor product between the computation qubit results and ancillary measurement qubit. In Grover's searching process the ancillary qubit does not change during the quantum computing.

As described above operator U_f is constructed from two Hadamard transformations and the Hadamard transformation H (modeling the constructive interference) applied on the state of the standard computational basis can be seen as implementing a fair coin tossing. Thus, if the matrix $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ is applied to the states of the standard basis then $H^2|0\rangle = -|1\rangle$, $H^2|1\rangle = |0\rangle$ and therefore H^2 acts in measurement process of computational result as a *NOT*-operation up to the phase sign. In this case the measurement basis separated with the computational basis (according to tensor product).

Figures 38 and 39 shows the measurement result and final results of entropy dynamic evolution interpretation of Grover's QSA for search of successful results with different number of marked states (in computational basis $\{|0\rangle, |1\rangle\}$). These results represent the possibility of the classical efficient simulation of Grover's QSA.

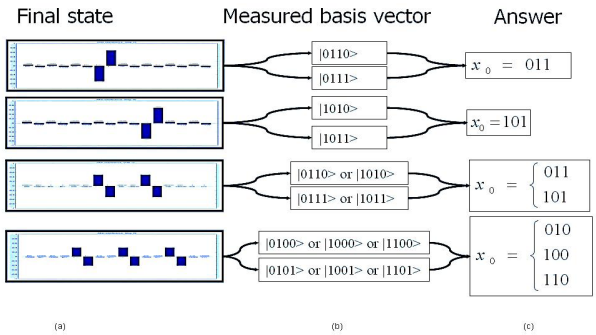


Figure 38. Interpretation of Measurement Results of QSA

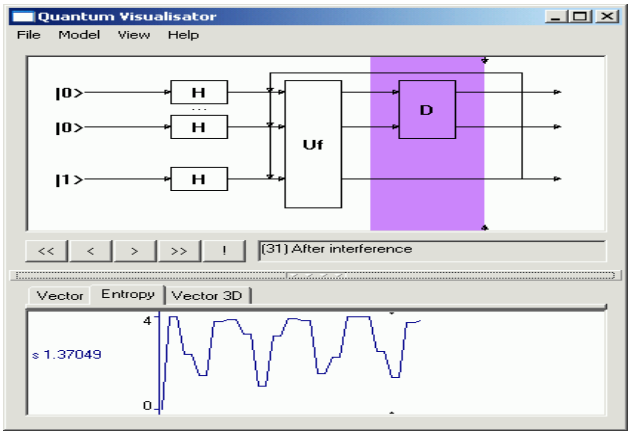


Figure 39. Shannon Entropy Dynamics after 31 Steps of Grover's QSA

Remark. Figure 38 (b) shows the results of computation on a classical computer and shows two possibilities:

and $\left\{ |0110\rangle = \underbrace{|011\rangle}_{Result} \otimes \underbrace{|0\rangle}_{measurement\ qubit} \right\}$

$$\Updownarrow \qquad \Updownarrow$$

$$\left\{ |0111\rangle = \underbrace{|011\rangle}_{Result} \otimes \underbrace{|1\rangle}_{measurement\ qubit} \right\}$$

Figure 38 (b) demonstrates also two searching marked states:

$$\left\{ |0110\rangle = |011\rangle \otimes \underbrace{|0\rangle}_{measurement\ qubit} \quad or \quad |1010\rangle = |101\rangle \otimes \underbrace{|0\rangle}_{measurement\ qubit} \right\}$$

and

$$\Updownarrow \qquad \Updownarrow$$

$$\left\{ |0111\rangle = |011\rangle \otimes \underbrace{|1\rangle}_{measurement\ qubit} \quad or \quad |1011\rangle = |101\rangle \otimes \underbrace{|1\rangle}_{measurement\ qubit} \right\}$$

A similar situation is shown for two and three searching marked states in Fig. 37 (b).

Using a random measurement strategy based on a fair coin tossing in the measurement basis $\{|0\rangle, |1\rangle\}$ one can independently receive with certainty the searched marked states from the measurement basis result.

The measurement results based on a fair coin tossing measurement are shown in Fig. 38 (c) and shows accurate results of searching of corresponding marked states. Figure 38 (c) shows also that for both possibilities in implementing a fair coin tossing type of measurement process the search for the answer is successful.

Final results of interpretation for Grover's algorithm are shown in Fig. 38.

Let us describe briefly the main blocks in Fig. 2: i) unified operators; ii) problem-oriented operators; iii) Benchmarks of QA simulation on classical computers; and iv) quantum control algorithms based on quantum fuzzy inference (QFI) and quantum genetic algorithm (QGA) as new types of QSA (see, Part II of this article).

Let us consider problem-oriented operators description.

Problem-oriented approach based on structural pattern of QA state vector with compressed vector allocation.

Let n be the input number of qubits. In the Grover algorithm (as mentioned above) half of all 2^{n+1} elements of a vector making up its even components always take values symmetrical to appropriate odd components and, therefore, need not be computed.

Odd 2^n elements can be classified into two categories:

• The set of *m* elements corresponding to truth points of input function (or oracle); and

• The remaining $2^n - m$ elements.

The values of elements of the same category are always equal.

As discussed above, the Grover QA only requires two variables for storing values of the elements. Its limitation in this sense depends only on a computer representation of the floating-point numbers used for the state vector probability amplitudes. For a double-precision software realization of the state vector representation algorithm, the upper reachable limit of q-bit number is approximately 1024 [13].

Figure 40 shows a state vector representation algorithm for the Grover QA.

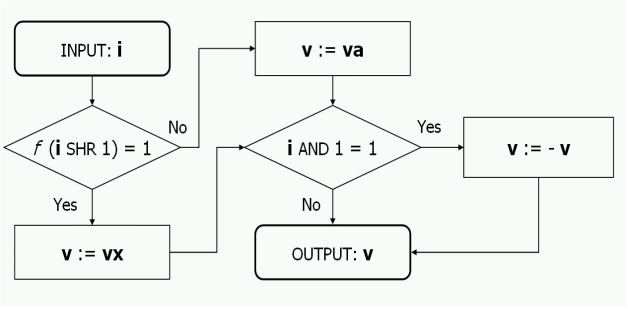


Figure 40. State Vector Representation Algorithm for Grover' Quantum Search

Remark. In Fig. 40, i is an element index, *f* is an input function, *vx* and *va* corresponds to the elements' category, and *v* is a temporal variable. The number of variables used for representing the state variable is constant. A constant number of variables for state vector representation allow reconsideration of the traditional schema of quantum search simulation.

Classical gates are used not for the simulation of appropriate quantum operators with strict one-to-one correspondence but for the simulation of a quantum step that changes the system state. Matrix product operations are replaced by arithmetic operations with a fixed number of parameters irrespective of qubit number.

Figure 41 shows a generalized schema for efficient simulation of the Grover QA built upon three blocks, a superposition block *H*, a quantum step block *UD* and a termination block *T*.

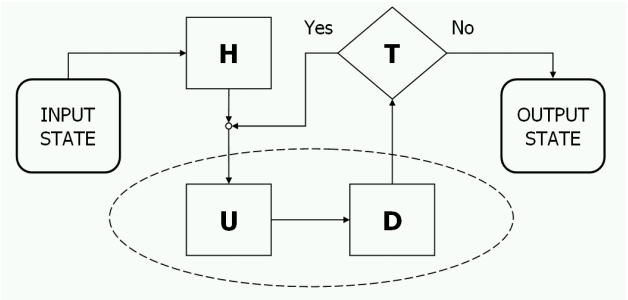


Figure 41. Generalized Schema of Simulation for Grover' QSA

Figure 41 also shows an input block and an output

DOI: https://doi.org/10.30564/aia.v1i1.619

block.

Remark. The *UD* block includes a *U* block and a *D* block. The input state from the input block is provided to the superposition block. *A* superposition of states from the superposition block is provided to the *U* block. An output from the *U* block is provided to the *D* block. An output from the *D* block is provided to the termination block. If the termination block terminates the iterations, then the state is passed to the output block; otherwise, the state vector is returned to the *U* block for iteration.

As shown in Fig. 42, the *superposition* block *H* for Grover QSA simulation changes the system state to the state obtained traditionally by using $n + 1$ times the tensor product of Walsh-Hadamard transformations. In the process shown in Fig. 41, *vx:= hc*, *va:= hc*, and *vi:*= 0, where $hc = 2^{-(n+1)/2}$ is a table value.

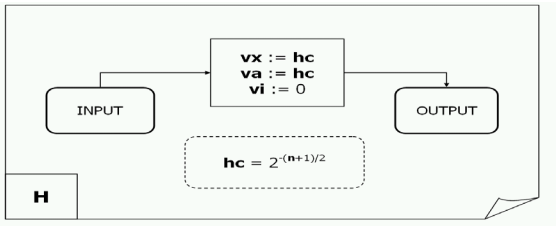


Figure 42. Superposition Block for Grover's QSA

The quantum step block UD that emulates the *entanglement* and *interference* operators is shown on Figs 43 (a - c).

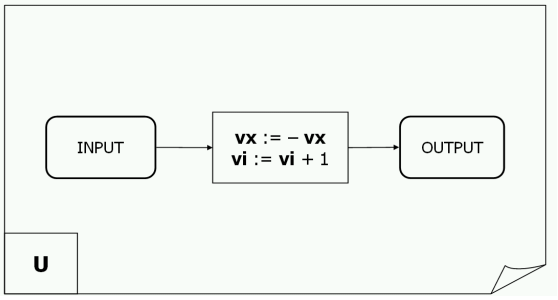


Figure 43 (a). Emulation of the Entanglement Operator Application of Grover's QSA

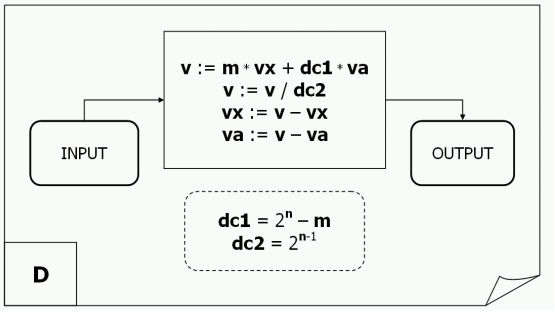


Figure 44 (b). Emulation of Interference Operator Application of Grover's QSA

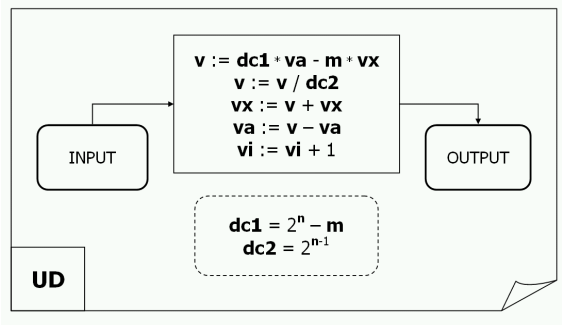


Figure 44 (c). Quantum Step Block for Grover' Quantum Search

The *UD* block reduces the temporal complexity of the quantum algorithm simulation to linear dependence on the number of executed iterations. The *UD* block uses recalculated table values $dc1 = 2^n - m$ and $dc2 = 2^{n-1}$.

Remark. In the *U* block shown in Fig. 44 (a), *vx:*= - *vx* and *vi:*= *vi* + 1. In the *D* block shown in Fig. 44 (b), *v:*= *m***vx*+*dc*1**va, v:*= *v*/*dc*2, *vx:*= *v* - *vx*, and *va:*= *v* - *va* in the *UD* block shown in Fig. 44 (c), *v:*= *dc*1**va* = *m***vx*, *v:*=*v*/*dc*2, *vx:*=*v* + *vx, va:*= *v* - *va*, and *vi:*= *vi* + 1.

The termination block *T* is general for all QAs, independently of the operator matrix realization. Block *T* provides intelligent termination condition for the search process. Thus, the block *T* controls the number of iterations through the block *UD* by providing enough iteration to achieve a high probability of arriving at a correct answer to the search problem. The block *T* uses a rule based on observing the changing of the vector element values according to two classification categories. The *T* block during a number of iterations, watches for values of elements of the same category monotonically increase or decrease while values of elements of another category changed monotonically in reverse direction. If after some number of iteration the direction is changed, it means that an extremum point corresponding to a state with maximum or minimum uncertainty is passed. The process can using direct values of amplitudes instead of considering Shannon entropy value, thus, significantly reducing the required number of calculations for determining the minimum uncertainty state that guarantees the high probability of a correct answer.

The *termination* algorithm realized in the block T can be used one or more of five different termination models:

o *Model* 1: Stop after a predefined number of iterations;

o *Model* 2: Stop on the first local entropy minimum;

o *Model* 3: Stop on the lowest entropy within a predefined number of iterations;

o *Model* 4: Stop on a predefined level of acceptable entropy; and/or

o *Model* 5: Stop on the acceptable level or lowest

 DOI: https://doi.org/10.30564/aia.v1i1.619

reachable entropy within the predefined number of iterations.

Note that models 1 - 3 do not require the calculation of an entropy value.

Figures 45 – 47 show the structure of the termination condition blocks *T*.

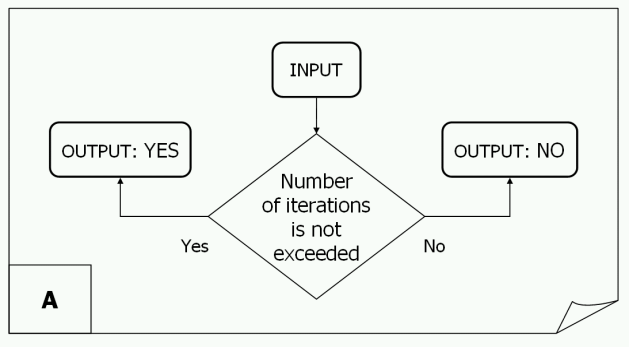


Figure 45. Termination Block for Method 1

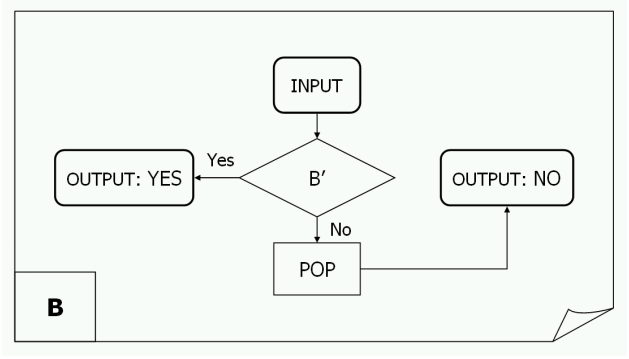


Figure 46. Component B for the Termination Block

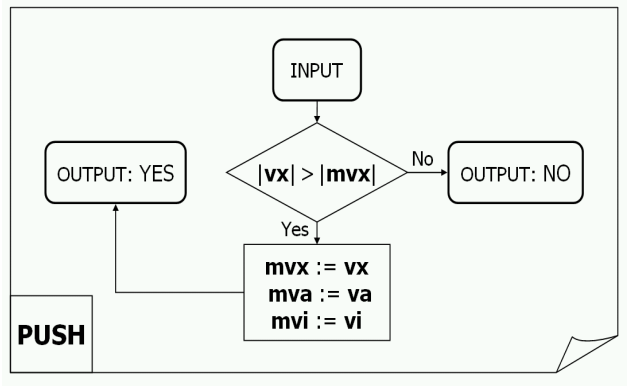


Figure 47 (a). Component PUSH for the Termination Block

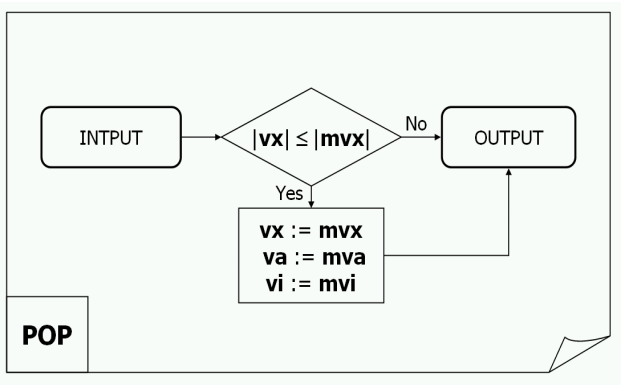


Figure 47 (b). Component POP for the Termination Block

Since time efficiency is one of the major demands on such termination condition algorithm, each part of the termination algorithm is represented by a separate module, and before the termination algorithm starts, links are built between the modules in correspondence to the selected termination model by initializing the appropriate functions' calls.

Table 2 shows components for the termination condition block T for the various models. Flow charts of the termination condition building blocks are provided in Figs 45 – 50

Table 2. Termination Block Construction

Model	T	B'	C'
1	A	--	--
2	B	PUSH	--
3	C	A	B
4	D	--	--
5	C	A	E

The entries *A, B, PUSH, C, D, E,* and *PUSH* in Table 2 correspond to the flowcharts in Figs 40 – 45 respectively.

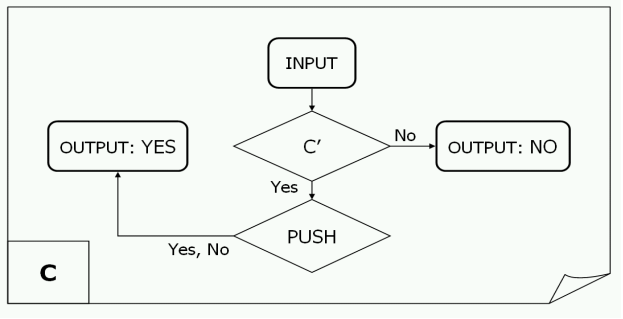


Figure 48. Component *C* for the Termination Block

DOI: https://doi.org/10.30564/aia.v1i1.619

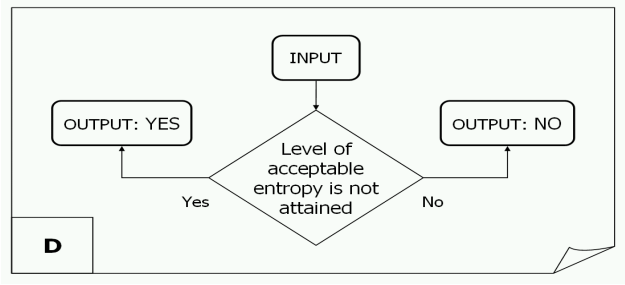


Figure 49. Component D for the Termination Block

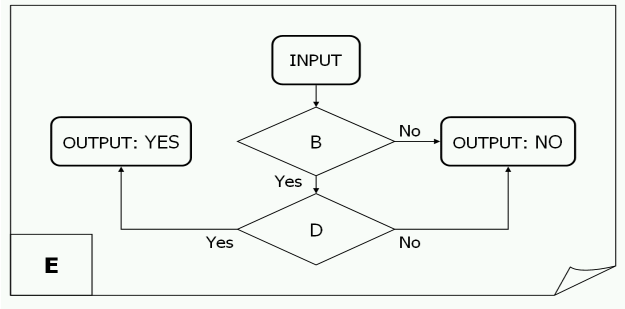


Figure 50. Component *E* for the Termination Block

Remark: Peculiarities of QA termination models in *model* 1, only one test after each application of quantum step block *UD* is needed. This test is performed by block *A*. So, the initialization includes assuming *A* to be *T*, i.e., function calls to *T* are addressed to block *A*. Block *A* is shown in Fig. 45 and checks to see if the maximum number of iterations has been reached, if so, then the simulation is terminated, otherwise, the simulation continues.

In *model* 2, the simulation is stopped when the direction of modification of categories' values are changed. Model 2 uses the comparison of the current value of *vx* category with value *mvx* that represents this category value obtained in previous iteration:

(i) If *vx* is greater than *mvx*, its value is stored in *mvx*, the *vi* value is stored in *mvi*, and the termination block proceeding to the next quantum step;

(ii) If *vx* is less than *mvx*, it means that the *vx* maximum is passed and the process needs to set the current (final) value of *vx*: = *mvx*, *vi* := *mvi*, and stop the iteration process. So, the process stores the maximum of *vx* in *mvx* and the appropriate iteration number *vi* in *mvi*. Here block *B*, shown in Fig. 46 is used as the main block of the termination process.

The block PUSH, shown in the Fig. 47 (a) is used for performing the comparison and for storing the *vx* value in *mvx* (*case a*). A POP block, shown in Fig. 47 (b) is used for restoring the *mvx* value (*case b*). In the PUSH block of Fig. 47 (a), *if* |*vx*| > |*mvx*|, then *mvx*: = *vx*, *mva*: = *va*, *mvi*: = *vi*, and the block returns true; otherwise, the block returns . In the POP block of Fig. 47 (b), if |*vx*| <= |*mvx*|, then *vx*: = *mvx*, *va*:= *mva*, and *vi*:= *mvi*.

The *model* 3 termination block checks to see that a predefined number of iterations do not exceed (using block *A* in Fig. 43):

• If the check is successful, then the termination block compares the current value of *vx* with *mvx*. If *mvx* is less than, it sets the value of *mvx* equal to *vx* and the value of *mvi* equal to *vi*. If *mvx* is less using the PUSH block, then perform the next quantum step;

• If the check operation fails, then (if needed) the final value of *vx* equal to *mvx*, *vi* equal to *mvi* (using the POP block) and the iterations are stopped.

The *model* 4, the termination block uses a single component block *D*, shown in Fig. 48. The *D* block compares the current Shannon entropy value with a predefined acceptable level. If the current Shannon entropy is less than the acceptable level, then the iteration process is stopped; otherwise, the iterations continue.

The *model* 5 termination block uses the *A* block to check that a predefined number of iterations do not exceeded (see, Fig. 45). If the maximum number is exceeded, then the iterations are stopped. Otherwise, the *D* block is then used to compare the current value of the Shannon entropy with the predefined acceptable level. If acceptable level is not attained, then the PUSH block is called and the iterations continue. If the last iteration was performed, the POP block is called to restore the *vx* category maximum and appropriate *vi* number and the iterations are ended.

Figure 51 shows measurement of the final amplitudes in the output state to determine the success or failure of the search.

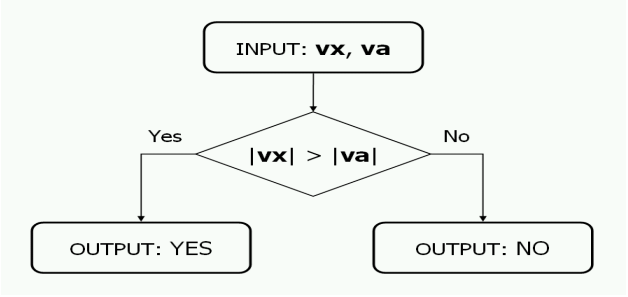


Figure 51. Final Measurement Emulation

If |*vx*| > |*va*|, then the search was successful; otherwise, the search was not successful.

Table 3 lists results of testing the optimized version of Grover QSA simulator on personal computer with Pentium 4 processor at 2GHz.

Table 3 High Probability Answers for Grover QSA

Qbits	Iterations	Time
32	51471	0.007

 DOI: https://doi.org/10.30564/aia.v1i1.619

36	205887	0.018
40	823549	0.077
44	3294198	0.367
48	13176794	1.385
52	52707178	2.267
56	210828712	20.308
60	843314834	81.529
64	3373259064	328.274

Simulation results

The theoretical boundary of this approach is not the number of qubits, but the representation of the floating-point numbers. The practical bound is limited by the front side bus frequency of the personal computer. Using the above algorithm, a simulation of a 1000 qubit Grover QSA requires only 96 seconds for 10^8 iterations.

Figure 52 shows the simulation result of Grover's algorithm [problem-oriented approach with compressed vector allocation] [14].

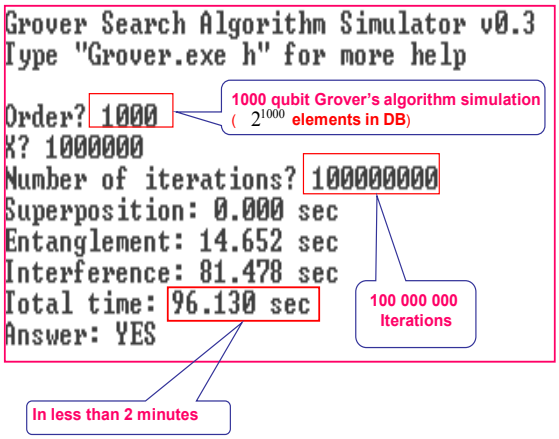


Figure 52. Simulation Results of Problem Oriented Grover's QSA According to Approach 4 with 1000 Qubit (Simulator Window Snapshot)

The described method is differed from [23-44].

Let us discuss briefly the applications of QAG approach in design of new types of quantum search algorithm as quantum genetic algorithm.

Quantum fuzzy inference and quantum genetic algorithm: Quantum simulator

Intelligent control systems (ICS) based on the use of soft computing, fuzzy logic, evolutionary algorithms and neural networks. Basis of management systems - proportional–integral–derivative (PID) controller, which is used in 70% of the industrial automation, but often can`t cope with the task of managing and does not work well in unpredicted control situations. The use of quantum computing and quantum search algorithms, as a special example, quantum fuzzy inference (QFI), allows increasing robustness without the cost of a time resource - in on line.

Figure 53 show an ICS with the integration of several fuzzy controllers and QFI which allows create a new quality in management – online self-organization of knowledge base (KB) [45].

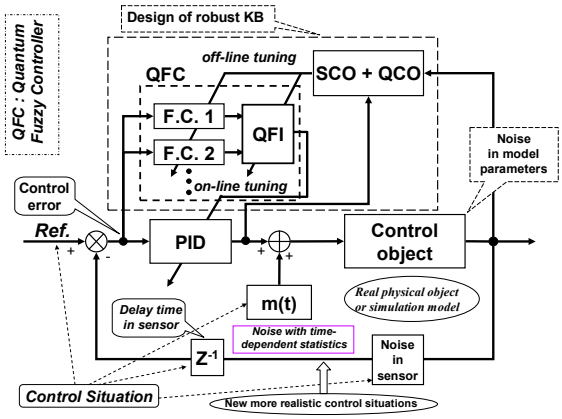


Figure 53. Structure of robust ICS based on QFI

In general, the structure of a quantum algorithmic gate (QAG) based on a quantum genetic algorithm (QGA) described in (9) in the form:

$$QAG = \left[\left(Int \otimes^{n} I\right) \cdot U_F\right]^{h+1} \cdot [QGA]\left[{}^{n}H \otimes^{m} S\right]. \quad (9)$$

Structure of corresponding QAG on Fig. 54 is shown.

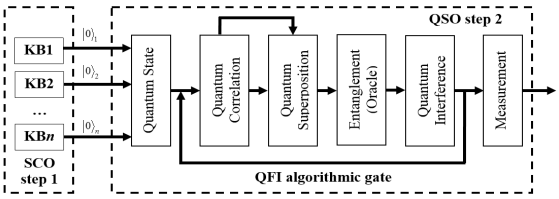


Figure 54. QAG Structure of QFI

The first part in designing Eq. (9) is the choice of the type of the entangled state of operator U_F. The basic unit of such an ICS is the quantum genetic search algorithm (QGPA) (see, Fig. 55).

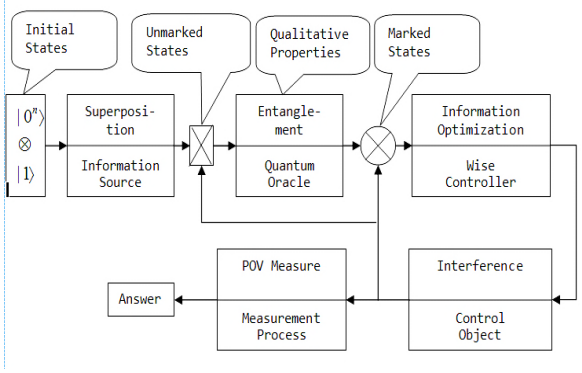


Figure 55. Intelligent Self-organizing Quantum Search Algorithm for Intelligent Control Systems

DOI: https://doi.org/10.30564/aia.v1i1.619

Results of simulation show computing effectiveness of robust stability and controllability of (QFI + QGA)-controller and new information synergetic effect: from two fuzzy controllers with imperfect knowledge bases can be created robust intelligent controller (extracted hidden quantum information from classical states is the source of value work for controller [45]). Intelligent control systems with embedding intelligent QFI-controller can be realized either on classical or on quantum processors (as an example, on D-Wave processor type).

Two classes of quantum evolution (9) are described: quantum genetic algorithm (QGA) and hybrid genetic algorithm (HGA). The QFI algorithm for determining new PID coefficient gain schedule factors K (see Fig.56) consists of such steps as normalization, the formation of a quantum bit, after which the optimal structure of a QAC is selected, the state with the maximum amplitude is selected, decoding is performed and the output is a new parameter *K*.

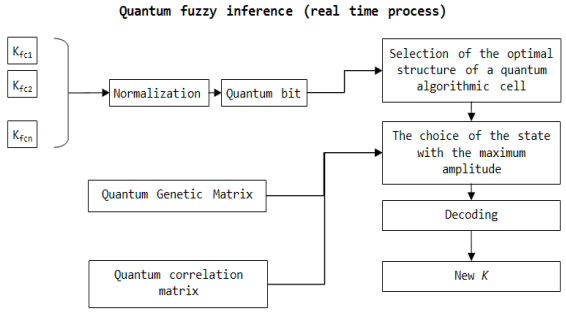


Figure 56. QFI Algorithm Structure on Line

At the input, the QFI obtains coefficients from the fuzzy controller knowledge bases formed in advance based on the KB optimizer on soft calculations.

The next step is carried out normalization of the received signals [0, 1] by dividing the current values of control signals at their maximum values (*max k*), which are known in advance.

Formation of quantum bits. The probability density functions are determined. They are integrated and they make the probability distribution function. They allow defining the virtual state of the control signals for generating a superposition via Hadamard transform of the current state of the entered control signals.

The law of probability is used: $(|0\rangle)+(|1\rangle)=1$, where $p(|0\rangle)$ is the probability of the current real state and $p(|1\rangle)$ is the probability of the current virtual state. The superposition of the quantum system "*real* state - *virtual* state" has the form: $|\psi\rangle=\frac{1}{2}\left(\sqrt{p\left(|0\rangle\right)}|0\rangle+\sqrt{1-p\left(|0\rangle\right)}|1\rangle\right)$.

The next step is selection of the type of quantum correlation - constructing operation of entanglement. Three types of quantum correlation are considered: spatial, temporal and spatial-temporal. Each of them contains valuable quantum information hidden in a KB.

Quantum correlation considered as a physical computational resource, which allows increasing the successful search for solutions of algorithmically unsolvable problems. In our case, the solution of the problem of ensuring global robustness of functioning of the control object under conditions of unexpected control situations by designing the optimal structure and laws of changing the PID controller gain factors by classical control methods is an algorithmically unsolvable problem. The solution of this problem is possible based on quantum soft computing technologies [17]. The output parameters of the PID-regulators are considered as active information-interacting agents, from which the resulting controlling force of the control object is formed. In a multi-agent system, there is a new synergistic effect arising from the exchange of information and knowledge between active agents (swarm synergetic information effect) [17].

Types and operators of quantum genetic algorithms. There are several different types of quantum genetic algorithms. All of them are built on a combination of quantum and classical calculations. Quantum computing includes quantum genetic operators performing genetic operations on quantum chromosomes. These operators are called interference gates.

There are several update operators, but Q-gate interference (rotation) is the most popular [42-44]. The quantum interference operator is denoted as gate $U(t)$:

$$U(t)=\begin{pmatrix}\cos(\delta\theta_j) & -\sin(\delta\theta_j)\\ \sin(\delta\theta_j) & \cos(\delta\theta_j)\end{pmatrix}.$$

Using this operator, the evolution of a population is the result of a process of unitary transformations. In particular, rotations, which approximate the state of chromosomes to the state of the optimal chromosome in the population. The gate enhances or reduces the amplitude of qubits or genes in accordance with the chromosome with the maximum fitness function: $f_i(x_1,x_2,x_3,...,x_j)$ (*maximum*). The best individuals determine the evolution of the quantum state. We considered the quantum genetic algorithm (QGA) [35, 36], the gate of rotations and a quantum gate of mutation and a crossover operator is added to the HGA between them.

Remark. In the classical genetic algorithm (GA), the choice operator mimics Darwinian natural selection, improving populations, promoting individuals with better fitness and "punishing" those who have the worst performance. In the QGA, the choice is replaced by changing all individuals to the best. Therefore, when the rotation operator updates the population, the population converges, but

 DOI: https://doi.org/10.30564/aia.v1i1.619

usually the CGA falls into local optima that undergo premature convergence. To avoid this, the QGA often include either a tape measure or an elite selection. For example, QGA with a selection step is used in an improved K-means clustering algorithm. There are even more extreme approaches, for example, when the QGA includes the selection and simulation algorithm for annealing, precluding premature convergence. In other cases, the selection step is enabled without resorting to the operators commonly used in GA. Such a case of a semiclassical GA (where the selection (choice) operator tends to maximize its suitability through a quantum approach) for example using the Grover algorithm [14, 18].

Quantum mutation operator (inversion). In the GA simulation, there is also a quantum version of the classical mutation operator. The gate performs the inter qubit mutation of the j-th qubit, replacing the amplitudes with Pauli's quantum gate.

Quantum mutation operator (insertion). This gate resembles the biological mechanism for introducing chromosomes. Chromosome insertion means that a chromosome segment has been inserted into an unusual position on the same or a different chromosome. The quantum version of this genetic mechanism involves a permutation or exchange between two randomly selected qubits (left, right). For example, suppose that, given the following chromosome, the first and third qubits are chosen randomly.

The quantum transition operator (classical). A quantum crossover is modeled similar to the classic recombination algorithm used in GA, but it works with probability amplitudes. Although the quantum version of the mutation can be implemented on a quantum computer, there are theoretical reasons that prevent crossover.

Quantum crossover operator (interference). This quantum operator performs crossover by recombination in accordance with a criterion based on drawing diagonals. As a result, all individuals mix with each other, resulting in progeny. Both (QGA and HQGA) quantum algorithms are tested on example of the roots searching task of equation as: $f(x) = |x - \frac{5}{2} + \sin(x)|$.

QGA resulting performance indicates the following (see, Fig. 57).

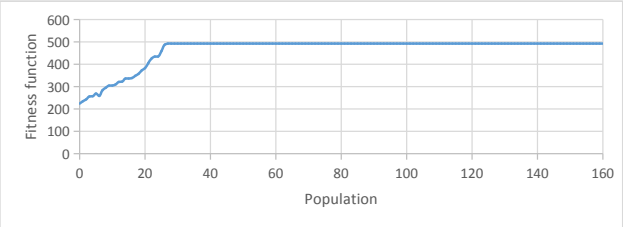


Figure 57. Result of Quantum Genetic Algorithm

It can be seen that after about 30 populations, the value of the fitness function ceases to change. HGA shows the following results (see, Fig. 58).

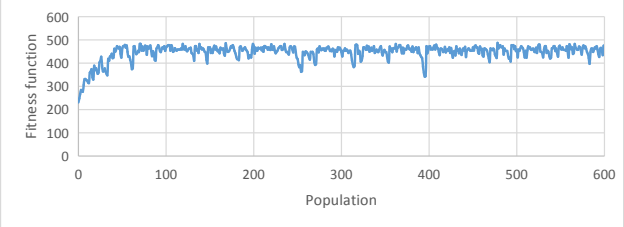


Figure 58. Result of Hybrid Genetic Algorithm

Remark. One of the interesting ideas was proposed in 2004, taking the first steps in implementing the genetic algorithm on a quantum computer [45]. The author proposed this quantum evolutionary algorithm, which can be called the reduced quantum genetic algorithm (RQGA).

The algorithm consists of the following steps: 1) Initialization of the superposition of all possible chromosomes; 2) Evaluation of the fitness function by the operator F; 3) Using Grover's algorithm; 4) Quantum oracle; 5) Using of the diffusion operator Grover G; 6) Make an evaluation of the decision. The search for solutions in RQGA is performed in one operation.

In this case the matrix form is the result of RQGA action as following (see, Fig. 59)

```
[[0.078125]
 [0.078125]
 [0.078125]
 [0.078125]
 [0.078125]
 [0.078125]
 [0.078125]
 [0.078125]
 [0.078125]
 [0.078125]
 [0.078125]
 [0.953125]
 [0.078125]
 [0.078125]
 [0.078125]
 [0.078125]]
```

Figure 59. The Result of the RQGA Algorithm

After action of GQA more than 1000 generation we can see on Fig. 60 that around 70% spatio-temporal correlation have best probability choice.

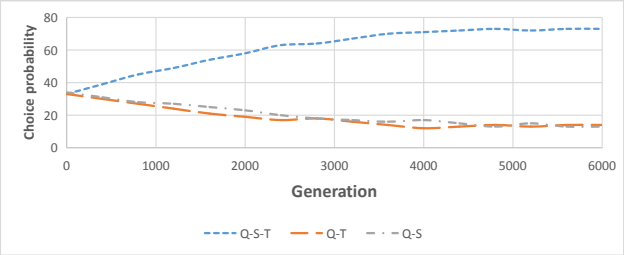


Figure 60. The Result of the QGA

DOI: https://doi.org/10.30564/aia.v1i1.619

Temporal and spatial correlations have similar quality. After 5000 generations probability value is not changing. QGA after 200 generations the probability choice of spatio-temporal correlation decreases to 60% (see, Fig. 61).

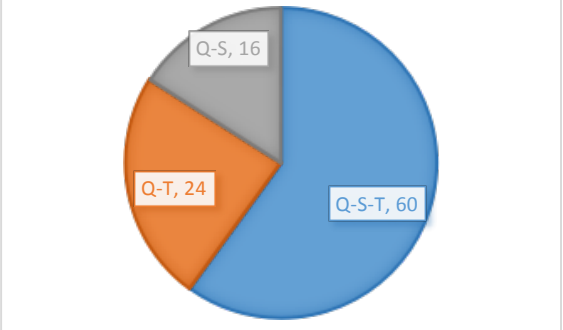


Figure 61. The Result of the Quantum Genetic Algorithm 200 Times

The overall strategy for improving the quality of QGA is to use small improvements in the algorithm. For example, including new operators: "quantum disaster", disturbance, or other customized algorithms [17]. But in many cases, these operators are only useful in highly specific applications [47-53].

Simulator structure and examples of applications

The use of simulators has long been used in various industries: motor racing, aviation, surgery and many others. The development of virtual reality technology and augmented reality adds the ability to create simulators with full immersion.

Remark. In the development of quantum genetic algorithm in this article on the model of the inverted pendulum (autonomous robot) was discovered a few problems. *Firstly*, testing a written algorithm on a robot takes a lot of time. *Secondly*, you may encounter an incorrectly working HW, and it is rather difficult to identify the malfunction itself. *Thirdly*, the GA is the selection of parameters that work best in a particular situation, but it's quite common that these parameters were very bad, which makes it difficult to set up a dynamically unstable object.

Description of the problem. The main goal of the simulator development is SW testing, educational goals, and the ability to observe the pendulum's behavior when using various intelligent control algorithms with different parameters: using only the PID controller, adding a fuzzy controller to the ICS, using the GA and neural network, using QGA. The simulator is interesting because it covers many areas required for its implementation. There are also many different ways of development: improvement of the 2*D* model or even implementation in 3*D*, control of the pendulum in on line (changing the parameters of the pendulum, adding various noises), making the simulator more universal for simply creating simulations of other tasks based on the prepared project.

Selection of development toolkit. Simulator access is as simple as possible and it is implemented as a non-typical web application. The diagram of the sequence of the user's work with the system and the interaction of the model, the presentation and the template are presented on Fig. 62.

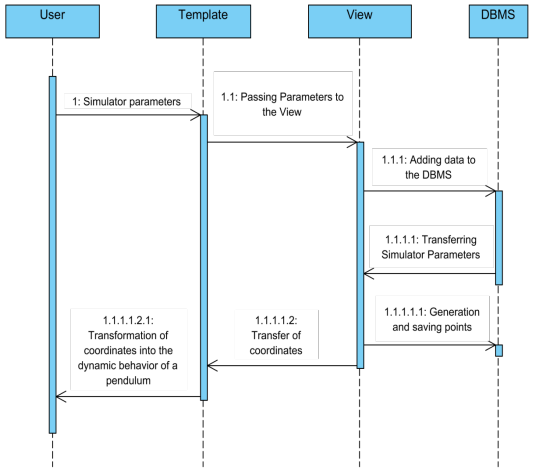
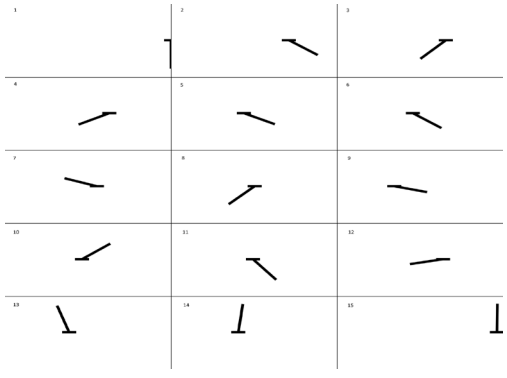


Figure 62. Sequence Diagram of System

Most of the server side work is math. It is necessary to calculate the position of the carriage, the angle of inclination of the pendulum in space. For this reason, Python and the Django framework, which implements the model-view-controller (MVC) approach, were chosen as the programming language (or in Django, this is the model-view-template (MVT)). MySQL is used to store all data, and the architecture has been developed for adding Redis to be faster, if the MySQL operation speed is insufficient.

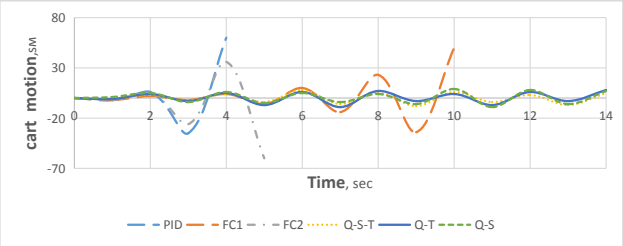
Figure 63 show Benchmark results of quantum intelligent control simulation of "cart - pole" system with QGA (box for the type choice of "*Quantum correlation*" on Fig. 54).

Partial rendering performance of the simulator is shown in Fig. 63.



(a) Visualization of Inverted Pendulum Behavior

 DOI: https://doi.org/10.30564/aia.v1i1.619

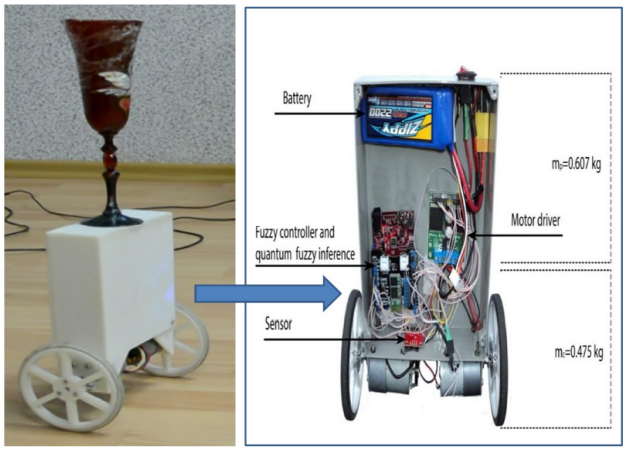


(b) Cart Behavior (Q-S-T – Quantum Spatio-temporal Correlation; Q-T – Quantum Temporal Correlation; Q-S – Quantum Spatial Correlation; FC – Fuzzy Controller)

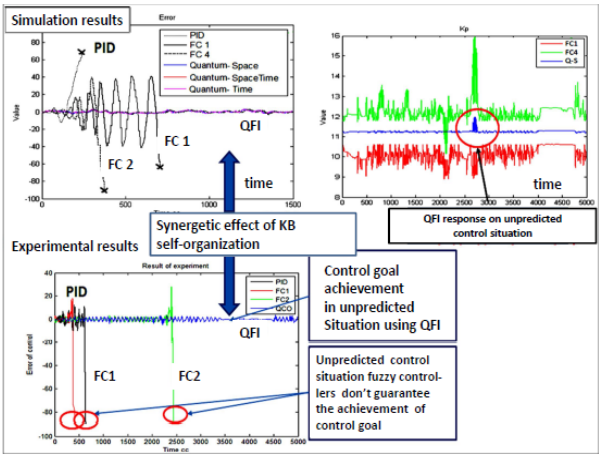
Figure 63. Simulation results of "cart - pole" system with PID - controller, fuzzy controllers and QFI-QGA-controller with different quantum correlation

The described method is differed from [47-53].

Example: *Application of quantum computing optimizer of knowledge base* (QCOPTKB™) *for the case of experimental teaching signal from control object* Control object shown on Fig. 64 a. Structure of robust ICS based on QFI is shown on Fig. 53 and on Fig. 54 is shown QAG structure of QFI that used in the simulation and experiment. On Fig. 64 b are demonstrated results of simulation and experimental results comparison. Mathematical modeling and experimental results are received for the case of unpredicted control situation and knowledge base of fuzzy controller was designing with SW of QCOPTKB™ for teaching signal measured directly from control object (autonomous robot on Fig. 64 a). As model of unpredicted control situation on Fig. 53 (Box Z^{-1}) was the situation of feedback sensor signal delay on three times.



(a)



(b)

Figure 64. Autonomous Robot with Inverted Pendulum (a) and Simulation & Experimental Results Comparison for Unpredicted Control Situation in Cases of PID-controller, Fuzzy Controller and QFI-controller (b)

Results of controllers behavior comparison confirm the existence of synergetic self-organization effect in the design process of robust KB on the base of imperfect (non robust) KB of fuzzy controllers on Fig. 53. In unpredicted control situation control error is dramatically changing and KB responses of fuzzy controllers (FC 1 and FC 2) that designed in learning situations with soft computing are imperfect and do not can achieve the control goal. Using responses of imperfect KB (as control signals for design the schedule of time dependent coefficient gain in PID-controller on Fig. 53) in Box QFI the robust control is formed in on line. This effect is based on the existence of additional information resource that extracted by QFI as quantum information hidden in classical states of control signal as response output of imperfect KB's on new control error (QFI algorithm structure on line in Fig. 56). QGA in Fig. 56 for this case recommended the spatial quantum correlation as was early received in [54, 55].

2. Discussion

The described design method of ICS based on QAG-approach let to achieve global robustness in the case of unpredicted control situations in online using new types of computational intelligence toolkit as quantum and soft computing and based on computational resource of classical computers. The introduced model of QFI is a new type of quantum search algorithm based on sophisticated structure of quantum genetic algorithm embedded in his structure. Such on an approach to the solution of robust control design problems of classical nonlinear control objects

DOI: https://doi.org/10.30564/aia.v1i1.619

(in general globally unstable and essentially nonlinear) is considered as Benchmark for effective application of developed design information technology of ICS [14, 54-58]. The results of simulation and experiment show unconventional (for classical Boolean logic) conclusion: from response of two non-robust imperfect KB of FCs in the structure of ICS on Fig. 53 with new quantum search algorithm QFI possible to design in online robust quantum FC.

With RQGA based on reduced Grover's QSA used spatial quantum correlation between two coefficient gain schedules of FCs in Fig. 53 and quantum self-organization of imperfect KBs in online effectively on classical standard chip realized and described on concrete example.

This synergetic information effect has pure quantum nature, used hidden in classical states quantum information as additional information resource and does not have classical analogy.

3. Conclusions

o New circuit implementation design method of quantum gates for fast classical efficient simulation of QAs is developed. Benchmarks of design application as Grover's QSA and QFI based on QGA demonstrated. Applications of QAG approach in intelligent control systems with quantum self-organization of imperfect knowledge bases are described on concrete examples.

o The results demonstrate the effective application possibility of end-to-end quantum technologies and quantum computational intelligence toolkit based on quantum soft computing for the solution of intractable classical [59] and algorithmically unsolved problems as design of global robustness of ICS in unpredicted control situations and intelligent robotics.

o Efficient simulation on classical computer quantum soft computing algorithms, robust fuzzy control based on quantum genetic (evolutionary) algorithms and quantum fuzzy neural networks (that can realized as modified Grover's QSA), AI-problems as quantum gate simulation approaches and quantum deep learning, quantum optimization in Part II are considered.

o Thus, positive application results of mutual technologies based on soft and quantum computing give the possibility of application Feymann - Manin thesis to study classical physical system as inverse problem "quantum control system – classical control object" solve effectively classical intractable and algorithmic unsolved problems.

References

[1] O. Kyriienko, Quantum inverse iteration algorithm for near-term quantum devices [J]. arXiv:1901.09988v1 [quant-ph], 2019.

[2] L. Gyongyosi, Quantum circuit designs for gate-model quantum computer architectures [J]. arXiv:1803.02460v1 [quant-ph], 2018.

[3] R. M. Parrish, J. T. Iosue, A. Ozaeta, and P. L. McMahon, A Jacobi diagonalization and Anderson acceleration algorithm for variational quantum algorithm parameter optimization [J]. arXiv: 1904.03206v1 [quant-ph],2019.

[4] V. Dunjko and H. J. Briegel, Machine learning & artificial intelligence in the quantum domain: a review of recent progress [J]. Rep. Prog. Phys. 2018, 81(7): 074001 (67pp)
DOI: 10.1088/1361-6633/aab406

[5] C.D. Bruzewicz, J. Chiaverini, R. McConnell, and J. M. Sage, Trapped-ion quantum computing: Progress and challenges [J]. arXiv: 1904.04178v1 [quant-ph], 2019.

[6] V. Bergholm, J. Izaac, M. Schuld et all, PennyLane: Automatic differentiation of hybrid quantum-classical computations [J]. arXiv: 1811.04968v2 [quant-ph], 2019.

[7] K. Bertels, I. Ashraf, R. Nane et all, Quantum computer architecture: Towards full-stack quantum accelerators [J]. arXiv: 1903.09575v1 [quant-ph], 2019.

[8] T. Peng, A. W. Harrow, M. Ozols and X. Wu, Simulating large quantum circuits on a small quantum computer [J]. arXiv: 1904.00102v1 [quant-ph], 2019.

[9] P. Krantz, M. Kjaergaard, F. Yan et all, A Quantum engineer's guide to superconducting qubits [J]. arXiv:1904.06560v1 [quant-ph], 2019.

[10] National Academies of Sciences, Engineering, and Medicine. 2018. Quantum Computing: Progress and Prospects (E. Grumbling and M. Horowitz, Eds) [B]. The National Academies Press, Washington, DC.
DOI: https://doi.org/10.17226/25196.

[11] M.A Nielsen and I.L Chuang, Quantum computation and quantum information [M]. Cambridge University Press, Cambridge, England, 2000.

[12] S. Ulyanov, V. Albu and I. Barchatova, Quantum algorithmic gates: Information analysis & design system in MatLab [M]. LAP Lambert Academic Publishing, Saarbrücken, 2014.

[13] S. Ulyanov, V. Albu and I. Barchatova, Design IT of Quantum Algorithmic Gates: Quantum search algorithm simulation in MatLab [M]. LAP Lambert Academic Publishing, Saarbrücken, 2014.

[14] S.V. Ulyanov, System and method for control using quantum soft computing [P]. US Patent No 7,383,235 B1, 2003; EP PCT 1 083 520 A2, 2001; Efficient simulation system of quantum algorithm gates on classical computer based on fast algorithm [P]. US

 DOI: https://doi.org/10.30564/aia.v1i1.619

Patent No 2006/0224547 A1, 2006.

[15] S. Ulyanov, Method and hardware architecture for controlling a process or for processing data based on quantum soft computing (Inventors: Ulyanov S., Rizzotto G.G., Kurawaki I., Amato P. and Porto D.) [P]. PCT Patent WO 01/67186 A1, 2000.

[16] D. M. Porto and S.V. Ulyanov, Hardware implementation of fast quantum searching algorithms and its application in quantum soft computing and intelligent control [C]. In: Proc. World Automation Congress (5th Intern. Symp. on Soft Computing for Industry). Seville, Spain, 2004 (paper ISSC I31).

[17] S.V.Ulyanov, et al. Quantum information and quantum computational intelligence: Classically efficient simulation of fast quantum algorithms (SW / HW Implementations). [M] Note del Polo, Milan Univ, 2005, 79.

[18] L.K. Grover, A fast quantum mechanical algorithm for database search [P]. US Patent US 6,317,766 B1, 2001.

[19] S. Bose, L. Rallan, V. Vedral, Communication capacity of quantum computation [J]. Phys Rev Lett. 2000, (85): 5448-5451.

[20] E. Arikan, An information-theoretic analysis of Grover's algorithm [J]. arXiv:quant-ph/0210068v2, 2002.

[21] F. Ghisi and S. Ulyanov, The information role of entanglement and interference operators in Shor quantum algorithm gate dynamics [J]. J. of Modern Optics, 2000. 47(12): 2079-2090.

[22] M. Branciforte, A. Calabrò, D. M. Porto, and S.V. Ulyanov, Hardware design of main quantum algorithm operators and application in quantum search algorithm of unstructured large data bases [C]. In: Proc. of the 7th World Multi-Conference on Systems, Cybernetics and Informatics (SCI '2003), Florida, Orlando, USA, 2003.

[23] J. Niwa, K. Matsumoto, H. Imai, General-purpose parallel simulator for quantum computing [J]. Physical Review A, 2002. 66(6).

[24] L. Valiant, Quantum computers that can be simulated classically in polynomial time [C]. In: ACM Proc. STOC'01, Greece, 2001: 114-123.

[25] L. Valiant, Quantum circuits that can be simulated classically in polynomial time [J]. SIAM J. Comput. 2002, 31(4):1229-1254.

[26] C. Huang, M. Newman, and M. Szegedy, Explicit lower bounds on strong quantum simulation [R]. arXiv:1804.10368v2 [quant-ph], 2018.

[27] A. Rybalov, E. Kagan, A. Rapoport and I. Ben-Gal, Fuzzy implementation of qubits operators [J]. Computer Science and Systems Biology. 2014. 7(5): 163-168.
DOI:10.4172/jcsb.1000151

[28] T.M. Forcer, et all Superposition, entanglement and quantum computation [J]. Quantum Information and Computation, 2002, 2(2): 97-116.

[29] J. Pilch, J. Długopolski, An FPGA-based real quantum computer emulatorю [J]. J. of Computational Electronics, 2018. available: https://doi.org/10.1007/s10825-018-1287-5

[30] A.J. McCaskey, E.F. Dumitrescu, D. Liakh, M. Chen, W. Feng, T.S. Humble, A language and hardware independent approach to quantum–classical computing [J]. Software X 7, 2018, 2: 245-254.

[31] A.R. Colm, R.J. Blake R., B.D. Diego and T. A. Ohki, Hardware for dynamic quantum computing [R]. arXiv:1704.08314v1 [quant-ph], 2017.

[31] Zeng-Bing Chen, Quantum Neural network and soft quantum computing [R]. arXiv:1810.05025v1 [quant-ph], 2018.

[32] J. B. Vega, D. Hangleiter, M. Schwarz, R. Raussendorf, and J. Eisert, Architectures for quantum simulation showing a quantum speedup [J]. Physical Review, 2018, X 8: 021010.

[33] A.M. Childs, D. Maslov, Y. Nam, N. J. Ross and Y. Su, Toward the first quantum simulation with quantum speedup [J]. PNAS, 2018. 115(38): 9456-9461 https://doi.org/10.1073/pnas.1801723115

[34] K. Michielsen, M. Nocon, D. Willsch, F. Jin, T.Lippert, H. De Raedt, Benchmarking gate-based quantum computers [J.] . Computer Physics Communication, 2017, 220: 44-55. http://dx.doi.org/10.1016/j.cpc.2017.06.011

[35] Patrick J. Coles et all, Quantum algorithm implementations for beginners [R]. arXiv:1804.03719v1 [cs. ET], 2018

[36] K. A. Britt, F. A. Mohiyaddin, and T. S. Humble, Quantum accelerators for high-performance computing systems [R]. arXiv:1712.01423v1 [quant-ph], 2017.

[37] Zhao-Yun Chen and Guo-Ping Guo [R], QRunes: High-level language for quantum-classical hybrid programming [R]. arXiv:1901.08340v1 [quant-ph], 2019.

[38] Y. H. Lee, M. Khalil-Hani, and M. N.Marsono, An FPGA-based quantum computing emulation framework based on serial-parallel architecture [J]. Intern. J. of Reconfigurable Computing, 2016. Vol. 2016, Article, ID: 5718124. http://dx.doi.org/10.1155/2016/5718124

[39] F. K. Wilhelm et all, Entwicklungsstand Quantencomputer [M]. Federal Office for Information Security. Bonn, 2017.

DOI: https://doi.org/10.30564/aia.v1i1.619

https://www.bsi.bund.de

[40] G. D. Paparo, V. Dunjko, A Makmal, M. A. Martin-Delgad, and H. J. Briegel [J], Quantum speedup for active learning agents. Pysical Review, 2014, X 4: 031002.

[41] Yi-Lin Ju, I-Ming Tsai, and Sy-Yen Kuo, Quantum circuit design and analysis for database search applications [J]. IEEE Trans. On Circuits and Systems. 2007, 54(11): 2552-2563.

[42] M. Suchara, Y. Alexeev, F. Chong, H. Finkel, H. Hoffmann, J. Larson, J. Osborn, and G. Smith, Hybrid quantum-classical computing architectures [C]. In: Proc. 3rd INTERN. WORKSHOP ON POST-MOORE'S ERA SUPERCOMPUTING (PMES), PMES Workshop, Dallas, 2018. http://j.mp/pmes18

[43] D. Koch, L. Wessing, P. M. Alsing, Introduction to coding quantum algorithms: A tutorial series using Qiskit [R]. arXiv:1903.04359v1 [quant-ph]. 2019

[44] S.V. Ulyanov, Quantum soft computing in control processes design: Quantum genetic algorithms and quantum neural network approaches [C]. In: Proc. WAC (ISSCI') 2004 (5th Intern. Symp. on Soft Computing for Industry), Seville Spain, 2004, 17: 99-104.

[45] S.V. Ulyanov, Self-organizing quantum robust control methods and systems for situations with uncertainty and risk [P]. Patent US 8788450 B2, 2014.

[46] P. Chandra Shill, F. Amin, K. Murase, Parameter optimization based on quantum genetic algorithms for fuzzy logic controller [R]. Department of System Design Engineering University of Fukui, 3-9-1 Bunkyo, Fukui-910-8507, 2011.

[47] P. Chandra Shill, B. Sarker, M. Chowdhury Urmi, K. Murase. Quantum fuzzy controller for inverted pendulum system based on quantum genetic optimization [J] Intern. J. of Advanced Research in Computer Science, 2012.

[48] R. Lahoz-Beltra, Quantum genetic algorithms for computer scientists [J]. Computers. – 2016, 5: 24.

[49] Cheng-Wen Lee, Bing-Yi Lin. Applications of the chaotic quantum genetic algorithm with support vector regression in load forecasting [M], Switzerland, 2017.

[50] A. Arjmandzadeh, M. Yarahmadi, Quantum genetic learning control of quantum ensembles with Hamiltonian uncertainties [R], Switzerland, 2017.

[51] H. Wang, J. Liu, J. Zhi. The improvement of quantum genetic algorithm and its application on function optimization [R]. College of Field Engineering, PLA University of Science and Technology, Nanjing 210007, China, 2013.

[52] A. Malossini, E. Blanzieri, T. Calarco, QGA: A quantum genetic algorithm [R]. Technical Report # DIT-04-105. – University of Toronto, 2004.

[53] S.V. Ulyanov, K. Takahashi, L.V. Litvintseva and T. Hagiwara, Design of self-organized intelligent control systems based on quantum fuzzy inference: Intelligent system of systems engineering approach [C]. Proc. of IEEE Intern. Conf. SMC' , Hawaii, USA, 2005, 4: 3835- 3840.

[54] L.V. Litvintseva and S.V. Ulyanov, Quantum fuzzy inference for knowledge base design in robust intelligent controllers [J]. J. of Computer and Systems Sciences Intern. 2007, 46(6): 908-961.

[55] S.V. Ulyanov, K. Takahashi, G.G. Rizzotto and I. Kurawaki, Quantum soft computing: Quantum global optimization and quantum learning processes – Application in AI, informatics and intelligent control processes [C]. In: Proc. of the 7th World Multi-Conference on Systems, Cybernetics and Informatics, (SCI '2003), Florida, Orlando, USA, 2003.

[56] S. Ulyanov, F. Ghisi, V. Ulyanov, I. Kurawaki and L. Litvintseva [M] Simulation of Quantum Algorithms on Classical Computers, Universita degli Studi di Milano, Polo Didattico e di Ricerca di Crema, Note del Polo, 2000, 32.

[57] D.M. Porto, S.V. Ulyanov, K. Takahashi, and I.S. Ulyanov, Hardware implementation of fast quantum searching algorithms and its applications in quantum soft computing and intelligent control [C]. Proc. Word Automation Congress (WAC'2004), Seville, Spain, 2004.

[58] Ch. H. Papadimitriou and J. Tsitsiklis, Intractable problems in control theory [J]. SIAM J. Control and Optimization, 1986. 24 (4): 639-654.

 DOI: https://doi.org/10.30564/aia.v1i1.619