Quantum Circuit Automatic Synthesizer (QCAS)

Ali Moghadaszadeh, Majid Mohamadi, Ali Akbar Niknafs, Peyman Keshavarzian

***Abstract*— Quantum and reversible circuit synthesis has been important concern of designers in recently passed decades. Considering the application of garbage bit in optimization of quantum circuit, GA-Based techniques have been introduced. This project provides an extensible infrastructure for quantum circuit automatic synthesis and optimization with the enhancement of interoperability features. The problem domain begins from complex computing model up to a synthesis of reversible quantum circuit. Optimization criterias are whole considered separately from software processing logic which cause capability of applying variable optimization critria in different problem domains.**

The new introduced methodology utilize PSO technique for circuit synthesis evolutionary computation in optimization step. A software library has been developed which implements this technique.

It provides a simple flexible software for researchers in order to automatic synthesis of quantum circuit that implements an open source library of prerequisite of quantum circuit synthesis.

Index Terms-Quantum circuits, gate, PSO, Automatic synthesis.

I. INTRODUCTION

Synthesis of reversible logic circuit is very different from classical logic circuit. The output capacity of reversible gate is supposed to be one [1]. The common gates like "AND" and "OR" cannot be applied in reversible circuit synthesis process because of irreversible nature.

Considering reversible characteristics of quantum circuits, synthesis process differs from classical and reversible logic. The reasons are variety of quantum gates, distinct representation model and properties like entanglement or superposition in quantum circuits [2].

Therefore, the synthesis methods of reversible circuits are used only for a certain set of quantum circuits. It is possible to use insignificant values in corresponding input and output vectors of truth table to optimize the circuit [3].

Extra inputs and garbage outputs which are insignificant values for result function may cause a simpler circuit with lower synthesis cost [4].

Circuit optimization is considered with different aspects, e.g. criteria such as quantum cost, number of constant inputs and garbage outputs and the distance between control inputs and goal output are used for comparing reversible and quantum circuits. The major objective is to provide a method that can automatically find the best way to optimize different aspects of reversible and quantum circuits [5]- [7].

Manuscript received April, 2013.

Ali Moghadaszadeh, Computer Engineering Department, Science and Research branch, Islamic Azad University, Kerman, Iran

Majid Mohamadi, Computer Engineering Department, Shahid Bahonar University of Kerman, Kerman, Iran.

Ali Akbar Niknafs, Computer Engineering Department, Shahid Bahonar University of Kerman, Kerman, Iran.

Peyman Keshavarzian, Computer Engineering Department, Science and Research branch, Islamic Azad University, Kerman, Iran.

The rest of the paper is organized as follows: Section II introduces QCAS. Section III presents Experimental Results Finally, section IV concludes the paper.

II. QCAS

We will address the problem of quantum circuit automatic synthesis given a desired truth table. The automatic circuit synthesis problem is considered as a computing problem of optimization and quantum computation.

The framework structure represented in Fig.1, includes 5 main components named "Process Controller", "PSO Actor", "QCL", "Model Convertor" and "Visualizer". We will explain each Component in detailed below.

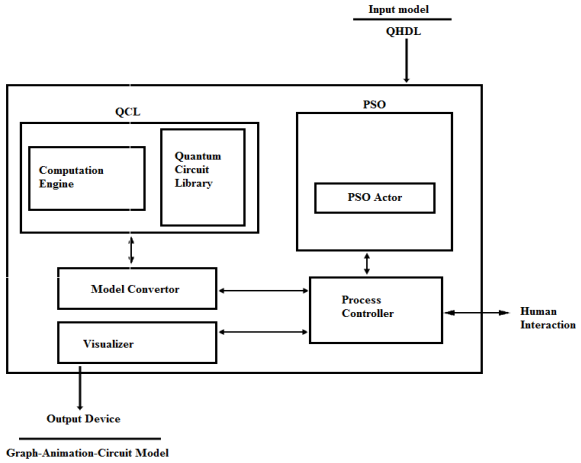

Fig.1: The general structure of QCAS software

A. Process Controller

Process Controller agent manages whole process of quantum circuit synthesis. It provides initial states for other actors in the system. Initial variables like process time, maximum iteration and maximum inertia are initiated in this module.

Controller adapts process parameters by back-tracking and assigning values to some parameters in stochastic manner. It dispatches events and information from PSO Actors to Visualizer. It also prepares input model from a user-defined truth table to a reversible by adding don't care values.

B. PSO Actor

This component provides functions and classes which are used to apply PSO algorithm on automatic quantum circuit synthesis problem in order to optimize the result.

With regards to comprehensive features of this actor, it is possible to utilize software library of classes developed in this section for other problem domains. The application technique of PSO Actor in a quantum circuit synthesis is depicted in Fig.2 as an abstract model.

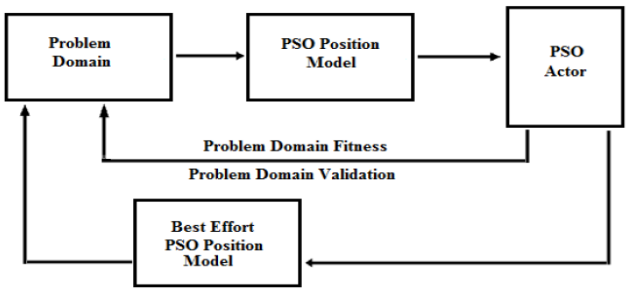

Fig.2: PSO employment in solution

Synthesis of classical reversible or quantum circuit with application of PSO technique requires a coding approach which represents gate positions and states on a circuit [8].
As reversible circuits, quantum circuits have equal inputs and outputs. So we considered plane and section representation method (Fig.3) for coding a quantum circuit to particle positions in PSO.

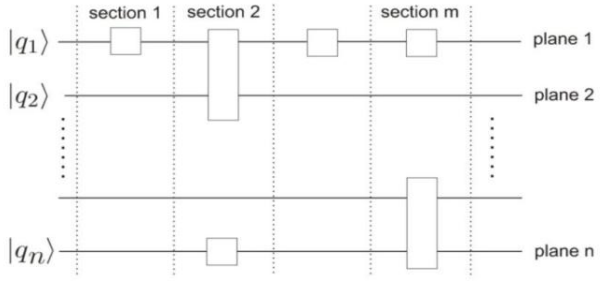

Fig.3: Display model of Plane and Section for quantum circuits

A gate position is converted to a particle position by structure which is shown in Fig.4, each gate input is mapped to a plane. The gate kind is obtained by an integer number that points to that gate in the possible gate collection list.

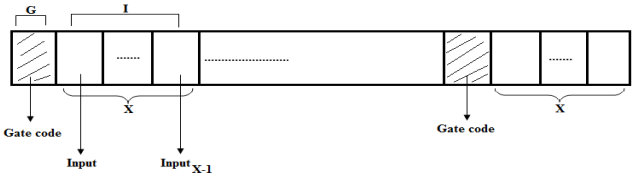

Fig.4: Coding of circuits into particles position in PSO method

In (1), maximum allowed gate count named GL with maximum gate input named X, we obtain GL * (X+1) dimensions for particle position space.

$$n=(X+1)\times GL,\quad X=\max(\text{InputCount}(G)),\quad GL=\text{GateLevel} \tag{1}$$

Remember the PSO workflow [9], the velocity of particles are updated as (2).

$$V_{(p)} = K \times [\, V(p) \times D(w_i) + [\, (\text{Best}(\text{position}(p)) - \text{position}(p)) \times CA \times D(r_1)\,] + [\, (\text{Best}(\text{position}(i,p)) - \text{position}(p)) \times SA \times D(r_2)\,]\,] \tag{2}$$

i = Iteration , r_1 = Random , SA = Social Acceleration, P =Particle

D = Discret Probability Function , CA = Cognitive Acceleration

Like GA-Based techniques in quantum circuit synthesizing [10] by employing mutation operator to change position of a particle in one dimension suddenly, we can improve particle chance to see new positions.
We also used a velocity tuner called VTU that improves algorithm performance by reduction of invalid space which particles meet them. It prevents particles to move out from PSO search space boundary.
Some positions where particles moving to, are not a valid position in PSO search space because of the impossibility of realization e.g. a position like “1,1,1,1” is invalid position because all input of gate is located on a same Plane which its index is “1”.
Fitness function determination is a challenging problem and mostly related to optimization criteria.
We applied an object oriented fitting technique. Events called “OnMeasure” and “OnFit” are raised when PSO Actor needs to evaluate each particle position in order to update internal variables. Events are dispatched to evaluators which are assigned to this process.
External SDK developments can easily participate in evaluating metric and fitness of particles that are indeed coded circuits.
Here Metric structure is introduced to propel process to an optimized space, e.g. Target Control Distance (TCD) could be considered as metric.
“OnFit” causes more fitted result and “OnMeasure” directs to best metrics. More fitted result means the result which it is possible circuit that obeys the preferred input and output model such as zero hamming distance of circuit output with expected one.

C. *QCL*

Quantum Circuit computing Logic which is briefly called “QCL” is main computing agent to evaluate output function of circuit as sequence of quantum gate. QCL has two level computations, one is considered as Gate Level and the other one as circuit level (plane level). Gate Level computation is handled by internal corresponding matrix in each gate that effectively computes outputs of a quantum gate after applying a custom input state. Second level computation (plane level) affects the gate outputs on total circuit state in a specific index in sequence. Each quantum state is represented by (3):

$$\varphi = \alpha.|0\rangle + \beta.|1\rangle \tag{3}$$

$$|\alpha^2| + |\beta^2| = 1$$

Since QCL shall be able to compute φ, so we consider “QS” as:

```
Struct QS
    {
        AlfaReal
        AlfaComplex
        BetaReal
        BetaImaginary
    }
```

Circuit as sequence of quantum gate maintains QS for each qubit and takes aware of gate effect at sequence stages.
By affecting first level computation, new qubit states are obtained, it

goes on till output function to be computed, as described in (4), circuit state is initiated by an initial quantum state and last to final quantum state.

(4)

$$QS_{initial}(n,double[2^n],double[2^n]) \rightarrow QS(n,double[2^n],double[2^n])$$

"n" is count of plane in representing model. A quantum circuit with "n" plane requires a matrix by 2^n rows, in order to describe circuit state in "kth" stage of computation.
Quantum states are represented by a matrix like (5):

$$P = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_{2^n-1} \end{pmatrix} \quad (5)$$

$a_j = x + yi \qquad x:Matrix(2^n,1) \qquad y:Matrix(2^n,1)$

QCL detaches "P" into two different matrixes to simplify complex computation of $\alpha_0,\ldots,\alpha_{2^n-1}$, that hides symbol "i" in preserving complex matrixes.
"P" has a growth function of (Ω (2^n)), if n=2 or n=3 the matrix is in dimension of 4 or 8 but for n=32 it would be 2^{32}, that is massive to store and computing that results to lower performance.
It suffers from resource consumption problem. To overcome this problem. We consider only significant states which are distinguished by none-zero value in imaginary or real part rows of matrixes.

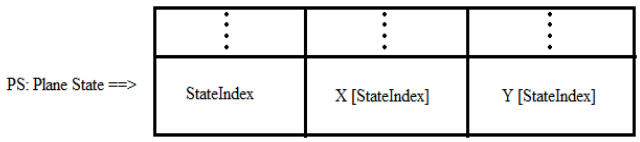

Fig.5: QCL data structure used to store the quantum circuit

If quantum circuit is simultaneously in "η" different quantum state ($0<\eta<2^n$) only "η" records will be maintained in structure that illustrated in Fig.5. We should consider each section (gate) changes Plane-State from an elementary state to a secondary state [11] as (6).

$$PS[\eta] \xrightarrow{Section[K]} PS[\delta] \quad (6)$$

"δ" is count of new quantum states of the circuit to be synthesized. Considering a Plane-State named State-Index, inputs of "Kth" gate are in state that computed by (7):

(7)

$$GateMapIndex = \sum_{i=1}^{i=j} 2^{j-i} \times [\,2^{GateMap[i]} \wedge StateIndex]$$

$j = GateInput(k)$

$$[\,V\,]^u = \begin{cases} 0 & \text{if } (v=0) \\ 1 & \text{if } (v\neq 0) \end{cases}$$

Obtaining gate result matrix, Plane-States are found by calculative mapping of State-Indexes in "Kth" gate to circuit Plane State. New Plane-States are achieved by repeating computations in (8), for each output state of specific gate in stage K.

(8)

$$MaxGateState = \sum_{i=1}^{i=j=GateInput(k)} 2^{j-i}$$

$$PlaneState = PlaneState \wedge (\sim MaxGateState)$$

D. *Model convertor*

This component is in role of adapting PSO Actor modeling domain with QCL modeling domain.

E. *Visualizer*

Running activities in other section are illustrated by visualizer, it includes 3 main parts:
Process visualizer which represents the particle position in 2D graph. Scaling the dimensions, coloring particles and position is another task of this module.
A visualizer module which shows controller state in synthesis process.
A visualizer module that shows result circuit with standard symbols.

III. EXPERIMENTAL RESULTS

Using QCAS library we had setup an experiment. Experiment environment parameters are according to Table.1.

Table.1: Experiment Setup Parameters

Symbol	Abbreviation	Value
INR	Inertia	0.7*
MIR	Maximum Iteration	5000
MME	Maximum Metric Error	0.01*
IGL	Initial Gate Level	1*
MPE	Maximum Plane Exchange	5*
MPT	Maximum Process Time	2100(S)
MPR	Mutation Probability	0.7*
DAP	Dynamic Adaptive Parameter	True
PGS	Probilistic Gate Selection	True
DME	Default Metric Evaluation	Cost↓ CTD↓ CCD↓
VTA	VTU Activation	True
* Means adaptive variables		

A one-bit full adder was synthesized and validated by QCL. The two final results are shown in Figs.6,7 by visualizer.

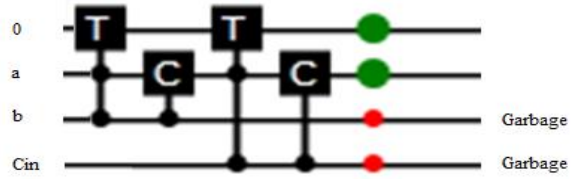

Fig.6: One-bit full adder (reversible gates)

Result (Fitness=1, Metrics=10,8,3)
It is improved by quantum cost,TCD and CCD in Fig.7

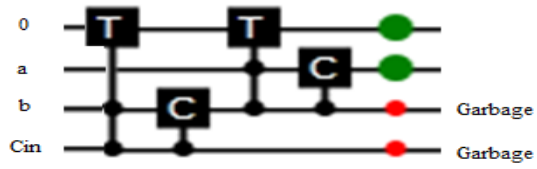

Fig.7: Improved One-bit full adder (reversible gates)
Result (Fitness=1, Metrics=10,8,2)

Metric selector in PSO Actor, selects best metric by ordering logic specified in initial parameter definition table (Table.1). Metrics were considered as quantum cost, TCD (Target Control Distance) and CCD (Control-Control Distance). Results with less metric measures are obviously better results. Fig.7 shows circuit which its metrics are improved in comparison to Fig.6 by reduction of CCD.

The same circuit in Figs.6,7 is synthesized by quantum gates in Fig.8.

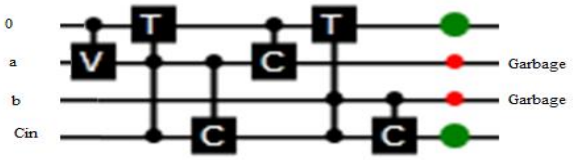

Fig.8: One-bit full adder (quantum gate)

It is improved by quantum cost,TCD and CCD in Fig.9

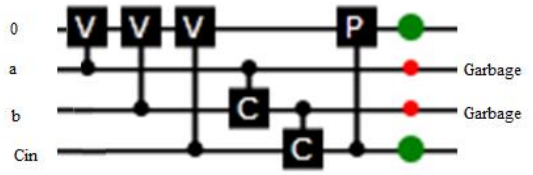

Fig.9: Improved One-bit full adder (quantum gate)

PSO particles may meet PSO search space positions more than once and also some of them are inherently invalid positions that could not represent a circuit by QCL. mAs Fig.10 shows, duplication ratio is increased when Gate Level growth. Reversely valid space is decreased by Gate Level growth.

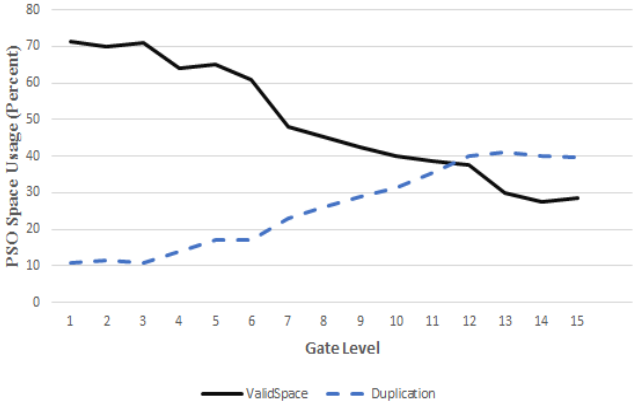

Fig.10: PSO space usage

Applying VTU (Velocity Tuner) the average valid space ratio is improved and almost being constant by Gate Level growth (Fig.11).

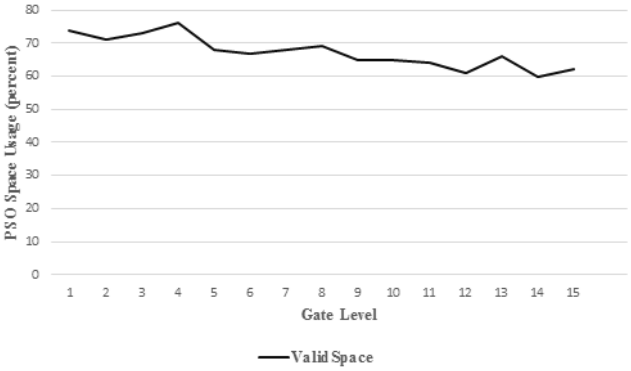

Fig.11: PSO valid space usage

According to Fig.12, VTU causes a considerable reduction of algorithm response time.

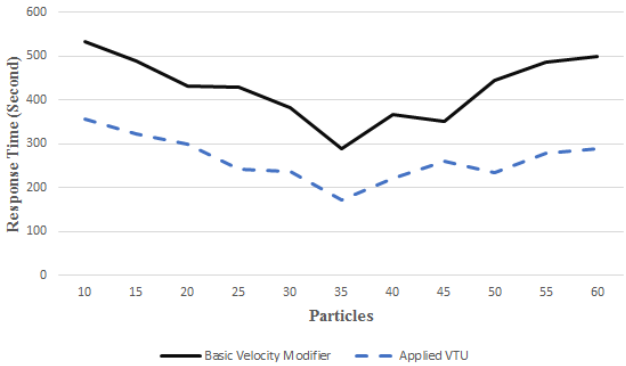

Fig.12: Response time

VTU affects the performance of algorithm by decreasing about 52% of the iteration count that illustrated in Fig.13.

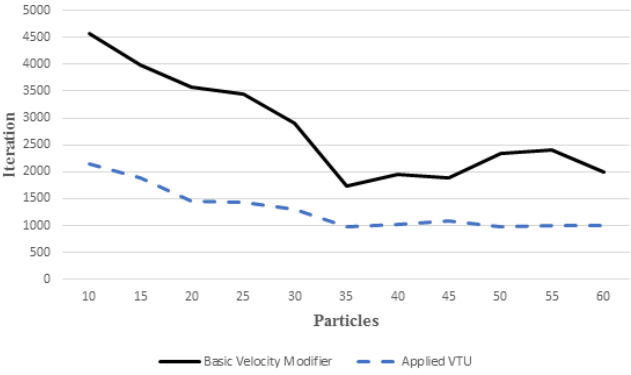

Fig.13: Iteration

IV. CONCLUSION

In this paper we explained synthesis and optimization of reversible quantum circuits in a logical manner. We detach synthesis problem into logical operational agents. The process of synthesizing is handled by agent interactions, although GA_Based methods have been already applied for quantum and mostly reversible circuit automatic synthesis, we utilized PSO algorithm.

A flexible software framework developed for managing interaction of individual modules in synthesis process also an object oriented view of optimization
criteria was employed which makes it possible to take advantage of physical quantum simulators in case of metrics in finding optimum result. Metric objects are passed to evaluators by the internal event dispatcher in controller agent then forwarded to optimizer agent in order to be applied in heuristic search computation. Dynamic fitness evaluators, no-limited metrics and various gate definition capabilities are important features of this framework.
Controller module tunes adaptive parameters of synthesis in each cycle of iterations and uses back-track operation, so it strengthens the optimizer module by providing different configuration and initialization at start time.
VTU (Velocity Tuner) cause a considerable improvement in algorithm performance and expected response time.

REFERENCES

[1] Miller, D.M., Dueck, G.W., Maslov, D., "A Transformation Based Algorithm for Reversible Logic Synthesis", Proceedings of the 40th Design Automation Conference, Anaheim, CA, pp. 318–323, 2003.
[2] Peres, A.: "Reversible Logic and Quantum Computers", Physical Review, A 32: 3266–3276, 1985.
[3] Prathibha, T.R., Bhagyalakshmi, H.R., Venkatesha, M.K., "Optimized Reversible carry select BCD adders using Reversible logic gates", Computer Technology & Applications,Vol 3 (4), 1550-1556, IJCTA, July-August 2012.
[4] Mohammadi, M., Eshghi, M., "Heuristic Methods to use don't cares in automated design of reversible and quantum logic circuits", Quantum Information Processing , Springer, Volume 7, Issue 4 (August 2008) pp.175 - 192.
[5] Donald, J., JHA, N., "Reversible Logic Synthesis with Fredkin and Peres Gates", ACM Journal on Emerging Technologies in Computing Systems, Vol. 4,No. 1, Article 2, March 2008.
[6] Maslov, D., Dueck, G.W., Miller, M., Negrevergne, C., "Quantum Circuit Simplification and Level Compaction", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 27(3):436–444, March 2008.
[7] Wille, R., Drechsler R., "BDD-based Synthesis of Reversible Logic for Large Functions", DAC, San Francisco, California, USA, July 26-31, 2009.
[8] Ogri. J., Etim, O., prosper, I., " Optimising digital combinational circuit using particle swarm optimisation technique", Lat. Am. J. Phys. Educ. Vol. 6, No. 1, March 2012.
[9] Zhong, Y., Shuzhi, N., "An Improved Particle Swarm Optimization Algorithm based on Membrane Structure", IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No 2, 2013.
[10] Ruican, C., Udrescu, M., Prodan, L., Vladutiu, M., "Genetic Algorithm Based Quantum Circuit Synthesis with Adaptive Parameters", CEC'09 "IEEE Congress on Evolutionary Computation", Trondheim, Norway, May 2009, pp. 896-903, ISBN 978-1-4244-2959-2.
[11] Patel, K., Markov, I., Hayes, P., "Optimal Synthesis Of Linear Reversible Circuits", Quantum Information and Computation, Vol. 8, No. 3&4 (2008) 0282–0294.