YIDDISH Vol 10 Issue 02,June 2020
ISSN NO: 0364-4308

An evolutionary algorithm based on quantum swarm theory for mining associations in huge datasets

Dr. D SRUJAN CHANDRA REDDY[1], AKSHAYAM PRASMITA[2],
PROFESSOR[1], ASSISTANT PROFESSOR[2],
DEPARTMENT OF CSE
PBR VISVODAYA INSTITUTE OF TECHNOLOGY AND SCIENCE::KAVALI

Abstract

Association rule mining seeks to extract the fundamental cause structure between a set of commonly recurring items or traits in a database. These relationships are represented by rules. Association rule mining provides a strong, non-linear method of associations. The search for link rules is an NP-complete problem. Most of the challenges arise when trying to exploit a very large number of database transactions and objects. In this article, we propose a new approach to efficiently deriving the most applicable principles without obligatorily ensuring optimal responses in all circumstances. The innovative derived algorithm is created using the Quantum Swarm Evolutionary technique, which yields better results than genetic algorithms.

Introduction

As a result of their efficacy and simplicity, data mining techniques like association rule mining (Agra- wall et al., 1993a,b) are becoming increasingly widespread. Methods for learning by associating data points offer a method that is both non-linear and durable for discovering causative structures among groups of frequently-used objects or characteristics in a database. Apriorism and similar association rule algorithms (Agrawal et al., 1993a,b) look at a large set of transactions to identify which products are commonly bought together. Database science, machine learning, and optimization studies all come together to provide cutting-edge clever solutions to the problem of finding relationship patterns from data. Anguilla et al. (2001) demonstrated that the methods for association rule mining are NP-complete by showing that the problem can be simplified to the NP-complete problem of finding a CLIQUE in a network. Exploiting a massive number of things and database interactions is where most of the difficulties emerge. There are a wide variety of algorithms for mining association rules, but they can be broken down into two main categories: (1) Exact algorithms, like Apriorism (Agrawal et al., 1993a,b) and FP-Growth, and (2) approximate algorithms. (Pei et al., 2000). These algorithms ensure the best possible answer, regardless of how long it takes to get there. (2) Evolutionary algorithms (Lopes et al., 1999; Melba and El-Ghazali, 2000), which provide acceptable solutions that are not necessarily optimum but can be executed in a sensible amount of time (polynomial).Association rule mining in big datasets is a complex process that requires the use of precise methods that can be prohibitively costly. Our research suggests that evolving computing can be of great assistance in this field. In this piece, we tackle the challenge of mining association rules with a Quantum Swarm Evolutionary Algorithm (QSE) (Wang et al., 2006). QSE combines the best features of the Quantum Evolutionary Algorithm (QEA; Han and Kim, 2002) and the particle swarm optimization (PSO; Kennedy and Eberhart, 1995) to create a powerful optimization tool.When compared to traditional evolving algorithms like the genetic algorithm, QEA's use of a Q-bit as the tiniest unit of information in its probability depiction makes it far superior. Multiple Q-bits refer to a series of Q-bits used to identify a single entity. One distinct benefit of the Q-bit individual is its ability to probabilistically describe a linear combination of states (binary answers) in the search space. This means that when compared to the chromosome representation used in a genetic program, the Q-bit representation provides a more accurate picture of population variety. To encourage individuals to work together toward a common goal, we can think of a Q-gate as a variant operator of QEA that pushes them to improve their answers. To autonomously adjust the Q-bit of QEA, QSE (Wang et al., 2006) uses a new quantum bit expression method dubbed quantum angle and adopts the enhanced PSO. QSE is superior to QEA, as demonstrated by Wang et al. (2006).

Here's how the rest of the story is laid out:

In Section 2, the foundations of mining for associations are laid out. We provide an overview of quantum computing and particle swarm optimization in Section 3. We introduce a novel method for mining association principles in Section 4. Our testing findings are presented in Section 5.

Association rule mining

Association rule mining is formally defined as follows. Let I ¼fi1 ; i2 ; ...; I'm g be a set of Boolean attributes called items and S ¼fs1 ; s2 ; ...; son g be a multi-set of records representing data instances or transactions, where each record or data in- stance is 2S is constituted from the non-repeatable attributes from I. The presence of a Boolean attribute in a data instance s I means that its value is 1, if it is absent, its value is set to 0. For example, let I ¼fA; B; Cg be a set of Boolean attributes

and let S ¼fhA; Bi; chip; chi be a multi-set of data instances,

the multi-set S can be rewritten as follows:

$$S = \{\langle A = 1, B = 1, C = 0\rangle, \langle A = 0, B = 0, C = 1\rangle,$$
$$\langle A = 0, B = 0, C = 1\rangle\}$$

For categorical attribute, instead of having one attribute in I, we have as many attributes as the number of attribute values. For example, the more general multi-set of data instances S

go-Ven by:

{⟨height-166 = 1, height-170 = 0, height-174 = 0, gender-male = 0, gender-female = 1⟩,
⟨height-166 = 0, height-170 = 1, height-174 = 0, gender-male = 1, gender-female = 0⟩,

Æheight-166 = 0, height-170 = 0, height-174 = 1, gender- male = 0, gender-female = 1æ}is intended to abstract a multi-set of three data instances have- Ing two categorical attributes: height and gender. The values of (height, gender) are {(166, female), (170, male), (174, female)},

respectively.An association rule is denoted by IF C THEN P when C states for Condition(s) and P for Prediction(s) where C, P I and C \ P= B.In this article we are particularly interested by the conjunct- tie association rules where C is a conjunction of one or more condition(s) and P is also a conjunction of one or more predict- ton(s). The following notations are used in the remainder of the

article:

- $|C|$: The number of data instances which are covered by (i.e. satisfying) the *C* part of the rule.
- $|P|$: The number of data instances which are covered by the *P* part of the rule.
- $|C\&P|$: The number of data instances which are covered by both the *C* part and the *P* part of the rule.
- *N*: The total number of data instances being mined.

The confidence *b* of a rule is the probability of the occurrence of *P* knowing that *C* is observed; *b* is equal to $\frac{|C\&P|}{|C|}$. The prediction frequency *a* is equal to $\frac{|P|}{N}$. Note that the support is equal to the fraction $\frac{|C\&P|}{|S|}$.

Fitness function

The quality of a candidate rule is evaluated by means of a fit- ness function. Several fitness functions have been defined in the literature (Agrawal et al., 1993a,b; Lopes et al., 1999). They can be basic or complex. An example of a basic function is the support of a rule (the percentage of data instances satisfy- Ing the C part of the rule) and the confidence facto(the percentage of data instances satisfying the implication IF C THEN P). It is claimed that such basic fitness function is not sufficient. In this article we adopt the complex fitness function of Lopes et al. (1999). This function isderivedfrom.

information theory and it is based on J-measure JM given by:

$$J_m = \frac{|C|}{N} * \left(b * \log \left(\frac{b}{a} \right) \right)$$

The fitness function F is the following:

$$F = \frac{w_1 * (J_m) + w_2 * \left(\frac{n_{pu}}{n_T} \right)}{w_1 + w_2}$$

where input is the number of potentially useful attributes. A given attribute A is said to be potentially useful if there is at least one data instance having both the A's value specified in the part C and the prediction attribute(s). The term is not the total number of attributes in the part C of the rule; w1, w2 are userdefined weights set to 0.6 and 0.4, respectively.

Quantum computing and particle swarm optimization

Quantum computing (QC) is an emergent field calling upon sav- earl specialties: physics, engineering, chemistry, computer sci- once and mathematics. QC uses the specificities of quantummechanics for processing and transformation of data stored in two-state quantum bits or Q-bit(s) for short. A Q-bit can take state value 0, 1 or a superposition of the two states at the same time. The state of a Q-bit can be represented as We = aŒ0æ + bŒ1æ where a and b are the amplitudes of Œ0æ and Œ1æ, respect- timely, in this state. When we measure this Q-bit, we see Œ 0æ with probability ŒaŒ2, and Œ1æ with probability ŒbŒ2 such that.

A 2 + ŒbŒ2 =1. The idea of superposition makes it possible to represent an exponential whole of states with a small number of Q-bits. According to the quantum laws like interference, the linearity of quantum operations makes the quantum computing more powerful than the classical machines.

In order to exploit effectively the power of quantum com-putting, it is necessary to create efficient quantum algorithms. A quantum algorithm consists in applying a succession of quantum operations on quantum systems. Shor (1994) demon-started that QC could solve efficiently NP-complete problems by describing a polynomial time quantum algorithm for factor- Ingnumbers. One of the most known algorithms is Quantum-inspiredEvolutionary Algorithm (QEA) (Han and Kim, 2002), which is inspired by the concept of quantum computing. This algorithm has been first used to solve knapsack problem (Han and Kim, 2002) and then it has first used to solve different NP-complete problems like Traveling Salesman Problem (Talbi et al., 2004) and Multiple Sequence Alignment (Layer et al., 2006, 2008).Meanwhile, particle swarm optimization (PSO) has demon- started a good performance in many functions and parameter optimization problems. PSO is a population-based optimize- ton strategy. It is initialized with a group of random particlesand then updates their velocities and positions with the

following formula:

$$\begin{aligned} v(t+1) &= v(t) + c1 * rand() * (pbest(t) - present(t)) \\ &\quad + c2 * rand() * (gbest(t) - present(t)) \\ present(t+1) &= present(t) + v(t+1) \end{aligned}$$

where vet is the particle velocity, present is the current par- tickle. best and gestate are defined as individual best and global best. rind is a random number between [0, 1]. c1, c2 are learning factors; usually c1=c2=2 (Wang et al., 2006). In the next section we will tailor the hybrid Quantum Swarm Evolutionary Algorithm (QSE) (Wang et al., 2006)to the problem of mining association rules.

The QSE-RM approach

In this section we first present QEA-RM for association rule mining and then we give a PSO version of QEA-RM named

QSE-Remain order to show how QEA concepts have been tailored to the problem of association rule mining, a formulation of the problem in terms of quantum representation is presented and a Quantum Swarm Evolutionary Algorithm for association rules mining QSE-RM is derived.

Quantum representation

QEA-RM uses the novel representation based on the concept of string of Q-bits called multiple Q-bit defined as below:

$$Q = \left[\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \cdots & \alpha_m \\ \beta_1 & \beta_2 & & \beta_m \end{array} \right]$$

where ŒatŒ2 + ŒbtŒ2 =1, t ¼ 1; ...; m, m is the number of Q-bits. Quantum Evolutionary Algorithm with the multiple Q-bit representation has a better diversity than classical genetic algorithm since it can represent superposition of states. Only

one multiple Q-bit with three Q-bits such as:

$$\left[\begin{array}{c|c|c} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{2} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & \frac{\sqrt{3}}{2} \end{array} \right]$$

is enough to represent the following system with eight states:

$$\frac{1}{4}|000\rangle + \frac{\sqrt{3}}{4}|001\rangle - \frac{1}{4}|010\rangle - \frac{\sqrt{3}}{4}|011\rangle + \frac{1}{4}|100\rangle + \frac{\sqrt{3}}{4}|101\rangle - \frac{1}{4}|110\rangle - \frac{\sqrt{3}}{4}|111\rangle$$

This means that the probabilities to represent the states $|000\rangle$, $|001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle$ are 1/16, 3/16, 1/16, 3/16, 1/16, 3/16, 1/16, 1/16 respectively. However in genetic algorithm one needs eight chromosomes for encoding.

For the data instances S of Section 2.1 given by $S = \{\langle A=1, B=1, C=0\rangle, \langle A=0, B=0, C=1\rangle, \langle A=0, B=0, C=1\rangle\}$ one would have a multiple Q-bits representation constituted from 3 Q-bits.

Measurement

he measurement of single Q-bit projects the quantum state onto one of the basis states associated with the measuring de-vice. The process of measurement changes the state to that measured. The multiple Q-bit measurement can be treated as a series of single Q-bit measurements to yield a binary solution P. In association rules, the occurrence of 1 in P means that the corresponding item or the attribute value is present in P how- ever 0 means that the corresponding item or attribute value is absent from P.

Structure of QEA-RM

The Quantum-inspired Evolutionary Algorithm for asocial- ton rules mining (QEA-RM) is described as follows:

ISSN NO: 0364-4308

```
Procedure QEA-RM
begin
   t ← 0
   initialize population of Q-bit individuals Q(t)
   project Q(t) into binary solutions P(t)
   compute fitness of P(t)
   generate association rule from each P(t) if there is any
   store the best solutions among P(t)
   while (not end-condition) do
      t ← t + 1
      project Q(t − 1) into binary solutions P(t)
      compute fitness from P(t)
      generate association rule from each P(t) if there is any
      update Q(t) using Q-gate
      store the best solutions among P(t)
   end
end
```

Table 1 Lookup table.

x_i	b_i	$f(x) \geq f(b)$	$\Delta\theta_i$	$s(\alpha_i, \beta_i)$			
				$\alpha_i\beta_i > 0$	$\alpha_i\beta_i < 0$	$\alpha_i = 0$	$\beta_i = 0$
0	0	False	0	0	0	0	0
0	0	True	0	0	0	0	0
0	1	False	0	0	0	0	0
0	1	True	Delta	−1	+1	±1	0
1	0	False	Delta	−1	+1	±1	0
1	0	True	Delta	+1	−1	0	±1
1	1	False	Delta	+1	−1	0	±1
1	1	True	Delta	+1	−1	0	±1

In the step ‘‘initialize population of Q-bit individuals Qtothe values of ai and bi are initialized with 1=p 2. The step ‘‘pro-jet into binary solutions Put’’ generates binary solutions by observing the states of populationfor each bit in mule-tiple Q-bit we generate a random variable between 0 and 1; if random(0, 1) < ŒbiŒ2 then we generate 1 else 0 is generated. In the step ‘‘compute fitness of Put’’, each binary solution Put is evaluated for the fitness value computed by the formula F of Section 2.2. The step ‘‘update Qto using Q-gate’’ is intoeduced.

as follows (Han and Kim, 2002):

```
Procedure update Q(t)
begin
   i ← 0
   while (i < m) do
      i ← i + 1
      determine Δθ_i with the lookup table
      [α'_i  β'_i]^T = U(Δθ_i)[α_i  β_i]^T
   end
end
```

Quantum gate UðDh1pti Þ is a variable operator, it can be chosen according to the problem. We use the quantum gate de-.

fined in Han and Kim (2002) as follows:

$$U(\Delta\theta_i) = \begin{vmatrix} \cos(\xi(\Delta\theta_i)) & -\sin(\xi(\Delta\theta_i)) \\ \sin(\xi(\Delta\theta_i)) & \cos(\xi(\Delta\theta_i)) \end{vmatrix}$$

where $\xi(\Delta\theta_i) = s(\alpha_i, \beta_i) * \Delta\theta_i$; $s(\alpha_i, \beta_i)$ and $\Delta\theta_i$ represents the rotation direction and angle, respectively. The lookup table is presented in Table 1. Delta is the step size and should be designed in compliance with the application problem. However, it has not had the theoretical basis till now, even though it usually is set as small value. Many applications set Delta = 0.01π. The function $f(x)$ (resp. $f(b)$) is the profit of the binary solution x (resp. best solution b). For example, if the condition $f(x) \geqslant f(b)$ is satisfied and xi, bi are 1 and 0, respectively, we can set the value of $\Delta\theta_i$ as 0.01π and $s(\alpha_i, \beta_i)$ as +1, −1, or 0 according to the condition of α_i, β_i; so as to increase the probability of the state $|1\rangle$.

Structure of QSE-RM

In order to introduce QSE-RM we present quantum angle. A quantum angle (Wang et al., 2006) is defined as an arbitrary angle θ and a Q-bit is presented as $[\theta]$. Then $[\theta]$ is equivalent to the original Q-bit as $\begin{bmatrix}\sin(\theta)\\ \cos(\theta)\end{bmatrix}$. It satisfies the condition:

$$|\sin(\theta)|^2 + |\cos(\theta)|^2 = 1.$$

Then a multiple Q-bit $\left[\begin{matrix}\alpha_1\\ \beta_1\end{matrix}\middle|\begin{matrix}\alpha_2\\ \beta_2\end{matrix}\middle| \cdots \begin{matrix}\alpha_m\\ \beta_m\end{matrix}\right]$ could be replaced by: $[\theta_1 | \theta_2 | \ldots | \theta_m]$.

The common rotation gate

$$[\alpha_i' \ \ \beta_i']^T = U(\Delta\theta_i)[\alpha_i \ \ \beta_i]^T$$

where $U(\Delta\theta_i) = \begin{vmatrix}\cos(\xi(\Delta\theta_i)) & -\sin(\xi(\Delta\theta_i))\\ \sin(\xi(\Delta\theta_i)) & \cos(\xi(\Delta\theta_i))\end{vmatrix}$, is replaced by $[\theta_i'] = [\theta_i + \xi(\Delta\theta_i)]$.

QSE-RM uses the concept of swarm intelligence of the PSO and regards all multiple Q-bit in the population as an intel-gent group, which is named quantum swarm. First QSE-RM finds the local best quantum angle and the global best value from the local ones. Then according to these values, quantum angles are updated by quantum gate. The QSE-RM based on

QEA-RM is given as follows:

1. Use quantum angle to encode Q-bit $Q(t)$ using $Q(t) = \{q_1^t, q_2^t, \ldots, q_m^t\}$ and $q_i^t = [\theta_{j1}^t | \theta_{j2}^t | \ldots | \theta_{jm}^t]$
2. Project $Q(t)$ into binary solutions $P(t)$ by observing the state of $Q(t)$ through $|\cos(\theta)|^2$ as follows: for quantum angle, we generate a random variable between 0 and 1; if $\mathrm{random}(0, 1) > |\cos(\theta)|^2$ then we generate 1 else 0 is generated.
3. The "update $Q(t)$ using Q-gate" is modified with the following PSO formula (Wang et al., 2006):

$$v_{ji}^{t+1} = \chi * (\omega * v_{ji}^t + c1 * rand() * (\theta_{ji}^t(pbest) - \theta_{ji}^t) + c2 * rand() * (\theta_i^t(gbest) - \theta_{ji}^t))$$

$$\theta_{ji}^{t+1} = \theta_{ji}^t + v_{ji}^{t+1}$$

where v_{ji}^t, θ_{ji}^t, $\theta_{ji}^t(pbest)$ and $\theta_i^t(gbest)$ are the velocity, current position, individual best and global best of the ith Q-bit of the jth multiple Q-bit. The parameters χ, ω, $c1$, $c2$ are, respectively, set to 0.99, 0.7298, 1.42, 1.57.

Vol 10 Issue 02,June 2020
ISSN NO: 0364-4308

Test and evaluation

in this section we compare Quantum Swarm Evolutionary Algorithm (QSE-RM) to the non-parallel version of Genetic Algorithm (GA-PVMINER) (Lopes et al., 1999). Since the parameters of QSE-RM are different from the parametersof GA-PVMINER, the comparison between QSE-RM and GA-PVMINER is done by fixing a threshold of time

Table 2 Structure of the Nursery School database.

	Attribute name	Attribute values
1	Parents	Usual, pretentious, great_pret
2	Has_nurs	Proper, less_proper, improper, critical, very_crit
3	Form	Complete, completed,incomplete, foster
4	Children	1, 2, 3, more
5	Housing	Convenient, less_conv, critical
6	Finance	Convenient, inconv
7	Social	Non-prob, slightly_prob, problematic
8	Health	Recommended, priority, not_recom
9	Recommendation	Not_recom, recommend, very_recom, priority, spec_prior

execution. In the remainder of this section, we will see that for the same goal and for the same time of execution, QSE-RM has generated rules with fitness better than the fitness of rules given by GA-PVMINER. Recall that QSE-RM and GA- PVMINER algorithms belong to the class of evolutionary algorithms. Evolutionary algorithms give good solution and may be non-optimal ones but in a reasonable time (polio- mail) of execution. All the tests were performed on 1.86 GHz.

InteCentrinoPC machine with 1.00 GB RAM, running.

on Windows XP platform. QSE-RM algorithm is wrytenwith.

MATLAB programming language. The dataset used for test-

Ing, namely the nursery school dataset, is a public domain

and available from UCI repository (http://www.archive.ics.u-

ci.edu/ml/) of machine learning. Nursery database wasderived.

from a hierarchical decision model originally developed to

rank applications for nursery schools (Bosanac and Rajkovic,

1990).

The Nursery database contains 12,960 instances and atria-

buttes, all of them categorical. The structure of Nursery data-

base is given in Table 2.

As it is done in Lopes et al. (1999) we have specified three

goal attributes, namely Recommendation, Social and Finance.

A threshold of execution time is fixed. In all cases, our results

are better than those found by GA-PVMINER.

Vol 10 Issue 02,June 2020
ISSN NO: 0364-4308

Table 3 Results for goal Recommendation = not_recom.

	Rule	$\|C\&P\|$	b	Fitness	J-measure
1	IF Housing = convenient AND Finance = inconv THEN Recommendation = not_recom	720	0.33	0.40003	0.0005344
2	IF Parents = great_pret AND Has_nurs = proper AND Children = 2 AND Housing = less_conv AND Finance = inconv AND Social = nonprob AND Health = not_recom THEN Recommendation = not_recom	4	1	0.40036	0.0009339
3	IF Parents = great_pret AND Health = not_recom THEN Recommendation = not_recom	1440	1	0.40010	0.0016954
4	IF Health = not_recom THEN Recommendation = not_recom	4320	1	0.40005	0.0000476

Table 4 Results for goal Recommendation = spec_prior.

	Rule	$\|C\&P\|$	b	Fitness	J-measure
1	IF Has_nurs = very_crit AND Health = priority THEN Recommendation = spec_prior	855	0.98	0.40011	0.0001762
2	IF Parents = pretentious AND Has_nurs = very_crit AND Children = 1 AND Housing = critical AND Finance = convenient AND Social = slightly_prob AND Health = priority THEN Recommendation = spec_prior	4	1	0.40038	0.0006295

For the goal ‘‘Recommendation = notarikon’’, the best rule found by GA-PVMINER is given in the first row of Table 3. In addition to this rule, our algorithm QSE-RM has discovered other more interesting rules, which are given in rows 2, 3 and 4 of Table 3. For example, the following rule is very important than the best rule given by

GA-PVMINER:

$$\text{“}IF\ Health = not\_recom\ THEN\ Recommendation = not\_recom\text{”}$$

with support $|C\&P| = 4320$, confidence $b = 1$ and fitness = 0.40005.

For the goal ‘‘*Recommendation = spec_prior*’’, the best rule found by GA-PVMINER is given in the first row of Table 4. In addition to this rule, our algorithm QSE-RM has discovered other more interesting rule with fitness = 0.40038 (see row 2 of Table 4).

The authors of Lopes et al. (1999) stated that the best rule found by their GA-PVMINER algorithm is:

$$\text{“}IF\ Has\_nurs = very\_crit\ AND\ Health = priority\ THEN\ Recommendation = spec\_prior\text{”}$$

$$\text{“}IF\ Finance = inconv\ AND\ Health = not\_recom\ THEN\ Recommendation = not\_recom\text{”}$$

with support $|C\&P| = 2160$, confidence $b = 1$ and fitness = 0.400.

Concerning the goals Social and Finance our results are also better than those found by GA-PVMINER.

Conclusion

In this piece, we looked at how the Quantum Swarm Evolute- ternary method (Wang et al., 2006) can be used to better mine association rules. Algorithm QSE-RM is developed from it. proposed. The efficacy of the QSE-RM algorithm is demonstrated by practical experiments in comparison to PVMINER (Lopes et al., 1999). In the same vein as PGA-RM (Melba and El-Ghazali, 2000), we intend to further hybridize QSE-RM and are currently studying its response to parallelization.

References

1. ***Agrawal, R., Malinski, T., Swami, S., 1993a. Mining association rules between sets of items in large databases. In: Benemann, P., Jahoda, S. (Eds.), Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, DC, May 26–28, pp. 207–216.***

2. ***Agrawal, R., Malinski, T., Swami, S., 1993b. Mining association rules between sets of items in large databases. SIGMOD Record 22 (2), 207–216 (ACM Special Interest Group on Management of Data).***

3. ***Anguilla, F., Ianni, G., Palopoli, L., 2001. On the complexity of mining association rules. In: Proc. Nono Convegno Nazionale usSistemaEvolute di Basi di Dati (SEBD), pp. 177–184.***

4. ***Bosanac, M., Rajkovic, V., 1990. Expert system for decision making. Systemic 1 (1), 145–157.***

5. ***Han, K.H., Kim, J.H., 2002. Quantum-inspired Evolutionary Algo- rhythm for a class of combinatorial optimization. IEEE Transaction on Evolutionary Computation 6 (6), 580–593.***

6. ***Kennedy, J., Eberhart, R.C., 1995. Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, vol. 9, Australia, pp. 2147–2156.***

7. ***Layer, A., Mosholu, S., Barouche, M., 2006. Multiple sequence alignment by quantum genetic algorithm. In: Proceedings of the IEEE Conference of the International Parallel and Distributed Processing Symposium (IPDPS'2006), Rhodes Island, Greece, April 25–29.***

8. ***Layer, A., Mosholu, S., Barouche, M., 2008. Quantum genetic algorithm for multiple RNA structural alignment. In: IEEE Proceedings of the Second Asia International Conference on Modelling and Simulation (AMS 2008), Kuala Lumpur, Malaysia, May 13–15***

9. ***Lopes, H.S., Araujo, D.L.A., Freitas, A.A., 1999. A parallel genetic algorithm for rule discovery in large databases. In: IEEE Systems, Man and Cybernetics Conf., pp. 940–945.***

10. ***Melba, M., El-Ghazali, T., 2000. A parallel genetic algorithm for rule mining. In: IPDPS, IEEE Computer Society.***
11. ***Pei, J., Han, J., Yin, Y., 2000. Mining frequent patterns without candidate generation. In: ACM SIGMOD Int. Conference on Management of Data.***

12. ***Shor, P.W., 1994. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. In: Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, November 20–22.***