New Method For Geometric Constraint Solving Based on the Genetic Quantum Algorithm

Cao Chunhong
College of computer science and technology, Jilin University
Changchun 130012, P.R.China
Chunhongcao_li@163.com

Li Wenhui
College of computer science and technology, Jilin University
Changchun 130012, P.R.China
liwh@public.cc.jl.cn

ABSTRACT

This paper proposes a novel genetic quantum algorithm (GQA) to solve geometric constraint problems. Instead of binary, numeric or symbolic representation, we introduce qubit chromosome representation. GQA is based on qubit and superposition of states and is used in the process of geometric constraint solving in order to get the solution sequence. As GQA has diversity caused by the qubit representation, there is no need to use the genetic operator. Qubit chromosome can be updated by proper quantum gate in the circulation. The experiment indicates GQA can solve the geometric constraint problem effectively.

Keywords

Geometric constraint solving geometric decomposing genetic algorithm quantum computing genetic quantum algorithm quantum gate qubit chromosome

1. Geometric Constraint Solving

Geometric constraint solving is a remarkable problem based on the constraint design. Once a user defines a series of relations, the system will satisfy the constraints by selecting proper state after the parameters are modified. The idea is named model-based constraints. Constraint solver is a segment for the system to solve the constraints. Many scholars worked over the constraint solving by numeric computing approach, artificial intelligence approach, degree of freedom approach and graph-based approach. To conclude there are whole solving, sparse matrix solving, joint analysis solving, constraint diffuse solving, symbolic algebra solving and guides solving. [Aba95a]

Geometric constraint problem can be expressed in a set of nonlinear equations. Nonlinear equations can be solved by Newton-Raphson algorithm. When the size of geometric constraint problem is large, the scale of set of equation and variable of nonlinear equation is very large. It will be difficult for the efficiency and stability to meet the requirement of interactive design. Furthermore, the geometric information in the geometric system will not be treated properly and we cannot deal with under- and over-constrained design system well when all the equations must be solved. In order to solve problems hereinbefore, we must decompose the geometric constraint system into a series of small ones. So the geometric constraint decomposition is the key of geometric constraint solving technique.

We can decompose large constraint problem into a series of small problems that can be solved step by step. The subsidiary problems and their solving sequence constitute a new geometric constraint-solving problem. In many conditions, subsidiary problem can be solved by geometric method instead of numeric overlap. The constraint solving process based on the geometric constraint decomposition consists of two phases:(1) decomposing phase and (2) execution phase. [Aba95a]

Geometric problems defined by constraints have an exponential number of solution instances in the number of geometric elements involved. Generally, the user is only interested in one instance such that besides fulfilling the geometric constraints, exhibits some additional properties. Selecting a solution instance amounts to selecting one among a

WSCG POSTERS proceedings
WSCG'2004, February 2-6, 2004, Plzen, Czech Republic.
Copyright UNION Agency – Science Press

number of different roots of a nonlinear equation or system of equations. The problem of selecting a given root was christianized in as the Root Identification Problem. We introduce Genetic Quantum Algorithm (GQA) to solve the root identification problem by searching in the solution space automatically. The user specifies the intended solution instance by defining a set of additional constraints or predicates on the geometric elements that drive the search of genetic quantum algorithm. [Rjo02a]

2. Genetic Quantum Algorithm (GQA)

2.1 Genetic Algorithm

Genetic Algorithm imitates life's evolution process from low-grade to high-grade. It applies evolution operation in a group of genes that codes searching space. In every generation of genetic algorithm, the algorithm can search different areas of parameter space. Then it focuses on the highest part of expectation value in the solution space.

In the genetic algorithm, we carry genetic operation after forming initial generation by coding. The task of genetic operation is carrying a certain operation according to the fit degree to the circumstance. The genetic operation consists of three basic genetic operators: 1) selection 2) crossover 3) mutation.

The genetic algorithm needn't external information in the evolution search and it is only based on the fit degree function. The zoom adjusting to the fit degree is named fit degree decision. The decision mode is classified: (1) linear decision, f'=af+b； （2）σ truncation, f'=f-（$\overline{f}$-cσ）；（3）power decision, f'=f^k。[Che01a]

2.2 Quantum Algorithm

Quantum computer is a system that can fulfill computing task following quantum mechanics. It codes information with quantum states and the basic union to storage quantum information is a quantum double-state system (or to say two dimensional Hilbert space) named qubit. We can consider the quantum computer as a circuit made up of a series of quantum gates. A qubit can be 0 or 1 state, also their superposition state. The qubit superposition |ψ> can be expressed: $|\psi\rangle = p_0|0\rangle + p_1|1\rangle$, here p_0 and p_1 are both pluralism to represent the swing of basic sates 0 and 1. $|p_0|^2$和$|p_1|^2$ represent the probability of the basic state 0 and 1 of the system respectively and can be induced to 1, i.e. $\sum|p_i|^2=1$. When we measure the superposition |ψ> of quantum system, the state of the quantum register is in |0> state in probability of $|p_0|^2$ and |1> state in probability of $|p_1|^2$. [Zho02a]

A qubit string of the length m represents a linear superposition of solutions to the problem. The length of a qubit string is the same as the number of items. The i-th item can be selected with the probability $|\beta_i|^2$ or $(1-|\alpha_i|)^2$. Thus, a binary string of the length m is formed from the qubit string. For every bit in the binary string, we generate a random number γ from the range [0.. 1]; if $\gamma>|\alpha_i|^2$, we set the bit of the binary string. The binary string x_j^t, j=1,2…n, of P (t) represents a j-th solution to the problem. For notational simplicity, x is used instead of x_j^t in the following.

Genetic Algorithm with qubit representation has a better characteristic of diversity than classical approaches, since it can represent superposition of states. Only one qubit chromosome such as (1) is enough to represent eight states, but in classical representation at least eight chromosomes（000），（001），（010），（011），（100），（101），（110），（111）are needed. Convergence can be also obtained with the qubit representation. As $|\alpha_i|^2$ or $|\beta_i|^2$ approaches to 1 or 0, the qubit chromosome converges to a single state and the property of diversity disappears gradually.

2.3 Genetic Quantum Algorithm

The GQA can be written as follows:

```
Procedure GQA
Begin
t←0
initialize Q (t)
make P (t) by observing Q (t) states
evaluate P (t)
store the best solution among P (t)
while（not cease condition）do
  begin
    t←t+1
    make P (t) by observing Q (t-1) states
    evaluate P (t)
    update Q (t) by U (t)
    store the best solution among P (t)
  end
end
```

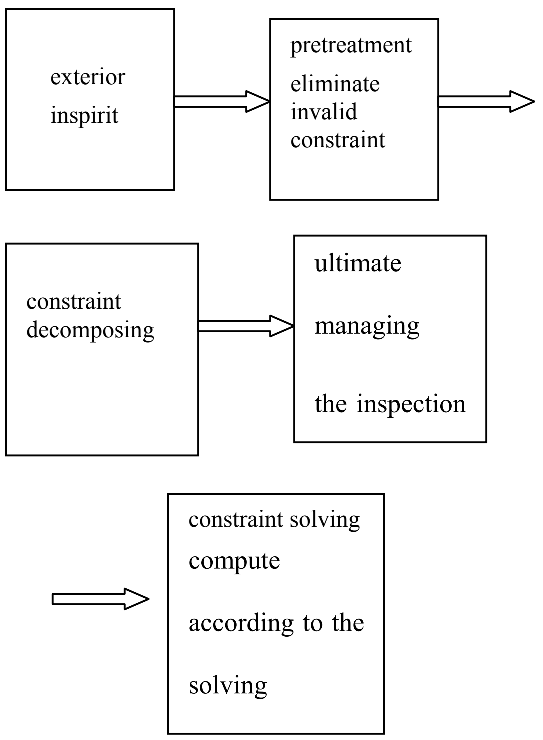
Definition 1. qubit: is the length of quantum chromosome.

Definition 2. qubit chromosome q_j^t is defined as: $q_j^t=\begin{bmatrix}\alpha_1^t & \alpha_2^t & \cdots & \alpha_m^t \\ \beta_1^t & \beta_2^t & \cdots & \beta_m^t\end{bmatrix}$，Q (t)={$q_1^t$，$q_2^t$，…，$q_n^t$}，at generation t, where n is the size of population, m is the number of qubits, i.e., the string length of the qubit string, and j=1，2，…n.

In the step of initialize Q (t), α_i^t和β_i^t, i=1，2，…，m, of all q_j^t , j=1，2，…，n, in Q (t) are initialized with $\frac{1}{\sqrt{2}}$. It means that one quantum chromosome, $q_i^t|_{t=0}$ represents the linear superposition of all possible states with the probability $|\psi_{qi0}\rangle = \sum_{k=1}^{2^m} \frac{1}{\sqrt{2^m}} |S_k\rangle$, where S_k is the k-th state represented by the binary string （$x_1x_2x_m$）, where x_i , i=1，2，…，m, is either 0 or 1. The next step makes a set of binary solutions, P (t), by observing Q（t）states, where P（t）=$\{x_1^t,x_2^t,\ldots,x_n^t\}$ at generation t. One binary solution, x_j^t， j=1，2，…，n, is a binary string of the length m, and is formed by selecting each bit using the probability of qubit, either $|\alpha_i^t|^2$ or $|\beta_i^t|^2$, i=1，2，…，m of q_i^t. Each solution x_j^t is evaluated to give some measures of its fitness. The initial best solution is then selected and stored among the binary solutions P (t).

In the process of updating Q (t), qubit chromosome is updated by some appropriate quantum gates U (t). It should be noted that some genetic operators could be applied, such as mutation and crossover that can make the probability of linear superposition of states change. But as GQA has diversity caused by the qubit representation, there is no need to use the genetic operator. If the probabilities of mutation and crossover are high, the performance of GQA decreased notably.

The figure applying the genetic quantum into geometric constraint solving is as follows:



exterior inspirit
pretreatment
eliminate invalid constraint
constraint decomposing
ultimate managing the inspection
constraint solving
compute according to the solving

Figure1. Applying the genetic quantum algorithm into geometric constraint solving

Geometric constraint decomposing and geometric constraint solving are the kernel processes and Genetic quantum algorithm（GQA）is used to construct the solving sequence. Then we solve the geometric constraint according to the solving sequence in turn.

2.4 Quantum Gate

A quantum computer is equivalent to a quantum Turing. How to construct a quantum computer? It has been proved theoretically [Aya93a] that quantum Turing is equivalent to a quantum logic circuit, so we can form a quantum computer by the combination of quantum logic gates.

The quantum gate is an I/O equipment whose input and output are separate quantum variable (such as spin) [Dde89a][Aba95a]. Its operating input variable is expressed in a unitary functor, viz. the state vector from |ψ〉 to |ψ′〉=U|ψ〉. As we know, to a digital computer, if there were linear logic gates such as NOT gate and COPY gate, as well as nonlinear logic gates such as AND gate, it will be able to finish arbitrary logic or arithmetic task. For the quantum computing, it is also true. Barenco [Efr82a] in the Oxford has proved it true. In 1982 Fredkin and Toffoli brought forward a reversible computing logic gate (for short Frendkin gate, namely Fredingate)[Pot90a], which is an equipment possessing three input lines and three output lines. One of lines is input line (assuming it is line a) and its logic state has not been changed when passing the gate. If the bit of controlling line is 0, then the bit of other two lines is invariable; and if the bit of controlling line is 1, then the bit of other two lines should be exchanged.

We can get Frendkin gate by basic one bit and two bits quantum gates. [Hfc95a] In this paper we adopt Frendkin gate.

3.Application instance and result analysis

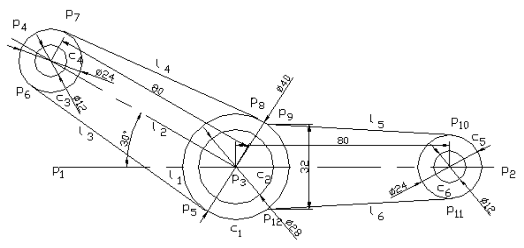


Figure 2. A design instance

In the figure 2, there 24 geometric elements, the freedom degree is 54 and the constraint degree is 49. We change DIST_PP（p_{c1}，p_{c3}，80） to DIST_PP（p_{c1}，p_{c3}，60）, VDIST_PP (p_9，p_{12}, 32) to VDIST_PP (p_9，p_{12}, 30) ，ANGLE_LL (l_1, l_2, $\pi/6$) to ANGLE_LL (l_1, l_2, $\pi/4$). If we use 49 nonlinear equations to superpose in order to solve 54 position variables corresponding to 24 geometric elements, the solving will be very inconvenient. The solution sequence decomposed by genetic quantum algorithm is as follows:

$p_9 \rightarrow p_{12} \rightarrow c_1 \rightarrow l_1 \rightarrow p_3 \rightarrow p_{11} \rightarrow c_5 \rightarrow l_6 \rightarrow p_{10} \rightarrow l_2 \rightarrow c_3 \rightarrow l_4 \rightarrow p_7 \rightarrow p_8 \rightarrow l_3 \rightarrow c_4 \rightarrow p_5 \rightarrow p_6 \rightarrow c_2 \rightarrow c_4 \rightarrow c_6 \rightarrow l_5 \rightarrow p_1 \rightarrow p_2 \rightarrow p_4$

Figure 3. the final solving sequence

We realize GQA starts with a random search and this shows the initial quantum chromosome includes all solutions with the same probability. Then the probability of the cases adjacent to the solution increases. It changes into a local search step by step after 50 generations. Finally, the probability of the qubit chromosome converges to the best solution. That is, GQA starts with a global search and changes automatically into a local search.

The final solving result is as follows:

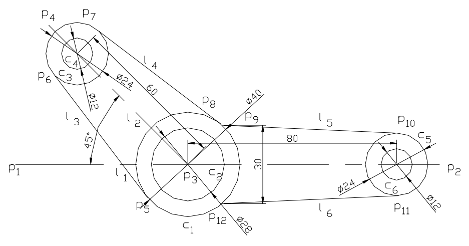


Figure 4. Solving result

4. Result

The genetic quantum algorithm can finish decomposing the geometric constraints by getting the sequence of geometric elements. Combining genetic algorithm and quantum computation, it is a new idea by itself. Now we apply it into the geometric constraint solving and have proved it can improve the speed of solving and reduce the complication degree of computation.

REFERENCES

[Aba95a] A．Barenco et al．，Phys．Rev．，A52（1995），3457

[Aya93a] A．Yao， in Proceedings of the 34th Annual Symposium of Foundation of Computer Science. (IEEE Computer Society, Los Almamitos, CA), 1993, 352

[Che01a] Chen Guoliang, Wang Xufa, Zhuang Zhenquan, Wang Dongsheng genetic algorithm and its application Post &Telecom Press, February, 2001: 71-73

[Dde89a] D．Deusch，Proc．Roy．Soc London Ser．，A425（1989），73

[Efr82a] E．Fredkin and T．Toffoli，Int．J．Theor．Phys．，21（1982），219

[Hfc95a] H．F．Chau and F．Wilczek，Phys．Rev．lett．，75（1995），748

[Pot90a] Pothen A and Fan C J.Computing the blocktriangular form of a sparse matrix. ACM Trans.Mathematical Software, 1990, 16(4): 303-324

[Rjo02a] R．Joan-Arinyo， M．V．Luzon， and A．Soto Constructive Geometric Constraint Solving：A New Application of Genetic Algorithms， September 2002

[Yua99a] Yuan Bo, Research and Implementation of Geometric Constraint Solving Technology, doctor dissertation of Tsinghua University, 1999

[Zho02a] H．F．Chau and F．Wilczek，Phys．Rev．lett．，75（1995），748