

Algoritmos Evolutivos inspirados en Computación Cuántica

13 de mayo de 2015

Rodrigo Arias Mallo
Carlos Pérez Ramil

Resumen

Este estudio tiene como objetivo presentar una serie de técnicas que tratan de combinar la forma de resolver problemas de los Algoritmos Evolutivos con los conceptos básicos de la Computación Cuántica, así como su repercusión y posibles aplicaciones en el mundo de la Robótica.

Índice

1. Introducción	3
1.1. Computación Evolutiva	3
1.2. Computación Cuántica	4
2. Estado del arte	5
3. El concepto de QEA	6
3.1. Representación	6
3.2. Algoritmo	6
3.3. Rendimiento	8
4. Aplicaciones en la robótica	9
4.1. Motricidad	9
4.2. Planificación de rutas	9
4.3. Otras aplicaciones	10
5. Conclusiones	11
Referencias	12

1. Introducción

Los Algoritmos Evolutivos son un método de resolución de problemas ampliamente desarrollado desde la década de 1980 hasta la actualidad. Utilizan los principios de la evolución biológica para operar sobre un conjunto de individuos que representa distintas soluciones a un problema determinado. Entre sus múltiples variantes encontramos una relativamente reciente, que toma algunos conceptos del campo emergente de la Computación Cuántica para modelar el individuo y la manera en que este evoluciona. Hay que tener presente que no se trata de un algoritmo cuántico, sino de un algoritmo clásico, pensado para ser ejecutado en un computador digital, pero que utiliza conceptos de Computación Cuántica como son el *qbit* o la superposición de estados. Por eso se les denomina «Algoritmos Evolutivos inspirados en Computación Cuántica» (QEA, por sus siglas en inglés).

1.1. Computación Evolutiva

La Computación Evolutiva es una rama de la Inteligencia Artificial que trata principalmente problemas de optimización. Surge con la idea de aplicar los principios de la evolución biológica a la resolución de problemas complejos. En particular, se basa en los conceptos de *selección natural*, *recombinación genética* y *mutación*. Las primeras aproximaciones datan de los años 50, pero no se consagra como disciplina hasta la década de los 70. Desde entonces se ha realizado mucha investigación en este campo, y encontramos innumerables aplicaciones en todo tipo de contextos, no sólo de carácter científico-técnico, sino también artístico, económico o social.

Existen tres grandes subcampos de la Computación Evolutiva, entre muchos otros de distinta relevancia:

- *Algoritmos Genéticos*. Modelan las posibles soluciones de un problema como individuos de una población, y las distintas variables como su información genética. Funcionan de forma iterativa, simulando las distintas generaciones de la población, y aplicando los operadores genéticos de selección, cruce y mutación de manera estocástica.
- *Programación Genética*. Similar a los Algoritmos Genéticos, pero en este caso las soluciones son programas, y su calidad se determina en función de su capacidad para resolver un problema computacional.
- *Estrategias Evolutivas*. Utilizan variables adicionales para controlar el proceso de evolución de forma dinámica.

Los algoritmos evolutivos son adecuados para problemas en los que el espacio de búsqueda es muy grande y no hay una solución analítica conocida. Esto los hace especialmente adecuados para ciertos problemas de Robótica, como son el diseño de morfologías o de controladores. Sin embargo, presentan ciertos inconvenientes. En primer lugar, no garantizan que se encuentre el óptimo global. Y en caso de alcanzar a una solución aceptable, resulta muy difícil (o casi imposible, en algunos casos) averiguar cómo y por qué se ha llegado hasta esa solución.

1.2. Computación Cuántica

La Computación Cuántica es un nuevo paradigma de computación. Emplea las propiedades especiales de la materia que suceden en una escala más pequeña.

El objetivo que busca la Computación Cuántica es el de hacer uso de las propiedades inherentes a la naturaleza para realizar computaciones. De la misma forma que los ordenadores clásicos que acostumbramos a emplear, existe una analogía cuántica; el ordenador cuántico. Los bits se convierten en qbits.

Sin embargo, los qbits tienen la propiedad de encontrarse en varios estados al mismo tiempo. Esta propiedad se denomina *superposición*. Así, la representación del estado de un qbit $|\psi\rangle$ es:

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \alpha |0\rangle + \beta |1\rangle$$

Donde $\alpha, \beta \in \mathbb{C}$ y también $|\alpha|^2 + |\beta|^2 = 1$. La medición de un qbit resultará en 0 o bien en 1. De forma que la probabilidad de que resulte 0 es $|\alpha|^2$ y de que resulte 1 es $|\beta|^2$.

Una ventaja frente a la computación clásica es el entrelazamiento de estados. Para un sistema $|\psi\rangle$ de n qbits:

$$|\psi\rangle = \gamma_1 |00\dots 0\rangle + \gamma_2 |00\dots 1\rangle + \dots + \gamma_{2^n} |11\dots 1\rangle$$

Sólo debe cumplir la restricción:

$$\sum_{i=1}^{2^n} |\gamma_i|^2 = 1$$

De forma que se pueden construir estados especiales, que no son posibles en los bits clásicos, por ejemplo el estado de Bell, denominado $|B_{00}\rangle$.

$$|B_{00}\rangle = \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle)$$

La probabilidad de que se obtenga como resultado de una medición 01 o 10, es nula. Sin embargo, la probabilidad de obtener 00 o bien 11 es $\frac{1}{2}$.

2. Estado del arte

Para entender el tipo de algoritmos que se va a analizar, hay que entender primero la diferencia entre QEA y otros conceptos que pueden parecer similares, pero no lo son en absoluto. Las palabras *quantum*, *evolutionary* y *genetic*, aparecen juntas en decenas de publicaciones que nada tienen que ver unas con otras.

Existen tres líneas de investigación que relacionan los conceptos de la Computación Cuántica con la Computación Evolutiva: una muy popular es la generación automática de circuitos cuánticos mediante algoritmos genéticos tradicionales; otra, que es la que se denomina como *Quantum-Inspired Evolutionary Algorithms*, consiste en el diseño de algoritmos evolutivos (para computadores digitales) que toman algunos conceptos de la Computación Cuántica para modelar la población; y una última, quizás la menos desarrollada de todas por su elevadísima complejidad, que estudia algoritmos evolutivos cuánticos en sentido literal, es decir, para ser ejecutados en una máquina cuántica. La que se va a estudiar aquí es la segunda, pero es interesante conocer y distinguir las demás.

A. Narayanan y M. Moore [8] fueron los primeros en proponer la idea de los QEA en 1996. Sin embargo, se utilizará el trabajo de Kuk-Hyun Han y Jong-Hwan Kim [1, 2, 3] como guía para exponer el algoritmo, por el mayor nivel de detalle que hay en su trabajo. En su tesis, publicada en 2003, Han propone una estructura genérica para los QEA, y después implementa varias versiones adaptadas a distintos problemas.

Desde su primera aparición en 1996, los QEA han sido objeto de estudio de un número considerable de publicaciones, pero no se puede decir que sea una línea de investigación especialmente popular. Después de más de una década, su desarrollo no parece avanzar demasiado, y las novedades consisten solamente en pequeñas mejoras del algoritmo original. En cualquier caso, se ha llevado a la práctica en numerosos problemas reales con aparente éxito, normalmente mejorando los resultados obtenidos por técnicas evolutivas tradicionales.

3. El concepto de QEA

A continuación se detalla la propuesta de Han y Kim para el esquema general de un QEA.

3.1. Representación

Un individuo posee un solo cromosoma que se modela como un vector de qbits:

$$\left[\begin{pmatrix} \alpha_1 \\ \beta_1 \end{pmatrix}, \begin{pmatrix} \alpha_2 \\ \beta_2 \end{pmatrix}, \dots, \begin{pmatrix} \alpha_m \\ \beta_m \end{pmatrix} \right]$$

donde $|\alpha_i|^2 + |\beta_i|^2 = 1$ para $i = 1, 2, \dots, m$. Se utilizan los valores de probabilidad de cada qbit para simular el concepto de superposición de estados, de forma que un individuo contiene información acerca de todas las soluciones simultáneamente. Lo que se pretende con esta representación es aumentar la variabilidad genética en la población, ya que la información transmitida a lo largo de las generaciones son valores de probabilidad de todas las secuencias posibles, y no secuencias concretas.

3.2. Algoritmo

Algoritmo 1 Esquema del algoritmo QEA

```
procedure QEA
begin
  t = 0
  1) initialize Q(t)
  2) make P(t) by observing the states of Q(t)
  3) evaluate P(t)
  4) store the best solutions among P(t) into B(t)
  while (not termination-condition) do
    begin
      t = t + 1
      5) make P(t) by observing the states of Q(t-1)
      6) evaluate P(t)
      7) update Q(t) using Q-gates
      8) store best solutions among B(t-1) and P(t) into B(t)
      9) store best solution b among B(t)
    end
  end
end
```

$Q(t) = \{q_1^t, q_2^t, \dots, q_n^t\}$ es la población en el instante t .

$P(t) = \{x_1^t, x_2^t, \dots, x_n^t\}$ es un conjunto de soluciones en el instante t , resultante de efectuar una medición en cada individuo de $Q(t)$.

$B(t) = \{b_1^t, b_2^t, \dots, b_n^t\}$ es el conjunto de mejores soluciones obtenidas hasta el instante t . \mathbf{b} es la mejor solución global.

El algoritmo no es un algoritmo cuántico, si no que se ejecuta en un ordenador clásico, por lo que es posible determinar el estado de un qbit, sin que este se colapse. De esta forma, de un vector de qbits $\mathbf{q} \in Q(t)$ es posible realizar una medición en el vector binario $\mathbf{x} \in P(t)$, sin modificar \mathbf{q} .

1. Los qbits de los cromosomas se inicializan con valores de igual probabilidad para todos los estados.
2. Para cada individuo $j = 1, 2, \dots, n$, se realiza una medición de los qbits del cromosoma \mathbf{q}_j . En el vector \mathbf{x}_j se almacena el estado del cromosoma colapsado, que formará la solución.
3. Evalúa el resultado de cada individuo \mathbf{x}_j empleando una función de *fitness*.
4. Selecciona las mejores soluciones tras evaluar cada \mathbf{x}_j , y se almacenan en $B(t)$. Obviamente, $B(t) = P(t)$ cuando $t = 0$.

A continuación comienza un bucle que se repetirá hasta que lo indique la condición de parada.

5. Repite el procedimiento del paso 2, que consiste en medir el estado de los cromosomas de $Q(t - 1)$ en $P(t)$. Sin alterar el estado del cromosoma.
6. Evalúa \mathbf{x}_j para cada $j = 1, 2, \dots, n$.
7. Realiza una modificación en el valor de los qbits de cada individuo. Consiste en una transformación aplicando una puerta cuántica de rotación. Se acerca el valor de cada qbit a $|0\rangle$ o a $|1\rangle$, aumentando o reduciendo el valor de α y de β en función de la variación del *fitness* con respecto al mejor actual. De esta forma, cada vez existen más probabilidades de que el valor resultante de medir proporcione un resultado más acercado al indicado por la función de *fitness*.
8. Almacena las mejores soluciones de entre $B(t - 1)$ y $P(t)$ en $B(t)$. Vuelve al paso 5.

Al finalizar, el vector \mathbf{b} contendrá la mejor solución alcanzada, en forma de vector binario.

El algoritmo propuesto por Han [1, 2, 3] no emplea las operaciones habituales de cruce ni de mutación para producir la siguiente generación de individuos. Sin embargo, muchas otras publicaciones que se citan más adelante sí hacen uso de ellos.

3.3. Rendimiento

Se han realizado experimentos con distintos problemas de optimización combinatoria y numérica. Han y Kim [3] hacen una comparativa entre algoritmos genéticos tradicionales y QEA para el problema de la mochila. En este caso, el algoritmo utiliza a mayores un operador de *migración*, que se define como el proceso de copiar un \mathbf{b}_j^t en otros elementos de $B(t)$ con un valor de *fitness* inferior. Nótese que el operador de migración se aplica sobre el conjunto de mejores soluciones, y no sobre el conjunto de individuos. De todas formas, una migración en la generación t provoca variaciones en los individuos de la generación $t + 1$ por medio de la actualización de los qbits en el paso 7. Los resultados que obtienen son en general favorables a los QEA, ya que estos no convergen de forma tan prematura como los tradicionales, alcanzando valores de beneficio superiores. Narayanan y Moore [8] aplican QEA al problema del viajante con resultados similares.

Mohammed *et al.* [7] experimentan con combinaciones de distintos tipos de cruce (*single-point*, *two-point*, *multi-point* y *quantum interference crossover*) y de mutación (*quantum inversion* y *quantum swap mutation*) para tratar de determinar las que producen mejores resultados en problemas de búsqueda de mínimos/máximos de funciones. En general, concluyen que la utilización de operadores genéticos es beneficiosa para el algoritmo.

4. Aplicaciones en la robótica

Existen varios ejemplos de QEA aplicados al campo de la robótica. En general se centran en problemas con un amplio espacio de búsqueda, donde los algoritmos evolutivos son apropiados. Dado que es un campo de reciente desarrollo, las posibles aplicaciones aún no han sido exploradas en profundidad.

4.1. Motricidad

Gwo-Ruey Yu y Lun-Wei Huang [13] propusieron un controlador para un brazo robótico de dos articulaciones, empleando lógica difusa y basado en un problema de optimización LMI (linear matrix inequality).

Los resultados son comparados con el mismo controlador, con algoritmos genéticos. El resultado para QEA es mejor que para GA, sin embargo no incluye una comparación cuantitativa.

In-Won Park, Ki-Baek Lee y Jong-Hwan Kim [9] emplean el algoritmo MQEA, que mejora QEA añadiendo varios objetivos para minimizar. Lo han aplicado al control de la postura de un robot humanoide, cuando recibe una fuerza externa, de forma que pueda mantenerse de pie. Utilizan el método ILQR (iterative linear quadratic regulator) en el que los pesos de la matriz se obtienen de forma evolutiva.

Obtuvieron las trayectorias óptimas que mejor satisfacen los criterios de calidad (ángulo, torque y tiempo).

4.2. Planificación de rutas

En 2009, Ye-Hoon Kim y Jong-Hwan Kim [5] usan el método MQEA para diseñar de forma eficiente, un sistema de diseño de rutas, basado en lógica difusa. Emplean el tiempo, la dirección global y local como criterios de medida de calidad. Tratan de golpear una pelota en un juego de fútbol para robots.

Al comparar los resultados con el algoritmo NSGA-II (nondominated sorting genetic algorithm-II), obtienen unos resultados más cercanos al frente óptimo de Pareto.

Un año más tarde, Yangguang Sun y Mingyue Ding [11] analizan el problema de trazar rutas para evitar obstáculos (path planning). Codifican la ruta en los propios qbits y añaden operadores de mutación. Los resultados son de mayor diversidad poblacional, mejores capacidades de búsqueda y una mejor convergencia.

En Julio de 2013, Shen Xiaoning [12] mejoró las soluciones al problema de trazado de rutas, para entornos dinámicos. A través de un mapa de casillas, trata de buscar una ruta, en el que los obstáculos cambian de posición y morfología. Además añade operaciones de mutación y cruce, para evitar la convergencia prematura.

Los resultados muestran otra vez que QEA supera a los algoritmos genéticos clásicos. Además muestra también los resultados de QEA sin mutación ni cruce, y se aprecia que en el caso en el que aparece un nuevo obstáculo, no consiguen encontrar el camino óptimo. Por el contrario, con dichos operadores, la ruta óptima se encuentra el 100 % de los casos.

4.3. Otras aplicaciones

Si-Jung Ryu, Rituparna Datta y Jong-Hwan Kim [10] proponen emplear otra variante de QEA denominada DMQEA (dual multiobjective quantum-inspired evolutionary algorithm) para la colocación óptima de sensores de proximidad en un entorno 2D. En los experimentos tratan de minimizar el número de sensores, la cobertura y el ratio de interferencia. Sin embargo no proporcionan una comparación con otros algoritmos. Mencionan que el algoritmo DMQEA es capaz de indicar las posiciones óptimas de la colocación de cada sensor, para minimizar los objetivos, en las proporciones que se le indiquen.

Deok-Hwa Kim et. al [4] han propuesto una solución para la detección automática de robots, en el programa MiroSOT, enfocado a campeonatos de fútbol para robots. Dado que el ajuste de los colores es un proceso difícil por motivo de las sombras y cambios de luz, usan QEA para determinar de forma automática los parámetros. Han conseguido resolver el problema, pero la simulación es demasiado lenta como para realizarla en tiempo real, durante la competición. Para solventarlo proponen emplear procesamiento por GPU, o realizar en preprocesado bayesiano de la imagen, que agilice el análisis posterior.

Zhiyong Li et. al [6] aplican QEA a entornos multi-robot, para trabajar de forma cooperativa en una formación de coalición. Los robots deben trabajar en equipo para solucionar una tarea. Enfocan la solución de forma genérica, donde existe una función objetivo a minimizar. En comparación con los algoritmos genéticos clásicos, la calidad de esta solución no sólo es mejor, si no que se encuentra de forma más rápida.

5. Conclusiones

Los Algoritmos Evolutivos inspirados en Computación Cuántica son una nueva vertiente de la Computación Evolutiva que constituye una alternativa viable a las técnicas tradicionales. No se pone en duda su eficacia ni su eficiencia a la hora de resolver problemas. Lo que sí es cuestionable es su grado de «inspiración» en los conceptos de la cuántica.

El QEA utiliza números reales para representar los qbits, desechando por completo la parte compleja. Esto implica que la puerta cuántica aplicada en el paso 7 del algoritmo sólo realiza rotaciones con 1 grado de libertad, cuando las operaciones cuánticas sobre qbits en estados puros permiten hasta 2 grados de libertad. Tampoco se aprovecha una de las propiedades más interesantes de la información cuántica, que es el entrelazamiento de qbits que se expone en la sección 1.2. En QEA, los qbits siempre se tratan individualmente, sujetos a la restricción ya mencionada que establece $|\alpha_i|^2 + |\beta_i|^2 = 1$. Por todo esto, la clase de algoritmos aquí expuesta se parece más a un método probabilista clásico que a uno cuántico. Los elementos denominados como «qbits» son totalmente equivalentes a un valor de probabilidad en el intervalo $[0, 1]$, y las operaciones referidas como «puertas cuánticas» son simplemente desplazamientos dentro de ese intervalo. Si la idea original proviene o no de los conceptos cuánticos es irrelevante para explicar el algoritmo, pues se podría hacer en términos probabilistas.

A pesar de todo, se trata de una nueva revisión del algoritmo evolutivo que parece dar buenos resultados a la hora de aplicarlo a problemas reales. Posibles investigaciones futuras pasarían por explorar otros subcampos de la Computación Evolutiva más minoritarios desde la perspectiva de QEA. Y a largo plazo, cuando se pueda explotar verdaderamente todo el potencial de la Computación Cuántica, el diseño e implementación de algoritmos genéticos cuánticos constituirá un área de investigación verdaderamente prometedora.

Referencias

- [1] Kuk-Hyun Han. Quantum-inspired evolutionary algorithm. *Korea Advanced Institute of Science and Technology (KAIST)*, June 2003.
- [2] Kuk-Hyun Han and Jong-Hwan Kim. Introduction of quantum-inspired evolutionary algorithm. *Proceedings of the 2002 FIRA Robot World Congress*, May 2002.
- [3] Kuk-Hyun Han and Jong-Hwan Kim. Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE Transactions on Evolutionary Computation*, December 2002.
- [4] Deok-Hwa Kim, Yong-Ho Yoo, Si-Jung Ryu, Woo-Young Go, and Jong-Hwan Kim. Automatic color detection for mirosot using quantum-inspired evolutionary algorithm.
- [5] Ye-Hoon Kim and Jong-Hwan Kim. Multiobjective quantum-inspired evolutionary algorithm for fuzzy path planning of mobile robot. *IEEE*, 2009.
- [6] Zhiyong Li, Bo Xu, Lei Yang, Jun Chen, and Kenli Li. Quantum evolutionary algorithm for multi-robot coalition formation. *ACM*, June 2009.
- [7] Amgad M. Mohammed, N. A. Elhefnawy, Mahmoud M. El-Sherbiny, and Mohiy M.Hadhoud. Quantum crossover based quantum genetic algorithm for solving non-linear programming. *The 8th International Conference on INFOrmatics and Systems (INFOS2012)*, May 2012.
- [8] Ajit Narayanan and Mark Moore. Quantum-inspired genetic algorithms. *IEEE International Conference on Evolutionary Computation*, May 1996.
- [9] In-Won Park, Ki-Baek Lee, and Jong-Hwan Kim. Multi-objective evolutionary algorithm-based optimal posture control of humanoid robots. *WCCI 2012 IEEE World Congress on Computational Intelligence*, June 2012.
- [10] Si-Jung Ryu, Jong-Hwan Kim, and Ki-Baek Lee. Dmqea: Dual multiobjective quantum-inspired evolutionary algorithm.
- [11] Yangguang Sun and Mingyue Ding. Quantum genetic algorithm for mobile robot path planning. *Fourth International Conference on Genetic and Evolutionary Computing*, 2010.
- [12] SHEN Xiaoning. A quantum evolutionary algorithm for robot path planning in dynamic environment. *Proceedings of the 32nd Chinese Control Conference*, July 2013.
- [13] Gwo-Ruey Yu and Lun-Wei Huang. Design of lmi-based fuzzy controller for robot arm using quantum evolutionary algorithms. *Fourth International Conference on Innovative Computing, Information and Control*, 2009.