

一個快速的量子演化法應用於資料分群問題

廖宥勳

國立中山大學
資訊工程學系

m993040087@student.nsysu.edu.tw

蔡崇煒

嘉南藥理科技大學
資訊多媒體應用系

cwtsai0807@gmail.com

江明朝

國立中山大學
資訊工程學系

mcchiang@cse.nsysu.edu.tw

摘要

量子演化法應用於最佳化問題的設計，是基於模擬量子力學相關概念，與整合演化計算方式，所設計的搜尋演算法。但目前這類研究所遇到的問題是，此搜尋演算法需大量的計算時間，求得較好品質的近似解。本研究將著重於發展一個改良方法，稱作快速量子演化法，用來解決傳統量子演化法應用在最佳化問題時，於短時間內難以收斂至較佳解的問題。為了評估快速量子演化法的效能，我們在本文中將以資料分群問題作為效能評估之工具。

關鍵詞：量子演化法、快速量子演化法、分群

Abstract

Quantum-inspired evolutionary algorithm (QEA) applied on optimization problems, the quality will be good enough based on the quantum simulated method. Once, however iterations are too short to converge, the good quality is missing. We proposed an improved quantum-inspired evolutionary algorithm, called fast-quantum-inspired evolutionary algorithm (FQEA). Through the advantages of exploration and exploitation, the FQEA provides higher convergence efficiency and robustness compared to QEA and K-means Algorithm when solving some clustering problems.

Keywords: Quantum-inspired evolutionary algorithm (QEA) and clustering

1 簡介

量子演化法 (quantum-inspired evolutionary algorithm; QEA) 是一種運用量子力學相關計算概念，並整合演化式演算法，所發展出的新型演化式演算法。量子演算法一般而言經常被用以增進演化式演算法解決問題的能力[2]。初期提出相關研究報告的 Han and Kim，在他們的研究 [6][7][2]當中，運用了量子位元 (quantum bit)、量子線性疊加態

(superposition) 測量模擬、以及量子個體 (quantum individual) 等概念。目前在許多相關領域也已經開始透過這樣計算方式，來尋求突破並得到一定的成效 [15]，例如：多目標最佳化問題 [14]、組合最佳化問題 [2]、人臉辨識 [11]、旅行銷售員問題，等真實世界的難題等 [2][3][4][12]。

在這些傳統的最佳化問題中，分群問題一直以來，都是非常重要的一項研究。現今面臨數位化的時代，龐大的資訊量往往分門別類地儲存在各個資訊中心當中，要分析如此巨大的資料量並從中擷取有利資訊，分群是最具直觀性且有效性的方法之一。

由於量子演化式方法，是近期所提出的最佳化搜尋方法，有關於量子演化法應用在分群問題上的研究 [15]的部分，目前仍顯不足。因此量子演化法應用在分群問題的研究，目前尚有許多需要討論的空間。在研究過程當中，我們發現傳統量子演化法應用在分群問題時，需要有足夠的代數 (1000 代以上) 演化，才能比較有機會得到較佳的解品質，

本研究中將著重於發展改良量子演化法之收斂過程，稱作快速量子演化法 (fast quantum-inspired evolutionary algorithm; FQEA)，主要用來解決傳統量子演化法應用在分群最佳化問題時，因代數不足而難以得到較佳的解品質所造成的問題。主要概念是於傳統量子演化法的步驟中加入區域搜尋的機制，用以減少過多的隨機性對收斂速度帶來的影響。

本文架構如下：第二章將概述量子運算原理與量子演化法，第三章將針對所提出的演算法作進一步的說明，包括如何實作、以及如何將其運用在資料分群問題上。第四章為實驗設計與實驗結果以及綜合分析討論。最後第五章為結論以及未來展望。

2 量子運算與量子演化法

量子演化法主要是基於機率模型表示法的增強式演化式演算法 [2][3][5]。不同於在傳統數位電腦當中，基本資訊儲存單位是二進位表

示法的位元 (bit, 其狀態非 0 即 1), 在二進位的量子電腦當中, 基本資訊儲存單位稱為量子位 (quantum bit 或 qubit) [9]。量子位的狀態處於 0 和 1 這兩者之間的任意線性疊加狀態 (superposition)。一個量子位狀態的定義如下:

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad (1)$$

其中 $|\alpha|^2 + |\beta|^2 = 1$ 。係數 α 與 β 皆為複數, 分別代表相關狀態的機率幅 (probability amplitudes), 機率幅的絕對值的平方代表某物理量的機率或機率密度 [1]。在此 $|\alpha|^2$ 與 $|\beta|^2$ 分別代表測量一個量子位的狀態時, 可量測到 0 狀態的機率與量測到 1 狀態的機率。

當進行觀測或量測一個量子位的狀態時, 其狀態會由線性疊加狀態塌縮 (collapse) 至一個穩定狀態, 使觀測結果始終以單一穩態呈現 [13]。

如下圖 1 所示, QEA 演算法分別以 $Q(t)$ 、 $P(t)$ 、 $B_{best}(t)$ 、 $G_{best}(t)$ 表示量子個體的集合即量子群體、量測的二元解集合、最佳的二元解集合、以及全域最佳解, 經由多次的疊代 t , 在解空間中, 搜尋較合適的解。

一般而言, 量子位元 (Q-bit)、量子個體 (Q-bit individual)、量子團隊 (quantum team) 及量子群體 (quantum swarm) 將會在上述的搜尋過程中被使用。為了更進一步說明量子演化法的流程 [2], 下列將分別介紹這些編碼方式。

Begin

$t \leftarrow 0$

(1) 初始化 $Q(t)$ 與 $B_{best}(t)$

while (未達中止條件) **do**

(2) 量測步驟: 對 $Q(t)$ 進行量測得到 $P(t)$

(3) 評估步驟: 評估 $P(t)$

(4) 更新步驟: 根據 $P(t)$ 與 $B_{best}(t)$ 比較並透過 Q -gate 對 $Q(t)$ 進行更新

(5) 儲存步驟: 將最好的 $B_{best}(t)$ 存入 $G_{best}(t)$

(6) 同步遷移步驟: 同步最佳解 $B_{best}(t)$ 或 $G_{best}(t)$

$t \leftarrow t + 1$

End while

End

圖 1 量子演化法

量子位元 (Q-bit)

QEA 效法量子運算當中量子位的概念, 在表示法的編碼方式採取了一種機率性表示法, 稱為量子位元 (Q-bit) 編碼。此為 QEA 中的最基本層級。一個量子位元可以表示為:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \quad (2)$$

其中 $|\alpha|^2 + |\beta|^2 = 1$ 。 $|\alpha|^2$ 與 $|\beta|^2$ 之意義如前所述。

量子個體 (Q-bit individual)

QEA 中最重要的角色就是量子位元個體 (Q-bit individual), 多個量子位元經由串連組合構成一個量子位元個體。此為 QEA 當中個體層級的最基本元素, 一個長度為 m 的量子位元個體由 m 個量子位元組成, 可表示為:

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_m \\ \beta_1 & \beta_2 & \cdots & \beta_m \end{bmatrix} \quad (3)$$

量子團隊 (Quantum team)

一個量子團隊 (quantum team) 的組成基本元素是量子個體, 每個團隊由 g 個量子個體組成。量子團隊的一個重要特徵就是在每個團隊當中都存在一個最佳解, 記為 $B_{best}^l(t)$ 。此最佳解是經由該團隊中的所有量子個體在每一代競爭後所得出的最佳解, 其目的是引導搜尋方向朝更好的方向搜尋。

量子群體 (Quantum swarm)

量子群體 (quantum swarm) 是 QEA 編碼的最上一層, 包含多個量子團隊。而每個量子團隊是由 g 個量子個體組成。假設一個量子群體由 n 個量子個體組成, 則此量子群體包含的量子團隊個數即為 (n/g) 個。在一個量子群體當中, 只存在唯一一個最佳解, 記為 G_{best} 。此最佳解是由所有量子團隊最佳解當中選取。因此, 根據此 QEA 的編碼方式, 可以假設在第 t 代時, 量子群體狀態表示為:

$$Q(t) = \{q_1(t), q_2(t), \dots, q_j(t), \dots, q_n(t)\}, \quad (4)$$

$$1 \leq j \leq n$$

其中第 j 個量子個體 $q_j(t)$ 狀態可描述為:

$$q_j(t) = \begin{bmatrix} \alpha_j^1(t) & \alpha_j^2(t) & \cdots & \alpha_j^i(t) & \cdots & \alpha_j^m(t) \\ \beta_j^1(t) & \beta_j^2(t) & \cdots & \beta_j^i(t) & \cdots & \beta_j^m(t) \end{bmatrix}, \quad (5)$$

$1 \leq j \leq n, \quad 1 \leq i \leq m$

在圖 1 中的量子邏輯閘 (Quantum gate, $Q-gate$)，是量子演化法當中的更新機制，其作用相當於傳統演化式演算法的交配 (crossover) 與突變 (mutation) 機制。量子邏輯閘作用於量子個體並對每一個量子位元作運算，以達到更新的效果。第 t 代的第 j 個量子個體的第 i 個量子位元的更新可以如下表示：

$$\begin{bmatrix} \alpha_j^i(t') \\ \beta_j^i(t') \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \begin{bmatrix} \alpha_j^i(t) \\ \beta_j^i(t) \end{bmatrix}, \quad (6)$$

$1 \leq j \leq n, \quad 1 \leq i \leq m, \quad t' \equiv t+1$

其中關於旋轉角度 $\Delta\theta$ 的選用可參考[2]。

3 快速量子演化法

3.1 概述

為了避免在代數時間有限之情況下，無法再找尋到一個較佳解的情況，本研究透過修改其檢查合法解的策略以達成改善傳統量子演化法 (QE) 的目標。將在檢查合法解的策略中引入區域搜尋的概念，用來判斷經過量測步驟所產生的解是否為合理的解。若判斷為不合理的解，則採取「區域搜尋概念」原則，直接針對不合理的點作修正，調整至當前區域最佳的合法解內容，即可進入下一階段。

值得注意的是此處的「區域搜尋概念」會隨著所選定的問題與評估函數的不同，而有不一樣的作法與定義。例如：應用在分群問題上，評估函數為計算 SSE (sum of squared error) 時，一旦偵測到不合理的點，例如一個資料點同時被分配到兩個群以上，則採取直接將此點分配到距離此點最接近質心的作法。

如圖 2 所示，在這個章節裡將提出一個改良量子演化法的方式，稱為快速量子演化法 (fast quantum-inspired evolutionary algorithm; FQEA)。下列將說明如何將 FQEA 進行適當的轉換，應用至分群問題。

Begin

$t \leftarrow 0$

(1)初始化 $Q(t)$ 與 $B_{best}(t)$

while (未達中止條件) **do**

(2)量測步驟: 對 $Q(t)$ 進行量測得到 $P(t)$

(3)檢查步驟: 以區域搜尋概念檢查 $P(t)$
是否合法

(4)評估步驟: 評估 $P(t)$

(5)更新步驟: 根據 $P(t)$ 與 $B_{best}(t)$ 比較並
透過 $Q-gate$ 對 $Q(t)$ 進行更新

(6)儲存步驟: 將最好的 $B_{best}(t)$ 存入 $G_{best}(t)$

(7)同步遷移步驟:

Local: 在 Team 層級內同步 $B_{best}(t)$

Global: 在 Swarm 層級內同步 $G_{best}(t)$

$t \leftarrow t+1$

End while

End

圖 2 快速量子演化法

3.2 快速量子演化法

編碼

在 FQEA 中，機率模型：Population (quantum swarm)，是每個量子群體由 n 個量子個體組成。Individual (quantum individual)，每個量子個體由 m 個量子位元組成。其中 $m = |clusters| \times |patterns|$ ， $|clusters|$ 代表欲求得的目標分群數， $|patterns|$ 代表資料點的個數。透過此方式編碼，可以順利將分群問題轉換成二元表示法，再透過快速量子演化法 (FQEA) 來求解資料分群問題。

評估函數

Pattern 之間的相似度評量採用 Euclidean distance ($p = 2$):

$$d_p(X_i, X_j) = \sum_{k=1}^d |x_{i,k} - x_{j,k}|^p \quad (7)$$

並透過 Sum of squared error (SSE) 來評估分群結果的品質，如表 1:

$$w_{ij} = \begin{cases} 1, & \text{if } x_i \in C_j \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

$$SSE = \sum_{i=1}^n \sum_{j=1}^m w_{ij} \|x_i - C_j\|^2 \quad (9)$$

表 1 評估函數說明

Sum of squared error (SSE)	
m	: 分群數目
n	: 樣式數目
x_i	: 第幾個樣式
C_j	: 第幾群
w_{ij}	: 用來判斷樣式是否在群內

解的表示法

分群的解採用 cluster ID 編碼，每個資料點 (pattern) 對應到所屬的分群編號，二元解中的 $G_{best}(t)$ 即代表分群的結果。透過對應再將二元解轉換成為分群編號即可得到分群解的表示法。

3.3 實例說明

1. 初始化 $Q(0)$: 首先初始化 $Q(0)$, $\alpha = \sqrt{1/2}, \beta = \sqrt{1/2}$ 說明在初始階段觀測到 1 狀態或是 0 狀態的機率都是相同的 1/2。接著初始化質心位置，作法是由資料點 (patterns) 中任意挑一個點來當作初始質心。
2. 量測步驟 (make $P(t)$) : 藉由量測量子位元的值用以產生二元解 $P(t)$ 。作法採 QEA 中提到的方式：

$$\begin{aligned} & \text{if } \text{random}[0,1) < |\beta_i|^2 \\ & \text{then } x_i \leftarrow 1 \\ & \text{else } x_i \leftarrow 0 \end{aligned} \quad (10)$$
3. 檢查步驟 (Repair $P(t)$) : 此為本方法的設計核心。利用區域搜尋概念原則來判斷經過量測步驟所產生的解是否為合理的解，以確保每個資料點只會分配到一個群。其作法是當偵測到不合理的點時，例如一個資料點同時被分配到兩個以上的群時，採取直接將此點分配到距離最近的質心的方式。
4. 分群步驟 (clustering) : 根據二元解 $P(t)$ 來進行分群。經過 repair() 步驟可確保每個資料點只可能被分到一個群中。而 $P(t)$ 代表的就是分群的結果。假設第 k 群不包含任何點，則隨機挑選一個包含兩個點以上的任一群，從中隨機選出一點放到第 k 群當中，使得分群結果不會產生空群。做完空群調整後必須再次調整 $P(t)$ ，以確保二元解與分群結果一致。確定沒有空

群之後即代表這份二元解為當然合法解，接著進入下一步驟。

5. 評估步驟: 計算總體的 Sum of squared error (SSE) 以作為評估 $P(t)$ 好壞的依據。
6. 更新步驟: 根據 $P(t)$ 與 $B_{best}(t)$ 比較的結果並透過 Q -gate 對 $Q(t)$ 進行更新。利用量子旋轉邏輯閘 (Rotation gate) $U(\Delta\theta_i)$ 來更新量子位元。其主要精神在於調整機率模型-量子群體- $Q(t)$ ，使得經由觀察 $Q(t)$ 所量測出來的 $P(t)$ 將隨著演化時間的推進而與 $B_{best}(t)$ 越來越相近。意即解的品質會愈來愈好。
7. 儲存步驟: 首先於量子團隊層級中各自比較 $B_{best}(t-1)$ 與 $P(t)$ ，再把比較好的結果存入 $B_{best}(t)$ 。再於量子群體層級中相互比較所有的 $B_{best}(t)$ 後，將最好的 $B_{best}(t)$ 存入 $G_{best}(t)$ 。
8. 區域同步遷移步驟 (local migration) : 主要作用於量子團隊層級，從每個 team 當中挑出一個最好的 $B_{best}(t)$ ，然後將該 team 的 $b_j(t)$ 全部都換成該 team 當中最好的那個 $B_{best}(t)$ 。(實驗參數設定: 每一代都做一次 local migration)
9. 全域同步遷移步驟 (global migration) : 用 $G_{best}(t)$ 把 $B_{best}(t)$ 中所有的解都取代掉 (Globally)。(實驗參數設定: 每 100 代做一次 global migration)

4 實驗結果

本章節將分析討論所提出的快速量子演化法 (FQEA) 對於效能上的影響。k-means [8][10] 及 quantum-inspired evolutionary algorithm (QEA) [2] 將被使用來與 FQEA 進行比較。實驗資料則來自 UCI (Machine Learning Repository)。

4.1 參數設定以及所使用的資料集

此部分將快速量子演化法應用在分群問題上，實驗中所需要的參數除了原本 QEA 的參數以及搭配欲分群資料集的特定參數之外，不需要再加上其他參數，而實驗目的主要是為了瞭解 FQEA 應用在分群問題上的收斂速度快

表 2 參數設定

$m = K \times P$ ， K 代表欲求得的目標分群數， P 代表資料點的個數、 D 代表資料維度。

Population Size	10
Length of chromosome	m
Iteration	1000
Round	30
Global migration period	100
Number of Clusters	K
Dimension	D
Rotation Angle	0.01π
Logical group size	2

表 3 資料集

Benchmark	K	D	P
Ecoli	8	7	336
Iris	3	4	150
Yeast	10	8	1484
Wine	3	13	178
Abalone	29	8	4177
SPECT	2	22	267

表 4 k-means、QEA、FQEA 的解品質(SSE)

Benchmark	k-means	QEA	FQEA
Ecoli	15.220	41.324	14.194
Iris	102.558	131.193	78.941
Yeast	50.389	116.831	46.233
Wine	49.856	59.320	48.962
Abalone	35.678	654.596	35.075
SPECT	1009.485	1021.874	1010.951

慢與最終解品質的好壞，參數設定如表 2。實驗程式是以 C++ 所撰寫並且使用編譯器 g++ 編譯。實驗所使用的硬體資訊為 2.67GHz、Intel-Core i7 CPU、4GB 記憶體、Linux 核心版本為 2.6.31.5、而所使用的資料集如表 3，樣式數最小從 150 點到 4177 點，維度最小從 4 維到 22 維，群數則是從 2 群到 29 群，而這些資料集各自具有不同的特性，如 Abalone 這個資料集為較大群數及樣式數，而 Iris 則非線性分布的資料集，本實驗將採用這些資料集來進行測試。

4.2 解的品質分析

本實驗比較三種演算法在解品質上的差異，表 4 說明 k-means、QEA、以及 FQEA 應用在六個分群問題資料集的實驗結果。實驗採用 Sum of squared error (SSE) 為評估標準

以分析分群結果的品質。根據表 4 的數據顯示，FQEA 在 Ecoli、Iris、Yeast、Wine、Abalone 的 SSE 表現皆優於 k-means 與 QEA。其中值得一提的現象是 QEA 不管在哪一個資料集所得到的 SSE 都不甚理想。實驗結果顯示：當演化代數為有限次的情況下，FQEA 能大幅且有效地改善傳統 QEA 在解品質的表現。

4.3 收斂分析

經由本實驗結果也顯示 FQEA 在收斂趨勢的效果比其他兩種分群演算法好。觀察圖 3 及圖 4 的收斂趨勢圖可發現，FQEA 曲線之收斂趨勢於一開始時會介於 QEA 與 k-means 之間，隨著代數的推移，FQEA 的曲線逐漸收斂，途中與 k-means 曲線交叉，最終收斂在優於 k-means 曲線之處。這代表 FQEA 在此類問題上，比其他兩種演算法更能達到深度與廣度的搜尋平衡。QEA 曲線雖顯示其優異的探勘能力，但卻無法在代數時間結束前找到有效的收斂趨勢，而 k-means 曲線則是顯示出極容易陷入局部最佳解的趨勢。

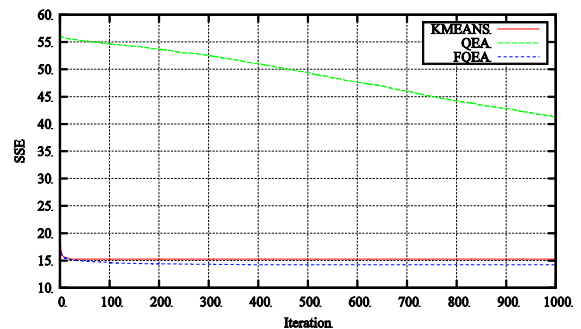


圖 3 收斂曲線(資料集：Ecoli)

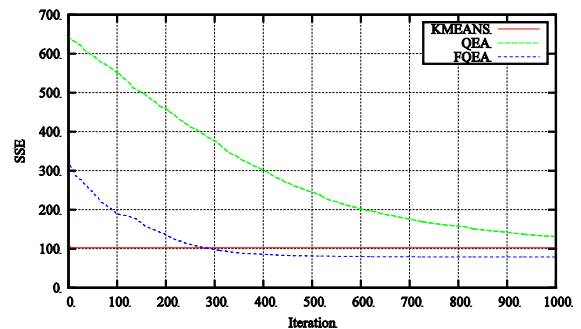


圖 4 收斂曲線(資料集：Iris)

5 結論以及未來展望

本論文提出了一個有效的解決方案，稱為快速量子演化法 (FQEA)，其改良量子演化法在檢查合理解的步驟，在一定的代數時間內可以有效加快收斂速度並提升最終解的品質。將快速量子演化法應用於分群問題的結果顯示，與傳統量子演化法 (QEA) 和 k-means 演算法相較之下，快速量子演化法展現出良好的實驗結果且能有效提升傳統量子演化法的性能。未來的研究方向將會整合跳脫機制，提升快速量子演化法找到更好的解的機會。同時也將可尋求應用於更多的現實生活問題，讓此方法，可應用至更多領域。

誌謝

本論文是國科會計畫(101-2221-E-041-012)研究成果的一部份，我們在此感謝國科會經費支持這個計畫的研究。

參考文獻

- [1] Born, M., “The Statistical Interpretation of Quantum Mechanics,” *Nobel Lecture*, 1954.
- [2] Han, K-H., and Kim, J-H., “Quantum-inspired evolutionary algorithm for a class of combinatorial optimization,” *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 6, pp. 580-593, 2002.
- [3] Han, K-H., and Kim, J-H., “Quantum-inspired evolutionary algorithms with a new termination criterion, H_ϵ gate, and two phase scheme,” *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 2, pp. 156-169, 2004.
- [4] Han, K-H., and Kim, J-H., “On setting the parameters of quantum-inspired evolutionary algorithm for practical application,” in *Proceedings of the IEEE Congress on Evolutionary Computation*, CEC 2003, Vol. 1, pp. 178-184, 2003.
- [5] Han, K-H., and Kim, J-H., “On the analysis of the quantum-inspired evolutionary algorithm with a single individual,” in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 9172-9179, 2006.
- [6] Han, K-H., and Kim, J-H., “Genetic Quantum Algorithm and its Application to Combinatorial Optimization Problem,” in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 1354-1360, 2000.
- [7] Han, K-H., and Kim, J-H., “Analysis of Quantum-inspired Evolutionary Algorithm,” in *Proceedings of the International Conference on Artificial Intelligence*, Vol. 2, pp. 727-730, 2001.
- [8] Hartigan, J. A., and Wong, M. A., “Algorithm as 136: A k-means clustering algorithm,” *Royal Statistical Society*, Vol. 28, No. 1, pp. 100-108, 1979.
- [9] Hey, T., “Quantum computing: An introduction,” *Computing & Control Engineering Journal*, Vol. 10, No. 3, pp. 105-112, 1999.
- [10] Macqueen, J. B., “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the Fifth Berkeley Symposium on Math, Statistics, and Probability*, Vol. 1, pp. 281-297, University of California Press, 1967.
- [11] Kim, J-H., Jang, J-S., and Han, K-H., “Face detection using quantum-inspired evolutionary algorithm,” in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 2100-2106, 2004.
- [12] Kima, K-H., Hwang, J-Y., Han, K-H., Kim, J-H., and Park, K-H., “A quantum-inspired evolutionary computing algorithm for disk allocation method,” *IEICE Transactions on Information and Systems*, Vol. E86-D, No. 3, pp. 645-649, 2003.
- [13] Narayanan A., “Quantum computing for beginners,” in *Proceedings of the IEEE Congress on Evolutionary Computation*, CEC 1999, Vol. 3, pp. 2231-2238, 1999.
- [14] Talbi, H., Draa, A., and Batouche, M., “A novel quantum-inspired evaluation algorithm for multi-source affine image registration,” *The International Arab Journal of Information Technology*, Vol. 3, No. 1, pp. 9-15, 2006.
- [15] Zhang, G., “Quantum-inspired evolutionary algorithms: a survey and empirical study,” *Journal of Heuristics*, Vol. 17, No. 3, pp. 303-351, 2011.