



یک روش مبتنی بر بازمقداردهی به جامعه جواب در الگوریتم‌های تکاملی کوانتوم برای افزایش تنوع در جامعه جواب

* محمد طبرانی محمد رضا اکبرزاده توتونچی

* دانشگاه آزاد اسلامی واحد مشهد، دانشگاه فردوسی مشهد

* akbarzadeh@ieee.org, tayarani@ieee.org

چکیده - الگوریتم‌های ژنتیک کوانتوم از ساختاری احتمالاتی برای نمایش پاسخ‌ها استفاده می‌کنند و چنین نمایشی موجب افزایش تنوع در جامعه جواب می‌شود. ولی در مجموع این الگوریتم‌ها نیز به خاطر استفاده از کوانتوم گیت و حرکت جمعی پاسخ‌ها به سوی بهترین پاسخ دچار مشکل گیر کردن در قله محلی، و کندی سرعت همگرایی هستند. در اینجا برای افزایش کارایی الگوریتم‌های تکاملی کوانتومی روشی را ارائه می‌کنیم که در آن هرگاه الگوریتم در قله محلی گرفتار شد با مقداردهی تصادفی به کوانتوم-کروموزوم‌ها، الگوریتم را بازمقدار دهی می‌کنیم. نتایج روی مسئله کوله‌پشتی، *Trap* و توابع عددی نشان می‌دهد که روش پیشنهادی هر چند از نظر پیچیدگی زمانی برابر با الگوریتم تکاملی کوانتوم ساده می‌باشد ولی از نظر کارایی برتری قابل ملاحظه‌ای نسبت به الگوریتم‌های تکاملی کوانتوم دارد.

کلیدواژه- الگوریتم‌های تکاملی، الگوریتم‌های ژنتیک کوانتومی، الگوریتم‌های کوانتومی

۱- مقدمه

موجب طولانی شنزهان رسیدن به پاسخ مطلوب [۴].

برای افزایش تنوع جامعه جواب و افزایش قدرت جستجو، در الگوریتم‌های تکاملی روش‌های گوناگونی پیشنهاد شده است. یکی از این روش‌ها بازمقداردهی به جامعه جواب است. در [۵] یک روش مبتنی بر بازمقداردهی برای الگوریتم‌های ژنتیک ارائه شده است که به محض درگیر شدن الگوریتم در قله محلی، جامعه جواب بازمقداردهی می‌شود. برای حل مسئله انتخاب سنسور، [۶] یک الگوریتم ژنتیک مبتنی بر بازمقداردهی ارائه نموده است که در آن بازمقداردهی بر اساس یک تحلیل آماری روی جامعه جواب صورت می‌گیرد. یک الگوریتم ژنتیک با یک جامعه جواب خیلی کوچک و یک فرایند بازمقداردهی در [۷] پیشنهاد شده است که از سه شیوه نخبه‌گرایی استفاده می‌کند. روش پیشنهادی در این مرجع از نظر محاسباتی کارایی بسیار بالایی دارد. در [۸] یک الگوریتم ژنتیک پیشنهاد شده است که در آن اندازه جامعه جواب طبق یک تابع دندان‌اره‌ای

الگوریتم‌های تکاملی، الگوریتم‌های جستجوی تصادفی هستند که از نظریه تکاملی داروین الهام گرفته‌اند [۱]. این الگوریتم‌ها در مقایسه با روش‌های دیگر بهینه‌سازی مانند روش‌های محاسباتی و روش‌های مبتنی بر گرادیان، مقاوم‌تر هستند. بسیاری روش‌های دیگر، از مشکل گیر کردن در قله‌های محلی رنج می‌برند، حال آنکه الگوریتم‌های تکاملی به علت جستجوی موازی مبتنی بر جمعیت تا حدی با این مشکل کنار آمده‌اند [۲].

در الگوریتم‌های تکاملی برای ایجاد توازن میان تعمق، در برابر جستجو، بایستی مقادیر پارامترهای الگوریتم مانند نرخ برش، نرخ جهش، اندازه جامعه را به درستی تعیین کرد. تعمق به معنی تأکید بر پاسخ‌های موجود است و جستجو به معنی تلاش برای یافتن پاسخ‌های جدید. افزایش تعمق، موجب گیر افتادن در قله محلی می‌شود و افزایش جستجو

۲- الگوریتم‌های تکاملی کوانتوم

کوچک‌ترین واحد اطلاعاتی که در یک کامپیوتر کوانتومی ذخیره می‌شود کوانتوم-بیت نامیده می‌شود. یک کوانتوم-بیت ممکن است مقدار "۱" یا "۰" به خود بگیرد. حالت یک کوانتوم بیت را می‌توان به صورت زیر نمایش داد:

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

در حالی که α و β اعداد مختلطی هستند که دامنه احتمالی حالات مربوط را مشخص می‌کند. $|\alpha|^2$ احتمال اینکه کوانتوم-بیت مقدار "۰" به خود بگیرد و $|\beta|^2$ احتمال اینکه کوانتوم-بیت مقدار "۱" به خود بگیرد را مشخص می‌کند. با توجه به اینکه کوانتوم-بیت یکی از دو حالت "۰" یا "۱" را خواهد گرفت داریم:

$$|\alpha|^2 + |\beta|^2 = 1 \quad (2)$$

الگوریتم‌های ژنتیک کوانتوم از نمایش پاسخ‌ها در قالبی احتمالاتی استفاده می‌کنند. هر کوانتوم-کروموزوم q_j^t در فضای جواب با n کوانتوم-بیت نمایش داده می‌شود:

$$q_j^t = \begin{bmatrix} \alpha_{j1} & \alpha_{j2} & \dots & \alpha_{jn} \\ \beta_{j1} & \beta_{j2} & \dots & \beta_{jn} \end{bmatrix} \quad (3)$$

که در آن $j = (1, 2, \dots, m)$ شماره کوانتوم کروموزوم را در جامعه جواب مشخص می‌کند، m معرف تعداد کروموزوم‌ها در جامعه جواب است، n تعداد کوانتوم-بیت‌های کوانتوم-کروموزوم یا همان بُعد مسئله بهینه‌سازی را نشان می‌دهد و t معرف شماره نسل الگوریتم تکاملی می‌باشد. در واقع $q_j^t = [\alpha_{ji}^t \ \beta_{ji}^t]^T$ کوانتوم بیت i ام از کروموزوم j ام در نسل t ام را نشان می‌دهد. مزیت اصلی نمایش داده‌ها با کوانتوم-بیت این است که یک کروموزوم در جامعه جواب می‌تواند مقادیر مختلفی به خود بگیرد. برای نمونه کوانتوم-کروموزوم زیر را در نظر بگیرید:

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{\sqrt{3}}{2} \end{bmatrix} \quad (4)$$

حالات این کوانتوم-کروموزوم می‌تواند به شکل زیر نمایش داده شود:

نوسان می‌کند و جامعه جواب طبق یک روال تکراری بازمقداردهی می‌شود. برای حل مسئله جایگذاری سلول در فرایند طراحی تراشه، مرجع [۹] یک روش تطبیقی برای اندازه جامعه جواب پیشنهاد کرده است که در آن اندازه جامعه جواب به شکلی تطبیقی تغییر می‌کند. در بسیاری مسائل بهینه‌سازی، محیط دائماً در حال تغییر است. در این شرایط الگوریتم باید به گونه‌ای طراحی شود که خود را با تغییر محیط تطبیق دهد. یکی از بهترین راه‌ها برای حل این مسئله، بازمقداردهی به جامعه جواب است. در [۱۰] برای حل مسائل بهینه‌سازی در محیط‌های متغیر، یک الگوریتم ژنتیک مبتنی بر بازمقداردهی ارائه شده است. بازمقداردهی همچنین در الگوریتم‌های هوش جمعی گروه ذرات PSO مورد استفاده قرار گرفته است. در [۱۱] یک الگوریتم بازمقداردهی برای سرعت ذرات پیشنهاد شده است. در [۱۲] از الگوریتم هوش گروه ذرات برای یافتن کوتاهترین مسیرها در گراف استفاده شده است. کارایی الگوریتم به کمک بازمقداردهی جامعه جواب به همراه یک الگوریتم جستجوی محلی، افزایش یافته است.

الگوریتم‌های ژنتیک کوانتوم (QGA) که اخیراً معرفی شده‌اند [۳] در مقایسه با الگوریتم‌های ژنتیک مرسوم توانایی زیادی برای ایجاد توازن میان تعمق و جستجو دارند [۴]. در الگوریتم‌های ژنتیک کوانتوم پس از همگرایی الگوریتم، کروموزوم‌های جامعه جواب به مقادیر نزدیک به هم همگرا می‌شوند. این همگرایی به معنی گیر افتادن الگوریتم در یک قله است. این قله ممکن است یک قله محلی و یا یک قله سراسری باشد. در این مرحله گریز الگوریتم از قله محلی با دشواری صورت می‌گیرد. در اینجا روشی را پیشنهاد می‌کنیم که با بازمقداردهی به جامعه جواب موجب بهبود کارایی الگوریتم ژنتیک کوانتوم می‌شود. در الگوریتم پیشنهادی ما (Reinitialization QEA) هر بار اجرای الگوریتم، شرط همگرایی آن بررسی می‌شود. هرگاه الگوریتم به همگرایی رسید، برای گریز از قله محلی جامعه جواب را باز مقداردهی می‌کنیم.

در بخش ۲ الگوریتم‌های ژنتیک کوانتوم را معرفی می‌کنیم، در بخش ۳ الگوریتم پیشنهادی را مطرح می‌کنیم و در بخش ۴ نتایج آزمایشی را ارائه می‌کنیم.

$$\begin{aligned} & \frac{1}{4}|000\rangle + \frac{\sqrt{3}}{4}|001\rangle - \frac{1}{4}|010\rangle - \frac{\sqrt{3}}{4}|011\rangle \\ & + \frac{1}{4}|100\rangle + \frac{\sqrt{3}}{4}|101\rangle - \frac{1}{4}|110\rangle \\ & - \frac{\sqrt{3}}{4}|111\rangle \end{aligned} \quad (5)$$

کروموزوم دودویی x_j^0 ، $j = 1, 2, \dots, n$ یک پاسخ دودویی به طول n است. ایجاد یک کروموزوم دودویی x_j^t از روی کوانتوم-کروموزوم با استفاده از عمل مشاهده انجام می‌گیرد. برای ایجاد یک بیت دودویی x_{ji}^t از روی یک کوانتوم-بیت $[\alpha_{ji}^t \ \beta_{ji}^t]^T$ ، از معادله ۶ استفاده می‌کنیم درحالی‌که $U(\cdot, \cdot)$ یک تابع تولید کننده اعداد تصادفی یکنواخت می‌باشد:

$$x_{ji}^t = \begin{cases} 0 & U(0,1) < \alpha_{ji}^t{}^2 \\ 1 & otherwise \end{cases} \quad (6)$$

iii. ارزیابی مجموعه کروموزوم‌های دودویی بدست آمده از مرحله دوم توسط تابع برازندگی.

iv. بهترین جواب یافته شده در میان $X(0)$ در \mathbf{b} ذخیره می‌شود.

v. تا ارضای شرط خاتمه، الگوریتم اجرا می‌شود.

vi. جواب‌های دودویی $X(t)$ با مشاهده حالت‌های $Q(t-1)$ ایجاد می‌شوند.

vii. ارزیابی $X(t)$.

viii. به‌روزرسانی کوانتوم-بیت‌ها با استفاده از کوانتوم-گیت.

ix. ذخیره بهترین جواب یافته شده در \mathbf{b} .

۲-۱- کوانتوم-گیت

کوانتوم-گیت یک عملگر است که مقدار یک کوانتوم-بیت را به‌روز می‌کند. برای به‌روز کردن کوانتوم-بیت $[\alpha_{ji}^t \ \beta_{ji}^t]^T$ از کوانتوم-گیت $U(\cdot)$ استفاده می‌کنیم که معادله (۷) آن را نشان می‌دهد:

$$[\alpha_{ji}^{t+1} \ \beta_{ji}^{t+1}]^T = \begin{cases} U(\Delta\theta_i)[\alpha_{ji}^t \ \beta_{ji}^t]^T & \text{if } \alpha\beta > 0 \\ U(-\Delta\theta_i)[\alpha_{ji}^t \ \beta_{ji}^t]^T & \text{otherwise} \end{cases} \quad (7)$$

درحالی‌که $\Delta\theta_i, i = 1, 2, \dots, n$ زاویه دوران است:

$$U(\Delta\theta_i) = \begin{bmatrix} \cos(\Delta\theta_i) & -\sin(\Delta\theta_i) \\ \sin(\Delta\theta_i) & \cos(\Delta\theta_i) \end{bmatrix} \quad (8)$$

مقدار زاویه دوران، $\Delta\theta$ از جدول ۱ به‌دست می‌آید [۳].

از (۵) مشخص می‌شود که احتمال رویداد حالات $|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle$ به ترتیب برابر است با $1/16, 3/16, 1/16, 3/16, 1/16, 3/16, 1/16, 3/16$ همان‌طور که مشاهده می‌شود کوانتوم-کروموزوم (۴) توانایی ایجاد هشت حالت مختلف را داراست. این بدین معنی است که در الگوریتم‌های ژنتیک کوانتومی، هر کروموزوم اطلاعات موجود در جواب‌های زیادی را در خود نهفته دارد، بنابراین تنها یک کوانتوم-کروموزوم برای نمایش هشت حالت مختلف کافیست.

روال یک الگوریتم ژنتیک کوانتومی را می‌توان به صورت زیر نمایش داد [۳]:

Procedure QEA

```
begin
  t=0
  i. initialize Q(0)
  ii. make X(0) by observing Q(0)
  iii. evaluate X(0)
  iv. store the best solutions among X(0) into b
  v. while (not termination-condition) do
    begin
      t=t+1
      vi. make X(t) by observing the states of Q(t-1)
      vii. evaluate X(t)
      viii. update Q(t) using Q-gate
      ix. store the best solution among X(t) into b
    end
  end
```

مراحل اجرای الگوریتم فوق به شکل زیر است:

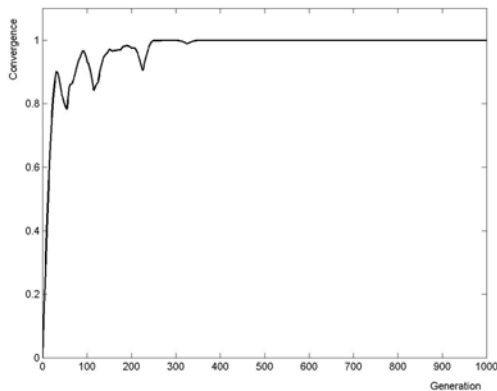
i. در مرحله مقدار دهی اولیه به $Q(0)$ ، به تمامی کوانتوم-بیت‌های α_{ji}^0 و β_{ji}^0 ($i = 1, 2, \dots, n$) برای همه کوانتوم-کروموزوم‌های q_j^0 ($j = 1, 2, \dots, m$) مقدار اولیه $1/\sqrt{2}$ داده می‌شود.

ii. این مرحله شامل ایجاد مجموعه کروموزوم‌های دودویی $X(0)$ از کوانتوم-کروموزوم‌های $Q(0)$ می‌باشد. در این

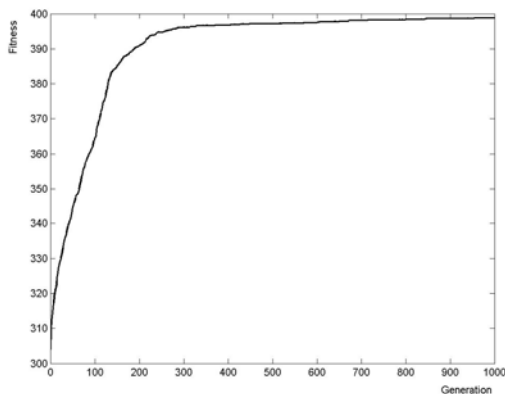
مرحله به‌سسته به مقادیر $|\alpha_{ji}^0|^2, |\beta_{ji}^0|^2$

کروموزوم‌های $(i = 1, 2, \dots, n)$ دودویی

$X(0) = \{x_1^0, x_2^0, \dots, x_m^0\}$ در نسل صفر ایجاد می‌شوند. یک



شکل ۱. روند همگرایی کروموزوم‌ها در طول فرایند تکامل. محور افق شماره نسل‌ها و محور قائم میزان همگرایی الگوریتم را نشان می‌دهد.



شکل ۲. روند تکامل الگوریتم

شکل ۲ روند تکامل الگوریتم را نشان می‌دهد. از مقایسه شکل ۱ و شکل ۲ پیداست که با همگرایی کوانتوم کروموزوم‌ها و قرارگیری الگوریتم در یک قله محلی، روند تکامل الگوریتم کندتر می‌شود. با بذل توجه به این موضوع می‌توان دریافت که مقدار دهی به الگوریتم پس از هر بار گیر افتادن الگوریتم در قله محلی، می‌تواند موجب بهبود کارایی الگوریتم شود.

برای این منظور الگوریتم زیر را پیشنهاد می‌کنیم:

```

Procedure Reinitialization-QGA
begin
  t = 0
  i. initialize  $Q(t)$ 
  ii. make  $P(t)$  by observing  $Q(t)$ 
  iii. evaluate  $P(t)$ 
  iv. store the best solutions among  $P(t)$  into b
  v. while (not termination-condition) do
    begin
  vi.   t = t+1
  vii.  if algorithm is converged then
        reinitialize the population

```

جدول شماره ۱. مقدار $\Delta\theta$ برای به‌روز رسانی کوانتوم-بیت. $f(x)$ معرف میزان برازش کروموزوم دودویی x و $f(b)$ میزان برازش کروموزوم دودویی b می‌باشد.

x_i	b_i	$f(x) \geq f(b)$	$\Delta\theta_i$
۰	۰	False	۰
۰	۱	True	۰
۰	۰	False	$۰,۰ \ ۱\pi$
۰	۱	True	۰
۱	۰	False	$-۰,۰ \ ۱\pi$
۱	۱	True	۰
۱	۰	False	۰
۱	۱	True	۰

۳- بازمقداردهی برای گریز از قله‌های محلی

در الگوریتم‌های ژنتیک کوانتوم پس از گذشت مدتی از اجرای الگوریتم، کوانتوم-بیت‌ها به مقادیر $[0 \ 1]^T$ یا $[1 \ 0]^T$ همگرا می‌شوند. در این شرایط الگوریتم به یک قله محلی همگرا شده است و گریز از آن دشوار می‌باشد. در الگوریتم‌های ژنتیک معمول، روش‌های مختلفی برای ایجاد تنوع بیشتر در الگوریتم و گریز از قله‌های محلی پیشنهاد شده است. یکی از این روش‌ها، بازمقداردهی به الگوریتم است [۸]. شکل ۱ میزان همگرایی جامعه جواب را در طول روند تکامل جامع نشان می‌دهد. اگر $Q = \{q_1, q_2, \dots, q_m\}$ جامعه جواب و $q_j = \begin{bmatrix} \alpha_{j1} & \alpha_{j2} & \dots & \alpha_{jn} \\ \beta_{j1} & \beta_{j2} & \dots & \beta_{jn} \end{bmatrix}$ کوانتوم-کروموزوم j ام از جامعه جواب باشد میزان همگرایی جامعه جواب از رابطه (۸) بدست می‌آید [۴]:

$$C = \frac{1}{n \times m} \sum_{i=1}^m \sum_{j=1}^n |1 - 2|\alpha_{ij}|^2| \quad (۸)$$

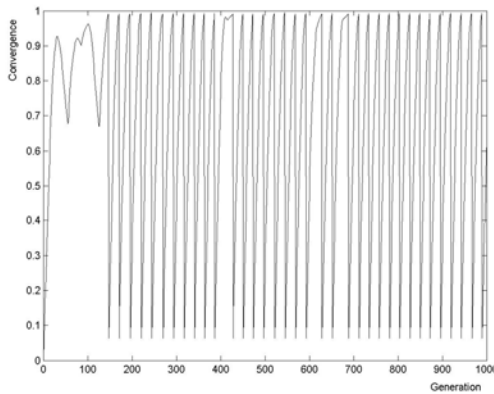
که در آن C میزان همگرایی جامعه جواب، m اندازه جامعه جواب و n اندازه هر کروموزوم است.

شکل ۱ همگرایی الگوریتم را برای حل مسئله کوله‌پشتی نشان می‌دهد. همان‌طور که از این نمودار پیداست پس از نزدیک ۲۰۰ نسل از اجرای الگوریتم، کوانتوم-کروموزوم‌ها همگرا می‌شوند و حول قله محلی به دام می‌افتند. در این شرایط با بازمقداردهی به الگوریتم می‌توان به کارایی الگوریتم افزود.

۴-۱- مقایسه کارایی الگوریتم پیشنهادی با

الگوریتم ژنتیک کوانتوم مرسوم

در الگوریتم پیشنهادی برای افزایش تنوع در جامعه جواب، پس از همگرا شدن جامعه، آن را با مقدار دهی می‌کنیم. شکل ۳ همگرایی در جامعه را در طی تکامل الگوریتم پیشنهادی نشان می‌دهد. همان‌طور که مشاهده می‌شود به محض همگرایی الگوریتم، جامعه جواب را با مقداردهی می‌کنیم. با این عمل جامعه جواب از قله محلی دور می‌شود.



شکل ۳. همگرایی در جامعه جواب در الگوریتم پیشنهادی

حفظ تنوع در جامعه جواب موجب می‌شود الگوریتم کمتر در دام قله‌های محلی بیفتد.

جدول شماره ۲ مقایسه روش پیشنهادی را با الگوریتم کوانتوم ژنتیک ساده نشان می‌دهد. همان‌طور که از این جدول پیداست روش پیشنهادی توانسته است به پاسخی بهتر نسبت به الگوریتم کوانتوم ژنتیک ساده دست یابد. الگوریتم پیشنهادی برای افزایش تنوع، جامعه جواب را پس از همگرایی با مقداردهی می‌کند. تست همگرایی الگوریتم تنها عملگری است که بر بار محاسباتی الگوریتم افزوده است. تست همگرایی الگوریتم نسبت به اندازه جامعه جواب از مرتبه زمانی خطی $O(n)$ است و نسبت به محاسبه تابع برازندگی از زمان اجرای کمتری برخوردار است. از این رو از نظر مرتبه زمانی تأثیری بر سرعت اجرای الگوریتم ندارد. الگوریتم پیشنهادی هر چند از نظر زمان اجرا پیچیدگی چندان بیشتری نسبت به نسخه مرسوم الگوریتم تکاملی کوانتوم ندارد ولی کارایی بسیار بالاتری از خود نشان می‌دهد.

- viii. make $P(t)$ by observing the states of $Q(t-1)$
 - ix. evaluate $P(t)$
 - x. update $Q(t)$ using Q-gate
 - xi. store the best solutions among $P(t)$ into \mathbf{b}
- end

end

در دستور VII همگرایی الگوریتم محاسبه می‌شود. اگر میزان همگرایی از مقدار ثابت δ بیشتر بود، جامعه جواب را با مقداردهی می‌کنیم.

هدف از دستور VII این است که تنها هنگامی جامعه را با مقداردهی کنیم که جامعه به اندازه کافی به یک پاسخ همگرا شده باشد و به کوانتوم-کروموزوم‌هایی که اخیراً ایجاد شده‌اند به اندازه کافی فرصت برای همگرا شدن داده شود. در این پژوهش مقدار $\delta=0.99$ در نظر گرفته شده است.

۴- نتایج آزمایشی

در تمامی آزمایشات تعداد اعضای جامعه برابر ۲۵ در نظر گرفته شده است. تعداد اجرا برای تمامی آزمایش‌ها برابر ۵۰ تنظیم شده است، نتایج حاصل، میانگین ۵۰ بار اجرای الگوریتم است و مقدار $\Delta\theta$ بر اساس جدول شماره ۱ تعیین شده است. برای مقایسه الگوریتم مطروحه و الگوریتم ژنتیک کوانتوم مرسوم، از مسئله کوله‌پشتی ۰-۱، مسئله Trap و ۱۴ تابع عددی استفاده نموده‌ایم. شیوه استفاده از الگوریتم‌های تکاملی کوانتوم برای حل مسئله کوله‌پشتی، Trap و توابع عددی در [۴] آمده است. در اینجا از چهار روش برای حل مسئله کوله‌پشتی استفاده نموده‌ایم دو روش مبتنی بر پنالته و دو روش مبتنی بر ترمیم. جزئیات این روش‌ها در [۴] ارائه شده‌اند. در بهینه‌سازی توابع عددی از رمزگذاری دودویی با طول کروموزوم‌ها برابر ۱۰ استفاده کرده‌ایم. مجموعه توابع عددی که برای ارزیابی الگوریتم پیشنهادی مورد استفاده قرار گرفته است در جدول شماره ۲ لیست شده‌اند. این توابع برخی دارای یک مینیمم سراسری و برخی دارای چندین مینیمم محلی و یک مینیمم سراسری هستند که با افزایش بُعد مسئله بر تعداد مینیمم‌های محلی به سرعت افزوده می‌شود. در تمامی آزمایش‌ها هدف یافتن ماکزیمم توابع است. برای همین منظور در آزمایشات تابع برازندگی برابر $-f(x)$ در نظر گرفته شده است.

جدول شماره ۲. نتایج آزمایشی روی مسئله کوله‌پشتی، Trap و ۱۴ تابع عددی. m بُعد مسئله است. KPI به معنی استفاده از تابع پنتالی ۱ برای مسئله کوله-پشتی و KR1 به معنی استفاده از تابع ترمیم ۱ برای مسئله کوله‌پشتی است. ضابطه هر یک از توابع را می‌توانید در مرجع ذکر شده بیابید.

Dimension	$m=100$				$m=250$				$m=500$				$m=1000$			
	QEA		RQEA		QEA		RQEA		QEA		RQEA		QEA		RQEA	
Function	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
KR1	398.8	6.9	415.3	1.8	908.8	20.8	967.2	9.6	1675.4	25.8	1780.7	15.4	3314.3	47.2	3492.7	38.3
KR2	418.0	2.2	421.2	0.64	996.7	16.9	1028.3	4.2	1839.8	26.3	1930.0	12.8	3637.8	82.5	3836.8	40.9
KP1	533.1	8.5	549.8	3.4×10^{-13}	1163.5	40.1	1261.1	12.6	2082.5	58.5	2275.7	34.1	3919.6	106.3	4250.3	61.0
KP2	403.4	6.4	416.8	1.9	908.2	20.8	970.2	11.3	1673.1	37.2	1778.6	19.3	3303.9	52.4	3503.7	37.2
Trap	71.2	4.3	83.6	1.9	158.3	9.8	191.9	4.0	278.2	16.5	330.5	4.6	539.9	31.8	570.2	9.6
Schwefel [13]	3.8×10^4	4.1×10^3	4.9×10^4	1.7×10^3	3.9×10^4	2.7×10^3	5.0×10^4	1.8×10^3	3.8×10^4	5.0×10^3	5.0×10^4	1.6×10^3	3.7×10^4	4.1×10^3	5.1×10^4	2.6×10^3
Rastrigin [13]	-1.9×10^3	2.4×10^2	-1.1×10^3	7.6	-1.7×10^3	2.6×10^2	-1.1×10^3	38.1	-1.9×10^3	1.8×10^2	-1.1×10^3	72.5	-1.9×10^3	3.7×10^2	-1.1×10^3	9.8×10^1
Ackley [13]	-17.9	9.7×10^{-2}	-17	4.6×10^{-2}	-17.9	1.1×10^{-1}	-17.6	4.6×10^{-2}	-17.9	0.11	-17.6	0.05	-18.0×10^2	9.6×10^2	-17.6	7.1×10^{-3}
Griewank [13]	1.2	-11.9	5.3	-35.9	-35.6	6.9	-11.9	1.3	-35.8	8.4	-10.8	0.9	-34.3	8.0	-11.5	1.06
Penalized 1 [13]	-1.5×10^5	1.3×10^4	-5.3×10^4	6.8×10^3	-1.4×10^5	3.7×10^4	-5.3×10^4	9.1×10^3	-1.4×10^5	2.5×10^4	-4.9×10^4	7.5×10^3	-1.4×10^5	2.8×10^4	-5.2×10^4	4.4×10^3
Penalized 2 [13]	-3.6×10^4	7.2×10^3	-1.4×10^4	1.8×10^3	-3.4×10^4	6.6×10^3	-1.5×10^4	2.3×10^3	-3.4×10^4	4.5×10^3	-1.3×10^4	1.6×10^3	-3.8×10^4	6.4×10^3	-1.3×10^4	1.6×10^3
Michalewicz [14]	25.5	2.53	35.7	2.26	25.1	3.2	35.5	2.4	26.9	3.4	34.8	2.6	27.3	1.9	35.1	2.7
Goldberg [8]	42.3	3.1	57.6	2.1	43.4	4.4	57.4	2.2	43.4	3.3	55.0	2.9	40.1	3.4	55.8	2.6
Sphere Model [13]	-3.7×10^5	6.5×10^4	-1.3×10^5	1.7×10^4	-4.0×10^5	5.8×10^4	-1.2×10^5	1.7×10^4	-3.9×10^5	9.5×10^4	-1.2×10^5	1.2×10^4	-4.2×10^5	6.9×10^4	-1.3×10^5	13×10^3
Schwefel 2.22 [13]	-4.7	0.42	-2.4	1.7×10^{-1}	-4.6	0.28	-2.4	0.2	-4.63	0.46	-2.48	0.16	-4.7	0.46	-2.4	0.17
Schwefel 2.21 [13]	-1.7×10^2	3.8	-1.4×10^2	8.9	-1.7×10^2	3.2	-1.4×10^2	8.6	-1.7×10^2	6.8	-1.4×10^2	12.2	-1.7×10^2	4.12	-1.4×10^2	9.9
Dejong [14]	-2.15×10^7	5.5×10^6	-2.6×10^6	6.8×10^5	-2.2×10^7	7.2×10^6	-2.7×10^6	4.9×10^5	-2.17×10^7	4.3×10^6	-2.31×10^6	6.6×10^5	-2.1×10^7	7.9×10^6	-2.8×10^6	9.2×10^5
Rosenbrock [8]	-8.5×10^4	2.8×10^4	-1.5×10^4	3.2×10^3	-8.9×10^4	1.9×10^4	-1.9×10^4	5.0×10^3	-1.0×10^5	3.9×10^4	-1.8×10^4	5.2×10^3	-8.2×10^4	1.9×10^4	-1.9×10^4	3.7×10^3
Kennedy [8]	-1.6	1.6	-1.5×10^{-2}	3.2×10^{-3}	-1.4	1.1	-1.1×10^{-2}	8.6×10^{-3}	-1.1	6.7×10^1	-1.7×10^{-2}	1.1×10^{-2}	-1.47	1.86	-1.5×10^{-2}	1.2×10^{-2}

- [4] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithm with a new termination criterion, H_g gate and two phase scheme" *IEEE Trans. Evol. Comput.* vol. 8, pp. 156–169, April, 2004.
- [5] Hisashi Shimodaira, "Methods for Reinitializing the Population to Improve the Performance of a Diversity-Control-Oriented Genetic Algorithm," *IEICE Transactions on Information and Systems* Vol. E84-D No.12 pp.1745-1755, 2001.
- [6] Héctor C. Goicoechea, and Alejandro C. Olivieri, Wavelength Selection for Multivariate Calibration Using a Genetic Algorithm: A Novel Initialization Strategy," pp 1146-1153, 2002.
- [7] Carlos A. Coello Coello, Gregorio Toscano Pulido, "A Micro-Genetic Algorithm for Multiobjective Optimization," *Springer, First International Conference Evolutionary Multi-Criterion Optimization*, 2001, p. 126, 2001.
- [8] Vlasios K. Koumoussis, Christos P. Katsaras. "A Saw-Tooth Genetic Algorithm Combining the Effects of Variable Population Size and Reinitialization to Enhance Performance," *IEEE Trans. Evol. Comput.* Vol. 10, No. 1 pp. 19-28, February 2006.
- [9] Al Qasim, Rose Eldos, Taisir, "Population Sizing Scheme for Genetic Algorithm," *IEEE International Conference on Computer Systems and Applications*, 2007.
- [10] Aimin Zhou, Yaochu Jin, Qingfu Zhang, Bernhard Sendhoff and Edward Tsang, "Prediction-Based Population Re-initialization for Evolutionary Dynamic Multiobjective Optimization," *Springer, bookchapter, Evolutionary Multi-Criterion Optimization*, pp. 832-846, 2007.
- [11] Ratnaweera, A. Halgamuge, S.K. Watson, H.C, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 3, pp. 240- 255. 2004.
- [12] Mohemmed, A.W. Sahoo, N.C, "Particle Swarm Optimization Combined with Local Search and Re-Initialization for Shortest Path Computation in Networks," *IEEE Swarm Intelligence Symposium*, pp. 266 – 272, 2007.
- [13] Weicai Zhong, Jing Liu, Mingzhi Xue "A Multiagent Genetic Algorithm for Global Numerical Optimization", *IEEE Trans on Cyber* Vol. 34, No. 2, pp. 1128-1141.
- [14] A.-R. Khorsand, M.-R. Akbarzadeh-T. "Quantum Gate Optimization in a Meta-Level Genetic Quantum Algorithm," *IEEE International Conference on Systems, Man and Cybernetics*, 2005.

۵- نتیجه‌گیری

در اینجا الگوریتم ژنتیک کوانتومی پیشنهاد کردیم که در آن، هنگام گیر افتادن الگوریتم در قله محلی، با بازمقدار دهی به الگوریتم موجب گریز آن از قله محلی می‌شویم. برای بررسی اینکه الگوریتم در قله محلی گیر افتاده است یا نه از ملاک همگرایی الگوریتم استفاده می‌کنیم. با این کار الگوریتم هیچگاه در دام قله محلی گرفتار نمی‌شود و می‌تواند در نهایت به پاسخی مناسب‌تر دست یابد. در این الگوریتم از ملاک همگرایی $\delta=0.99$ استفاده نمودیم. مقدار $\delta=0.99$ به معنی این است که ۹۹٪ کوانتوم گیت‌ها به مقادیر ۰ و ۱ همگرا شده‌اند. یکی از پژوهش‌های آینده می‌تواند تعیین مقدار مناسب برای ملاک همگرایی δ باشد. همچنین در مرحله بازمقدار دهی به جامعه جواب، از عملگر تصادفی استفاده می‌کنیم. به این معنی که اگر شرط بازمقدار دهی ارضا شود تمامی کوانتوم کروموزوم‌ها به شکل تصادفی مقدار می‌گیرند. استفاده از دانشی که در جستجوی پیشین بدست آمده است در بازمقدار دهی به الگوریتم می‌تواند از پژوهش‌های بعدی به حساب آید.

۶- مراجع

- [1] Randy L. Haupt, Sue Ellen Haupt, *Practical Genetic Algorithms. Second Edition*. Wiley.
- [2] Thomas Back, David B Fogel, Zbigniew Michalewicz, *Evolutionary Computation 1. Basic Algorithms and Operators*. IOP, 2000.
- [3] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Trans. Evol. Comput.* vol. 6, pp. 580–593, Dec. 2002.