

# الگوریتم ترکیبی یادگیری افزایشی مبتنی بر جمعیت، بهینه سازی حدی و اتوماتای یادگیر (PBIL-EO-LA)

میترا هاشمی<sup>1</sup>، محمد رضا میبدی<sup>2</sup>

<sup>1</sup> دانشگاه آزاد اسلامی قزوین، دانشکده برق رایانه و فن آوری اطلاعات، قزوین

Mitra.hash@yahoo.com

<sup>2</sup> آزمایشگاه سیستم‌های نرم دانشکده مهندسی کامپیوتر دانشگاه صنعتی امیر کبیر، تهران

Mmeybodi@aut.ac.ir

## چکیده

الگوریتم PBIL یکی از انواع الگوریتم‌های تخمین توزیع است. این الگوریتم در مقایسه با الگوریتم‌هایی مانند الگوریتم ژنتیک از نظر سرعت، حافظه مصرفی و دقت جواب‌های بدست آمده، کارایی بهتری دارد؛ و به خوبی می‌تواند نقاط ناشناخته فضای جستجو را با به کار بردن روش‌های آماری جستجو نماید. این الگوریتم از یک بردار احتمال استفاده کرده و اعضای جمعیت را از طریق نمونه برداری از آن ایجاد می‌کند. از طرفی الگوریتم EO برای جستجوهای محلی و نزدیک بهینه مناسب می‌باشد و می‌تواند بهترین جواب را از میان جواب‌های محلی بدست آورده و در بهینه محلی گیر نکند. لذا با ترکیب این دو الگوریتم می‌توان تعاملی بین دو مفهوم اساسی مطرح در الگوریتم‌های تکاملی، اکتشاف و بهره برداری، ایجاد نمود و نتایج بهتری بدست آورد. در این مقاله یک نسخه تطبیق پذیر از الگوریتم  $\tau$ -EO به نام EO-LA استفاده شده که در آن وظیفه انتخاب جزء تعویضی به یک اتوماتای یادگیر محول شده است. در طول اجرای این الگوریتم، با توجه به شایستگی جواب‌های تولید شده، سیگنال‌های بازخوردی به اتوماتای یادگیر ارسال می‌شود تا به این وسیله نحوه انتخاب جزء تعویضی هرچه بیشتر با مساله تطبیق پیدا کند. نتایج موجود در این مقاله نشان دهنده کارایی الگوریتم پیشنهادی (ترکیب سه روش) بر روی مسئله بگرنج، دو تکه کردن گراف می‌باشد.

## کلمات کلیدی

الگوریتم یادگیری افزایشی مبتنی بر جمعیت، بهینه سازی حدی، اتوماتای یادگیر، الگوریتم‌های تخمین توزیع، مسائل بهینه سازی.

بهینه‌سازی را کد می‌کند. این جمعیت‌ها در نسل‌های بعدی با پیشرفت جستجو تکامل می‌یابند.

## 1- مقدمه

در این الگوریتم‌ها یک جمعیت ممکن است با یک توزیع احتمال تخمین زده شود و راه‌حل‌های کاندید جدید توسط نمونه‌برداری از این توزیع بدست آیند. این روش مزایای زیادی دارد مانند، اجتناب از همگرایی زودرس و استفاده از یک نمایش فشرده و کوتاه.

الگوریتم یادگیری افزایشی مبتنی بر جمعیت<sup>[1]</sup> یک الگوریتم بهینه‌سازی و تخمین توزیع است. الگوریتم توسط بالوجا در سال 1994 بدست آمد. الگوریتم ساده‌تر از یک الگوریتم ژنتیک استاندارد است و در اکثر مواقع به سمت نتایج بهتر از آن هدایت می‌کند.

در الگوریتم یادگیری افزایشی، ژن‌ها به صورت مقادیر حقیقی در رنج [0,1] نمایش داده می‌شود که مشخص کننده احتمال هر ال

در طول دهه 1990 اکثر مسائل بهینه‌سازی ترکیبی با موفقیت توسط الگوریتم‌های ژنتیک حل می‌شدند. اما وجود مسائل فریبنده و سخت و بدلیل عملکرد ضعیف الگوریتم ژنتیک باعث جست‌وجوی الگوریتم‌های بهینه‌سازی جدید شد. به خاطر وجود این مشکلات یک تعداد از محققان گروهی از الگوریتم‌های جدید که الگوریتم‌های تخمین توزیع<sup>[1,2,3]</sup> نامیده می‌شود را پیشنهاد دادند.

این الگوریتم‌ها توسط ماهرلن بین و پاب مطرح شدند، الگوریتم‌های تخمین توزیع یک نمونه الهام گرفته از مکاشفه تصادفی مبتنی بر جمعیت‌هایی از افراد هستند که هر کدام یک راه‌حل ممکن از مساله

## 2- الگوریتم یادگیری افزایشی مبتنی بر جمعیت (PBIL)

الگوریتم یادگیری افزایشی [1,2] یک ترکیبی از بهینه‌سازی تکاملی و یادگیری رقابتی است. این الگوریتم کمک می‌کند تا یک بردار احتمال با مقادیر حقیقی  $\bar{P} = \{P_1, \dots, P_L\}$  (طول کدگذاری باینری) ایجاد شود، بطوریکه وقتی نمونه برداری می‌شود، راه‌حل‌های با کیفیت بالا و با احتمال بالا را تولید کند.

از یک بردار احتمال که مقدار 0.5 برای هر مکان بیت در نظر می‌گیرد، شروع می‌شود. این بردار احتمال، بردار احتمال مرکزی به خاطر اینکه در نقطه مرکز فضای جستجو می‌افتد، نامیده می‌شود. نمونه‌برداری از این بردار احتمال اولیه راه‌حل‌های تصادفی تولید می‌کند. یک راه‌حل از یک بردار احتمال  $\bar{P}$  به این صورت نمونه‌برداری می‌شود، برای هر محل ژن  $i$ ، اگر یک عدد تصادفی ایجاد شده  $r = \text{rand}(0,1) < p_i$  (1) می‌شود، در غیر این صورت آن محل ژن (0) قرار داده می‌شود.

در تکرار  $t$ ، یک مجموعه  $S(t)$  متشکل از  $n$  راه‌حل از بردار احتمال  $\bar{p}(t)$  نمونه‌برداری می‌شود. نمونه‌ها با استفاده از تابع شایستگی مبتنی بر مسأله ارزیابی می‌شوند. سپس بردار احتمال برای یافتن بهترین راه‌حل  $\bar{B}(t)$  از مجموعه  $S(t)$  یادگیری انجام می‌دهد، به صورت زیر:

(1)

$$P_i(t+1) = (1 - \alpha) * P_i(t) + \alpha * B_i(t),$$

بطوریکه  $\alpha$  نرخ یادگیری است که فاصله بردار احتمال برای هر تکرار را معین می‌کند. بعد از اینکه بردار احتمال برای رسیدن به بهترین نمونه بروز رسانی می‌شود، به منظور حفظ تنوع نمونه‌ها ممکن است نیاز به انجام عمل جهش بی‌بی باشد.

عمل جهش به این صورت انجام می‌شود. برای محل ژن  $i = \{1, \dots, L\}$ ، اگر یک عدد تصادفی  $r = \text{rand}(0,1) < P_m$  (احتمال جهش) تولید شود، پس  $P_i$  با رابطه زیر تغییر داده می‌شود:

(2)

$$P'_i = \begin{cases} P_i * (1.0 - \delta_m) & P_i > 0.5 \\ P_i & P_i = 0.5 \\ P_i * (1.0 - \delta_m) + \delta_m & P_i < 0.5 \end{cases}$$

بطوریکه  $\delta_m$ ، شیفت جهش است که تعداد عملیات جهش را کنترل می‌کند. بعد از عمل جهش، یک مجموعه جدید از نمونه‌ها توسط بردار احتمال جدید تولید می‌شوند و این حلقه تکرار می‌شود. با پیشروی فرآیند جستجو، عناصر در بردار احتمال از مقدار اولیه 0.5 به سمت 0 یا 1 می‌روند و راه‌حل‌های با کیفیت بالا را نشان

خاص در آن ژن است. در این الگوریتم به سوی بهترین فرد عمل یادگیری با تغییر بردار احتمال انجام می‌گیرد و نهایتاً جوابهای مناسبی ارائه می‌دهد.

تفاوت اساسی آن با الگوریتم ژنتیک حفظ آمارهای جمعیت به جای حفظ یک تعداد زیاد از نمونه‌ها می‌باشد. چون الگوریتم یادگیری افزایشی مجبور نیست یک جمعیت بزرگ را نگه‌داری کند، بنابراین حافظه کمتری را نیاز دارد و اجراهایش سریعتر و نتایج در اکثر موارد بهتر از یک الگوریتم ژنتیک معادل است.

ایرادی که در مورد این الگوریتم وجود دارد، آمارهای مرتبه دوم (وابستگی متقاطع) جمعیت را حفظ نمی‌کند. بنابراین این الگوریتم بطور موثر در فضا‌هایی که آمارهای مرتبه دوم وجود دارد و یک امر مهم در بهینه‌سازی رشته بی‌بی است دچار مشکل می‌شود، بنابراین با ترکیب این روش با روشی مانند بهینه‌سازی حدی و اتوماتای یادگیر می‌توان تا حدی این مشکل را حل کرده و به جوابهای بهینه و با دقت بیشتر دست یافت.

الگوریتم  $\tau$ -EO [8,9] یک الگوریتم تکاملی جدید است که بر اساس ویژگی بحران‌های خود سازمانده [5,6] عمل می‌کند. این الگوریتم که برای حل مسائل بهینه‌سازی استفاده شده و جواب‌های مطلوبی ارائه داده است، بر خلاف بسیاری از الگوریتم‌های تکاملی که در هر لحظه از چندین جواب تشکیل شده‌اند تنها از یک جواب تشکیل شده و در هر مرحله آن را بهبود می‌دهد. بهبود جواب در این الگوریتم با انتخاب یکی از اجزای جواب (جزء تعویضی) و جایگزین کردن {مقدار آن} به‌عبارت دیگر انجام می‌شود.

در این مقاله با ترکیب این دو روش به گونه‌ای که بهترین فرد بدست آمده با الگوریتم یادگیری افزایشی مبتنی بر جمعیت، توسط الگوریتم بهینه‌سازی حدی دوباره بهبود می‌یابد و مانع از گیر افتادن الگوریتم در بهینه‌های محلی و دست‌یابی به بهینه سراسری می‌شود. و نیز با ترکیب سه روش بهبود بیشتری در نتایج حاصل می‌شود. این الگوریتم بر روی مسئله دو تکه کردن گراف انجام گرفته است. نتایج نشان دهنده عملکرد مناسب الگوریتم پیشنهادی در مقایسه با الگوریتم‌های مطرح همانند: الگوریتم ژنتیک، بهینه‌سازی حدی، بهینه‌سازی حدی و اتوماتای یادگیر و الگوریتم یادگیر افزایشی مبتنی بر جمعیت استاندارد و نیز الگوریتم ترکیبی یادگیری افزایشی و بهینه‌سازی حدی است.

ادامه مقاله بدین صورت سازماندهی شده است: در بخش 2 و 3 و 4 الگوریتم PBIL و  $\tau$ -EO و LA به اختصار معرفی می‌شوند. بخش 5 به معرفی الگوریتم جدید PBIL-EO-LA اختصاص دارد. در بخش 6 مسائل آزمایشی و روش حل آنها توسط الگوریتم‌های مقایسه شده شرح داده می‌شوند. در بخش 7 نتایج حاصل از اجرای الگوریتم‌ها بر روی مسائل انتخاب شده ارائه شده و با هم مقایسه می‌شوند. بخش 8 جمع‌بندی می‌باشد.

می‌دهند. روند جستجو وقتی که تعدادی از شرایط پایانی ارضاء می‌شود متوقف می‌گردد. برای مثال، رسیدن به حداکثر تعداد دفعات تکرار  $t_{max}$  یا وقتی که بردار احتمال به مقادیر 0 یا 1 در هر محل از بیهتا همگرا می‌شود.

### 3- الگوریتم بهینه سازی حدی (τ-EO)

الگوریتم τ-EO [9و8] یک الگوریتم اکتشافی مبتنی بر پدیده بحران‌های خود سازمانده [5,6] است. در این الگوریتم، جواب مساله به آرایه‌ای تبدیل می‌شود که هر یک از عناصر آن معادل یکی از اجزای جواب است.

این الگوریتم در هر مرحله (هر تکرار حلقه اصلی) با انتخاب یکی از اجزای جواب (جزء تعویضی) و تغییر مقدار آن جواب را بهبود می‌دهد. در این الگوریتم، برای پیدا کردن جزء تعویضی، ابتدا اجزای جواب به صورت صعودی بر اساس شایستگی محلی آنها مرتب می‌شوند. سپس به هر جزء با توجه به مکان آن در فهرست مرتب شده ( $n$ ) احتمالی متناسب با  $P(n)$  (رابطه 1) نسبت داده شده و در انتها یکی از اجزا با توجه به این احتمال‌ها انتخاب می‌شود؛ انتخاب جزء تعویضی در این روش به نحوی است که اجزا با شایستگی کمتر با احتمال بالاتری انتخاب می‌شوند. این روش موجب می‌شود تا الگوریتم τ-EO بتواند از بسیاری از نقاط بهینه محلی فرار کند.

(3)

$$P(n) \sim n^{-\tau}$$

### 4- اتوماتای یادگیر

اتوماتای یادگیر [10] ماشینی است که می‌تواند تعداد محدودی عمل را انجام دهد؛ این ماشین عملی را انتخاب می‌کند، عمل انتخاب شده توسط محیط ارزیابی شده و نتیجه آن به صورت یک سیگنال بازخوردی مثبت (در صورت مناسب بودن عمل) یا منفی (در صورت نامناسب بودن عمل) به اتوماتا بازگردانده می‌شود. مقدار این سیگنال در انتخاب اعمال بعدی تاثیر می‌گذارد. هدف این فرایند این است که اتوماتا بعد از گذشت مدتی به سمت مناسب‌ترین عمل خود در محیط میل کرده و یا به عبارت دیگر یاد می‌گیرد که کدام عمل بهترین عمل است. برای اطلاعات بیشتر درباره اتوماتاهای یادگیر می‌توانید به [11] مراجعه نمایید.

### 5- الگوریتم پیشنهادی (PBIL-EO-LA)

با توجه به مزایا و معایب ذکر شده از الگوریتم یادگیری افزایشی مبتنی بر جمعیت و برای رفع معایب آن الگوریتم مفروض را با الگوریتم بهینه سازی حدی ترکیب می‌کنیم. در الگوریتم بهینه

سازی حدی نیز قدرت جستجوی کل فضای جستجو در مقایسه با الگوریتم‌های دیگر کمتر می‌باشد لذا با ترکیب این دو الگوریتم از قدرت مکاشفه کل فضای جستجوی الگوریتم یادگیری افزایشی مبتنی بر جمعیت و نیز قدرت الگوریتم بهینه سازی حدی در یافتن جوابهای دقیق در قسمتی از فضای جستجو عمل می‌کنیم. با ترکیب این دو الگوریتم با اتوماتای یادگیر نتایج حاصل بهتر می‌شود. این الگوریتم که از یک اتوماتای یادگیر برای انتخاب جزء تعویضی استفاده می‌کند در طول اجرا با بررسی جواب‌های تولید شده و تولید سیگنال‌های بازخوردی مناسب برای این اتوماتا، با مساله مورد حل تطبیق پیدا می‌کند.

در ابتدا بردار احتمال را با مقادیر اولیه 0.5 مقدار دهی می‌نماییم و سپس یک تعداد افراد بر اساس اینکه سائز جمعیت چقدر باشد با استفاده از بردار احتمال نمونه برداری می‌شوند.

در الگوریتم پیشنهادی از نخبه‌گزینی نیز استفاده شده و بهترین فرد به عنوان نخبه انتخاب شده و بعد از تغییر در یادگیری بردار احتمال، مطابق رابطه 1 شرکت می‌نماید. با انتخاب بهترین فرد از قسمتی از فضای جستجو سعی داریم با اعمال ترکیب بهینه سازی حدی و اتوماتای یادگیر بر روی بهترین فرد و اعمال یک جستجوی محلی در فضای بهترینها، شایسته‌ترین فرد را بدست آورده و از آن در عمل یادگیری استفاده نماییم. می‌توان مراحل الگوریتم پیشنهادی را به صورت زیر نمایش داد:

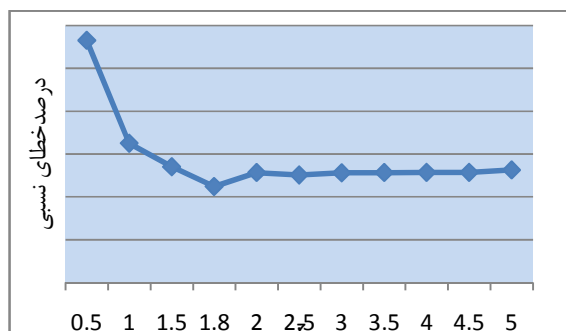
- 1- مقداردهی اولیه
  - 2- مقداردهی بردار احتمال با مقدار 0.5 (بردار احتمال مرکزی)
  - 3- نمونه برداری N فرد از بردار احتمال
  - 4- پیدا کردن و حفظ (نخبه‌گزینی) بهترین فرد در جمعیت
  - 5- اعمال الگوریتم EO-LA داخلی (با توجه به مسئله) بر روی بهترین فرد
  - 6- عمل یادگیری بردار احتمال (رابطه 1) بر اساس بهترین فرد حاصل از مرحله قبل
  - 7- اعمال جهش بر روی بردار احتمال (رابطه 2)
  - 8- تکرار از مرحله 3 تا توقف الگوریتم
  - 9- برگرداندن بهترین جواب
- الگوریتم EO-LA داخلی:

1. مقداردهی بردار احتمال اتوماتای یادگیر
2. محاسبه شایستگی اجزای جواب با توجه به مساله
3. مرتب کردن اجزای جواب بر اساس شایستگی آنها به صورت صعودی
4. انتخاب یک عمل توسط اتوماتای یادگیر و انتخاب جزء متناظر با آن به عنوان جزء تعویضی
5. انتخاب مقدار جدید برای جزء تعویضی با توجه به مساله

6. جایگزین کردن مقدار جدید جزء تعویضی و تولید جواب جدید
7. پاداش دادن و یا جریمه کردن اتوماتای یادگیر با توجه به جواب جدید و جواب‌های قبلی
8. اجرای مرحله 2 تا وقتی که بهبود حاصل می شود.

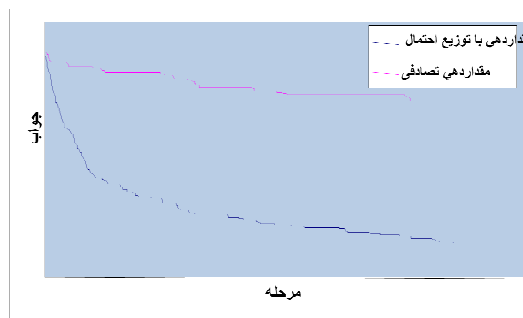
الگوریتم  $\tau$ -EO در [8] و [9] و استفاده شده است در این مراجع از یک خوشه بندی اولیه نیز استفاده شده است؛ در این روش برای محاسبه شایستگی هر جزء جواب، از نسبت تعداد گره‌های همسایه گره در دو تکه گراف استفاده می‌شود. برای اینکه جوابها با مفروضات مسئله مطابقت داشته باشند (تعداد گره های مساوی در هر بخش) از الگوریتم KL [12] نیز استفاده شده است. در پیاده‌سازی انجام شده برای حل مساله دو تکه کردن گراف، با توجه به اهمیت نرخ یادگیری موجود در رابطه 1 الگوریتم مورد نظر با مقادیر مختلف نرخ یادگیری بر روی نمونه مسائل انجام شده و مقداری که خطای نسبی کمتری در مقایسه با مقادیر دیگر داشت (0.25) انتخاب گردید. در الگوریتم مفروض احتمال جهش (0.02) و شیفت جهش (0.2) نیز به روش گفته شده و محاسبه خطای نسبی بدست آمده است. در ترکیب اتوماتای یادگیر و بهینه سازی حدی، اتوماتا هیچگاه جریمه نشده (پارامتر جریمه همواره برابر صفر است) و در صورتی که جواب تولید شده در هر مرحله بهترین جواب تا آن مرحله باشد به عمل انتخاب شده توسط اتوماتای یادگیر پاداش داده می‌شود.

همچنین برای مقایسه الگوریتم‌های PBIL-EO و PBIL-EO-LA،  $\tau$ -EO و مشاهده تاثیر تغییرات اعمال شده، مقدار پارامتر  $\tau$  در تمامی اجراها برای 1.8 که بهترین مقدار این پارامتر برای الگوریتم  $\tau$ -EO می‌باشد قرار داده شده است. شکل 2 نشان‌دهنده مجموع خطای نسبی به ازای مقادیر مختلف  $\tau$  می‌باشد. همانطور که مشاهده می‌شود الگوریتم به ازای مقدار 1.8 برای پارامتر  $\tau$  کمترین خطا را تولید کرده است.



شکل 2: تعیین مقدار مناسب پارامتر  $\tau$

مزیت الگوریتم EO-LA بر الگوریتم  $\tau$ -EO در تطبیق‌پذیری این الگوریتم با نمونه مساله مورد حل است. این الگوریتم ابتدا با بردار احتمالی همانند توزیع احتمال مورد استفاده در الگوریتم  $\tau$ -EO شروع به کار کرده و در طول اجرا با تغییر مقادیر عناصر این بردار، آن را هرچه بیشتر برای حل نمونه مساله مورد نظر تنظیم می‌کند. نتایج بر روی مسئله آزمایش نشان دهنده کارایی روش مفروض در مقایسه با روش‌های دیگر است. شکل 1 چگونگی حرکت الگوریتم EO-LA به سمت جواب در حل نمونه‌ای از مساله دو تکه کردن گراف را برای حالتی که بردار احتمال اتوماتای یادگیر با توزیع احتمال رابطه 3 و برای حالتی که این بردار به صورت تصادفی مقداردهی اولیه شده باشد نمایش می‌دهد.



شکل 1: نحوه عملکرد EO-LA بر اساس نحوه مقدار دهی اولیه بردار احتمال اتوماتای یادگیر

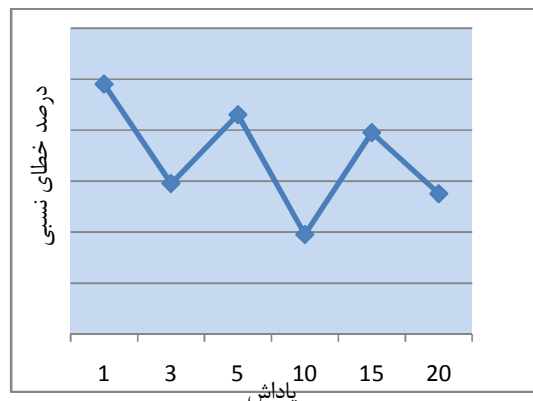
## 6- مسائل آزمایشی و روش حل آنها

برای ارزیابی کارایی الگوریتم پیشنهادی و مقایسه آن با روش‌های دیگر از مساله دو تکه کردن گراف<sup>۲</sup> استفاده شده است. در مساله دو تکه کردن گراف هدف، تقسیم کردن گراف به دو قسمت است به شکلی که تعداد گره‌های دو قسمت برابر باشد (در صورت فرد بودن تعداد گره‌های گراف، اختلاف تعداد گره‌های دو قسمت برابر 1 باشد) و ثانیاً تعداد یال‌های بین دو قسمت کمترین مقدار ممکن باشد. نمونه‌های استفاده شده برای مقایسه عملکرد الگوریتم‌ها در حل این مساله از مرجع [7] آورده شده‌اند.

### 6-1- حل مسئله دو تکه کردن گراف

برای حل مساله دو تکه کردن گراف توسط الگوریتم PBIL-EO از رشته‌ای از بیت‌ها با مقادیر 0 و 1 به منظور تعیین دو بخش متفاوت در گراف استفاده می‌شود و نیز از روش حل این مساله توسط

به ازای هر مقدار از پارامتر پاداش، خطاهای نسبی تولید شده توسط الگوریتم محاسبه شده و مقداری از پارامتر (10) که کمترین خطا را تولید کرده باشد به عنوان مقدار بهینه این پارامتر در نظر گرفته می‌شود.



شکل 3: بدست آوردن بهترین پارامتر پاداش در اتوماتای یادگیر

در الگوریتم پیشنهادی هم چنین سائز جمعیت 60 و تعداد دفعات تکرار برابر 100 در نظر گرفته شده است.

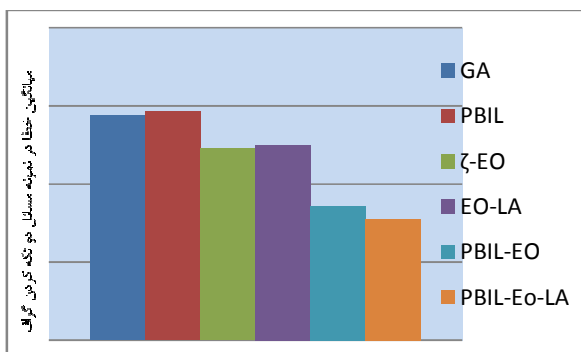
## 7- نتایج

برای بررسی و مقایسه الگوریتم‌ها در حل مساله دو تکه کردن گراف از نمونه مسائل انتخاب شده در بخش قبل استفاده شده است. برای مشاهده تغییر عملکرد الگوریتم‌های PBIL-EO-LA و  $\tau$ -EO و PBIL-EO مقدار پارامتر  $\tau$  برای آنها برابر 1.8 در نظر گرفته شده است.

مقایسه الگوریتم‌ها بر اساس مجموعه جواب‌های تولید شده توسط آنها برای نمونه مسائل انتخاب شده در 10 اجرا انجام شده است. نتایج آماری جواب‌های تولید شده توسط این الگوریتم‌ها در جدول 1 نمایش داده شده است. همانطور که در این جدول مشاهده می‌شود الگوریتم PBIL-EO-LA تقریباً در تمامی موارد از بقیه الگوریتم‌ها بهتر می‌باشد.

با توجه به جدول نتایج همانطور که ذکر شد ملاحظه می‌شود، الگوریتم PBIL نسبت به GA کارایی بهتری دارد. در مقایسه PBIL و EO تقریباً در نصف موارد PBIL و نصف دیگر EO عملکرد بهتری داشته است. در مقایسه با EO-LA (ترکیب بهینه سازی حدی با اتوماتای یادگیر) نیز عملکرد بهتری داشته و این بدلیل تاثیر اتوماتا در بهبود عملکرد جستجوی محلی EO بوده و هنوز امکان اینکه نقاط مکاشفه نشده وجود داشته باشد است. اما تقریباً در تمام موارد الگوریتم پیشنهادی (PBIL-EO-LA) عملکرد مناسبی داشته است. شکل 4 نیز نشان می‌دهد، میانگین خطا در نمونه مسائل دو تکه کردن گراف برای الگوریتم پیشنهادی کمتر از بقیه الگوریتم‌ها است.

نتایج مناسب این الگوریتم بدلیل استفاده از مزایای هر دو الگوریتم و رفع عیوب آنها از طریق ترکیب با یکدیگر است. الگوریتم PBIL عمل جستجوی نقاط ناشناخته در فضا را انجام می‌دهد و به نوعی می‌توان گفت بر روی عمل اکتشاف<sup>4</sup> تاکید دارد، الگوریتم EO روی استفاده از تجربیات قبلی<sup>5</sup> و جستجو در نزدیکی نقاط بهینه و یافتن نقطه بهینه سراسری تاکید دارد و با ترکیب اتوماتای یادگیر در الگوریتم عمل جستجوی جواب بهینه با دقت بیشتر انجام گرفته و نتایج بهتری حاصل شده است.



شکل 4: نمودار میله ای مقایسه روش‌ها با در نظر داشتن میانگین خطا نمونه مسائل دو تکه کردن گراف

## 8- جمع‌بندی

با توجه به نتایج حاصل ملاحظه می‌شود الگوریتم پیشنهادی (PBIL-EO-LA) بر روی مسئله جهان واقعی، دو تکه کردن گراف عملکرد خوبی داشته است و این بدلیل ترکیب سه روش و استفاده از مزایای هر سه روش و ایجاد تعادل بین دو مفهوم مکاشفه و بهره برداری از اطلاعات قبلی است. الگوریتم PBIL عمل کشف نقاط ناشناخته در فضای جستجو و EO جستجو در نزدیکی نواحی بهینه را برای یافتن بهینه سراسری انجام می‌دهند که از طریق ترکیب با اتوماتای یادگیر با دقت بیشتری سعی در پیدا کردن بهینه سراسری دارد.

## مراجع

- [1] Yang S. and Yao X., "Population-Based Incremental Learning with Associative Memory for Dynamic Environments", IEEE Trans. Evolutionary Computation, vol. 12, No. 5, pp. 542-561, 2008.
- [2] Tianshi Chen, Tang Ke, Guoliang Chen, Xin Yao, "Analysis of Computational Time of Simple Estimation of Distribution Algorithms", IEEE Trans., Evolutionary computation, vol. 14, No. 1, 2010
- [3] Hons Robin, Estimation of Distribution Algorithms and Minimum Relative Entropy, phd. Thesis, university of Bonn, 2005.

Estimation of Distribution Algorithm<sup>1</sup>  
 Population Based Incremental Learning<sup>2</sup>  
 Graph Bipartitioning Problem<sup>3</sup>  
 exploration<sup>4</sup>  
 exploitation<sup>5</sup>

- [4] Boettcher, S. and Percus, A. G., "Extremal Optimization: An Evolutionary Local-Search Algorithm", <http://arxiv.org/abs/cs.NE/0209030>
- [5] [http://en.wikipedia.org/wiki/Self-organized\\_criticality](http://en.wikipedia.org/wiki/Self-organized_criticality)
- [6] Per Bak, Chao Tang and Wiesenfeld, K. "Self-organized Criticality", Physical Review A, Vol. 38, No. 1, 1988.
- [7] <http://staffweb.cms.gre.ac.uk/~c.walshaw/partition>
- [8] Boettcher, S., "Extremal Optimization of Graph Partitioning at the Percolation Threshold", Physics A, vol 32, No.28, pp.5201-5211, 1999.
- [9] Boettcher, S. and Percus, A. G., "Extremal Optimization for Graph Partitioning", Physical Review E, vol. 64, pp. 021114, 2001.
- [10] Narendra K. S. and Thathachar M.A.L., "Learning Automata: An Introduction", Prentice Hall, 1989.
- [11] Thathachar, M. A. L. and Sastry, P. S., "Varieties of Learning Automata: An Overview", IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 32, No. 6, PP. 711-722, 2002.
- [12] Mühlenbein H. and Mahnig Th., "Evolutionary Optimization and the Estimation of Search Distributions with Applications to Graph Bipartitioning", Journal of Approximate Reasoning, Vol. 31, 2002.

جدول 1: نتایج آزمایشات مسئله دو تکه کردن گراف

نام گراف	GA	PBIL	ζ-EO	EO-LA	PBIL-EO	PBIL-EO-LA
3elt	335.5	332	330.5	327	267	209.5
4elt	772	734.5	728	722.5	666.5	615
Add20	4287	4774	4942	4683	3168	3331.5
Add32	2179.5	2425	254.5	216	583.5	488
Crack	445	442	380	378.5	382	367.5
CS4	1016	1018	769	790	643.5	579
CTI	2433.5	2294	1747	1794	1442	1372.5
data	510	504	672	674	483	482
FE-4elt2	196	200	494	502	196	199
FE-sphere	665	664	677	678	654	643
UK	67	66	35	31	37	34
Whitaker3	224.5	217.5	442	453	205	201.5
Wing-nodal	9863	9636	9287	9263.5	6710	5842.5