

الگوریتم‌های ترکیبی (آtomاتاهای یادگیر + الگوریتم‌های ژنتیکی) برای حل مسئله مینیمم کردن پهنهای باندگراف

محمد رضا میبدی

دانشکده مهندسی کامپیوتر و فناوری اطلاعات
دانشگاه صنعتی امیرکبیر
تهران ایران
mmeybodi@aut.ac.ir

علی صفری ممقانی

دانشکده مهندسی برق، رایانه و فناوری اطلاعات
دانشگاه آزاد اسلامی
قزوین ایران
safari_m_61@yahoo.com

چکیده- در این مقاله، سه الگوریتم تقریبی برای حل مسئله مینیمم کردن پهنهای باند در گراف‌ها بکار گرفته شده است که بواسیله تعییر دادن ترتیب سطراها و ستون‌های ماتریس مجاورت، باعث کاهش پهنهای باند می‌گردد. الگوریتم اول مبتنی بر آtomاتاهای یادگیر مهاجرت اشیا می‌باشد. دومین الگوریتم ترکیبی می‌باشد که از ترکیب آtomاتای یادگیر مهاجرت اشیا و ژنتیک حاصل شده است. الگوریتم سوم نیز از ترکیب آtomاتای یادگیر ساختار متغیر و ژنتیک حاصل شده است. الگوریتم‌ها بر روی ۱۱۳ نمونه از مسئله‌های واقعی ارزیابی شده‌اند و نتایج آن با تعدادی از الگوریتم‌های مشهور مقایسه شده است که نتایج بهبود یافته‌ای نسبت به چندین مورد از بهترین الگوریتم‌ها دیده می‌شود. نشان داده شده است که با استفاده همزمان از الگوریتم‌های ژنتیکی و آtomاتای یادگیر در فرایند جستجو، سرعت رسیدن به جواب افزایش می‌یابد و همچنین از بدام افتادن الگوریتم در بهینه‌های محلی جلوگیری می‌شود. یکی دیگر از نکات مثبت الگوریتم‌های جدید ارایه شده این است که بحث سرعت اجرایی و کیفیت نتایج را در حالت متعادل نگه می‌دارند یعنی این الگوریتم‌ها قادر هستند که در مدت زمان کم، جواب‌های معقول بدست آورند.

کلید واژه- آtomاتای یادگیر مهاجرت اشیا، الگوریتم ژنتیکی، پهنهای باند گراف، الگوریتم ترکیبی.

راس v که بصورت $B_L(v)$ نشان داده می‌شود برابر با بیشترین اختلاف بین $L(v)$ و برچسب‌های اختصاص یافته به راس‌های همسایه v می‌باشد. بنابراین داریم:

$$B_L(v) = \max\{|L(v) - L(u)| : (v, u) \in E\} \quad (1)$$

پهنهای باند گراف G با در نظر گرفتن برچسب‌گذاری L برابر با بیشترین مقدار پهنهای باند راس‌های گراف خواهد بود. پس خواهیم داشت:

$$B_L(G) = \max\{B_L(v) : v \in V\} \quad (2)$$

با در نظر گرفتن تعاریف (1) و (2)، پهنهای باند مطلوب برای گراف G برابر با کمترین مقدار $B_L(G)$ از بین برچسب‌گذاری‌های ممکن L برای گراف G خواهد بود. بنابراین، مسئله مینیمم کردن پهنهای باند گراف بصورت مسئله‌ای شامل یافتن یک برچسب‌گذاری L که مقدار $B_L(G)$ را مینیمم می‌سازد تعریف می‌گردد. توسط پاپا دیمیتریو نشان داده شده است که مسئله مینیمم کردن

۱- مقدمه

مسئله مینیمم کردن پهنهای باند در گراف دارای کاربردهای فراوانی می‌باشد که می‌توان به حل سیستم‌های خطی بزرگ، طراحی VLSI، قابلیت ماندگاری شبکه، الکترومغناطیس‌های صنعتی، روش‌های المان محدود برای جواب‌های تخمینی معادلات دیفرانسیل جزئی، سیستم‌های تبدیل قدرت با اندازه بزرگ، طراحی مدار، جنبش‌های شیمیایی، ژئوفیزیک عددی و بازیابی اطلاعات از ابر متن‌ها اشاره کرد [۱۴] [۱۵] [۱۶] [۱۷] [۱۸].

فرض کنید که $G = \{V, E\}$ ، گرافی با مجموعه راس‌های V و مجموعه یال‌های E باشد که در آن $|V| = n$ می‌باشد. یک برچسب‌گذاری L از گراف G ، اعداد صحیح $\{1, 2, 3, \dots, n\}$ را به راس‌های گراف G اختصاص می‌دهد، به عبارت دیگر برچسب‌گذاری L نگاشتی یک به یک بصورت $L : V \rightarrow \{1, 2, 3, \dots, n\}$ می‌باشد. اگر $(v, u) \in E$ ، برچسب تخصیص یافته به راس v باشد، در این صورت پهنهای باند

۲- الگوریتم LA مبتنی بر آtomاتای مهاجرت اشیا برای حل مسئله

همانطور که می‌دانیم برای برچسب‌گذاری گراف $G = (V, E)$ که $|V| = n$ می‌باشد، $n!$ جایگشت مختلف وجود دارد. بنابراین، اگر از آtomاتاهای یادگیر برای حل مساله استفاده شود، آtomاتای یادگیر باید $n!$ اقدام داشته باشد که تعداد زیاد اقدام‌ها باعث کاهش سرعت همگرایی می‌شود. به همین منظور برای حل مسئله می‌توانیم از آtomاتای مهاجرت اشیا دریافت پاسخ از محیط باعث تغییر وضعیت آtomاتا نمی‌گردد بلکه باعث گردش اشیا بین وضعیت‌های مختلف آtomاتا می‌شود. برای حل این مسئله، آtomاتا را می‌توان به صورت شش‌تایی $\langle V, \underline{a}, \underline{\phi}, \underline{\beta}, \underline{E}, \underline{G} \rangle$ نشان داد که در آن: $\{V_1, V_2, \dots, V_n\}$ مجموعه اشیای بکار رفته می‌باشند که این اشیا، همان برچسب‌های منتبه به راس‌ها می‌باشد که دارای مقادیری بین ۱ و n می‌باشند. این اشیا بین وضعیت‌های مختلف آtomاتا حرکت می‌کنند و برچسب‌گذاری‌های مختلفی از گراف را ایجاد می‌کنند. $\{\alpha_1, \alpha_2, \dots, \alpha_k\}$ مجموعه اقدام‌های مجاز برای آtomاتای یادگیر است. هر یک از این اقدام‌ها نمایانگر راسی از گراف می‌باشد، مثلاً a_k نشان‌دهنده راس k ام گراف است. این آtomاتا K اقدام دارد (تعداد اقدام‌های این آtomاتا با تعداد راس‌های گراف می‌باشد). $\{\phi_1, \phi_2, \dots, \phi_{KN}\}$ مجموعه وضعیت‌ها و N عمق حافظه برای آtomاتا می‌باشد. مجموعه وضعیت‌های این آtomاتا به K زیر مجموعه $\{\phi_1, \phi_2, \dots, \phi_N\}$ و $\{\phi_{N+1}, \phi_{N+2}, \dots, \phi_{2N}\}$ و ... و $\{\phi_{(K-1)N+1}, \phi_{(K-1)N+2}, \dots, \phi_{jN}\}$ تقسیم می‌شود و اشیا بر اساس این که در کدام وضعیت قرار داشته باشند دسته‌بندی می‌گردد. اگر شی u در مجموعه وضعیت‌های $\{\phi_{(j-1)N+1}, \phi_{(j-1)N+2}, \dots, \phi_{jN}\}$ قرار داشته باشد در اینصورت برچسب u منتبه به راس z خواهد بود. در مجموعه وضعیت‌های اقدام z ، به وضعیت $\phi_{(j-1)N+1}$ و وضعیت داخلی ϕ_j و به وضعیت ϕ_{jN} ، وضعیت مرزی گفته می‌شود. شئی که در وضعیت $\phi_{(j-1)N+1}$ قرار دارد شی با اهمیت بیشتر و شئی که در وضعیت ϕ_{jN} قرار دارد، شی با اهمیت کمتر نامیده می‌شود. $\beta = 0,1$ مجموعه ورودی‌های آtomاتا می‌باشد. در این مجموعه ۱ شکست و مقدار صفر موفقیت را نشان می‌دهد. $\phi \rightarrow \beta$: F تابع نگاشت وضعیت‌ها می‌باشد.

پهنهای باند گراف یک مسئله NP-Complete است [۸] و گاری و همکارانش نشان داده‌اند که حتی اگر گراف ورودی درختی با حداقل درجه ۳ باشد، مسئله مینیمم کردن پهنهای خواهد بود [۹]. با خاطر اهمیت مسئله مینیمم کردن پهنهای باند گراف، الگوریتم‌هایی توسط افراد مختلف برای حل آن پیشنهاد داده شده است. به نظر می‌رسد که هیوریستیک‌ها روش مناسبی برای حل این مسئله باشند که سعی در یافتن جواب‌های مناسب در مدت زمان معقول دارند. تعدادی الگوریتم تقریبی از دهه ۱۹۶۰ برای حل مسئله ارایه شده است. نمونه این الگوریتم‌ها، الگوریتم کاتھیل- مک‌کی [۱۱]، الگوریتم کاتھیل- مک‌کی معکوس [۱۲] و الگوریتم GPS [۱۰] می‌باشند. اخیراً، الگوریتم‌های متاهیوریستیک برای حل این مسئله بکار رفته‌اند که الگوریتم‌های مبتنی بر شبیه‌سازی سرد کردن فلزات [۱۳]، جستجوی تابو [۱۵]، الگوریتم ژنتیکی با تپه‌نوردی [۱۷]، الگوریتم GRASP-PR [۱۶] نمونه‌هایی از این دسته می‌باشند. در این مقاله، سه الگوریتم تقریبی جدید برای حل این مسئله ارایه شده است. اولین الگوریتم مبتنی بر آtomاتای یادگیر مهاجرت اشیا می‌باشد و الگوریتم دوم و سوم به ترتیب الگوریتم ترکیبی مبتنی بر ترکیب ژنتیک با آtomاتای مهاجرت اشیا و الگوریتم ترکیبی مبتنی بر ترکیب ژنتیک با آtomاتای ساختار متغیر می‌باشد. ما این الگوریتم‌ها را با تعدادی از بهترین الگوریتم‌های موجود مانند: الگوریتم GPS، الگوریتم شبیه‌سازی سرد کردن فلزات، جستجوی تابو، الگوریتم ژنتیکی با تپه‌نوردی و الگوریتم GRASP-PR کرده ایم که نتایج بهبود یافته‌ای نسبت به چندین مورد از این الگوریتم‌ها دیده می‌شود. نشان داده شده است که با استفاده همزمان از الگوریتم‌های ژنتیکی و آtomاتای یادگیر در فرایند جستجو، سرعت رسیدن به جواب افزایش می‌یابد و همچنین از بدام افتادن الگوریتم در بهینه‌های محلی جلوگیری می‌شود. بقیه مقاله به این صورت سازماندهی شده است که در بخش دوم به توصیف الگوریتم پیشنهادی مبتنی بر آtomاتای مهاجرت اشیا برای حل مسئله می‌پردازیم و در بخش سوم، الگوریتم پیشنهادی ترکیبی مبتنی بر ژنتیک و آtomاتای مهاجرت اشیا را توصیف می‌کنیم. در بخش چهارم، الگوریتم پیشنهادی بعدی که مبتنی بر ترکیب ژنتیک با آtomاتای ساختار متغیر می‌باشد نشان داده شده است. بخش پنجم و ششم نیز بترتیب اختصاص به ارزیابی نتایج آزمایش‌ها و نتیجه‌گیری مقاله دارد.

به نوع آتماتای یادگیر متفاوت خواهد بود. اکنون، با توجه به توصیفات قبلی می‌توانیم الگوریتم بکار رفته برای حل مسئله را نشان دهیم. الگوریتم با تولید تصادفی و یا با استفاده از پیمایش عمقی BFS گراف و ایجاد جایگشتی از راس‌های گراف شروع به کار می‌کند. در ابتدا، همانطور که در شکل ۱ دیده می‌شود، برچسب‌ها را بعنوان اشیا هر یک از اقدام‌های آتماتا (به ترتیب با شروع از اقدام اول) در حالات مرزی آنها قرار می‌دهیم. سپس، به طور تصادفی یکی از اشیا آتماتا انتخاب می‌شود و میزان برازنده‌گی آن محاسبه می‌شود، در صورتی که شی در جای مناسب قرار گرفته باشد، پاداش می‌گیرد و در غیر اینصورت جریمه می‌شود. سپس این کار را برای یک شی تصادفی دیگر ادامه می‌دهیم. الگوریتم تا زمانی ادامه پیدا می‌کند که شرط توقف الگوریتم برآورده شود (شرط توقف الگوریتم، تکرار الگوریتم تا حداقل تکرارها یا رسیدن به جواب بهینه و یا قرار گرفتن تمامی اشیا در حالت‌های داخلی می‌باشد). شبه کد این الگوریتم در شکل ۲ آمده است.

```

Procedure Bandwidth-Minimization-OMA(G);
Begin
  n = |V|;
  randomly or with BFS create a permutation of Graph
  nodes.
  Assign labels as objects to the boundary states of
  corresponding actions;
  EvalFitness(f);
  Iteration=1;
  Repeat
    For a random node u do //  $1 \leq u \leq n$ 
      If  $B_f(u) \geq \beta \times B_f(G)$  then reward(u); //  $0 < \beta \leq 1$ 
      Else penalize(u); //Critical Node
      Inc(iteration);
      EvalFitness(f);
    Until(iteration=Max_iterations or all of objects appear in
    states of maximumcertainty);
  End Procedure

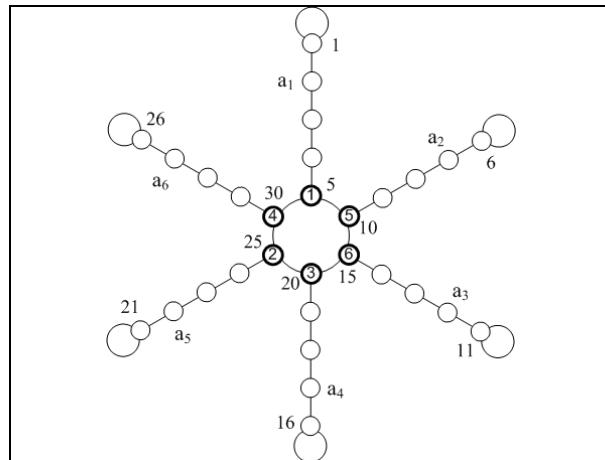
```

شکل ۲: الگوریتم مبتنی بر آتماتای مهاجرت اشیا برای حل مسئله مینیمم کردن پهنهای باند گراف

۳- الگوریتم ترکیبی GALA برای حل مسئله

با ترکیب الگوریتم ژنتیکی و آتماتای یادگیر [۴ و ۵] و تلفیق مفاهیم ژن، کروموزوم، اقدام و عمق، می‌توان به یک روش جستجوی کارا برای حل مساله مینیمم کردن پهنهای باند گراف دست یافت. خودترمیمی، تولید مثل، جریمه و پاداش (هدایت) از ویژگی‌های مهم الگوریتم ترکیبی است. جهت اطلاعات بیشتر می‌توانید به مراجع [۴ و ۵] مراجعه کنید.

باشد. اینتابع از روی وضعیت فعلی و ورودی آتماتا وضعیت بعدی آن را تولید می‌نماید. اینتابع چگونگی گردش اشیا را در وضعیت‌های آتماتا مشخص می‌نماید. $\underline{\phi} \rightarrow G$: تابع نگاشت خروجی می‌باشد. اینتابع تصمیم می‌گیرد که به ازای هر وضعیت آتماتا چه اقدامی را انجام دهد. پس اگر شی u در مجموعه وضعیت‌های $\{\phi_{(j-1)N+1}, \dots, \phi_{jN}\}$ قرار داشته باشد، اقدام j انتخاب می‌گردد. بعنوان مثال شکل ۱، یک آتماتای مهاجرت اشیا با اتصالات ستینین برای برچسب گذاری گرافی با ۶ راس را نشان می‌دهد.



شکل ۱: آتماتای یادگیر مهاجرت اشیا مبتنی بر اتصالات ستینین

برچسب‌های تخصیص یافته به راس‌ها به صورت زیر است:

$$L(1)=1, \quad L(2)=5 \quad L(3)=6 \quad L(4)=3 \quad L(5)=2 \quad L(6)=4$$

عملگر جریمه و پاداش : در جواب ایجاد شده اولیه، یک شی (برچسب راس) به صورت تصادفی انتخاب می‌شود. سپس آن شی پاداش یا جریمه می‌شود. اگر پهنهای باند راسی که شی (برچسب) مورد نظر به آن تخصیص یافته است از مقدار آستانه (مقدار آستانه بصورت تطبیقی محاسبه می‌شود و در هر لحظه مقدار آن برابر با β برابر پهنهای باند گراف است که $0 < \beta \leq 1$. خود یک پارامتر ورودی به الگوریتم می‌باشد) کمتر باشد به این راس پاداش داده می‌شود و به سمت وضعیت‌های داخلی‌تر این اقدام حرکت می‌کند، در غیر اینصورت راس جریمه می‌شود (راس بحرانی). در اثر پاداش دادن یا جریمه کردن یک راس، وضعیت راس در مجموعه وضعیت‌های اقدام مربوطه، تغییر می‌کند. اگر راسی در وضعیت مرزی یک اقدام قرار داشته باشد، جریمه شدن آن باعث تغییر اقدام آن و در نتیجه باعث ایجاد جایگشت جدیدی می‌شود. عملگر جریمه و پاداش با توجه

آتماتای والد خوانده می شوند. در شکل ۳ شبه کد این عملگر نشان داده شده است. از آنجا که در این الگوریتم از آتماتا (کروموزوم) استفاده می شود و هر آتماتا دارای مشخصه های اختصاصی مربوط به خود (وضعیت، اقدام و شئ متناظر هر اقدام) می باشد، جهت خوانایی بیشتر شبه کد، این مشخصه ها را با پیشوند نام آتماتا و جداساز نقطه نشان می دهیم. مثلا برای نشان دادن وضعیت شی u از آتماتا i از نمایش $LA_i.State(u)$ استفاده شده است.

```

Procedure SX1 (LA1, LA2)
Begin
Generate two random numbers r1 and r2 between 1 to n
r1 = Random *n; r2 = Random *n;
r1 = Min(r1, r2); r2 = Max(r1, r2);
for i = r1 to r2 do
    j = Action of LA1 where
        LA1.Object(LA1.Action(j))= LA2.Object(LA2.Action(i));
        Swap(LA1.State(LA1.Action(i)),LA1.State(LA1.Action(j)));
        Swap(LA1.Object(LA1.Action(i)),LA1.Object(LA2.Action(i)));
    j = Action of LA2 where
        LA2.Object(LA2.Action(j))= LA1.Object(LA1.Action(i));
        Swap(LA2.State(LA2.Action(i)), LA2.State(LA2.Action(j)));
        Swap(LA2.Object(LA1.Action(i)),LA2.Object(LA2.Action(i)));
    end for
End Procedure

```

شکل ۳: شبه کد عملگر ترکیب SX1

در شکل ۴، شبه کد عملگر ترکیب دوم نشان داده شده است.

```

Procedure SX2 (LA1, LA2)
Begin
Generate two random numbers r1 and r2 between 1 to n
r1 = Random *n; r2 = Random *n;
r1 = Min(r1, r2); r2 = Max(r1, r2);
for i = r1 to r2 do
    if ((BLA1(u) > BLA2(u))) ) then
        j = Action of LA2 where
            LA2.Object (LA2.Action(j)) = LA1.Object(LA1.Action(i));
            Swap(LA2.Object(LA2.Action(i)), LA2.Object (LA2.Action(j)));
    end if
    else
        j = Action of LA1 where
            LA1.Object(LA1.Action(j)) = LA2.Object(LA2.Action(i));
            Swap(LA1.Object(LA1.Action(i)), A1.Object(LA1.Action(j)));
    end else
    end for
End Procedure

```

شکل ۴: شبه کد عملگر ترکیب SX2

ج) عملگر جهش: در عملگر جهش Swap Mutation ، دو اقدام(ژن) از یک آتماتا (کروموزوم) به صورت تصادفی انتخاب شده و جابجا می شوند. شکل ۵ این عملگر را نشان می دهد.

کدگذاری کروموزومها: در الگوریتم پیشنهادی برخلاف الگوریتم های ژنتیک کلاسیک، از کدگذاری دودویی برای کروموزومها استفاده نمی شود. هر کروموزوم، توسط یک آتماتای یادگیر از نوع مهاجرت اشیاء نشان داده می شود به طوریکه هر کدام از ژن های کروموزوم به یکی از اقدام های آتماتا نسبت داده می شوند و در یک عمق مشخصی از آن اقدام قرار می گیرند.

جمعیت اولیه: با فرض اینکه تعداد اعضای جمعیت برابر با pop_size باشد، می توانیم جمعیت اولیه را بطور تصادفی ایجاد کنیم یا بر اساس پیمایش BFS تولید کنیم. روش دیگر برای تولید جمعیت اولیه، ادغامی از دو روش تصادفی و پیمایش BFS است، یعنی pop_size عضو جمعیت با ایجاد جایگشت تصادفی تولید می شوند. برای تولید آخرین عضو جمعیت، می توانیم از جایگشت ایجاد شده توسط پیمایش BFS استفاده کنیم. آخرین عضو اضافه شده به جمعیت بیشترین تشابه را با جواب نهایی دارد.

عملگرهای از آنجا که در الگوریتم ترکیبی، هر کروموزوم به صورت یک آتماتای یادگیر نمایش داده می شود، عملگرهای جابجایی و جهش مشابه عملگرهای سنتی ژنتیک نیستند.

الف) عملگر انتخاب: پیاده سازی ما برای انتخاب آتماتای یادگیر (کروموزوم) برای عملگرهای جهش و یا ترکیب مبتنی بر چرخ رولت می باشد. در انتهای هر تکرار از الگوریتم از مفهوم نخبه سالاری نیز استفاده شده است.

ب) عملگر ترکیب یا جابجایی: برای انجام دادن این عملگر می توان از هر یک از عملگرهای ترکیب که برای کار با جایگشت ها مناسب هستند استفاده کرد. در اینجا، دو عملگر ترکیب جدید توصیف می گردد. در عملگر ترکیب اول، دو کروموزوم والد انتخاب شده و به صورت تصادفی دو ژن r_1 و r_2 در یکی از دو کروموزوم والد انتخاب می شوند. سپس همین دو ژن در کروموزوم دیگر انتخاب می شوند. مجموعه ژنهای با شماره های بین r_1 و r_2 را مجموعه جابجایی می نامیم. سپس ژن های هم شماره در دو مجموعه جابجایی با یکدیگر جابجا می شوند (مثلا ژن شماره r_1 از مجموعه جابجایی اول با ژن شماره r_1 از مجموعه جابجایی دوم، ژن شماره r_{1+1} از مجموعه جابجایی اول با ژن شماره r_{1+1} از مجموعه جابجایی دوم و ...). با این عمل دو کروموزوم جدید حاصل می شوند که اصطلاحا فرزندان دو

که احتمال حالتها برای تمامی اقدام‌های موجود به روز شود. برای حل این مشکل یک الگوریتم یادگیری ترکیبی به نام STGA را بکار می‌گیریم [۱۸] و که در آن عملگرهای الگوریتم ژنتیکی را در داخل فرایند یادگیری آtomاتای یادگیر بکار می‌روند. در این الگوریتم، تعدادی از اقدام‌های موجود را نمونه‌برداری می‌کنیم و آtomاتای یادگیر را بصورت محلی بر روی آنها اعمال می‌کنیم. با اعمال آtomاتای یادگیر بر روی مجموعه کوچک، جمعیتی از اقدام‌های ایجاد می‌شود و سپس جهت تولید اقدام‌های جدید عملگرهای ژنتیکی بر روی جمعیت موجود آمده بکار می‌بریم. هر اقدام بصورت یک رشته جایگشتی کدگذاری شده است. هر رشته دارای یک مقدار برازنده‌گی می‌باشد که نشان دهنده احتمال انتخاب اقدام می‌باشد. پس از اینکه یک اقدام انتخاب گردید و بر روی محیط اعمال شد، بسته به اینکه از محیط پاداش دریافت کند و یا جریمه شود، مقدار برازنده‌گی بروز می‌گردد. در ضمن با بروز کردن احتمال اقدام انتخابی، بقیه اقدامات موجود نیز بروز می‌شود جهت تولید جواب‌های جدید پس از یک تعداد مشخصی از انتخاب‌ها و بروز رسانی احتمالات اقدام‌ها، الگوریتم ژنتیکی را بر روی جمعیت حاصل اعمال می‌کنیم. در پایان الگوریتم ژنتیکی چک می‌کنیم که عناصر تکراری در جمعیت نباشد. در صورت وجود عنصر تکراری با اعمال جهش بر روی یکی از این عناصر، عنصر جدیدی ایجاد می‌کنیم. شکل ۷، الگوریتم ترکیبی STGA را نشان می‌دهد. الگوریتم دارای پارامترهای ورودی G و α و β می‌باشد که بترتیب نشان دهنده گراف ورودی و پارامترهای پاداش و جریمه آtomاتای یادگیر با ساختار متغیر می‌باشند.

Function STGA(G, α, β):Minimum Bandwidth of G ;
Begin
Initialization;
Iteration=1;
Repeat
For $i=1$ **to** LA-max-iter **do**
Action;
Evaluation(α, β);
Endfor ;
Select(parent1,parent2);
Crossover;
Mutation;
Replace;
Inc(iteration);
Until(iteration=Max_iterations);
End Function

شکل ۷: توصیف الگوریتم STGA

```

Procedure Mutation (LA)
Begin
    i = Random *n;
    j = Random *n;
    Swap( LA.Object(LA.Action(i)),LA.Object(LA.Action(j)));
End Procedure

```

شکل ۵: شبیه کد عملگر جهش

در ادامه در شکل ۶، الگوریتم ترکیبی نشان داده شده است.

```

Function Bandwidth Minimization_Hybrid1(G):Minimum
Bandwidth of G;
Begin
    Pop_size = Size of Population;
    Create the initial population LA1 ... LApop_size;
    For i=1 to pop_size do
        EvalFitness(LAi);
    Endfor
    Iteration=1;
    repeat
        NewLA1 = NewLA2 = LA with minimum Value of
        Bandwidth;
        for i = 2 to n do
            Select LAi; Select LA2 ;
            if (Random ≤ CrossoverRate) then
                Crossover ( LA1, LA2 );
            if (Random ≤ MutationRate) then
                Mutation ( LA1 ); Mutation ( LA2 );
            NewLAi+1 = LA1;
            NewLAi+2 = LA2 ;
            i=i+2;
        end for
        for i = 1 to pop_size do
            LAi = NewLAi;
            u = Random *n; //n=|V|
            If (  $B_{LA_i}(u) < \beta \times B_f(G)$  ) then
                Reward(LAi , u );
            else
                Penalize(LAi , u );
        End for
        For i=1 to pop_size do
            EvalFitness(NewLAi);
        Endfor
        Merge the new chromosomes with old chromosomes;
        Select Fittest chromosomes from chromosomes as the
        next generation;
        Inc(iteration);
        Until(iteration=Max_iterations);
    End Function

```

شکل ۶: الگوریتم ترکیبی GALA برای حل مسئله مینیمم کردن پهنهای باند گراف

۴- الگوریتم ترکیبی STGA برای حل مسئله

با در نظر گرفتن عدد اقدام‌های موجود در مورد مسئله مینیمم کردن پهنهای باند در گراف، در صورتی که بخواهیم از آtomاتای یادگیر با ساختار متغیر برای حل این مسئله استفاده کنیم، مشکل کند بودن همگرایی یادگیری وجود خواهد داشت. زیرا در هر مرحله از یادگیری لازم خواهد بود

شامل ۸۰ نمونه با تعداد راس‌های بین ۲۰۰ تا ۱۰۰۰ راس می‌باشد. تمامی این نمونه ماتریس‌ها از سیستم‌های خطی و سایر مسائل مهندسی و علمی بدست آمده‌اند. این مجموعه از سایت <http://math.nist.gov/MatrixMarket/data/Harwell-Boeing> قابل دریافت می‌باشند. ما نتایج بدست آمده از الگوریتم‌های ارایه شده جدید را با الگوریتم کلاسیک GPS و الگوریتم شبیه‌سازی سرد کردن فلزات (SA-DJ) و جستجوی تابو (TS) و روش جستجوی حریصانه تصادفی تطبیقی با پیوند دوباره مسیر (GRASP-PR) و الگوریتم ژنتیکی به همراه تپه‌نوردی (GA-HC) مقایسه کردیم. جدول ۱، نتایج کلی ۵ روش قبلی را به همراه سه روش جدید را از دیدگاه مینیمم پهنه‌ای باند بدست آمده و مدت زمان اجرای الگوریتم مورد بررسی قرار می‌دهد. همانطور که در جدول ۱ مشاهده می‌کنید، کیفیت نتایج الگوریتم‌های GA-HC و GRASP-PR از بقیه الگوریتم‌ها بهتر است، بخارطه اینکه میانگین مقادیر پهنه‌ای باند و انحراف این مقادیر از بهترین جواب‌های بدست آمده بسیار کم می‌باشد و تعداد بهترین جواب‌های یافته شده توسط این الگوریتم‌ها نسبت به بقیه الگوریتم‌ها بیشتر می‌باشد. با دقت در مورد الگوریتم‌های جدید ارایه شده یعنی LA و دو الگوریتم ترکیبی GALA و STGA به نکات امیدوار کننده‌ای دست می‌یابیم.

الگوریتم ترکیبی جدید GALA، یک الگوریتم بسیار مناسب می‌باشد که کیفیت نتایج آن تقریباً نزدیک به الگوریتم TS و GRASP می‌باشد ولی این الگوریتم از الگوریتم‌های GRASP و TS در حدود سه برابر سریع‌تر است (مخصوصاً در مورد نمونه‌های مجموعه B). در ضمن با دقت در میانگین مقادیر پهنه‌ای باند و انحراف از بهترین جواب و تعداد بهترین جواب‌های یافته شده توسط الگوریتم GALA مشاهده خواهیم کرد که کیفیت نتایج این الگوریتم از الگوریتم GPS و SA-DJ و LA و STGA بهتر و مناسب‌تر می‌باشد. بنابراین با توجه به برتری لگوریتم STGA نسبت به این الگوریتم‌ها و از طرف دیگر با توجه به اینکه نتایج الگوریتم STGA نزدیک به الگوریتم‌های TS و GRASP می‌باشد ولی بسیار سریع‌تر از آنها می‌باشد، می‌توان الگوریتم GALA را عنوان یک الگوریتم کارا که جواب‌های مناسبی را در مدت زمان معقول ارایه می‌دهد، در نظر گرفت، بعبارت دیگر این الگوریتم بحث کیفیت و مدت زمان را در حالت موازن نگه می‌دارد.

هر یک از بخش‌های الگوریتم به صورت زیر خواهد بود:

رویه Initialization: در ابتدا جمعیت اولیه از جواب‌ها را بصورت تصادفی یا با استفاده از پیمایش BFS ایجاد می‌کنیم و مقدار احتمال آنها را برابر $f_i(0) = \frac{1}{r}$ $i=1,2,\dots,r$ قرار می‌دهیم که r برابر با تعداد جواب‌های موجود در جمعیت می‌باشد.

رویه Action: بر اساس مقادیر احتمال جواب‌ها، جوابی از جمعیت موجود بعنوان اقدامی برای آtomاتیک یادگیر انتخاب می‌شود. سپس، اقدام انتخاب شده توسط آtomاتا بر روی محیط اعمال می‌شود و بسته به مقدار اقدام انتخابی جوابی بعنوان پاداش یا جریمه از سمت محیط بر روی آن اعمال می‌گردد.

رویه Evaluation : بسته به نتیجه اعمال اقدام بر روی محیط، احتمالات تمامی اقدام‌های موجود در جمعیت مطابق با طرح یادگیری خطی آtomاتیک یادگیر بروز می‌شود. در صورتی که اقدام i در تکرار tام، موفق به گرفتن پاداش گردد، مقادیر احتمال اقدام‌ها در تکرار t+1 برابر با معادله (۳) خواهد بود.

$$\begin{aligned} f_i(t+1) &= f_i(t) + \sum_{j \neq i}^r af_j(t) \\ f_j(t+1) &= f_j(t) - af_j(t) \quad (\forall j \neq i) \end{aligned} \quad (3)$$

در صورتی که اقدام i در تکرار t+1 برابر با معادله (4) خواهد بود:

$$\begin{aligned} f_i(t+1) &= f_i(t) - \sum_{j \neq i}^r \left(\frac{\beta}{r-1} - \beta f_j(t) \right) \\ f_j(t+1) &= f_j(t) + \frac{\beta}{r-1} - \beta f_j(t) \quad (\forall j \neq i) \end{aligned} \quad (4)$$

۵- نتایج آزمایش‌ها

این بخش شامل تحلیل و مقایسه کارایی الگوریتم‌های جدید ارایه شده در مقابل الگوریتم‌های قبلی موجود می‌باشد. آزمایش‌ها بر روی کامپیوترا با مشخصات Intel Core Due 2GHz و 2GB Ram و بر روی ۱۱۳ نمونه از مجموعه ماتریس‌های اسپارس هارول-بوئینگ انجام شده است. این مجموعه شامل دو دسته می‌باشد. دسته اول شامل ۳۳ نمونه با تعداد راس‌های بین ۳۰ تا ۱۹۹ راس می‌باشد. دسته دوم

الگوریتم، کیفیت نتایج بدست آمده از الگوریتم LA نسبت به الگوریتم های GPS و SA-DJ نیز بهتر می باشد. بنابراین با توجه به این توصیفات، می توان گفت الگوریتم مبتنی بر LA نیز الگوریتمی خواهد بود که بحث کیفیت و مدت زمان اجرایی الگوریتم را در حالت موازن نگه می دارد.

نکته بعدی در مورد الگوریتم جدید STGA می باشد. کیفیت نتایج الگوریتم ترکیبی STGA نسبت به الگوریتم های GPS و SA-DJ و الگوریتم جدید LA مناسب تر است، زیرا میانگین پنهانی باند و انحراف از بهترین جواب در این الگوریتم کمتر می باشد. در ضمن، مدت زمان صرف شده در الگوریتم ترکیبی STGA یک مدت زمان کم نسبت به سایر الگوریتم های موجود یعنی: GA-HC و GRASP-PR و TS و SA-DJ و GALA و SA-DJ می باشد ولی مدت زمان آن از الگوریتم LA بطور نامحسوس بیشتر است ولی همانطور که اشاره شد، میانگین نتایج در آن کمتر از الگوریتم LA می باشد. بنابراین می توان الگوریتم STGA را در کنار الگوریتم های ترکیبی GALA و LA عنوان سه الگوریتم جدید نامید که توازن بین سرعت اجرایی و کیفیت نتایج را در نظر می گیرند.

نکته دیگری که از جدول ۱ نتیجه می شود این است که گرچه الگوریتم کلاسیک GPS سریع ترین الگوریتمی است که تاکنون برای حل مسئله پیشنهاد شده است ولی میانگین پنهانی باند بدست آمده در آن بسیار کمتر از بهترین الگوریتم پیشنهادی می باشد و دارای میزان انحراف زیادی از بهترین جوابها می باشد. بنابراین با توجه اینکه به مدت زمان صرف شده در الگوریتم جدید مبتنی بر آtomاتای یادگیر مهاجرت اشیا(LA) یک مدت زمان بسیار کم نسبت به سایر الگوریتم های موجود یعنی: GRASP- GA-HC و PR و TS و الگوریتم های جدید پیشنهادی یعنی: GALA و STGA می باشد، می توان این الگوریتم را بعنوان یک الگوریتم سریع مبنی جهت مقایسه مدت زمان اجرایی الگوریتم ها بکار برد. به همین دلیل، در جدول ۱، نسبت مدت زمان اجرایی سایر الگوریتم ها به مدت زمان اجرایی الگوریتم LA محاسبه گردیده است. در ضمن با دقت در میانگین مقادیر پنهانی باند و انحراف از بهترین جواب و تعداد بهترین جواب های یافته شده توسط الگوریتم LA مشاهده خواهیم کرد که علاوه بر بحث مدت زمان اجرایی

جدول ۱: نتایج حاصل از مقایسه الگوریتم ها از نظر کیفیت و سرعت اجرایی

مجموعه A : شامل ۳۳ نمونه با اندازه های ۳۰ و ... و ۱۹۹								
الگوریتم	GA-HC	GRASP-PR	TS	GPS	SA-DJ	LA	GALA	STGA
میانگین مقادیر پنهانی باند	22.48	22.67	23.33	28.58	29.36	26.91	24.88	28.85
انحراف از بهترین جواب	1.22	2.07	5.04	28.68	32.19	21.16	12.02	20.89
تعداد بهترین جواب ها	24	24	14	4	—	6	10	6
مدت زمان اجرایی (ثانیه)	2.54	2.70	4.99	0.002	2063.85	0.61	4.8	1.02
نسبت مدت زمان اجرایی به مدت زمان LA	4.16	4.43	8.18	0	3383.36	1	7.87	1.67

مجموعه B : شامل ۸۰ نمونه با اندازه های ۲۰۰ و ... و ۱۰۰۰

مجموعه B : شامل ۸۰ نمونه با اندازه های ۲۰۰ و ... و ۱۰۰۰								
الگوریتم	GA-HC	GRASP-PR	TS	GPS	SA-DJ	LA	GALA	STGA
میانگین مقادیر پنهانی باند	97.01	99.91	100.78	152.36	164.59	142.74	123.9	140.06
انحراف از بهترین جواب	1.43	4.46	5.37	59.31	72.09	49.25	29.55	46.45
تعداد بهترین جواب ها	46	23	21	9	—	4	7	2
مدت زمان اجرایی (ثانیه)	85.33	235.29	263.97	0.08	26448.83	6.17	94.25	9.84
نسبت مدت زمان اجرایی به مدت زمان LA	13.83	38.13	42.78	0.01	4286.68	1	15.28	1.59

۶- نتیجه‌گیری

در این مقاله، یک الگوریتم مبتنی بر آtomاتای یادگیر مهاجرت اشیا و دو الگوریتم دیگر که مبتنی بر ترکیب آtomاتای یادگیر مهاجرت اشیا و ژنتیک و ترکیب آtomاتای با ساختار متغیر و ژنتیک می‌باشد، برای حل مسئله مینیمم کردن پهنای باند در گراف بکار گرفته شده است. نشان داده شده است که با استفاده همزمان از الگوریتم‌های ژنتیکی و آtomاتاهای یادگیر در فرایند جستجو، سرعت رسیدن به جواب افزایش می‌یابد و همچنین از بدام افتادن الگوریتم در بهینه‌های محلی جلوگیری می‌کند.

مراجع

- [9] M. Garey, R. Graham, D. Johnson and D. E. Knuth, "Complexity results for band-width minimization", SIAM Journal of Applied Mathematics, Vol. 34, pp. 477-495, 1978.
- [10] N. E. Gibbs, W. G. Poole and P. K. Stockmeyer, "An Algorithm for reducing the bandwidth and profile of sparse matrix", SIAM Journal on Numerical Analysis, Vol. 13, No. 2, pp. 236-250, 1976.
- [11] E. Cuthill and J. McKee, "Reducing the bandwidth of sparse symmetric matrices", in Proc. Of the ACM National Conf. Association for Computing Machinery, pp. 157-172, 1969.
- [12] A. George and J. Liu, **Computer Solution of Large Sparse Positive Definite Systems**, Prentice-Hall, 1981.
- [13] G. H. Deuck and J. Jeffs, "A heuristic bandwidth minimization algorithm", Journal of Combinatorial Mathematics and Combinatorial Computing, Vol. 18, pp. 97-108, 1995.
- [14] A. Esposito, M. S. Catalano, F. Malucelli and L. Tarricone, "Sparse matrix band-width reduction: Algorithms, Applications and real industrial cases in electromagnetics, high performance algorithms for structured matrix problems", Advances in the theory of Computation and Computational Mathematics, Vol. 2, pp. 27-45, 1998.
- [15] R. Marti, M. Laguna, F. Glover and V. Campos, "Reducing the bandwidth of a sparse matrix with tabu search", European Journal of Operational Research, Vol. 135, No. 2, pp. 211-220, 2001.
- [16] E. Pinana, I. Plana, V. Campos and R. Marti, "GRASP and path relinking for the matrix bandwidth minimization", European Journal of Operational Research, Vol. 153, pages 200-210, 2004.
- [17] A. Lim, B. Rodrigues and F. Xiao, "Heuristics for matrix bandwidth reduction", European Journal of Operational Research, Vol. 174, pp. 69-91, 2006.
- [18] M. Munetomo, "STGA: An Application of A Genetic Algorithm to Stochastic Learning Automata", Systems and Computers in Japan, Vol. 27, No. 10, 1996.
- [19] B. J. Oommen, and D. C .Y. Ma, "Deterministic Learning Automata Solutions to the equipartitioning problem", IEEE Transactions on Computers, Vol. 37 , pp. 2-13, 1998.
- [20] K. Narendra and M. A. L. Thathachar, **Learning Automata: An Introduction**, Prentice Hall, 1989.
- [۱] ح. بیگی و م. میبدی, "حل مسئله نتاظر گراف توسط آtomاتونهای یادگیر", در مجموعه مقالات پنجمین کنفرانس بین‌المللی انجمن کامپیوتر ایران, ص ۴۰۲-۴۱۵، دانشگاه شهرید بهشتی، ۱۳۷۸.
- [۲] م. رضایپور, حل مسئله نتاظر گراف به کمک آtomاتای یادگیر، پایان نامه کارشناسی ارشد، دانشکده مهندسی کامپیوتر و فن‌آوری اطلاعات، دانشگاه صنعتی امیر کبیر، ۱۳۸۲.
- [۳] ع. صفری ممقانی، طراحی الگوریتم‌های ترکیبی برای حل مسائل مشکل در گراف‌ها، پایان نامه کارشناسی ارشد، دانشکده برق ، مهندسی کامپیوتر و فن‌آوری اطلاعات، دانشگاه آزاد اسلامی قزوین، ۱۳۸۷.
- [۴] ع. صفری ممقانی، ک. اصغری، م. ر. میبدی و ف. محمودی، "رویکرد نوین مبتنی بر الگوریتم ژنتیک در کمینه کردن هزینه اجرای عملگرهای پیوند در پایگاه داده"، در مجموعه مقالات سیزدهمین کنفرانس ملی انجمن کامپیوتر ایران، جزیره کیش، ۸۶.
- [۵] K. Asghari, A. Safari Mamaghani and M. R. Meybodi, "An Evolutionary Approach for Query Optimization Problem in Database", In Proc. of Int. Joint Conf. on Computers, Information and System Sciences, and Engineering (CISSE2007), University of Bridgeport, 2007.
- [۶] M. Berry, B. Hendrickson and P. Raghavan, "Sparse matrix reordering schemes for browsing hypertext", Lectures in Applied Mathematics, Vol. 32, pp. 99-123, 1996.
- [۷] P. Chinn, J. Chavatalova, A.K.Dewdney and N.E.Gibbs, "The bandwidth problem for graphs and matrices-a survey", Journal of Graph Theory, Vol. 6, pp. 223-254, 1982.
- [۸] C. H. Papadimitriou, "The NP-completeness of the bandwidth minimization problem", Computing, Vol. 16, pp. 263-270, 1976.