

## یک روش ترکیبی برای حل مساله مرتب سازی ترتیبی

محمد رضا میبدی

باقر زارعی

دانشکده برق، مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه  
صنعتی امیر کبیر، تهران، ایران

دانشکده مهندسی کامپیوتر، دانشگاه آزاد اسلامی واحد  
شبستر، ایران

MMeybodi@aut.ac.ir

Zarei\_Bager@yahoo.com

چکیده - یکی از مسائل بسیار مهم در تئوری گراف‌ها، مساله مرتب سازی ترتیبی می‌باشد. آtomاتاهای یادگیر و الگوریتم‌های ژنتیکی هر دو از ابزارهای جستجویی باشند که برای حل بسیاری از مسائل NP-Complete بکار برده می‌شوند. در این مقاله یک الگوریتم ترکیبی برای حل مساله مرتب سازی ترتیبی پیشنهاد شده است. این الگوریتم از دو روش الگوریتم‌های ژنتیکی و آtomاتاهای یادگیر بطور همزمان برای جستجو در فضای حالت استفاده می‌نماید. نشان داده شده است که با استفاده از همزمان از آtomاتاتای یادگیر و الگوریتم ژنتیک در فرایند جستجو، سرعت رسیدن به جواب افزایش چشمگیری پیدا می‌کند و همچنین از بدام افتادن الگوریتم در حداقل‌های محلی جلوگیری می‌نماید. نتایج آزمایش‌ها، برتری الگوریتم ترکیبی را نسبت به الگوریتم ژنتیکی و آtomاتاهای یادگیر نشان می‌دهد.

کلمات کلیدی: مساله مرتب سازی ترتیبی، آtomاتاتای یادگیر، الگوریتم ژنتیک

حلشان به طور نمایی افزایش می‌یابد. این مسائل، مسائل بهینه سازی ترکیبی هستند، که زمان حل آنها به صورت تابعی غیر چند جمله‌ای است. مساله مرتب سازی ترتیبی یکی از آنها می‌باشد که حل مساله به معنای پیدا کردن بهترین تور، در مقایسه با تورهای شناخته شده قبلی نمی‌باشد بلکه همچنین باید ثابت کرد که توری با هزینه کمتر از تور پیدا شده نیز وجود ندارد.

آtomاتاهای یادگیر و الگوریتم‌های ژنتیکی، هر دو ابزار جستجوی عمومی می‌باشند که برای حل بسیاری از مسائل NP-Complete بکار برده شده است. در این مقاله یک الگوریتم ترکیبی برای حل مساله مرتب سازی ترتیبی پیشنهاد شده است. این الگوریتم از دو روش الگوریتم‌های ژنتیکی و آtomاتاهای یادگیر بطور همزمان برای جستجو در فضای حالت استفاده می‌نماید. نشان داده شده است که با استفاده از همزمان از آtomاتاتای یادگیر و الگوریتم ژنتیک در فرایند جستجو، سرعت رسیدن به جواب افزایش چشمگیری پیدا می‌کند و همچنین از بدام افتادن الگوریتم در حداقل‌های محلی جلوگیری می‌نماید.

### ۱- مقدمه

گراف‌ها ابزارهای قدرتمندی هستند که به طور گستردگی در کاربردهای متعددی مورد استفاده قرار می‌گیرند. یکی از مسائل بسیار مهم در تئوری گراف‌ها، مساله مرتب سازی ترتیبی می‌باشد. بسیاری از کاربردهای عمومی از جمله طراحی حلقه‌های شبکه‌های Sonet، کابل‌های برق، مسیر هوایی‌ها، مسیریابی وسائط نقلیه و ... را می‌توان با مساله مرتب سازی ترتیبی مدل کرد.

مساله مرتب سازی ترتیبی، نسخه‌ای از مساله فروشنده دوره گرد نامتنازن می‌باشد که یکسری محدودیت‌های اولویتی باید روی رئوس رعایت شود. مساله در Escudero (۱۹۸۸) معرفی شد. فرض کنید که یک گراف کامل داریم که هر یال  $E \in E(u, v)$ ، هزینه نامنفی  $C(u, v)$  را دارد. هدف یافتن سیکل همیلتونی با حداقل هزینه می‌باشد بطوریکه محدودیت‌های اولویتی در بین رئوس ارضاء شود.

بعضی از مسائل وجود دارند که با افزایش بعد آنها، زمان

## ۴- الگوریتم جستجوی ترکیبی برای حل مساله مرتب سازی ترتیبی

با ترکیب الگوریتم ژنتیک و آtomاتای یادگیر و تلفیق مفاهیم ژن، کروموزوم، اقدام و عمق، سابقه تاریخی تکامل راه حل مساله، به شکل کارا استخراج شده و در روند جستجو مورد استفاده قرار می گیرد. خاصیت مهم الگوریتم ترکیبی، مقاومت آن در مقابل تغییرات سطحی جواب هاست، به عبارتی دیگر تعادلی انعطاف پذیر بین کارایی الگوریتم ژنتیک و پایداری آtomاتای یادگیر در الگوریتم ترکیبی وجود دارد. خود ترمیمی، تولید مثل، جریمه و پاداش (هدایت) از ویژگیهای الگوریتم ترکیبی است. در ادامه پارامترهای اصلی این الگوریتم توضیح داده شده است.

### ژن و کروموزوم:

در الگوریتم پیشنهادی برخلاف الگوریتم های ژنتیک کلاسیک، از کدگذاری دودویی برای کروموزوم ها استفاده نمی شود. هر کروموزوم توسط یک آtomاتای یادگیر از نوع مهاجرت اشیاء نشان داده می شود. بطوریکه هر کدام از ژنهای کروموزوم به یکی از اقدامهای آtomاتا نسبت داده می شود و در یک عمق مشخصی از آن اقدام قرار می گیرد.

در این آtomاتا  $\{\alpha_1, \dots, \alpha_k\}$  مجموعه اقدام های مجاز برای آtomاتای یادگیر است. این آtomاتا  $k$  اقدام دارد (تعداد اقدام های این آtomاتا با تعداد راس های گراف برابر است). اگر راس  $v$  از گراف در اقدام  $m$  قرار گرفته باشد، در اینصورت راس  $v$  در ترتیب ملاقات کردن شهرها،  $m$ امین شهر می باشد.

$\{\phi_1, \phi_2, \dots, \phi_{KN}\}$  مجموعه وضعیت ها و  $N$  عمق حافظه برای آtomاتا می باشد. مجموعه وضعیت های 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  $\{\phi_{(K-1)N+1}, \phi_{(K-1)N+2}, \dots, \phi_{KN}\}$  افزایش می شود و راس های گراف بر اساس این که در کدام وضعیت قرار داشته باشند دسته بندی می گردند. اگر گره  $v$  از گراف در مجموعه وضعیت های  $\{\phi_{jN}, \phi_{(j-1)N+1}, \dots, \phi_{(j-1)N+2}\}$  قرار داشته باشد در اینصورت راس  $v$  در ترتیب ملاقات کردن شهرها،  $j$ امین شهر می باشد. در مجموعه وضعیت های اقدام  $v$ ، به وضعیت  $\phi_{(j-1)N+1}$  وضعیت داخلی و به

های محلی جلوگیری می نماید.

### ۲- الگوریتم های ژنتیک

الگوریتم های ژنتیکی که بر مبنای ایده تکامل در طبیعت عمل می نماید، بر روی جمعیتی از راه حل های بالقوه به جستجوی راه حل نهایی می پردازد. در هر نسل، بهترین های آن نسل انتخاب می شوند، و پس از زاد و ولد، مجموعه جدیدی از فرزندان را تولید می کنند. در این فرایند افراد مناسبتر با احتمال بیشتری در نسل های بعدی باقی خواهند ماند.

در آغاز الگوریتم، تعدادی از افراد<sup>1</sup> (جمعیت اولیه) به صورت تصادفی ساخته شده و تابع هدف برای تک آنها ارزیابی می شود. اگر شرط رسیدن به جواب برقرار نباشد (به جواب بهینه نرسیده باشیم)، نسل بعدی با انتخاب والدین براساس میزان برآزندگی شان تولید می شود و فرزندان با احتمال ثابتی دچار جهش می شوند. سپس میزان برآزندگی فرزندان جدید محاسبه شده و جمعیت جدید، از جایگزینی فرزندان با والدین ایجاد می شود و این فرآیند تا برقرار شدن شرط خاتمه تکرار می شود.

### ۳- آtomاتاهای یادگیر

یادگیری در آtomاتاهای یادگیر، انتخاب یک اقدام<sup>2</sup> بهینه از میان یک مجموعه از اقدام های مجاز آtomاتا می باشد. این اقدام روی یک محیط تصادفی اعمال می شود و محیط به این اقدام آtomاتا بوسیله یک پاسخ تصادفی از مجموعه پاسخ های مجاز جواب می دهد. پاسخ محیط بصورت آماری به اقدام آtomاتا واپسیته است. واژه محیط شامل اجتماع تمام شرایط خارجی و تاثیرات آنها روی عملکرد آtomاتا می باشد.

برای یک گراف با اندازه  $n$ ! جایگشت مختلف از رئوس وجود دارد و در صورتیکه از آtomاتاهای یادگیر برای حل کردن مساله مرتب سازی ترتیبی استفاده شود، آtomاتا باید  $n!$  اقدام داشته باشد که تعداد زیاد اقدام ها سرعت همگرایی آtomاتا را کاهش می دهد. به همین جهت آtomاتای مهاجرت اشیاء<sup>3</sup> توسط اولمن<sup>4</sup> و ما<sup>5</sup> پیشنهاد شده است.

<sup>1</sup> Individuals

<sup>2</sup> Action

<sup>3</sup> Object Migrating Automata

<sup>4</sup> Oommen

<sup>5</sup> Ma

شوند، ۶ جایگشت تصادفی  $\langle b, d, e, a, f, c \rangle$ ،  $\langle c, f, b, e, d, a \rangle$ ،  $\langle e, f, b, d, a, c \rangle$ ،  $\langle d, e, f, b, c, a \rangle$  و  $\langle b, e, d, f, c, a \rangle$  که محدودیت فوق را ارضاء می کنند، می باشد. جمعیت اولیه حاصل از گراف شکل ۱ در شکل ۳ نشان داده شده است. در ابتدا هرگره در وضعیت مرزی اقدام خود قرار دارد.

#### تابع برازنده‌ی؟

در الگوریتم های ژنتیک تابع برازنده‌ی، شاخص زنده ماندن کروموزوم ها است. لذا برازنده‌ی یک آtomاتا در مساله مرتب سازی ترتیبی به صورت زیر تعریف می شود.

$$f(LA_i) = 1 / \text{Length of Specified Tour by } LA_i$$

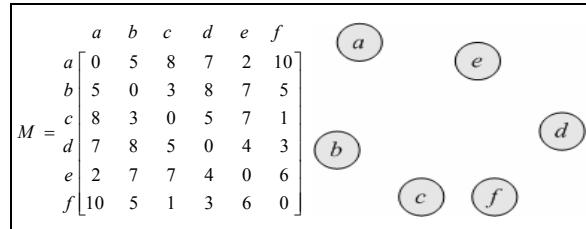
#### عملگرها:

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

(الف) عملگر انتخاب<sup>۷</sup>: برای انتخاب آtomاتاهای یادگیر (کروموزوم ها) برای عملگرهای جهش و ترکیب می توان از یکی از روشهای رتبه بندی، سازوکار چرخ رولت و یا Tournament استفاده کرد.

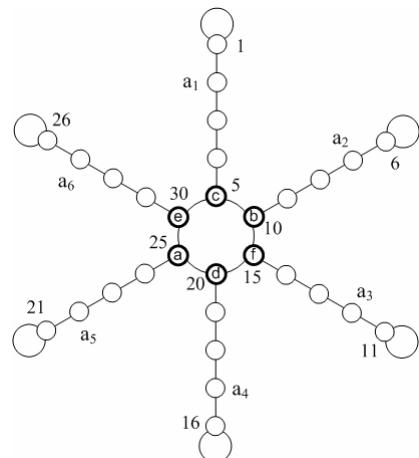
وضعیت  $\phi_{jN}$ ، وضعیت مرزی گفته می شود.

به عنوان مثال گراف کامل شکل ۱ را که شامل ۶ راس می باشد در نظر بگیرید.



شکل ۱- گراف کامل با ۶ راس

جایگشت  $\langle c, b, f, d, a, e \rangle$  از گراف شکل ۱ را در نظر بگیرید. این جایگشت توسط یک آtomاتای یادگیر با اتصالات مشابه آtomاتای ستلین در شکل ۲ نشان داده شده است. این آtomاتا دارای ۶ اقدام  $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6\}$  (به تعداد راسهای گراف) و عمق ۵ می باشد. مجموعه وضعیت های  $\{1, 6, 11, 16, 21, 26\}$  وضعیت های  $\{5, 10, 15, 20, 25, 30\}$  وضعیت های مرزی آtomاتا هستند. در ابتدا هر یک از راسهای گراف در وضعیت مرزی اقدام مربوطه قرار دارد.



شکل ۲- نمایش جایگشت  $\langle c, b, f, d, a, e \rangle$  به وسیله آtomاتای یادگیر با اتصالات مشابه آtomاتای ستلین

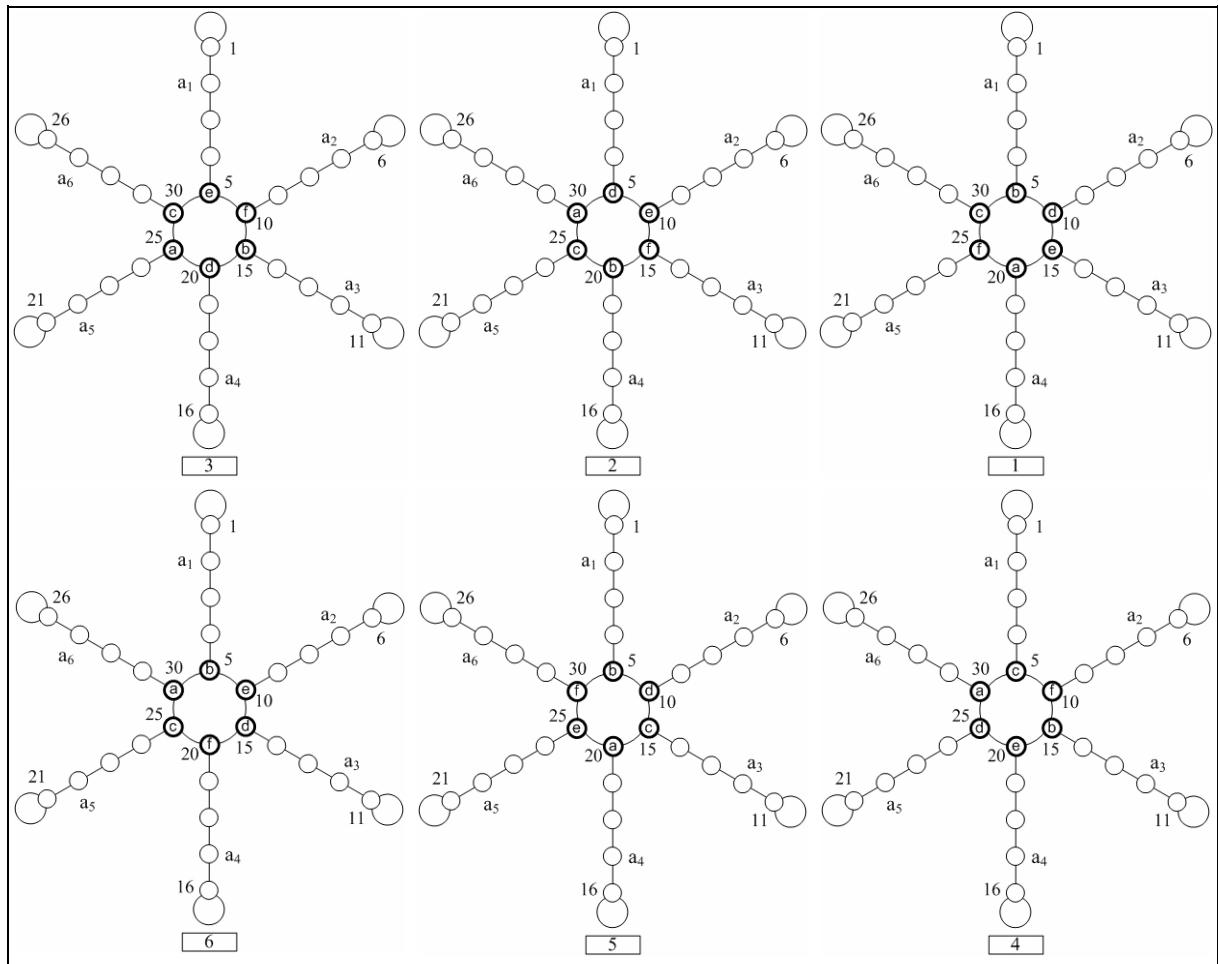
#### جمعیت اولیه:

با فرض اینکه تعداد اعضای جمعیت  $n$  باشد، جمیعت اولیه با ایجاد  $n$  جایگشت تصادفی که محدودیت های اولویتی در بین رئوس را ارضاء می کنند، تولید می شود.

به عنوان مثال جمعیت اولیه برای گراف شکل ۱ با فرض  $n=6$  و این محدودیت که رئوس  $b$  و  $d$  قبل از راس  $a$  ملاقات

<sup>6</sup> Fitness Function

<sup>7</sup> Selection Operator



شکل ۳- جمعیت اولیه برای گراف شکل ۱

```

Procedure Crossover (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 (Ji(LA1) < Ji(LA2)) 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)), LA1.Object(LA1.Action(j)));
        end else
    end for
End Procedure
// Ji(LAk) = (Weight of Edge (i-1,i) in LAk + Weight of Edge
//(i,i+1) in LAk)/2 - (Length of Specified Tour By LAk/n);

```

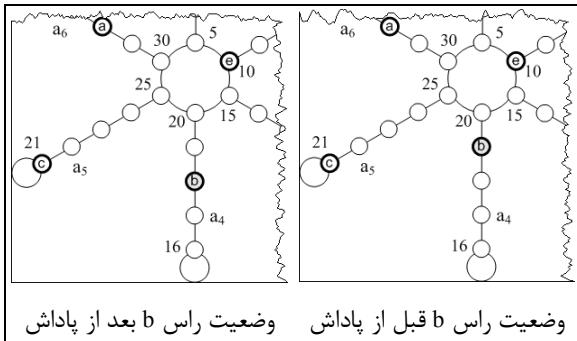
شکل ۴- شبه کد عملگر جابجایی (New Crossover)

به عنوان مثال فرض کنید که آutomataهای LA<sub>2</sub> و LA<sub>5</sub> از جمعیت تشکیل شده قبل به عنوان والد انتخاب شوند. با انتخاب تصادفی دو محل a<sub>2</sub> و a<sub>3</sub>، مجموعه جابجایی {a<sub>2</sub>, a<sub>3</sub>} حاصل می شود و در نهایت مطابق شکل ۵ با

ب) عملگر ترکیب یا جابجایی<sup>۸</sup>: برای انجام دادن این عملگر می توان از یکی از روش‌های .Partially Mapped Crossover و New Cycle Crossover و Ordered Crossover که برای کار با جایگشت ها مناسب هستند استفاده کرد. در اینجا فقط روش پیشنهادی یعنی روش New Crossover توضیح داده می شود. در این روش دو کروموزوم والد انتخاب شده و به صورت تصادفی دو زن ۱ و ۲ در یکی از دو کروموزوم والد انتخاب می شوند. سپس همین دو زن در کروموزوم والد دیگر نیز انتخاب می شوند. مجموعه زنها با شماره های بین i و زرا مجموعه جابجایی می نامیم. سپس زن های هم شماره در دو مجموعه جابجایی با یکدیگر جابجا می شوند. با این عمل دو کروموزوم جدید حاصل می شوند که اصطلاحا فرزندان دو آtomataی والد خوانده می شوند. در شکل ۴ شبه کد این عملگر نشان داده شده است.

جستجوی تصادفی است و اگر با نرخ بالا اعمال شود باعث کاهش در کارایی الگوریتم می‌شود. عملگر جریمه و پاداش ببا توجه به نوع آتماتای یادگیر متفاوت می‌باشد.

به عنوان مثال در آtomاتای با اتصالات مشابه آtomاتای سنتی، اگر راس b در مجموعه وضعیت های {16,17,18,19,20} قرار داشته باشد و میانگین هزینه یالهای ورودی و خروجی به راس b (هزینه یال ورودی به راس b + هزینه یال خروجی از راس b تقسیم بر ۲) از مقدار آستانه (مقدار آستانه بصورت تطبیقی مشخص می گردد و مقدار آن در هر لحظه برابر است با نسبت هزینه کل تور به تعداد راسها) کوچکتر باشد به این راس پاداش داده می شود و به سمت وضعیت های داخلی تر این اقدام حرکت می کند. اگر راس b در داخلی ترین وضعیت (وضعیت شماره 16) قرار داشته باشد و پاداش بگیرد در همان وضعیت باقی می ماند. نحوه حرکت چنین راسی در شکل ۶ نشان داده شده است.



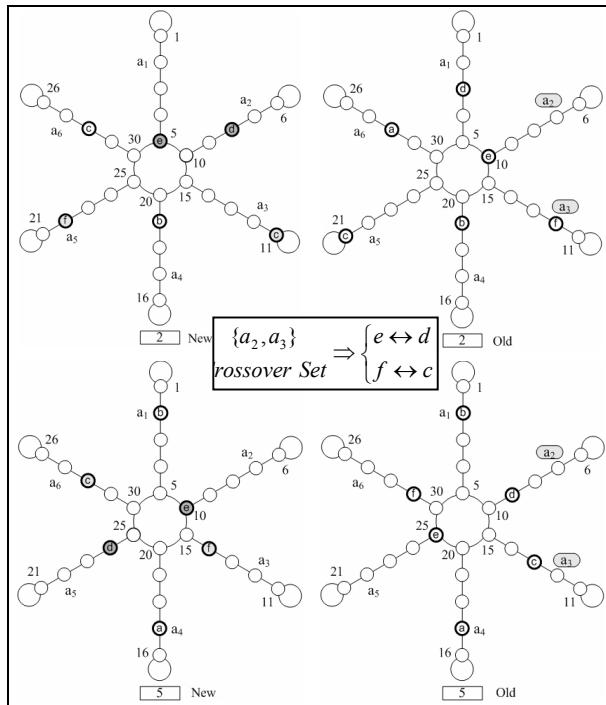
شکل ۶- نحوه پاداش دادن به راس b

اگر میزان برازنده‌گی یک راس از مقدار آستانه بزرگتر باشد در اینصورت تور برقرار شده مناسب نبوده و این راس جریمه می‌شود. در شکل ۷ شبیه کد عملگر جریمه اقدام LA با اتصالات مشابه آتماتای ستلین نشان داده شده است. نحوه حرکت چنین راسی برای دو حالت مختلف در زیر آمده است.

الف) راس در وضعیتی غیر از وضعیت مرزی قرار داشته باشد: جریمه نمودن این راس سبب کم اهمیت شدن این راس نشده و راس به سمت وضعیت های مرزی حرکت می کند.

ب) راس در وضعیت مرزی قرار داشته باشد: در این حالت راسی از گراف را پیدا می کنیم بطوریکه اگر در جایگشت مربوطه جای دو راس عوض شوند بیشترین کاهش در هزینه

جبجایی اقدام های متناظر در فاصله جابجایی، دو کروموزوم جدید حاصل می شود.



شكل ٥- نحوه انجام عملگر جابجاگی (New Crossover)

پ) عملگر جهش<sup>۹</sup>: برای انجام دادن این عملگر می‌توان از یکی از روش‌های Insertion Mutation، Swap Mutation، Scramble Mutation و Inversion Mutation که برای کار با جایگشت‌ها مناسب هستند استفاده کرد. به عنوان مثال در روش Swap Mutation، دو اقدام (زن) از یک آتماتا کروموزم) به صورت تصادفی انتخاب شده و جابجا می‌شوند.

ت) عملگر جرمیه و پاداش<sup>۱۰</sup> : از آنجاییکه هر کروموزوم به صورت یک آتماتای یادگیر نشان داده شده است، در هر یک از آتماتها پس از بررسی میزان برازنده‌گی یک ژن (راس یا اقدام) که به صورت تصادفی انتخاب می‌شود، آن ژن پاداش یا جرمیه می‌شود. در اثر پاداش دادن یا جرمیه کردن یک ژن، وضعیت ژن در مجموعه وضعیت‌های اقدام مربوطه، تغییر می‌کند. اگر ژنی در وضعیت مرزی یک اقدام قرار داشته باشد، جرمیه شدن آن باعث تغییر اقدام آن و در نتیجه باعث ایجاد جایگشت جدیدی می‌شود. نرخ این عملگر باید پایین باشد زیرا این عملگر، یک عملگر

---

9 Mutation Operators

## 10 Penalty and Reward

افرايش نيايد.

در شکل ۹ شبه کد الگوريتم ترکيبي برای حل مساله مرتب سازی ترتيبی آورده شده است.

```

Function SOP_Solver(G) : SOP_Tour
Begin
    n = Size of Population; // n = |VG|
    Create the initial population LA1 ... LAn;
    EvalFitness();
    while (All (Length of Specified Tour By LAi > Constant-Value) ) do
        NewLA1 = NewLA2 = LA with minimum Value of Tour-Lenght;
        for i = 2 to n do
            Select LA1; Select LA2 ;
            if (Random > 1 - CrossoverRate) then
                Crossover ( LA1, LA2 );
            if (Random > 1 - MutationRate) then
                Mutation ( LA1 ); Mutation ( LA2 );
            NewLAi+1 = LAi;
            NewLAi+2 = LA2 ;
            i=i+2;
        end for
        for i = 0 to n do
            LAi = NewLAi;
            u = Random *n;
            if ( Ju(LAi) < threshold Threshold(LAi ) ) then
                Reward(LAi , u );
            else
                Penalize(LAi , u );
            end for
            EvalFitness();
        end while
        ModificationProcess(); //for observe precedence constraints
    End Function
//Threshold(LAi) = Length( Specified Tour by LAi ) / |VG|;
//Ju(LAi) = (length of edge (u-1,u) in LAi + length of edge (u,u+1) in LAi ) / 2;

```

شکل ۹- شبه کد الگوريتم ترکيبي برای حل مساله مرتب سازی ترتيبی

## ۵- نتایج آزمایش ها

در اين بخش نتایج آزمایشي الگوريتم های حل مساله مرتب سازی ترتيبی که براساس آتماتاتی يادگیر، الگوريتم ژنتيك و الگوريتم ترکيبي پياده سازی شده اند، نشان داده شده است. اين نتایج بهبود قابل توجه الگوريتم ترکيبي را نسبت به روشهاي مبتنی بر آتماتاتی يادگیر و الگوريتم ژنتيك نشان می دهد. در آزمایش های انجام گرفته اندازه گراف ها (گراف ها از TSPLIB انتخاب شده اند) از ۲۲ تا ۲۸۰ راس و تعداد تکرارها از ۵۰ تا ۵۰۰ تکرار در نظر گرفته شده است. در الگوريتم ترکيبي و آتماتاتی يادگير عمق های ۱، ۴، ۷، ۱۰ و ۱۵ آزمایش شده اند. در الگوريتم ترکيبي و الگوريتم ژنتيكی روش Swap Mutation با نرخ ۲۵٪ و روش رتبه بندی برای انتخاب کروموزومها استفاده شده است و همچنین سایز جمعیت برابر با تعداد نودهای گراف در نظر گرفته شده است.

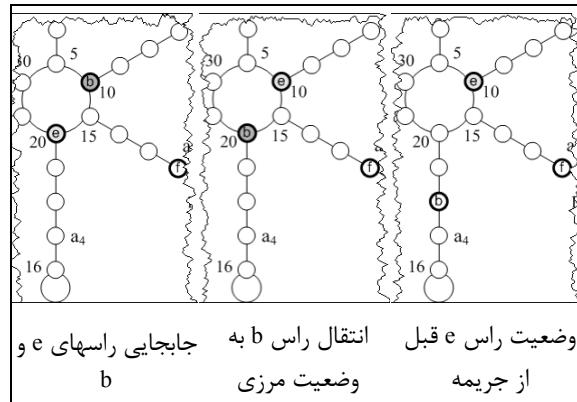
تور حاصل گردد. در اينصورت اگر راس پيدا شده در وضعیت مرزی قرار داشته باشد جای دو راس عوض می شود و در غير اينصورت ابتدا راس مشخص شده به وضعیت مرزی اقدام خود منتقل و سپس جابجايي صورت می پذيرد. نحوه حرکت چنین راسي در شکل ۸ نشان داده شده است.

```

Procedure Penalize( LA, u )
repeat
    for u = 1 to n do
        if (LA.State(U)) mod N <> 0 then
            Inc(LA.State(U));
        end for
    until at least one node appears in the boundary state
    bestTourLenght =  $\infty$ ;
    for U = 1 to n do
        Create permutation LA' from LA by swapping u and U
        if Length(Specified Tour by LA') < bestTourLenght then
            bestTourLenght = Length(Specified Tour by LA');
            bestNode = U;
        end if
    end for
    LA.State(bestNode) = LA.Action(bestNode)*N;
    LA.State(u) = LA.Action(u)*N;
    Swap(LA.State(u),LA.State(bestNode));
End Procedure

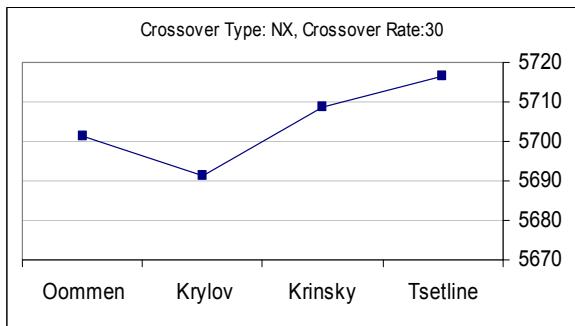
```

شکل ۷- شبه کد عملگر جريمه اقدام u از آتماتاتی LA با اتصالات مشابه آتماتاتی ستلين

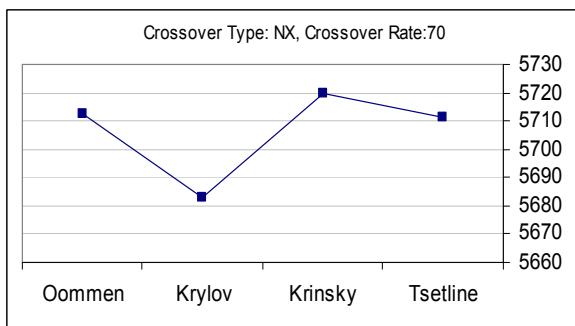


شکل ۸- نحوه جريمه کردن راسی که در وضعیت مرزی قرار دارد

با توجه به اينكه جايگشت های توليد شده بعد از اعمال عملگرهای ژنتيكی (ترکيب، جهش، پاداش و جريمه) ممکن است محدوديت های اوليتي در بين رؤوس را ارضاء نکند، بنابراین اين جايگشت ها باید بررسی شده و در صورت لزوم اصلاح شوند. بدليل اينكه اصلاح جايگشت های توليد شده برای ارضاء محدوديت های اوليتي بعد از هر عملگر ژنتيكی هزينه بر می باشد، بنابراین فقط در انتهای الگوريتم ترکيبي تور بدست آمده برای ارضاء محدوديت های اوليتي اصلاح می شود. فرآيند اصلاح زمان چند جمله ای دارد و باید طوري اعمال شود که طول (هزينه) تور، تا حد ممکن

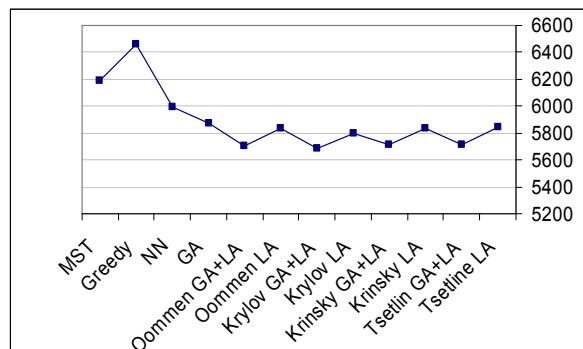


نمودار ۴- میانگین طول (هزینه) تور بدست آمده از الگوریتم ترکیبی با روش ترکیب New Crossover و نرخ ترکیب ۳۰

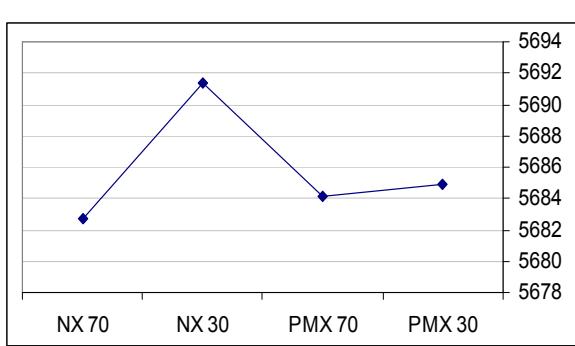


نمودار ۵- میانگین طول (هزینه) تور بدست آمده از الگوریتم ترکیبی با روش ترکیب New Crossover و نرخ ترکیب ۷۰

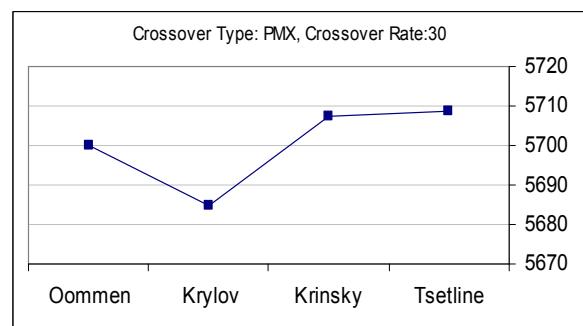
در نمودارهای ۱ الی ۷ مقایسه الگوریتم ترکیبی با سایر الگوریتم های حل مساله مرتب سازی ترتیبی بطور خلاصه آورده شده است. همانطور که از نتایج معلوم است، الگوریتم New ترکیبی مبتنی بر آtomاتای کرایلو با روش ترکیب Crossover و نرخ ترکیب ۷۰ بهتر از بقیه الگوریتم ها و سایر روشها و نرخ های ترکیب هم از لحاظ زمان اجرا و هم از لحاظ طول تور بدست آمده عمل می کند.



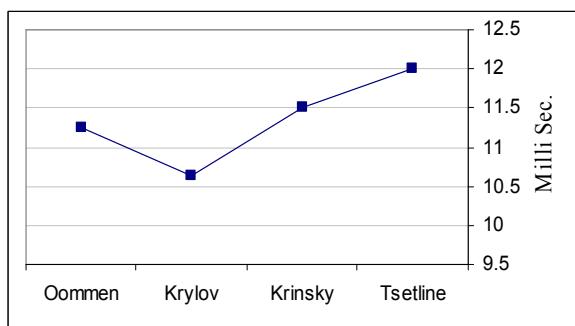
نمودار ۱- میانگین طول (هزینه) تور بدست آمده از الگوریتمهای مختلف



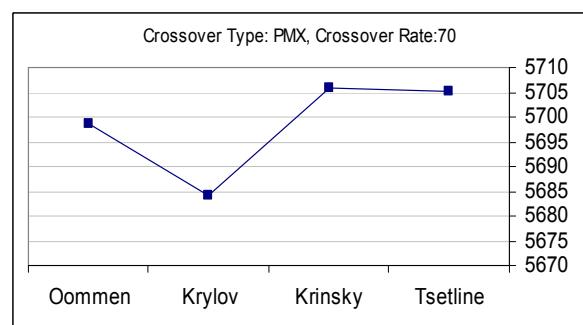
نمودار ۶- میانگین طول (هزینه) تور بدست آمده توسط الگوریتم ترکیبی مبتنی بر آtomاتای کرایلو با اعمال روشها و نرخ های ترکیب مختلف



نمودار ۲- میانگین طول (هزینه) تور بدست آمده از الگوریتم ترکیبی با روش ترکیب Partially Mapped Crossover و نرخ ترکیب ۳۰



نمودار ۷- میانگین زمان لازم برای الگوریتم ترکیبی مبتنی بر آtomاتاهای ستلین، کرینسکی، کرایلو و اومن



نمودار ۳- میانگین طول (هزینه) تور بدست آمده از الگوریتم ترکیبی با روش ترکیب Partially Mapped Crossover و نرخ ترکیب ۷۰

## ۶- نتیجه گیری و پیشنهادها

- گراف ها، بویژه گراف های برچسب دار، ابزار های قدرتمند و پراستفاده ای هستند که به طور گسترده در کاربردهای کامپیوتر مورد استفاده قرار می گیرند. یکی از مسائل بسیار مهم در تئوری گراف ها، پیدا کردن تور مرتب سازی ترتیبی می باشد. محققان بیش از دو دهه بر روی این مساله کار کرده اند، ولی با توجه به این حقیقت که هنوز الگوریتمی از درجه چند جمله ای برای حل این مساله وجود ندارد، با استفاده از پژوهش ها در این زمینه همچنان ادامه دارد. با استفاده از روش های جستجوی مناسب و ترکیب آنها، می توان الگوریتم های بهینه برای این مساله پیدا نمود. همچنین با خوش بندی گره های گراف و اجرای الگوریتم ترکیبی بر روی هر خوش بطور مستقل، می توان به نتایج بهتری رسید و همچنین استفاده از الگوریتم ژنتیک چند جمعیته می تواند نتایج را بهبود دهد.
- ### مراجع
- [۱] میبدی، محمد رضا و بیگی، حمید. "حل مساله تناظر گراف توسعه آtomاتاهای یادگیر". دانشکده مهندسی کامپیوتر. دانشگاه صنعتی امیرکبیر. تهران. ایران. ۱۳۷۹.
  - [۲] میبدی، محمد رضا و رضاپور میرصالح، مهدی. "یک روش ترکیبی (GA+LA) برای حل مساله تناظر گراف". دانشکده مهندسی کامپیوتر. دانشگاه صنعتی امیرکبیر. تهران. ایران. ۱۳۸۲.
  - [۳] Bager Zarei, M. R. Meybodi, and Mortaza Abbaszadeh, "A Hybrid Method for Solving Traveling Salesman Problem", Proceedings of the 6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007), IEEE Computer Society, 11-13 July 2007, Melbourne, Australia, 394-399.
  - [۴] D. S. Johnson, and L. A. McGeoch, "Experimental Analysis of Heuristics for the STSP", in the Traveling Salesman Problem and its Variations, G. Gutin and A. Punnen, Editors, Kluwer Academic Publishers, 2002, Boston, 369-443.
  - [۵] D. S. Johnson, G. Gutin, L. A. McGeoch, A. Yeo, W. Zhang, and A. Zverovich, "Experimental Analysis of Heuristics for the ATSP", in the Traveling Salesman Problem and its Variations, G. Gutin and A. Punnen, Editors, Kluwer Academic Publishers, 2002, Boston, 445-487.
  - [۶] D. S. Johnson, "A Theoretician's Guide to the Experimental Analysis of Algorithms", to appear in Proceedings of the 5th and 6th DIMACS Implementation Challenges, M. Goldwasser, D. S. Johnson, and C. C. McGeoch, Editors, American Mathematical Society, Providence, 2002.
  - [۷] J. Cirasella, D.S. Johnson, L.A. McGeoch, and W. Zhang, "The Asymmetric Traveling Salesman Problem: Algorithms, Instance Generators, and Tests", in Algorithm Engineering and Experimentation, Third International Workshop, ALENEX 2001, Lecture Notes in Computer Science, Vol. 2153, Springer, 2001, Berlin, 32-59.
  - [۸] K. Bryant, "Genetic Algorithms and the Traveling Salesman Problem", Thesis, 2000, Harvey Mudd College, Dept. of Mathematics.
  - [۹] H. Beigy, and M. R. Meybodi, "Optimization of Topology of Neural Networks Using Learning Automata", Proceedings of