

یک روش ترکیبی برای حل مساله فروشنده دوره گرد

محمد رضا میبیدی

باقر زارعی

دانشکده مهندسی 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 گراف ها، مساله فروشنده دوره گرد می باشد. آتوماتاهای یادگیر و الگوریتم های ژنتیکی هر دو از ابزارهای جستجو می باشند که برای حل بسیاری از مسائل $NP-Complete$ بکار برده می شوند. در این مقاله یک الگوریتم ترکیبی برای حل مساله فروشنده دوره گرد پیشنهاد شده است. این الگوریتم از دو روش الگوریتم های ژنتیکی و آتوماتاهای یادگیر بطور همزمان برای جستجو در فضای حالت استفاده می نماید. نشان داده شده است که با استفاده همزمان از آتوماتای یادگیر و الگوریتم ژنتیک در فرایند جستجو، سرعت رسیدن به جواب افزایش چشمگیری پیدا می کند و همچنین از بدام افتادن الگوریتم در حداقل های محلی جلوگیری می نماید. نتایج آزمایش ها، برتری الگوریتم ترکیبی را نسبت به الگوریتم ژنتیکی و آتوماتاهای یادگیر نشان می دهد.

کلمات کلیدی: مساله فروشنده دوره گرد، آتوماتای یادگیر، الگوریتم ژنتیک

۱- مقدمه

آتوماتاهای یادگیر و الگوریتم های ژنتیکی، هر دو ابزار جستجوی عمومی می باشند که برای حل بسیاری از مسائل $NP-Complete$ از جمله افزایش اشیاء، افراز گراف، بهینه سازی صفحه کلید و پیدا کردن ساختار بهینه شبکه عصبی و ... بکار برده شده است. در این مقاله یک الگوریتم ترکیبی برای حل مساله فروشنده دوره گرد پیشنهاد شده است. این الگوریتم از دو روش الگوریتم های ژنتیکی و آتوماتاهای یادگیر بطور همزمان برای جستجو در فضای حالت استفاده می نماید. نشان داده شده است که با استفاده همزمان از آتوماتای یادگیر و الگوریتم ژنتیک در فرایند جستجو، سرعت رسیدن به جواب افزایش چشمگیری پیدا می کند. نتایج آزمایش ها، برتری این الگوریتم را نسبت به الگوریتم ژنتیکی [8]، الگوریتم مبتنی بر آتوماتاهای یادگیر، الگوریتم نزدیکترین همسایه ملاقات نشده [9] و الگوریتم حریصانه [4] نشان می دهد.

ادامه مقاله بدین صورت سازماندهی شده است. بخش دوم به تعریف مساله اختصاص داده شده است. توضیح مختصری از

گراف ها ابزارهای قدرتمندی هستند که به طور گسترده در کاربردهای متعددی مورد استفاده قرار می گیرند. یکی از مسائل بسیار مهم در تئوری گراف ها، مساله فروشنده دوره گرد می باشد. مساله فروشنده دوره گرد، تعمیم یافته مساله مشهور سیکل همیلتنی است. فرض کنید که یک گراف کامل داریم که هر یال $(u, v) \in E$ یک هزینه صحیح نامنفی $c(u, v)$ را دارد. در مساله فروشنده دوره گرد، راسهای گراف معادل شهرها، یالهای گراف معادل مسیر ارتباطی بین شهرها و هزینه یالها نشاندهنده طول مسیر ارتباطی بین شهرها می باشد. فروشنده باید با شروع از یک مبدا، تمامی شهرها را دقیقاً یک بار ملاقات کرده و به شهر مبدا بازگردد طوری که هزینه کل تور حداقل گردد. اگر گراف متقارن (غیر جهتدار) باشد مساله را مساله فروشنده دوره گرد متقارن و اگر گراف نامتقارن (جهتدار) باشد مساله را مساله فروشنده دوره گرد نامتقارن می گویند [9-1].

الگوریتم های ژنتیک و اتوماتاهای یادگیر در بخش های ۳ و ۴ آورده شده است. بخش ۵، الگوریتم ترکیبی را که برای حل مساله فروشنده دوره گرد بکار گرفته شده است، توضیح می دهد. در بخش ۶ نتایج آزمایش ها نشان داده شده اند و در بخش ۷ نتیجه گیری به عمل آمده است. مراجع استفاده شده در این مقاله در بخش ۸ آورده شده است.

۲- تعریف مساله

یک گراف وزن دار به صورت سه تایی $G = (V, E, \alpha)$ نشان داده می شود که در آن V مجموعه غیر تهی از راسها، $E \subset V \times V$ مجموعه یالها و $\alpha: V \rightarrow R_V$ تابعی از V به R_V است که R_V مجموعه وزن یالها است.

بعضی از مسائل وجود دارند که با افزایش بعد آنها، زمان حلشان به طور نمایی افزایش می یابد. این مسائل، مسائل بهینه سازی ترکیبی هستند، که زمان حل آنها به صورت تابعی غیر چند جمله ای است. مساله فروشنده دوره گرد یکی از آنها می باشد که حل مساله به معنای پیدا کردن بهترین تور، در مقایسه با تورهای شناخته شده قبلی نمی باشد بلکه همچنین باید ثابت کرد که توری با هزینه کمتر از تور پیدا شده نیز وجود ندارد. با توجه به اینکه مساله فروشنده دوره گرد، یک مساله بهینه سازی ترکیبی است، بنابراین هدف یافتن جایگشتی از رئوس می باشد، بطوریکه جایگشت مورد نظر حداقل هزینه ممکن را داشته باشد یا به عبارت دیگر طول تور (مجموع وزن یالهای) مشخص شده توسط جایگشت حداقل باشد.

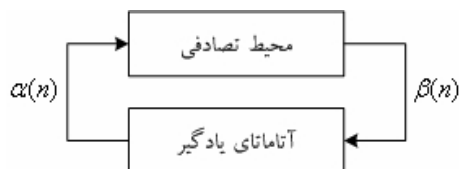
۳- الگوریتم های ژنتیک

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^۱ (جمعیت اولیه) به صورت تصادفی ساخته شده و تابع هدف برای تک تک آنها ارزیابی می شود. اگر شرط رسیدن به جواب برقرار نباشد (به جواب بهینه نرسیده باشیم)، نسل بعدی با انتخاب والدین

براساس میزان برازندگی شان تولید می شود و فرزندان با احتمال ثابتی دچار جهش می شوند. سپس میزان برازندگی 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. عمده ترین مزایای این روش در مقایسه با روش های متداول عبارتند از: جستجوی موازی در عوض جستجوی ترتیبی، عدم نیاز به هرگونه اطلاعات کمکی نظیر روش حل مساله، قطعی نبودن الگوریتم، پیاده سازی آسان و رسیدن به چند گزینه مطلوب. برای اطلاعات بیشتر در باره الگوریتم های ژنتیک می توان به [11] مراجعه کرد.

۴- اتوماتاهای یادگیر

یادگیری در اتوماتاهای یادگیر، انتخاب یک اقدام^۲ بهینه از میان یک مجموعه از اقدام های مجاز اتوماتا می باشد. این اقدام روی یک محیط تصادفی اعمال می شود و محیط به این اقدام اتوماتا بوسیله یک پاسخ تصادفی از مجموعه پاسخ های مجاز جواب می دهد. پاسخ محیط بصورت آماری به اقدام اتوماتا وابسته است. اصطلاح محیط شامل اجتماع تمام شرایط خارجی و تاثیرات آنها روی عملکرد اتوماتا می باشد. اتصال یک اتوماتای یادگیر با محیط در شکل ۱ نشان داده شده است. برای اطلاعات بیشتر در باره اتوماتاهای یادگیر می توان به [12][14][15][17] مراجعه کرد.



شکل ۱- اتصال اتوماتای یادگیر با محیط

اتوماتاهای یادگیر دارای کاربردهای فراوانی می باشد. بعضی از این کاربردها عبارتند از: مسیریابی در شبکه های ارتباطی، فشرده سازی تصاویر، شناسایی الگو، زمانبندی فرآیندها در شبکه های کامپیوتری، تئوری صف، کنترل دسترسی در شبکه های انتقال ناهمزمان، کمک به آموزش شبکه های عصبی، دسته بندی و افراز اشیاء و پیدا کردن ساختار بهینه برای شبکه های عصبی [10][13][16][18][17].

برای یک گراف با اندازه n ، $n!$ جایگشت مختلف از رئوس

$\{\phi_{N+1}, \phi_{N+2}, \dots, \phi_{2N}\}$ و ... و $\{\phi_{(K-1)N+1}, \phi_{(K-1)N+2}, \dots, \phi_{KN}\}$ افزای می شود و راس های گراف بر اساس این که در کدام وضعیت قرار داشته باشند دسته بندی می گردند. اگر گره u از گراف در مجموعه وضعیت های $\{\phi_{(j-1)N+1}, \phi_{(j-1)N+2}, \dots, \phi_{jN}\}$ قرار داشته باشد در اینصورت راس u در ترتیب ملاقات کردن شهرها، زامین شهر می باشد. در مجموعه وضعیت های اقدام j ، به وضعیت $\phi_{(j-1)N+1}$ وضعیت داخلی و به وضعیت ϕ_{jN} ، وضعیت مرزی گفته می شود. گره ای که در وضعیت $\phi_{(j-1)N+1}$ قرار دارد گره با اهمیت بیشتر و گره ای در وضعیت ϕ_{jN} گره با اهمیت کمتر نامیده می شود.

در اثر پاداش دادن یا جریمه کردن یک اقدام، وضعیت راس وابسته به آن اقدام، تغییر می کند. اگر راسی در وضعیت مرزی یک اقدام قرار داشته باشد، جریمه شدن آن باعث تغییر اقدام آن و در نتیجه باعث ایجاد جایگشت جدیدی می شود. به عنوان مثال شکل ۲ را که ماتریس مجاورت یک گراف کامل را نشان می دهد در نظر بگیرید.

	a	b	c	d	e	f
a	0	5	8	7	2	10
b	5	0	3	8	7	5
c	8	3	0	5	7	1
d	7	8	5	0	4	3
e	2	7	7	4	0	6
f	10	5	1	3	6	0

شکل ۲- گراف کامل با ۶ راس

جایگشت $\langle c, b, f, d, a, e \rangle$ از گراف شکل ۲ توسط یک اتوماتای یادگیر با اتصالات مشابه اتوماتای ستلین در شکل ۳ نشان داده شده است. این اتوماتا 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 داخلی و مجموعه وضعیت های $\{5, 10, 15, 20, 25, 30\}$ وضعیت های مرزی اتوماتا هستند. در ابتدا هر یک از راسهای گراف در وضعیت مرزی اقدام مربوطه قرار دارند. در الگوریتم ترکیبی هر ژن از کروموزوم معادل یک اقدام اتوماتا می باشد و لذا می توان در ادامه این دو واژه را به جای یکدیگر بکار برد. این اتوماتای یادگیر (کروموزوم ژنتیک) دارای ۶ اقدام (ژن) می باشد و هر اقدام دارای ۵ وضعیت داخلی است.

وجود دارد و در صورتیکه از اتوماتاهای یادگیر برای حل کردن مساله فروشنده دوره گرد استفاده شود، اتوماتا باید $n!$ اقدام داشته باشد که تعداد زیاد اقدام ها سرعت همگرایی اتوماتا را کاهش می دهد. به همین جهت اتوماتای مهاجرت اشیاء^۳ توسط اومن^۴ و ما^۵ پیشنهاد شده است.

۵- الگوریتم جستجوی ترکیبی برای حل مساله فروشنده دوره گرد

با ترکیب الگوریتم ژنتیک و اتوماتای 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 اتوماتای 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 اتوماتای یادگیر از نوع مهاجرت اشیاء نشان داده می شود. بطوریکه هر کدام از ژنهای کروموزوم به یکی از اقدامهای اتوماتا نسبت داده می شود و در یک عمق مشخصی از آن اقدام قرار می گیرد.

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

$\phi = \{\phi_1, \phi_2, \dots, \phi_{KN}\}$ مجموعه وضعیت ها و N عمق حافظه برای اتوماتا می باشد. مجموعه وضعیت های این اتوماتا به K زیر مجموعه $\{\phi_1, \phi_2, \dots, \phi_N\}$ و

3 Object Migrating Automata

4 Oominnen

5 Ma

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

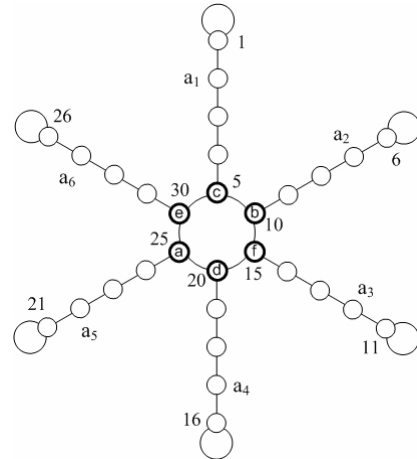
عملگرها:

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

الف) عملگر انتخاب^۸: برای انتخاب اتوماتاهای یادگیر (کروموزوم ها) برای عملگرهای جهش و ترکیب می توان از یکی از روشهای رتبه بندی، سازوکار چرخ رولت و یا Tournament استفاده کرد.

ب) عملگر ترکیب یا جابجایی^۹: برای انجام دادن این عملگر می توان از یکی از روشهای Partially Mapped Crossover، Ordered Crossover و Cycle Crossover 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 ژن i و j در یکی از دو کروموزوم والد انتخاب می شوند. سپس همین دو ژن در کروموزوم والد دیگر نیز انتخاب می شوند. مجموعه ژنهای با شماره های i و j را مجموعه جابجایی می نامیم. سپس ژن های هم شماره در دو مجموعه جابجایی با یکدیگر جابجا می شوند. با این عمل دو کروموزوم جدید حاصل می شوند که اصطلاحاً فرزندان دو اتوماتای والد خوانده می شوند. به عنوان مثال فرض کنید که اتوماتاهای LA2 و LA5 از جمعیت تشکیل شده قبل به عنوان والد انتخاب شوند. با انتخاب تصادفی دو محل a_2 و a_3 مجموعه جابجایی $\{a_2, a_3\}$ حاصل می شود و در نهایت مطابق شکل ۵ با جابجایی اقدام های متناظر در فاصله جابجایی، دو کروموزوم جدید حاصل می شود.

پ) عملگر جهش^{۱۰}: برای انجام دادن این عملگر می توان از یکی از روشهای Insertion Mutation، Swap Mutation و Inversion Mutation استفاده کرد. به عنوان مثال در روش Swap Mutation، دو اقدام (ژن) از یک اتوماتا (کروموزوم) به صورت تصادفی انتخاب شده و جابجا می شوند.



شکل ۳- نمایش جایگشت $\langle c, b, f, d, a, e \rangle$ به وسیله اتوماتای یادگیر با اتصالات مشابه اتوماتای ستلین

جمعیت اولیه:

با فرض اینکه تعداد اعضای جمعیت n باشد، $n-1$ عضو جمعیت با ایجاد $n-1$ جایگشت تصادفی تولید می شوند. برای تولید آخرین عضو جمعیت، از روش نزدیکترین همسایه ملاقات نشده استفاده می کنیم. به این جایگشت، جایگشت تقریبی^۶ می گوئیم. آخرین عضو اضافه شده به جمعیت بیشترین تشابه را با جواب نهایی دارد.

به عنوان مثال نحوه تشکیل جمعیت اولیه برای گراف شکل ۲ با فرض $n=6$ در ادامه توضیح داده شده است. پنج عضو اول جمعیت به وسیله پنج جایگشت تصادفی $\langle e, f, b, d, a, c \rangle$ ، $\langle d, e, f, b, c, a \rangle$ ، $\langle b, d, e, a, f, c \rangle$ ، $\langle c, f, b, e, d, a \rangle$ و $\langle b, d, c, a, e, f \rangle$ ایجاد می شود. برای ایجاد جایگشت ششم از روش نزدیکترین همسایه ملاقات نشده استفاده می کنیم. با فرض اینکه راس شروع a باشد جایگشت ششم بصورت $\langle a, e, d, f, c, b \rangle$ می باشد. جمعیت اولیه حاصل از گراف شکل ۲ در شکل ۴ نشان داده شده است. در ابتدا هرگره در وضعیت مرزی اقدام خود قرار دارد.

تابع برازندگی^۷:

در الگوریتم های ژنتیک تابع برازندگی، شاخص زنده ماندن کروموزوم ها است. در مساله فروشنده دوره گرد هدف یافتن جایگشتی (توری) مثل σ است که هزینه آن کمینه باشد، لذا برازندگی یک اتوماتا در مساله فروشنده دوره گرد به صورت زیر تعریف می شود.

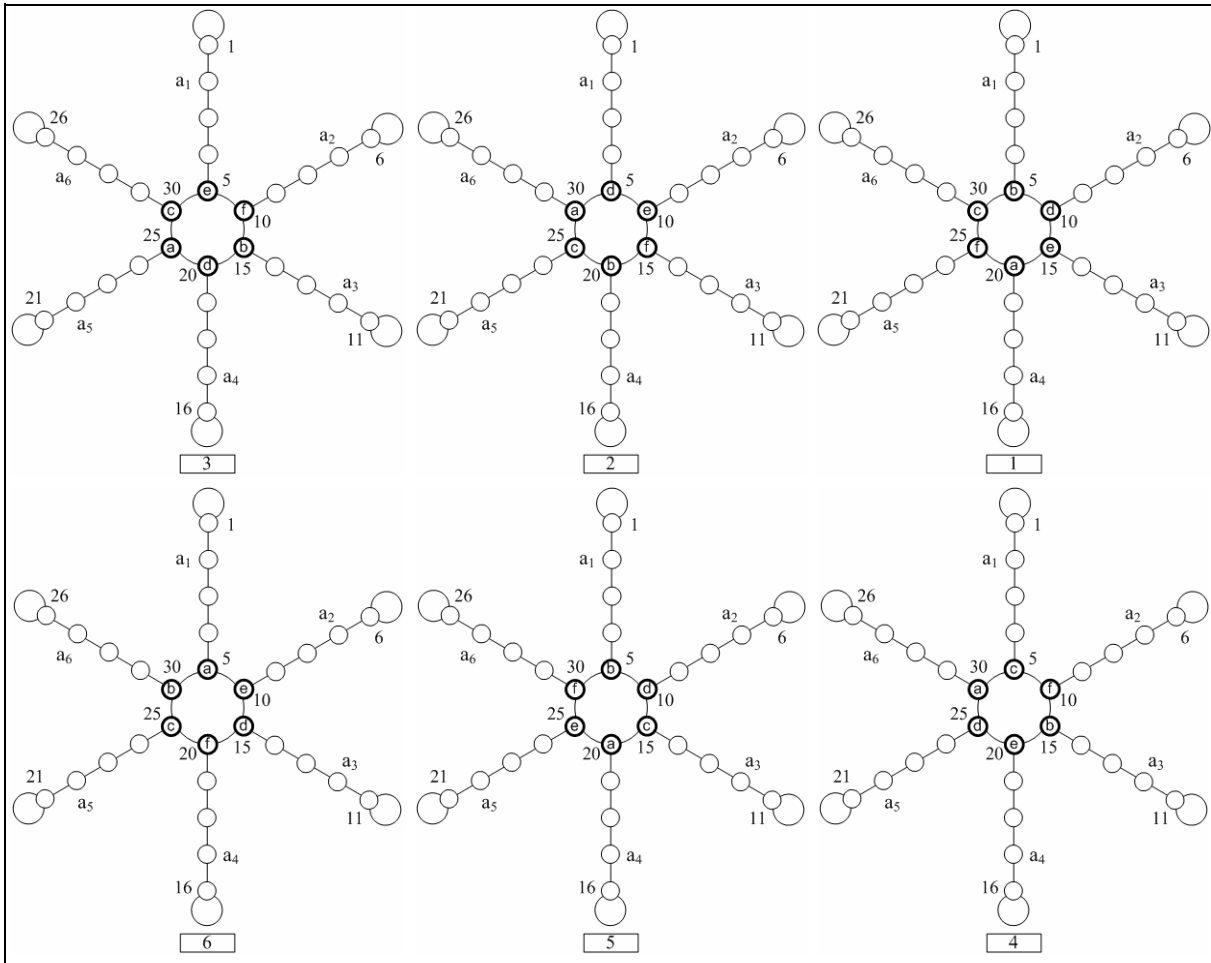
8 Selection Operator

9 Crossover Operator

10 Mutation Operator

6 Approximate Mapping

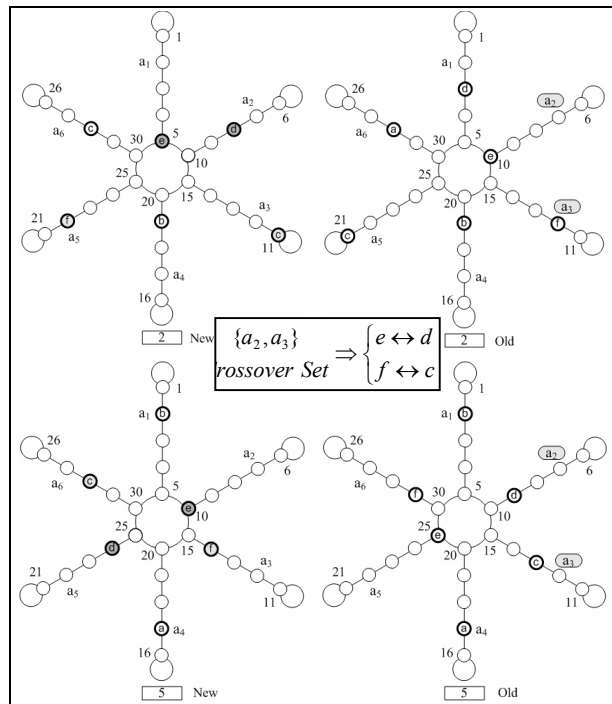
7 Fitness Function



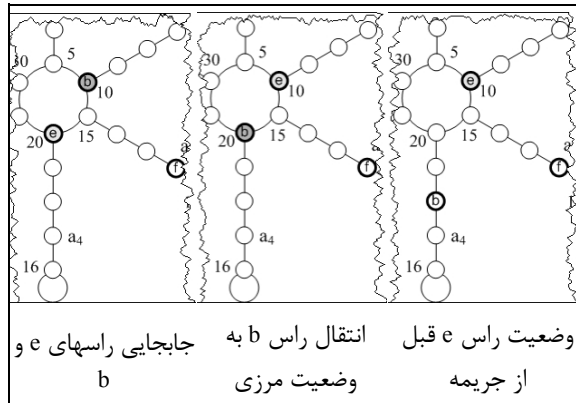
شکل ۴- جمعیت اولیه برای گراف شکل ۲

ت) عملگر جریمه و پاداش^{۱۱}: از آنجاییکه هر کروموزوم به صورت یک اتوماتای یادگیر نشان داده شده است، در هر یک از اتوماتاها پس از بررسی میزان برازندگی یک ژن (راس یا اقدام) که به صورت تصادفی انتخاب می شود، آن ژن پاداش یا جریمه می شود. در اثر پاداش دادن یا جریمه کردن یک ژن، وضعیت ژن در مجموعه وضعیت های اقدام مربوطه، تغییر می کند. اگر ژنی در وضعیت مرزی یک اقدام قرار داشته باشد، جریمه شدن آن باعث تغییر اقدام آن و در نتیجه باعث ایجاد جایگشت جدیدی می شود. نرخ این عملگر باید پایین باشد زیرا این عملگر، یک عملگر جستجوی تصادفی است و اگر با نرخ بالا اعمال شود باعث کاهش در کارایی الگوریتم می شود. عملگر جریمه و پاداش با توجه به نوع اتوماتای یادگیر متفاوت می باشد.

به عنوان مثال در اتوماتای با اتصالات مشابه اتوماتای ستلین، اگر راس b در مجموعه وضعیت های



شکل ۵- نحوه انجام عملگر جابجایی (New Crossover)



شکل ۷- نحوه جرمه کردن راسی که در وضعیت مرزی قرار دارد

در شکل ۸ شبه کد الگوریتم ترکیبی برای حل مساله فروشنده دوره گرد آورده شده است.

```

Function TSP_Solver(G) : TSP_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 = LA1;
      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
  End Function
  //Threshold(LAi) = Lenght( Specified Tour by LAi ) / |VG|;
  //Ju(LAi) = (lenght of edge (u-1,u) in LAi + lenght of edge (u,u+1) in LAi) / 2;

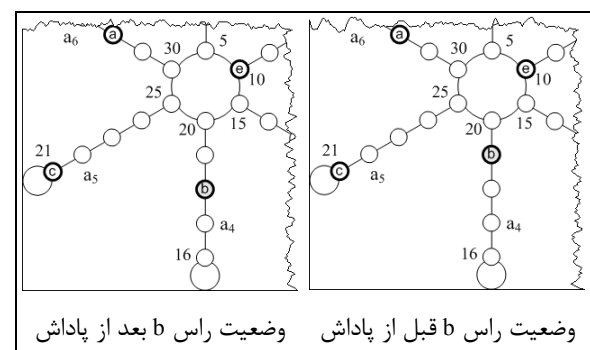
```

شکل ۸- شبه کد الگوریتم ترکیبی برای حل مساله فروشنده دوره گرد

۶- نتایج آزمایش ها

در این بخش نتایج آزمایشی الگوریتم های حل مساله فروشنده دوره گرد که براساس اتوماتای یادگیر، الگوریتم ژنتیک و الگوریتم ترکیبی پیاده سازی شده اند، نشان داده شده است. این نتایج بهبود قابل توجه الگوریتم ترکیبی را نسبت به روشهای مبتنی بر اتوماتای یادگیر و الگوریتم ژنتیک نشان می دهد. در آزمایش های انجام گرفته اندازه گراف ها (گراف ها از TSPLIB انتخاب شده اند) از ۲۲ تا

{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 برازندگی یک راس از مقدار آستانه بزرگتر باشد در اینصورت تور برقرار شده مناسب نبوده و این راس جرمه می شود. نحوه حرکت چنین راسی برای دو حالت مختلف در زیر آمده است.

(الف) راس در وضعیتی غیر از وضعیت مرزی قرار داشته باشد: جرمه نمودن این راس سبب کم اهمیت شدن این راس شده 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 تور حاصل گردد. در اینصورت اگر راس پیدا شده در وضعیت مرزی قرار داشته باشد جای دو راس عوض می شود و در غیر اینصورت ابتدا راس مشخص شده به وضعیت مرزی اقدام خود منتقل و سپس جابجایی صورت می پذیرد. نحوه حرکت چنین راسی در شکل ۷ نشان داده شده است.

- C.L. Monma, and G. Nemhauser, eds), Elsevier Science B.V., 1995, 225-330.
- [6] P. Moscato, and M.G. Norman, "An Analysis of the Performance of Traveling Salesman Heuristics on Infinite-Size Fractal Instances in the Euclidean Plane", Oct. 1994.
- [7] P. Merz, and B. Freisleben, "Genetic Local Search for the TSP: New Results", in Proceedings of the 1997 IEEE.
- [8] B. Freisleben, and P. Merz, "A Genetic Local Search Algorithm for Solving Symmetric and Asymmetric Traveling Salesman Problems", appeared in Proceedings of the 1996 IEEE International Conference on Evolutionary Computation, 1996, Nagoya, Japan, 616-621.
- [9] B. Freisleben, and P. Merz, "New Genetic Local Search Operators for the Traveling Salesman Problem", in Proceedings of the 4th Conference on Parallel Problem Solving from Nature - PPSN IV, (H.-M. Voigt, W. Ebeling, I. Rechenberg, H.-P. Schwefel, eds.), Vol. 1141 of Lecture Notes in Computer Science, 1996, 890-899.
- [10] H. Beigy, and M. R. Meybodi, "Optimization of Topology of Neural Networks Using Learning Automata", Proceedings of 3th Annual International Computer Society of Iran Computer Conference CSICC-98, 1999, Tehran, Iran, 417-428.
- [11] D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning", Reading, MA, Addison-Wesley, 1989.
- [12] P. Mars, K. S. Narendra, and M. Chrystall, "Learning Automata Control of Computer Communication Networks", Proceedings of Third Yale Workshop on Application of Adaptive Systems Theory, 1983, Yale University.
- [13] A. A. Hashim, S. Amir, and P. Mars, "Application of Learning Automata to Data Compression", in Adaptive and Learning Systems, K. S. Narendra, Editor, New York, Plenum Press, 1986, 229-234.
- [14] K. S. Narendra, and M. A. L. Thathachar, "Learning Automata: An Introduction", Prentice-hall, Englewood cliffs, 1989.
- [15] M. R. Meybodi, and S. Lakshmivarhan, "A Learning Approach to Priority Assignment in a Two Class M/M/1 Queuing System with Unknown Parameters", Proceedings of Third Yale Workshop on Applications of Adaptive System Theory, 1983, Yale University, 106-109.
- [16] M. R. Meybodi, and H. Beigy, "New Class of Learning Automata Based Scheme for Adaptation of Backpropagation Algorithm Parameters", Proceedings of EUFIT-98, 7-10 Sep. 1998, Aachen, Germany, 339-344.
- [17] B. J. Oommen, and D. C. Y. Ma, "Deterministic Learning Automata Solution to the Keyboard Optimization Problem", IEEE Transaction on Computers, Vol. 37, No. 1, 1988, 2-3.
- [18] 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.
- [19] K. Bryant, "Genetic Algorithms and the Traveling Salesman Problem", Thesis, 2000, Harvey Mudd College, Dept. of Mathematics.

۲۸۰ راس و تعداد تکرارها از ۵۰ تا ۵۰۰ تکرار در نظر گرفته شده است. در الگوریتم ترکیبی و اتوماتای یادگیر عمق های ۱، ۴، ۷، ۱۰ و ۱۵ آزمایش شده اند. در الگوریتم ترکیبی و الگوریتم ژنتیکی روش Swap Mutation با نرخ ۲۵٪ و روش رتبه بندی برای انتخاب کروموزمها استفاده شده است و همچنین سائز جمعیت برابر با تعداد نودهای گراف در نظر گرفته شده است.

در جداول ۱ و ۲ مقایسه الگوریتم ترکیبی با سایر الگوریتم های حل مساله فروشنده دوره گرد بطور خلاصه آورده شده است. همانطور که از نتایج معلوم است، الگوریتم ترکیبی مبتنی بر اتوماتای کرایلو با روش ترکیب New 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.

۷- نتیجه گیری

در این مقاله یک الگوریتم ترکیبی برای حل مساله فروشنده دوره گرد پیشنهاد شده است. این الگوریتم از دو روش الگوریتم های ژنتیکی و اتوماتاهای یادگیر بطور همزمان برای جستجو در فضای حالت استفاده می نماید. نشان داده شده است که با استفاده همزمان از اتوماتای یادگیر و الگوریتم ژنتیک در فرایند جستجو، سرعت رسیدن به جواب افزایش چشمگیری پیدا می کند و همچنین از بدام افتادن الگوریتم در حداقل های محلی جلوگیری می نماید. نتایج آزمایش ها، برتری الگوریتم ترکیبی را نسبت به الگوریتم ژنتیکی و اتوماتاهای یادگیر نشان می دهد.

مراجع

- [1] 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.
- [2] 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.
- [3] M. Grötschel, and O. Holland, "Solution of Large-Scale Symmetric Traveling Salesman Problems", Mathematical Programming 51, 1991, 141-202.
- [4] M. Padberg, and G. Rinaldi, "A Branch-and-Cut Algorithm for the Resolution of Large-Scale Symmetric Traveling Salesman Problems", SIAM Review 33, 1991, 60-100.
- [5] M. Jünger, G. Reinelt, and G. Rinaldi, "The Traveling Salesman Problem", in Handbooks in Operations Research and Management Science, Vol. 7 (M.O. Ball, T. Magnanti,

MST	Greedy	NN	GA	آتوماتای اومن		آتوماتای کرایلو		آتوماتای کرینسکی		آتوماتای ستلین		ترکیب روش	ترکیب وزن
				GA+LA	LA	GA+LA	LA	GA+LA	LA	GA+LA	LA		
۶۱۹۲,۶۲۵	۶۴۵۶,۶۲۵	۵۹۸۹,۵	۵۸۷۴,۵۶۳	۵۷۰۰,۱۰۴	۵۸۴۲,۰۴۶	۵۶۸۴,۸۷۵	۵۷۹۹,۴	۵۷۰۷,۲۵۸	۵۸۳۲,۷۷۹	۵۷۰۸,۷۶۳	۵۸۴۲,۵۱۳	۳۰	PM X
			۵۸۷۴,۶۰۴	۵۶۹۸,۵۶۳	۵۸۳۸,۲۹۲	۵۶۸۴,۱۱	۵۸۰۰,۷۱۷	۵۷۰۵,۹۱۳	۵۸۳۸,۹۴۶	۵۷۰۵,۰۴	۵۸۴۰,۳۲۱	۷۰	
			۵۸۸۶,۴۵۸	۵۷۰۱,۴۵۸	۵۸۳۸,۳۳۸	۵۶۹۱,۳۷۹	۵۸۰۲,۴۷۵	۵۷۰۸,۶۶۳	۵۸۴۴,۴۹۵	۵۷۱۶,۴۵۸	۵۸۳۸,۷۲۱	۳۰	NX
			۵۸۶۵,۹۷۹	۵۷۱۲,۴۷۵	۵۸۳۴,۵۸۳	۵۶۸۲,۶۹۲	۵۷۹۷,۰۶۳	۵۷۱۹,۴۵۴	۵۸۳۶,۰۰۴	۵۷۱۱,۴	۵۸۳۶,۷۳۳	۷۰	
۶۱۹۲,۶۲۵	۶۴۵۶,۶۲۵	۵۹۸۹,۵	۵۸۷۵,۴۰۱	۵۷۰۳,۱۵	۵۸۳۸,۳۱۵	۵۶۸۵,۷۶۴	۵۷۹۹,۹۱۴	۵۷۱۰,۳۲۲	۵۸۳۸,۰۵۶	۵۷۱۰,۴۱۵	۵۸۳۹,۵۷۲	میانگین	

جدول ۱- میانگین طول (هزینه) تور بدست آمده از الگوریتم ترکیبی (GA+LA) مبتنی بر آتوماتاهای ستلین، کرینسکی، کرایلو و اومن و مقایسه آن با الگوریتم مبتنی بر آتوماتاهای یادگیر (LA) مبتنی بر آتوماتاهای ستلین،

کرینسکی، کرایلو و اومن، الگوریتم ژنتیک (GA)، نزدیکترین همسایه ملاقات نشده (NN)، حریصانه (Greedy) و درخت پوشای حداقل (MST)

GA+LA				الگوریتم	طول تور مورد نظر	گراف (از TSPLIB)
Oommen	Krylov	Krinsky	Tsetline			
۱	۱	۱	۱		۸۰	ulysses22 -۱
۲	۱	۱	۲		۱۹۰۰	Random30 -۲
۲	۲	۵	۳		۸۶۵۰	berlin52 -۳
۸	۷	۱۰	۱۰		۳۷۵۰	Random80 -۴
۶	۴	۵	۵		۱۹۸۵۰	lin105 -۵
۱۰	۱۰	۱۱	۱۰		۴۵۵۰	Random150 -۶
۱۹	۲۲	۲۱	۲۴		۴۷۸۰	tsp225 -۷
۴۵	۳۹	۳۷	۳۸		۳۱۴۰	a280 -۸
۱۱,۲۵	۱۰,۶۲۵	۱۱,۵	۱۲		میانگین زمان لازم برای رسیدن به توری با طول مورد نظر	

جدول ۲- میانگین زمان لازم برای الگوریتم ترکیبی مبتنی بر آتوماتاهای ستلین، کرینسکی، کرایلو و اومن