اولین همایش منطقه ای رویکردهای نوین در مهندسی کامپیوتر و فناوری اطلاعات
۵ خرداد ۱۳۹۰ - رودسر

یک روش مبتنی بر خوشه بندی با ترکیب الگوریتم ژنتیک و الگوریتم رقابت استعماری برای حل مسئله فروشنده دوره گرد با مقیاس بزرگ

محمد صادق گرشاسبی[۱]، مریم گرشاسبی[۲]

[۱] دانشجوی کارشناسی کامپیوتر- نرم افزار، موسسه آموزش عالی غیرانتفاعی سبلان اردبیل
M.S.Garshasbi@gmail.com
[۲] دانشجوی کارشناسی کامپیوتر- نرم افزار ، دانشگاه آزاد اسلامی واحد خلخال
Maryam.Gharshasbi@yahoo.com

چکیده

یکی از مسایل مهم در تئوری گراف ها مساله فروشنده دوره گرد می باشد که یک مساله NP-COMPLETE است اکثرمسائلی که می توان آنها را با مساله فروشنده دوره گرد مدل کرد دارای مقیاس خیلی بزرگ هستند که الگوریتم های موجود قادر به حل آنها در یک زمان قابل قبول نیستند. الگوریتم رقابت استعماری و الگوریتم ژنتیک هر دو از الگوریتم های تکاملی هستند که برای حل مسائل NP-COMPLETE به کار برده می شوند. در این مقاله یک روش ترکیبی از الگوریتم ژنتیک و الگوریتم رقابت استعماری برای حل مسئله فروشنده دوره گرد در مقیاس بزرگ پیشنهاد شده است. در این روش ابتدا مساله اصلی را به چند خوشه با مقیاس کوچک تقسیم کرده و سپس هر یک از خوشه ها را از طریق الگوریتم ژنتیک حل می کنیم. هر خوشه یک گره اصلی(استعمارگر) و تعدادی گره عضو (مستعمره) دارد که گره اصلی مناسبترین عضو برای هر خوشه می باشد. سپس خوشه ها برای تصاحب گره های فرعی با همدیگر رقابت می کنند و در نهایت خوشه ای که ضعیف تر است به صورت تدریجی گره های فرعی خود را از دست خواهد داد و حذف خواهند شد و خوشه های قوی تری با جذب این گره ها بر قدرت خود خواهند افزود و در نهایت یک خوشه واحد بوجود خواهد آمد که گره اصلی مناسبترین جواب مساله خواهد شد. نشان داده شده است که با استفاده از این روش در مقیاس های بزرگ، سرعت رسیدن به جواب افزایش پیدا می کند.

کلمات کلیدی

الگوریتم ژنتیک، الگوریتم رقابت استعماری، خوشه بندی، مساله فروشنده دوره گرد

۱- مقدمه

گراف ها ابزار های قدرتمندی هستند که به طور گسترده در کاربرد های متعددی مورد استفاده قرار می گیرند. یکی از مسائل بسیار مهم در تئوری گراف ها، مساله فروشنده دوره گرد می باشد. بسیاری از کاربرد های عمومی از جمله طراحی حلقه های شبکه های Sonet، مسیر هواپیما ها، مسیر یابی و ... را می توان با مساله فروشنده دوره گرد مدل کرد.

مساله فروشنده دوره گرد، تعمیم یافته مساله مشهور سیکل همیلتنی است. فرم کلی این مساله برای اولین بار در سال ۱۹۳۰ توسط Karl Menger مطرح شد و بعدا توسط Hassler Whitney و Merrill Flood ترویج داده شد. فرض کنید که یک گراف کامل داریم که هر یال $(u,v)\in E$ یک هزینه صحیح نامنفی $c(u,v)$ را دارد. فروشنده دوره گرد باید با شروع از مبدا، تمامی شهر ها را دقیقا یک بار ملاقات کرده و به شهر مبدا باز گردد بطوریکه هزینه کل تور حداقل گردد[۲].

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.

الگوریتم ژنتیک و الگوریتم رقابت استعماری، هر دو جزو الگوریتم های جستجوی عمومی می باشند که برای حل بسیاری از مسائل NP-Complete بکار برده شده است. در این مقاله یک روش ترکیبی (الگوریتم ژنتیک + الگوریتم رقابت استعماری) مبتنی بر خوشه بندی برای حل مساله فروشنده دوره گرد با مقیاس بزرگ پیشنهاد

شده است. در این روش ابتدا با استفاده از تکنیک خوشه بندی، مساله را به چند زیر مساله با مقیاس کوچک تقسیم کرده و سپس از طریق الگوریتم ژنتیک جواب بهینه هر خوشه بدست می آید، سپس از طریق الگوریتم رقابت استعماری جواب کل مساله با ترکیب خوشه ها و تبدیل آنها به یک خوشه بدست می آید. با استفاده از این روش در فرایند جستجو، سرعت رسیدن به جواب افزایش چشمگیری پیدا می کند و همچنین از به دام افتادن الگوریتم در حداقل های محلی جلوگیری می نماید. همچنین با استفاده از تکنیک خوشه بندی و اجرای الگوریتم ترکیبی بطور همزمان بر روی هر خوشه با یک سیستم چند پردازنده ای می توان زمان لازم برای حل مساله را به حداقل مقدار ممکن کاهش داد.

۲- معرفی الگوریتم ژنتیک

الگوریتم ژنتیک تکنیک جستجویی در علم کامپیوتر برای یافتن راه حل تقریبی برای بهینه سازی و مسائل جستجو است. الگوریتم ژنتیک نوع خاصی از الگوریتم های تکامل است که بر اساس ساختار ژن ها و کوروموزوم ها تشکیل شده است. این الگوریتم بر اساس اصل بقای بهترین ها است که طی آنها در یک ساختار تکاملی یک مجموعه تصادفی از جواب های اولیه مساله به تکامل می رسند و در واقع بهینه می شوند. الگوریتم ژنتیک در ابتدا با جمعیتی کاملا تصادفی آغاز می شود و در نسل ها ادامه می یابد. در هر نسل گنجایش تمام جمعیت ارزیابی می شود، چندین فرد مناسب در فرایندی تصادفی از نسل جاری انتخاب می شود (بر اساس شایستگی ها) و برای شکل دادن نسل جدید اصلاح می شود و در تکرار بعدی الگوریتم به نسل جاری تبدیل می شود. الگوریتم ژنتیک از قسمت های زیر تشکیل شده است:

- Gene (ژن)
- Chromosome (کروموزوم)
- Population (جمعیت)

در این الگوریتم هر کروموزوم از آرایه ای از ژن ها تشکیل شده است که به مجموعه ای از این کروموزوم ها جمعیت گفته می شود.

مناسب بودن یا نبودن جواب، با معیاری که از تابع هدف بدست می آید، سنجیده می شود. هر چه که یک جواب مناسب تر باشد، مقدار برازندگی بزرگتری دارد. برای آنکه شانس بقای چنین جوابی بیشتر شود، احتمال بقای آن، متناسب با مقدار برازندگی آن در نظر گرفته می شود. بنابراین کروموزومی که برازنده ترین است با احتمال بیشتری در تولید فرزندان شرکت می کند و دنباله های بیشتری از آن به وجود می آید[۵].

مهمترین عملگر در الگوریتم ژنتیک، عملگر ترکیب است. ترکیب، فرایندی است که در آن نسل قدیمی کروموزوم ها با یکدیگر مخلوط و ترکیب می شوند تا نسل تازه ای از کروموزوم ها بوجود بیاید. جفت هایی که در مرحله انتخاب، به عنوان والد در نظر گرفته شدند، در این قسمت ژنهایشان را با هم مبادله می کنند و اعضایی جدید بوجود می آورند. ترکیب اعضا با برازندگی بالا باعث بوجود آمدن اعضایی می شود که از برازندگی میانگین، برازندگی بیشتری دارند. ترکیب در الگوریتم ژنتیک باعث از بین رفتن پراکندگی یا تنوع ژنتیکی جمعیت می شود. زیرا اجازه می دهد ژن های خوب یکدیگر را بیابند.

عملگر جهش در الگوریتم ژنتیک روی هر یک از کروموزوم های حاصل از عملگر ترکیب عمل می کند. بدین ترتیب که به ازای هر بیت از کروموزوم، یک عدد تصادفی تولید می گردد. اگر مقدار این عدد تصادفی از مقدار Pm (احتمال انجام جهش) کمتر باشد، در آن بیت عمل جهش انجام می شود و در غیر اینصورت، در آن بیت عمل جهش انجام نمی گیرد. ژنی که انتخاب می شود به تصادف مقدار جدید به این ژن انتساب داده می شود. در واقع جهش یک ژن به معنای تغییر آن ژن است و وابسته به نوع کدگذاری، از روش های متفاوت جهش استفاده می شود[۵].

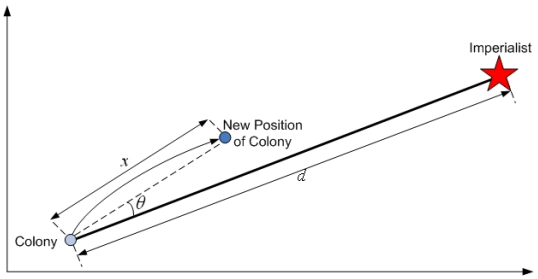
این الگوریتم با استفاده از عملگر های انتخاب، ترکیب و جهش ما را به حل بهینه ای می رساند که با روش های دیگر امکان پذیر نیست. این مراحل آنقدر تکرار می شود که به حل بهینه ای از جواب برسیم.

۳- معرفی الگوریتم رقابت استعماری

این الگوریتم با الهام‌گیری از یک فرایند اجتماعی سیاسی، نسبت به روش‌های مطرح شده دارای توانایی بالایی بوده و تا حد بسیار زیادی نیز، سریع می‌باشد. این الگوریتم همانند سایر روش‌های بهینه‌سازی تکاملی، با تعدادی جمعیت اولیه شروع می‌شود. در این الگوریتم، هر عنصر جمعیت، یک کشور نامیده می‌شود. کشورها به دو دسته مستعمره و استعمارگر تقسیم می‌شوند. هر استعمارگر، بسته به قدرت خود، تعدادی از کشورهای مستعمره را به سلطه خود درآورده و آن‌ها را کنترل می‌کند. سیاست جذب و رقابت استعماری، هسته اصلی این الگوریتم را تشکیل می‌دهند.

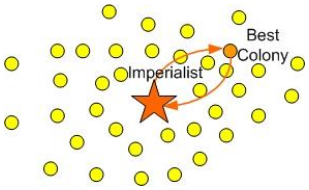
قدرت کل هر امپراطوری، به هر دو بخش تشکیل دهنده آن یعنی کشور امپریالیست (به عنوان هسته مرکزی) و مستعمرات آن، بستگی دارد.

در این الگوریتم کشور امپریالیست کشور مستعمره را جذب می‌کند. همانطور که در شکل (۱) دیده می شود، اگر فاصله بین کشور مستعمره تا استعمارگر برابر d باشد، حرکت کشور مستعمره به اندازه x و به سمت محل استعمارگر نظیر آن خواهد بود. البته این حرکت با زاویه θ منحرف می شود که مقدار حرکت x و زاویه θ به طور تصادفی تعیین می گردد .معمولا مقدار زاویه θ به طور یکنواخت در بازه $[-\gamma,\gamma]$ و مقدار حرکت x به طور یکنواخت در بازه $[0,\beta d]$ انجام می شود[۴].



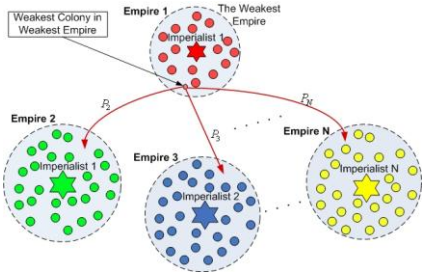
شکل (۱) شمای کلی حرکت مستعمرات به سمت امپریالیست[۱]

اگر در طول انجام الگوریتم و حرکت کشورها، یک کشور مستعمره قدرت بیشتری از استعمارگر نظیر خود پیدا کند، جای کشور مستعمره و استعمارگر عوض خواهد شد. به عبارت دیگر در مراحل بعدی اجرای الگوریتم، تمام کشورهای مستعمره استعمارگر قبلی، به استعمارگر جدید تعلق خواهند گرفت و حرکت این مستعمرات به سمت استعمارگر جدید خواهد بود[۱]. شکل (۲).



شکل (۲) تغییر جای استعمارگر و مستعمره[۱]

در هر مرحله از تکرار این الگوریتم، رقابتی استعماری میان استعمارگران برقرار است. در این رقابت، استعمارگری که نسبت به دیگر استعمارگران قدرت کمتری دارد، یکی از مستعمرات خود را از دست می دهد. به این ترتیب ضعیف ترین مستعمره از ضعیف ترین استعمارگر به طور تصادفی به یکی از استعمارگران دیگر ملحق می شود. احتمال انتساب این مستعمره جدید به هر یک از استعمارگران متناسب با میزان قدرت استعمارگران خواهد بود[۱]. شکل (۳).



شکل (۳) شمای کلی رقابت استعماری[۱]

اگر استعمارگری به دلیل از دست دادن مستعمرات خود، هیچ مستعمره ای نداشته باشد، آن استعمارگر خود به صورت مستعمره یک استعمارگر دیگر در خواهد آمد[۳].

مراحل الگوریتم رقابت استعماری به همین ترتیب ادامه می یابد تا بالاخره تعداد استعمارگران به یک برسد. در این حالت تمام کشورها، مستعمره یک استعمارگر هستند و الگوریتم به پایان می رسد[۱].

۴- روش پیشنهادی

در این روش جهت پیدا کردن تور فروشنده دوره گرد برای گراف هایی با مقیاس بزرگ (گراف هایی با تعداد گره های بسیار زیاد)، ابتدا گراف مورد نظر را با استفاده از یکی از روش های خوشه بندی گراف بنام Kmean، به خوشه هایی با اندازه کوچک تقسیم کرده و سپس با استفاده از الگوریتم ژنتیک جواب بهینه هر خوشه را بدست می آوریم. بهترین جمعیت بوجود آمده در هر خوشه که از طریق الگوریتم ژنتیک بدست می آید را به عنوان یک امپراتوری در نظر می گیریم، به این صورت که بهترین کروموزوم در جمعیت را استعمارگر و بقیه کروموزوم ها را به عنوان مستعمره در نظر می گیریم. از این مرحله رقابت استعماری شروع می شود به این صورت که امپراتوری هایی با قدرت بیشتر در صدد جذب مستعمره های امپراتوری های ضعیف تر خواهند بود و در نهایت به تدریج امپراتوری های ضعیف تر حذف خواهند شد و یک امپراتوری واحد بوجود می آید.

وقتی یک استعمارگر قویتر یکی از مستعمره های استعمارگر ضعیف تر را جذب می کند در این صورت استعمارگر قویتر مستعمره جدید را با هر یک از اعضای خود (خودش و مستعمره ها) ترکیب می کند که در این صورت فضای جستجوی آن امپراتوری افزایش پیدا می کند و فضای جستجوی امپراتوری ضعیف تر را در بر می گیرد.

رقابت استعماری به همین صورت تکرار می شود یعنی امپراتوری ها با جذب مستعمره ها و ترکیب آن با اعضای خود فضای جستجوی خود را افزایش می دهند و با هم رقابت می کنند تا زمانی که یک امپراتوری واحد بوجود بیاید یا به جواب بهینه قابل قبول برسیم.

۴-۱- هزینه کل یک امپراتوری

برای محاسبه هزینه کل یک امپراطوری داریم:

$$T.C_n = \frac{Cost(imperialist_n) + \xi\, mean\{Cost(colonies\ of\ empire_n)\}}{K} \quad (۱)$$

که در آن $T.C.\ n$ هزینه کل امپراطوری nام و ξ عددی مثبت است که معمولاً بین صفر و یک و نزدیک به صفر در نظر گرفته می‌شود. در حالت نوعی $\xi = 0.05$ در اکثر پیاده‌سازی ها به جوابهای مطلوبی منجر شده است.

نکته ای که در اینجا حائز اهمیت است این است که وقتی یک امپراتوری قوی تر مستعمره ای را از امپراتوری ضعیف تر جذب می کند و آن را با اعضای خودش ترکیب می کند در این صورت فضای جستوجوی آن افزایش پیدا می کند که باید هزینه کل امپراتوری متناسب با فضای جستجوی آن محاسبه شود که K فضای جستجو را

تعیین می کند که در اینجا مقدار K تعداد شهرهایی است که هر امپراتوری، عمل جستجو را در آن تعداد شهر انجام می دهد.

۴-۲- رقابت استعماری

فرض می‌کنیم که امپراطوری در حال حذف، ضعیف‌ترین امپراطوری موجود است. بدین ترتیب، در تکرار الگوریتم، یکی یا چند تا از ضعیف‌ترین مستعمرات ضعیف‌ترین امپراطوری را برداشته و برای تصاحب این مستعمرات، رقابتی را میان کلیه امپراطوری‌ها ایجاد می‌کنیم. مستعمرات مذکور، لزوماً توسط قویترین امپراطوری، تصاحب نخواهند شد، بلکه امپراطوری‌های قویتر، احتمال تصاحب بیشتری دارند. برای مدل‌سازی رقابت میان امپراطوری‌ها برای تصاحب این مستعمرات، ابتدا احتمال تصاحب هر امپراطوری را با در نظر گرفتن هزینه کل امپراطوری، به ترتیب زیر محاسبه می‌کنیم. ابتدا از روی هزینه کل امپراطوری، هزینه کل نرمالیزه شده آن را تعیین می‌کنیم[۱].

$$N.T.C._n = \max_i\{T.C._i\} - T.C._n \qquad (۲)$$

در این رابطه $T.C._n$، هزینه کل امپراطوری nام و $N.T.C._n$ نیز، هزینه کل نرمالیزه شده آن امپراطوری می‌باشد. هر امپراطوری که $T.C._n$ کمتری داشته باشد $N.T.C._n$ بیشتری خواهد داشت.

با داشتن هزینه کل نرمالیزه شده، احتمال (قدرت) تصاحب مستعمره رقابت، توسط هر امپراطوری، به صورت زیر محاسبه می‌شود[۱].

$$p_{p_n} = \left| \frac{N.T.C._n}{\sum_{i=1}^{N_{imp}} N.T.C._i} \right| \qquad (۳)$$

با داشتن احتمال تصاحب هر امپراطوری، مکانیزمی همانند چرخه رولت در الگوریتم ژنتیک مورد نیاز است تا مستعمره مورد رقابت را با احتمال متناسب با قدرت امپراطوری ها در اختیار یکی از آنها قرار دهد.

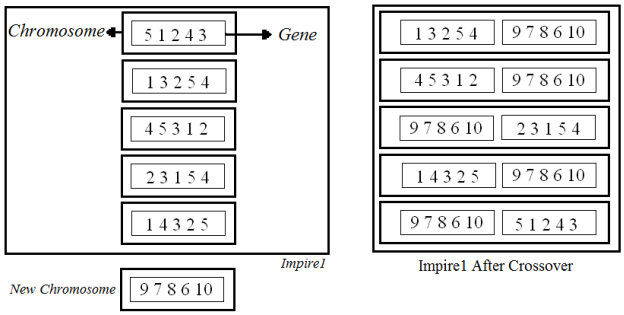
۴-۳- حذف امپراتوری های ضعیف

در جریان رقابت‌های استعماری، خواه ناخواه، امپراطوری های ضعیف به تدریج سقوط کرده و مستعمراتشان به دست امپراطوری‌های قوی‌تر می‌افتد. شروط متفاوتی را می‌توان برای سقوط یک امپراطوری در نظر گرفت. در اینجا، یک امپراطوری زمانی حذف شده تلقی می‌شود که مستعمرات خود را از دست داده باشد.

۴-۴- ترکیب مستعمره جذب شده با اعضای قبلی

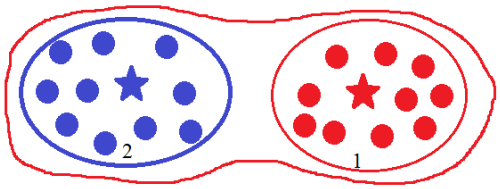
برای انجام این عمل همانند عملگر ترکیب (Crossover) در الگوریتم ژنتیک عمل می شود به این صورت که مستعمره جذب شده را به عنوان یک کروموزوم در نظر می گیریم که ممکن است متناسب با فضای جستجو خوشه خودش، تعداد متفاوتی ژن داشته باشد، و همچنین هر یک از اعضای قبلی امپراتوری را به عنوان یک کروموزوم در نظر می گیریم که می تواند در طول اجرای الگوریتم با توجه به فضای جستجویی که در بر می گیرد تعداد متفاوتی ژن داشته باشد، یعنی $1 \le Gene \le n$ که n تعداد خوشه ها در ابتدای شروع الگوریتم است و تعداد ژن می تواند در هر بار تکرار الگوریتم با توجه به جذب مستعمره های جدید متفاوت باشد، که تعداد ژن ها هرگز کاهش پیدا نمی کند.

برای مثال در اولین گام این روش، در مرحله رقابت استعماری وقتی مستعمره ای جذب می شود در این صورت یک کروموزوم با یک ژن بوجود می آید و اعضای امپراتوری هم فقط کروموزوم هایی با یک ژن هستند که با ترکیب اعضای امپراتوری با مستعمره جدید کروموزم هایی بوجود می آیند که از دو ژن تشکیل شده اند. یعنی هر ژن نشانگر مسیری در یک خوشه است، و به همین صورت در مرحله بعدی اگر این امپراتوری مستعمره ای را جذب کند مستعمره جدید را با اعضای خودش که کروموزوم هایی با دو عدد ژن هستند ترکیب خواهد کرد. بنابراین فضای جستجوی امپراتوری ها به همین ترتیب افزایش پیدا می کند. شکل (۴ و ۵).



شکل (۴)

در شکل (۴) منظور از ژن 51243 این است که هزینه این مسیر یعنی از ۵ به ۱ و از ۱ به ۲ و تا آخر در خوشه مورد نظر چقدر است. و با ترکیب این ژن با کروموزم جدید کروموزمی با دو ژن بوجود می آید که نشانگر هزینه مسیر در دو خوشه است(خوشه اول شامل شهر های ۱، ۲، ۳، ۴ و ۵ است و خوشه ۲ شامل شهر های ۶، ۷، ۸، ۹ و ۱۰ است).



شکل (۵): امپراتوری ۱ با جذب یک مستعمره از امپراتوری ۲ فضای جستجوی خود را افزایش می دهد

به همین ترتیب امپراتوری هایی که قدرتمند هستند به تدریج فضای جستجوی خود را افزایش می دهند و ممکن است که همه امپراتوری ها همه فضای جستجو را در بر بگیرند.

۴-۵- شرط پایان الگوریتم

زمانی که در جریان رقابت‌های استعماری امپراتوری های ضعیف سقوط کردند و یک امپراتوری واحد بوجود آمد یا به یک جواب قابل قبول رسیدیم در این صورت استعمارگر امپراتوری به عنوان جواب مسئله در نظر گرفته می شود.

۵- نتایج آزمایش ها

در این بخش نتایج آزمایشی روش پیشنهاد شده را که بر اساس الگوریتم ژنتیک و الگوریتم رقابت استعماری پیاده سازی شده اند در جدول (۱) نشان داده شده است. نتایج بدست آمده در جدول (۲) بهبود قابل توجه این روش نسبت به الگوریتم ژنتیک و الگوریتم رقابت استعماری که به تنهایی مسئله فروشنده دوره گرد را حل می کنند نشان می دهد. در آزمایش های انجام گرفته تعداد شهرها ۱۰۰۰۰ در نظر گرفته شده است که به ۱۳ خوشه تقسیم می شود، بنابراین ۱۳ امپراتوری خواهیم داشت. در الگوریتم ژنتیک احتمال جهش 0.3 و نرخ ترکیب 0.6 استفاده شده است، همچنین سایز جمعیت برابر با تعداد شهرهای هر خوشه در نظر گرفته شده است.

تمام آزمایش ها با استفاده از نرم افزار MATLAB، روی یک کامپیوتر شخصی با پردازنده Intel CORE Duo 2.10 GHz با حافظه 2 GB و در شرایط یکسان انجام شده است.

جدول (۱): مدت زمان رسیدن به جواب بهینه در هر خوشه

زمان رسیدن به طور بهینه (ثانیه)	تعداد شهر خوشه	شماره خوشه ها
۱۳.۴۹	۷۶۹	۱
۱۱.۷۲	۷۸۵	۲
۱۲.۶۶	۷۸۲	۳
۱۴.۴۸	۸۱۵	۴
۱۵.۱۲	۷۹۴	۵
۱۶.۱۷	۸۱۰	۶
۱۴.۷۷	۸۰۵	۷
۱۲.۸۶	۷۷۹	۸
۲۳.۱۴	۸۱۹	۹
۱۲.۸۷	۷۸۹	۱۰
۱۵.۶۱	۸۱۴	۱۱
۱۷.۱۶	۸۰۳	۱۲
۱۲.۵۴	۴۳۶	۱۳
۴.۸۹	-	رقابت استعماری
۱۹۲.۵۹ + ۴.۸۹	۱۰۰۰۰	مجموع

جدول (۲): مدت زمان رسیدن به جواب بهینه با الگوریتم های مختلف

روش پیشنهادی	الگوریتم ژنتیک	الگوریتم رقابت استعماری	
~۱۹۸	~۲۴۰	~۲۵۴	زمان(ثانیه)

۶- نتیجه

گراف ها، بویژه گراف های برچسب دار، ابزار های قدرتمند و پر استفاده ای هستند که به طورگسترده در کاربرد های کامپیوتر مورد استفاده قرار می گیرند. یکی از مسائل مهم در تئوری گراف ها، پیدا کردن تور فروشنده دوره گرد می باشد. با توجه به این حقیقت که هنوز الگوریتمی از درجه چند جمله ای برای حل این مسائل وجود ندارد، 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 خوشه بندی گره های گراف و اجرای الگوریتم ترکیبی بطور همزمان بر روی هر خوشه، می توان به نتایج بهتری رسید و در صورت استفاده از یک سیستم چندپردازنده ای می توان زمان لازم برای حل مساله اصلی را چندین برابر کاهش داد.

مراجع

[۱] آتش پز گرگری، **معرفی الگوریتم رقابت استعماری**، پایان نامه ی کارشناسی ارشد، مرکز عالی هوش مصنوعی و مهندسی کنترل دانشکدهی مهندسی برق، دانشگاه تهران.

[۲] میبدی، محمد رضا و اصغری، کیوان و زارعی، باقر، **یک روش ترکیبی مبتنی بر خوشه بندی برای حل مسئله TSP**، دانشکده برق، مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران، ۱۳۸۶.

[۳] بابایی، مرتضی و دادگر، حسن و کیمیا قلم، بهرام وکارولوکس، **بکارگیری الگوریتم رقابت استعماری برای حل مسئله فروشنده دوره گرد**.

[۴] سیگاری، محمد حسین و کارولوکس، **کاربرد الگوریتم رقابت استعماری برای انتخاب ویژگی در سیستم تشخیص چهره**، قطب کنترل و پردازش هوشمند دانشکده مهندسی برق و کامپیوتر، دانشگاه تهران.

[۵] گرشاسبی، محمد صادق، **الگوریتم ژنتیک و حل مسئله فروشنده دوره گرد**، موسسه آموزش عالی غیر انتفاعی سبلان اردبیل.