دسترسی در سایتِ http://jnrm.srbiau.ac.ir
سال سوم، شماره نهم، بهار ۱۳۹۶
شماره شاپا: ۱۶۸۲-۰۱۶۹

دانشگاه آزاد اسلامی، واحد علوم و تحقیقات

پژوهش‌های نوین در ریاضی

یک روش رقابت استعماری و یک مدل برنامه‌ریزی صحیح-آمیخته برای مساله مسیریابی وسیله نقلیه ظرفیت‌دار

مجید یوسفی خوشبخت*

استادیار، گروه ریاضی، دانشکده علوم، دانشگاه بوعلی سینا، همدان، ایران

تاریخ دریافت مقاله: ۹۶/۰۱/۱۵ تاریخ پذیرش مقاله: ۹۶/۰۲/۳۰

چکیده

مساله مسیریابی وسیله نقلیه یکی از مشهورترین مسائل تحقیق در عملیات است که از جایگاه بسیار مهمی در مسائل بهینه‌سازی ترکیباتی برخوردار است. در این مسئله ناوگانی از وسایل نقلیه با ظرفیت Q از گره‌ای به نام انبار شروع به حرکت می‌کنند و بعد از سرویس‌دهی به مشتریان به آن باز می‌گردند به شرط آنکه هر کدام از مشتریان را فقط یک‌بار مورد ملاقات قرار دهند و در هیچ زمانی بیشتر از ظرفیت محدود Q بارگذاری نکنند. هدف کمینه‌کردن مسیرهای پیموده شده توسط وسایل نقلیه است. این مقاله کاربرد روش رقابت استعماری، را برای حل مساله مسیریابی وسیله نقلیه ارائه می‌کند. برخلاف روش‌های دیگر بهینه‌سازی، این روش از فرآیند اجتماعی-سیاسی جوامع الهام گرفته شده است و از رقابت بین کشورهای استعمارگر و مستعمره برای رسیدن به جواب استفاده می‌کند. برای آزمایش کارایی الگوریتم، دو دسته مثال استاندارد در نظر گرفته شده و الگوریتم بر روی آن مورد اجرا قرار گرفته است. نتایج محاسباتی روی این مثال‌ها که دارای اندازه‌ای از ۵۰ تا ۲۰۰ می‌باشند نشان می‌دهد که الگوریتم پیشنهادی توانسته رقابت خوبی با الگوریتم‌های مشهور فراابتکاری از نظر کیفیت جواب‌ها داشته باشد. به علاوه جواب‌های نزدیک به بهترین جواب‌های تاکنون بدست آمده برای بیشتر مثال‌ها بدست آورده شده است.

واژه‌های کلیدی: الگوریتم رقابت استعماری، مسائل NP-تام، مساله مسیریابی وسیله نقلیه.

*. Email: khoshbakht@aut.ac.ir

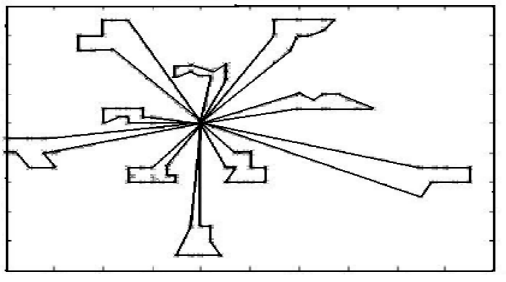
عهده‌دارمکاتبات

۱. مقدمه

مساله مسیریابی وسیله نقلیه[1] (VRP) یکی از مهم‌ترین مسائل تحقیق در عملیات و برنامه‌ریزی حمل و نقل است که امروزه بسیار مورد توجه محققان و دانشمندان قرار می‌گیرد. در این مساله چندین وسیله نقلیه بطور هم‌زمان از گره‌ای معین به نام انبار کالا شروع به حرکت می‌کنند و بعد از ملاقات کردن مشتری‌ها به همان گره باز می‌گردند، به شرط آنکه هر گره فقط توسط یکی از این وسایل نقلیه ملاقات شود و هر وسیله نقلیه بیشتر از ظرفیت خود در طول مسیر بارگذاری نکند. هدف تعیین کمینه هزینه جابجایی برای وسائل نقلیه بکار گرفته شده با توجه به محدودیت‌های مساله است. بطور مثال در شکل ۱ نمونه‌ای از حل یک مساله مسیریابی وسیله نقلیه با تعدادی گره و وسیله نقلیه نشان داده شده است. یکی از عواملی که سبب شده است این مساله یکی از مسائل مهم بهینه‌سازی ترکیباتی لقب گیرد، کاربرد این مساله و گسترش‌های بسیار زیاد آن همانند حالت کلاسیک [۱]، باز [۲]، دریافت و تحویل همزمان [۳-۴]، با ناوگان ناهمگن بار [۵] و غیره در دنیای واقعی است. بطور مثال فرض کنید که یک کارخانه قادر باشد، طول مسیر تحویل کالا به مشتریان و یا تعداد وسایل نقلیه و در نتیجه هزینه حمل و نقل را کاهش دهد. در نتیجه شرکت مربوطه قدرت رقابت خود را در مقابل شرکت‌های مشابه دیگر افزایش داده و بازار کالای خود را گسترش می‌دهد و در نهایت سود بیشتری را کسب خواهد کرد.

از زمانی که مساله مسیریابی وسیله نقلیه در دهه ۶۰ مورد بررسی قرار گرفت، حالت‌های بسیاری از آن بر اساس کاربردهای متفاوتی که در دنیای واقعی داشت، مشتق شدند. تنوع این‌گونه از مسائل آنقدر زیاد است که دسته‌بندی آنها و بیان حالت‌های مختلفی که در آن رخ می‌دهد، بسیار مشکل و زمان گیر است [۶]. بنابراین در اینجا فقط حالت ظرفیت‌دار آن در نظر گرفته می‌شود. این مسئله اولین بار به وسیله دانتزیک و همکاران معرفی شد. آن‌ها در آن مقطع فرمول‌بندی مسئله را مورد توجه قرار دادند و یک الگوریتم ریاضی برای آن در قالب مسئله مسیریابی کامیون‌های گازوئیل از پایانه سوخت‌رسانی به ایستگاه‌های متقاضی گازوئیل، مطرح کردند [۷].

به علاوه آن‌ها نشان دادند که به‌کارگیری روش‌های مدیریتی و مباحث بهینه‌سازی در بحث حمل‌ونقل تأثیر به سزایی در کاهش هزینه نهایی کالا دارد. در این مقاله توسط همین افراد روشی برای یافتن یک جواب نزدیک به بهینه ارائه شد. بعلاوه در این مسئله فرض شد که ظرفیت وسایل نقلیه به طور یکسان و برابر با بی‌نهایت است و مدت زمان توقف در گره‌ها برابر با صفر در نظر گرفته می‌شود، هدف تعیین مسیری است که هر خودرو باید طی کند تا به دسته‌ای از مشتریان در یک گستره جغرافیایی خدمتی را ارائه نماید. لذا تعیین مجموعه مشتریان اختصاص یافته به هر خودرو و هم تعیین مسیر رفت و برگشت و تقدم و تأخر مراجعه به آن‌ها دو سؤالی است که باید در این مسئله به آن پاسخ داده شود. به عبارت دیگر، علاوه بر دسته‌بندی مشتریان و تخصیص دادن هر مشتری به وسیله‌ای خاص، باید مسیریابی خودروها هم انجام شود. بنابراین این مسئله ابتدایی همان مسئله چندین فروشنده دوره‌گرد است که در آن محدودیتی برای ظرفیت وسایل نقلیه وجود ندارد [۸].



شکل ۱. حل نمونه‌ای از مساله مسیریابی وسیله نقلیه

1. Vehicle Routing Problem (VRP)

بطور کلی روش‌های زیادی برای حل کردن این مساله وجود دارند که می‌توان این روش‌ها را به دو دسته بزرگ الگوریتم‌های دقیق و تخمینی تقسیم‌بندی کرد. در الگوریتم‌های دقیق سرانجام می‌توان به جواب بهینه مساله دست یافت اما الگوریتم باید زمان بیشتری و در نتیجه هزینه بیشتری را برای رسیدن به این جواب صرف کند. برای مثال از این نوع الگوریتم‌ها می‌توان به پیمایش درخت [۹]، روش مینیمم k- درخت [۱۰] و شاخه و کران [۱۱] اشاره کرد.

در نقطه مقابل، چون مساله مسیریابی وسیله نقلیه NP-تام[1] [۱۲] است، بنابراین با افزایش اندازه مساله تعداد جواب‌های شدنی بطور ناگهانی و با شتاب بسیار زیاد شروع به رشد می‌کند و الگوریتم نمی‌تواند در یک زمان قابل قبول به جواب بهینه مساله دست یابد. بنابراین برای اینگونه مسائل الگوریتم‌های تخمینی، که خود به دو دسته الگوریتم‌های ابتکاری و فراابتکاری دسته‌بندی می‌شوند، مورد توجه بیشتری قرار گرفتند و نسخه‌های زیادی از این نوع الگوریتم‌ها پیشنهاد شدند.

الگوریتم‌های ابتکاری برخلاف روش‌های دقیق در یک زمان اندک به جواب می‌رسند هر چند که ممکن است جواب مربوطه متناسب به اندازه مساله دارای دقت کمتری نسبت به جواب الگوریتم‌های دقیق باشد. توجه بدین نکته ضروری است که الگوریتم‌های ابتکاری دارای راهکار موثری برای فرار از نقاط بهینه محلی نیستند و در بیشتر مواقع در بهینه‌های محلی گیر می‌افتند. برای نمونه می‌توان به الگوریتم صرفه‌جو و نوع‌های اصلاحی [۱۳] و همچنین الگوریتم بندرز [۱۴] آن اشاره کرد.

در روش‌های فراابتکاری که تقریبا ۳۰ سال است که مورد توجه قرار گرفته شده است، جواب‌ها تقریبا در یک زمان بیشتر از الگوریتم‌های ابتکاری و کمتر از روش‌های دقیق بدست می‌آیند زیرا میزان اجرای الگوریتم، برخلاف روش‌های ابتکاری، به تصمیم کاربر وابسته است، در مقایسه این روش‌ها با روش‌های ابتکاری، روش‌های فراابتکاری دارای پارامترهای زیادی هستند که باید توسط کاربر و بصورت تجربی بدست آیند. بنابراین الگوریتم‌ها نمی‌توانند در تکرارهای مشابه جواب‌های یکسانی را بدست آورند. به عبارت دیگر اینگونه از الگوریتم‌ها دارای یک روال ثابت برای رسیدن به جواب نبوده و پارامترهای تصادفی نقش زیادی را در این الگوریتم‌ها بازی می‌کنند. بطور مثال از این الگوریتم‌ها می‌توان به روش‌های مورچگان [۴]، جستجوی ممنوع [۳]، ممتیک [۱۵]، ترکیبی ژنتیک و ممتیک [۱۶]، پرندگان [۱۷]، رقابت استعماری [۱۸] و غیره اشاره کرد.

الگوریتم رقابت استعماری[2] (ICA) یکی از جدیدترین الگوریتم‌های فراابتکاری است که تاکنون بیشتر بر روی مسائل گوناگون بهینه‌سازی ترکیباتی پیوسته مورد استفاده قرار گرفته و نتایج بسیار خوبی را بدست آورده است. از طرف دیگر به علت اینکه مسئله *VRP* همان‌طور که گفته شد از اهمیت بسیار زیادی در مسائل تحقیق در عملیات است، هدف این مقاله آن است که به این مسئله پرداخته شود و برای اولین بار الگوریتم *ICA* برای آن پیشنهاد گردد. در روش پیشنهادی برای مرحله جذب، از الگوریتم نزدیک‌ترین همسایه اصلاحی استفاده شده است و در مرحله انقلاب روش بهبود دهنده دوگانه بکار گرفته می‌شود. این الگوریتم دارای کارایی بسیار خوبی در مقایسه با سایر روش‌های فراابتکاری است که در بخش‌های آینده به آن پرداخته می‌شود.

در این مقاله بخش ۲ فرمول‌بندی مساله را ارائه می‌کند و بخش ۳ به توضیح الگوریتم رقابت استعماری و روش پیشنهادی می‌پردازد. سپس نتایج محاسباتی الگوریتم که بر روی مثال‌های استاندارد مسائل حمل و نقل اجرا شده است در بخش ۴ مورد تجزیه و تحلیل قرار می‌گیرد و با الگوریتم‌های ابتکاری و فراابتکاری مانند صرفه‌جو، ژنتیک، مورچگان، جستجوی ممنوع و غیره مورد مقایسه قرار می‌گیرد. در انتها نتیجه‌گیری و جهت‌گیری‌های آینده در بخش ۵ ارائه می‌شود.

۲. فرمول‌بندی مساله

فرض کنید:

$V=\{0,1,...,n\}$ و $A=\{(i,j)|\,i,j\in V \text{ and } i\neq j\}$

به ترتیب نشان دهنده مجموعه گره و مجموعه یال‌های گراف $G(V,A)$ باشد. گره 0 نمایش‌دهنده انبار کالا و

1. NP-Complete
2. Imperialist Competitive Algorithm (ICA)

بقیه گره‌های نشان دهنده مشتری‌ها هستند که هر کدام دارای مقدار تقاضای مشخص می‌باشند. همچنین تقاضاهای هر یک از مشتری‌ها فقط باید به وسیله یکی از سرویس دهنده مسئله برآورده شود. باید توجه کرد که در این مسئله همان‌طور که گفته شد نکات قابل توجه دیگری مانند این‌که همه وسایل نقلیه در ابتدای الگوریتم باید در گره انبار باشند، هیچ سرویس دهنده‌ای در هیچ زمان مجاز نیست که بیشتر از ظرفیت معین شده‌ای بارگذاری کند و تقاضای هر مشتری نباید بیشتر از ظرفیت مجاز باشد، وجود دارد. اگر ماتریس هزینه برای رفتن هر یک از سرویس دهنده‌ها از گره‌ای به گره دیگر با ماتریس متقارن C نمایش داده شود، هدف مسئله عبارت است از یافتن مسیرهایی با کمترین هزینه برای سرویس‌دهنده‌ها به طوری که بعد از سرویس‌دهی به انبار باز گردند. برای ارائه مدل، معرفی چندین متغیر و پارامتر به عنوان ورودی لازم است که به شرح زیر می‌باشند.

n: تعداد مشتریان که معمولاً به شکل اندیس‌های ۱ تا n همراه سایر پارامترها دیده می‌شود.

m: تعداد وسایل نقلیه مورد استفاده در هر مسئله که در ابتدا همگی در انبار مستقر شده‌اند.

Q: ظرفیت هر وسیله نقلیه که برای همه وسایل نقلیه یکسان در نظر گرفته می‌شود.

q_i: مقدار تقاضای مشتری iام است که فرض می‌شود برای هر i از مقدار Q بیشتر نباشد.

c_{ij}: هزینه سفر بین دو گره i و j است.

x_{ij}^k: اگر وسیله نقلیه k بعد از گره i گره j را ملاقات کند آن‌گاه برای $(\forall i,j=0,1,2,\ldots,n,\forall k=1,\ldots,m)$ مقدار $x_{ij}^k=1$ است و در غیر این صورت $x_{ij}^k=0$ می‌باشد.

بنابراین مدل مسئله VRP عبارت است از:

$$\min \sum_{k=1}^{m}\sum_{i=0}^{n}\sum_{j=0}^{n} c_{ij}x_{ij}^k \qquad (۱)$$

$$\sum_{k=1}^{m}\sum_{i=0}^{n} x_{ij}^k = 1 \qquad \forall j=1,\ldots,n \qquad (۲)$$

$$\sum_{i=0}^{n} x_{iu}^k - \sum_{j=0}^{n} x_{uj}^k = 0, \qquad (۴)$$

$$\forall u=1,\ldots,n,\forall k=1,\ldots,m$$

$$\sum_{k=1}^{m}\sum_{i=0}^{n} y_{ij}^k - \sum_{k=1}^{m}\sum_{i=0}^{n} y_{ji}^k = q_j, \quad \forall j=1,2,\ldots,n \qquad (۴)$$

$$q_j x_{ij}^k \le y_{ij}^k \le (Q-q_i)x_{ij}^k \quad \forall i,j=0,1,\ldots,n, i\ne j, \forall k=1,2,\ldots,m \qquad (۵)$$

$$\sum_{j=1}^{n} x_{0j} = m \qquad (۶)$$

$$y_{ij}^k \ge 0, x_{ij}^k \in \{0,1\} \quad \forall i,j=0,1,\ldots,n, \forall k=1,2,\ldots,m \qquad (۷)$$

در این مدل (۱) نشان دهنده تابع هدف مسئله است که باید تا حد امکان مینیمم شود در حالی که (۲) نشان‌دهنده آن هستند که به هر گره بجز انبار فقط یک وسیله نقلیه وارد می‌شود در حالی که رابطه (۳) بیان می‌کند که هر وسیله نقلیه‌ای که به گره i وارد می‌شود، باید از آن خارج شود. در واقع این قید پیوستگی مسیرها را تضمین می‌کند. رابطه (۴) نشان می‌دهد که تقاضای هر مشتری حتماً به‌طور کامل برآورده می‌شود؛ رابطه (۵) نشان می‌دهد که محدودیت ظرفیت هر وسیله نقلیه رعایت می‌شود و رابطه (۶) به این نکته اشاره دارد که تعداد یال‌های خروجی از انبار برابر تعداد وسایل نقلیه (m) است و در نهایت روابط (۷) دامنه متغیرها را تعیین می‌کنند.

۳. الگوریتم ارائه شده

در این بخش به ترتیب الگوریتم رقابت استعماری و الگوریتم پیشنهادی مطرح می‌شوند. بدین منظور در زیر بخش ۱ ابتدا مقدمه‌ای بر الگوریتم رقابت استعماری ارائه می‌گردد. سپس در زیر بخش ۲ الگوریتم پیشنهادی شرح داده می‌شود و با جزئیات بیشتر مورد بررسی قرار می‌گیرد.

۱،۳. الگوریتم رقابت استعماری

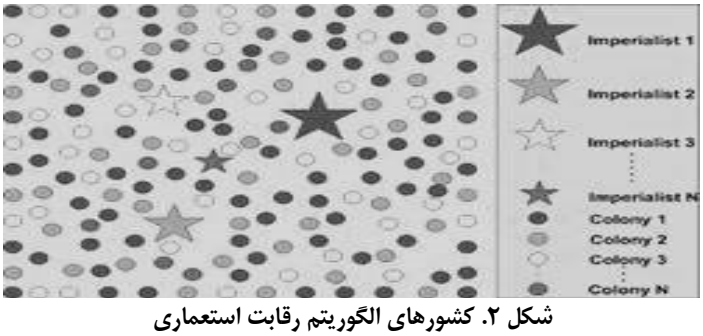
الگوریتم رقابت استعماری یکی از روش‌های فراابتکاری است که توسط آتش‌پز و همکارانش در سال ۲۰۰۸ [۱۹] ابداع شد. اگر چه این الگوریتم تنها هفت سال است که از ابداع آن می‌گذرد اما تاکنون در مسائل زیادی مانند

طراحی ساختار اسکلتی [۲۰]، خوشه‌بندی داده ها [۲۱]، موتور القایی خطی [۲۲]، تعادل نش [۲۳] و غیره استفاده شده است. دلایل استقبال بالا از این الگوریتم در کنار کارایی بالای آن، سرعت همگرایی، توانایی بهینه سازی بالا در مقایسه با الگوریتم‌های موجود، بیشتر به جنبه نوآوری و جذاب بودن آن برای متخصصین حوزه بهینه سازی است. این روش شبیه روش‌های پرندگان و ژنتیک برای رسیدن به جواب از سیستم‌های تکاملی استفاده می‌کند و از نظر قابلیت تعمیم پذیری در مسائل کاربردی بسیار موفق بوده است. در این مسائل با توجه به اینکه الگوریتم هیچ اطلاعاتی در مورد مساله مربوط ندارد، می‌تواند با ایجاد جواب‌های تصادفی در مسیر یافتن جواب‌های بهتر حرکت کند و در فضای جستجو تا حد امکان به خوبی پیشروی نماید.

الگوریتم کلی روش رقابت استعماری به این روال است که همانند روش ژنتیک، با تعدادی جمعیت اولیه شروع می‌شود که هر کدام یک کشور نامیده می‌شود. حال طبق مساله مورد بررسی و تابع هدف آن، مقدار تابع هزینه برای هر کدام از کشورها بدست می‌آید و تعدادی از بهترین کشورها با مقدار تابع هدف بهتر انتخاب می‌گردد. حال این کشورهای انتخاب شده کشورهای استعمارگر و بقیه کشورها، کشورهای مستعمره نامیده می‌شوند. باید توجه کرد که که در شرایط مساوی که چندین کشور استعمارگر با مقدار تابع هدف یکسان وجود دارد به تصادف از همین کشورها، کشورهای استعمارگر انتخاب می‌شوند. حال مانند حالت واقعی، هر چه قدرت یک کشور استعمارگر بیشتر باشد، تعداد کشورهای مستعمره شده توسط آن کشور بیشتر می‌شود. به عبارت دیگر بسته به قدرت هر کشور، تعدادی از کشورهای مستعمره به آن کشور اختصاص می‌یابد.

بطور مثال در شکل ۲ نمونه‌ای از این تقسیم‌بندی نشان داده شده است [۲۱]. در این شکل ستاره‌های $1,2,...,N$ کشورهای استعمارگر را نشان می‌دهد. در حالیکه دایره‌ها نشان دهنده کشورهای مستعمره شده توسط هر کدام از کشورهای استعمارگر است. همانطور که از شکل مشخص است کشور استعمارگر قویتر که در اینجا ستاره بزرگ‌تر است دارای تعداد بسیار زیادی کشور تحت سلطه است که با دایره‌های سیاه نشان داده شده است و کشور استعمارگر از همه ضعیف‌تر که با ستاره کوچکتر نمایان است دارای کمترین کشورهای تحت نفوذ می‌باشد.

به علاوه کشورهای استعمارگر سعی در وابسته کردن کشور مستعمره با دگرگونی فرهنگ و رسوم آن کشور می‌کنند در نتیجه این عمل سبب می‌گردد که کشور مستعمره در جهت همسان شدن فرهنگی با کشور استعمارگر حرکت کند. باید توجه کرد که در ارائه الگوریتم، این سیاست با استفاده از تابع جذب صورت می‌گیرد. به عبارت دیگر هر کدام از کشورهای مستعمره شده توسط یک استعمارگر، به ترتیب با استفاده از یک روال معین به کشور استعمارگر نزدیک می‌شوند. برای اینکه از همگرایی زودرس الگوریتم جلوگیری شود و در حین تکرار الگوریتم تنوع جواب‌ها حفظ شود، ρ درصد از کشورهای مستعمره انتخاب و دچار انقلاب می‌شوند. این عمل سبب می‌گردد که این تعداد از جواب‌ها دارای مشخصات متنوع‌تری نسبت به حالت عادی شوند و انتخاب صحیح آنها برای هر مساله سبب افزایش کیفیت و سرعت الگوریتم شود. باید توجه کرد که در این روش استراتژی‌های تابع جذب و انقلاب از اهمیت زیادی برخوردار بوده و نوع هر کدام از آنها در سرعت رسیدن به جواب موردنظر از اهمیت زیادی برخوردار است.



شکل ۲. کشورهای الگوریتم رقابت استعماری

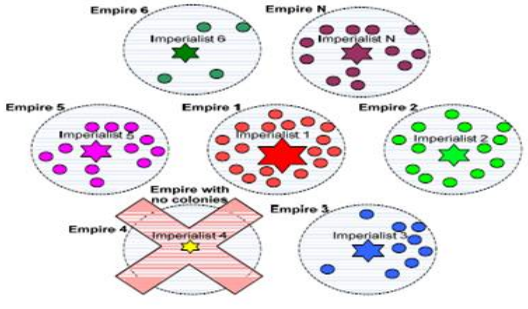
بعد از اینکه توابع جذب و انقلاب برای کشورهای مستعمره نسبت به کشورهای استعمارگر خودشان انجام شد، تمامی توابع هدف برای هر کدام از این کشورها دوباره محاسبه می‌شود. حال برای هر کدام از امپراتورها (کشور استعمارگر و کشورهای مستعمره آن)، اگر بهترین تابع هدف کشورهای مستعمره از تابع هدف کشور استعمارگر مقدار بهتری داشته باشد، آنگاه جای آن دو کشور با هم عوض می‌شود. در ضمن، بعد از انجام این عمل برای تمامی امپراتوری‌ها بهترین جواب و مقدار همه کشورهای استعمارگر به عنوان بهترین جواب جاری در این تکرار الگوریتم ذخیره می‌شود. حال اگر این مقدار جواب جاری از بهترین مقدار تاکنون بدست آمده دارای کیفیت بیشتری باشد آنگاه جواب و مقدار جدید جایگزین جواب قبلی می‌شود. از طرف دیگر باید توجه کرد که بعد از پایان الگوریتم، این جواب و مقدار به عنوان نتیجه نهایی الگوریتم ارائه می‌شود.

در مرحله بعدی قدرت کل یک امپراتوری به صورت مجموع تابع هدف کشور استعمارگر به اضافه ضریبی مانند β از میانگین توابع هدف مستعمرات آن تعریف می‌شود. توجه به این نکته ضروری است که این ضریب سبب می‌گردد که کاربر بتواند تاثیر اهمیت کشور استعمارگر یا میانگین کشورهای مستعمره را در قدرت نهایی هر امپراتور معین کند. همچنین طی رقابت استعماری که براساس مقایسه قدرت امپراتوری‌ها صورت می‌گیرد، امپراتوری‌های ضعیف که دارای قدرت کمتری نسبت به سایر امپراتوری‌ها است همانند شکل ۳ به تدریج قدرت خود را از دست داده و به مرور زمان از بین می‌رود [۲۴].

به عبارت دیگر در هر مرحله از الگوریتم ضعیف‌ترین مستعمره امپراتوری ضعیف انتخاب شده و به قوی‌ترین امپراتوری انتقال می‌یابد. حال اگر فقط یک امپراتوری وجود نداشته باشد، دوباره الگوریتم به مرحله تابع جذب و انقلاب برمی‌گردد. بنابراین رقابت استعماری باعث می‌شود که به مرور زمان، حالتی بدست آید که در آن تنها یک امپراتوری وجود دارد که دنیا را اداره می‌کند و این حالت وقتی اتفاق می‌افتد که شرط پایانی الگوریتم بوجود آید.

۲،۳. روش پیشنهادی

الگوریتم رقابت استعماری یکی از جدیدترین الگوریتم‌ها در بین الگوریتم‌های فراابتکاری است که با این روش می‌توان حرکتی بهتر در فضای شدنی مسائل بهینه‌سازی ترکیباتی داشت و فضای جواب را با دقت بیشتری مورد بررسی قرار داد و در نتیجه جواب‌های باکیفیتی را بدست آورد. در این الگوریتم روش نشان دادن هر کشور برای مساله مسیریابی وسیله نقلیه بدین شکل است که از یک آرایه دوبخشی که در شکل ۴ نشان داده شده است، استفاده می‌شود. در قسمت اول، مشتری‌هایی که باید بوسیله هر وسیله نقلیه به ترتیب مورد ملاقات قرار گیرند نشان داده شده است در حالیکه در قسمت دوم، تعداد مشتری‌هایی که هر یک از وسائل نقلیه ملاقات می‌کنند، ارائه شده است. بطور مثال در شکل ۴ نمونه‌ای از این مسئله نشان داده شده است که در آن پنج مشتری بوسیله وسیله نقلیه اول، شش مشتری توسط وسیله نقلیه دوم و سه مشتری بوسیله وسیله نقلیه سوم مورد ملاقات قرار می‌گیرند.



شکل ۳. حذف امپراتور ضعیف

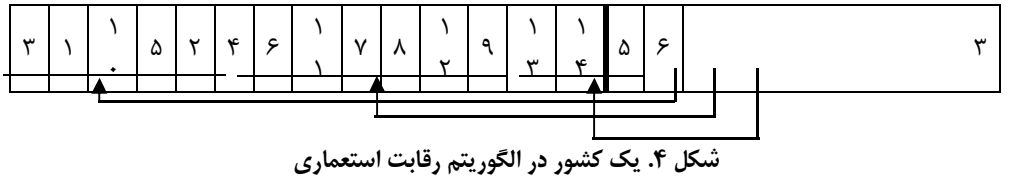
حال به تعداد n کشور، جواب شدنی برای مساله مسیریابی وسیله نقلیه به تصادف انتخاب شده و مقادیر تابع هدف f_i برای هر $i=1,...,n$ بدست می‌آید. سپس با مقایسه مقادیر تابع هدف بدست آمده برای هر کدام از کشورها، m کشور که دارای مقادیر تابع هدف بهتر هستند به عنوان کشورهای استعمارگر در نظر گرفته می‌شوند و با جابجایی کشورها، اندیس‌های ۱ تا m جمعیت اولیه به کشورهای استعمارگر تعلق می‌یابد. حال باید کشورهای مستعمره به کشورهای استعمارگر اختصاص یابد تا بدین طریق به تعداد m امپراتوری ایجاد شود. برای این منظور از رابطه (۸) استفاده می‌شود. این رابطه سبب می‌شود که هر کدام از کشورهای استعمارگر که دارای قدرت بیشتری هستند، تابع هدف کمتری داشته باشند، دارای تعداد بیشتری از کشورهای مستعمره باشند و امپراتوری بزرگ‌تری را تشکیل دهند. باید توجه کرد که در این رابطه از تابع جزء صحیح (int) استفاده می‌شود زیرا این عمل سبب می‌گردد که به هر کدام از امپراتوری‌ها تعداد صحیحی از کشورهای مستعمره اختصاص یابد. از طرف دیگر امکان دارد که در انتها کشورهای وجود داشته باشند که به هیچ کدام از امپراتوری‌ها تعلق نداشته باشند که در این صورت این کشورها به قوی‌ترین امپراتوری اختصاص می‌یابند.

$$k_j = \text{int}\left(\frac{1/f_j}{\sum_{i=1}^{n} 1/f_i}.(n-m)\right) \quad j=1,...,m \qquad (۸)$$

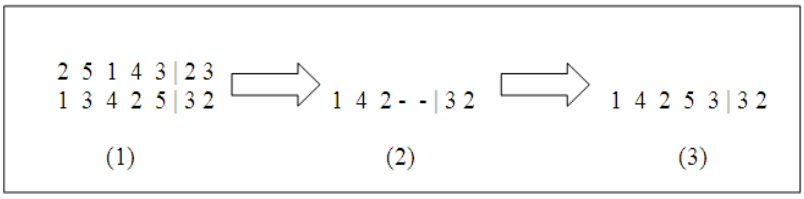
بعد از اینکه امپراتوری‌ها تشکیل شدند طبق حالت طبیعی باید کشورهای مستعمره از لحاظ فرهنگی و اجتماعی شبیه کشورهای استعمارگر شوند. به عبارت دیگر در الگوریتم، کشورهای ضعیف هر امپراتوری با استفاده از کشورهای استعمارگر که نقش بهینه‌های محلی را بر عهده دارند، باید کیفیت خود را افزایش دهند. بنابراین در اینجا باید از راهکاری استفاده کرد که تا حدی کشورهای مستعمره در جهت کشورهای استعمارگر حرکت کنند. بدین منظور از روش نزدیکترین همسایه اصلاحی استفاده می‌شود. بطور مثال در شکل ۵ جزئیات کامل این روش نشان داده شده است. در این شکل کشور [2 5 1 4 3 | 2 3] نشان‌دهنده کشور استعمارگر و کشور [1 3 4 2 5 | 3 2] نشان دهنده کشور مستعمره است. حال از ۱ که اولین گره کشور مستعمره است شروع به حرکت کرده و در دو کشور، همسایه‌های ۱ تاکنون ملاقات نشده که گره‌های ۳، ۴ و ۵ هستند و بطور مثال دارای فاصله‌های ۱، ۲ و ۲ می‌باشند در نظر گرفته می‌شوند. طبق روش نزدیک‌ترین همسایه اصلاحی احتمال ملاقات شدن گره‌های ۳، ۴ و ۵ به ترتیب طبق فرمول (۹) زیر بدست می‌آید. باید توجه کرد که اگر همسایه‌های مربوطه قبلا مورد ملاقات قرار گرفته باشند آن‌گاه بقیه گره‌های ملاقات نشده در مجموعه همسایگی در نظر گرفته می‌شوند.

$$\frac{1/c_{1j}}{\sum_{j\in\{3,4,5\}} 1/c_{1j}} \qquad (۹)$$

که در آن c_{1j} فاصله اقلیدسی گره ۱ با سه گره مربوطه است که در اینجا ۱، ۲ و ۲ در نظر گرفته شده است. فرض کنید ۴ برگزیده شود پس در جمعیت اولیه تاکنون [3 2 | - - - 4 1] بوجود می‌آید. بعلاوه برای ۴، سه گره ۱، ۲ و ۳ به عنوان همسایه در دو کشور محسوب می‌شوند اما چون قبلا ۱ مورد ملاقات قرار گرفته شده است. از بین ۲ و ۳ نزدیکترین همسایه طبق (۲) انتخاب می‌گردد. این عمل تا یافتن بقیه گره‌ها در [3 2 | - - - 4 1] ادامه می‌یابد. بنابراین این تابع جذب برای همه کشورهای مستعمره نسبت به کشورهای استعمارگر انجام و جواب و مقدار بدست آمده فقط در صورتی جایگزین جواب و مقدار کشور مستعمره می‌شود که در محدودیت‌های مساله صدق کند.



شکل ۴. یک کشور در الگوریتم رقابت استعماری



شکل ۵. تابع جذب

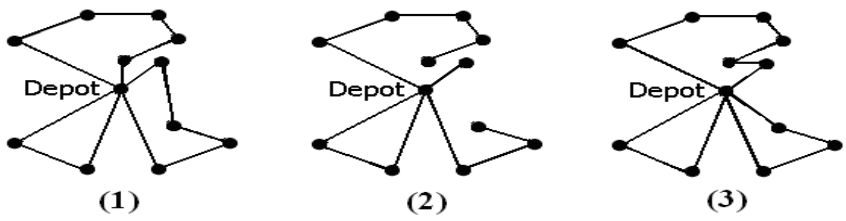
در مرحله بعدی ρ درصد از کشورهای مستعمره دچار انقلاب می‌شوند. این عمل سبب می‌گردد که کشورهای مستعمره در امپراتوری‌ها دارای تنوع کافی باشند و در هر مرحله کیفیت آنها افزایش یابد. به همین علت در هر امپراتوری ρ درصد از مستعمرات انتخاب شده و الگوریتم جستجوی بهبود دهنده دوگانه برای آنها اجرا می‌گردد. این روش، همانطور که در شکل ۶ نشان داده شده است، براساس حذف‌کردن دو یال غیر همسایه از تور و دوباره متصل‌کردن آن دو یال از طریقی دیگر، کار می‌کند. باید توجه کرد که برای متصل‌کردن مسیر و ایجادکردن دوباره تور چندین راه وجود دارد اما فقط حالتی پذیرفته می‌شود که در محدودیت‌های مساله صدق کند و تور جدید مقدار بهتری برای مساله نسبت به جواب قبلی بدست آورد. توجه به این نکته ضروری است که عمل حذف دو کمان و متصل‌کردن دوباره آنها بطور متوالی تا جایی ادامه پیدا می‌کند که دیگر هیچ حرکت بهبوددهنده دوگانه برای الگوریتم یافت نشود.

برای تمامی کشورهای مستعمره شده بدست آمد، امکان دارد که این کشورها دارای تابع هدف بهتری نسبت به کشورهای استعماگر خود شوند. بنابراین بهترین کشور مستعمره در هر امپراتوری انتخاب شده و در صورت داشتن تابع هدف بهتر، جایگزین کشور استعمارگر می‌شود. باید توجه داشت که در صورتیکه چند کشور مستعمره دارای تابع هدف یکسان باشند، آنگاه یکی از آنها به تصادف انتخاب شده و با کشور استعمارگر امپراتوری خود مقایسه می‌شود.

بعلاوه دو متغیر در اجرای الگوریتم وجود دارند که مقادیر بهترین جواب و مقدار تابع هدف را تا تکرار جاری ذخیره می‌کنند. هنگامیکه در هر تکرار عمل جایگزینی کشورهای استعمارگر انجام شد، آنگاه بهترین مقدار و جواب کشورهای استعمارگر برای مساله مسیریابی وسیله نقلیه بعنوان بهترین جواب جاری انتخاب می‌گردد. حال در صورتی که این مقدار بدست آمده دارای مقداری بهتر نسبت به جواب‌های بدست آمده در تکرارهای قبلی باشند، آنگاه جواب و مقدار جدید جایگزین جواب و مقدار قبلی می‌گردد.

از طرف دیگر در این مرحله باید قدرت امپراتوری‌ها سنجیده شود تا اینکه امپراتوری قوی‌تر بتواند قلمرو خود را بوسیله گرفتن کشورهای دیگر افزایش دهد. برای این منظور ابتدا باید قدرت هر امپراتوری بدست آورده شود. این قدرت علاوه بر اینکه باید نشان دهنده قدرت کشور استعمارگر باشند، باید قدرت کشورهای مستعمره آن امپراتوری را نیز در نظر بگیرد. بدین منظور از فرمول زیر استفاده می‌شود:

$$p_j = f_j + \lambda(s_j) \qquad j = 1,\ldots,m \quad (۱۰)$$



شکل ۶. حرکت بهبوددهنده دوگانه برای حل مساله *VRP* بعد از اینکه مقدار تابع هدف

در این فرمول s_j, p_j و λ به ترتیب نشان دهنده قدرت کل هر امپراتوری، میانگین قدرت کشورهای مستعمره در هر امپراتوری و در نهایت ضریب تاثیر، بین صفر و یک، میانگین قدرت کشورهای مستعمره در قدرت یک امپراتوری است.

حال در این هنگام امپراتوری ضعیف‌تر قدرت خود با از دست دادن ضعیف‌ترین مستعمره خود به قوی‌ترین امپراتوری از دست می‌دهد. سپس در این مرحله این شرط مورد بررسی قرار می‌گیرد که آیا امپراتوری وجود دارد که دیگر هیچ مستعمره‌ای برایش باقی مانده است یا نه؟ اگر در این مرحله امپراتوری یافت شد که حذف می‌شود و در غیر این صورت الگوریتم با برگشت به مرحله تابع جذب دوباره تکرار می‌گردد تا اینکه شرط پایانی حلقه برقرار گردد.

برای شرط پایانی حلقه از دو شرط تکرار الگوریتم به تعداد معین t بار یا باقی ماندن فقط یک امپراتوری استفاده می‌شود که بطور همزمان در پایان هر تکرار الگوریتم مورد بررسی قرار می‌گیرد. حال هر کدام از این دو شرط که زودتر اتفاق بیفتد الگوریتم به پایان می‌رسد و بهترین جواب و مقداری که تاکنون بدست آمده است به عنوان بهترین جواب و مقدار معرفی می‌گردد.

۴. محاسبات عددی

تمام کدهای برنامه الگوریتم رقابت استعماری برای مساله مسیریابی وسیله نقلیه به زبان متلب ۷ نوشته شده است و کامپیوتری که این برنامه‌ها بر روی آن اجرا شده دارای قدرت پردازش *AMD 3.5 GHz* با یک گیگا بایت حافظه است. نتایج محاسباتی از مقایسه این الگوریتم با سایر الگوریتم‌های ابتکاری و فراابتکاری در این بخش برای دو دسته از مسائل نشان داده است. در دسته اول و در جدول ۱ روش‌های ابتکاری شامل الگوریتم‌های صرفه‌جو و نوع اصلاحی آن ارائه شده است که بهترین الگوریتم‌ها در این دسته محسوب می‌شوند و الگوریتم‌های فراابتکاری شامل سه الگوریتم معروف ژنتیک، جستجوی ممنوع و مورچگان هستند. از طرف دیگر بهترین جواب‌هایی که تاکنون بدست آمده است هم ارائه شده تا بتوان الگوریتم‌ها را با جزئیات بیشتری مورد تجزیه و تحلیل قرار داد.

جدول ۱ نشان دهنده ۷ مساله استاندارد مساله مسیریابی وسیله نقلیه به همراه خصوصیات آنها است که می‌توان آنها را از آدرس زیر دانلود کرد:

http://branchandcut.org/VRP/data/

این جدول که از پنج ستون تشکیل شده است در ستون اول مشخصات نمونه‌های مورد آزمایش را نشان می‌دهد که شامل مثال‌های *E-n51-k5, E-n76-k10, E-n101-k8, E-n101-k10, E-n121-k7, M-n151-k12, M-n200-k17* است. باید توجه کرد که این دسته از مثال‌ها دارای یک دامنه خوب از تعداد مشتری‌ها مابین ۵۰ تا ۱۹۹ و تعداد وسائل نقلیه مابین ۵ تا ۱۷ عدد است که تاکنون الگوریتم‌های متنوعی بر روی آنها اجرا شده است. توجه به این نکته ضروری است که گره انبار جزء تعداد مشتری‌ها قرار نمی‌گیرد.

در ستون دوم و سوم جدول ۱ مشخصات این مثال‌ها آورده شده است بطوریکه در ستون دوم تعداد مشتری‌های هر مساله و در ستون سوم تعداد وسائل نقلیه نشان داده شده است. بعلاوه ظرفیت وسائل مورد استفاده در هر کدام از مثال‌ها در ستون چهارم ارائه شده است. در اینجا وسائل نقلیه همگن بوده و همگی دارای ظرفیت یکسان می‌باشند. همچنین الگوریتم پیشنهادی برای هر مساله ۱۰ بار بطور مستقل مورد آزمایش قرار گرفته است. بعلاوه در این مثال‌ها محدودیت زمان سرویس‌دهی به مشتریان و همچنین محدودیت حرکت هر کامیون به مقداری خاص وجود ندارد. بنابراین هر وسیله نقلیه دارای ظرفیت کافی، در هر زمانی می‌تواند هر گره دلخواه را مورد ملاقات قرار دهد.

نتایج کامل الگوریتم پیشنهادی بر روی مثال‌های استاندارد در جدول ۲ نشان داده شده است. در این جدول بدترین جواب الگوریتم (*Worst solution*)، میانگین جواب‌ها (*Mean solutions*) و بهترین جواب‌ها (*Best solution*) در ۱۰ بار آزمایش هر مثال نشان داده است. همچنین با وجود بهترین جواب‌های تاکنون بدست آمده در ستون پنجم جدول می‌توان کیفیت جواب‌های الگوریتم را مشاهده کرد.

در جدول ۳ مقایسه جواب‌های الگوریتم با دیگر روش‌های ابتکاری و فراابتکاری نشان داده شده است.

بعلاوه میانگین جواب‌های بدست آمده هر الگوریتم برای هفت مثال مربوطه در سطر آخر جدول ارائه شده است. نتایج به این نکته اشاره دارد که روش پیشنهادی اگر چه نتوانسته است که در ۶ مثال از هفت مثال مربوطه به جواب بهین هر مساله دست یابد و فقط در مثال *E-n51-k5* به جواب بهین دست یافته است، اما در حالت کلی توانسته است که رقابت خوبی را با سایر الگوریتم‌ها داشته باشد.

بعلاوه الگوریتم پیشنهادی در مقایسه با الگوریتم‌های ابتکاری مانند صرفه‌جو و ترکیب همین الگوریتم با روش بهبود دهنده سه‌گانه توانسته است که در پنج مثال از هفت مثال مربوطه جواب‌های بهتری را نسبت به این دو الگوریتم بدست آورد. اما فقط در دو مثال *E-n101-k10* و *E-n200-k17* است که دو الگوریتم مربوطه جواب‌های بهتری را بدست آورده‌اند. همچنین الگوریتم رقابت استعماری پیشنهادی با بدست آوردن میانگین ۹۵۹،۴۰ در مقابل میانگین‌های ۹۷۲،۲۱ و ۹۶۲،۰۳ که به ترتیب برای الگوریتم صرفه جو و صرفه جو ترکیبی با بهبود دهنده سه‌گانه بدست آمده است، توانسته از این نظر نیز جواب‌های باکیفیت‌تری بدست آورد.

جدول۱: مشخصات مسائل استاندارد *VRP*

مثال	تعداد مشتری	تعداد وسیله نقلیه	ظرفیت وسیله نقلیه	تعداد تکرار الگوریتم
E-n51-k5	۵۰	۵	۱۶۰	۱۰
E-n76-k10	۷۵	۱۰	۲۰۰	۱۰
E-n101-k8	۱۰۰	۸	۲۰۰	۱۰
E-n101-k10	۱۰۰	۱۰	۲۰۰	۱۰
E-n121-k7	۱۲۰	۷	۲۰۰	۱۰
M-n151-k12	۱۵۰	۱۲	۲۰۰	۱۰
M-n200-k17	۱۹۹	۱۷	۲۰۰	۱۰

جدول۲: بررسی جواب‌های الگوریتم پیشنهادی برای مسائل استاندارد*VRP*

مثال	*Worst solution*	*Mean solutions*	*Best solution*	*Best*
E-n51-k5	۵۳۶،۹۳	۵۲۹،۹۱	۵۲۱	۵۲۱
E-n76-k10	۸۶۱،۸۴	۸۵۳،۴۳	۸۴۲،۰۴	۸۳۲
E-n101-k8	۸۸۲،۹۳	۸۶۷،۵۴	۸۴۵،۱۳	۸۱۵
E-n101-k10	۸۹۳،۷۶	۸۶۱،۱۴	۸۴۰،۵۲	۸۲۰
E-n121-k7	۱۰۹۹،۵۴	۱۰۶۵،۳۲	۱۰۴۷،۱۹	۱۰۴۲،۱۱
M-n151-k12	۱۲۵۲،۱۶	۱۱۴۸،۷۵	۱۱۲۷،۶۲	۱۰۲۸،۴۲
M-n200-k17	۱۵۸۷،۸۲	۱۵۱۲،۴۰	۱۴۹۲،۳۲	۱۲۹۱،۴۵

جدول۳: مقایسه الگوریتم‌ها برای مسائل استاندارد*VRP*

مثال	الگوریتم صرفه‌جو+ 3-opt [۲۵]	الگوریتم صرفه‌جو [۲۶]	الگوریتم ترکیبی ژنتیک [۲۷]	روش الگوریتم مورچگان [۲۸]	روش جستجوی ممنوع [۲۹]	روش پیشنهادی	بهترین جواب
E-n51-k5	۵۷۸،۵۶	۵۸۴،۶۴	۵۲۴،۸۱	۵۲۱	۵۲۴،۶۱	۵۲۱	۵۲۱
E-n76-k10	۸۸۸،۰۴	۹۰۰،۲۶	۸۴۹،۷۷	۸۷۷	۸۴۴	۸۴۲،۰۴	۸۳۲
E-n101-k8	۸۷۸،۷۰	۸۸۶،۸۳	۸۴۰،۷۲	۸۴۵	۸۳۵	۸۴۵،۱۳	۸۱۵
E-n101-k10	۸۲۴،۴۲	۸۳۳،۵۱	۸۷۷،۸	۸۳۸	۸۲۰	۸۴۰،۵۲	۸۲۰
E-n121-k7	۱۰۴۹،۴۳	۱۰۷۱،۰۷	۱۰۶۰،۲۴	۱۱۸۹	۱۰۴۲،۱۱	۱۰۴۷،۱۹	۱۰۴۲،۱۱
M-n151-k12	۱۱۲۸،۲۴	۱۱۳۳،۴۳	۱۱۹۳،۰۵	۱۱۰۵	۱۰۵۲	۱۱۲۷،۶۲	۱۰۲۸،۴۲
M-n200-k17	۱۳۸۶،۸۴	۱۳۹۵،۷۴	۱۴۸۳،۰۶	۱۶۰۶	۱۳۵۴	۱۴۹۲،۳۲	۱۲۹۱،۴۵
میانگین	۹۶۲،۰۳	۹۷۲،۲۱	۹۷۵،۶۴	۹۹۷،۲۹	۹۲۴،۵۳	۹۵۹،۴۰	۹۰۷،۱۴

از طرف دیگر این روش توانسته است که رقابت خوبی با روش ترکیبی ژنتیک داشته باشد. با مقایسه این دو روش می‌توان نتیجه گرفت که الگوریتم پیشنهادی دارای کارایی بالاتری نسبت به این الگوریتم است زیرا فقط در دو مثال *E-n101-k8* و *E-n200-k17* نتوانسته است که جواب بهتری را تولید کند اما در ۵ مثال دیگر توانسته است که کیفیت جواب‌ها را به مقدار قابل قبولی افزایش دهد. باید اضافه کرد که از منظر میانگین جواب‌ها نیز مقدار بدست آمده برای الگوریتم پیشنهادی دارای مقدار بهتری نسبت به میانگین ۹۷۵،۶۴ است که توسط الگوریتم ژنتیک ترکیبی بدست آمده است.

همچنین الگوریتم دیگر فراابتکاری که در اینجا نتایج آن آورده شده است الگوریتم مورچگان است که علارغم اینکه در مسائل با اندازه کوچک نتایج قابل قبولی را بدست آورده است اما در مسائل بزرگ‌تر متاسفانه کارایی خود را از دست داده و به نتایج بسیار بی‌کیفیتی قناعت کرده است و در نهایت از نظر بدست آوردن میانگین بی‌کیفیت‌ترین الگوریتم در جدول ۳ است. با مقایسه الگوریتم پیشنهادی نسبت به این الگوریتم می‌توان دریافت که اگر چه الگوریتم مورچگان با بدست آوردن میانگین ۹۹۷،۲۹ فاصله زیادی را با الگوریتم پیشنهادی دارد و از کیفیت قابل قبولی برخوردار نیست اما در مقایسه جداگانه در مثال‌ها می‌توان دید که الگوریتم دارای ۲ مقدار بهتر، ۲ مقدار برابر و ۳ مقدار بدتر نسبت به الگوریتم پیشنهادی است و از این دید دارای رقابت نزدیکی با الگوریتم پیشنهادی به عنوان یکی از جدیدترین الگوریتم‌های فراابتکاری می‌باشد.

روش جستجوی ممنوع یکی از قویترین الگوریتم‌های فراابتکاری محسوب می‌شود که در این جدول توانسته است کیفیت قابل قبولی را ارائه کند و در دو مثال *E-n101-k10* و *E-n121-k7* به بهترین جواب‌های تاکنون بدست آمده، دست یابد. از طرف دیگر با مقایسه الگوریتم پیشنهادی و الگوریتم جستجوی ممنوع می‌توان نتیجه گرفت که الگوریتم پیشنهادی دارای کیفیت بهتری در مسائل کوچک همچون ۵۰ یا ۷۵ گره نسبت به این الگوریتم است اما هنگامیکه اندازه مساله افزایش می‌یابد الگوریتم نمی‌تواند کارایی خود را حفظ کند و به جواب‌های بی‌کیفیتی‌تری دست می‌یابد. بدین ترتیب در ۵ مثال دیگر بالای ۱۰۰ گره شامل *E-n101-k8*, *E-n101-k10*, *E-n121-k7*, *M-n151-k12*, *M-n200-k17* الگوریتم جستجوی ممنوع جواب‌های بهتری را نسبت به الگوریتم رقابت استعماری بدست آورد. همچنین در مقایسه میانگین جواب‌ها نیز الگوریتم پیشنهادی فاصله قابل ملاحظه‌ای را با الگوریتم جستجوی ممنوع با میانگین ۹۲۴،۵۳ دارد.

در جدول ۴، دسته دوم شامل ۱۴ مثال استاندارد

کریستوفیدز در نظر گرفته شده است که دارای یک بازه مناسب از گره‌ها بوده است و الگوریتم‌های زیادی بر روی این مثال‌ها مورد آزمایش قرار گرفته‌اند و بدین جهت می‌توان مقایسه مناسبی بین الگوریتم پیشنهادی و الگوریتم‌های دیگر انجام داد. در این ۱۴ مثال، ۱۰ مثال ابتدایی $C1$-$C10$ شامل مشتری‌هایی هستند که به طور تصادفی در اطراف انبار توزیع شده‌اند، در حالِی که در ۴ مثال باقیمانده مشتری‌ها در کلاس‌ترهایِی که در آن انبار کالا در مرکز واقع نشده است، قرار گرفته شده‌اند. از طرف دیگر همه مثال‌های در نظر گرفته شده دارای محدودیت ظرفیت بوده ولی مثال‌های $C6$-$C10$, $C13$, $C14$ علاوه بر داشتن محدودیت ظرفیت، دارای محدودیت‌هایِ طول مسیر و زمان سروِیس‌دهِی نیز هستند. به عبارت دیگر در این مثال‌ها هر وسیله نقلیه نمی‌تواند بیشتر از یک مقدار معین مسیر پیمایش کند. بایِد توجه داشت که در اِین جدول بهترِین جواب‌هایِ شناخته شده **BKS** نشان داده شده است.

جدول ۴: خصوصیات مثال‌های کریستوفیدز و همکارانش

مثال	تعداد گره	تعداد وسیله نقلیه	ظرفیت	ماکزیمم طول هر تور	زمان سرویس‌دهی	BKS
C1	۵۰	۵	۱۶۰	∞	–	۵۲۴٫۸۱
C2	۷۵	۱۰	۱۴۰	∞	–	۸۳۵٫۲۶
C3	۱۰۰	۸	۲۰۰	∞	–	۸۲۶٫۱۴
C4	۱۵۰	۱۲	۲۰۰	∞	–	۱۰۲۸٫۴۲
C5	۱۹۹	۱۷	۲۰۰	∞	–	۱۲۹۱٫۴۵
C6	۵۰	۶	۱۶۰	۲۰۰	۱۰	۵۵۵٫۴۳
C7	۷۵	۱۱	۱۴۰	۱۶۰	۱۰	۹۰۹٫۶۸
C8	۱۰۰	۹	۲۰۰	۲۳۰	۱۰	۸۶۵٫۹۴
C9	۱۵۰	۱۴	۲۰۰	۲۰۰	۱۰	۱۱۶۲٫۵۵
C10	۱۹۹	۱۸	۲۰۰	۲۰۰	۱۰	۱۳۹۵٫۸۵
C11	۱۲۰	۷	۲۰۰	∞	–	۱۰۴۲٫۱۱
C12	۱۰۰	۱۰	۲۰۰	∞	–	۸۱۹٫۵۶
C13	۱۲۰	۱۱	۲۰۰	۷۲۰	۵۰	۱۵۴۱٫۱۴
C14	۱۰۰	۱۱	۲۰۰	۱۰۴۰	۹۰	۸۶۶٫۳۷

در جدول ۵، جواب و زمان (برحسب ثانِیه) حاصل از مقایسه الگوریتم پیشنهادی با تعدادی از بهترین روش‌های موجود برای مسئله VRP نشان داده شده است. در این جدول، SA روش شبیه‌سازی آنیلی عثمان [۲۹]، TS روش جستجوی ممنوع عثمان [۲۹]، GA روش الگوریتم ژنتیک ارائه شده توسط برگر و براکااوِی [۳۰]، PSO روش الگوریتم پرندگان به وسِیله آی و همکارش [۳۱] و در نهایت ACO-SS روش ترکیبی ACO به همراه روش جستجوی محلی پراکنده به وسِیله زانگ و تانک [۳۲] است.

جدول ۵: مقایسه الگوریتم با بهترین الگوریتم‌هایِ فراابتکاری

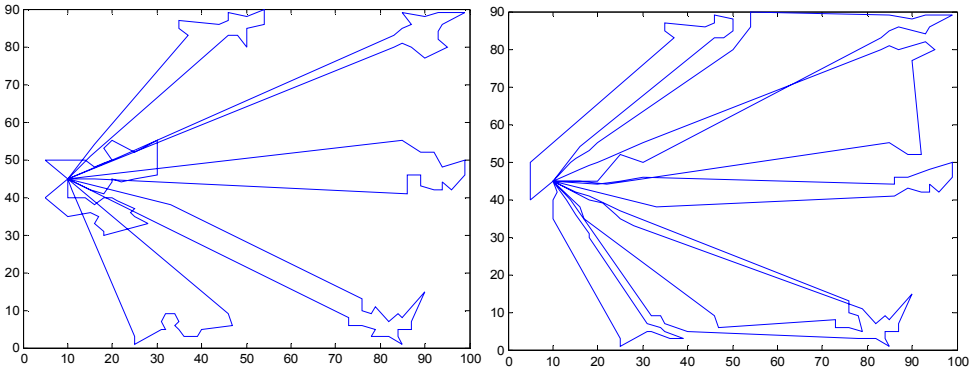
مثال	تعداد گره	SA		TS		GA		ACO-SS		PSO		الگوریتم پیشنهادی		BKS
		جواب	زمان	جواب	زمان	جواب	زمان	جواب	زمان	جواب	زمان	جواب	زمان	
C1	۵۰	۵۲۸	۱۶۷	**۵۲۴**	۱۱۴	**۵۲۴،۸۱**	۲۱۳	**۵۲۴،۸۱**	۵۵	**۵۲۴،۸۱**	--	**۵۲۴،۸۱**	۳۵	۵۲۴،۸۱
C2	۷۵	۸۳۸	۶۴۳۴	۸۴۴	۱۷۹	۸۴۹،۷۷	۷۶۵	**۸۳۵،۲۶**	۷۰	۸۴۴،۴۲	–	**۸۳۵،۲۶**	۶۵	۸۳۵،۲۶
C3	۱۰۰	۸۲۹	۹۳۳۴	۸۳۵	۱۵۴۳	۸۴۰،۷۲	۱۱۴۸	۸۳۰،۱۴	۱۲۰	۸۲۹،۴۰	–	۸۲۹،۴۰	۹۶	۸۲۶،۱۴
C4	۱۵۰	۱۰۵۸	۵۰۱۲	۱۰۵۲	۳۵۶۰	۱۰۵۵،۸۵	۲۴۷۵	۱۰۳۸،۲۰	۲۵۱	۱۰۴۸،۸۹	–	۱۰۵۲	۱۶۷	۱۰۲۸،۴۲
C5	۱۹۹	۱۳۷۶	۲۳۱۸	۱۳۵۴	۳۲۴۶	۱۳۷۸،۷۳	۳۹۹۹	۱۳۰۷،۱۸	۷۰۸	۱۳۲۳،۸۹	–	۱۳۲۳،۸۹	۵۶۲	۱۲۹۱،۴۵
C6	۵۰	**۵۵۵**	۳۴۱۰	۵۵۵	۱۷۳	۵۶۰،۲۹	۲۱۷	۵۵۹،۱۲	۶۵	**۵۵۵،۴۳**	–	**۵۵۵،۴۳**	۴۷	۵۵۵،۴۳
C7	۷۵	**۹۰۹**	۶۲۷	۹۱۳	۱۰۵۷	۹۱۴،۱۳	۷۸۶	۹۱۲،۶۸	۹۰	۹۱۷،۶۸	–	۹۰۹،۶۸	۷۶	۹۰۹،۶۸
C8	۱۰۰	۸۶۶	۹۵۷	۸۶۶	۲۹۹۸	۸۷۲،۸۲	۱۱۳۴	۸۶۹،۳۴	۲۱۰	۸۶۷،۰۱	–	۸۶۹،۳۴	۱۶۲	۸۶۵،۹۴
C9	۱۵۰	۱۱۶۴	۸۴۳۰۱	۱۱۸۸	۴۷۵۶	۱۱۹۳،۰۵	۲۲۵۸	۱۱۷۹،۴	۵۲۱	۱۱۸۱،۱۴	–	۱۱۸۱،۱۴	۳۳۹	۱۱۶۲،۵۵
C10	۱۹۹	۱۴۱۸	۵۷۰۸	۱۴۲۲	۴۵۶۱	۱۴۸۳،۰۶	۳۶۸۷	۱۴۱۰،۲۶	۱۰۱۲	۱۴۲۸،۴۶	–	**۱۳۹۵،۸۵**	۸۶۱	۱۳۹۵،۸۵
C11	۱۲۰	۱۱۷۶	۳۱۶	**۱۰۴۲**	۱۴۴۵	۱۰۶۰،۲۴	۱۶۳۳	۱۰۴۴،۱۲	۲۳۲	۱۰۵۱،۸۷	–	**۱۰۴۲،۱۱**	۱۷۶	۱۰۴۲،۱۱
C12	۱۰۰	۸۲۶	۶۳۲	**۸۱۹**	۸۹۲	۸۷۷،۸	۱۱۶۰	۸۲۴،۳۱	۱۵۶	**۸۱۹،۵۶**	–	۸۲۴،۳۱	۷۹	۸۱۹،۵۶
C13	۱۲۰	۱۵۴۵	۷۶۲۳	۱۵۴۷	۲۸۳۴	۱۵۶۲،۲۵	۱۶۹۴	۱۵۵۶،۵۲	۴۶۷	۱۵۴۶،۲۰	–	**۱۵۴۱،۱۴**	۴۱۹	**۱۵۴۱،۱۴**
C14	۱۰۰	۸۹۰	۳۰۵	**۸۶۶**	۱۱۷۶	۸۷۲،۳۴	۱۱۹۷	۸۷۰،۲۶	۳۶۹	**۸۶۶،۳۷**	–	**۸۶۶،۳۷**	۲۷۱	**۸۶۶،۳۷**

مثال‌های نشان داده شده در اِین جدول را می‌توان از نظر تعداد مشترِی به سه دسته کمتر از ۱۰۰ (کوچک)، مابین ۱۰۰ تا ۱۵۰ (متوسط) و بالاتر از ۱۵۰ مشترِی (بزرگ) تقسیم‌بندی نمود. بنابراین می‌توان نتایج الگوریتم را روِی این سه دسته به طور مجزا مورد بررسی قرار داد. در دسته اول شامل چهار مثال الگوریتم دارای موفقیت کامل بوده و توانسته است که در ۰/۱۰۰ مثال‌ها به بهترین جواب‌های تاکنون یافته شده، دست پیدا کند. این نشان می‌دهد که الگوریتم در مثال‌های کوچک از کارایی خوبی برخوردار است. در دسته دوم و در شش مثال موجود، در ۰/۶۷ مثال‌ها الگوریتم توانسته است که به بهترین جواب‌ها دست پیدا کند. بنابراِین مِی‌توان گفت که الگوریتم کارایی بسِیار خوبِی برای حل مثال‌هایِ متوسط دارد. نهایِتاً در دسته سوم شامل چهار مثال الگوریتم از کارایِی مناسبِی برخوردار است و توانسته در یِک مثال موجود به بهترِین جواب‌هایِی تاکنون بدست آمده، دست یِابد.

با دِیدن نتایِج الگوریتم‌ها مِی‌توان گفت که *SA* فقط در دو مثال توانسته است که جواب‌هایِ بهِینه را تولِید کند ولِی الگوریتم *TS* در چهار مثال توانسته به جواب‌هایِ *BKS* دست یِابد. الگوریتم *GA* از اِین نظر جواب‌هایِ ضعِیف‌ترِی نسبت به دو الگوریتم قبلِی بدست آورده است و فقط در یِک مثال توانسته که به جواب بهِینه برسد. الگوریتم دِیگرِی که در اِین جدول توانسنه نتایِج مشابهِی را با الگوریتم *SA* بدست آورد الگوریتم *ACO-SS* است که فقط در دو مثال توانسته به جواب‌هایِ *BKS* دست یِابد. از طرف دِیگر الگوریتم *PSO* همانند الگوریتم *TS* در چهار مثال به بهترِین جواب‌ها دست پِیدا کرده است و تواسنته نسبت به سه الگوریتم قبلِی، جواب‌هایِ بهترِی تولید کند. باید توجه کرد که در اِین جدول الگوریتم *PA* بهترِین جواب‌ها را بدست آورده است و در هشت مثال همانطور که گفته شده به بهترِین جواب ها دست پِیدا کرده است.

باید توجه کرد که اگر الگوریتم‌ها را از نظر میانگین جواب‌ها مورد مقاِیسه قرار گِیرد، الگوریتم در مقایسه با الگوریتم‌هایِ *GA* و *SA* دارای کارایی بسِیار بیشتری بوده و به جواب‌هایِ بهتری دست پیدا کرده است در حالِی که این الگوریتم در مقایسه با الگوریتم‌هایِ *PSO*، *ACO-SS* و *TS* دارای رقابت بسیار خوبی بوده و با فاصله کمی از آن‌ها قرار دارد و توانسته جواب‌هایِ بهترِی را بدست

آورد. در نتیجه می‌توان گفت که الگوریتم پیشنهادِی یک روش کاملاً رقابتی برای حل این مسئله محسوب می‌شود که توانسته است به جواب‌های باکیفیتی دست پیدا کند به طوری که از بین ۱۴ مثال آزمایش شده به ۸ تا از بهترین جواب‌های تاکنون بدست آمده، دست یافته است. به طور نمونه دو جواب نهایِی که به وسیله الگوریتم پیشنهادِی بدست آمده است در شکل ۷ نشان داده شده است.



C۱۱ C۱۳

شکل ۷: چندین جواب بدست آمده بوسیله الگوریتم پیشنهادِی

نتیجه‌گیری

در این مقاله یک روش جدید به نام روش رقابت استعماری برای حل مساله مسیریابی وسیله نقلیه ارائه شد. با مقایسه نتایج بدست آمده می‌توان دریافت که الگوریتم از کارایی قابل قبولی برای مسائل با اندازه کوچک است اما هنگامیکه اندازه مساله افزایش پیدا می‌کند و به بیش از ۱۵۰ مشتری می‌رسد کارایی الگوریتم کاهش می‌یابد. به نظر می‌رسد که استفاده از روش‌های ترکیبی فراابتکاری مانند ترکیب این روش با روش‌های الگوریتم مورچگان، جستجوی ممنوع و شبیه‌سازی آنیلی و غیره یا استفاده از الگوریتم‌های قوی محلی می‌تواند راهکارهای مناسب برای افزایش کارایی الگوریتم باشد. همچنین گسترش این روش به گسترش‌هایِ دیگر مساله مسیریابی وسیله نقلیه می‌تواند نتایج خوبی را در پی داشته باشد. استفاده از این الگوریتم و کاربردی کردن آنها به مقاله‌های بعدی موکول می‌شود.

فهرست منابع

[1] Yousefikhoshbakht, M., Didehvar, F., Rahmati, R., "An Efficient Solution for the Vehicle Routing Problem by Using a Hybrid Elite Ant Colony Optimization", International Journal of Computers Communications & Control, Vol. 9, No. 3, 2014, pp. 340-347.

[2] Yousefikhoshbakht, M., Didehvar, F., Rahmati, R. & Sedighpour, M. (2012). An effective imperialist competitive algorithm for solving the open vehicle routing problem. Transportation Research Journal, 9(1), 83-95.

[3] Yousefikhoshbakht, M., Didehvar, F., Rahmati, R., "Optimization of Vehicle Routing Problem with Simultaneous Pickup and Delivery Using a Modified Tabu Search," Journal of Advances in Computer Research, Vol. 3. No. 4, 2012. pp. 55-66.

[4] Yousefikhoshbakht, M., Didehvar, F., Rahmati, R., "A Combination of Modified Tabu Search and Elite Ant System to Solve the Vehicle Routing Problem with Simultaneous Pickup and Delivery", Journal of Industrial and Production Engineering, Vol. 31, No. 2, 2014, 65–75.

[5] Yousefikhoshbakht, M., Didehvar, F., Rahmati, R., Solving the Heterogeneous Fixed Fleet Open Vehicle Routing Problem by a Combined Meta-heuristic Algorithm, International of production research, Vol. 52, No. 9, 2014, pp. 2565-2575.

[6] Contardo, C., Desaulniers, G. and Lessard, F., Reaching the elementary lower bound in the vehicle routing problem with time windows. Networks, 65: 88–99. doi: 10.1002/net.21594, 2015.

[7] Yousefikhoshbakht, M., Didehvar, F., Rahmati, R. & Saadati Eskandari, Z. (2012). A hybrid ant colony system for the heterogeneous fixed fleet vehicle routing problem. Transportation Research Journal, 9(2), 191-207.

[8] Yousefikhoshbakht, M., Didehvar, F., Rahmati, R., Modification of the Ant Colony Optimization for Solving the Multiple Traveling Salesman Problem, Journal of information science and technology, Romanian journal of information science and technology, Vol. 16, No. 1, 2013, pp. 65–80.

[9] Christofides, N., Mingozzi, A., Toth, P., "*Exact algorithms for the vehicle routing problem based on spanning tree and shortest path relaxations*", Mathematical Programming, vol. 20, pp. 255–282, 1981.

[10] Fisher, M., "*Optimal solution of vehicle routing problems using minimum K-trees*", Operations Research, vol. 42 (4), pp. 626–642, 1994.

[11] Toth, P., Vigo, D., "*An exact algorithm for the vehicle routing problem with backhauls*", Transportation Science, vol. 31, pp. 372–385, 1997.

[12] Christofides N, Mingozzi A, Toth P. *The vehicle routing problem*. In: Christofides N, Mingozzi A, Toth P, Sandi C, editors. Combinatorial Optimization. Chichester: Wiley; pp 315–38, 1979.

[13] Yellow, P., "*A computational modification to the saving method of vehicle scheduling*", Operational Research Quarterly, vol. 21, pp. *2*81-283, 1970.

[۱۴] چهارسوقی، کمال؛ اشرفی، مهدی، انتخاب تأمین کننده پایدار و تخصیص سفارش با الگوریتم تغییر شکل یافته بندرز، مجله مدلسازی پیشرفته ریاضی، دوره ۳، شماره ۲، ص ۸۱–۱۰۱، ۱۳۹۲.

[15] Prins, C., "*Two memetic algorithms for heterogeneous fleet vehicle routing problems*", Engineering Applications of Artificial Intelligence, vol. 22, pp. 916–928, 2009.

[16] Marinakis, Y., Marinaki, M., "*A hybrid genetic – Particle Swarm Optimization Algorithm for the vehicle routing problem*", Expert Systems with Applications, 2009.

[17] Kaiwartya, O., Kumar, S., Lobiyal, D. K., Kumar Tiwari, P., Hanan Abdullah, A. and Nazar Hassan, A., Multiobjective Dynamic Vehicle Routing Problem and Time Seed Based Solution Using Particle Swarm Optimization, Journal of Sensors, Article ID 189832, 14 pages, 2015. doi:10.1155/2015/189832, 2015.

[۱۸] یوسفی خوشبخت، مجید؛ دیده ور، فرزاد؛ رحمتی، فرهاد، کاربرد یک الگوریتم اصلاحی رقابت استعماری برای حل مسأله‌ی فروشنده دوره گردمجله مدلسازی پیشرفته ریاضی، دوره ۱، شماره ۲، ص ۲۹–۴۹، ۱۳۹۰.

[19] Rajabioun, R., Hashemzadeh, F., Atashpaz-Gargari, E., "*Colonial competitive algorithm A novel approach for PID controller design in MIMO distillation column process*", International Journal of Intelligent Computing and Cybernetics, vol. 1 (3), pp. 337–355, 2008.

[20] Kaveh, A., Talatahari, S., "*Optimum design of skeletal structures using imperialist competitive algorithm*", Computers & Structures, vol. 88(21-22), pp. 1220-1229, 2010.

[21] Niknam, T., Taherian Fard, E., Pourjafarian, N., Rousta, A., "*An efficient hybrid algorithm based on modified imperialist competitive algorithm and K-means for data clustering*", Engineering Applications of Artificial Intelligence, 2010.

[22] Lucas, C., Nasiri-Gheidari, Z. and Tootoonchian, F., "*Application of an imperialist competitive algorithm to the design of a linear induction motor*", Energy Conversion and Management, vol. 51(7), pp. 1407-1411, 2010.

[23] Rajabioun, R., Atashpaz-Gargari, E. and Lucas, C., "*Colonial Competitive Algorithm as a Tool for Nash Equilibrium Point Achievement*", Lecture Notes in Computer Science, vol. 5073, pp. 680-695, 2008.

[24] Nazari-Shirkouhi, S., Eivazy, H., Ghodsi, R., Rezaie, K., Atashpaz-Gargari, E., "*Solving the integrated product mix-outsourcing problem using the Imperialist Competitive Algorithm*", Expert Systems with Applications, vol. 37, pp. 7615–7626, 2010.

[25] Toth, P., Vigo(eds), D., "*The Vehicle Routing Problem*", SIAM monographs on discrete mathematics and applications, 2002.

[26] Clark G, Wright JW., "*Scheduling of vehicles from a central depot to a number of delivery points*", Operations Research, vol. 12, pp. 568–581, 1964.

[27] Berger, J., Barkaoui, M., "*A hybrid genetic algorithm for the capacitated vehicle routing problem"*, in: Proceedings of the International Genetic and Evolutionary Computation

Conference – GECCO03, LNCS 2723, pp. 646–656, 2003.

[28] Mazzeo, S., Loiseau, I., "*An Ant Colony Algorithm for the Capacitated Vehicle Routing*", Elestronic Notes in Discrete Mathematics, vol.18, pp. 181–186, 2004.

[29] Osman LH., "*Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem*", Ann Operations Research, vol. 41, pp. 421–51, 1993.

[30] Berger, J. and Barkaoui, M., A hybrid genetic algorithm for the capacitated vehicle routing problem, in: Proceedings of the International Genetic and Evolutionary Computation Conference GECCO03, LNCS 2723, 2003, pp. 646–656.

[31] Ai, T. J. and Kachitvichyanukul, V., A Particle Swarm Optimization for the Heterogeneous Fleet Vehicle Routing Problem, International Journal of Logistics and SCM Systems, Vol. 3, No. 1, 2009, pp. 32-39.

[32] Zhang, X. and Tang, L., A new hybrid ant colony optimization algorithm for the vehicle routing problem, Pattern Recognition Letters, Vol. 30, No. 9, 2009, pp. 848-855.