مجله تحقیق در عملیات و کاربردهای آن، سال هشتم، شماره ۳ (پیاپی ۳۰)، پاییز ۹۰، ص ص ۹۶-۸۳
شاپا ۷۲۸۶-۲۲۵۱

الگوریتم نمونه اصلاحی مورچگان برای حل مساله چندین فروشنده دوره‌گرد

مجید یوسفی خوشبخت*[1]، محمد صدیق‌پور[2]
[1] دانشگاه آزاد اسلامی، واحد همدان، باشگاه پژوهشگران جوان ، همدان، ایران
[2] مربی ، دانشکده علوم پایه، واحد همدان، دانشگاه آزاد اسلامی، همدان، ایران

رسید مقاله: سی ویکم فروردین ماه ۱۳۹۰
پذیرش مقاله: بیست و هشتم مرداد ماه ۱۳۹۰

چکیده

یکی از کاربردی‌ترین مسایل بهینه‌سازی ترکیباتی مساله چندین فروشنده دوره‌گرد است که در آن، $m>1$ فروشنده از یک نقطه مشترک به نام انبار شروع به حرکت می‌کنند و بعد از ملاقات کردن $n>m$ مشتری به آن باز می‌گردند، به طوری که هر مشتری تنها یک‌بار به وسیله یک فروشنده مورد ملاقات قرار می‌گیرد. هدف در این مساله کمینه‌کردن مسیر کلی پیموده شده توسط همه فروشنده‌های دوره‌گرد است. در این مقاله، یک روش اصلاحی الگوریتم نمونه مورچگان برای این مساله به کار گرفته می‌شود که در ضریب تشویق بهترین مسیر با الگوریتم معمولی نمونه مورچگان تفاوت دارد. این ضریب سبب می‌شود که الگوریتم دارای قدرت بیشتری برای فرار از نقاط بهینه محلی شود و بتواند به جواب‌های با کیفیت‌تری دست یابد. نتایج محاسباتی روی مثال‌های استاندارد به وضوح کارایی الگوریتم جدید را نسبت به الگوریتم‌های دیگر فراابتکاری نشان می‌دهد.

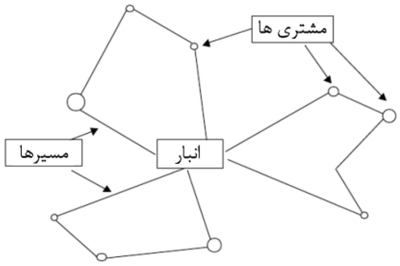
کلمات کلیدی: مساله چندین فروشنده دوره‌گرد، الگوریتم نمونه مورچگان، ضریب فرمون، مسایل بهینه‌سازی ترکیباتی.

۱ مقدمه

مساله چندین فروشنده دوره‌گرد یکی از مهم‌ترین گسترش‌های مساله فروشنده دوره‌گرد است که از اهمیت بسیار زیادی در مسایل بهینه‌سازی ترکیباتی برخوردار است. در این مساله m فروشنده از گره‌ای به نام انبار شروع به حرکت کرده و بعد از ملاقات کردن n مشتری به محل حرکت خود باز می‌گردند. هدف در این مساله معین‌کردن مجموعه‌ای از مسیرها برای فروشنده‌ها است به طوری که هزینه کلی آن‌ها کمینه شده و هر مشتری فقط به وسیله یک فروشنده مورد ملاقات قرار گیرد (شکل ۱). بعضی از عللی که سبب شده است این مساله از جایگاه بسیار مهمی در مسایل بهینه‌سازی ترکیباتی برخوردار باشد عبارت است از:

* عهده دار مکاتبات
آدرس الکترونیکی: *khoshbakht@iauh.ac.ir*

۱. وجود چندین مساله گسترش یافته از آن مانند مساله مسیریابی چندین وسیله نقلیه و گسترش‌های آن.
۲. قابلیت تبدیل مسایل گوناگون به این مساله و حل کردن آن‌ها به وسیله الگوریتم‌های موجود و تفسیر جواب‌های به دست آمده در مساله اصلی مانند مساله ترتیب‌دهی کارها.
۳. استاندارد بودن این مساله برای الگوریتم‌هایی که به طور موازی کار می‌کنند، زیرا این‌گونه از الگوریتم‌ها به راحتی در این مساله قابل اجرا هستند و می‌توان بدین وسیله کارایی الگوریتم‌های جدید موازی را بررسی کرد.



شکل ۱. نمونه‌ای از حل مساله چندین فروشنده دوره‌گرد با $m=3$ و $n=10$

اگر چه مساله فروشنده دوره‌گرد بسیار مورد توجه دانشمندان و محققین قرار می‌گیرد، اما مساله چندین فروشنده دوره‌گرد علیرغم جایگاه بسیار مهمی که دارد کمتر مورد بررسی قرار می‌گیرد و تحقیقات اندکی درباره آن انجام شده است [۱]. با این حال می‌توان روش‌های حل این مساله را به سه دسته الگوریتم‌های دقیق، تقریبی و ابتکاری تقسیم کرد. در الگوریتم‌های دقیق مانند روش تخفیف لاکرانژ یدلاپالی و همکارانش [۲]، روش شاخه و کران گاویش و اسریکانت [۳] و روش شبه تخصیص گرومیکو و همکارانش [۴] که بیشتر برای مسایل با اندازه تقریباً کوچک بکار می‌رود، جواب بهینه مساله به دست می‌آید علیرغم آنکه زمان زیادی برای به دست آمدن این جواب باید صرف شود. از طرف دیگر روش‌های دقیق دیگری نیز وجود دارند که ایده اصلی آن‌ها تخفیف چندین محدودیت و ساده‌سازی مساله مربوطه است مانند روش ارایه شده در [۵] که بر اساس روش شاخه و کران می‌باشد.

به علاوه در روش‌های تقریبی اگر چه نمی‌توان مانند روش‌های دقیق به جواب بهینه مساله رسید اما می‌توان با یک کران بالای خطا برای جواب به دست آمده یا زمان اجرا به جواب رسید. برای نمونه از این‌گونه روش‌ها می‌توان به [۶] و [۷] مراجعه کرد. در مقابل، روش‌های ابتکاری روش‌هایی هستند که برخلاف روش‌های دقیق معمولاً به جواب بهینه مساله نمی‌رسند اما می‌توانند جواب زیر بهینه مساله را در یک زمان قابل قبول به دست آورند. این الگوریتم‌ها علی‌رغم اینکه در یک زمان اندک به جواب زیر بهینه می‌رسند دارای ساختار خوبی برای فرار از نقاط بهینه محلی نیستند و معمولاً به جواب خوبی همگرا نمی‌شوند. به طور نمونه از این روش‌ها می‌توان به موارد [۸] و [۹] اشاره کرد که از اولین الگوریتم‌های ابتکاری برای حل مساله چندین فروشنده دوره‌گرد هستند. از طرف دیگر در چندین دهه اخیر نسخه‌ای جدید از الگوریتم‌های ابتکاری پیشنهاد شدند که دارای ساختار تصادفی برای رسیدن به جواب هستند. این الگوریتم‌ها که فراابتکاری نامیده می‌شوند می‌توانند تا حد امکان از بهینه‌های محلی فرار کرده و به جواب‌های بسیار خوبی همگرا شوند. به طور مثال برای این‌گونه از روش‌ها می‌توان به روش‌های

شبکه‌های عصبی ماسوتی و همکارش [۱۰]، جستجوی ممنوع رایان و همکارانش [۱۱]، الگوریتم‌های مورچگان یوسفی خوشبخت و همکارش [۱۲] و همچنین غفوریان و همکارش [۱۳] و الگوریتم شبیه‌سازی آنیلی پایدار و همکارانش [۱۴] اشاره کرد.

در این مقاله در بخش دوم برای این مساله یک فرمول‌بندی بر اساس مساله فروشنده دوره‌گرد ارایه می‌شود و سپس در بخش‌های سوم و چهارم در ابتدا الگوریتم معمولی مورچگان (ACO) ارایه شده و سپس اصلاح انجام شده بر روی الگوریتم نمونه مورچگان (AS_{elite}) به طور مبسوط شرح داده می‌شود. تحلیل حساسیت پارامترهای روش پیشنهادی در بخش پنجم مورد آزمایش قرار می‌گیرد و مقدار بهینه پارامترها در این بخش به دست می‌آید. در بخش ششم بین الگوریتم پیشنهادی و الگوریتم‌های دیگر فراابتکاری مقایسه‌ای بر روی مثال‌های استاندارد انجام می‌شود، در حالی که در بخش هفتم نتیجه‌گیری و جهت‌گیری‌های آینده ذکر می‌گردد.

۲ فرمول‌بندی مساله

مساله چندین فروشنده دوره‌گرد روی گراف غیر جهت دار کامل $G(V,A)$ تعریف می‌شود که در آن $V=\{0,1,\ldots,n\}$ مجموعه‌ای از n مشتری به همراه انبار و $A=\{(i,j):i,j\in V, i\neq j\}$ مجموعه‌ای از کمان‌ها است (اگر گراف کامل نبود آن گاه هر یال وجود نداشته به وسیله یالی با اندازه بی‌نهایت جایگزین می‌شود). فرض کنید $C=(c_{ij})_{(n+1)\times(n+1)}$ نشان‌دهنده ماتریس هزینه روی گراف G باشد. اگر برای هر $i,j=0,1,2,\ldots,n$ $c_{ij}=c_{ji}$ آن گاه ماتریس C را متقارن گویند. همچنین اگر برای هر $i,j,k\in V$ $c_{ik}\leq c_{ij}+c_{jk}$ باشد آن گاه ماتریس C در نامساوی مثلثی صدق می‌کند.

علیرغم اینکه مساله چندین فروشنده دوره‌گرد جزء مسایل NP-سخت محسوب می‌شود، می‌تواند به صورت یک مدل برنامه‌ریزی خطی صحیح نمایش داده شود، برای این منظور فرض می‌شود که ماتریس هزینه C متقارن و در نامساوی مثلثی صدق می‌کند. همچنین اگر برای هر $(i,j=0,1,2,\ldots,n; i\neq j)$ فروشنده به طور مستقیم از i به j حرکت کند $x_{ij}=1$ و در غیر این صورت $x_{ij}=0$ در نظر گرفته می‌شود.

بنابراین مدل مساله چندین فروشنده دوره‌گرد عبارت است از:

$$Min \sum_{i=0}^{n}\sum_{j=0}^{n} c_{ij}x_{ij} \tag{۱}$$

$$s.t.$$

$$\sum_{i=0}^{n} x_{ij}=\begin{cases}1 & j=1,\ldots,n\\ m & j=0\end{cases} \tag{۲}$$

$$\sum_{j=0}^{n} x_{ij}=\begin{cases}1 & i=1,\ldots,n\\ m & i=0\end{cases} \tag{۳}$$

$$\sum_{i,j\in S} x_{ij} \le |S|-1 \;\; (S \subset \{1,...,n\}, |S| \ge 2) \tag{۴}$$

$$x_{ij} \in \{0,1\} \qquad i,j = 0,...,n. \tag{۵}$$

در این مدل، (۱) تابع هدف مساله را نشان می‌دهد. در حالی که محدودیت (۲) نشان‌دهنده این است که علیرغم اینکه به هر مشتری j به غیر از انبار فقط یک یال وارد می‌شود، به گره انبار m یال وارد می‌شود. همچنین محدودیت (۳) مانند محدودیت (۲) به این نکته اشاره دارد که از هر مشتری i به غیر از انبار فقط یک یال خارج می‌شود در حالی که این تعداد یال برای گره انبار m می‌باشد. محدودیت (۴) محدودیت حذف زیرگذر استاندارد نامیده می‌شود که سبب می‌گردد جواب‌هایی که دارای زیر دوری هستند که شامل گره انبار نیست، مورد قبول قرار نگیرد. در نهایت محدودیت (۵) به شرایط دودویی متغیر x_{ij} اشاره می‌کند.

۳ الگوریتم مورچگان

بهینه‌سازی جمعیت مورچگان نوعی روش فراابتکاری است که در حل تعداد زیادی از مسایل بهینه‌سازی ترکیباتی کاملاً موفق عمل کرده است. ACO به عنوان یک ابزار جهت حل مساله فروشنده دوره‌گرد توسط دوریگو و همکارانش در سال ۱۹۹۲ مطرح شد [۱۵]. این الگوریتم که نوعی از سیستم‌های چندعامله‌ای است، از رفتار غذایابی مورچه‌های واقعی الهام گرفته شده است به طوری که هر عامل، یک مورچه مصنوعی می‌باشد. همچنین این الگوریتم نمونه موفقی از سیستم‌های هوشمند گروهی است، که در آن هر عامل، عمل ساده‌ای را انجام می‌دهد ولی انجام این عمل ساده در کل باعث حل شدن مسایل NP-سخت می‌شود. گروهی از مهم‌ترین مسایلی که در آن از این روش استفاده شده است مسایل کلاسیک فروشنده دوره‌گرد، تخصیص، ترتیب‌دهی کارها و مسیریابی در شبکه های ارتباطی از راه دور می‌باشند.

آزمایشات زیست‌شناسان بر روی مورچه های آرژانتینی نشان داد که اگر دو راه برای مورچه‌ها از لانه تا منبع غذایی در نظر گرفته شود، اغلب مورچه‌ها بعد از مدتی ناپایداری (در حدود چند دقیقه) مسیر کوتاه‌تر را انتخاب می‌کنند و این نسبت مورچه‌ها با افزایش تفاوت طول دو مسیر، افزایش می‌یابد. این عمل از این ناشی می‌شود که مورچه‌ها در طول مسیر رفت و برگشت ماده شیمیایی به نام فرمون آزاد می‌کنند، که تبخیر پذیر می‌باشد. به عبارت دیگر وقتی مورچه‌ها به نقطه تصمیم‌گیری می‌رسند، مسیر را در ابتدا به صورت تصادفی انتخاب می‌کنند زیرا هیچ فرمونی بر روی مسیرها موجود نمی‌باشد. اما بعد از مدتی که مورچه‌های قبلی بر اساس مسیر کوتاه‌تر و منبع غذایی بهتر فرمون‌ریزی کردند آن گاه انتخاب در این زمان به صورت احتمالی صورت می‌گیرد. یعنی مسیر دارای فرمون بیشتر، مورد انتخاب تعداد بیشتری از مورچه‌ها قرار می‌گیرد. این رفتار در حقیقت یک اثر تحریک کننده دارد، زیرا انتخاب یک مسیر در تکرار جاری، احتمال انتخاب دوباره آن را در آینده افزایش خواهد داد. به این ترتیب با تکرار این عمل به وسیله مورچه‌ها، فرمون آزاد شده روی مسیر کوتاه‌تر با مقدار بالاتری ذخیره می‌شود تا مسیر کوتاه‌تر نسبت به مسیر بلندتر دارای جذابیت بیشتری باشد.

دوریگو این ایده ساده را برای یافتن راه‌حل‌های خوب در مسایل بهینه‌سازی سخت مورد استفاده قرار داد و روش الگوریتم سیستم مورچگان (AS) را به عنوان اولین نسخه از الگوریتم ACO ارایه کرد. در این الگوریتم وظیفه اصلی هر مورچه مصنوعی مانند همتای طبیعی خود، یافتن کوتاه‌ترین مسیر بین یک جفت گره در یک گراف است که در آن، مساله به نحو مناسبی روی آن نگاشته شده است. بنابراین مساله به زیر مساله‌هایی تبدیل می‌شود که در آن مورچه‌های مصنوعی وظیفه دارند که انتخاب گره بعدی را بر اساس فرمون ریخته شده و فاصله تا گره بعدی، انجام دهند. قانون تصمیم برای مورچه k واقع در گره i، که می‌خواهد یکی از گره‌ها را از مجموعه گره‌های ملاقات نشده N_i انتخاب کند از فرمول (۶) به دست می‌آید. در اینجا τ_{ij} نشان دهنده مقدار فرمون روی یال (i,j) بوده، در حالی که η_{ij} نشان‌دهنده عکس مقدار فاصله بین دو گره i و j است. البته هر کدام از این دو دارای توانی هستند که می‌توان با تغییر آن‌ها میزان اهمیت هر یک را نسبت به دیگری تغییر داد.

$$P_{ij}^{k}=\begin{cases}\dfrac{{\tau_{ij}}^{\alpha}{\eta_{ij}}^{\beta}}{\sum\limits_{j\in N_i}{\tau_{ij}}^{\alpha}\eta_{ij^{\beta}}} & if \quad j\in N_i\\ 0 & if \quad j\notin N_i\end{cases} \tag{۶}$$

همچنین مورچه‌ها در حالی که از یک گره i به گره دیگر j می‌روند، اطلاعات فرمون $\Delta\tau_{ij}$، معمولاً به اندازه عکس مقدار یال (i,j)، را روی یال مربوطه می‌ریزند. این عمل با استفاده از فرمول (۷) انجام می‌شود.

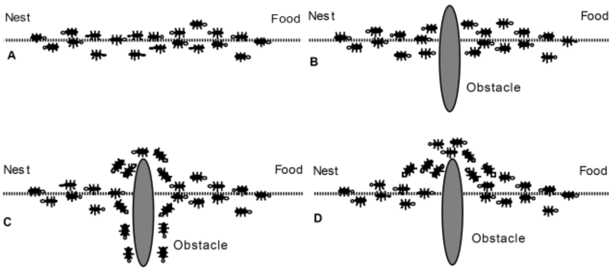
$$\tau_{ij}(t)\leftarrow\tau_{ij}(t)+\Delta\tau_{ij} \tag{۷}$$

به علاوه الگوریتم مانند نسخه طبیعی آن و برای اجتناب از همگرایی سریع همه مورچه‌ها به یک مسیر زیر بهینه، از مکانیزم تبخیر فرمون استفاده می‌کند. یعنی غلظت فرمون به طور خودکار و در هر تکرار به مقدار ρ کاهش می‌یابد. به عبارت دیگر اگر τ ماتریس فرمون موجود بر روی یال‌های گراف مربوطه باشد، آن گاه این ماتریس در هر تکرار به وسیله فرمول (۸) بروز می‌شود.

$$\tau\leftarrow(1-\rho)\tau \qquad \rho\in(0,1] \tag{۸}$$

اولین مساله‌ای که با AS حل شد، مساله فروشنده دوره‌گرد بود زیرا به راحتی با این الگوریتم تطبیق یافته و از مسایلی است که بسیار به آن پرداخته می‌شود. به علاوه چون توضیح و درک آن ساده است، تاکنون الگوریتم‌های زیادی بر روی آن اجرا شده است. از طرف دیگر الگوریتم AS اولین نسخه از الگوریتم ACO بود که برای این مساله مورد استفاده قرار گرفت و متأسفانه نتایج خوبی را در مقایسه با سایر الگوریتم‌های فراابتکاری آن زمان تولید نکرد. در نتیجه دانشمندان به تکاپو افتادند که نسخه‌های جدیدتری از الگوریتم را ایجاد کنند که بتواند در مقایسه با سایر الگوریتم‌های فراابتکاری نتایج بهتری را به دست آورد. تلاش‌ها سبب شد که الگوریتم‌هایی مانند AS_{elite}، رتبه بندی مورچگان (AS_{rank})، جمعیت مورچگان (ACS) و ماکزیمم – مینیمم

مورچگان (MMAS) ارایه شوند که همگی نسخه‌هایی اصلاح شده از این الگوریتم بودند که در ریختن و بروز کردن فرمون با یکدیگر تفاوت دارند.



شکل ۲. A: مورچه‌های واقعی که بین لانه و منبع غذایی حرکت می‌کنند.B: یک مانع بر روی مسیر بوجود می‌آیدC: مورچه‌ها با احتمال یکسان مسیرهای چپ و راست را انتخاب می‌کنند.D: فرمول بیشتر روی مسیر کوتاه‌تر سبب می‌شود که مورچه‌ها با احتمال بیشتری آن راانتخاب کنند.

۴ الگوریتم اصلاحی نمونه مورچگان

اولین بهبود از الگوریتم AS، استراتژی نخبه‌گرایی AS_{elite} بود که در سال ۱۹۹۶ به وسیله دوریگو، منیزو و کلورنی معرفی شد [۱۶]. این الگوریتم به دلیل معرفی مکانیسم‌هایی جدید از لحاظ کارایی نسبت به AS به بهبودهایی دست یافته بود. باید توجه کرد که روند استراتژی نخبه بدین صورت است که یکی از مورچه‌ها، بهترین مورچه‌ای که تاکنون توانسته است بهترین جواب را به دست آورد، بتواند در هر تکرار مسیر خود را دوباره فرمون‌ریزی کند و یا به عبارت دیگر در هر تکرار بهترین جواب تا سفر حاضر مورد تشویق قرار گیرد. این کار باعث می‌شود تا حدودی کارایی الگوریتم بهتر شود زیرا در روش AS تمام مورچه‌ها بعد از مدتی تورهای یکسانی تولید می‌کردند و این کار باعث می‌شد که اگر این تور مطلوب نباشد عملاً الگوریتم توانایی خود را برای یافتن جواب بهتر از دست داده و روند جستجو و استخراج جواب متوقف شود. در نتیجه الگوریتم برای مسایل نسبتاً بزرگ، کارایی خود را از دست می‌دهد.

تفاوت عمده‌ای که AS_{elite} نسبت به الگوریتم AS دارد عبارت است از:

فرض کنید که S_{gb} بهترین جوابی است که تاکنون در اجرای کامل الگوریتم به دست آمده است بنابراین در هنگام به‌روزرسانی فرمون، مسیر طی شده به وسیله مورچه‌ای که مسیر S_{gb} را تولید کرده است یک مقدار اضافی از فرمون با مقدار $e.\Delta\tau_{ij}^{gb}(t)$ را جذب می‌کند. بنابراین معادله بروز رسانی فرمون به صورت (۹) اصلاح می‌شود.

$$\tau_{ij}(t+1) = (1-\rho).\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) + e.\Delta\tau_{ij}^{gb}(t) \qquad (۹)$$

که در آن

e: وزن تور S_{gb} بوده و دارای یک مقدار ثابت است که توسط کاربر تعیین می‌شود.

$L^k(t)$: طول توری است که مورچه kام پیموده است.

ρ : نرخ تبخیر ثابت در دامنه [۰،۱] می‌باشد که کاربر بدین وسیله کاهش فرمون روی یال‌ها را تنظیم می‌کند.

$$\Delta\tau_{ij}^{\ k}(t)=\begin{cases}1/L^k(t) & (i,j)\in T^k\\ 0 & (i,j)\in T^k\end{cases} \qquad (۱۰)$$

T^k: مجموعه یال‌هایی که توسط مورچه kام مورد ملاقات قرار گرفته است.

$\Delta\tau_{ij}^{\ k}(t)$: مقدار فرمون‌ریزی محلی است که در آن مورچه‌ها در حین حرکت مابین هر گره i و j، مقداری فرمون، به اندازه عکس اندازه‌ای که هر کدام تاکنون پیموده‌اند، را روی یال مربوطه می‌ریزند.

$\Delta\tau_{ij}^{\ gb}(t)$: مقدار فرمون سراسری است که روی یال‌های متعلق به بهترین تور تاکنون شناخته شده ریخته می‌شود و برابر $1/f(s_{gb})$ می‌باشد که در آن $f(s_{gb})$ مقدار طول بهترین مسیر به دست آمده است.

از طرف دیگر لازم به ذکر است که ساختار الگوریتم AS_{elite} به نحوی است که دقت جواب‌ها در ابتدا بسیار پایین است اما به تدریج با افزایش تکرارها و ریختن فرمون توسط مورچه‌ها دقت جواب‌ها افزایش پیدا می‌کند. بنابراین ضریب ثابت e نمی‌تواند ضریب مناسبی برای تشویق بهترین مسیر تاکنون به دست آمده، باشد زیرا تفاوتی در اینکه بهترین جواب در چه تکراری و با چه دقتی به دست آمده، نمی‌گذارد. در روند بهبود AS_{elite} مشخص شد که استفاده از تابع چند جمله‌ای k^2، k عددی صحیح و در ابتدا دارای مقدار ۱ است و هر زمانی که الگوریتم جوابی بهتر را به دست آورد یک واحد افزایش پیدا می‌کند، سبب می‌شود که نتایج بهتری برای الگوریتم به دست آید که در بخش بعدی به تفصیل شرح داده می‌شود.

علاوه بر این، این تابع چند جمله‌ای از آن جهت انتخاب نسبتاً مناسبی به نظر می‌آید که اولاً یک تابع صعودی است و ثانیاً دارای یک شیب متغیر است، به عبارت دیگر با افزایش تکرارها و به دست آمدن جواب‌های بهتر، تابع مربوطه رشد کرده و بهترین مسیر به دست آمده با قدرت بیشتری نسبت به جواب‌های قبلی تشویق می‌شود. توجه به این نکته ضروری است که مقدار کم تابع در ابتدای الگوریتم باعث می‌شود که فرمون ریخته شده بر روی بهترین مسیر تأثیر کمتری بر روی انتخاب مسیرها در تکرارهای بعدی الگوریتم داشته باشد و مورچه‌ها نوعی از فراموشی را به اجرا بگذارند، به عبارت دیگر این کار باعث می‌شود که اگر مورچه‌ها جواب‌های ضعیفی را در ابتدای الگوریتم به دست آوردند، این جواب‌ها را به فراموشی بسپارند، اما با گذشت زمان که الگوریتم به جلو می‌رود و جواب‌ها با دقت بیشتری به دست می‌آیند این مقدار چند جمله‌ای نیز به سرعت افزایش پیدا کرده و یال‌های متعلق به بهترین جواب، فرمون بیشتری را جذب کنند.

۵ تحلیل حساسیت پارامترها

برای یافتن مقادیر بهینه برای پارامترهای روش پیشنهادی، چندین نمونه از مسایل مسیریابی وسیله نقلیه از کتابخانه مساله فروشنده دوره‌گرد (TSPLIB) در نظر گرفته شده [۱۷] و مقادیر تقاضا برای هر کدام از مشتری‌ها حذف می‌شوند تا این مثال‌ها تبدیل به مثال‌هایی قابل استفاده برای مساله چندین فروشنده دوره‌گرد شوند. خصوصیات این مثال‌ها در جدول ۱ نشان داده شده است. در این جدول به ترتیب ستون‌های اول و دوم مثال‌های مورد آزمایش به همراه تعداد گره‌های هر کدام از مثال‌ها (n) را نشان می‌دهد در حالی که ستون‌های سوم و

چهارم تعداد فروشنده‌های هر کدام از مثال‌ها (m) به همراه تعداد مورچه‌های مورد استفاده در الگوریتم (r) را ارایه می‌کند. باید توجه کرد که در این مثال‌ها تعداد مورچه‌های مورد استفاده برابر تعداد فروشنده‌های دوره‌گرد در نظر گرفته شده است. سرانجام ستون پنجم تعداد اجرای الگوریتم را به طور مستقل نشان می‌دهد (t) که در آن بهترین جواب به دست آمده به وسیله الگوریتم به عنوان جواب نهایی نشان داده می‌شود.

جدول ۱. خصوصیات مسایل

نمونه	تعداد مشتری‌ها	تعداد فروشنده دوره‌گرد	تعداد مورچه‌های مورد استفاده	تعداد اجرای الگوریتم
E-۰۲۳-۰۳	۲۲	۳	۳	۱۰
E-۰۳۱-۰۹	۳۰	۹	۹	۱۰
E-۰۴۱-۱۴	۴۰	۱۴	۱۴	۱۰
E-۰۴۵-۰۴	۴۴	۴	۴	۱۰
E-۰۵۱-۰۵	۵۰	۵	۵	۱۰
E-۰۷۶-۰۷	۷۵	۷	۷	۱۰
E-۱۰۱-۰۸	۱۰۰	۸	۸	۱۰

برای بررسی تأثیر پارامترهای روش الگوریتم مورچگان بر روی جواب‌های مسایل، سه پارامتر این الگوریتم به طور مجزا در نظر گرفته شده و با ثابت گرفتن بقیه پارامترها مورد تجزیه و تحلیل قرار می‌گیرند. باید توجه کرد که α و β به ترتیب توان‌های میزان تأثیر فرمون و اطلاعات ابتکاری مساله در فرمول (۶) و ω میزان تکرار بهترین جواب الگوریتم در یک‌بار تکرار الگوریتم می‌باشند.

در جدول ۲ تغییرات پارامتر α نشان داده شده است. در این جدول مقدار پارامتر β برابر ۴ و تعداد تکرار بهترین جواب الگوریتم، به عنوان شرط پایانی، برابر ۱۲ در نظر گرفته شده است. همان‌طور که مشخص است بهترین جواب‌ها برای مقادیر مختلف α برای مقدار ۱ به دست آمده است. همچنین اگر جواب‌های مسایل برای مقادیر آلفای بزرگ‌تر یا مساوی یک در نظر گرفته شود آن گاه رابطه معناداری بین افزایش مقدار α و کیفیت جواب‌های به دست آمده به وجود دارد. در این مسایل هر چه مقدار آلفا افزایش پیدا می‌کند مقادیر جواب‌ها نیز افزایش پیدا می‌کند. بنابراین با افزایش α در ۷ مثال مربوطه، کیفیت جواب‌ها کاهش می‌یابد.

جدول ۲. جواب‌های الگوریتم برای آلفاهای متفاوت

نمونه	$\alpha = 0.4$	$\alpha = 1$	$\alpha = 1.6$	$\alpha = 2.2$	$\alpha = 2.8$
E-۰۲۳-۰۳	۵۱۷,۴۲	**۵۱۶,۴۲**	۵۱۹,۶۵	۵۳۱,۴۵	۵۵۰,۲۴
E-۰۳۱-۰۹	۴۴۱,۵۲	**۴۳۹,۹۶**	۴۴۵,۳۴	۴۴۹,۵۲	۴۶۸,۲۵
E-۰۴۱-۱۴	۷۲۵,۹۶	**۷۲۱,۹۹**	۷۲۹,۲۵	۷۴۵,۱۸	۷۶۹,۳۶
E-۰۴۵-۰۴	۶۴۴,۹۷	۶۴۴,۲۱	**۶۴۳,۱۵**	۶۵۸,۶۹	۶۶۴,۲۵
E-۰۵۱-۰۵	۴۷۶,۲۱	**۴۷۳,۵۲**	۴۷۶,۳۸	۴۷۹,۲۴	۴۸۷,۹۸
E-۰۷۶-۰۷	۶۲۵,۳۱	۶۱۸,۴۸	**۶۱۶,۱۷**	۶۱۹,۶۸	۶۳۶,۱۴
E-۱۰۱-۰۸	۷۷۱,۲۸	**۷۵۸,۱۲**	۷۶۰,۱۲	۷۶۵,۹۸	۸۰۲,۶۴

در جدول ۳ میزان تغییرات پارامتر β نشان داده شده است. در اینجا مقدار α برابر ۱، به علت به دست آمدن بهترین جواب‌ها در جدول ۲، و مقدار تکرار بهترین جواب الگوریتم برابر ۱۲ در نظر گرفته شده است. همچنین در این جدول مقدار تغییر پارامتر β ما بین ۰٫۲ تا ۳٫۲ در نظر گرفته شده است. باید توجه کرد با تنظیم مقدار β به اندازه ۳٫۲ بهترین جواب‌ها و به اندازه ۱ بی‌کیفیت‌ترین جواب‌ها به دست آمده است. همچنین هر چه مقدار بتا افزایش می‌یابد. تقریباً کیفیت جواب‌های الگوریتم با یک شیب صعودی افزایش پیدا می‌کند. بنابراین برعکس پارامتر α، پارامتر β دارای یک رابطه مستقیم با کیفیت جواب‌ها بوده و افزایش آن باعث افزایش کیفیت جواب‌ها می‌شود.

جدول۳. جواب‌های الگوریتم برای بتاهای متفاوت

نمونه	$\beta = 0.2$	$\beta = 1$	$\beta = 1.8$	$\beta = 2.6$	$\beta = 3.2$
E-۰۲۳-۰۳	۸۲۴,۹۴	۵۲۴,۸۴	۵۲۰,۴۵	۵۱۸,۲۱	**۵۱۵,۲۴**
E-۰۳۱-۰۹	۴۴۶,۱۲	۴۴۵,۵۴	۴۴۵,۳۱	۴۴۱,۹۸	**۴۴۰,۰۱**
E-۰۴۱-۱۴	۷۲۳,۴۵	۷۲۵,۱۴	۷۲۱,۶۴	**۷۱۹,۹۵**	۷۲۰,۱۲
E-۰۴۵-۰۴	۶۶۲,۴۵	۶۶۰,۸۷	۶۵۳,۱۲	۶۴۷,۵۴	**۶۴۲,۶۴**
E-۰۵۱-۰۵	۵۰۳,۱۲	۵۰۱,۹۶	۴۸۹,۳۴	۴۸۰,۲۱	**۴۷۵,۹۶**
E-۰۷۶-۰۷	۶۵۹,۱۲	۶۵۴,۱۲	۶۳۰,۵۴	**۶۱۵,۵۳**	۶۱۶,۵۴
E-۱۰۱-۰۸	۸۳۲,۱۵	۸۱۲,۱۳	۷۹۶,۱۵	۷۶۵,۴۲	**۷۵۷,۵۵**

نهایتاً در جدول ۴ میزان تغییرات تکرارهای بهترین جواب الگوریتم، که شرط اصلی برای پایان الگوریتم است، نشان داده شده است. در این جدول تعداد تکرارهای بهترین جواب در بازه‌ای بین ۳ تا ۲۰ در نظر گرفته شده است. همچنین برای این جدول $\alpha = 1$ و $\beta = 3.2$ در نظر گرفته شده است. نتایج الگوریتم در این جدول به این نکته اشاره دارد که علیرغم اینکه تصور می‌شود که تکرار بیشتر بهترین جواب الگوریتم در مدت طولانی سبب به دست آمدن جواب‌های بهتر می‌شود، اما در اینجا این فرضیه صادق نمی‌باشد. در این جدول برای سه مثال ۰۳-E-۰۲۳، E-۰۴۱-۱۴ و ۰۵-۰۵۱- E مقدار $\omega = 10$، برای مثال E-۰۴۵-۰۴ مقدار $\omega = 7$ و برای بقیه مثال‌ها مقدار $\omega = 20$ بهترین جواب‌ها را پیدا کرده‌اند. علت این امر را باید در تعداد زیاد متغیرها و پارامترهای زیادی که در الگوریتم مورچگان وجود دارد جستجو کرد. بنابراین برای به دست آوردن جواب‌های باکیفیت‌تر برای این مساله به وسیله الگوریتم مورچگان بهتر است که تعداد انجام آزمایش الگوریتم را افزایش داد تا اینکه تعداد تکرار بهترین جواب الگوریتم زیاد شود. بنابراین میزان نرخ بهینه برای پارامترهای β, α و ω به ترتیب ۱، ۳٫۲ و ۱۰ در نظر گرفته می‌شود زیرا زمان مصرفی برای $\omega = 10$ کمتر از $\omega = 20$ است.

جدول ۴. جواب‌های الگوریتم برای ω های متفاوت

نمونه	۳	۷	۱۰	۱۵	۲۰
E-۰۲۳-۰۳	۵۲۰,۱۴	۵۱۶,۶۴	**۵۱۴,۹۸**	۵۱۵,۱۴	۵۱۶,۳۴
E-۰۳۱-۰۹	۴۴۲,۲۵	۴۳۹,۶۸	۴۴۲,۸۵	۴۴۰,۴۵	**۴۳۹,۲۴**
E-۰۴۱-۱۴	۷۲۵,۲۱	۷۲۰,۲۱	**۷۱۹,۹۴**	۷۲۱,۵۴	۷۲۰,۰۰
E-۰۴۵-۰۴	۶۵۲,۱۴	**۶۴۲,۱۵**	۶۴۲,۲۵	۶۴۴,۵۶	۶۴۵,۹۷
E-۰۵۱-۰۵	۴۷۹,۱۲	۴۷۵,۹۹	**۴۷۵,۹۳**	۴۷۶,۰۱	۴۷۷,۷۹
E-۰۷۶-۰۷	۶۲۵,۱۴	۶۲۰,۵۴	۶۱۶,۹۷	۶۱۵,۹۶	**۶۱۵,۷۳**
E-۱۰۱-۰۸	۷۶۹,۱۵	۷۵۹,۶۴	۷۵۷,۹۸	۷۵۹,۱۲	**۷۵۷,۴۵**

۶ نتایج محاسباتی

برای بررسی کارایی الگوریتم پیشنهادی تعدادی از مثال‌های استاندارد مساله چندین فروشنده دوره‌گرد در نظر گرفته شده است تا بدین وسیله مقایسه کاملی بین الگوریتم اصلاحی، الگوریتم قدیمی و دو نوع از الگوریتم‌های فراابتکاری انجام شود. این دسته از مثال‌ها از کتابخانه مساله فروشنده دوره‌گرد در نظر گرفته شده است با این تفاوت که تعداد فروشنده دوره‌گرد در هر کدام از مثال‌ها برابر ۵ در نظر گرفته می‌شود. همچنین باید اضافه کرد که تمام کدهای الگوریتم پیشنهادی به زبان $C++$ نوشته شده است و کامپیوتری که این برنامه‌ها بر روی آن اجرا شده از نوع پنتیوم ۴ با قدرت $3GHz$ و با یک گیگا بایت حافظه می‌باشد.

این دسته از مثال‌ها که شامل *pr*۷۶، *pr*۱۵۲، *pr*۲۲۶، *pr*۲۲۹، *pr*۴۳۹ و *pr*۱۰۰۲ می‌باشند دارای بازه بسیار خوبی از تعداد گره‌ها می‌باشند که مقادیر بین ۷۶ تا ۱۰۰۲ را اختیار می‌کنند. از طرف دیگر بهترین مقدار پیدا شده برای هر کدام از الگوریتم‌ها در ۱۰ تکرار الگوریتم در جدول ۵، میانگین مقادیر پیدا شده برای هر کدام از مثال‌ها در جدول ۶ و در نهایت مقدار زمان مصرفی برای یافتن جواب‌ها برای هر کدام از الگوریتم‌ها در جدول ۷ نشان داده شده است.

در جدول ۵ خصوصیات کامل مثال‌های مورد آزمایش و همچنین نتایج الگوریتم‌های فراابتکاری برای این مثال‌ها آورده شده است. در این جدول همانند جدول ۱، ستون دوم نشان دهنده تعداد گره‌های هر مثال (m) می‌باشد درحالی که ستون سوم تعداد فروشنده‌های مورد استفاده برای هر مثال (n) را ارایه می‌کند. برای اینکه یک تعادل بین تعداد گره‌های ملاقات شده توسط هر فروشنده برقرار باشد، برای حداکثر تعداد گره‌های قابل ملاقات برای هر فروشنده مقداری تعیین شده است. ستون چهارم در جدول ۵ نشان دهنده همین تعداد یعنی حداکثر گره‌های قابل ملاقات توسط هر فروشنده می‌باشد (l). از طرف دیگر ستون‌های پنجم و ششم جواب‌های الگوریتم پیشنهادی برای این مساله را ارایه می‌کنند. باید توجه کرد که در ستون پنجم الگوریتم نمونه مورچگان معمولی برای مساله ارایه شده است در حالی که در ستون ششم الگوریتم پیشنهادی یعنی الگوریتم اصلاحی نمونه مورچگان برای مساله نشان داده شده است. برای اینکه یک مقایسه کامل بین الگوریتم پیشنهادی و الگوریتم‌های دیگر فراابتکاری صورت گیرد در ستون‌های هفتم و هشتم به ترتیب الگوریتم‌های اصلاحی ژنتیک (MGA) و

اصلاحی مورچگان ($MACO$) از ادبیات موضوع آورده شده است. باید توجه کرد که الگوریتم اصلاحی ژنتیک توسط تانگ و همکارش ارایه شده [۱۸] در حالی که الگوریتم اصلاحی مورچگان توسط جانجیل و دینگوی پیشنهاد شده است [۱۹].

با مقایسه نتایج به دست آمده در جدول ۵ می‌توان نتیجه گرفت که الگوریتم پیشنهادی نسبت به الگوریتم نمونه مورچگان معمولی از کارایی بسیار خوبی برخوردار است به طوری که توانسته است به جز مثال $pr۲۲۹$ در بقیه مثال‌ها کیفیت مقادیر جواب‌ها را به مقدار قابل قبولی افزایش دهد. البته شایان به ذکر است که الگوریتم نمونه مورچگان نیز در مقایسه با دو الگوریتم دیگر دارای رقابت خوبی بوده و به طور متوسط با مقدار ۱۸۶۲۶۰,۷ برای شش مثال در مقایسه با مقدار ۱۸۹۲۳۲,۳ برای الگوریتم ژنتیک اصلاحی و مقدار ۱۸۳۹۰۹,۲ برای الگوریتم اصلاحی مورچگان، جواب‌های باکیفیتی را ارایه کرده است. بنابراین الگوریتم نمونه معمولی دارای مقادیری بهتر نسبت به الگوریتم ژنتیک اصلاحی است اما در مقایسه با الگوریتم اصلاحی مورچگان نتوانسته است که برتری خود را به اثبات برساند و به جواب‌های بی‌کیفیت‌تری رضایت داده است. از طرف دیگر با مقدار متوسط ۱۷۹۸۸۶,۵ که به وسیله الگوریتم پیشنهادی به دست آمده است می‌توان نتیجه گرفت که این الگوریتم اصلاحی از قدرت بسیار خوبی برای فرار از نقاط بهینه محلی برخوردار بوده و بدین علت توانسته است که بهترین جواب‌ها را برای این مثال‌ها به دست آورد. باید توجه کرد که از نظر قدرت یافتن بهترین جواب‌ها می‌توان الگوریتم‌های جدول ۵ را به ترتیب از ضعیف به قوی به صورت الگوریتم اصلاحی ژنتیک، الگوریتم نمونه مورچگان معمولی، الگوریتم اصلاحی مورچگان و الگوریتم نمونه مورچگان اصلاح شده ترتیب‌بندی نمود.

جدول ۵. مقایسه بین الگوریتم‌های فراابتکاری

نمونه	n	v	l	AS_{elite}	MAS_{elite}	MGA	$MACO$
Pr۷۶	۷۶	۵	۲۰	۱۶۳۲۵۷	۱۵۶۹۸۲	۱۵۷۴۴۴	۱۷۸۵۹۷
Pr۱۵۲	۱۵۲	۵	۴۰	۱۴۳۲۳۱	۱۳۰۸۳۲	۱۲۷۸۳۹	۱۳۰۹۵۳
Pr۲۲۶	۲۲۶	۵	۵۰	۱۶۷۵۱۲	۱۶۷۴۳۹	۱۶۶۸۲۷	۱۶۷۶۴۶
Pr۲۹۹	۲۹۹	۵	۷۰	۸۲۲۳۱	۸۲۳۲۱	۸۲۱۷۶	۸۲۱۰۶
Pr۴۳۹	۴۳۹	۵	۱۰۰	۱۶۳۸۷۲	۱۶۰۹۸۹	۱۷۳۸۳۹	۱۶۱۹۵۵
Pr۱۰۰۲	۱۰۰۲	۵	۲۲۰	۳۸۵۴۶۱	۳۸۰۷۵۶	۴۲۷۲۶۹	۳۸۲۱۹۸

جدول ۶ نشان دهنده مقایسه الگوریتم‌های ذکر شده از جنبه‌ای دیگر است. در این جدول مقادیر متوسط به دست آمده برای هر الگوریتم نشان داده شده است. چون برای یافتن بهترین جواب برای هر مثال طبق روش گفته شده الگوریتم ده بار مورد آزمایش قرار می‌گیرد بنابراین می‌توان الگوریتم‌ها را از جهتی دیگر نیز مورد مقایسه قرار داد. نتایج این جدول نیز حاکی است که الگوریتم در مقایسه با نسخه عادی خود دارای پیشرفت خوبی بوده است و در هر ۶ مثال، مقدار به دست آمده از کیفیت بالاتری برخوردار است. نکته قابل توجهی که وجود دارد این است که با وجود آن که مقادیر به دست آمده برای الگوریتم نمونه معمولی برای بهترین جواب در جدول ۵ دارای کیفیت کمتری نسبت به الگوریتم $MACO$ است اما از نظر متوسط مقادیر الگوریتم برای ۱۰ بار آزمایش،

الگوریتم نمونه معمولی توانسته است که کارایی بیشتری از خود نشان دهد و به جواب‌های باکیفیت‌تری دست پیدا کند. در آخر باید توجه کرد که نتایج به دست آمده در جدول ۶ برای الگوریتم پیشنهادی نیز به این نکته تاکید دارد که کارایی الگوریتم در مقایسه با سه الگوریتم دیگر از کیفیت بیشتری برخوردار است.

جدول۶. متوسط مقادیر برای ۱۰ بار اجرا الگوریتم‌ها

نمونه	AS_{elite}	MAS_{elite}	MGA	$MACO$
Pr۷۶	۱۵۷۳۴۲	۱۵۷۱۲۱	۱۶۰۵۷۴	۱۸۰۶۹۰
Pr۱۵۲	۱۳۵۹۸۲	۱۳۲۶۷۱	۱۳۳۳۳۷	۱۳۶۳۴۱
Pr۲۲۶	۱۷۱۲۶۵	۱۶۹۷۴۵	۱۷۸۵۰۱	۱۷۰۸۷۷
Pr۲۹۹	۸۳۱۱۱	۸۲۹۹۴	۸۵۷۹۶	۸۳۸۴۵
Pr۴۳۹	۱۶۲۹۷۳	۱۶۲۷۷۴	۱۸۳۶۹۸	۱۶۵۰۳۵
Pr۱۰۰۲	۳۸۸۷۴۱	۳۸۲۰۰۵	۴۵۹۱۷۹	۳۸۷۲۰۵

در جدول ۷ میزان زمان مصرفی برای یافتن جواب هر الگوریتم برای مثال‌های داده شده، آورده شده است. باید توجه کرد که این نتایج بر حسب ثانیه می‌باشند. همچنین توجه به این نکته ضروری است که شرط پایان الگوریتم در دو روش پیشنهادی برای این مساله ۱۰ بار تکرار شدن بهترین جواب در حین اجرای الگوریتم است. با توجه به نتایج به دست آمده در جدول ۷ می توان نتیجه گرفت که از لحاظ زمان مصرفی، الگوریتم نمونه مورچگان معمولی دارای بهترین جواب است و بعد از آن الگوریتم نمونه مورچگان اصلاحی، الگوریتم اصلاحی مورچگان و الگوریتم اصلاحی ژنتیک قرار دارند. نکته قابل توجهی که در این جدول وجود دارد در این است که علیرغم اینکه الگوریتم پیشنهادی دارای جواب‌های بهتر نسبت به الگوریتم نمونه مورچگان معمولی است اما از نظر طول زمان اجرا دارای مقدار بیشتری می‌باشد. البته این نتیجه غیر قابل پیش‌بینی نبود زیرا بر اساس توضیحات ذکر شده با تغییر مقدار تشویق ضریب بهترین مسیر، الگوریتم دارای کیفیت بیشتری برای فرار از نقاط بهینه محلی می‌شود و بدین ترتیب فضای جواب مساله با گستردگی بیشتری مورد تجزیه و تحلیل قرار می گیرد. این عمل سبب می‌شود که علیرغم اینکه زمان بیشتری صرف جستجو در فضای جواب مساله شود، الگوریتم بتواند به جواب‌های بهتری دست پیدا کند.

جدول۷. مقایسه زمان اجرای الگوریتم‌ها فراابتکاری

نمونه	AS_{elite}	MAS_{elite}	MGA	$MACO$
Pr۷۶	۲۷	۳۲	۴۳	۵۱
Pr۱۵۲	۷۸	۷۵	۹۱	۱۲۸
Pr۲۲۶	۱۲۵	۱۳۶	۱۶۵	۱۴۳
Pr۲۹۹	۲۲۴	۲۵۶	۳۶۳	۲۸۸
Pr۴۳۹	۴۰۵	۴۶۳	۶۲۳	۵۶۳
Pr۱۰۰۲	۱۷۹۰	۲۰۰۹	۲۸۹۲	۲۶۲۰

۷ نتیجه‌گیری و جهت‌گیری آینده

در این مقاله ابتدا یک فرمول‌بندی بر اساس مساله فروشنده دوره‌گرد برای مساله چندین فروشنده دوره‌گرد پیشنهاد شد و سپس الگوریتم اصلاحی نمونه مورچگان برای حل مساله مورد نظر مورد استفاده قرار گرفت که در چگونگی تشویق بهترین مسیر با الگوریتم نمونه مورچگان معمولی تفاوت داشت. در اینجا از تابعی استفاده شد که تا حد ممکن از همگرایی زودرس جلوگیری می‌کرد و باعث می‌شد که الگوریتم بتواند تا حد امکان از افتادن در بهینه‌های محلی نجات یابد. در نتیجه اگر چه الگوریتم زمان بیشتری را برای یافتن جواب‌های بهتر صرف می‌کرد اما توانست به جواب‌های با کیفیت‌تری دست پیدا کند. به نظر می‌رسد که توابع بهتری از این تابع پیشنهادی برای این ضریب وجود داشته باشد و همچنین این الگوریتم اصلاحی را بتوان برای مسایل دیگر بهینه‌سازی ترکیباتی همچون مساله مسیریابی وسیله نقلیه یا گسترش‌های آن به کار برد. از طرف دیگر ترکیب این الگوریتم با روش‌های فراابتکاری دیگر مانند ژنتیک، باز پخت شبیه‌سازی یا جستجوی ممنوع راهکار مناسب دیگری برای بهبود الگوریتم و یافتن جواب‌های باکیفیت‌تر می‌باشد که کاربردی کردن این پیشنهادها به مقاله‌های بعدی موکول می‌شود.

قدردانی و تشکر

این مقاله نتیجه انجام یک طرح پژوهشی است که هزینه های اجرای این طرح توسط باشگاه پژوهشگران جوان دانشگاه آزاد اسلامی واحد همدان تأمین و تخصیص یافته است.

منابع

[1] Bektas, T., (2006). The multiple traveling salesman problems: an over-view of formulations and solution procedures. Omega, 34, 209-219.

[2] Yadlapalli, S., Malik, W.A., Darbhaa, S., Pachter, M., (2009). A Lagrangian-based algorithm for a Multiple Depot, Multiple Traveling Salesmen Problem. Nonlinear Analysis: Real World Applications 10(4), 1990-1999.

[3] Gavish, B., Srikanth, K., (1986). An optimal solution method for large-scale multiple traveling salesman problems. Operations Research, 34(5), 698–717.

[4] Gromicho, J., Paixao, J., Branco, I., (1992). Exact solution of multiple traveling salesman problems. Springer.

[5] AI, A., Kennington, J. L., (1986). Exact solution of multiple traveling salesman problems. Discrete Applied Mathematics, 13, 259–276.

[6] Malik, W., Rathinam, S., Darbha, S., (2007). An approximation algorithm for a symmetric Generalized Multiple Depot, Multiple Travelling Salesman Problem. Operations Research Letters, 35(6), 747-753.

[7] Zhanga, T., Yin, Y., Li, J., (2010). An improved approximation algorithm for the maximum TSP. Theoretical Computer Science, 411, 2537-2541.

[8] Potvin, J., Lapalme, G., Rousseau, J., (1989). A generalized k-opt exchange procedure for the mtsp. INFOR, 21, 474–481.

[9] Russell, R.A., (1977). An effective heuristic for the m-tour traveling salesman problem with some side conditions. Operations Research, 25(3), 517–524.

[10] Masutti, T. A. S., Castrom L. N., (2009). Neuro-immune approach to solve routing problems. Neurocomputing, 72(10-12), 2189-2197.

[11] Ryan, J. L., Bailey, T. G., Moore, J. T., Carlton, W. B., (1998). Reactive Tabu search in unmanned aerial reconnaissance simulations. Proceedings of the 1998 winter simulation conference 1, 873–879.

[12] Yousefi Khoshbakht, M., Zafari, A., (2008). A new ant colony algorithm for solving multiple traveling salesman problem. The 2th Joint Congress on Intelligent and Fuzzy Systems (ISFS2008), 28-30 October, Malek-Ashtar University of Technology, Tehran, Iran.
[13] Ghafurian, S., Javadian, N., (2011). An ant colony algorithm for solving fixed destination multi-depot multiple traveling salesmen problems. Applied Soft Computing, 11(1), 1256-1262.
[14] Paydar, M. M., Mahdavi, I., Sharafuddin, I., Solimanpur, M., (2010). Applying simulated annealing for designing cellular manufacturing systems using MDmTSP. Computers & Industrial Engineering, 59(4), 929-936.
[15] Dorigo, M., (1992). Optimization, Learning and natural algorithms. Ph.D Thesis, Dip.Electtronica e Informazion, Politecnico di Milano Italy.
[16] Dorigo, M., Maniezzo, V., Colorni, A., (1996). The ant system: optimization by a colony of cooperating agents. IEEE Trans on Sys 26.
[17] http://www.or.deis.unibo.it/research pages/orinstances/vrplib/vrplib.html
[18] Tang, T., Liu, J., (2000). Multiple traveling salesman problem model for hot rolling scheduling in Shanghai Baoshan Iron & Steel Complex. European Journal of Operational Research, 24, 267- 282.
[19] Junjie, P., Dingwei, W., (2006). An ant colony optimization algorithm for multiple travelling salesmen Problem. In ICICIC ’06: Proceedings of the First International Conference on Innovative Computing, Information and Control, Washington, DC, USA, IEEE Computer Society, 210–213.