

یک روش بهینه‌سازی ترکیبی برای حل مساله چندین فروشنده دوره گرد

مجید یوسفی خوشبخت*

دانشگاه آزاد اسلامی، واحد همدان، باشگاه پژوهشگران جوان و نخبگان، همدان، ایران

چکیده

این مقاله یک الگوریتم اصلاحی ژنتیک برای حل مساله چندین فروشنده دوره گرد را ارائه می‌کند. در اینجا برای این که جواب‌های ابتدایی دارای کیفیت و تنوع خوبی باشند، جواب‌های اولیه الگوریتم ژنتیک به وسیله الگوریتم نمونه مورچگان تولید می‌شود. به علاوه در هر زمان از اجرای الگوریتم برای جلوگیری از کاهش تنوع جواب‌ها تعدادی از کروموزوم‌ها بصورت تصادفی انتخاب می‌گردد و در نهایت از دو الگوریتم جهش متفاوت، که احتمال اجرای هر کدام در هر تکرار برابر ۰/۵ است، برای ارتقای جواب‌ها استفاده می‌شود. نتایج بدست آمده روی چندین مثال استاندارد، کارایی الگوریتم جدید را نسبت به الگوریتم‌های فراابتکاری دیگر نشان می‌دهد.

کلمات کلیدی: الگوریتم ژنتیک، الگوریتم نمونه مورچگان، مساله چندین فروشنده دوره گرد، همگرایی زودرس.

۱ مقدمه

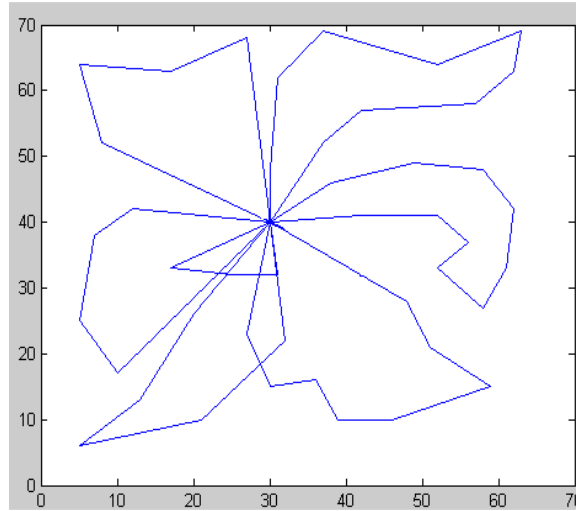
مساله چندین فروشنده دوره گرد (MTSP) یکی از مهم‌ترین گسترش‌های مساله فروشنده دوره گرد (TSP) است که دارای کاربردهای فراوانی در مسایل بهینه‌سازی ترکیباتی بوده و می‌تواند برای مدل‌بندی بسیاری از مسایل عملی مورد استفاده قرار گیرد. این مساله که دارای تشابه بسیار زیادی با مساله TSP است به صورت زیر تعریف می‌شود.

فرض کنید که گراف غیرجهت‌دار کامل $G = (V, A)$ دارای مجموعه گره $V = \{0, 1, \dots, n\}$ و مجموعه کمان $A = \{(i, j) : i, j \in V, i \neq j\}$ باشد (اگر گراف کامل نبود آنگاه هر یال وجود نداشته را به وسیله یالی با اندازه مثبت بی‌نهایت جای‌گزین کنید). در این مساله اگر $C = (c_{ij})_{(n+1) \times (n+1)}$ نشان‌دهنده ماتریس هزینه روی گراف G باشد، هدف کمینه‌کردن هزینه برای $m > 1$ فروشنده است که از گره انبار، به عنوان مکان شروع، حرکت می‌کنند و بعد از ملاقات کردن هر گره دیگر به آن باز می‌گردند (شکل ۱) به شرط آن که:

* عهده دار مکاتبات

آدرس الکترونیکی: khoshbakht@iauh.ac.ir

- هر گره فقط به وسیله یک فروشنده مورد ملاقات قرار گیرد.
- هر فروشنده حداقل یک گره را مورد ملاقات قرار دهد.



شکل ۱. یک جواب شدنی مسأله MTSP

همان طوری که مشاهده می‌کنید در این مسأله n گره باید به m تور طور تقسیم‌بندی شود به طوری که قوانین مسأله TSP برای هر تور برقرار باشد. به علاوه مسأله $MTSP$ به علت این که هر گره ممکن است به وسیله هر فروشنده دلخواه مورد ملاقات قرار گیرد، دارای پیچیدگی بیشتر از مسأله TSP می‌باشد. از طرف دیگر این مسأله یک آزمون استاندارد برای الگوریتم‌هایی است که به طور موازی کار می‌کنند و کارایی این الگوریتم‌ها در این مسأله به نحوی کارایی آن‌ها را در مسایل دیگر نشان می‌دهد. اهمیت این مسأله فقط به این موضوع محدود نبوده و از چندین جهت دیگر نیز حائز اهمیت است که بعضی از آن‌ها عبارت است از:

۱- وجود مسایل گسترش یافته از آن مانند:

- مسأله مسیریابی چندین وسیله نقلیه که در آن چندین وسیله نقلیه با ظرفیت‌های یکسان یا غیر یکسان از انبار شروع به حرکت کرده و بعد از ملاقات کردن بقیه گره‌ها، هر گره توسط فقط یک وسیله نقلیه مورد ملاقات قرار می‌گیرد، و بارگذاری کالا به انبار باز می‌گردند.
- مسأله زمان‌بندی چندین وسیله نقلیه که در آن مجموعه‌ای از وسایل نقلیه با ظرفیت محدود در چندین انبار وجود دارند. هدف انجام چندین مسافرت است از جایی به جای دیگر است به طوری که هزینه کلی این مسافرت‌ها مینیمم شده و همه محدودیت‌ها رعایت شود [۱].

۲- کاربردهای فراوان این مسأله در مسایل گوناگون مانند:

- مسأله زمان‌بندی کارها در چند خط تولید [۲]
- زمان‌بندی کارکنان کشتی‌ها یا هواپیماها [۳]

- زمان‌بندی چاپ در مطبوعات [۴]
- طراحی ماموریت برای روبات‌های مستقل متحرک [۵]
- زمان‌بندی مصاحبه [۶]
- زمان‌بندی غلطک‌های داغ [۷]

به طور کلی روش‌هایی که برای حل مساله *MTSP* مورد استفاده قرار گرفته شده است را می‌توان به دو دسته دقیق و ابتکاری تقسیم‌بندی کرد که هر کدام دارای معایب و محاسنی هستند.

۱- روش‌های دقیق: در این گونه از روش‌ها، جواب بهینه به دست می‌آید ولی به علت پیچیدگی محاسباتی مساله *MTSP* که از نوع *NP*-تام است، الگوریتم‌های بهینه‌سازی برای حل این مساله با ابعاد بزرگ ناکارآمد بوده و نمی‌توانند در یک زمان قابل قبول به جواب بهینه برسند. به طور مثال می‌توان به روشی که به وسیله لاپرت و نابرت در سال ۱۹۸۰ ارائه شد، اشاره کرد که بر پایه تخفیف بعضی از محدودیت‌های مساله بود [۸]. روش دقیق دیگری که به وسیله گاویش و اسریکانت ارائه شد روش شاخه و کران بود که در سال ۱۹۸۶ مورد توجه قرار گرفت و برای مسایل متقارن با بیش از ۱۰۰ گره به کار برده شد [۹]. همچنین گرومیکو و همکارانش روش دیگری در سال ۱۹۹۲ ارائه دادند که بر اساس روش شبه تخصیص بنا نهاده شده بود. آن‌ها این روش را برای مساله *MTSP* نامتقارن مورد استفاده قرار دادند [۱۰].

۲- روش‌های ابتکاری: این گونه از روش‌ها، که امروزه توجه قابل ملاحظه‌ای را به خود جلب کرده است، همیشه نمی‌توانند جواب بهین مساله را بیابند اما برای مسایل بزرگ قادرند که جواب نزدیک به بهینه را در زمان قابل قبولی به دست آورند. برای نمونه از این الگوریتم‌ها می‌توان به روش شبکه‌های عصبی و اچولدر و همکارانش اشاره کرد که در سال ۱۹۸۹ ارائه شد [۱۱]. در این روش که گسترش روش هافیلد-تانک برای مساله *TSP* بود، روش قادر به دست آوردن جواب‌های شدنی به علت پیچیدگی بالا مساله نبود. برای اطلاع از این روش و سایر اصلاحات صورت گرفته بر روی آن می‌توان به [۱۸-۱۲] مراجعه کرد. روش دیگر فراابتکاری که در سال ۱۹۹۸ ارائه شد، الگوریتم جستجوی ممنوع بود که همراه با یک فرمول‌بندی برنامه‌ریزی خطی صحیح به وسیله رایان و همکارانش ارائه شد [۱۹]. سپس در سال ۱۹۹۹ زانگ و همکارانش الگوریتم ژنتیک (*GA*) را برای حل این مساله ارائه دادند [۲۰] در حالی که در سال ۲۰۰۳ روش بازپخت شبیه‌سازی شده توسط سانگ و همکارانش برای مسایلی با ۴۰۰ گره مورد استفاده قرار گرفت [۲۱]. نهایتاً یوسفی خوشبخت و همکارش در سال ۲۰۱۱ یک روش الگوریتم مورچگان را برای مساله فوق پیشنهاد دادند [۲۲]. در روش مربوطه از الگوریتم نمونه مورچگان (*EAS*) برای حل مساله استفاده شده بود. در این روش اصلاحی، ضریبی جدید برای تشویق بهترین مسیر به دست آمده در هر تکرار الگوریتم ارائه گردید که سبب شد الگوریتم تا حد امکان به جستجوی سراسری به پردازد و با یک روال منظم به سمت جستجوی محلی پیش برود.

در این مقاله ابتدا در بخش ۲ مدل مساله $MTSP$ ارائه می‌شود. سپس روش پیشنهادی در بخش ۳ مورد بررسی قرار می‌گیرد و تحلیل پارامترها در بخش ۴ شرح داده می‌شود. در انتها در بخش ۵ یک مقایسه محاسباتی بین الگوریتم پیشنهادی و الگوریتم‌های دیگر فراابتکاری بر روی مثال‌های استاندارد انجام می‌شود.

۲ مدل بندی مساله

اگر چه مساله $MTSP$ NP -تام بوده و جواب‌های شدنی با افزایش بعد مساله به سرعت افزایش پیدا می‌کند، اما می‌تواند به صورت یک مدل برنامه‌ریزی خطی صحیح نمایش داده شود. برای مدل‌سازی مساله از علامت‌گذاری‌های زیر استفاده می‌شود:

- C که نشان‌دهنده ماتریس هزینه روی گراف G است، یک ماتریس متقارن است و در نامساوی مثلثی صدق می‌کند. یعنی برای هر $(i, j, k) \in V$ ، $c_{ij} = c_{ji}$ و $c_{ij} + c_{jk} \geq c_{ik}$ است.
- اگر برای $(i, j) \in V$ ، $i \neq j$ ، $i, j = 1, \dots, n$ فروشنده به طور مستقیم از i به j حرکت کند $x_{ij} = 1$ است و در غیر این صورت $x_{ij} = 0$ می‌باشد.

بنابراین مدل مساله $MTSP$ که براساس فرمول‌بندی مساله تخصیص می‌باشد عبارت است از:

$$Min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (1)$$

$$\sum_{i=1}^n x_{i.} = m \quad (2)$$

$$\sum_{j=1}^n x_{.j} = m \quad (3)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad j = 1, \dots, n \quad (4)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad i = 1, \dots, n \quad (5)$$

$$\sum_{i \in S} \sum_{j \in N-S} x_{ij} \geq 1 \quad \emptyset \neq S \subset \{1, \dots, n\}, |S| \geq 2 \quad (6)$$

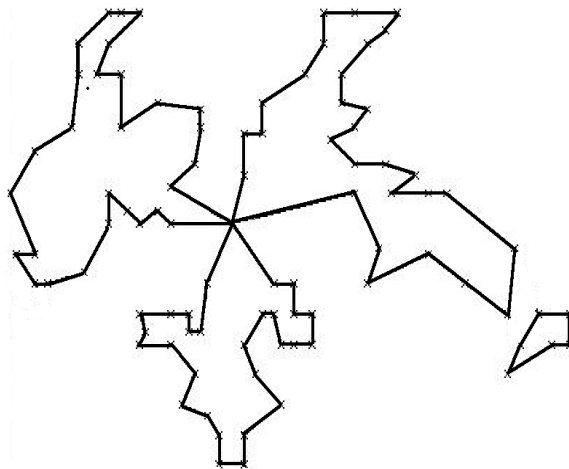
$$\sum_{i \in N-S} \sum_{j \in S} x_{ij} \geq 1 \quad \emptyset \neq S \subset \{1, \dots, n\}, |S| \geq 2 \quad (7)$$

$$x_{ij} \in \{0, 1\} \quad (8)$$

در این مدل محدودیت‌های دوم و سوم نشان‌دهنده آن است که به هر گره انبار m یال وارد و m یال خارج می‌شود، به عبارت دیگر این محدودیت تضمین می‌کند که جواب به دست آمده شرایط لازم برای وجود m زیر تور را دارا می‌باشد.

محدودیت‌های چهارم و پنجم نشان‌دهنده آن است که به هر گره $i \in V - \{0\}$ فقط یک یال وارد و یک یال خارج می‌شود، به عبارت دیگر این محدودیت تضمین می‌کند که جواب به دست آمده یک جواب هامیلتونی (هر گره فقط یکبار ملاقات شود) باشد.

محدودیت ششم و هفتم برگرفته شده از محدودیت حذف زیر گذر استاندارد است که اولین بار به وسیله دانتزیک برای مساله TSP در سال ۱۹۵۴ ($(S \subset \{1, \dots, n\}, |S| \geq 2)$) $\sum_{i,j \in S} x_{ij} \leq |S| - 1$ ارائه شد [۲۳]. این محدودیت از ایجاد زیر دور که شامل گره صفر نباشد، در پیمایش گره‌ها جلوگیری می‌کند و باعث می‌شود که گراف مربوطه یک گراف همبند باشد. در شکل ۲ یک جواب نشدنی برای مساله $MTSP$ پیشنهاد شده است که در همه شرایط به جز شرایط ششم و هفتم صدق می‌کند. نهایتاً محدودیت هفتم به شرایط دودویی متغیر x_{ij} اشاره می‌کند. باید توجه کرد که محدودیت‌های سوم، چهارم و هفتم مساله بالا از فرمول‌بندی مساله تخصیص الهام گرفته شده است.



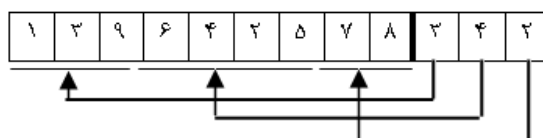
شکل ۲. یک جواب نشدنی مساله $MTSP$ در محدودیت‌های ۵ و ۶

۳ الگوریتم پیشنهادی

اگر چه الگوریتم GA دارای برخی مزایا بوده و هنوز هم در مقایسه با سایر الگوریتم‌های فراابتکاری یکی از بهترین روش‌ها برای حل مسایل بهینه‌سازی ترکیباتی محسوب می‌شود و از لحاظ سادگی و اجرا دارای مزایای زیادی است، اما متأسفانه زمان اجرای طولانی برای رسیدن به یک جواب مناسب و رکود سریع در بدست آوردن جواب‌های بهتر از معایب آن محسوب می‌شوند.

در این مقاله سعی شده است که تا حد امکان بر این ضعف غلبه کرده و الگوریتم تا حد امکان ارتقاء یافته و به جواب‌های بهتری دست یابد. برای این منظور در ابتدا از الگوریتم EAS برای یافتن جواب‌های اولیه استفاده می‌شود. روش کار بدین صورت است که ابتدا مساله $MTSP$ با الگوریتم EAS مورد حل قرار می‌گیرد (شرط پایانی الگوریتم انجام حلقه الگوریتم برابر با تعداد گره هر مساله است). چون این الگوریتم دارای همگرایی خوبی نسبت به سایر نسخه‌های الگوریتم مورچگان است، می‌تواند جواب‌های نسبتاً خوبی را در زمانی مورد قبول بدست آورد. سپس با در نظر گرفتن بهترین جواب هر تکرار الگوریتم به عنوان کروموزوم‌های ابتدایی، تقریباً بهترین جواب هر منطقه از فضای جواب در نظر گرفته می‌شود. حال وقتی که الگوریتم EAS به اتمام رسید، این جواب‌ها با تعدادی جواب‌های تصادفی، به عنوان جمعیت اولیه برای الگوریتم GA ، در نظر گرفته می‌شوند تا بدین وسیله

یک مجموعه مناسب از جواب‌های اولیه پدیدار گردد و الگوریتم بتواند با توجه به کیفیت این جواب‌های اولیه تا حد امکان به جستجوی دقیق‌تر فضای شدنی پردازد و به جواب‌های بهتری دست پیدا کند. شکل ۳ یک جواب یا کروموزوم ابتدایی، برای مساله *MTSP* را نشان می‌دهد. در این شکل کروموزوم از دو قسمت تشکیل شده است به طوری که در قسمت اول شماره گره‌های ملاقات شده به وسیله هر فروشنده و در قسمت دوم تعداد گره‌های مورد ملاقات قسمت اول نشان داده شده است. در این مثال تعداد گره‌ها به جز انبار ۹ و تعداد فروشنده‌ها ۳ در نظر گرفته شده است.



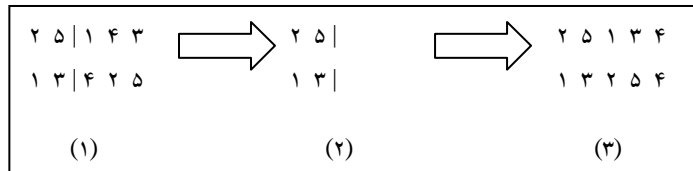
شکل ۳. یک جواب شدنی

علت اینکه چرا این عمل موفقیت بیشتری نسبت به روش‌های دیگر دارد این است که هر کدام از روش‌های فراابتکاری دارای مزایا و معایبی هستند به طور مثال الگوریتم *EAS* سرعت همگرایی خوبی دارد ولی متاسفانه بعد از تعدادی تکرار، نمی‌تواند جواب‌های بهتری تولید کند زیرا در تکرارهای قبلی بر روی یال‌هایی خاص به مقدار قابل توجهی فرمون ریخته شده است و الگوریتم با احتمال بسیار بالایی دوباره آن یال‌ها را انتخاب می‌کند. در حالیکه در الگوریتم *GA* اگر جواب‌های اولیه ترکیبی از کروموزوم‌های نخبه و تصادفی باشند آن گاه علی‌رغم اینکه دارای سرعت همگرایی کمتری نسبت به الگوریتم *EAS* است، دارای قدرت بیشتری برای فرار از نقاط بهینه محلی می‌باشد. باید توجه کرد که ترکیب این دو الگوریتم سبب می‌شود که تا حد امکان از مزایای آن‌ها استفاده شود و کارایی الگوریتم به مقدار قابل توجهی افزایش پیدا کند. در نتیجه الگوریتم دارای قدرت بیشتری برای جستجو در فضای شدنی مساله می‌شود و می‌تواند به جواب‌های باکیفیت‌تری دست پیدا کند.

دو عملگر بسیار مهم و پایه‌ای الگوریتم *GA* که در تمام نسخه‌های الگوریتم وجود دارد، عملگرهای تقاطع و جهش می‌باشند. باید توجه کرد که برای این دو عملگر که باعث می‌شود نسل بعدی به وجود آید حالت‌های بسیار زیادی وجود دارد که هر کدام به نوبه خود دارای نقاط ضعف و قوتی هستند. اما ایده اصلی در تمامی آن‌ها این است که تا حد امکان باید سعی شود که والدین باکیفیت‌تر با احتمال بیشتری انتخاب شوند. به این امید که فرزندان بهتری تولید کنند و در نهایت جواب‌های بهتری بوجود آید.

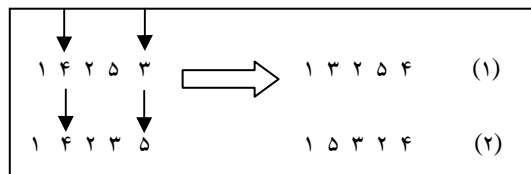
در این مقاله برای عملگر تقاطع از نسخه تک نقطه‌ای استفاده شده است. در این روش یک ژن از یک کروموزوم را در نظر گرفته و نقطه شکست نامیده می‌شود. سپس ژن‌های بعدی آن‌ها در نظر گرفته شده و به ترتیبی که در کروموزوم دوم وجود دارد، جایگزین می‌شود. به طور مثال در شکل ۴ و در کروموزوم اول، عدد ۵ به عنوان محل شکستن کروموزوم در نظر گرفته می‌شود. حال اعداد بعد از ۵ یعنی ۱، ۴ و ۳ را در نظر گرفته و ترتیب این سه عدد

در ژن دوم به دست می آید که همان طور که مشاهده می کنید این ترتیب برابر با ۱، ۳ و ۴ است. سپس این ترتیب به اعداد قبل از شکست کروموزوم اول اضافه می شود. به همین ترتیب برای کروموزوم دوم نیز همین کار انجام شده تا جواب مربوطه به دست آید.



شکل ۴. ترکیب دو کروموزوم

همچنین در این الگوریتم از دو روش جهش متفاوت، شکل ۵، برای بالا بردن بیشتر کیفیت جوابها استفاده شده است. این دو روش به ترتیب تعویض دو ژن (۱) و انتخاب دو ژن و برعکس کردن ترتیب ژنها در بین این دو ژن در کروموزوم می باشند (۲). باید توجه کرد که در هر تکرار هر دو این جهشها مورد استفاده قرار نمی گیرد بلکه یکی از این دو با احتمال برابر انتخاب می شود.



شکل ۵. جهش در یک کروموزوم

اگر چه در ابتدای الگوریتم GA تنوع مناسبی بین جوابهای اولیه وجود دارد و به سرعت در اجراهای ابتدایی کیفیت جوابها افزایش می یابد اما هر چه الگوریتم به پیش می رود کروموزومها دارای کیفیت بهتری می شوند و تنوع جوابها کاهش می یابد و الگوریتم به اصطلاح دچار رکود ناخواسته یا همگرایی زودرس می شود. این عمل سبب می شود که جستجو به طور زود هنگام به ناحیه خاصی از فضای شدنی همگرا شده و جستجو در اطراف بهینه های محلی متمرکز شود. بنابراین قسمت قابل توجهی از جوابهای فضای شدنی مورد بررسی قرار نمی گیرد. در این صورت الگوریتم کارایی خود را برای یافتن کروموزومهای غیر تکراری از دست داده و دیگر نمی تواند راه حل های بهتری تولید کند. در نتیجه الگوریتم در یک بهینه محلی را به عنوان بهینه سراسری در نظر می گیرد. یکی از روشهایی که در این مقاله برای حل این مشکل ارائه می شود، استفاده از کروموزومهای تصادفی در هر تکرار است [۲۴]. این ایده سبب می شود که تنوع کروموزومها در هر تکرار الگوریتم به مقدار مناسب حفظ شود. به عبارت دیگر این عمل باعث افزایش کارایی الگوریتم می شود زیرا تنوع کروموزومها در تمامی تکرارها حفظ شده و ترکیب آنها باعث ایجاد جوابهای جدید و متنوع می شود. در نتیجه الگوریتم می تواند فضای بیشتری از جوابها را مورد بررسی قرار داده و به جوابهای با کیفیت تری دست یابد.

۴ تحلیل پارامترها

به علت وجود پارامترهای زیادی که در الگوریتم GA وجود دارد و همچنین مسایل متنوعی که هر کدام دارای ویژگی‌های منحصر به فرد هستند، بهتر است که این پارامترها مورد بررسی بیشتر قرار گرفته تا مقدارهای مناسب آن‌ها برای حل مسأله $MTSP$ بدست آید. برای این منظور ۵ مثال استاندارد مسأله مسیریابی وسیله نقلیه با ظرفیت از ۳۱ تا ۱۰۱ گره در نظر گرفته می‌شوند تا با آزمایش مقادیر گوناگون برای پارامترهای آن، مقدار مناسب به دست آید. باید توجه کرد که مقدار تقاضاهای هر مشتری برای این پنج مثال صفر در نظر گرفته می‌شود تا مثال‌های برای $MTSP$ بدست آید. برای دسترسی به این مثال‌ها می‌توان به آدرس زیر مراجعه نمود:

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

در جدول ۱ خصوصیات مثال‌ها و نتایج الگوریتم پیشنهادی برای چندین جمعیت اولیه کروموزوم‌ها نشان داده شده است به طوری که در سه ستون اولیه، خصوصیات پنج مثال مورد آزمایش مسأله $MTSP$ شامل نام، تعداد مشتری و تعداد فروشنده نشان داده شده است. به علاوه در ستون‌های دیگر این جدول با ثابت نگهداشتن میزان تکرار الگوریتم به میزان $8n$ ، n تعداد گره هر مثال است، نتایج الگوریتم برای جمعیت‌های اولیه گوناگون از $2n$ تا $8n$ نشان داده شده است. نکته بسیار مهمی که در اینجا وجود دارد این است که کیفیت جواب‌ها و تعداد جمعیت اولیه دارای روال قابل پیش‌بینی و منظمی نیستند و علی‌رغم آن که به نظر می‌رسید که هر چه تعداد جمعیت اولیه بیشتر شود، باید مقدار جواب به دست آمده نیز بهتر شود ولی این فرض اتفاق نیفتاده است. با مقایسه جواب‌های به دست آمده در جدول ۱ می‌توان نتیجه گرفت که بهترین مقادیر برای سه مثال $6n$ و برای دو مثال $8n$ است. در نتیجه بهتر است که برای جدول ۲ مقدار بهینه جمعیت اولیه برابر $6n$ در نظر گرفته شود.

جدول ۱. نتایج آزمایشات برای چندین نوع جمعیت اولیه (بهترین مقدار در ۱۰ بار آزمایش در نظر گرفته شده است)

مثال‌ها	تعداد مشتری	تعداد فروشنده	$p = 2n$	$p = 4n$	$p = 6n$	$p = 8n$
E-۰۳۱-۰۹	۳۰	۹	۴۵۱/۶۲	۴۴۲/۸۲	۴۳۸/۹۱	۴۳۸/۹۹
E-۰۴۱-۱۴	۴۰	۱۴	۷۰۳/۴۲	۶۹۸/۷۲	۶۸۷/۴۱	۶۸۶/۸۲
E-۰۵۱-۰۵	۵۰	۵	۵۰۰/۶۱	۴۸۶/۹۲	۴۷۵/۶۱	۴۷۶/۳۱
E-۰۷۶-۰۷	۷۵	۷	۶۸۴/۴۱	۶۳۶/۷۸	۶۳۶/۵۱	۶۳۶/۷۱
E-۱۰۱-۰۸	۱۰۰	۸	۸۱۵/۳۱	۷۸۲/۶۱	۷۷۴/۵۱	۷۷۵/۳۱

در جدول ۲ همانند جدول ۱ پارامتر دیگری در الگوریتم GA مورد بررسی قرار گرفته شده است. این پارامتر تعداد اجرای الگوریتم به عنوان شرط پایانی است که از $2n$ تا $8n$ در نظر گرفته شده است. در این جدول نیز جواب‌ها دارای یک روال منطقی نبوده و در بعضی از مواقع با تعداد بیشتر حلقه برای الگوریتم، جواب‌های باکیفیت کمتری به دست آمده است. بنابراین به طور کلی با مقایسه جواب‌ها و با توجه به میانگین هر پنج مثال می‌توان نتیجه گرفت که تعداد $8n$ مقدار بهتری برای تعداد تکرار الگوریتم است.

جدول ۲. نتایج آزمایشات برای شرط همگرایی الگوریتم (بهترین مقدار در ۱۰ بار آزمایش در نظر گرفته شده است)

مثال‌ها	$l=2n$	$l=4n$	$l=6n$	$l=8n$
E-۰۳۱-۰۹	۴۴۸/۲۵	۴۴۸/۵۲	۴۳۸/۵۴	۴۳۵/۳۲
E-۰۴۱-۱۴	۶۷۱/۳۲	۶۹۸/۵۴	۶۸۳/۱۷	۶۸۳/۴۲
E-۰۵۱-۰۵	۴۹۶/۵۶	۴۸۳/۵۸	۴۷۲/۵۹	۴۶۹/۷۲
E-۰۷۶-۰۷	۶۸۱/۵۳	۶۴۶/۴۲	۶۳۴/۸۵	۶۲۷/۵۹
E-۱۰۱-۰۸	۸۱۱/۵۲	۷۸۲/۶۳	۷۷۲/۷۳	۷۶۸/۴۲

به علت اهمیتی که کروموزوم‌های تصادفی ابتدایی در تنوع جواب‌ها و رسیدن به جواب‌های با کیفیت دارند، مقادیر درصد‌های تصادفی کروموزوم‌ها در هر تکرار و جواب‌هایی که الگوریتم برای مثال‌های مختلف به دست آورده، در جدول ۳ نشان داده شده است. در این جدول میزان درصد کروموزوم‌ها تصادفی ما بین ۱ تا ۱۰ در نظر گرفته شده است. این اعداد بدین معنی هستند که به طور مثال در جمعیت اولیه کروموزوم‌ها ۳ درصد آن‌ها بصورت تصادفی و بدون هیچ قاعده‌ای ایجاد شده است.

با توجه به مقادیری که در جدول ۳ بدست آمده است، می‌توان نتیجه گرفت که علی‌رغم این که استفاده مناسب از کروموزوم‌های ابتدایی، کیفیت جواب‌ها را افزایش می‌دهد اما استفاده زیاد از این کروموزوم‌ها نیز سبب می‌گردد که جواب‌های خوب نیز از بین بروند و الگوریتم نتواند به جواب‌های بهتری دست پیدا کند. همچنین با بررسی مقادیر به دست آمده به وسیله الگوریتم در جدول ۳ می‌توان نتیجه گرفت که بهترین مقادیر جواب‌ها برای مقدار ۳ درصد بدست آمده است و علی‌رغم اینکه کیفیت جواب‌ها از درصد‌های ۱ به ۳ افزایش پیدا می‌کند اما وقتی که مقادیر درصد‌های تصادفی از ۳ بیشتر می‌شود، کیفیت جواب‌ها نیز کاهش پیدا کرده تا این که بدترین مقادیر خود را در مقدار ۱۰ درصد کسب می‌کنند.

جدول ۳. درصد‌های متفاوت از جواب‌های تصادفی در هر تکرار (بهترین جواب در ۱۰ آزمایش نشان داده شده است).

مثال‌ها	$O=۰/۰۱$	$O=۰/۰۳$	$O=۰/۰۶$	$O=۰/۱۰$
E-۰۳۱-۰۹	۴۳۵/۶۲	۴۳۳/۵۲	۴۴۶/۱۱	۴۵۲/۵۳
E-۰۴۱-۱۴	۶۸۳/۲۱	۶۸۲/۷۲	۶۹۵/۴۳	۷۰۵/۶۲
E-۰۵۱-۰۵	۴۶۸/۹۳	۴۶۷/۵۴	۴۸۲/۴۱	۴۹۵/۶۹
E-۰۷۶-۰۷	۶۲۶/۶۳	۶۲۵/۸۳	۶۴۵/۳۲	۶۶۱/۷۳
E-۱۰۱-۰۸	۷۶۹/۹۹	۷۶۸/۱۱	۷۷۹/۴۲	۷۹۳/۸۳

۵ نتایج محاسباتی

برای بررسی کارایی الگوریتم پیشنهادی در مقایسه با سایر روش‌های فراابتکاری تعدادی از مثال‌های استاندارد مسأله *MTSP* شامل $pr152$ ، $pr226$ ، $pr229$ ، $pr439$ و $pr1002$ در نظر گرفته شده است که دارای بازه بسیار خوبی از تعداد مشتری‌ها می‌باشند که مقادیر بین ۷۶ تا ۱۰۰۲ را اختیار می‌کنند. این دسته از مثال‌ها از کتابخانه *TSP* انتخاب شده است با این تفاوت که تعداد فروشنده‌های دوره‌گرد در هر کدام از مثال‌ها برابر ۵ در نظر گرفته می‌شود. به علاوه باید اضافه کرد که در این مقایسه‌ها، پارامترهایی که برای الگوریتم پیشنهادی به کار گرفته شده است به صورت زیر می‌باشند.

$$l=\lambda n, p=\epsilon n, o=0.03$$

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. از طرف دیگر بهترین مقدار پیدا شده برای هر کدام از الگوریتم‌ها در ۱۰ تکرار الگوریتم در جدول ۵ پیدا شده است در حالی جدول ۶ مقدار زمان مصرفی برای یافتن جواب‌ها برای هر کدام از الگوریتم‌ها را نشان می‌دهد.

خصوصیات کامل مثال‌های ذکر شده و همچنین نتایج الگوریتم‌ها در جدول ۵ آورده شده است. در این جدول به ترتیب در ستون‌های دوم و سوم تعداد مشتری‌ها و فروشنده‌های هر مثال آورده شده است. در حالی که در ستون چهارم، حداکثر گره‌های قابل ملاقات توسط هر فروشنده نشان داده شده است. باید توجه کرد که در ستون پنجم نتایج الگوریتم *GA* معمولی و در ستون ششم نتایج الگوریتم پیشنهادی برای مثال‌ها نشان داده شده است. برای اینکه یک مقایسه کامل بین الگوریتم پیشنهادی و الگوریتم‌های دیگر فرا ابتکاری صورت گیرد، ستون‌های هفتم و هشتم به ترتیب نشان‌دهنده الگوریتم‌های اصلاحی ژنتیک و اصلاحی مورچگان از ادبیات موضوع می‌باشند. باید توجه کرد که الگوریتم اصلاحی ژنتیک توسط تانگ و همکارش ارایه [۲۵] در حالیکه الگوریتم اصلاحی مورچگان توسط جانجیل و دینگوی پیشنهاد شده است [۲۶].

نتایج به دست آمده در این جدول به این نکته اشاره دارد که الگوریتم پیشنهادی نسبت به الگوریتم *GA* معمولی از کارایی بسیار خوبی برخوردار است به طوری که توانسته است در همه مثال‌ها کیفیت مقادیر جواب‌ها را به مقدار قابل قبولی افزایش دهد. البته شایان به ذکر است که الگوریتم ژنتیک معمولی در مقایسه با دو الگوریتم دیگر دارای رقابت خوبی نبوده و نتوانسته است که جواب‌های با کیفیتی را بدست آورد. از طرف دیگر الگوریتم پیشنهادی دارای رقابت خوبی با الگوریتم‌های دیگر فراابتکاری است. در این مثال‌ها الگوریتم پیشنهادی توانسته است که نسبت به الگوریتم اصلاحی *GA* کیفیت جواب‌ها را به مقدار قابل توجهی افزایش داده و با بدست آوردن میانگین ۱۸۴۳۴۹ در مقابل ۱۹۵۵۹۰ برای ۵ مثال برتری خود را با اختلاف قابل توجهی ثابت کند. باید توجه کرد که رقابت اصلی در این جدول بین الگوریتم پیشنهادی و الگوریتم اصلاحی مورچگان وجود دارد. به طوری که در بین ۵ مثال مورد آزمایش در ۳ مثال الگوریتم پیشنهادی توانسته است که به جواب‌های بهتری دست یابد اما در مقابل در ۲ مثال، الگوریتم پیشنهادی نتوانسته است که به جواب‌های با کیفیت تری دست یابد. به علاوه از نظر به دست آوردن میانگین جواب‌ها برای مثال‌های مورد آزمایش، الگوریتم ترکیبی پیشنهادی با بدست

آوردن ۱۸۴۳۴۹ در مقابل ۱۸۴۹۷۱/۶ توانسته است که در این محک نیز جواب‌های بهتری را کسب کند. بنابراین می‌توان نتیجه گرفت که این الگوریتم از قدرت بسیار خوبی برای فرار از نقاط بهینه محلی برخوردار بوده و بدین علت توانسته است که بهترین میانگین جواب‌ها را برای این مثال‌ها به دست آورد. باید توجه کرد که از نظر قدرت یافتن بهترین جواب می‌توان الگوریتم‌های جدول ۴ را به ترتیب از ضعیف به قوی به صورت الگوریتم ژنتیک، الگوریتم اصلاحی ژنتیک، الگوریتم اصلاحی مورچگان و الگوریتم پیشنهادی ترتیب‌بندی نمود.

جدول ۴. مقایسه بین الگوریتم‌های فراابتکاری

MACO	MGA	PA	GA	l	v	n	نمونه
۱۳۰۹۵۳	۱۲۷۸۳۹	۱۲۷۶۲۴	۱۵۳۴۲۵	۴۰	۵	۱۵۲	Pr152
۱۶۷۶۴۶	۱۶۶۸۲۷	۱۶۶۶۲۴	۱۷۳۸۷۲	۵۰	۵	۲۲۶	Pr226
۸۲۱۰۶	۸۲۱۷۶	۸۲۱۲۱	۸۳۶۲۷	۷۰	۵	۲۹۹	Pr299
۱۶۱۹۵۵	۱۷۳۸۳۹	۱۶۱۹۵۱	۱۷۳۴۲۵	۱۰۰	۵	۴۳۹	Pr439
۳۸۲۱۹۸	۴۲۷۲۶۹	۳۸۳۴۲۵	۳۴۳۹۷۶	۲۲۰	۵	۱۰۰۲	Pr1002

میزان زمان مورد احتیاج برحسب ثانیه برای یافتن جواب هر مثال به وسیله الگوریتم‌های ذکر شده، در جدول ۵ آورده شده است. باید توجه کرد که شرط پایان الگوریتم در دو روش پیشنهادی برای حل مثال‌ها، ۱۰ بار تکرار شدن بهترین جواب در حین اجرای الگوریتم است. به عبارت دیگر الگوریتم مربوطه تا زمانی اجرا می‌شود که بهترین جواب به دست آمده در ۱۰ تکرار متوالی تغییری نکند.

جدول ۵. مقایسه زمان اجرای الگوریتم‌ها فراابتکاری

MACO	MGA	PA	GA	نمونه
۱۲۸	۹۱	۸۹	۸۲	Pr152
۱۴۳	۱۶۵	۱۶۸	۱۳۱	Pr226
۲۸۸	۳۶۳	۳۵۲	۲۲۰	Pr299
۵۶۳	۶۲۳	۵۶۱	۳۹۵	Pr439
۲۶۲۰	۲۸۹۲	۲۱۵۲	۱۶۵۲	Pr1002

با توجه به نتایج به دست آمده در این جدول می‌توان نتیجه گرفت که از لحاظ زمان مصرفی، الگوریتم GA معمولی دارای بهترین جواب است و بعد از آن الگوریتم پیشنهادی، الگوریتم اصلاحی مورچگان و الگوریتم اصلاحی ژنتیک قرار دارند. به علاوه باید توجه کرد که علی‌رغم اینکه الگوریتم پیشنهادی دارای جواب‌های بهتر

نسبت به الگوریتم *GA* معمولی است اما از نظر زمان اجرا دارای کیفیت کمتری می‌باشد. البته این نتیجه غیر قابل پیش‌بینی نبود زیرا براساس اصلاحات ذکر شده، الگوریتم دارای کیفیت بیشتری برای فرار از نقاط بهینه محلی می‌شود و بدین ترتیب فضای جواب مساله را با دقت بیشتری مورد تجزیه و تحلیل قرار می‌دهد. این عمل سبب می‌شود که علی‌رغم این که زمان بیشتری صرف جستجو در فضای جواب مساله می‌شود، الگوریتم بتواند به جواب‌های بهتری دست پیدا کند.

۶ نتیجه‌گیری و جهت‌گیری‌های آینده

در این مقاله در ابتدا یک مدل برنامه‌ریزی خطی صحیح برای مساله *MTSP* ارائه شد و سپس الگوریتم ترکیبی ژنتیک بر پایه‌ی اجتناب از همگرایی زودرس برای حل مساله *MTSP* مورد استفاده قرار گرفت. همان طوری که نشان داده شد استفاده از الگوریتم نمونه مورچگان سبب شد که الگوریتم پیشنهادی از یک روند منظم برای جستجو در فضای جواب استفاده کند و در یک زمان اندک به جواب‌های بسیار خوبی دست پیدا کند. به علاوه استفاده از جواب‌های به دست آمده توسط الگوریتم نمونه مورچگان به همراه جواب‌های تصادفی دیگر، به عنوان کروموزوم‌های اولیه؛ سبب شد که الگوریتم از کارایی بیشتری برای فرار از نقاط بهینه محلی برخوردار شود. به نظر می‌رسد که ترکیب این روش با روش‌های فراابتکاری دیگر مانند بازپخت شبیه‌سازی، جستجوی ممنوع و غیره می‌تواند سبب شود که کیفیت جواب‌های الگوریتم به مقدار بیشتری ارتقاء یابد. همچنین گسترش این روش به مسایل دیگر مانند *MTSP* همراه با دریافت و تحویل کالا و مسیریابی وسیله نقلیه به مقاله‌های بعدی موکول می‌شود.

منابع

- [۲۲] یوسفی خوشبخت، م.، صدیق‌پور، م.، (۱۳۹۰). الگوریتم نمونه اصلاحی مورچگان برای حل مساله چندین فروشنده دوره‌گرد. مجله تحقیق در عملیات و کاربردهای آن، ۸ (۳)، ص ۸۳-۹۶.
- [۲۴] یوسفی خوشبخت، م.، صدیق‌پور، م.، خرم، ا. (۱۳۹۰). یک راهکار جدید برای فرار از نقاط بهینه محلی در الگوریتم ژنتیک، پذیرفته شدن در مجله علوم پایه دانشگاه آزاد علوم و تحقیقات.
- [1] Park, Y. B. (2001). A hybrid genetic algorithm for the vehicle scheduling problem with due times and time deadlines. *International Journal of Productions Economics*, 73 (2), 175–188.
- [2] Carter, A. E., Ragsdale, C. T. (2002). Scheduling pre-printed newspaper advertising inserts using genetic algorithms. *Omega*, 30(6), 415–421.
- [3] Lenstra, J. K., Rinnooy Kan, A. H. G. (1975). Some simple applications of the traveling salesman problem, *Operations Research Quarterly*, 26, 717–733.
- [4] Gorenstein, S. (1970). Printing press scheduling for multi-edition periodicals. *Management Science*, 16(6), B373–B383.
- [5] Brummit, B., Stentz, A. (1996). Dynamic mission planning for multiple mobile robots, in: *Proceedings of the IEEE International Conference on Robotics and Automation*.
- [6] Gilbert, K. C., Hofstra, R. B. (1992). A new multiperiod multiple traveling salesman problem with heuristic and application to a scheduling problem. *Decision Sciences*, 23, 250–259.

- [7] Tang, L., Liu, J., Rong, A., Yang, Z. (2000). A multiple traveling salesman problem model for hot rolling scheduling in Shanghai Baoshan Iron & Steel Complex. *European Journal of Operational Research*, 124, 267–282.
- [8] Laporte, G., Nobert, Y. A. (1986). Cutting planes algorithm for the m -salesmen problem. *Journal of the Operational Research Society*, 31, 1017–1023.
- [9] Gavish, B., Srikanth, K. (1986). An optimal solution method for large-scale multiple traveling salesman problems. *Operations Research*, 34(5), 698–717.
- [10] Gromicho, J., Paixão, J., Branco, I. (1992). Exact solution of multiple traveling salesman problems. In: Mustafa Akgül, et al., editors. *Combinatorial optimization*. NATO ASI Series, vol. F82. Berlin: Springer, 291–292.
- [11] Wacholder, E., Han, J., Mann, RC. (1989). A neural network algorithm for the multiple traveling salesmen problem. *Biology in Cybernetics*, 61, 11–19.
- [12] Goldstein, M. (1990). Self-organizing feature maps for the multiple traveling salesmen problem. In: *Proceedings of the IEEE international conference on neural network*. San Diego, CA, 258–261
- [13] Hsu, C., Tsai, M., Chen, W. (1991). A study of feature-mapped approach to the multiple travelling salesmen problem. *IEEE International Symposium on Circuits and Systems*, 3, 1589–1592.
- [14] Vakhutinsky, IA., Golden, L. B. (1994). Solving vehicle routing problems using elastic net. *Proceedings of the IEEE international conference on neural network*, 4535–4540.
- [15] Torki, A., Somhon, S., Enkawa, T. (1997). A competitive neural network algorithm for solving vehicle routing problem. *Computers and Industrial Engineering*, 33(3–4), 473–647.
- [16] Modares, A., Somhom, S., Enkawa, T. (1999). A self-organizing neural network approach for multiple traveling salesman and vehicle routing problems. *International Transactions in Operational Research*, 6, 591–606.
- [17] Somhom, S., Modares, A., Enkawa, T. (1999). Competition-based neural network for the multiple traveling salesmen problem with minmax objective. *Computers and Operations Research*, 26(4), 395–407.
- [18] Kara, I., Bektas, T. (2006). Integer linear programming formulations of multiple salesman problems and its variations. *European Journal of Operational Research*, 174, 1449–1458.
- [19] 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.
- [20] Zhang, T., Gruver, W. A. (1999). Smith MH. Team scheduling by genetic search. *Proceedings of the second international conference on intelligent processing and manufacturing of materials*, 2, 839–844.
- [21] Song, C., Lee, K., Lee, W. D. (2003). Extended simulated annealing for augmented TSP and multi-salesmen TSP. *Proceedings of the international joint conference on neural networks*. 3, 2340–2343.
- [23] Dantzig, G. B., Fulkerson, D. R., Johnson, S. M. (1954). Solution of a large-scale traveling salesman problem. *Operations Research*, 2, 393–410.
- [24] 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.
- [25] Junjie, P., Dingwei, W. (2006). An ant colony optimization algorithm for multiple travelling salesman Problem. In *ICICIC '06: Proceedings of the First International Conference on Innovative Computing, Information and Control*, Washington, DC, USA, 2006. IEEE Computer Society, 210–213.