

# یک الگوریتم بهبود یافته جمعیت مورچگان برای حل مسئله مسیریابی وسیله نقلیه همراه با دریافت و تحویل همزمان کالا

مجید یوسفی خوشبخت، دانشجوی دکترا، دانشکده ریاضی و علوم کامپیوتر، دانشگاه صنعتی امیرکبیر، تهران، ایران  
فرهاد رحمتی\*، دانشیار، دانشکده ریاضی و علوم کامپیوتر، دانشگاه صنعتی امیرکبیر، تهران، ایران

E-mail: frahmatai@aut.ac.ir

دریافت: ۱۳۹۰/۰۱/۲۰ - پذیرش: ۱۳۹۰/۰۴/۱۹

## چکیده

مسئله مسیریابی وسیله نقلیه<sup>۱</sup> (VRP) یکی از مهم‌ترین مسایل بهینه‌سازی ترکیباتی است که بسیار مورد توجه دانشمندان و محققان قرار گرفته است. این مسئله شامل مسیریابی برای یک ناوگان وسیله نقلیه است که در آن هر کدام از وسایل، مجموعه‌ای از مشتری‌ها را مورد ملاقات قرار می‌دهند، به شرط آن‌که هر مشتری فقط و تنها فقط به وسیله یک وسیله نقلیه ملاقات شود. هدف در این مسئله کمینه‌سازی فاصله‌های پیموده شده توسط همه وسایل نقلیه است. مسئله مسیریابی وسیله نقلیه همراه با دریافت و تحویل همزمان کالا<sup>۲</sup> (VRP\_SPD) یک گسترش از مسئله VRP که در آن وسایل نقلیه نه تنها کالاهایی را به مشتریان تحویل می‌دهند، بلکه همزمان کالاهایی را نیز از مشتریان تحویل می‌گیرند. روش بهینه‌سازی مورچگان<sup>۳</sup> (ACO) یک تکنیک احتمالی است که رفتار غذایی مورچگان را شبیه‌سازی کرده و آن‌را برای حل مسایلی که تاکنون الگوریتم‌های کارایی برای آن وجود ندارد، مورد استفاده قرار می‌دهد. الگوریتم جمعیت مورچگان<sup>۴</sup> (ACS) یکی از مشهورترین روش‌های فراابتکاری است که در قانون انتقال و بروزرسانی فرمون با سایر نسخه‌های ACO تفاوت دارد. این مقاله یک روش بهبود یافته روش ACS ترکیب شده با الگوریتم جستجوی محلی سه‌گانه را برای حل مسئله VRP\_SPD ارائه می‌کند که در اطلاعات ابتکاری و ضریب تبخیر فرمون الگوریتم ACS با روش معمولی تفاوت دارد. این عمل سبب جلوگیری از همگرایی زودرس، فرار از نقاط بهینه محلی و یافتن جواب‌های بهتر برای الگوریتم می‌شود. الگوریتم پیشنهادی روی ۳۴ مثال استاندارد از ادبیات موضوع اجرا شده است. نتایج محاسباتی نشان می‌دهد که الگوریتم توانسته است که بهترین جوابی که تاکنون به دست آمده را در چهار مثال ارتقا دهد و رقابت نزدیکی را با دیگر الگوریتم‌های فراابتکاری برای حل مسئله VRP\_SPD داشته باشد.

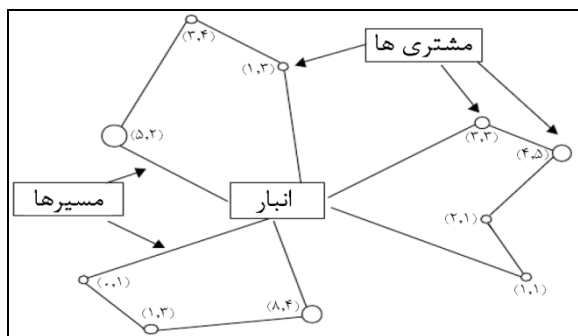
واژه‌های کلیدی: الگوریتم جمعیت مورچگان، دریافت و تحویل همزمان کالا، اطلاعات ابتکاری، مسئله مسیریابی وسیله نقلیه

## ۱- مقدمه

این‌گونه مسایل، مسئله مسیریابی وسیله نقلیه است که امروزه به‌علت گسترش‌ها و کاربردهای فراوانی که در مسایل تحقیق در عملیات دارد، بسیار مورد توجه قرار می‌گیرد (سپهری و حسینی مطلق، ۱۳۸۷). هدف این مسئله رساندن کالاها از یک انبار به تعدادی از مشتری‌ها است که در یک محدوده جغرافیایی پراکنده

مسایل حمل و نقل علاوه بر اینکه یکی از مهم‌ترین دسته‌های مسایل بهینه‌سازی ترکیباتی می‌باشند، مسایلی هستند که بسیار در زندگی و مشکلات روزمره زندگی انسان مؤثراند و بسیاری از مسایل کاربردی را می‌توان به نسخه‌ای از این مسایل تبدیل کرد و آن را حل کرد (ربانی، سپهری و ذگردی، ۱۳۸۷). یکی از

است و مختص اول مقدار دریافت کالا و مختص دوم مقدار تحویل کالا هر مشتری را نشان می‌دهد.



شکل ۱. یک جواب شدنی مسئله VRP\_SPD

الگوریتم ACS یکی از کاراترین نسخه‌های الگوریتم مورچگان است که تاکنون روی مسایل گوناگون تحقیق در عملیات مورد استفاده قرار گرفته است و نتایج بسیار خوبی را به دست آورده است. از طرف دیگر به علت وجود پارامترهای زیاد این الگوریتم، برای هر مسئله باید تحلیل پارامتری مناسبی صورت پذیرد تا الگوریتم از قابلیت کافی برای حل مسئله برخوردار شود. همچنین به علت ساختار پیچیده‌ای که مسئله VRP\_SPD دارد تاکنون تعداد کمی از کاربرد الگوریتم مورچگان بر این مسئله انجام شده است؛ به علاوه با توجه به قابلیت خوبی که الگوریتم ACS بر سایر مسائل مسیریابی و به خصوص نسخه‌های VRP دارد، در این مقاله یک روش بهبود یافته ACS ترکیب شده با الگوریتم جستجوی محلی سه‌گانه برای حل مسئله VRP\_SPD ارائه می‌گردد که در بخش‌های آینده با جزئیات کافی به آن پرداخته می‌شود.

در این مقاله ابتدا در بخش ۲ به تاریخچه این مسئله پرداخته می‌شود و کارهای انجام شده روی این مسئله مورد بررسی قرار می‌گیرد و سپس در بخش ۳ روش پیشنهادی و در بخش ۴ تحلیل پارامترها ارائه می‌گردد. نتایج محاسباتی که روی مثال‌های استاندارد اجرا شده است، در بخش ۵ مورد تجزیه و تحلیل قرار می‌گیرد و در انتها نتیجه‌گیری و جهت‌گیری‌های آینده در بخش ۶ آورده می‌شود.

## ۲- تاریخچه مسئله

شکل‌گیری مسئله VRP\_SPD به سال ۱۹۸۹ (Min, 1989) برمی‌گردد، به طوری که در این سال مین<sup>۱</sup> این نسخه از VRP را

شده‌اند. باید توجه کرد که اگر در مسئله ذکر شده علاوه بر اینکه وسایل نقلیه به مشتری‌ها کالا تحویل می‌دهند، مقداری کالا را از مشتری‌ها دریافت کنند. مسئله مربوطه، به مسئله مسیریابی وسیله نقلیه همراه با دریافت و تحویل کالا<sup>۲</sup> (VRPPD) تبدیل می‌شود که در دنیای واقعی در مسائلی مانند دریافت و تحویل شیشه‌های شیر و نوشابه، پشتیبانی از نیروها در زمان جنگ، توزیع و جمع‌آوری قطعات الکترونیکی و ... کاربرد فراوان دارد. به علاوه این مسئله مهم به صورت زیر تعریف می‌شود:

فرض کنید که  $V = \{0, 1, \dots, n\}$  مجموعه گره‌ها و  $A = \{(i, j) | i, j \in V \text{ and } i \neq j\}$  مجموعه یال‌های گراف  $G$  باشند. به علاوه هر یک از گره‌ها به جز گره 0 نمایش‌دهنده مشتری‌ها و دارای مقدار تقاضای  $q_i$  و مقدار تحویل  $d_i$  است. همچنین تقاضاهای هر یک از مشتری‌ها باید به وسیله فقط یکی از  $v$  وسیله نقلیه مسئله برآورده شود. نکات قابل توجه دیگری در این مسئله وجود دارند که عبارت است از:

- همه وسایل نقلیه در ابتدای الگوریتم باید در گره 0 باشند.
- هیچ وسیله نقلیه‌ای در هیچ زمان مجاز نیست که بیشتر از ظرفیت معین شده  $Q$  بارگذاری کند.
- مقدار  $d_i - q_i$  برای مشتری‌ها نباید بیشتر از ظرفیت مجاز برای وسایل نقلیه باشد. به عبارت دیگر برای هر  $i$  باید  $d_i - q_i \leq Q$  باشد.

در این صورت هدف مسئله عبارت است از یافتن مسیریابی با کمترین هزینه برای سرویس‌دهنده‌ها به طوری که شرایط ذکر شده برای مشتری‌ها و سرویس‌دهنده‌ها برقرار باشد و بعد از سرویس‌دهی همه وسایل نقلیه به انبار بازگردند. از طرف دیگر این مسئله را می‌توان به سه دسته کلی زیر طبقه‌بندی کرد که عبارتند از:

۱. مسایلی که دریافت و تحویل کالا باید همزمان انجام شود.
۲. مسایلی که دریافت و تحویل کالا به صورت مخلوط و غیرهمزمان انجام می‌شود.
۳. مسایلی که در آن دریافت کالا باید بعد از تحویل کالا صورت گیرد.

در اینجا مانند شکل ۱ فقط مسایلی در نظر گرفته می‌شوند که در آنها تحویل و دریافت کالا به صورت همزمان انجام شود. در این شکل مقدار ظرفیت سه وسیله نقلیه برابر ۱۰ در نظر گرفته شده

آنجللی<sup>۱۹</sup> و همکارش مورد استفاده قرار گرفت (Angelelli and Mansini, 2003)، در حالی که چندریو<sup>۲۰</sup> و همکارانش در سال ۱۹۹۹ از جستجوی ممنوع استفاده کردند (Gendreau, Laporte and Vigo, 1999).

از طرف دیگر نگی و سلحی در سال ۲۰۰۵ (Nagy and Salhi, 2005) احتمال جداکردن سرویس‌دهی مشتریان در دو قسمت یا ترکیب سرویس‌دهی در خلال مراحل حل را مورد توجه قرار دادند، اما این انعطاف‌پذیری به‌وسیله روش آنها ممکن نشد. همچنین در همین سال، کریسپیم<sup>۲۱</sup> و همکارش یک روش ترکیبی شامل روش جستجوی ممنوع و همسایگی متغیر برای مسئله پیشنهاد دادند (Crispim and Branda, 2005) که از کارایی بالایی برای حل مسایل برخوردار بود. به‌علاوه چن<sup>۲۲</sup> و وو<sup>۲۳</sup> یک روش ورودی به همراه یک روش ترکیبی رکورد به رکورد که شامل الگوریتم‌های بهبود دهنده و لیست ممنوع بود را در سال ۲۰۰۶ پیشنهاد دادند (Chen and Wu, 2006)، در حالی که تانگ و گالوا در همین سال یک روش جستجوی ممنوع را که دارای مراحل جریمه کردن متفاوتی بود برای مسئله ارائه کردند (Tang and Galvao, 2006).

همچنین در سال ۲۰۰۷ بیانچسی<sup>۲۴</sup> و همکارش یک سری از روش‌های ساختاری، جستجوی محلی، جستجوی ممنوع و ... را با هم مقایسه کردند (Bianchessi and Righini, 2007)، در حالی که در سال ۲۰۰۹ چندین محقق روش‌های گوناگونی را برای این مسئله ارائه دادند که در آنها می‌توان به کارهای انجام شده توسط کچیل<sup>۲۵</sup> و همکارش اشاره کرد که یک روش الگوریتم مورچگان بهبود یافته را برای حل مسئله استفاده می‌کرد. در روش آنها از یک ساختار مناسب برای افزایش کیفیت جواب‌ها استفاده شده بود و از دو جستجوی محلی چندگانه برای حل مسئله استفاده می‌کرد. مزیت این الگوریتم نسبت به الگوریتم‌های دیگر در آن بود که الگوریتم آنها قادر به حل نسخه‌های دیگر VRP نیز بود (Gajpal and Abad, 2009).

همچنین در همین سال آی<sup>۲۶</sup> و همکارش ابتدا یک فرمول‌بندی و سپس یک روش بهینه‌سازی ذرات برای مسئله ارائه دادند. فرمول‌بندی آنها به‌طوری بود که سه فرمول‌بندی قدیمی این مسئله را تولید می‌کرد. به‌علاوه الگوریتم پیشنهادی برای افزایش

مورد توجه قرار داد و آن را برای حل مسئله واقعی و کاربردی کتابخانه عمومی (تشکیل یافته از ۱ انبار، ۲ وسیله نقلیه و ۱۲ مشتری) و مسئله فروشنده دوره‌گرد به کار برد. روش مورد استفاده مین برای حل مسئله فروشنده دوره‌گرد براساس جریمه‌کردن کمان‌های نشدنی به‌وسیله قرار دادن هزینه آنها به اندازه بی‌نهایت و سپس دوباره حل کردن آنها بود.

جلس<sup>۲۷</sup> تعدادی از نسخه‌های VRP شامل مسئله مسیریابی وسیله نقلیه با بازگشت<sup>۲۸</sup> (برای ۱۵۰ مشتری) و VRPPD (برای بیش از ۱۰۰ مشتری) را مورد مطالعه قرار داد (Catay, 2010). روش مورد استفاده جلس از دو مرحله زیر تشکیل شده بود:

- تخصیص مشتری‌ها به وسایل نقلیه

- مسیریابی و استفاده از روش‌های بهبود دهنده

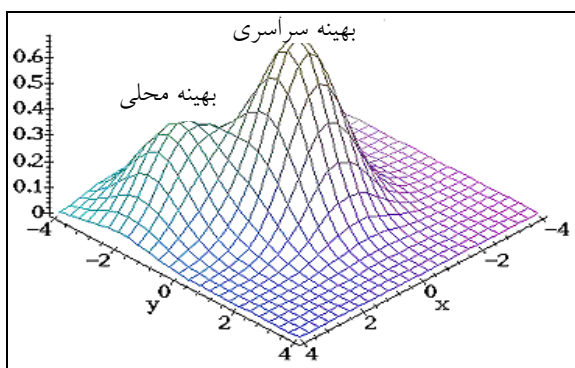
دلف<sup>۲۹</sup> بعد از چندین سال که مسئله VRPPD مورد فراموشی قرار گرفته بود، آن را دوباره مورد توجه قرار داد (Dethloff, 2001) و نوعی فرمول‌بندی ریاضی برای آن پیشنهاد کرد. سپس روش‌های ابتکاری ورودی که چهار معیار مختلف را برای حل مسئله استفاده می‌کرد، توسعه داد. به‌علاوه سلحی<sup>۳۰</sup> و نگی<sup>۳۱</sup> (Salhi and Nagy, 1999) چهار روش ابتکاری ورودی بر مبنای روش پیشنهاد شده توسط گلدن<sup>۳۲</sup> و همکارانش و همچنین کاسکو<sup>۳۳</sup> و همکارانش را ارائه دادند.

از این به بعد محققان به‌طور یکنواخت روش جستجوی محلی را مورد توجه قرار دادند. دو روش ابتکاری جستجوی محلی به‌وسیله تانگ<sup>۳۴</sup> و گالوا<sup>۳۵</sup> (Tang and Galvao, 2002) توسعه یافت به‌طوری‌که روش اول، روشی ابتکاری-افرازی است که به‌وسیله بسلی<sup>۳۶</sup> (Beasley, 1983) تعریف شده بود و روش دوم از الگوریتم جاروب ژیلت<sup>۳۷</sup> و میلر<sup>۳۸</sup> (Chalasanani and Motwani, 1999) استفاده می‌کرد.

باید توجه کرد که این دو روش برای حل مسئله مسیریابی وسیله نقلیه کلاسیک توسعه یافت. در همین زمان روش‌های توسعه‌یافته تانگ و گالوا برای حل مسئله فروشنده دوره‌گرد همراه با دریافت و تحویل کالا به‌طور همزمان مورد استفاده قرار گرفت. آنها همچنین یک فرمول‌بندی ریاضی تناوبی برای VRP\_SPD ارائه دادند. یک الگوریتم دقیق برای همین مسئله با پنجره‌های زمانی (محدودیت‌های زمانی) به‌وسیله

توجهات بسیاری را به خود جلب کرده است. در نتیجه در سال‌های اخیر الگوریتم‌های زیادی برای حل این مسئله ارائه شده است که می‌توان به‌طور کلی این الگوریتم‌ها را به دو دسته دقیق و ابتکاری تقسیم‌بندی کرد. در دسته اول که بیشتر برای مسایل با ابعاد کوچک مورد استفاده قرار می‌گیرد، جواب بهینه مسئله در یک زمان بسیار زیاد به دست می‌آید ولی در دسته دوم که تقریباً سه دهه است که به صورت جدی به آن پرداخته می‌شود، جواب بهینه مسئله در بیشتر مواقع به دست نمی‌آید، ولی الگوریتم می‌تواند به یک جواب زیر بهینه با کیفیت در یک زمان قابل قبول دست پیدا کند.

روش‌های جستجوی ابتکاری، روش‌هایی هستند که می‌توانند جواب را در زمانی محدود برای یک مسئله ارائه کنند و عمدتاً بر مبنای روش‌های شمارشی می‌باشند، با این تفاوت که از اطلاعات اضافی برای هدایت جستجو استفاده می‌کنند. این روش‌ها از نظر حوزه کاربرد، کاملاً عمومی هستند و می‌توانند مسایل پیچیده بهینه‌سازی ترکیباتی را حل کنند. توجه به این نکته ضروری است که در الگوریتم‌های ابتکاری برخلاف الگوریتم‌های فراابتکاری (نسخه‌ای دیگر از الگوریتم‌های ابتکاری) در هر تکرار الگوریتم یک جواب ثابت برای مسئله به دست می‌آید. بنابراین لازم نیست که برای یافتن جواب بهتر، الگوریتم را تکرار کرد و این خصوصیت از مزایای الگوریتم محسوب می‌شود اما در مقابل، این گونه از روش‌ها دارای راهکار مناسبی برای فرار از نقاط بهینه محلی (شکل ۲) نیستند و به این ترتیب به راحتی در این نقاط گیر می‌افتند.



شکل ۲. مسئله بهینه‌سازی ترکیباتی

از مهم‌ترین و امیدبخش‌ترین نسخه‌های الگوریتم‌های ابتکاری که مشکلات روش‌های شمارش کامل را ندارد، روش‌های

کیفیت جواب‌ها با یک ساختار اجتماعی ترکیب شده بود (Ai and Kachitvichyanukul, 2009). از طرف دیگر مینگیانگ<sup>۲۷</sup> و همکارش یک الگوریتم بهبود یافته تفاضلی تکاملی برای مسئله ارائه دادند. آنها در مقاله خود یک فرمول‌بندی مختلط صحیح کلی برای این مسئله استفاده کردند که در آن برای هر گره یک زمان سرویس‌دهی نیز در نظر گرفته شده بود. به‌علاوه در روش آنها ابتدا یک روش کدینگ اعشاری برای ساختاربندی جمعیت اولیه استفاده شده بود و سپس چندین عملگر متفاوت تکاملی جدید به کار گرفته می‌شد.

در نهایت جهت‌گیری صحیح مرتب براساس روش کدینگ عدد طبیعی برای بهبود بیشتر الگوریتم به کار برده می‌شد (Mingyong and Erbao, 2010). همچنین سابرامانیان<sup>۲۸</sup> و همکارانش یک روش ابتکاری موازی برای این مسئله ارائه دادند. جزئیات روش به این ترتیب است که الگوریتم آنها با چندین روش ابتکاری شامل روش‌های کاهنده همسایه متغیر، مرتب‌سازی همسایه تصادفی و استفاده از جستجوی محلی، شروع می‌شد (Subramanian et al., 2010).

به‌علاوه زاچارادیس<sup>۲۹</sup> و همکارانش یک روش ترکیبی مؤثر برای این مسئله ارائه دادند. این روش که از کارایی خوبی برخوردار بود از ترکیب روش‌های جستجوی ممنوع و جستجوی محلی تشکیل شده است و توانست به جواب‌های بسیار خوبی دست پیدا کند (Zachariadis, Tarantilis and Kiranoudis, 2009). همچنین در سال ۲۰۱۰ همین نویسندگان یک روش حافظه تطابقی برای مسئله ارائه دادند که در آن جواب‌های امیدبخش برای تولید جواب‌های باکیفیت‌تر با یکدیگر ترکیب می‌شدند. همچنین در این روش یک مکانیسم حافظه جدید برای ماکزیم کردن استخراج اطلاعات مسیریابی از حافظه تطابقی مورد استفاده قرار گرفت. با تست این روش بر مثال‌های استاندارد کارایی الگوریتم اثبات شد (Zachariadis, Tarantilis and Kiranoudis, 2010).

### ۳- روش پیشنهادی

چون مسئله VRP\_SPD همان‌طور که ذکر شد دارای کاربردهای فراوانی است، یافتن جواب بهینه این مسئله در کاهش هزینه شرکت‌های صنعتی و خدماتی از اهمیت زیادی برخوردار بوده و

۳-۱- الگوریتم جمعیت مورچگان

در سال ۱۹۹۷ دوریگو<sup>۳۰</sup> و گامباردلا<sup>۳۱</sup> سیستم جمعیت مورچگان را معرفی کردند (Dorigo and Gambardella, 1997) که تا حد زیادی از سیستم مورچه (AS) الهام گرفته بود. این الگوریتم به دلیل معرفی مکانیسم‌هایی جدید، از لحاظ کارایی، نسبت به AS به بهبودهایی دست یافته است. الگوریتم ACS دو تغییر عمده نسبت به قوانین به کار گرفته شده در AS دارد که عبارت‌اند از:

۱- یک قانون انتقال جدید معرفی شده است که به هر دو مورد استخراج و اکتشاف توجه دارد. به طور مثال حرکت از گره  $i$  به گره ملاقات نشده  $j$  و به‌وسیله مورچه  $k$  با  $J_i^k$  نشان داده می‌شود. مطابق قانون گذر نشان داده شده در زیر، احتمال ملاقات هر گره  $J_i^k$  برابر است با:

$$P_{ij}^k(t) = \begin{cases} 1 & q \leq q_0, j = j^* \text{ اگر} \\ 0 & q \leq q_0, j \neq j^* \text{ اگر} \\ \frac{\tau_{ij}^k(t) \eta_{ij}^k(t)}{\sum_{r \in J_i^k} \tau_{ir}^k(t) \eta_{ir}^k(t)} & \text{در غیر این صورت} \end{cases} \quad (1)$$

که در آن:

$j^*$ : گره‌ای است که دارای ماکزیمم مقدار  $([\tau_{ij}(t)][\eta_{ij}(t)]^p)$  در بین همه گره‌های ملاقات نشده در  $J_i^k$  است.

$\tau_{ij}(t)$ : مقدار فرمون روی یالی است که گره‌های  $i$  و  $j$  را به هم متصل می‌کند.

$\eta_{ij}(t)$ : اطلاعات ابتکاری برای اندازه‌گیری میدان دید مورچه است (مثلاً معکوس مسافت بین گره‌های  $i$  و  $j$ ).

$\alpha, \beta$ : پارامترهایی کنترلی هستند که نسبت اهمیت مقدار میدان دید مورچه را در برابر مقدار فرمون روی یالی که گره  $i$  و  $j$  را متصل کرده است، را تعیین می‌کند.

$q$ : یک پارامتر تصادفی است که به طور یکنواخت در  $[0,1]$  توزیع شده است.

$q_0$ : یک پارامتر آستانه ثابت در  $[0,1]$  است که نسبت اهمیت استخراج به اکتشاف را تعیین می‌کند. توجه کنید که انتخاب  $q_0 = 0$  روش ACS را به روش AS تبدیل می‌کند.

فراابتکاری هستند که اخیراً تمرکز بیشتری بر آنها صورت گرفته است. این روش‌ها شباهت‌هایی با سیستم‌های اجتماعی یا طبیعی دارند و کاربرد آنها برگرفته از روش‌های ابتکاری پیوسته می‌باشد که در حل مسایل مشکل ترکیباتی، نتایج بسیار خوبی داشته است.

در روش‌های فراابتکاری که میزان اجرای الگوریتم، برخلاف روش‌های ابتکاری، به تصمیم کاربر وابسته است، جواب‌ها تقریباً در یک زمان قابل قبول به‌دست می‌آید. اگر چه معمولاً جواب‌های این روش‌ها از جواب‌های روش‌های ابتکاری بهتر هستند و این‌گونه از الگوریتم‌ها از راه‌کارهایی استفاده می‌کنند که تا حد ممکن در بهینه‌های محلی گیر نیفتند، اما پارامترهای زیادی در این الگوریتم‌ها وجود دارند که باید توسط کاربر و به‌صورت تجربی به‌دست آیند. این حجم زیاد پارامترها سبب می‌شوند که این الگوریتم‌ها نتوانند در تکرارهای مشابه، جواب‌های یکسانی را به‌دست آورند. بنابراین می‌توان نتیجه گرفت که این‌گونه از الگوریتم‌ها دارای یک روال ثابت برای رسیدن به جواب‌ها نبوده و پارامترهای تصادفی نقش زیادی را در این الگوریتم‌ها بازی می‌کنند.

الگوریتم‌های ترکیبی فراابتکاری و ابتکاری، الگوریتم‌های جدیدی هستند که امروزه بسیار مورد توجه دانشمندان و محققان قرار می‌گیرند، زیرا ترکیب این روش‌ها سبب می‌شود که تا حد ممکن از مزایای هر کدام از الگوریتم‌ها به بهترین شکل استفاده شود و جواب‌های مناسب و با کیفیت در زمان اندک به‌دست آیند. باید توجه کرد که ترکیب این‌گونه از الگوریتم‌ها هم شامل مزایای الگوریتم ابتکاری می‌شود که جواب مناسبی را در یک زمان اندک به‌دست می‌آورد و هم دارای کارایی خوبی برای فرار از نقاط بهینه محلی هستند.

به عبارت دیگر، این‌گونه از الگوریتم‌ها دارای ساختار مناسب‌تری برای جستجو در فضای شدنی جواب‌ها بوده و دارای روال منظم‌تری برای یافتن جواب‌های نخبه هستند. در این الگوریتم‌ها ابتدا فضای شدنی تا حد ممکن و در یک زمان قابل قبول مورد جستجو قرار می‌گیرد و نواحی مستعد دارای جواب‌های نخبه به‌دست می‌آید، سپس روال جستجو از حالت سراسری به حالت محلی سوق داده می‌شود و الگوریتم با روالی دقیق‌تر به جستجوی همسایگی‌های جواب‌های نخبه می‌پردازد.

```

procedure ACS+3-Opt algorithm for solving
VRP_SPD
s* = φ; // s* is the best solution found yet //
f* := ∞; // f* is value of the s* //
n = the number of nodes; // n is the number of
ants //
loop = n;
initialize pheromone trails;
for k := 1 to loop do // main cycle // begin
S := none; // S is a matrix and population of
solutions //
for i := 1 to n do
begin
Construct a feasible solution si by using formula
(1)
S = S ∪ si;
If f(si) < f*
begin
f* := f(si);
s* = si;
end
end;
for i := 1 to length(S) do
begin
local update pheromone trails for each si
end
if the quality of the best solution is increased,
apply 3-opt local search to s*,
and obtain new s* and f* // save the best so far
solution //
global update pheromone trails for s*
end
show s* and f*
end // procedure //
    
```

شکل ۳. الگوریتم سیستم مورچگان برای حل مسئله VRP\_SPD

بنابراین موقعی که  $q$  کمتر یا مساوی  $q_0$  باشد مورچه‌ها اکتشاف را به کار می‌گیرند تا گره  $j^*$  را به عنوان گره بعدی در تورشان انتخاب کنند، در حالی که اگر  $q$  بزرگ‌تر از  $q_0$  باشد مورچه از استخراج بر پایه احتمال، برای انتخاب گره بعدی استفاده می‌کند.

۲- بروزرسانی ردپای فرمون به دو طریق انجام می‌شود:

- بروزرسانی محلی: همان‌طور که مورچه‌ها بین گره  $i$  و  $j$  حرکت می‌کنند، مقدار فرمون روی یال پیموده شده را، با استفاده از فرمول زیر، بروز می‌کنند

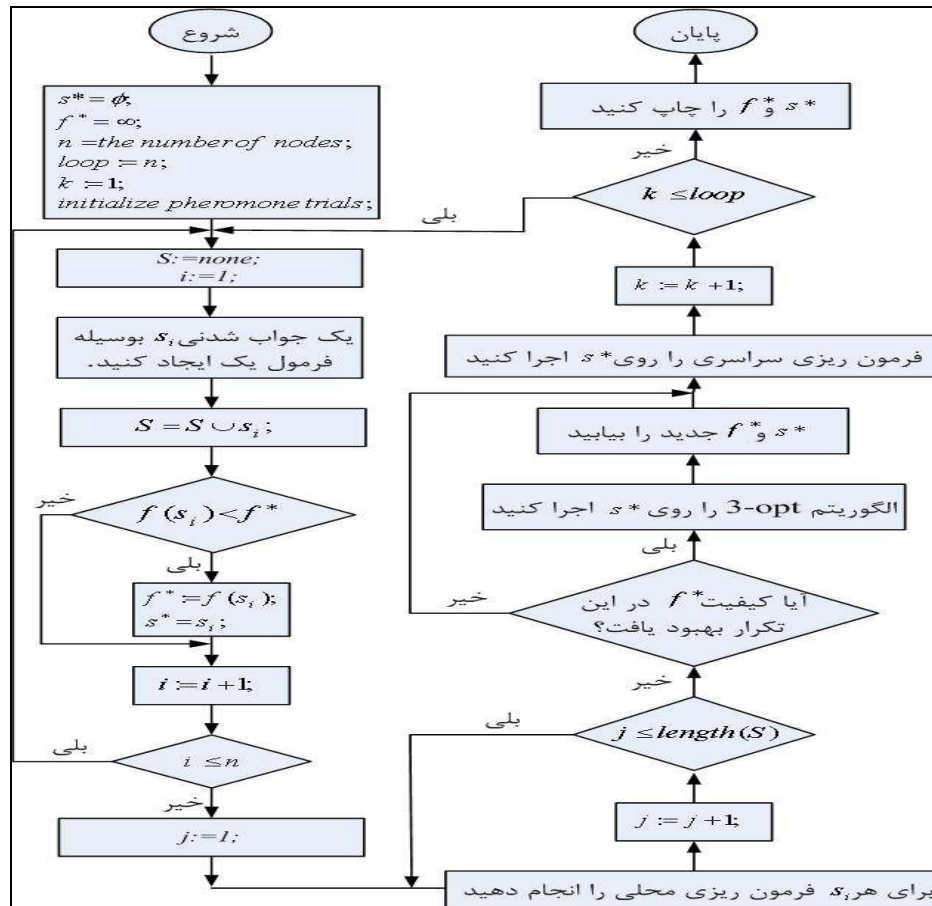
$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \rho \tau_0 \quad (2)$$

که در آن  $\tau_0$ ، مقدار اولیه فرمون است که به وسیله فرمول  $\tau_0 = (nC_i)^{-1}$  محاسبه می‌شود. در این فرمول  $n$  نشان‌دهنده تعداد گره‌ها و  $C_i$  هزینه تور اولیه است که به وسیله یک روش ابتکاری سازنده، مانند روش نزدیک‌ترین همسایه، تولید می‌شود. همچنین در فرمول (۲) نرخ تبخیر و پارامتری در دامنه  $[0, 1]$  می‌باشد که کاهش فرمون روی یال‌ها را تنظیم می‌کند. نتیجه بروزرسانی محلی این است که اگر در هر زمان یک مورچه یال  $(i, j)$  را پیمود، آن‌گاه ردپای فرمونش کاهش یابد تا یال‌ها برای مورچه‌های بعدی در تکرارهای آینده کمتر مطلوب باشند. این مطلب موجب تشویق بیشتر استخراج یال‌هایی که هنوز ملاقات نشده‌اند، می‌شود. باید توجه کرد که بروزرسانی محلی به اجتناب از موقعیت‌های رکود ضعیف کمک می‌کند.

- بروزرسانی سراسری: وقتی مورچه‌ها تورشان را تولید کردند، یال‌های متعلق به بهترین تور، به صورت زیر بروز می‌شوند:

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \rho(1/C_b) \quad (3)$$

که در آن،  $C_b$  هزینه بهترین تور،  $T_b$  پیدا شده از زمان شروع الگوریتم است. مهم است که توجه داشته باشیم بروزرسانی سراسری فقط فرمون روی یال‌های متعلق به بهترین تور را تنظیم می‌کند. این مطلب موجب تشویق مورچه‌ها برای جستجوی همسایگی‌های بهترین تور در تکرارهای بعدی می‌شود (Ellabib, Calamai and Basir, 2007). شکل ۳ و ۴ به ترتیب شبه کد و روند الگوریتم سیستم مورچگان را برای حل مسئله VRP\_SVP نشان می‌دهد.



شکل ۴. روند الگوریتم سیستم مورچگان برای حل مسئله VRP\_SPD

### ۳-۲- الگوریتم ارایه شده

اگر چه الگوریتم ACS یکی از قوی ترین نسخه های الگوریتم مورچگان برای حل مسائل بهینه سازی محسوب می شود، اما نمی تواند در حالت معمولی کارایی مناسبی برای حل مسئله VRP\_SPD داشته باشد. بنابراین به علت ضعف های موجود در الگوریتم ACS اصلاحاتی روی الگوریتم انجام می شود که در زیر آورده شده است و سبب می شود که کارایی الگوریتم افزایش پیدا کند و الگوریتم بتواند همان طور که در بخش ۵ نشان داده می شود به جواب های بسیار خوبی دست پیدا کند.

#### • اطلاعات ابتکاری مسئله:

یکی از ضرایبی که از اهمیت زیادی در الگوریتم ACS برخوردار و در انتخاب گره ها دارای نقش بسیار مهمی است، ضریب اطلاعات ابتکاری مسئله می باشد که با  $\eta_{ij}(t)$  نشان داده می شود.

این ضریب به خصوص در تکرارهای اولیه الگوریتم که در آن تأثیر فرمون از اهمیت کمتری برخوردار است، دارای نقش کلیدی می باشد و سمت و سوی مناطق جستجو را مشخص می کند. در الگوریتم معمولی مورچگان این مقدار برابر عکس مقدار هزینه بین دو گره در نظر گرفته می شود. بنابراین مانند روش نزدیک ترین همسایه اصلاحی، هرچه این هزینه کمتر باشد، احتمال انتخاب شدن آن یال بیشتر است. باید توجه کرد که الگوریتم نزدیک ترین همسایه و در نتیجه نوع احتمالی آن دارای کیفیت خوبی نیست و می توان این الگوریتم را با الگوریتم های ابتکاری کارا تر جایگزین کرد.

به همین علت در اینجا سعی شده است که از ایده الگوریتم صرفه جو استفاده شود که یکی از کارا ترین روش های ابتکاری بوده و دارای قدرت بیشتری برای یافتن جواب های با کیفیت در زمان اندک است. بنابراین در این مقاله این ضریب

صعودی از زمان اجرای الگوریتم در نظر گرفته شود. همچنین چون در این الگوریتم شرط پایانی برای اتمام الگوریتم، اجرای حلقه الگوریتم به تعداد مضربی از گره‌های هر مسئله تعیین شده است، این ضریب، تابعی از تعداد تکرار الگوریتم در نظر گرفته می‌شود.

بنابراین، پیشنهاد می‌شود که میزان تبخیر فرمون در ابتدای الگوریتم کم باشد و با افزایش تکرارها این مقدار افزایش یابد. زیرا این عمل سبب می‌گردد که الگوریتم از یک روند منطقی برای یافتن جواب‌های بهتر استفاده کند و به ترتیب در ابتدا و در انتهای الگوریتم به جستجوی سراسری و محلی پردازد، زیرا دقت جواب‌ها در ابتدای الگوریتم کم می‌باشد و مقدار تبخیر کم فرمون، سبب می‌شود که اختلاف فرمون بر یال‌های ملاقات شده و ملاقات نشده در تکرارهای الگوریتم، افزایش کمی داشته باشد تا اینکه شانس انتخاب شدن یال‌های ملاقات نشده در تکرارهای بعدی کاملاً از بین نرود. این عمل همچنین موجب می‌شود که الگوریتم در ابتدای روند به جستجوی سراسری پردازد، اما هنگامی که تکرارها افزایش پیدا می‌کند و جواب‌های باکیفیت‌تر تولید می‌شوند، جستجو بهتر است از حالت سراسری به حالت محلی تمایل پیدا کند تا به این وسیله به جستجوی همسایگی‌های جواب‌های نخبه پرداخته شود. بنابراین، افزایش تبخیر فرمون با یک تابع صعودی، سبب می‌گردد که این هدف نیز تحقق یابد؛ زیرا با افزایش ضریب فرمون، تفاوت فرمون بین یال‌های ملاقات شده و ملاقات نشده، تشدید گردیده و تقریباً شانس انتخاب شدن یال‌های ملاقات نشده در تکرارهای قبلی به صفر نزدیک می‌شود. در نتیجه الگوریتم با احتمال بیشتری یال‌های ملاقات شده قبلی را به همراه همسایگی‌های آنان در نظر می‌گیرد.

بنابراین، هدف این مقاله ارایه تابعی است که صعودی بوده و همواره براساس شرایط پارامترهای مسئله دارای برد  $[0, 1]$  باشد. در این روند و در جستجوهای انجام شده یک تابع مثلثاتی صعودی به جای ضریب ثابت  $\rho$ ، در رابطه (۴) پیشنهاد می‌شود که باعث می‌شود نتایج بهتری برای الگوریتم به دست آید که در بخش بعدی به طور مبسوط به آن پرداخته می‌شود.

$$\rho_t = 1 - \rho \cos\left(\frac{\pi t}{3loop}\right) \quad (4)$$

برابر  $\eta_{ij}(t) = ac_{i0} + bc_{0j} - cc_{ij} + d|c_{i0} - c_{0j}|$  در نظر گرفته می‌شود که در آن  $a, b, c$  و  $d$  اعداد ثابتی هستند و به وسیله کاربر تنظیم می‌گردند و  $C_{ij}$  هزینه بین دو گره  $i$  و  $j$  است. در جدول ۱ مقادیر مورد آزمایش و بهترین مقادیر پیدا شده برای این پارامترها نشان داده شده است. در این جدول با در نظر گرفتن ۳ پارامتر  $a, b, c$  و  $d$  به طور ثابت به مقدار ۱ ابتدا با آزمایش مثال‌ها و مقایسه جواب‌ها، بهترین مقدار  $a$  به دست می‌آید که در اینجا  $1/5$  به دست آمده است. سپس با در نظر گرفته شدن  $a$  به مقدار  $1/5$  و  $c$  و  $d$  به مقدار ۱ بهترین مقدار  $b$  به دست می‌آید و در نهایت بهترین مقادیر  $c$  و  $d$  نیز به همین ترتیب پیدا می‌شود.

جدول ۱. تحلیل چهار پارامتر

پارامتر	مقادیر در نظر گرفته شده	بهترین مقدار
$a$	$[0.05, 0.1, 0.2, 0.5]$	$1/5$
$b$	$[0.05, 0.1, 0.2, 0.5]$	$1/5$
$c$	$[0.05, 0.1, 0.2, 0.5]$	$2/5$
$d$	$[0.05, 0.1, 0.2, 0.5]$	$2/5$

#### • ضریب تبخیر فرمون:

این پارامتر در انتهای هر تکرار الگوریتم ACS به کار گرفته می‌شود که به صورت یک مقدار ثابت  $\rho$  نشان داده می‌شود. ضریب  $\rho$  باعث می‌شود که مقدار فرمون در تمامی یال‌ها به مقداری معین کاهش پیدا کنند. توجه به این نکته ضروری است که در این مقاله این ضریب به علت تحلیل درست از الگوریتم، تغییر می‌کند و از ضریبی ثابت به ضریبی متغیر تبدیل می‌شود. بنابراین با تغییر وضعیت الگوریتم، این ضریب تغییر می‌کند و باعث می‌شود که الگوریتم از کارایی بیشتری برخوردار شود. ضریب ثابت فرمون در هنگام اجرای الگوریتم، تفاوتی بین اینکه جواب‌ها دارای چه کیفیتی هستند قائل نمی‌شود، علی‌رغم اینکه جواب‌ها در ابتدای الگوریتم، دقت کمتری دارند و اطلاعات ابتکاری مسئله از قدرت بیشتری برای جهت‌گیری الگوریتم برخوردار هستند، اما هر چه به تعداد تکرار الگوریتم افزوده می‌شود، به علت جستجوهای انجام شده و استفاده مناسب از مقدار فرمون ریخته شده روی یال‌ها، جواب‌ها به کیفیت بالاتری می‌رسند؛ بنابراین بهتر است که این ضریب به صورت تابعی



#### ۴- تحلیل پارامترها

در این بخش تعدادی از پارامترهای الگوریتم پیشنهادی که تأثیر بسزایی در همگرایی الگوریتم دارند، در نظر گرفته می‌شوند و مورد تجزیه و تحلیل قرار می‌گیرند. برای این منظور برای هر پارامتر مورد نظر مقادیر مختلفی آزمایش می‌شود تا به این وسیله مقادیر نزدیک به بهینه آنها یافت شود. نتایجی که در جداول بعدی نشان داده شده است، بهترین مقادیری است که الگوریتم توانسته در ده بار تکرار الگوریتم به دست آورد. از طرف دیگر در هنگامی که یک پارامتر مورد بررسی قرار می‌گیرد، بقیه پارامترها ثابت در نظر گرفته می‌شوند تا به این ترتیب بتوان با دقت بیشتری پارامتر مورد آزمایش را بررسی کرد.

برای بررسی تأثیر پارامترهای روش الگوریتم جمعیت مورچگان بر جواب مسئله، سه پارامتر این الگوریتم شامل  $\alpha$ ،  $\beta$  و  $\text{loop}$  به طور مجزا در نظر گرفته می‌شوند. باید توجه کرد که  $\alpha$  و  $\beta$  به ترتیب توان‌های میزان تأثیر فرمون و اطلاعات ابتکاری مسئله در انتخاب یال بعدی به وسیله مورچه‌ها و  $\text{loop}$  میزان تکرار الگوریتم است.

در جدول ۲ پنج مثال استاندارد VRPPD در نظر گرفته شده است. این مثال‌ها دارای دامنه خوبی از گره‌ها شامل ۵۰ تا ۱۹۹ گره می‌باشند. از طرف دیگر تعداد وسایل نقلیه مورد استفاده و همچنین تغییرات پارامتر  $\alpha$  در ستون‌های سوم تا هشتم جدول نشان داده شده است. در نهایت ستون نهم این جدول بهترین مقادیری که تاکنون برای این مثال‌ها به وسیله الگوریتم‌های دیگر به دست آمده است، را نشان می‌دهد. در آزمایشات جدول مربوطه، مقدار پارامتر  $\beta$ ، برابر ۵ و تعداد اجرای الگوریتم، به عنوان شرط پایانی، برابر تعداد گره‌های هر مسئله در نظر گرفته شده است. همان‌طور که مشخص است بهترین جواب برای مقادیر مختلف  $\alpha$  مقدار ۱/۵ است. همچنین اگر جواب‌های مسایل برای مقادیر  $\alpha$ ، بزرگتر یا مساوی یک در نظر گرفته شود آنگاه رابطه معناداری بین افزایش مقدار  $\alpha$  و کیفیت جواب‌های به دست آمده وجود دارد. در این مسایل هرچه مقدار  $\alpha$  افزایش می‌یابد مقادیر جواب‌ها نیز افزایش پیدا می‌کند. بنابراین با افزایش  $\alpha$  در ۵ مثال مربوطه، کیفیت جواب‌ها کاهش می‌یابد.

در جدول ۳ میزان تغییرات  $\beta$  نشان داده شده است. در اینجا مقدار  $\alpha$  برابر ۱/۵، به علت به دست آمدن بهترین جواب در

$\text{loop}$ : تعداد کل تکرار حلقه الگوریتم که به عنوان شرط پایانی الگوریتم در نظر گرفته می‌شود.

$t$ : نشان‌دهنده شماره تکرار الگوریتم که حداقل برابر ۱ و حداکثر برابر  $n$  است.

$\rho_t$ : نرخ تبخیر جدید است که با زمان افزایش می‌یابد.

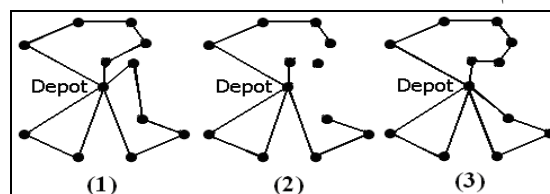
$\rho$ : نرخ تبخیر ثابت برای الگوریتم که همیشه از بازه  $[0, 1]$  در نظر گرفته می‌شود.

باید توجه کرد که چون  $t$  از ۱ تا  $n$  تغییر می‌کند، مقدار  $\pi t / 3n$  بین صفر تا  $\pi / 3$  تغییر کرده و در نتیجه  $\cos(\pi t / 3n)$  تابعی نزولی بین  $[1, 0.5]$  اختیار می‌کند و نهایتاً  $1 - \rho \cos(\pi t / 3n)$  بنا بر ضرب  $\rho$  تابعی صعودی در  $[0, 1]$  می‌باشد.

#### ۳-۲- الگوریتم بهبود دهنده سه گانه

برای به دست آوردن جواب‌های با کیفیت بالاتر الگوریتم پیشنهادی با الگوریتم جستجوی محلی بهبوددهنده سه گانه ترکیب می‌شود. این الگوریتم جستجوی محلی در هنگامی که جواب بهتری برای الگوریتم نسبت به تکرارهای قبلی به دست آید، فعال می‌گردد. علت این که چرا جستجوی محلی در موقعیت گفته شده فعال می‌شود این است که وقتی یک جواب بهتر نسبت به تکرارهای قبلی به دست می‌آید، امکان دارد که در همسایگی این جواب، جواب بهتر دیگری وجود داشته باشد که می‌توان با جستجوهای بیشتر در همسایگی جواب جاری به آن جواب دست پیدا کرد.

این روش، همان‌طور که در شکل ۵ نشان داده شده است، براساس حذف کردن سه یال از تور و دوباره متصل کردن آن سه یال از طریقی دیگر، کار می‌کند. باید توجه کرد که برای ایجاد کردن دوباره تور چندین راه وجود دارد اما فقط حالتی پذیرفته می‌شود که در محدودیت‌های مسئله صدق کند و تور جدید مقدار بهتری را نسبت به جواب قبلی به دست آورد. عمل حذف سه کمان و متصل کردن دوباره به طور متوالی تا جایی ادامه پیدا می‌کند که دیگر هیچ حرکت بهبوددهنده جدیدی برای الگوریتم یافت نشود.



شکل ۵. حرکت بهبوددهنده سه گانه

نتایج الگوریتم در این جدول به این نکته اشاره دارد که علی‌رغم اینکه تصور می‌شود که اجرای بیشتر الگوریتم در مدت طولانی حتماً سبب به‌دست آمدن جواب‌های بهتر می‌شود، اما در اینجا این فرضیه صادق نمی‌باشد. علت این امر را باید در تعداد زیاد متغیرها و پارامترهای موجود در الگوریتم جمعیت مورچگان جستجو کرد. بنابراین برای به‌دست آوردن جواب‌های باکیفیت‌تر برای این مسئله، بهتر است که تعداد آزمایش الگوریتم را افزایش داد تا اینکه زمان اجرای الگوریتم زیاد شود، زیرا این عمل دارای احتمال بیشتر برای رسیدن به جواب‌های باکیفیت‌تر است. بنابراین برای آزمایشات بعدی مقدار پارامترهای  $\alpha$ ،  $\beta$  و  $\text{loop}$  به ترتیب  $1/5$ ،  $2/5$  و  $5n/2$  در نظر گرفته شود.

جدول ۲ و مقدار تکرار الگوریتم برابر تعداد گره‌های هر مسئله در نظر گرفته شده است. همچنین در این جدول مقدار تغییر پارامتر  $\beta$  ما بین ۱ تا ۴ در نظر گرفته می‌شود. با مقایسه جواب‌های به‌دست آمده می‌توان نتیجه گرفت که بهترین و بدترین مقدار  $\beta$  به ترتیب برابر  $2/5$  و ۱ است. همچنین تقریباً هر چه مقدار  $\beta$  افزایش پیدا می‌کند کیفیت جواب‌های الگوریتم بهبود پیدا می‌کند. سرانجام در جدول ۴ میزان تغییرات تکرارهای الگوریتم نشان داده شده است. در این جدول تعداد تکرارها به عنوان مضربی از تعداد گره‌های هر مسئله در نظر گرفته شده است. این مضارب به ترتیب  $n/2$ ،  $3n/2$ ،  $2n$ ،  $5n/2$  و  $3n$  می‌باشند. همچنین برای این جدول  $\alpha = 1.5$  و  $\beta = 2.5$  در نظر گرفته می‌شود.

جدول ۲. جواب الگوریتم برای  $\alpha$  های متفاوت

نمونه	اندازه	$\gamma$	$0/5$	۱	$1/5$	۲	$2/5$	BKS [22]
CMT1X	۵۰	۳	۴۸۹/۴۲	۴۷۸/۳۱	۴۷۷/۱۲	۴۸۱/۵۲	۴۸۹/۰۱	۴۷۲
CMT2X	۷۵	۷	۷۱۴/۴۱	۷۱۰/۱۱	۷۱۲/۵۱	۷۳۱/۴۵	۷۵۱/۶۱	۶۹۵
CMT3X	۱۰۰	۵	۷۴۱/۵۶	۷۳۱/۷۲	۷۲۸/۷۱	۷۳۲/۸۶	۷۵۲/۱۸	۷۲۱
CMT4X	۱۵۰	۷	۹۲۱/۴۳	۹۱۲/۶۱	۹۱۰/۹۳	۹۱۵/۴۳	۹۳۴/۵۲	۸۸۰
CMT5X	۱۹۹	۱۱	۱۲۲۰/۰۱	۱۱۴۵/۷۳	۱۱۳۵/۱۵	۱۱۹۹/۸۲	۱۲۹۲/۴۵	۱۰۹۸

جدول ۳. جواب الگوریتم برای  $\beta$  های متفاوت

نمونه	۱	$1/75$	$2/5$	$3/25$	۴
CMT1X	۴۸۲/۵۲	۴۷۷/۶۲	۴۷۴/۳۲	۴۷۴/۳۲	۴۸۷/۴۲
CMT2X	۷۱۲/۶۵	۷۳۰/۰۴	۷۰۰/۵۳	۶۹۹/۵۴	۷۴۵/۵۲
CMT3X	۷۳۹/۴۳	۷۲۹/۶۱	۷۲۴/۴۲	۷۳۱/۴۳	۷۴۹/۵۱
CMT4X	۹۱۹/۵۳	۹۰۳/۳۲	۸۹۰/۹۱	۹۱۳/۶۲	۹۴۲/۸۳
CMT5X	۱۲۱۵/۶۹	۱۱۳۱/۸۹	۱۱۰۸/۴۱	۱۱۹۱/۶۲	۱۲۹۳/۱۲

جدول ۴. جواب الگوریتم برای  $\text{loop}$  های متفاوت

نمونه	$n/2$	$3n/2$	$2n$	$5n/2$	$3n$
CMT1X	۴۸۰/۶۵	۴۷۶/۱۴	۴۷۳/۱۲	۴۷۳/۰۱	۴۷۲
CMT2X	۷۱۴/۱۵	۷۱۴/۸۱	۷۰۱/۴۱	۶۹۵	۶۹۸/۳۱
CMT3X	۷۳۶/۲۱	۷۲۷/۶۳	۷۲۵/۱۸	۷۲۲/۹۲	۷۲۴/۱۱
CMT4X	۹۰۹/۴۲	۹۰۲/۹۷	۸۸۵/۸۲	۸۸۵/۹۸	۸۸۶/۵۲
CMT5X	۱۱۹۵/۳۲	۱۱۲۳/۵۴	۱۱۰۹/۶۵	۱۱۰۷/۴۳	۱۱۰۵/۲۴

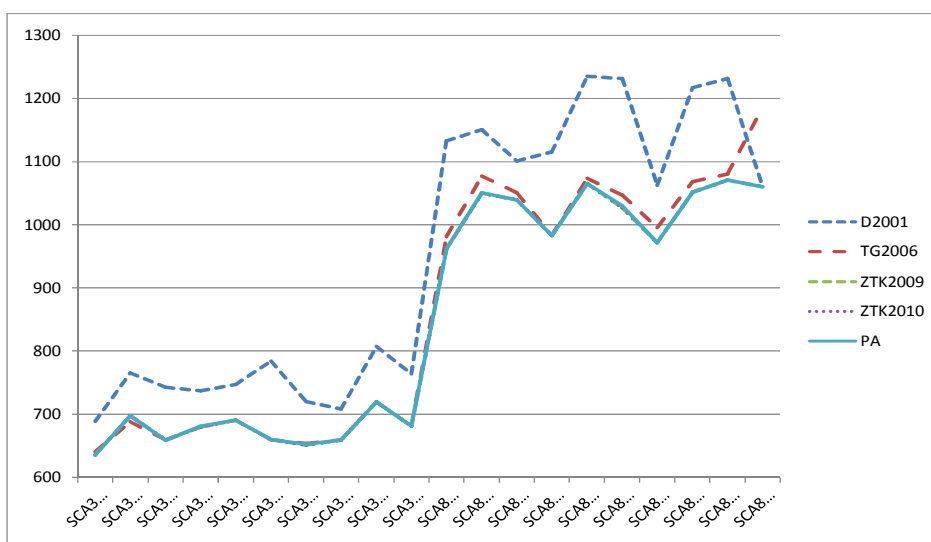
## ۵- محاسبات عددی

تمام کدهای این برنامه به زبان MATLAB 7 نوشته شده است و کامپیوتری که این برنامه‌ها روی آن اجرا شده از نوع AMD 3.5 GHz با یک گیگا بایت حافظه است. همچنین سیستم عاملی که روی این کامپیوتر نصب شده است ویندوز XP بوده است. در جدول ۵، ۲۰ مثال از روی مثال‌های استاندارد مسئله مسیریابی وسیله نقلیه همراه با دریافت و تحویل همزمان کالا که توسط دثلوف در سال ۲۰۰۱ ارائه شد، در نظر گرفته و نتایج الگوریتم‌های فراابتکاری انجام شده روی آن نشان داده شده است. در این جدول مثال‌ها در ستون اول نشان داده شده است. شماره سمت راست مثال‌ها فقط برای شناخته شدن هر مثال از مثال‌های دیگر آورده شده است و نشان‌دهنده تعداد گره‌ها و یا تعداد وسایل نقلیه نمی‌باشد. تعداد گره‌های هر مثال در ستون دوم آورده شده است که شامل گره انبار نیز می‌باشند. در ستون سوم و چهارم تعداد وسایل نقلیه مورد استفاده برای هر مثال (v) و همچنین مقدار جواب‌هایی که دثلوف در سال ۲۰۰۱ برای جواب‌ها به‌دست آورده، نشان داده شده است (Dethloff, 2001). توجه به این نکته ضروری است که در این روش، تعداد وسایل نقلیه مورد استفاده برای ۲۰ مثال مشخص نشده است. از طرف دیگر همان‌طور که ملاحظه می‌کنید ستون‌های پنجم و ششم تعداد وسایل نقلیه و نتایج روش مورد استفاده به‌وسیله تانگ و همکارانش را نشان می‌دهد (Tang and Galvao, 2006) به‌طوری‌که در ۱۰ مثال اول تعداد وسیله نقلیه برابر ۴، برای دو مثال SCA8-2، SCA8-7 برابر ۱۰ و برای بقیه موارد ۹ در نظر گرفته شده است. نکته‌ای که بعد از مقایسه این دو روش می‌توان به آن اشاره کرد این است که الگوریتم تانگ و همکارانش نسبت به الگوریتم دثلوف از کارایی بالاتری برخوردار بوده به‌طوری‌که نتوانسته است که در تمامی مثال‌ها جواب‌های بهتری را به‌دست آورد. در ستون‌های هفتم و هشتم تعداد وسایل نقلیه و بهترین جواب پیدا شده توسط الگوریتم جستجوی ممنوع و جستجوی محلی راهنما نشان داده شده است. این الگوریتم که توسط زاچاردیس و همکارانش ارائه شده (Zachariadis, Tarantilis and Kiranoudis, 2009)، از کارایی بسیار چشمگیری برخوردار است، به‌طوری‌که نسبت به الگوریتم دثلوف در تمامی مثال‌ها دارای کیفیت بالاتری بوده و در مقایسه با الگوریتم ارائه شده توسط تانگ و همکارانش دارای رقابتی

نزدیک است، به‌طوری‌که در ۲۰ مثال ارائه شده فقط در مثال SCA3-1 نتوانسته است که کیفیت جواب را افزایش دهد، ولی در بقیه مثال‌ها جواب‌هایی بهتر یا برابر به‌دست آورده است. به بیانی دقیق‌تر این الگوریتم در ۸ مثال دارای جواب‌هایی مشابه و در ۱۱ مثال دیگر دارای جواب‌هایی باکیفیت‌تر می‌باشد. در ستون‌های نهم و دهم روش دیگری به نام الگوریتم حافظه تطابقی همین نویسنده به همراه تعداد وسایل نقلیه معین برای هر مثال که در سال ۲۰۱۰ ارائه شده، نشان داده شده است (Zachariadis, Tarantilis, and Kiranoudis, 2009). با توجه به نتایج به‌دست آمده یکی از بهترین روش‌هایی که تاکنون برای این مسئله ارائه شده است همین روش است که نتوانسته است بهترین جواب‌ها را در جدول ۵ به‌دست آورد. در مقایسه این روش با سایر روش‌های ارائه شده، به خصوص روش ترکیبی فراابتکاری همین نویسنده در سال ۲۰۰۹، این روش نتوانسته است که در سه مثال SCA3-0، SCA8-1، SCA8-7 جواب‌های بهتری را ارائه کند و در بقیه مثال‌ها مقادیر برابر با الگوریتم قبلی را به‌دست آورد. توجه به این نکته ضروری است که با توجه به دو روش پیشنهاد شده توسط این نویسنده و جواب‌های با کیفیتی که این دو روش به‌دست آورده است، در روش پیشنهاد (PA)، که خصوصیات آن در ستون‌های ۱۱ و ۱۲ آورده شده است، تعداد وسایل نقلیه همانند این دو روش معین شده و برای ۱۰ مثال اول، ۴ و برای ۱۰ مثال دوم، ۹ در نظر گرفته شده است. همچنین با توجه به قابلیت خوبی که روش ACS در مسایل مسیریابی دارد، الگوریتم نتوانسته است که نتایج خوبی را برای این مسایل به‌دست آورد و رقابت خوبی را با سایر روش‌های فراابتکاری داشته باشد. به‌علاوه الگوریتم در مقایسه با بهترین روشی که در این جدول نشان داده شده است، نتوانسته است که در دو مثال SCA3-2 و SCA3-0 جواب‌های بهتری را به‌دست آورد و در ۱۲ مثال جواب‌های یکسانی را با جواب‌های ارائه شده در ستون ۱۰ ارائه کند. درحالی‌که فقط در ۶ مثال باقی‌مانده الگوریتم نتوانسته است که جواب‌های باکیفیت‌تری را تولید کند. بنابراین با توجه به این آزمایشات می‌توان نتیجه گرفت که الگوریتم پیشنهادی نسبت به الگوریتم‌های دثلوف، تانگ و زاچاردیس ۲۰۰۹ دارای کارایی بیشتری است، اما این الگوریتم در مقایسه با الگوریتم دوم ارائه شده، زاچاردیس علی‌رغم اینکه دارای رقابت خوبی بوده است اما کیفیت کمتری دارد.

جدول ۵. مقایسه الگوریتم پیشنهادی با سایر روش‌های فراابتکاری

PA	(V)	ZTK2010	(V)	ZTK2009	(V)	TG2006	(V)	D2001	(V)	تعداد گره‌ها	مثال
۸۳۵/۱۳	۴	۸۳۵/۶۲	۴	۸۳۶/۰۶	۴	۶۴۰/۵۵	۴	۶۸۹/۰	مشخص نیست	۵۰	SCA3-0
۶۹۷/۸۴	۴	۶۹۷/۸۴	۴	۶۹۷/۸۴	۴	۶۸۷/۸۴	۴	۷۵۶/۶	مشخص نیست	۵۰	SCA3-1
۶۵۸/۸۱	۴	۶۵۹/۳۴	۴	۶۵۹/۳۴	۴	۶۵۹/۳۴	۴	۷۴۲/۸	مشخص نیست	۵۰	SCA3-2
۶۸۱/۳۶	۴	۶۸۰/۰۴	۴	۶۸۰/۰۴	۴	۶۸۰/۰۴	۴	۷۳۷/۲	مشخص نیست	۵۰	SCA3-3
۶۹۰/۵۰	۴	۶۹۰/۵۰	۴	۶۹۰/۵۰	۴	۶۹۰/۵۰	۴	۷۴۷/۱	مشخص نیست	۵۰	SCA3-4
۶۶۰/۱۱	۴	۶۵۹/۹۰	۴	۶۵۹/۹۰	۴	۶۵۹/۹۰	۴	۷۸۴/۴	مشخص نیست	۵۰	SCA3-5
۶۵۲/۱۷	۴	۶۵۱/۰۹	۴	۶۵۱/۰۹	۴	۶۵۳/۸۱	۴	۷۲۰/۴	مشخص نیست	۵۰	SCA3-6
۶۵۹/۱۷	۴	۶۵۹/۱۷	۴	۶۵۹/۱۷	۴	۶۵۹/۱۷	۴	۷۰۷/۹	مشخص نیست	۵۰	SCA3-7
۷۱۹/۴۷	۴	۷۱۹/۴۷	۴	۷۱۹/۴۷	۴	۷۱۹/۴۷	۴	۸۰۷/۲	مشخص نیست	۵۰	SCA3-8
۶۸۱/۰۰	۴	۶۸۱/۰۰	۴	۶۸۱/۰۰	۴	۶۸۱/۰۰	۴	۷۶۴/۱	مشخص نیست	۵۰	SCA3-9
۹۶۱/۵۰	۹	۹۶۱/۵۰	۹	۹۶۱/۵۰	۹	۹۸۱/۴۷	۹	۱۱۳۲/۹	مشخص نیست	۵۰	SCA8-0
۱۰۵۰/۹۰	۹	۱۰۴۹/۶۵	۹	۱۰۵۰/۲۰	۹	۱۰۷۷/۴۴	۹	۱۱۵۰/۹	مشخص نیست	۵۰	SCA8-1
۱۰۳۹/۶۴	۹	۱۰۳۹/۶۴	۹	۱۰۳۹/۶۴	۹	۱۰۵۰/۹۸	۱۰	۱۱۰۰/۸	مشخص نیست	۵۰	SCA8-2
۹۸۳/۳۴	۹	۹۸۳/۳۴	۹	۹۸۳/۳۴	۹	۹۸۳/۳۴	۹	۱۱۱۵/۶	مشخص نیست	۵۰	SCA8-3
۱۰۶۵/۴۹	۹	۱۰۶۵/۴۹	۹	۱۰۶۵/۴۹	۹	۱۰۷۳/۴۶	۹	۱۲۳۵/۴	مشخص نیست	۵۰	SCA8-4
۱۰۳۰/۱۱	۹	۱۰۲۷/۰۸	۹	۱۰۲۸/۰۸	۹	۱۰۴۷/۲۴	۹	۱۲۳۱/۶	مشخص نیست	۵۰	SCA8-5
۹۷۱/۸۲	۹	۹۷۱/۸۲	۹	۹۷۱/۸۲	۹	۹۹۵/۵۹	۹	۱۰۶۲/۵	مشخص نیست	۵۰	SCA8-6
۱۰۵۱/۶۵	۹	۱۰۵۱/۲۸	۹	۱۰۵۲/۱۷	۹	۱۰۶۸/۵۶	۱۰	۱۲۱۷/۴	مشخص نیست	۵۰	SCA8-7
۱۰۷۱/۱۸	۹	۱۰۷۱/۱۸	۹	۱۰۷۱/۱۸	۹	۱۰۸۰/۵۸	۹	۱۲۳۱/۶	مشخص نیست	۵۰	SCA8-8
۱۰۶۰/۵۰	۹	۱۰۶۰/۵۰	۹	۱۰۶۰/۵۰	۹	۱۱۸۵/۶۰	۹	۱۰۶۰/۵	مشخص نیست	۵۰	SCA8-9



شکل ۶. مقایسه الگوریتم پیشنهادی با سایر روش‌های فراابتکاری

طرف دیگر این الگوریتم، اگر چه در هفت مثال باقیمانده نتوانسته است که به بهترین جواب دست یابد، اما توانسته است که تنها با دو درصد خطای نسبی به جواب‌های نسبتاً خوبی دست یابد. باید توجه کرد که مثال‌های نشان داده شده در جدول ۶ را می‌توان از نظر تعداد مشتری به سه دسته کمتر از ۵۰، ۵۰ تا ۱۰۰ و بالاتر از ۱۰۰ گره تقسیم‌بندی کرد. بنابراین می‌توان نتایج الگوریتم را روی این سه دسته به‌طور مجزا مورد بررسی قرار داد. در دسته اول الگوریتم دارای موفقیت نسبتاً خوبی بوده و نتوانسته است که در نیمی از مثال‌ها به بهترین جواب‌هایی که تاکنون یافت شده است، دست پیدا کند.

این نشان می‌دهد که الگوریتم در مثال‌های کوچک، از کارایی نسبتاً خوبی برخوردار است. در دسته دوم الگوریتم، کارایی ضعیف‌تری نسبت به دسته اول دارد و از میان شش مثال، فقط در یکی از مثال‌ها نتوانسته است به بهترین جواب‌ها دست پیدا کند.

باید توجه کرد که اگر چه الگوریتم در این دسته از مثال‌ها، نسبت به دسته اول نتوانسته است که به جواب‌های بسیار خوبی دست پیدا کند، اما همان‌طور که گفته شد، نتوانسته است که این جواب‌ها را فقط با ۲ درصد متوسط خطای نسبی تخمین بزند. نهایتاً در دسته سوم، الگوریتم از کارایی بسیار خوبی برخوردار است و نسبت به دو دسته قبلی از کیفیت بالاتری برخوردار است و از چهار مثال موجود نتوانسته است که در دو مثال به بهترین جواب‌هایی که تاکنون به‌دست آمده است، دست یابد و در دو مثال دیگر کیفیت جواب‌ها را ارتقا دهد.

بنابراین با توجه به نتایج به‌دست آمده، می‌توان نتیجه گرفت که اگرچه وقتی که تعداد گره‌ها افزایش می‌یابد، تعداد جواب‌های شدنی برای این مسئله به‌طور نمایی رشد می‌کند و سختی مسئله به مقدار قابل توجهی رشد می‌کند، اما الگوریتم کارایی خوبی از خود نشان می‌دهد و نسبت به سایر الگوریتم‌های قبلی از قدرت بیشتری برای یافتن جواب‌های خوب برخوردار است. در نتیجه و به‌طور کلی، می‌توان به این نکته اشاره کرد که الگوریتم از کارایی خوبی برای همه مثال‌ها برخوردار است، اگرچه که برای مثال‌های بزرگ این کارایی از مقدار بیشتری برخوردار است.

از طرف دیگر برای اینکه روند الگوریتم‌های ارایه شده در جدول ۵ برای مسایل VRP\_SPD به‌طور واضح‌تر نشان داده شود، در شکل ۶ نتایج آنها آورده شده است. باید توجه کرد که در این شکل محور افقی نشان‌دهنده ۲۰ مثال مورد آزمایش است درحالی‌که محور عمودی نشان‌دهنده جواب الگوریتم‌های مورد نظر برای آنها است.

توجه به این نکته ضروری است که در سمت راست شکل و براساس نامگذاری انجام شده روی الگوریتم‌ها در جدول ۵، خطوط نشان‌دهنده الگوریتم‌ها در شکل آورده شده است. این شکل نشان می‌دهد که الگوریتم پیشنهادی (خط ممتد در شکل) نتوانسته است که جواب‌های بسیار خوبی به‌دست آورد و جواب‌های نزدیکی را با دو الگوریتم زچاردیس و همکارانش در سال‌های ۲۰۰۹ و ۲۰۱۰ به‌دست آورد. به‌علاوه این الگوریتم نسبت به دو الگوریتم تانگ و دثلوف به جواب‌های بهتری دست پیدا کرده است.

همچنین برای اینکه کارایی الگوریتم برای مثال‌های کوچک، متوسط و بزرگ با دقت بیشتری مورد بررسی قرار گیرد، الگوریتم پیشنهادی روی دسته دوم از مثال‌ها مورد آزمایش قرار گرفته است. این مثال‌ها، نشان داده شده در جدول ۶، مثال‌هایی هستند که توسط نگی و سلحی در سال ۲۰۰۵ مورد استفاده قرار گرفتند (Nagy and Salhi, 2005). باید اضافه کرد که این مثال‌ها دارای بازه مناسبی از تعداد مشتری‌ها هستند و مقادیری از ۵۵ تا ۱۹۹ مشتری را شامل می‌شوند. از طرف دیگر، تعداد وسایل نقلیه مورد استفاده در این مثال‌ها بازه‌ای شامل ۳ تا ۱۰ وسیله نقلیه می‌باشد. خصوصیات کامل این مثال‌ها شامل تعداد مشتری، تعداد وسیله نقلیه، بهترین جواب تاکنون یافت شده برای مسئله و همچنین جواب الگوریتم پیشنهادی برای هر کدام از آنها در جدول ۶ نشان داده شده است.

با بررسی جواب‌های به‌دست آمده از جدول ۶ می‌توان به این نتایج رسید که الگوریتم از کارایی بسیار خوبی برای حل مسایل برخوردار است و از ۱۴ مثال نتوانسته است که در ۵ مثال به بهترین جوابی که تاکنون به‌دست آمده است، دست پیدا کند. به‌علاوه این الگوریتم نتوانسته است که در دو مثال با تعداد گره ۱۹۹، مقدار بهترین جواب به‌دست آمده تاکنون را ارتقا دهد. از

مناسبی برای بهبود هرچه بیشتر الگوریتم است. کاربردی کردن این ایده‌ها و همچنین گسترش این روش به نسخه‌های دیگر این مسئله مانند مسئله مسیریابی وسیله نقلیه همراه با دریافت و تحویل کالا به صورت مخلوط و غیرهمزمان و نیز مسئله مسیریابی وسیله نقلیه که تحویل کالا باید زودتر از دریافت کالا صورت گیرد، به مقاله‌های بعدی موکول می‌شود.

#### ۷- پی‌نوشت‌ها

1. Vehicle Routing Problem (VRP)
2. Vehicle Routing Problem with Simultaneously Pickup and Delivery (VRP\_SPD)
3. Ant Colony Optimization (ACO)
4. Ant Colony System (ACS)
5. Vehicle Routing Problem with Pickup and Delivery (VRPPD)
6. Min
7. Halse
8. Vehicle Routing Problem with Backhauls (VRPB)
9. Dethloff
10. Salhi
11. Nagy
12. Golden
13. Casco
14. Tang
15. Galvao
16. Beasley
17. Gillet
18. Miller
19. Angelelli
20. Gendreau
21. Crispim
22. Chen
23. Wu
24. Bianchessi
25. Gajpal
26. Ai
27. Mingyong
28. Subramanian
29. Zachariadis
30. Dorigo
31. Ganbardella

#### ۸- مراجع

- سپهری، محمدهدی و حسینی مطلق، سیدمهدی (۱۳۸۷) "مسیریابی بهینه سیستم‌های حمل و نقل در انبارهای

جدول ۶. نتایج الگوریتم پیشنهادی و مقایسه آن با بهترین

جواب‌های به‌دست آمده

مثال	تعداد گره	وسایل نقلیه	PA	بهترین جواب به‌دست آمده
CMT1X	۵۰	۳	۴۶۶/۷۷	۴۶۶/۷۷
CMT1Y	۵۰	۳	۴۶۶/۷۷	۴۶۶/۷۷
CMT2X	۷۵	۶	۶۷۹/۳۲	۶۶۸/۷۷
CMT2Y	۷۵	۶	۶۷۹/۳۲	۶۶۳/۲۵
CMT3X	۱۰۰	۵	۷۲۵/۴۵	۷۲۱/۲۷
CMT3Y	۱۰۰	۵	۷۲۵/۴۵	۷۲۱/۲۷
CMT12X	۱۰۰	۵	۶۵۳/۲۱	۶۴۴/۷۰
CMT12Y	۱۰۰	۵	۶۵۳/۲۱	۶۵۹/۵۲
CMT11X	۱۲۰	۴	۸۳۸/۶۶	۸۳۸/۶۶
CMT11Y	۱۲۰	۴	۸۳۸/۶۶	۸۳۰/۳۹
CMT4X	۱۵۰	۷	۸۵۲/۴۶	۸۵۲/۴۶
CMT4Y	۱۵۰	۷	۸۵۲/۳۵	۸۵۲/۳۵
CMT5X	۱۹۹	۱۰	۱۰۲۹/۲۵	۱۰۳۰/۵۵
CMT5Y	۱۹۹	۱۰	۱۰۲۹/۲۵	۱۰۳۰/۵۵

#### ۶- نتیجه‌گیری و جهت‌گیری‌های آینده

در این مقاله روش ترکیبی الگوریتم جمعیت مورچگان برای حل مسئله مسیریابی وسیله نقلیه همراه با دریافت و تحویل کالا به‌طور همزمان ارایه شد که در آن ابتدا چندین اصلاح برای این روش مانند تغییر در ضریب تبخیر فرمون و اطلاعات ابتکاری مسئله پیشنهاد شد و سپس الگوریتم مؤثر جستجوی محلی بهبود دهنده سه‌گانه برای بهبود بیشتر و حرکت از جستجوی سراسری به سمت جستجوی محلی به‌کار گرفته شد. باید توجه کرد که این الگوریتم جواب‌های خوبی برای مثال‌های استاندارد این مسئله به‌دست آورده است و در چهار مثال، بهترین جواب‌هایی که تاکنون به‌دست آمده است را ارتقا داد؛ به‌علاوه این روش رقابت نزدیکی با دیگر روش‌های فراابتکاری ذکر شده، دارد و از کارایی بیشتری نسبت به الگوریتم‌های دثلوف، تانگ و زاچاردیس ۲۰۰۹ دارد.

به نظر می‌رسد که استفاده از روش‌های دیگری مانند روش‌های الگوریتم ژنتیک، شبیه‌سازی آنبلی یا ... برای ترکیب با این الگوریتم جواب‌های باکیفیت‌تری را تولید کند. همچنین استفاده از الگوریتم‌های قوی محلی دیگر مانند لین-کرنیگان راهکار

- Dethloff, J. (2001) "Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up", *Operations Research Spectrum*, Vol. 23, pp. 79-96.
- Dorigo, M. and Gambardella, L. (1997) "Ant colony system: a cooperative learning approach to the traveling salesman problem", *IEEE trans, Evolutionary Computing*, pp. 53-66.
- Ellabib, I., Calamai, P. and Basir, O. (2007): "Exchange strategies for multiple ant colony system", *Information Sciences. an International Journal*, Vol. 177, No. 5, pp. 1248-1264.
- Gajpal, Y. and Abad, P. (2009) "An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup", *Computers & Operations Research*, Vol. 36, pp. 3215-3223.
- Gendreau, M., Laporte, G. and Vigo, D. (1999), "Heuristics for the traveling salesman problem with pickup and delivery", *Computers & Operations Research*, Vol. 23, pp. 501-508.
- Min, H. (1989), "The multiple vehicle routing problem with simultaneous delivery and pick-up points", *Transportation Research*, Vol. 23, pp. 377-386.
- Mingyong, L. and Erbao, C. (2010) "An improved differential evolution algorithm for vehicle routing problem with simultaneous pickups and deliveries and time windows", *Engineering Applications of Artificial Intelligence*, Vol. 23, No. 2, pp. 188-195.
- Nagy, G. and Salhi, S. (2005) "Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries", *European Journal of Operational Research*, Vol. 162, pp. 126-141.
- Salhi, S. and Nagy, G., (1999) "A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling", *Journal of the Operational Research Society*, Vol. 50, pp. 1034-1042.
- Subramanian, A., Drummond, L. M. A., Bentes, C., Ochi, L. S. and Farias, R. (2010) "A parallel heuristic for the Vehicle Routing Problem with Simultaneous Pickup and Delivery", *اتوماتیک*، پژوهشنامه حمل و نقل، سال پنجم، شماره دوم، تابستان ۱۳۸۷، ص ۱۴۴-۱۲۷.
- ربانی، یوسف، سپهری، محمدمهدی و ذگردی، سید حسام‌الدین (۱۳۸۷) "مسئله مسیریابی وسیله نقلیه متصل به حمل و نقل چندوجهی رویکرد یکپارچه"، پژوهشنامه حمل و نقل، سال پنجم، شماره چهارم، زمستان ۱۳۸۷، ص ۳۱۹-۳۰۱.
- Ai, T. J. and Kachitvichyanukul, V. (2009) "A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery", *Computers & Operations Research*, Vol. 36, pp. 1693-1702.
- Angelelli, E. and Mansini, R. (2003) "A branch-and-price algorithm for a simultaneous pick-up and delivery problem" Working Paper, Article presented at the EURO/INFORMS Meeting.
- Beasley, JE. (1983) "Route first-cluster second methods for vehicle routing", *Omega*, Vol. 11, pp. 403-408.
- Bianchessi, N. and Righini, G. (2007) "Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery", *Computers and Operations Research*, Vol. 34, pp. 578-594.
- Catay, B. (2010) "A new saving-based ant algorithm for the Vehicle Routing Problem with Simultaneous Pickup and Delivery", *Expert Systems with Applications*, Vol. 37, pp. 6809-6817.
- Chalasani, P. and Motwani, R. (1999) "Approximated capacitated routing and delivery problems", *SIAM Journal on Computing*, Vol. 28, pp. 2133-2149.
- Chen, J. F. and Wu, T. H. (2006) "Vehicle routing problem with simultaneous deliveries and pickups", *Journal of the Operational Research Society*, Vol. 57, pp. 579-587.
- Crispim, J. and Branda, O. J. (2005) "Metaheuristics applied to mixed and simultaneous extensions of vehicle routing problems with backhauls", *Journal of the Operational Research Society*, Vol. 56, pp. 1296-1302.

- Zachariadis, E. E., Tarantilis, C. D. and Kiranoudis, C. T. (2009) "A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service", Expert Systems with Applications, Vol. 36, pp. 1070-1081.
- Zachariadis, E. E., Tarantilis, C. D. and Kiranoudis, C. T. (2010) "An adaptive memory methodology for the vehicle routing problem with simultaneous pick-ups and deliveries", European Journal of Operational Research, Vol. 202, pp. 401-411.
- Computers & Operations Research, Vol. 37, No. 11, pp. 1899-1911.
- Tang, FA. and Galvao, RD. (2002) "Vehicle routing problems with simultaneous pick-up and delivery service", Journal of the Operational Research Society of India, Vol. 39, pp. 19-33.
- Tang, FA. and Galvao, RD. (2006) "A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service", Computers and Operations Research, Vol. 33, pp. 595-619.



## **An Improved Ant Colony System for Solving the Vehicle Routing Problem with Simultaneous Pickup and Delivery**

*M. Yousefi Khoshbakht, Ph.D. Candidate, Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran.*

*F. Rahmati, Associate Professor, Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran.*

*E-mail: frahmati@aut.ac.ir*

### **ABSTRACT**

The classical vehicle routing problem (VRP) is one of the most important combinatorial optimisation problems, and has received much attention from researchers and scientists. This problem involves routing a fleet of vehicles, each visiting a set of customers such that every customer is visited exactly once and by exactly one vehicle, with the objective of minimizing the total distance travelled by all the vehicles. The vehicle routing problem with simultaneous pickups and deliveries (VRP\_SPD) is an extension to the VRP, where the vehicles are not only required to deliver goods to customers, but also to collect some goods from the customer locations.

Ant Colony Optimisation (ACO) is a probabilistic technique that simulates the ants' food-hunting behaviour, which is used for solving problems that do not have a known efficient algorithm. The Ant Colony System (ACS) is one of the most famous metaheuristic algorithms, which differs from the other ACO methods due to its transition rule and updating pheromone.

This paper proposes an improved ACS, mixed with a 3-opt local search for solving the VRP\_SPD. The proposed method is different to the common ACS with regards to the coefficient pheromone and heuristic information for the ant visibility measure. This process will avoid the premature convergence and generate better solutions. The proposed algorithm is tested on 34 standard instances available in the literature. The computational result shows that the proposed algorithm improves the four best-known solutions of the benchmark problem, and it is competitive with other metaheuristic algorithms for solving the VRP\_SPD.

**Keywords:** Ant Colony System, Simultaneously Pickup and Deliver Goods, Heuristic Information, Vehicle Routing Problem