

یک روش اصلاحی جمعیت مورچگان ترکیب شده با الگوریتم‌های ابتکاری درج و جابه‌جایی برای حل مسئله مسیریابی وسیله‌نقلیه همراه با پنجره‌های زمانی

شاهرود اعظمی، استادیار، دانشگاه آزاد اسلامی، واحد پرنده، تهران، ایران

پیام بصیری، مربی، گروه ریاضی، دانشگاه پیام نور، تهران، ایران

فرزاد دیده‌ور، استادیار، دانشکده ریاضی و علوم کامپیوتر، دانشگاه صنعتی امیرکبیر، تهران، ایران

فرهاد رحمتی، دانشیار، دانشکده ریاضی و علوم کامپیوتر، دانشگاه صنعتی امیرکبیر، تهران، ایران

مجید یوسفی خوشبخت*، دانشگاه آزاد اسلامی، واحد همدان، باشگاه پژوهشگران و نخبگان، همدان، ایران

پست الکترونیکی نویسنده مسئول: khoshbakht@iauh.ac.ir

دریافت: ۱۳۹۴/۰۹/۰۶ - پذیرش: ۱۳۹۵/۰۱/۲۵

چکیده

مسئله مسیریابی وسیله‌نقلیه همراه با پنجره‌های زمانی ($VRPTW$) یکی از مشهورترین مسائل بهینه‌سازی ترکیباتی در حوزه حمل و نقل است. چون این مسئله متعلق به مسائل NP -سخت است، بسیاری از دانشمندان و محققین روش‌های فراابتکاری برای حل آن ارائه داده‌اند. در این مقاله، به علت ضعف‌های موجود در الگوریتم سیستم مورچگان (ACS)، نسخه‌ای اصلاحی از این الگوریتم به نام $HACS$ برای حل مسئله $VRPTW$ ارائه می‌گردد. به منظور افزایش کارایی الگوریتم، دو روش جستجوی همسایه به نام‌های درج و جابجایی مورد استفاده قرار گرفته شده است. این اصلاحات سبب می‌شود که الگوریتم جدید از همگرایی زودرس اجتناب کند و به جواب‌های بسیار خوبی دست پیدا کند. در نهایت برای تست کارایی الگوریتم، تعدادی از مجموعه مثال ۶۵ تایی سالامان در نظر گرفته و نتایج این الگوریتم با دیگر روش‌ها در ادبیات موضوع مقایسه شده است. نتایج نشان می‌دهد که نه تنها الگوریتم پیشنهادی توانسته جواب‌های بسیار خوبی را به دست آورد بلکه هفت عدد از بهترین جواب‌های تاکنون به دست آمده به وسیله الگوریتم $HACS$ حاصل می‌گردد.

واژه‌های کلیدی: مسئله مسیریابی وسیله‌نقلیه همراه با پنجره‌های زمانی، الگوریتم سیستم مورچگان، حرکت درج، حرکت جابجایی، مسائل بهینه‌سازی ترکیباتی

۱- مقدمه

که Q در نظر گرفته می‌شود، تجاوز نکند (ظفری، تشکری هاشمی، یوسفی خوشبخت، ۱۳۸۹).

از طرف دیگر به وجود آمدن مسائل جدید در دنیای واقعی سبب شده است که محدودیت‌های دیگری به مسئله $CVRP$ تحمیل شود و نسخه‌های دیگری از آن مانند مسئله مسیریابی وسیله‌نقلیه باز (وسائل‌نقلیه بعد از انجام مأموریت خود به انبار بازنگردند و مسیر خود را در همان گره‌های انتهایی به پایان برسانند) (یوسفی خوشبخت، دیده‌ور، رحمتی و صدیق‌پور، ۱۳۹۱)، مسئله مسیریابی وسیله‌نقلیه همراه با دریافت و تحویل کالا (هر مشتری می‌تواند علاوه بر درخواست دریافت کالا،

مسئله مسیریابی وسیله‌نقلیه ظرفیت‌دار^۱ ($CVRP$) یکی از مسائل استراتژیک در لجستیک و زنجیره تأمین است. در این مسئله تعدادی مشتری در یک منطقه جغرافیایی به همراه یک انبار وجود دارند به طوری که هر یک از مشتری‌ها به میزان خاصی کالا نیاز دارند که باید توسط ناوگانی ثابت از وسائل نقلیه به آن‌ها تحویل گردد. هدف تعیین مجموعه‌ای از تورها برای وسائل نقلیه با کمترین هزینه است به شرط آن‌که: هر وسیله‌نقلیه از انبار شروع به حرکت کرده و در انتها به آن بازگردد، هر مشتری دقیقاً یک‌بار توسط یک وسیله‌نقلیه بازدید شود و کل تقاضای مشتری‌های هر تور از ظرفیت وسیله‌نقلیه،

مراکز توزیع غذا یا خدمات و غیره مورد استفاده قرار می‌گیرد. باید توجه کرد که اگر بازه زمانی ارضای تقاضای هر مشتری محدودیت نداشته باشد آن‌گاه این مسئله تبدیل به مسئله *CVRP* می‌شود که یک مسئله *NP*-سخت^۳ است [Laporte, 2009]. بنابراین مسئله *VRPTW* نیز یک مسئله *NP*-سخت است. پیچیدگی این مسئله سبب می‌شود که نه تنها حل دقیق مسائل آن با اندازه بزرگ در زمانی قابل قبول امکان‌پذیر نباشد بلکه حل دقیق مسائل کوچک آن نیز به زمانی بسیار زیاد احتیاج دارد. از طرف دیگر الگوریتم سیستم مورچگان^۴ (*ACS*) یکی از مشهورترین روش‌های فراابتکاری است که تاکنون بر روی تعدادی از مسائل بهینه‌سازی مورد استفاده قرار گرفته و نتایج نسبتاً خوبی را به دست آورده است. باید توجه کرد که این الگوریتم در مقایسه با الگوریتم مورچگان^۵ (*ACO*) به علت اصلاحات صورت گرفته دارای کارایی بسیار بهتری است. به علت ساختار سخت مسئله *VRPTW*، در این مقاله یک روش ترکیبی-اصلاحی *MACS* برای حل این مسئله ارائه می‌گردد که با الگوریتم‌های جستجوی محلی درج^۶ و جابجایی^۷ ترکیب شده است. به علاوه برای افزایش کارایی الگوریتم از یک قانون تصمیم جدید، تبدیل پارامتر q_0 الگوریتم سیستم مورچگان به یک متغیر و بروزرسانی فرمون بر روی بهترین جواب تاکنون پیدا شده، استفاده می‌گردد. نتایج نشان می‌دهد که این اصلاحات سبب ایجاد یک روش کارایی فراابتکاری برای حل مسئله *VRPTW* شده است.

در این مقاله و در بخش دوم تاریخچه مسئله شرح داده می‌شود در حالی که در بخش سوم ابتدا روش *ACS* مورد بررسی قرار می‌گیرد و سپس روش پیشنهادی با جزئیات بیشتر شرح داده می‌شود. در بخش چهارم به تجزیه و تحلیل نتایج الگوریتم پیشنهادی و الگوریتم‌های دیگر فراابتکاری بر روی مثال‌های استاندارد پرداخته می‌شود و سرانجام در بخش پنجم جمع‌بندی و نتیجه‌گیری ارائه می‌گردد.

۲- تاریخچه مسئله

یکی از پرکاربردترین مسائل در بهینه‌سازی ترکیباتی، مسئله *CVRP* و گسترش‌های آن است. تنوع این‌گونه از مسائل آنقدر زیاد است که دسته‌بندی آن‌ها و بیان حالت‌های مختلفی که در آن رخ می‌دهد، بسیار مشکل و زمان‌گیر است. از زمانی که مسئله مسیریابی وسیله نقلیه بدون ظرفیت^۸ (*VRP*) در سال

درخواست تحویل کالا نیز داشته باشد و می‌تواند این درخواست‌ها را در یک‌بار و یا بیشتر توسط وسایل نقلیه برطرف سازد (یوسفی خوشبخت و رحمتی، ۱۳۹۰)، مسئله مسیریابی وسیله نقلیه ناهمگن (وسایل نقلیه می‌توانند با یکدیگر مشابه نبوده و از نظر ظرفیت با یکدیگر متفاوت باشند) (یوسفی خوشبخت، دیده‌ور، رحمتی و سعادت‌اسکندری، ۱۳۹۱) و مسئله مسیریابی وسیله نقلیه همراه با حمل در بازگشت (بعضی از گره‌ها باید قبل از برخی گره‌های دیگر مورد ملاقات قرار گیرند) (سپهری و ستاک، ۱۳۸۳) به وجود آید.

یکی دیگر از گسترش‌های مسئله *CVRP*، مسئله مسیریابی وسیله نقلیه با پنجره زمانی^۹ (*VRPTW*) است که با ناوگانی از وسایل نقلیه سروکار دارد که از انبار مسیره‌های خود را آغاز می‌کنند و می‌بایست به تعدادی از مشتریان با تقاضاهای متفاوت در یک بازه زمانی مشخص سرویس‌دهی کنند. در این مسئله هزینه c_{ij} و زمان t_{ij} برای پیمایش کمان‌ها و مقدار تقاضای کالای p_i برای هر مشتری i از قبل تعریف شده است. به علاوه هر مشتری باید در پنجره زمانی که از قبل تعریف شده است و محدود به زودترین زمان شروع e_i و دیرترین زمان سرویس l_i است، مورد ملاقات قرار گیرد. وسایل نقلیه‌ای که زودتر از زمان شروع به یک مشتری می‌رسند متحمل یک زمان انتظار اضافی می‌گردند اما این وسایل نقلیه باید مسیره‌های خود را در حداکثر زمان سفر مجاز به اتمام برسانند. هدف در این مسئله یافتن مسیره‌هایی برای وسایل نقلیه با حداقل هزینه است که در آن تمام محدودیت‌ها شامل ظرفیت، زمان سفر و محدودیت پنجره زمانی برای وسایل نقلیه برقرار باشند. باید توجه کرد که مسئله *VRPTW* خود به دو دسته پنجره‌های زمانی سخت و نرم تقسیم‌بندی می‌شود. در نسخه سخت اگر وسیله نقلیه‌ای زودتر از موعد ابتدایی بازه زمانی به یک مشتری برسد باید صبر کند اما نمی‌تواند به هیچ وجه دیرتر از بازه زمانی هر مشتری به آن سرویس‌دهی کند. در نسخه نرم، این مسئله طوری دیگر تعریف می‌شود. در این نسخه سرویس‌دهی خارج از هر بازه به هر مشتری امکان‌پذیر است اما چه در رسیدن زودتر از موعد و چه دیرتر از موعد، یک هزینه جریمه‌ای به تابع هدف مسئله تحمیل می‌شود.

مسئله *VRPTW* که برای اولین بار در دهه ۸۰ میلادی معرفی شد، امروز در مشاغل بسیاری مانند تحویل نامه و روزنامه، تحویل پول از بانک‌ها، مسیریابی اتوبوس مدرسه،

کار می‌کرد که مسئله را تبدیل به یک مسئله اصلی و چندین زیر مسئله شامل فروشنده دوره‌گرد به همراه پنجره‌های زمانی می‌کرد. سپس از روش ژنتیک برای حل مسئله اصلی و یک روش ساده ابتکاری برای حل زیر مسائل فروشنده دوره‌گرد با پنجره‌های زمانی استفاده می‌کرد. به علاوه جواب مسئله با یک تعامل تکراری بین مسئله اصلی و زیر مسائل به دست می‌آمد. نتایج نشان داد که این روش از کارایی مناسبی در مقایسه با روش‌های ابتکاری ورودی برخوردار است. به علاوه توکلی مقدم و همکاران این مسئله را مورد بررسی قرار دادند و گسترشی از این مسئله که علاوه بر کمینه‌سازی هزینه مسافرت سبب می‌شود که میزان فروش بیشینه گردد، را ارائه دادند [Tavakkoli-Moghaddam et al., 2011]. باید توجه داشت که افزایش فروش به علت کارایی و ارائه خدمات مناسب به مشتری‌ها قبل از سایر رقبا اتفاق می‌افتاد. همچنین بعد از آن‌ها یک مدل جدید برای این مسئله ارائه دادند، از روش شبیه‌سازی آنبلی و لینگو^{۱۴} برای حل مسئله استفاده کردند. در این مقاله روش شبیه‌سازی آنبلی بر روی مثال‌های سالامان نیز مورد استفاده قرار گرفت.

یک روش همسایگی تطابقی بزرگ توسط لی^{۱۵} و همکارانش برای گسترشی از مسئله *VRPTW* به نام مسئله مسیریابی وسیله نقلیه ظرفیت دار با پنجره‌های زمانی و تقاضاهای احتمالی ارائه شد [Lei, Laporte, and Guo, 2011]. به علاوه آن‌ها یک مدل برای این مسئله ارائه دادند و سپس مثال‌هایی اصلاحی را از مثال‌های استاندارد سالامان برای این مسئله ایجاد کردند و کارایی روش خود را در مقابل یک روش جستجوی تکراری مورد بررسی قرار دادند. یو^{۱۶} و همکارانش این مسئله را مورد بررسی قرار داده و یک روش بهبودیافته مورچگان را برای گسترشی از این مسئله ارائه کردند [Yu, and Yang, 2011]. در مسئله آن‌ها *VRPTW* برای تعدادی از روزها در نظر گرفته شده بود. به عبارت دیگر مشتریان باید در طی روزهایی متوالی و در یک بازه زمانی مورد بازدید قرار بگیرند. به همین منظور یک ماتریس فرمون چند بعدی برای انباشتن و بروزرسانی فرمون در نظر گرفته شد. همچنین دو عملگر برای بهبود الگوریتم مورد استفاده قرار گرفت. در نهایت این روش بر روی مثال‌های استاندارد این مسئله بکار گرفته و نتایج الگوریتم گزارش شد. آزمایشات نشان داد که الگوریتم از کارایی بسیار مناسبی برخوردار است به

۱۹۵۹ به وسیله دانتزیک^۹ و همکاران برای اولین بار معرفی شد، حالت‌های بسیاری از آن بر اساس کاربردهای متفاوتی که در دنیای واقعی داشتند، از آن مشتق شدند [Eksioglo et al., 2009]. آن‌ها در این مقاله یک فرمول‌بندی و الگوریتم ریاضی برای *VRP* ارائه دادند. به علاوه در مسئله ظرفیت وسایل نقلیه به طور یکسان و برابر با بی‌نهایت و مدت زمان توقف در گره‌ها صفر در نظر گرفته شده بود. هدف تعیین مسیریابی برای خودروها بود که باید هر یک دسته‌ای از مشتریان را در یک گستره جغرافیایی مورد ملاقات قرار دهند. باید توجه داشت که این مسئله همان مسئله چندین فروشنده دوره‌گرد^{۱۱} (*MTSP*) است که در آن محدودیتی برای بازدید از یک مشتری وجود ندارد [Yousefikhoshbakht et al., 2012].

برای اولین بار در سال ۱۹۸۷ کولن^{۱۱} و همکاران اولین روش بهینه‌سازی را برای مسئله *VRPTW* ارائه کردند. روش پیشنهادی آن‌ها یک کران پایین را با استفاده از روش برنامه‌ریزی پویا و تخفیف فضای حالت، ارائه کرد. به علاوه این روش از تصمیمات شاخه‌ای روش تخصیص مسیر-مشتری استفاده می‌کرد و توانست مثال‌هایی با ۱۵ مشتری را حل کند [Kolen, Rinnooy Kan, and Trienkens, 1987]. کهل^{۱۲} و همکارش یک روش بر اساس تخفیف‌سازی لاکرانژ برای این مسئله ارائه کردند. در روش پیشنهادی آن‌ها با تخفیف محدودیتی که سبب می‌گردد هر مشتری سرویس‌دهی شود، مسئله *VRPTW* تبدیل به زیر مسئله‌هایی مانند مسئله کوتاه‌ترین مسیر با محدودیت ظرفیتی و پنجره‌های زمانی می‌شد. برای تست کارایی روش مربوطه، تعدادی از مسائل مشهور در ادبیات موضوع مورد استفاده قرار گرفت.

برای نمونه از روش‌های ابتکاری برای حل مسئله *VRPTW* می‌توان به روش اصلاحی جستجوی ممنوع چن^{۱۳} و همکارانش اشاره کرد که از کارایی بسیار خوبی برخوردار بود [Chen et al., 2007]. هدف ارائه شده در این مسئله علاوه بر کمینه‌سازی هزینه مسیریابی وسایل نقلیه، کاهش تعداد وسایل نقلیه مورد استفاده نیز بود. الگوریتم بر روی مثال‌های سالامان مورد استفاده قرار گرفت و با [Kohl, and Madsen, 1997] مورد مقایسه قرار گرفت. نتایج نشان داد که الگوریتم پیشنهادی دارای کارایی بیشتری نسبت به الگوریتم یاد شده است. در [Cheng, and Wang, 2009] برای حل مسئله *VRPTW* از روش تجزیه استفاده شد. این روش با این روال

۳- روش تحقیق

الگوریتم سیستم مورچگان^{۱۹} (AS) که توسط دوریگو^{۲۰} ارائه شد یکی از اولین نسخه‌های الگوریتم ACO می‌باشد که از غذایابی کلونی مورچه‌های واقعی الهام گرفته شده است [Dorigo, 1992]. طبق آزمایشی ماده غذایی و لانه به وسیله فقط دو راه به هم وصل گردید. مشاهدات نشان داد که هر چه از شروع آزمایش می‌گذرد، مورچه‌های بیشتری راه کوتاه‌تر را انتخاب می‌کنند. در حقیقت علت اینکه مورچه‌ها بعد از یک مدت کم به سمت راه کوتاه‌تر مایل می‌شوند این است که آنان در ابتدا که از لانه شروع به حرکت می‌کنند راه خود را به صورت تصادفی انتخاب می‌کنند و در حین حرکت ماده‌ای شیمیایی، به نام فرمون که تبخیر پذیر است، از خود ترشح می‌کنند. باید توجه کرد که در یک زمان معین تعداد مورچه‌هایی که از راه کوتاه‌تر به منبع غذایی می‌رسند و برمی‌گردند بیشتر از راه بلندتر است و این سبب می‌شود که فرمون بیشتری بر روی راه کوتاه‌تر انباشته شود. در نتیجه بعد از یک مدت کم، وقتی مورچه‌های دیگر به محل تقاطع دو مسیر می‌رسند دیگر راه را نه به صورت تصادفی بلکه براساس میزان انباشتگی فرمون انتخاب می‌کنند و مسیر دارای فرمون بیشتر با احتمال بالاتری انتخاب می‌شود. این آزمایش نشان داد که عمل ساده‌ای که مورچه‌ها به طور مستقل انجام می‌دهند چگونه سبب می‌شود که مورچه‌ها بتوانند راه کوتاه‌تر به منبع غذایی را بیابند. دوریگو این عمل مورچه‌ها را شبیه‌سازی کرد و الگوریتم مورچگان را ارائه داد. باید توجه کرد که در این الگوریتم، وظیفه اصلی هر مورچه مصنوعی مانند همتای طبیعی خود، یافتن کوتاه‌ترین مسیر بین یک جفت گره در یک گراف است.

در سال ۱۹۹۷ دوریگو و گامباردلا^{۲۱} سیستم جمعیت مورچگان را معرفی کردند [Dorigo, and Gambardella, 1997] که تا حد زیادی از AS الهام گرفته شده بود. این الگوریتم به دلیل معرفی مکانیسم‌هایی جدید، نسبت به AS از لحاظ کارایی به بهبودهایی دست یافته است. در این روش در هر مرحله مورچه‌های مصنوعی وظیفه دارند که انتخاب گره بعدی j را از گره جاری i بر اساس فرمون ریخته شده روی یال (i, j) (τ_{ij}) و عکس فاصله یال (i, j) (η_{ij})، انجام دهند. قانون تصمیم برای مورچه k واقع در گره i ، که می‌خواهد یکی از گره‌ها را از مجموعه گره‌های ملاقات نشده N_i^k انتخاب کند از فرمول (۱) به دست می‌آید. باید توجه کرد که این قانون

طوری که چندین بهترین جواب تاکنون پیدا شده توسط این الگوریتم به دست آمد. همچنین یو و همکاران یک روش ترکیبی فراابتکاری شامل روش مورچگان، جستجوی ممنوع و یک الگوریتم جستجوی همسایگی جدید را برای مسئله VRPTW ارائه دادند [Yu, Yang, and Yao, 2011]. در این روش، ابتدا الگوریتم مورچگان مورد استفاده قرار گرفت و سپس از الگوریتم جستجوی محلی برای ارتقای جواب‌ها استفاده شد. هر موقع که کارایی روش رو به نزول می‌رفت و الگوریتم نمی‌توانست جواب‌ها را ارتقا بخشد، الگوریتم جستجوی ممنوع مورد استفاده قرار می‌گرفت. به‌کارگیری این الگوریتم کارا سبب می‌شود که الگوریتم بتواند مناطق دیگری را از فضای شدنی مسئله مورد جستجو قرار دهد و بتواند به جواب‌های بهتری دست یابد. این الگوریتم بر روی مثال‌های سالامان ۵۶ مورد استفاده قرار گرفت. نتایج این الگوریتم در مقایسه با سایر روش‌های مشهور فراابتکاری، کارایی این روش در به دست آوردن جواب‌های باکیفیت در زمان‌های قابل قبول را اثبات کرد. [Balseiro, Loiseau, and Ramonet, 2011] گسترشی دیگری از مسئله VRPTW را ارائه کرد که در آن نوع وابسته به زمان مسئله در نظر گرفته شد. این نسخه در مسائل شهری بسیار استفاده می‌شود زیرا مقدار هزینه پیمایش هر یال وابسته به زمانی است که وسیله نقلیه شروع به حرکت می‌کند. نویسندگان این مقاله بال‌سیرو^{۱۷} و همکاران یک روش ترکیبی مورچگان و ابتکاری ورودی را برای این مسئله ارائه کردند که توانست چندین بهترین جواب تاکنون به دست‌آمده را ارتقا دهد. باید توجه کرد که الگوریتم ابتکاری ورودی در انتهای الگوریتم بکار گرفته می‌شود که الگوریتم مورچگان شاید جواب‌های نشدنی تولید کرده است. ویدال^{۱۸} و همکاران این مسئله را برای اندازه‌های بزرگ در نظر گرفتند تا بتوانند الگوریتم کارایی را برای آن ارائه کنند [Vidal et al., 2012]. روش آن‌ها استفاده از الگوریتم تجزیه به منظور کوچک‌کردن مسئله به مسائل کوچک بود و سپس از الگوریتم ژنتیک ترکیبی با کنترل پراکندگی پیشرفته برای حل آن‌ها استفاده می‌شد. همچنین حرکت‌های تکاملی جدید و یک روش اصلاحی برای تبدیل جواب‌های نشدنی در پنجره‌های زمانی به جواب‌های شدنی در این روش ارائه شد. سرانجام الگوریتم برای گسترش‌های خاصی از این مسئله شامل حالت دوره‌ای، چند انباره و غیره مورد آزمایش قرار گرفت.

رسیدن زودتر وسیله نقلیه از زمان e_j

s_j : زمان سرویس در مشتری j

t_{j0} : زمان سفر از مشتری j به انبار

l_0 : کران بالای زمان برای بازگشت وسایل نقلیه به انبار

به علاوه همان طور که مورچه‌ها بین گره i و j حرکت می‌کنند، مقدار فرمون روی آن را با استفاده از فرمول (۲) بروز می‌کنند. که در آن τ_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 این فرمول و فرمول (۳)، ξ و ρ نرخ تبخیر و پارامتری در دامنه $[0,1]$ می‌باشد که کاهش فرمون روی یال‌ها را تنظیم می‌کند. باید توجه کرد که نتیجه بروز رسانی محلی این است که اگر در هر زمان یک مورچه یال (i, j) را پیمود، آن‌گاه ردپای فرمونش کاهش می‌یابد تا این یال برای مورچه‌های بعدی در تکرارهای آینده کمتر مطلوب باشند. این مطلب موجب تشویق بیشتر استخراج برای یال‌هایی که هنوز ملاقات نشده‌اند، می‌شود. باید توجه کرد که بروز رسانی محلی به اجتناب از موقعیت‌های رکود ضعیف کمک می‌کند.

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

همچنین وقتی مورچه‌ها تورشان را تولید کردند، یال‌های متعلق به بهترین تور بر اساس فرمول (۳) بروز می‌شوند (فرمون‌ریزی سراسری) که در آن C_b هزینه بهترین تور T_b پیدا شده از زمان شروع الگوریتم است. مهم است که توجه شود که بروز رسانی سراسری فقط فرمون را روی یال‌های متعلق به بهترین تور تنظیم می‌کند. این مطلب موجب تشویق مورچه‌ها برای جستجوی همسایگی‌های بهترین تور در تکرارهای بعدی می‌شود.

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

برای به‌کارگیری روش ACS برای مسئله VRPTW چندین اصلاح در نظر گرفته شد تا بتوان کارایی این الگوریتم را افزایش داد و بدین وسیله جواب‌های خوبی را برای مثال‌های استاندارد مسئله یافت. این اصلاحات به شرح زیر می‌باشند.

۱. همان طور که گفته شد انتخاب هر مشتری مانند j

هنگامی که مورچه k ام در گره i قرار دهد طبق فرمول (۱)

صورت می‌گیرد که در آن مشتری‌های کاندید، مجموعه

انتقال جدید به هر دو مورد استخراج و اکتشاف توجه دارد.

$$P_{ij}^k = \begin{cases} 1 & \text{if } q \leq q_0 \ \& \ j = j^k \\ 0 & \text{if } q \leq q_0 \ \& \ j \neq j^k \\ \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum_{j \in N_i^k} \tau_{ij}^\alpha \eta_{ij}^\beta} & q > q_0 \end{cases} \quad (1)$$

در این فرمول، به وسیله پارامترهایی α و β می‌توان میزان اهمیت τ_{ij} را نسبت به η_{ij} تغییر داد. به علاوه j^k گره‌ای است برای مورچه k ام که دارای ماکزیمم مقدار $\tau_{ij}^\alpha \eta_{ij}^\beta$ در بین تمام اعضای مجموعه N_i^k است و q یک متغیر تصادفی است که به طور یکنواخت در $[0,1]$ توزیع شده است. همچنین q_0 یک پارامتر آستانه ثابت در $[0,1]$ است که نسبت اهمیت استخراج به اکتشاف را تعیین می‌کند. بنابراین موقعی که q کمتر یا مساوی q_0 باشد مورچه‌ها اکتشاف را به کار می‌گیرند و گره j^k را به عنوان گره بعدی در تورشان انتخاب می‌کنند در حالی که اگر q بزرگ‌تر از q_0 باشد مورچه‌ها از استخراج بر پایه احتمال، برای انتخاب گره بعدی استفاده می‌کنند.

باید توجه کرد که در مسئله فروشنده دوره‌گرد مجموعه گره‌های ملاقات نشده N_i^k که $N_{i(TSP)}^k$ نامیده می‌شود، در هر مرحله با تمام گره‌هایی که تاکنون توسط مورچه k ام مورد ملاقات واقع نشده‌اند، برابر است در حالی که برای مسئله CVRP این مجموعه $N_{i(CVRP)}^k$ کوچک‌تر می‌شود و به تبدیل می‌شود که عبارت است از تمام مشتری‌هایی که تاکنون توسط مورچه k ام ملاقات نشده‌اند و اضافه شدن تقاضای آن‌ها به ظرفیت مورچه k ام، که نقش وسیله نقلیه را بازی می‌کند، از مقدار ظرفیت مورچه بیشتر نمی‌شود. در نهایت برای مسئله VRPTW مجموعه $N_{i(VRPTW)}^k$ دوباره کوچک‌تر می‌شود و نامیده می‌شود. بنابراین مشتریان علاوه بر صادق بودن در محدودیت ظرفیت که در CVRP شرح داده شد باید دو محدودیت دیگر $a_j + g_j + s_j + t_{j0} \leq l_0$ و $a_j \leq l_j$ نیز صدق کنند که در آن:

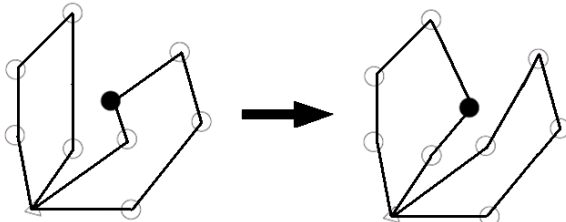
a_j : زمان رسیدن وسیله نقلیه به مشتری j (چون فقط در

هر تکرار یک وسیله نقلیه هر مشتری را ملاقات می‌کند بنابراین این زمان یکتا است).

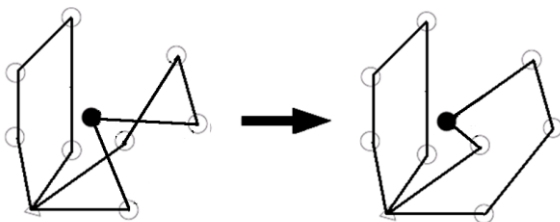
l_j : کران بالای پنجره زمانی مشتری j

g_j : زمان انتظار هر وسیله نقلیه در مشتری j در صورت

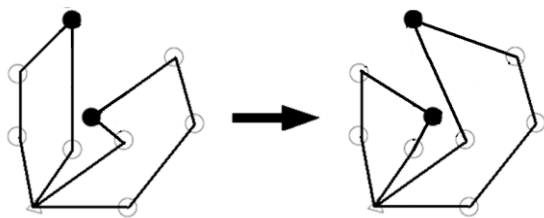
افزایش کارایی استفاده می‌کند. ده عدد همسایگی از هر کدام از این چهار روش تولید می‌شود و در این چهل همسایگی هر کدام از جواب‌های شدنی که بهبود بیشتری را در الگوریتم ایجاد کند، انتخاب می‌گردد.



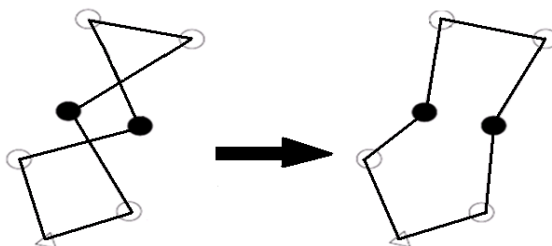
شکل ۲. حرکت درج (برداشتن یک مشتری از یک مسیر و ارسال آن به مسیر دیگر)



شکل ۳. حرکت درج (برداشتن یک مشتری از یک مسیر و ارسال آن به همان مسیر)



شکل ۴. حرکت جابجایی (جابجا کردن یک مشتری در یک مسیر با یک مشتری دیگر از مسیر دیگر)



شکل ۵. حرکت جابجایی (جابجا کردن یک مشتری در یک مسیر با یک مشتری دیگر از همان مسیر)

۴. اساس الگوریتم ژنتیک بر آن است که موجوداتی که دارای بیشترین تطبیق با محیط هستند بتوانند به حیات خود ادامه دهند. به عبارت دیگر موجودات قوی‌تر و سازگارتر توانایی بیشتری در گذشتن از خطرات داشته و می‌توانند نسل خودشان را حفظ کنند. در الگوریتم

مشتری‌های ملاقات نشده $N_{i(VRPTW)}$ هستند. به علت اینکه در مسئله $VRPTW$ طول بازه زمانی ملاقات هر مشتری نیز نقش بسیار مهمی را بازی می‌کند و هر مشتری که طول پنجره زمانی کوتاه‌تری دارد بهتر است که زودتر ملاقات شود، بنابراین متغیر w_j نیز در فرمول (۱) بکار گرفته می‌شود که با $(l_j - e_j)^{-1}$ برابر است. بنابراین در مسئله $VRPTW$ از تابع تصمیم‌گیری که در فرمول (۴) آمده به جای فرمول (۱) استفاده می‌شود.

$$P_{ij}^k = \begin{cases} 1 & \text{if } q \leq q_0 \text{ \& } j = j^k \\ 0 & \text{if } q \leq q_0 \text{ \& } j \neq j^k \\ \frac{\tau_{ij} \eta_{ij}^\beta w_j^\lambda}{\sum_{j \in N_i^k} \tau_{ij} \eta_{ij}^\beta w_j^\lambda} & q > q_0 \end{cases} \quad (4)$$

۲. پارامتر q_0 یکی از مهم‌ترین پارامترها در الگوریتم ACS است که نقش اساسی در یافتن جواب برای مسئله بازی می‌کند به طوری که مقدار نزدیک به یک آن سبب می‌شود که در بیشتر مواقع، الگوریتم بهترین مشتری دارای ماکزیمم مقدار $\tau_{ij}^\alpha \eta_{ij}^\beta w_j^\lambda$ را انتخاب کند. این عمل سبب می‌شود که ناحیه جستجو محدود گردد و الگوریتم در بهینه‌های محلی گیر بیفتند اما از طرف دیگر مقدار نزدیک به صفر آن سبب می‌گردد که قانون تصمیم آن همانند الگوریتم AS عمل کند و دایره‌ای وسیع از جواب‌های شدنی را در هر انتخاب در نظر بگیرد. این عمل سبب جستجوی تصادفی می‌شود و الگوریتم نمی‌تواند به جستجوی همسایگی‌های جواب‌های خوب بپردازد. در الگوریتم پیشنهادی مقدار q_0 نه یک پارامتر بلکه یک متغیر در نظر گرفته می‌شود. به نحوی که اگر الگوریتم پیشنهادی نتواند در سه تکرار متوالی بهترین جواب را بهبود بخشد آن‌گاه مقدار q_0 نصف می‌گردد. سپس این مقدار در طی الگوریتم ثابت می‌ماند تا اینکه مقدار بهترین جواب الگوریتم بهبود بخشد. در آن صورت دوباره مقدار q_0 به مقدار اولیه بازمی‌گردد.

۳. دو روش درج و جابجایی از بهترین الگوریتم‌های ابتکاری همسایه هستند که در بیشتر روش‌های کارا از آن‌ها استفاده شده است. الگوریتم پیشنهادی از این دو روش، که در شکل‌های ۲، ۳، ۴ و ۵ نشان داده شده است، برای

۴- تجزیه و تحلیل

الگوریتم حل به زبان *Matlab7* نوشته شده و روی کامپیوتری ۷ هسته‌ای با ۸ گیگا حافظه و 3500 Mhz قدرت پردازشگر اجرا گردیده است. در این کد، ورودی‌های ابتدایی مسئله شامل مختصات انبار و مشتری‌ها، مقدار تقاضای هر مشتری، پنجره‌های زمانی برای همه مشتری‌ها، مقدار زمان بارگیری مورد نیاز برای هر مشتری و مقدار زمان پیمایش هر کمان را وارد کرده و بهترین جواب و مقدار را به دست آورد. برای مقایسه کامل‌تر الگوریتم پیشنهادی، مسائل معروف سالامان که در ادبیات موضوع به آن اشاره شد و در آدرس اینترنتی زیر وجود دارند در نظر گرفته شده است.

<http://w.cba.neu.edu/~msolomon/problems.htm>

این مسائل براساس ویژگی‌ها و خصوصیات که دارا می‌باشند در سه گروه کلی C ، R و RC دسته‌بندی شده‌اند. هر گروه دارای ۸ تا ۱۲ مسئله می‌باشند که دارای اختلافات جزئی در مختصات نقاط، میزان تقاضا، زمان سرویس‌دهی و پنجره‌های زمانی هستند. به علاوه حداکثر تعداد مشتری‌ها در این مثال‌ها ۱۰۰ می‌باشد. در این مثال‌ها مجموعه C نشان‌دهنده مشتریانی هستند که پنجره‌های زمانی آن بر این اساس طراحی شده که جواب از قبل مشخص فرض شده است. در مجموعه R مشتریان به صورت تصادفی در داخل یک مربع توزیع شده‌اند. در نهایت در مجموعه RC ترکیبی از مشتریان در دو دسته R و C وجود دارند. به علاوه هر یک از این سه نوع به دو دسته فرعی ۱ و ۲ تقسیم می‌شوند. در مسائل نوع ۱ پنجره‌های زمانی کوتاه می‌باشند و ظرفیت وسایل نقلیه کوچک هستند در حالی که در نوع ۲ این شرایط کاملاً برعکس است. به عبارت دیگر در نوع ۲ پنجره‌های زمانی بلند می‌باشند و ظرفیت وسایل نقلیه بزرگ هستند. بنابراین جواب‌های مثال‌های نوع ۲ دارای مسیرهای کم و تعداد مشتری‌های زیاد اختصاص داده شده به هر مسیر می‌باشد. باید توجه کرد که در این بخش از هر شش دسته، دو مثال که همگی دارای ۱۰۰ مشتری هستند، انتخاب شده‌اند زیرا اولاً دارای ترکیبی مناسب از شش دسته از مسائل هستند و ثانیاً الگوریتم‌های زیادی بر روی این مثال‌ها مورد آزمایش قرار گرفته شده است و بدین جهت می‌توان مقایسه مناسبی بین الگوریتم پیشنهادی *HACS* و الگوریتم‌های دیگر انجام داد تا بدین وسیله کارایی الگوریتم با دقت بیشتری مورد بررسی قرار گیرد.

پیشنهادی علاوه بر فرمون‌ریزی سراسری برای بهترین جواب در تکرار جاری همانند روش معمولی *ACS*، بهترین جواب تاکنون پیدا شده نیز مورد توجه قرار می‌گیرد و بر روی این جواب نیز فرمون‌ریزی سراسری انجام می‌گیرد. این عمل سبب افزایش فرمون بر روی یال‌های متعلق به این دو بهترین مسیر جاری و بهترین مسیر تاکنون پیدا شده می‌شود و سبب می‌گردد که همسایگی‌های این یال‌ها با احتمال بیشتری در تکرارهای بعدی انتخاب شوند.

همانند سایر روش‌های فراابتکاری الگوریتم پیشنهادی دارای پارامترهای زیادی است که باید به وسیله کاربر قبل از شروع الگوریتم تنظیم شود. باید توجه داشت که چگونگی مقدار دهی به این پارامترها بسیار مهم و حیاتی در کارایی الگوریتم می‌باشد. با توجه به رفتار تصادفی الگوریتم و تنظیم بهتر این پارامترها، ممکن است نتایج بهتری به دست آید. بدین منظور مقادیر مورد آزمایش و بهترین مقادیر پیدا شده برای پنج پارامتر $\lambda, \rho, \beta, \alpha$ و q_0 در جدول ۱ نشان داده شده است. در این جدول با در نظر گرفتن ۴ پارامتر λ, ρ, β و q_0 با کمترین مقدار، بهترین مقدار α به دست می‌آید که در اینجا ۲ به دست آمده است. سپس با در نظر گرفته شدن α به مقدار ۲ و λ, ρ و q_0 با کمترین مقدار مورد آزمایش بهترین مقدار β به دست می‌آید و به همین ترتیب بهترین مقادیر λ, ρ و q_0 نیز به یافت می‌گردد. باید توجه داشت که در انتهای الگوریتم همانند سایر الگوریتم‌های فراابتکاری شرط پایانی حلقه در نظر گرفته می‌شود و در صورتی که این شرط برقرار باشد الگوریتم به پایان می‌رسد. این شرط تکرار متوالی بهترین جواب تاکنون به دست آمده به تعداد ۱۵ بار است.

جدول ۱. تحلیل حساسیت پارامترها

پارامتر	مقادیر در نظر گرفته شده	بهترین مقدار
α	[۰٫۵، ۱، ۱٫۵، ۲، ۲٫۵]	۲
β	[۱، ۲، ۳، ۴، ۵]	۵
λ	[۱، ۲، ۳، ۴، ۵]	۳
ρ	[۰٫۱، ۰٫۳، ۰٫۵، ۰٫۷، ۰٫۹]	۰٫۴
q_0	[۰٫۸۰، ۰٫۸۵، ۰٫۹۰، ۰٫۹۵، ۰٫۹۹]	۰٫۹۰

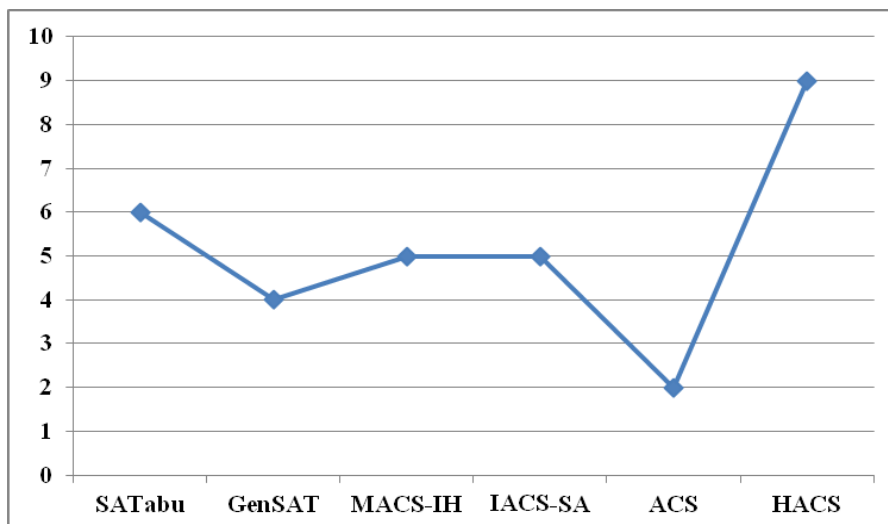
ترتیب مورد بررسی قرار می‌گیرند. با بررسی این نتایج می‌توان نتیجه گرفت که الگوریتم دارای جواب‌های بهتر نسبت به الگوریتم SATabu است زیرا این الگوریتم در شش مثال توانسته است که به بهترین جواب‌ها دست پیدا کند در حالی که الگوریتم HACS در هفت مثال توانسته به بهترین جواب‌ها برسد. همچنین باید توجه داشت که الگوریتم پیشنهادی در دو مثال R1-01 و RC1-01 توانسته که کیفیت جواب‌ها را بهبود بخشد و به جواب‌های بسیار خوبی دست یابد. از طرف دیگر می‌توان مشاهده کرد که الگوریتم GenSAT ضعیف‌ترین نتایج را در این جدول به دست آورده و فقط در چهار مثال توانسته که به بهترین جواب‌ها نائل گردد. دو الگوریتم دیگر در جدول ۱ توانسته‌اند که از این لحاظ جواب‌های یکسانی را به یافته و در پنج مثال بهترین جواب‌ها را به دست آورند. بنابراین در کل می‌توان به این نتیجه دست یافت که الگوریتم پیشنهادی در بین الگوریتم‌ها توانسته که به بهترین جواب‌ها دست یابد و همان‌طور که در شکل ۵ نشان داده شده است، قوی‌ترین الگوریتم محسوب می‌شود.

در جدول ۲ الگوریتم پیشنهادی HACS با الگوریتم‌های فراابتکاری مشهوری که تاکنون برای حل این مسئله مورد استفاده قرار گرفته شده‌اند، مقایسه شده است. در این جدول و در ستون‌های دوم تا پنجم نتایج چهار روش مهم فراابتکاری شامل الگوریتم جستجوی ممنوع (SATabu) [Tan, Lee,] [Thangiah,] [and Ou, 2001]، روش ژنتیک (GenSAT) [Osman, and Sun, 1994]، روش مورچگان ترکیب شده با الگوریتم ابتکاری ورودی (MACS-IH) [Loiseau, and Ramonet, 2011] و الگوریتم مورچگان ترکیبی (IACS-SA) [Chen, and Ting, 2005] را نشان می‌دهد. بعلاوه در ستون ششم نتایج الگوریتم پیشنهادی HACS و در ستون هفتم بهترین جواب‌هایی که تاکنون به وسیله سایر الگوریتم‌ها به دست آمده است (BKS)، ارایه شده است.

برای تجزیه و تحلیل جواب‌های به دست آمده سه روش مقایسه تعداد بهترین جواب‌های تاکنون به دست آمده، میانگین جواب‌های به دست آمده برای ۱۲ مثال و مقایسه با بهترین جواب‌های تاکنون به دست آمده در نظر گرفته می‌شود و به

جدول ۲. مقایسه الگوریتم با بهترین الگوریتم‌های فراابتکاری

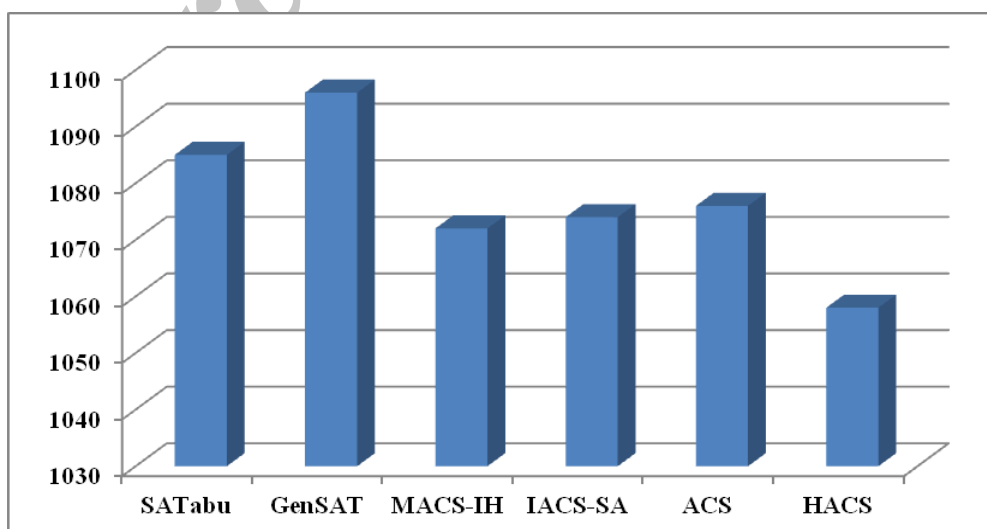
مثال	SATabu	GenSAT	MACS-IH	IACS-SA	ACS	HACS	BKS
R1-01	۱۶۵۵	۱۶۷۷	۱۶۵۱	۱۶۷۱	۱۶۵۵	۱۶۲۵	۱۶۴۶
R1-11	۱۰۹۱	۱۱۵۱	۱۰۹۷	۱۱۱۲	۱۰۹۷	۱۰۹۱	۱۰۹۱
C1-05	۸۲۹	۸۲۹	۸۲۹	۸۲۹	۸۳۸	۸۲۹	۸۲۹
C1-07	۸۲۹	۸۲۹	۸۲۹	۸۲۹	۸۲۹	۸۲۹	۸۲۹
RC1-01	۱۶۷۸	۱۶۶۹	۱۶۹۷	۱۶۵۳	۱۶۹۷	۱۶۳۱	۱۶۵۳
RC1-03	۱۱۹۶	۱۲۰۷	۱۲۶۲	۱۲۸۵	۱۲۰۷	۱۱۹۶	۱۱۹۶
R2-02	۱۰۷۸	۱۱۷۶	۱۱۹۲	۱۱۰۲	۱۱۰۲	۱۱۳۰	۱۰۷۸
R2-05	۱۰۴۹	۱۱۲۸	۹۹۴	۱۰۳۳	۱۰۴۹	۹۹۴	۹۹۴
C2-01	۵۹۲	۵۹۲	۵۹۲	۵۹۲	۶۰۵	۵۹۲	۵۹۲
C2-06	۶۵۴	۵۸۸	۵۸۸	۵۸۸	۵۸۸	۵۸۸	۵۸۸
RC2-05	۱۴۲۶	۱۳۸۹	۱۲۹۸	۱۳۴۴	۱۲۹۸	۱۳۴۶	۱۲۹۷
RC2-08	۹۴۳	۹۱۹	۸۲۹	۸۴۸	۹۴۳	۸۳۹	۸۲۸



شکل ۶. تعداد بهترین جواب‌های به دست‌آمده توسط هر الگوریتم

دیگر روش SATabu که از روش‌های کارا در به دست آوردن بهترین جواب تاکنون پیدا شده بود، نمی‌تواند کارایی خود را از در این محک نیز اثبات کند و با میانگین ۱۰۸۵ بعد از الگوریتم GenSAT، ضعیف‌ترین الگوریتم است. بعد از این دو الگوریتم، به ترتیب روش‌های ACS، IACS-SA و MACS-IH قرار دارند که به ترتیب با میانگین ۱۰۷۶، ۱۰۷۴ و ۱۰۷۲ به ترتیب در رده چهارم، سوم و دوم مقایسه الگوریتم‌ها قرار دارند. در نهایت الگوریتم پیشنهادی HACS است که توانسته از این نظر نیز کارایی خود را اثبات کند و همان طور که در شکل ۷ نشان داده شده است با میانگین ۱۰۵۸ بهترین جواب را به دست آورد.

برای مقایسه این الگوریتم‌ها در مرحله دوم از میانگین جواب‌های به دست‌آمده در ۱۲ مثال مربوطه استفاده شده است. باید توجه کرد که این معیار از اهمیت بسیار زیادی برخوردار است زیرا به این نکته اشاره می‌کند که رفتار الگوریتم به طور متوسط در بین مثال‌های استاندارد چگونه است. به علاوه با این روش می‌توان دید که هر الگوریتم دارای چقدر پایداری است و آیا می‌تواند به طور کلی برای این دسته از مثال‌ها جواب‌های مناسبی به دست آورد یا خیر. با توجه به جواب‌های به دست آمده در شکل ۷ می‌توان دید که الگوریتم GenSAT دارای ضعیف‌تری جواب است و دارای میانگین ۱۰۹۶ است و همانند معیار نخستین در انتهای رده‌بندی الگوریتم‌ها قرار دارد. از طرف



شکل ۷. میانگین جواب‌های به دست‌آمده توسط هر الگوریتم

جدول ۳. مقایسه الگوریتم با بهترین جواب‌های پیدا شده

RC2-08	RC2-05	C2-06	C2-01	R2-05	R2-02	RC1-03	RC1-01	C1-07	C1-05	R1-11	R1-01	مثال
۷۴.۸۲	۷۳.۴۲	۶۸.۲۷	۶۳.۰۱	۷۳.۵۲	۶۹.۵۳	۵۷.۹۲	۵۵.۲۶	۴۴.۱۲	۷۳.۱۱	۴۸.۱۲	۵۳.۲۲	زمان روش ACS
۱۱۵.۰۳	۱۲۰.۵۶	۱۲۱.۸۴	۱۰۴.۷۵	۹۳.۹۲	۸۹.۰۴	۷۹.۶۴	۱۰۲.۹۳	۱۰۱.۴۵	۱۰۲.۵۶	۷۸.۸۳	۹۵.۶۸	زمان روش HACS
۸۳۹	۱۳۴۶	۵۸۸	۵۹۲	۹۹۴	۱۱۳۰	۱۱۹۶	۱۶۳۸	۸۲۹	۸۲۹	۱۰۹۱	۱۶۲۵	نتایج HACS
۸۲۸	۱۲۹۷	۵۸۸	۵۹۲	۹۹۴	۱۰۷۸	۱۱۹۶	۱۶۵۳	۸۲۹	۸۲۹	۱۰۹۱	۱۶۴۶	BKS
-۱.۳۳	-۳.۷۸	۰	۰	۰	-۴.۸۲	۰	۰.۹۰	۰	۰	۰	۱.۲۸	Gap

الگوریتم ژنتیک و تبدیل کردن یک پارامتر الگوریتم به یک متغیر بود. برای تست کارایی الگوریتم چندین مسئله استاندارد از ادبیات موضوع در نظر گرفته شد و نتایج الگوریتم با چندین روش فرابتنکاری مشهور بر روی این مثال‌ها مورد مقایسه قرار گرفت. نتایج نشان داد که الگوریتم پیشنهادی کارایی بسیار خوبی از خود نشان می‌دهد و به جواب‌های بسیار خوبی دست می‌یابد. استفاده از الگوریتم جستجوی ممنوع برای ترکیب با این روش می‌تواند نتایج بهتری را برای الگوریتم در بر داشته باشد. به علاوه می‌توان این الگوریتم را برای مسائلی مانند مسیریابی وسیله نقلیه همراه با دریافت و تحویل کالا و سایر نسخه‌های *VRPTW* گسترش داد که به مقاله‌های بعدی موقوف می‌گردد.

۶- سپاسگزاری

این مقاله نتیجه اجرای طرح پژوهشی تحت عنوان "یک روش مورچگان ترکیب شده با الگوریتم‌های ابتکاری درج و جابجایی برای مسئله برنامه‌ریزی توزیع همراه با پنجره‌های زمانی" است که توسط دانشگاه آزاد اسلامی واحد پرند مورد حمایت مادی و معنوی قرار گرفته است.

۷- پی‌نوشت‌ها

1. Capacitated Vehicle Routing Problem (CVRP)
2. Vehicle Routing Problem with Time Window (VRPTW)
3. NP-Complete Problems
4. Ant Colony System (ACS)
5. Ant Colony Optimization (ACO)
6. Insert Move
7. Swap Move
8. Vehicle Routing Problem (VRP)
9. Dantzig
10. Multiple Traveling Salesmen Problem (MTSP)
11. Kolen
12. Kohl
13. Chen
14. Lingo 8.0
15. Lei
16. Yu

در جدول ۳ زمان اجرای الگوریتم ACS و HACS به ثابته نشان داده شده است. همچنین برای نشان دادن کارایی الگوریتم HACS، درصد انحراف نسبت به بهترین جواب تاکنون به دست آمده (*Gap*) برای هر مثال در این جدول ارائه گردیده است. این درصد از فرمول (۵) به دست می‌آید. در این فرمول $f(s)$ نشان دهنده مقدار به دست آمده به وسیله الگوریتم مورد بحث است در حالی که $f(s^*)$ نشان دهنده بهترین مقدار است که تاکنون برای آن مسئله به دست آمده است. بنابراین می‌توان نتیجه گرفت که اگر مقدار درصد انحراف برای یک الگوریتم دلخواه در یک مسئله کمینه‌سازی مثبت به دست آید، آنگاه مقدار به دست آمده از بهترین مقدار تاکنون به دست آمده دارای کیفیت بیشتری است. اما اگر این مقدار منفی به دست آید یعنی آنکه الگوریتم نتوانسته است که مقدار کیفیت جواب را افزایش دهد.

$$Gap = \frac{f(s^*) - f(s)}{f(s^*)} \times 100 \quad (5)$$

نتایج بر این نکته اشاره دارد که روش پیشنهادی نتوانسته است که جواب‌های بسیار خوبی برای مسائل استاندارد *VRPTW* به دست آورد. به خصوص همان‌طور که گفته شد الگوریتم نتوانسته که در هفت مثال از دوازده مثال مربوط به بهترین جواب‌هایی که تاکنون به دست آمده است، برسد. به علاوه در دو مثال نتوانسته که کیفیت جواب‌ها را افزایش دهد و در سه مثال دیگر نیز الگوریتم با خطای قابل قبولی به جواب‌هایی با کیفیت دست یافته است. همچنین اگر الگوریتم پیشنهادی از نظر به دست آوردن میانگین درصد انحراف در نظر گرفته شود، می‌توان نتیجه گرفت که الگوریتم HACS دارای میانگین ۰.۶۵- درصد انحراف است که کمتر از یک درصد می‌باشد.

۵- نتیجه‌گیری

نوعی اصلاح روش ACS به نام HACS در این مقاله برای مسئله *VRPTW* ارائه شد. این روش دارای اصلاحاتی شامل قانون تصمیم، استفاده از الگوریتم‌های کارایی ابتکاری، استفاده از ایده

- salesman problem. *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, pp. 53–66.
- Eksioğlu, B., Vural, A. V. and Reisman, A. (2009) "The vehicle routing problem: A taxonomic review", *Computers. & Industrial Engineering*, Vol. 57, pp. 1472–1483.
 - Kolen, A. W. J., Rinnooy Kan, A. H. G. and Trienkens, H. W. J. M. (1987) "Vehicle routing and scheduling with time windows", *Operations Research*, Vol. 35, No. 2, pp. 266-273.
 - Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science*, Vol. 43, pp. 408–416.
 - Lei, H., Laporte, G. and Guo, B. (2011) "The capacitated vehicle routing problem with stochastic demands and time windows", *Computers & Operations Research*, Vol. 38, No. 12, pp. 1775-1783.
 - Pesant, M. and Gendreau, M. (1999) "A constraint programming framework for local search methods", *Journal of Heuristics*, Vol. 5, No. 3, pp. 255-279.
 - Solomon, M. M. (1987) "Algorithms for the vehicle routing and scheduling problems with time window constraints", *Operations Research*, Vol. 35, No. 2, pp. 254-265.
 - Tan, K. C. Lee, L. H. and Ou, K. (2001) "Artificial intelligence heuristics in solving vehicle routing problems with time window constraints", *Eng. Appl. Artif. Intell*, Vol. 14, pp. 825–837.
 - Tavakkoli-Moghaddam, R., Gazanfari, M., Alinaghian, M., Salamatabkhsh, A. and Norouzi, N. (2011) "A new mathematical model for a competitive vehicle routing problem with time windows solved by simulated annealing", *Journal of Manufacturing Systems*, Vol. 30, No. 2, pp. 83-92.
 - Thangiah, S. R., Osman, I. H. and Sun, T. (1994) "Hybrid Genetic Algorithm, Simulated Annealing and Tabu Search Methods for Vehicle Routing Problems with Time Windows", *Technical Report UKC/OR94/4*, Institute of Mathematics and Statistics, University of Kent, Canterbury, UK.
 - Vidal, T., Crainic, T. G., Gendreau, M. and Prins, C. (2012) "A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows", *Computers & Operations Research*, In Press.
 - YousefiKhoshbakht, F., Sedighpour, M. (2012) "A Combination of Sweep Algorithm and Elite Ant Colony Optimization for Solving the Multiple Traveling Salesman Problem", accepted paper in *Proceedings of the Romanian Academy*, Vol. 13, No. 1, pp. 295-301.
 - Yu, B. and Yang, Z. Z., (2011) "An ant colony optimization model: The period vehicle routing problem with time windows", *Transportation Research Part E: Logistics and Transportation Review*, Vol. 47, No. 2, pp. 166-181.
 - Yu, B., Yang, Z. Z. and Yao, B. Z. (2011) "A hybrid algorithm for vehicle routing problem with time windows", *Expert Systems with Applications*, Vol. 38, No. 1, pp. 435-441.
 - 17. Balseiro
 - 18. Vidal
 - 19. Ant System (AS)
 - 20. Dorigo
 - 21. Gambardella
- ### ۸- مراجع
- سپهری، م.م، ستاک، م.، (۱۳۸۳) "ارایه روشهای ابتکاری و حل مسیریابی وسایل نقلیه با حمل در بازگشت در حالت غیر همگن" اولین کنفرانس لجستیک و زنجیره تامین اولین کنفرانس لجستیک و زنجیره تامین، ص ۱-۱۳.
 - ظفری، ع، تشکری هاشمی، س. م. و یوسفی خوشبخت، م. (۱۳۸۹) "الگوریتم ترکیبی موثر ژنتیک برای حل مسئله مسیریابی وسیله نقلیه"، *مهندسی صنایع و مدیریت تولید*، جلد ۲۱، شماره ۲، ص ۶۳-۷۶.
 - یوسفی خوشبخت، م. و رحمتی، ف. (۱۳۹۰) "یک الگوریتم بهبود یافته جمعیت مورچگان برای حل مسئله مسیریابی وسیله نقلیه همراه با دریافت و تحویل هم‌زمان کالا"، *پژوهشنامه حمل و نقل*، جلد ۸، شماره ۲، ص ۱۹۸-۱۸۳.
 - یوسفی خوشبخت، م.، دیده‌ور، ف.، رحمتی، ف. و صدیق‌پور، م.، (۱۳۹۱) "الگوریتم موثر رقابتی فراگیر برای حل مسئله مسیریابی وسیله نقلیه باز"، *پژوهشنامه حمل و نقل* ۹ (۱)، ۸۳-۹۵.
 - یوسفی خوشبخت، م.، دیده‌ور، ف.، رحمتی، ف. و سعادت‌ی اسکندری، ز.، (۱۳۹۱) "یک روش جمعیت مورچگان ترکیبی برای مسئله مسیریابی وسایل نقلیه با ناوگان ناهمگن ثابت"، *پژوهشنامه حمل و نقل* ۹ (۲)، ۱۳۹۱، ص ۱۹۱-۲۰۷.
 - Balseiro, S. R., Loiseau, I. and Ramonet, J. (2011) "An Ant Colony algorithm hybridized with insertion heuristics for the Time Dependent Vehicle Routing Problem with Time Windows", *Computers & Operations Research*, Vol. 38, No. 6, pp. 954-966.
 - Chen, C. H. and Ting, C. J. (2005) "A hybrid ant colony system for vehicle routing problem with time windows", *J. East. Asia Soc. Transp. Stud*, Vol. 6, pp. 2822–2836.
 - Chen, J. C., Chiang, C. S., Chen, B. B. and Chen, C. W. (2007) "Solving Vehicle Routing Problem with Time Window Constraints", 1-4244-0885-7/07/\$20.00 ©2007 IEEE.
 - Cheng, C. B. and Wang, K. P. (2009) "Solving a vehicle routing problem with time windows by a decomposition technique and a genetic algorithm", *Expert Systems with Applications*, Vol. 36, No. 4, pp. 7758-7763.
 - Dorigo M. (1992) "optimization, Learning and natural algorithms", *Ph.D Thesis, Dip.Electronica Informazion, Politecnico di Milano Italy*.
 - Dorigo, M., & Gambardella, L. M. (1997a). Ant colony system: A cooperative learning approach to the traveling