

حداقل دیرکرد در زمان‌بندی مسائل جریان کارگاهی با موعد تحویل میانی

لعیا الفت*

* استاد، گروه مدیریت صنعتی، دانشکده مدیریت و حسابداری، دانشگاه علامه طباطبائی، تهران، ایران

پذیرش: ۱۳۹۶/۵/۳۰

دریافت: ۱۳۹۵/۹/۱۸

چکیده

این مقاله به زمان‌بندی کارها در سیستم جریان کارگاهی با معیار عملکرد مجموع دیرکردهای مرحله‌ای می‌پردازد. این معیار بیانگر شرایطی است که کارها علاوه بر موعد نهایی، دارای مواعدهای تحویل میانی برای فعالیت‌ها هستند. در برخی از امور، مانند پروژه‌های تحقیقاتی، کارهای خدماتی، طراحی و مهندسی، خروجی گام‌های مختلف تعیین شده و زمان تحویل آن‌ها مشخص می‌شود. با توجه به طی نمودن یک مسیر توسط این پروژه‌ها، استفاده از منابع مشترک و همچنین تعهد به انجام به‌موقع مراحل کاری و عدم تأخیر آن‌ها، برنامه‌ریزی صحیح برای تخصیص منابع و زمان‌بندی مناسب جهت حداقل کردن مجموع دیرکردها ضروری می‌نماید. تاکنون، این هدف کمتر مدنظر قرار گرفته و استفاده از روش‌های فراابتکاری برای حل آن مشاهده نشده است. با توجه به NP-hard بودن چنین مسئله‌ای، در این مقاله نسبت به حل آن با روش‌های فراابتکاری، الگوریتم ژنتیک، شبیه‌سازی تبرید و ازدحام ذرات اقدام شد. ۹۶ مسئله در ابعاد مختلف و سه مقدار عامل فشردگی برای زمان‌های تحویل ایجاد و حل شدند. الگوریتم‌های شبیه‌سازی تبرید و الگوریتم ژنتیک در رابطه با دستیابی به هدف مسئله، یعنی حداقل مجموع دیرکرد، نتایج بهتری را نشان دادند. روش ازدحام ذرات زمان حل کمتری داشت. در کل با در نظر گرفتن هر دو شاخص، نتایج نشان از برتری الگوریتم ژنتیک در این مسئله دارد.

کلمات کلیدی: زمان‌بندی جریان کارگاهی، موعد تحویل میانی، دیرکرد کل، الگوریتم‌های فراابتکاری.

۱- مقدمه

زمان‌بندی از موارد مطرح در برنامه‌ریزی تولید و خدمات است که به تخصیص منابع محدود به کارها و تعیین توالی آن‌ها در راستای دستیابی به اهداف معیارهای عملکرد سیستم می‌پردازد.

سیستم جریان کارگاهی^۱ یکی از انواع سیستم‌های تولیدی است که در آن کلیه کارها مسیر مشابهی را روی ماشین‌ها طی می‌کنند. در مسائل زمان‌بندی، اهداف مختلفی ممکن است دنبال شود. بررسی متون ناظر بر این است که در ابتدا، اغلب تحقیقات حول محور حداقل کردن «زمان خاتمه کارها^۲» صورت گرفته و علیرغم اهمیت تأخیر یا دیرکرد در عمل، کمتر به آن پرداخته می‌شده است؛ اما در سال‌های اخیر پژوهش‌های مرتبط با این دسته از معیارهای عملکردی نیز مورد توجه قرار گرفته‌اند. گفتنی است در تحقیقاتی که در زمینه حداقل کردن دیرکرد انجام شده، تنها برای مرحله آخر هر کار، موعد تحویل در نظر گرفته شده است؛ درحالی‌که در بسیاری از مسائل طراحی محصول، پروژه‌های مطالعاتی و نظایر آن، زمان تحویل خروجی هر مرحله مشخص می‌شود و دیرکرد مرحله‌ای نیز مجاز نیست. این مقاله به این مهم می‌پردازد و حداقل کردن مجموع دیرکرد^۳ در یک سیستم جریان کارگاهی را با فرض اینکه هر کار متشکل از چندین عملیات یا مراحل با موعد تحویل است، مدنظر قرار می‌دهد؛ به عبارتی مواعدهای میانی در محاسبه دیرکرد هر کار لحاظ می‌شود. در صورتی‌که زمان تحویل هر مرحله مساوی با زمان تحویل عملیات آخر یا به عبارتی موعد تحویل نهایی قرار داده شود، مسئله تبدیل به مسئله معمول در این حوزه می‌شود.

با توجه به ماهیت مسائل زمان‌بندی و تعیین توالی، از روش‌های دقیق برای یافتن جواب بهینه (مگر مسائل کوچک) نمی‌توان استفاده کرد [۱]؛ از این رو، روش‌های تقریبی، ابتکاری و یا فراابتکاری بکار گرفته می‌شوند. در این مقاله، از روش‌های شبیه‌سازی تبرید^۴، الگوریتم ژنتیک^۵ و ازدحام ذرات^۶ برای حل مسئله استفاده شده است. در ادامه به پیشینه تحقیق، تعریف مسئله، شرح الگوریتم‌ها، یافته‌های پژوهش، بحث و نتیجه‌گیری پرداخته می‌شود.

۲- پیشینه پژوهش

پژوهش‌های انجام‌شده در سیستم جریان کارگاهی، ابتدا با معیار «زمان خاتمه کارها» صورت گرفته و سپس معیارهای مرتبط با تأخیر موردتوجه قرار گرفته است. سابقه تحقیقات در زمینه معیارهای مرتبط با دیرکرد به دهه ۱۹۷۰ برمی‌گردد [۲]. کلیه این پژوهش‌ها بر اساس موعد تحویل نهایی انجام‌گرفته‌اند. مسائل زمان‌بندی با در نظر گرفتن موعد تحویل مشترک توسط کنت مطرح شد [۳]. پس‌از آن، هال و همکاران ویژگی‌های بهینگی جواب مسئله مطرح‌شده را بیان و NP-Complete بودن آن را اثبات کردند. در ابتدا، مطالعات از جهت رویکرد، معطوف به روش‌های دقیق و ازجمله الگوریتم‌های انشعاب و تحدید بوده است [۴].

با عنایت به مشکلی که در حل دقیق این‌گونه مسائل وجود دارد، به روش‌های تقریبی، ابتکاری و فرا ابتکاری توجه شده است. با توجه به تنوع مسائل از جهت نوع سیستم و ویژگی‌های مسئله و اهداف موردنظر، مطالعات مختلفی گزارش شده است. در پژوهش برای حل مسائل زمان‌بندی ماشین‌های موازی از ترکیبی از الگوریتم‌های جستجوی ممنوعه، شبیه‌سازی تبرید و جستجوی همسایگی استفاده شده است [۵]. پژوهشگران دیگری سیستم ماشین‌های موازی را با هدف مجموع دیرکرد را با استفاده از الگوریتم‌های جستجوی ممنوعه و شبیه‌سازی تبرید حل کرده و با مقایسه نتایج بیان کرده‌اند که شبیه‌سازی تبرید عملکرد بهتری داشته است [۶]. دمیرل و همکاران برای حل مسئله ماشین موازی با هدف حداقل کردن دیرکرد، از الگوریتم ژنتیک استفاده نمودند [۷]. ولادا با مطالعه‌ای که در خصوص حداقل کردن مجموع دیرکرد در مسئله جریان کارگاهی انجام داد، عملکرد روش فراابتکاری شبیه‌سازی تبرید را بهتر از بقیه الگوریتم‌ها ارزیابی نمود [۸].

کریمی و داودپور مسئله جریان کارگاهی با دو هدف زمان خاتمه کارها و دیرکرد را با الگوریتم ژنتیک حل نموده و آن را به‌عنوان رویکردی قوی برای این نوع سیستم تولیدی معرفی می‌کنند [۹]. با در نظر گرفتن اینکه این روش‌ها در مسائلی گوناگون با مفروضات خاص به‌کار گرفته می‌شوند، تعیین بهترین روش دشوار می‌نماید، لکن می‌تواند نشان‌دهنده توانایی این روش‌ها در حل مسائل زمان‌بندی و تعیین توالی باشد.

۳- تعریف مسئله

در دنیای واقعی، بسیاری از پروژه‌های مطالعاتی، تحقیق و توسعه و مؤسسات خدماتی ملزم به ارائه مرحله‌ای کار بر اساس برنامه قبلی هستند که به ازای تأخیر جریمه تعلق می‌گیرد [۱۰]. طبیعت این پروژه‌ها به گونه‌ایست که گام‌های انجام پروژه‌ها مشابه بوده و نیازمند تخصیص منابع مشترک به هر مرحله آن‌هاست. این ویژگی‌ها، آن‌ها را با سیستم جریان کارگاهی قابل انطباق می‌سازد؛ لذا می‌توان مسئله‌ای را با n کار و m ماشین با هدف حداقل‌سازی مجموع تأخیرهای مرحله‌ای مدنظر قرار داد. با توجه به اینکه مسئله معمول زمان‌بندی جریان کارگاهی در دسته مسائل NP-hard قرار می‌گیرد [۱۱]، برای این مسئله نیز به‌جز در ابعاد کوچک نمی‌توان جواب دقیق به دست آورد؛ لذا از روش‌های فرا ابتکاری برای حل مسئله استفاده شده است.

جهت ارزیابی روش‌های فرا ابتکاری به کار گرفته شده، آزمایش‌های متعددی در نظر گرفته شد. در این راستا، مسائل به‌صورت زیر تعریف شدند:

اطلاعاتی که برای حل مسائل لازم است تا به‌طور تصادفی ایجاد شوند، شامل زمان عملیات هر کار در هر مرحله و موعدهای تحویل مرحله‌ای هر کار است. زمان‌های عملیات از توزیع یکنواخت در فاصله صفر و ده به‌طور تصادفی به دست می‌آید $p_{ij} \sim U[0,10]$. برای تعیین موعد تحویل نیز از توزیع یکنواخت استفاده شده است. ابتدا موعد نهایی تحویل به‌دست آمده، سپس با استفاده از آن موعدهای مرحله‌ای تعیین می‌شوند. موعد نهایی تحویل کارها از توزیع یکنواخت نمونه‌گیری می‌شود.

$$d_{im} \sim U \left[C \left(1 - TF - \frac{R}{2} \right), C \left(1 - TF + \frac{R}{2} \right) \right] \quad (1)$$

TF و R دو عامل هستند؛ TF جهت کنترل نزدیکی و دوری (فشرده‌گی) موعدهای تحویل بکار می‌رود. این عامل مقیاسی تقریبی از نسبت کارهای با تأخیر است و بیانگر متوسط تعداد کارهای معوقه در صورت انتخاب توالی تصادفی به‌طور تقریبی است. همچنین R مربوط به دامنه توزیع است و میزان تغییرات را کنترل می‌نماید. C

یک حد پائین برای زمان خاتمه کلیه کارهاست. از چنین توزیعی و یا مشابه آن در پژوهش‌های قبلی در مورد مسئله با یک ماشین یا چندین ماشین استفاده شده است. برای به دست آوردن تقریبی از مقدار C ، حداقل مجموع زمان هر کار روی ماشین‌های اول تا ماقبل آخر، با مجموع زمان کلیه کارها روی ماشین آخر، جمع می‌شود.

$$C = \min_i \left[\sum_{j=1}^{m-1} p_{ij} \right] + \sum_{i=1}^n p_{im} \quad (2)$$

تعیین موعدهای تحویل مرحله‌ای یک کار با توجه به موعد نهایی تحویل آن است و از مرحله آخر شروع و به مرحله اول ختم می‌شود. ابتدا برای هر کار (i) ضریبی مانند α_i محاسبه و از آن به شکل زیر استفاده می‌شود:

$$\alpha_i = \frac{d_i}{p_i} \quad (3)$$

d_i موعد نهایی است که از توزیع به دست آمده و در واقع d_{im} است. p_i مجموع زمان‌های عملیاتی کار i است.

$$d_{ij} = d_{i(j+1)} - \alpha_i P_{i(j+1)} \quad (4)$$

d_{ij} موعد تحویل کار i در مرحله j است که با کمک موعد تحویل کار i در مرحله $j+1$ و زمان کار i در مرحله $j+1$ و α_i محاسبه می‌شود.

عواملی که در ایجاد اطلاعات مسائل نمونه نقش دارند، عبارت‌اند از: N و M ، R ، TF که با تغییر هر یک از این‌ها مسائل متفاوتی به وجود آمده و می‌توان الگوریتم‌ها را در شرایط مختلف سنجید.

آزمایش‌هایی که جهت آزمون روش‌های فراابتکاری در نظر گرفته شدند با ترکیب عوامل بالا و مقادیر زیر است.

$$TF = 0/1 \text{ یا } 0/2 \text{ یا } 0/4$$

$$M = 5 \text{ و } 10 \text{ و } 15 \text{ و } 20$$

$$N = 5 \text{ و } 10 \text{ و } 15 \text{ و } 20 \text{ و } 25 \text{ و } 30 \text{ و } 35 \text{ و } 40$$

ترکیب‌های مختلف این عوامل جمعاً ۹۶ سناریو می‌سازد که روش‌های فراابتکاری برای هر مورد آن‌ها به کار گرفته شد. به این ترتیب در مجموع ۲۸۸ مسئله حل شده است.

۴- روش حل

همان‌طور که ذکر شد، جهت حل مسئله از الگوریتم‌های فراابتکاری استفاده شده است. بدین منظور، الگوریتم‌های ژنتیک (GA) و شبیه‌سازی تبرید (SA) به عنوان روش‌های رایج گسسته در حل مسائل زمان‌بندی انتخاب شد [۱۲؛ ۱۳؛ ۱۴؛ ۱۵] و الگوریتم ازدحام ذرات (PSO) به عنوان یکی از روش‌های رایج پیوسته، با توجه به کاربرد مؤثر این الگوریتم در مسائل جایگشتی از قبیل زمان‌بندی، زمان‌بندی پروژه‌ها و حرکت خودروها [۱۶؛ ۱۷]، مورد استفاده قرار گرفت.

در ادامه، قبل از تشریح ساختار الگوریتم‌های مورد استفاده، به ساختار پاسخ و تولید جواب اولیه اشاره می‌شود. با توجه به پیوسته بودن ساختار پاسخ در الگوریتم PSO و ماهیت گسسته بودن مسئله، روش کلید تصادفی جهت تعیین ساختار پاسخ به کار گرفته شد [۱۸]. در این روش، ابتدا یک ماتریس از اعداد پیوسته به اندازه کارهای مسئله، تولید شده و رتبه هر یک از اعداد، از کوچک به بزرگ، ترتیب کارها را در مسئله مشخص می‌کند. همچنین به منظور استاندارد کردن ساختار پاسخ و قابلیت مقایسه عملکرد روش‌های مورد استفاده، در الگوریتم‌های ژنتیک و شبیه‌سازی تبرید نیز از همین روش استفاده شد.

۴-۱- الگوریتم ژنتیک

الگوریتم‌های ژنتیک با اقتباس از فرآیندهای مشاهده شده در تکامل طبیعی، اولین بار توسط جان هولند^۷ در دهه ۷۰ پیشنهاد شد. این الگوریتم‌ها به طبقه بالاتری تحت عنوان الگوریتم‌های تکاملی^۸ تعلق دارند [۱۹]. در این الگوریتم، مناسب‌ترین کروموزوم‌ها از طریق جابجایی تصادفی ژن‌های نسل قبل به وجود می‌آیند و به

حیات خود ادامه می‌دهند. به این ترتیب، این روش با شبیه‌سازی فرآیندهای ژنتیک طبیعی سعی در حل مسائل پیچیده دنیای واقعی که با روش‌های بهینه‌سازی متداول قابل‌حل نیستند، دارد [۲۰]. امروزه، الگوریتم‌های ژنتیک جای خود را در میان روش‌های بهینه‌سازی مسائل پیچیده به‌خوبی بازکرده و در حل مسائلی مانند زمان‌بندی و توالی عملیات [۲۱]، فناوری گروهی، جایابی و چیدمان تسهیلات، مسیریابی و حمل‌ونقل [۲۲؛ ۲۳]، موفقیت‌های زیادی داشته است.

الگوریتم ژنتیک در حل مسائل بهینه‌سازی، پیش‌نیاز ریاضی خاصی نداشته و بدون توجه به عملکرد درونی مسئله به حل مسئله بهینه‌سازی می‌پردازد [۱۹]؛ در ضمن، ساختار عملگرهای آن، این الگوریتم را قادر می‌سازد تا در یافتن جواب‌های بهینه کلی موفق عمل کند و این درحالی است که در روش‌های مرسوم، جستجو از طریق مقایسه با نقاط همسایه انجام گرفته و حرکت به‌سوی نقاط بهینه نسبی صورت می‌گیرد. نکته قابل‌توجه در الگوریتم ژنتیک، ثابت بودن اندازه جمعیت در طی نسل‌های مختلف است. کروموزوم‌های جدید حاصل‌شده در نسل بعدی، نوزاد^۱ نامیده می‌شود که با اعمال عملگرهایی مانند عملگر بازتولید^۱، عملگر تقاطعی^{۱۱} و عملگر جهشی^{۱۲} روی نسل موجود حاصل می‌شوند. سپس برانزنگی کروموزوم‌ها توسط یکی از فنون انتخاب^{۱۳} ارزیابی می‌شود و کروموزوم‌های بهتر انتخاب شده و بدین ترتیب اندازه جمعیت در نسل بعدی نیز ثابت خواهد ماند [۲۴]. برای هر مسئله ژنتیک، یک تابع برانزنگی تعریف می‌شود که همان مقدار تابع هدف است. پس از چندین تکرار، الگوریتم به سمت بهترین کروموزوم همگرا می‌شود که جواب بهینه یا نزدیک به بهینه خواهد بود.

عملگر تقاطع، دو نقطه از فضای جواب را بر مبنای بیشترین میزان انطباق به‌عنوان والدین انتخاب کرده و اقدام به تبادل اجزاء آن نقاط می‌نماید تا نقاط جدید حاصل شوند، در ضمن برای آن احتمال در نظر گرفته می‌شود. چندین نوع مختلف از این عملگر وجود دارد که در این مسئله از عملگر تقاطع تک نقطه‌ای با توجه به قابلیت این عملگر در حل مسائل جایگشتی^{۱۴}، استفاده شده است.

گرچه عملگرهای تقاطع نقش ارزنده‌ای را در کاوش فضای جستجو ایفا کرده و موجب بهبود فرآیند جستجو می‌شوند، ولی دارای نواقصی نیز هستند، زیرا این عملگر صرفاً اطلاعات ژن‌های والدین را ترکیب می‌کند؛ لذا ممکن است در برخی مراحل

الگوریتم ژن‌های مفیدی حذف شده باشند و بازیابی آن‌ها امکان‌پذیر نباشد. در این نوشتار از عملگر تعویض^{۱۵} استفاده شده است. همچنین به هنگام استفاده از الگوریتم ژنتیک باید در خصوص تعیین اندازه جمعیت ($nPop$) تصمیم گرفته شود. اندازه جمعیت تأثیر فراوانی در عملکرد و راندمان الگوریتم‌های ژنتیک دارد. اگر تعداد جمعیت خیلی کم باشد، راندمان الگوریتم پایین خواهد آمد و اگر تعداد جمعیت خیلی زیاد در نظر گرفته شود، الگوریتم‌های ژنتیک جستجوی آگاهانه‌تری خواهند داشت ولی این تعداد زیاد باعث می‌شود که یک همگرایی زود هنگام به سمت نقاط زیر بهینه به دست آید؛ همچنین سرعت همگرایی کند و آهسته خواهد بود. در این نوشتار برای تعداد جمعیت اولیه عدد ۱۰۰ در نظر گرفته شده است. ساختار کروموزوم در الگوریتم ژنتیک، همان ساختار پاسخ است که در قسمت قبل تشریح شد.

۴-۲- الگوریتم شبیه‌سازی تبرید

الگوریتم شبیه‌سازی تبرید یا سرد کردن تدریجی در دهه هشتاد میلادی توسط کرک پاتریک و همکاران معرفی شد [۲۵]. شبیه‌سازی تبریدی یک روش جستجوی محلی است که سعی در به دست آوردن جواب بهینه کلی دارد [۲۶] و به دلیل سادگی و همچنین کارایی به‌طور گسترده در زمینه‌های گوناگون مسائل بهینه‌سازی ترکیبی به کار گرفته شده است. این الگوریتم می‌تواند با هر سیستم اختیاری و تابع هزینه ارتباط پیدا کند و به‌طور احتمالی جواب بهینه را ارائه نماید. الگوریتم شبیه‌سازی تبرید، تغییرات انرژی سیستم مذاب را با توجه به یک فرایند خنک‌سازی، به‌منظور دستیابی به وضعیت تعادلی پایدار، شبیه‌سازی می‌کند. این الگوریتم یک الگوریتم تصادفی است که توانایی پذیرش جواب‌هایی با کیفیت بدتر را دارد [۲۷]. هدف از این کار، فرار از نقطه بهینه محلی و به تأخیر انداختن همگرایی زودرس است. در این الگوریتم، همواره جواب‌هایی که باعث بهبود در تابع هدف می‌شوند، پذیرفتنی هستند. در غیر این صورت، همسایه به‌دست‌آمده با احتمالی که به تفاوت در تابع هدف (انرژی) جواب فعلی و جواب همسایه به‌دست‌آمده بستگی دارد، موردپذیرش واقع خواهد شد. با ادامه الگوریتم، احتمال بازگشت به عقب کمتر می‌شود [۲۵]. الگوریتم SA در هر تکرار t ، یک جواب x به‌صورت تصادفی از همسایگی $N(x_t)$ انتخاب می‌کند.

چنانچه $f(x) \leq f(x_t)$ باشد، x_{t+1} را برابر x قرار می‌دهیم؛ در غیر این صورت، رابطه زیر را داریم [۲۵]:

$$x_{t+1} = \begin{cases} x & \text{احتمال } p_t \\ x_t & \text{احتمال } 1-p_t \end{cases} \quad (5)$$

که در رابطه فوق، p_t به صورت زیر تعریف می‌شود:

$$P_t = \exp\left(-\frac{[f(x)-f(x_t)]}{\theta t}\right) \quad (6)$$

در رابطه فوق، θt بیان‌کننده دما و یا درجه حرارت در تکرار t است. در ابتدا، θt برابر با یک مقدار مثبت قرار می‌گیرد که بعد از هر T تکرار در یک ضریب a که $0 < a < 1$ است، ضرب می‌شود. در هر مرحله از الگوریتم SA، یک حالت جدید از سیستم ساخته می‌شود که اگر انرژی مربوط به این حالت جدید پایین‌تر از حالت فعلی باشد، عمل جایگزینی صورت می‌گیرد و حالت جدید به‌عنوان حالت فعلی پذیرفته می‌شود. در گام نخست الگوریتم پیشنهادی، یک جواب اولیه به صورت تصادفی و ابتکاری ساخته می‌شود که این پاسخ همانند آنچه در الگوریتم ژنتیک توضیح داده شد، است.

۳-۴- الگوریتم ازدحام ذرات

الگوریتم ازدحام ذرات^{۱۶} در سال ۱۹۹۵، برای اولین بار توسط ابره‌ارت^{۱۷} و کندی^{۱۸} به‌عنوان یک روش جستجوی غیرقطعی برای بهینه‌سازی مطرح شد. این الگوریتم بر اساس الگوهای حاکم بر پرواز هم‌زمان پرندگان و تغییر ناگهانی مسیر آن‌ها و تغییر شکل بهینه دسته‌جمعی به کار گرفته شد. در PSO، ذرات^{۱۹} در فضای جستجو جاری می‌شوند. تغییر مکان ذرات در فضای جستجو تحت تأثیر تجربه و دانش خودشان و همسایگانشان است [۲۸].

بنابراین، موقعیت توده^{۲۰} ذرات روی چگونگی جستجوی یک ذره اثر می‌گذارد. نتیجه مدل‌سازی این رفتار اجتماعی، فرآیند جستجویی است که در آن ذرات به سمت نواحی موفق میل می‌کنند. ذرات از یکدیگر می‌آموزند و بر مبنای دانش به‌دست‌آمده به سمت بهترین همسایگان خود می‌روند. در این روش، از مفهوم «جانمایی همسایگی جی بست^{۲۱}» که توسط کندی و دیگران معرفی شده، استفاده می‌شود؛ یعنی هر ذره بهترین موقعیت قبلی خود و هر یک از ذرات موجود در جمعیت را به خاطر می‌آورد و مکان خود را در فضای جستجو با توجه به بهترین مکانی که تاکنون در آن قرار گرفته و بهترین مکانی که در کل همسایگی‌اش وجود دارد، تنظیم می‌کند. هر راه‌حل که به آن یک ذره گفته می‌شود، توسط یک تابع شایستگی محاسبه می‌شود و دارای یک سرعت است که هدایت حرکت ذره را بر عهده دارد. در ابتدا، گروهی از ذرات به‌صورت تصادفی به وجود می‌آیند و با به‌روز کردن نسل‌ها سعی در یافتن راه‌حل بهینه می‌شود [۲۹].

در یک فضای D بعدی، i امین ذره از جمعیت می‌تواند با یک بردار سرعت و یک بردار موقعیت، نمایش داده شود. تغییر موقعیت هر ذره با تغییر در ساختار موقعیت و سرعت قبلی امکان‌پذیر است. هر ذره با استفاده از اطلاعاتی شامل بهترین مقداری که تاکنون به آن رسیده ($pbest$) و بهترین جوابی که تاکنون در گروه به‌دست‌آمده است ($gbest$)، سعی می‌کند برای رسیدن به بهترین جواب، موقعیت خود را تغییر دهد. بدین ترتیب سرعت هر ذره، طبق رابطه‌ی (۱) تغییر می‌کند:

$$v_{ij}(t+1) = w v_{ij}(t) + c_1 r_1 (pbest_{ij}(t) - x_{ij}(t)) + c_2 r_2 (gbest_{ij}(t) - x_{ij}(t)) \quad (7)$$

در این رابطه، موقعیت کنونی x_i ، سرعت کنونی v_i ، وزن اینرسی w ، عامل وزنی c_1 و c_2 و درنهایت r_1 و r_2 اعداد تصادفی بین ۰ و ۱ هستند. بنا به نتایج تجربی به‌دست‌آمده $w=0/7298$ و $c_1, c_2=1/49618$ منجر به همگرایی مناسب الگوریتم می‌شود [۳۰]. موقعیت جدید هر ذره به‌وسیله مجموع موقعیت گذشته و سرعت جدید تعیین می‌شود.

سمت راست معادله (۷) از سه قسمت تشکیل شده است که قسمت اول، سرعت فعلی ذره است و قسمت‌های دوم و سوم، تغییر سرعت ذره و چرخش آن به سمت بهترین تجربه شخصی و بهترین تجربه گروه را به عهده دارند. اگر قسمت اول را در این معادله در نظر نگیریم، آنگاه سرعت ذرات تنها با توجه به موقعیت فعلی و بهترین ذره و بهترین تجربه جمع تعیین می‌شود. به این ترتیب، بهترین ذره گروه در جای خود ثابت می‌ماند و سایرین به سمت آن ذره حرکت می‌کنند. در واقع، حرکت دسته‌جمعی ذرات بدون قسمت اول معادله (۷)، فرآیندی خواهد بود که طی آن فضای جستجو به تدریج کوچک می‌شود و جستجویی محلی حول بهترین ذره شکل می‌گیرد. در مقابل، اگر فقط قسمت اول معادله (۷) را در نظر بگیریم، ذرات راه عادی خود را می‌روند تا به دیواره محدوده برسند و به نوعی جستجویی سراسری را انجام می‌دهند. تعداد ذرات در این الگوریتم، همان تعداد پاسخ اولیه در الگوریتم ژنتیک است.

۴-۴- بررسی اثربخشی و کارایی الگوریتم‌ها

اثربخشی الگوریتم‌ها بر اساس مقدار تابع هدف و کارایی آن‌ها بر اساس مدت‌زمان حل مقایسه می‌شوند. جهت یافتن بهترین الگوریتم و مقایسه پاسخ‌ها، مطالعه‌ای آماری بر نتایج به دست آمده توسط الگوریتم‌ها، در کلیه مسائل با اندازه‌های مختلف، انجام می‌گیرد. در این راستا، به منظور آنکه زمینه‌ای برای مقایسه پاسخ‌های به دست آمده از الگوریتم‌ها در مسائل با ابعاد مختلف ایجاد شود، پاسخ‌ها به شاخص درصد انحراف نسبی^{۲۲} (RPD) تبدیل می‌شوند [۳۱]. نحوه محاسبه این شاخص به صورت زیر است:

$$RPD = \frac{|ALg_{sol} - Min_{sol}|}{Min_{sol}} \quad (۸)$$

در این رابطه، ALg_{sol} مقدار پاسخ به دست آمده توسط الگوریتم برای یک مسئله مفروض است و مقدار بهترین Min_{sol} پاسخ به دست آمده از Min_{sol} همان مسئله توسط کل الگوریتم‌هاست. همان‌طور که واضح است، مقدار کمتر RPD، مورد نظر است. کلیه مسائل حل و مقادیر تابع هدف در جدول زیر ارائه شده است.

جدول ۱ مقدار تابع هدف مسائل با استفاده از الگوریتمها

| m | n | PSO | | | SA | | | GA | | |
|----|----|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| | | TF=۰/۴ | TF=۰/۲ | TF=۰/۱ | TF=۰/۴ | TF=۰/۲ | TF=۰/۱ | TF=۰/۴ | TF=۰/۲ | TF=۰/۱ |
| ۵ | ۵ | ۹۱/۲ | ۸۲/۲ | ۲۲/۸ | ۹۱/۲ | ۸۲/۲ | ۲۲/۸ | ۹۱/۲ | ۸۲/۲ | ۲۲/۸ |
| | ۱۰ | ۱۲۱/۸ | ۱۳۴/۱ | ۱۲۸/۳ | ۱۲۱/۸ | ۱۳۴/۱ | ۱۲۸/۳ | ۱۲۱/۸ | ۱۳۴/۱ | ۱۲۸/۳ |
| | ۱۵ | ۳۴۹/۱ | ۶۴۶/۹ | ۴۵۷/۶ | ۳۴۹/۱ | ۶۴۶/۹ | ۴۵۷/۶ | ۳۴۹/۱ | ۶۴۶/۹ | ۴۵۷/۶ |
| | ۲۰ | ۷۲۰/۴ | ۷۵۸/۸ | ۴۳۶/۸ | ۷۲۰/۴ | ۷۵۸/۸ | ۴۳۶/۸ | ۷۲۰/۴ | ۷۵۸/۸ | ۴۳۶/۸ |
| ۱۰ | ۵ | ۱۱۳/۳ | ۹۷/۸ | ۱۱۵/۹ | ۱۱۰/۰ | ۱۱۲/۱ | ۱۱۸/۰ | ۱۰۰/۲ | ۸۹/۴ | ۱۱۵/۹ |
| | ۱۰ | ۵۶۸/۷ | ۶۵۲/۱ | ۳۵۲/۹ | ۵۴۵/۰ | ۶۱۰/۲ | -/۲۸۸ | ۵۴۵/۰ | ۶۱۰/۲ | ۲۸۱/۱ |
| | ۱۵ | ۱۹۳۴/۵ | ۱۳۰۶/۴ | ۱۰۳۳/۵ | ۱۹۳۴/۵ | ۱۲۴۰/۲ | ۹۷۲/۷ | ۱۹۳۴/۵ | ۱۲۴۰/۲ | ۹۷۲/۷ |
| | ۲۰ | ۲۰۸۳/۰ | ۲۰۵۰/۴ | ۲۱۳۲/۵ | ۲۰۸۳/۰ | ۲۰۰۲/۲ | ۲۰۷۹/۷ | ۲۰۸۳/۰ | ۱۹۴۱/۵ | ۲۰۷۹/۷ |
| ۱۵ | ۵ | ۵۶۱/۱ | ۴۴۸/۶ | ۲۶۶/۲ | ۴۰۳/۰ | ۳۱۴/۸ | ۷۰/۳ | ۳۹۴/۲ | ۳۱۴/۸ | ۵۸/۳ |
| | ۱۰ | ۲۳۸۵/۸ | ۵۹۵/۳ | ۶۳۱/۰ | ۲۱۹۰/۴ | ۴۷۶/۴ | ۴۲۴/۲ | ۲۱۹۳/۸ | ۴۵۰/۴ | ۳۹۰/۶ |
| | ۱۵ | ۱۹۵۶/۲ | ۱۵۵۰/۴ | ۲۰۸۵/۹ | ۱۷۰۹/۸ | ۱۳۶۰/۰ | ۱۵۷۸/۵ | ۱۷۱۲/۲ | ۱۲۶۶/۵ | ۱۶۲۵/۴ |
| | ۲۰ | ۳۰۸۱/۹ | ۲۶۶۹/۵ | ۲۱۵۵/۷ | ۲۷۴۸/۰ | ۲۱۲۷/۰ | ۱۷۹۳/۷ | ۲۷۴۸/۰ | ۲۱۲۷/۰ | ۱۶۷۳/۲ |
| ۲۰ | ۵ | ۱۵۶۲/۱ | ۱۳۴۴/۵ | ۷۶۳/۲ | ۹۳۴/۲ | ۶۸۴/۰ | ۵۵۴/۸ | ۸۵۴/۸ | ۶۴۳/۸ | ۶۲۵/۸ |
| | ۱۰ | ۳۲۷۹/۷ | ۱۰۵۷/۱ | ۱۸۹۰/۸ | ۲۳۶۵/۰ | ۶۶۹/۳ | ۱۴۰۰/۱ | ۲۴۹۴/۰ | ۸۴۲/۹ | ۱۱۶۹/۰ |
| | ۱۵ | ۴۹۰۷/۷ | ۱۹۲۵/۵ | ۲۹۴۷/۲ | ۴۰۰۱/۶ | ۱۵۲۷/۷ | ۲۲۳۰/۰ | ۴۱۵۸/۴ | ۱۵۴۰/۰ | ۲۱۲۵/۹ |
| | ۲۰ | ۷۷۳۰/۵ | ۶۵۰۱/۳ | ۵۰۶۷/۹ | ۶۳۲۷/۷ | ۴۶۰۷/۱ | ۳۱۸۹/۰ | ۶۳۲۴/۷ | ۴۷۱۸/۶ | ۲۹۹۴/۹ |
| ۲۵ | ۵ | ۲۲۲۱/۲ | ۲۵۹۰/۴ | ۳۲۵۰/۶ | ۱۲۶۸/۳ | ۱۶۲۵/۰ | ۱۹۶۰/۲ | ۱۳۳۸/۹ | ۱۷۱۸/۸ | ۱۸۷۸/۲ |
| | ۱۰ | ۳۵۷۱/۳ | ۳۱۵۰/۹ | ۲۲۶۶/۴ | ۲۵۱۹/۶ | ۱۴۴۹/۵ | ۱۲۱۹/۷ | ۲۵۸۲/۰ | ۱۴۵۲/۲ | ۱۲۷۴/۹ |
| | ۱۵ | ۹۲۹۴/۰ | ۳۴۲۷/۱ | ۵۰۴۰/۷ | ۷۴۸۱/۳ | ۲۲۳۷/۶ | ۲۷۳۹/۹ | ۷۶۲۰/۹ | ۲۴۴۷/۶ | ۲۸۵۳/۰ |
| | ۲۰ | ۹۹۱۱/۱ | ۱۰۵۲۱/۶ | ۷۷۹۶/۰ | ۸۳۶۸/۷ | ۸۴۴۰/۸ | ۵۵۵۴/۶ | ۸۶۲۷/۶ | ۸۳۵۱/۵ | ۵۸۵۵/۵ |
| ۳۰ | ۵ | ۶۲۰۷/۶ | ۴۵۴۳/۶ | ۶۱۷۳/۸ | ۴۶۷۳/۸ | ۳۲۲۶/۴ | ۳۶۸۳/۷ | ۴۶۶۱/۶ | ۳۱۵۳/۲ | ۳۸۶۰/۲ |
| | ۱۰ | ۳۷۸۹/۸ | ۵۲۹۶/۳ | ۴۹۳۰/۹ | ۱۷۱۰/۳ | ۲۱۷۰/۳ | ۱۹۴۲/۷ | ۲۰۳۹/۴ | ۲۵۶۹/۵ | ۲۰۴۰/۶ |
| | ۱۵ | ۸۴۳۵/۹ | ۶۱۱۷/۶ | ۷۲۴۰/۴ | ۵۵۸۷/۱ | ۳۰۴۴/۷ | ۴۸۵۵/۲ | ۵۷۹۵/۲ | ۳۲۷۶/۰ | ۴۶۷۲/۲ |
| | ۲۰ | ۱۰۷۴۳/۷ | ۹۳۰۵/۹ | ۱۰۱۹۰/۰ | ۶۹۸۴/۲ | ۶۱۷۷/۴ | ۶۸۲۱/۲ | ۷۳۷۶/۵ | ۵۶۶۷/۳ | ۷۴۶۶/۴ |
| ۳۵ | ۵ | ۱۱۶۸/۸ | ۹۱۹۹/۱ | ۱۱۳۳۱/۵ | ۹۵۹۶/۰ | ۷۵۲۹/۱ | ۸۷۶۰/۰ | ۹۸۶۵/۸ | ۷۸۶۱/۶ | ۹۰۵۵/۰ |
| | ۱۰ | ۹۴۲۵/۴ | ۸۲۰۹/۴ | ۵۳۱۷/۱ | ۵۵۴۸/۲ | ۳۱۳۰/۰ | ۱۶۶۷/۸ | ۵۸۶۱/۳ | ۳۵۳۲/۴ | ۲۰۹۶/۱ |
| | ۱۵ | ۱۲۱۰۲/۰ | ۱۰۳۶۸/۱ | ۱۱۰۰۱/۰ | ۶۵۶۷/۱ | ۵۹۶۹/۸ | ۴۷۳۰/۴ | ۶۶۶۸/۷ | ۵۹۶۵/۱ | ۵۸۵۴/۸ |
| | ۲۰ | ۱۳۸۷۵/۸ | ۱۰۶۱۰/۰ | ۱۵۶۴۹/۷ | ۹۳۳۲/۰ | ۷۱۲۲/۳ | ۷۷۰۱/۹ | ۱۰۴۵۱/۱ | ۷۳۵۹/۱ | ۹۹۳۲/۲ |
| ۴۰ | ۵ | ۱۹۷۵۸/۰ | ۱۵۸۶۱/۸ | ۱۹۸۹۱/۹ | ۱۵۰۱۶/۹ | ۱۳۲۱۱/۱ | ۱۵۴۵۶/۷ | ۱۵۶۴۰/۸ | ۱۳۴۲۵/۷ | ۱۵۵۳۲/۱ |
| | ۱۰ | ۱۱۶۸۳/۱ | ۷۴۶۶/۷ | ۱۰۵۷۳/۳ | ۵۱۳۳/۳ | ۲۲۷۶/۳ | ۵۳۱۵/۵ | ۵۸۵۰/۸ | ۲۸۶۶/۳ | ۵۹۳۹/۳ |
| | ۱۵ | ۲۲۰۸۵/۱ | ۱۲۳۱۲/۱ | ۱۲۸۰۱/۵ | ۱۴۴۳۶/۶ | ۴۵۳۱/۳ | ۴۵۵۹/۴ | ۱۴۹۶۰/۹ | ۵۹۸۲/۹ | ۶۱۸۹/۸ |
| | ۲۰ | ۲۱۹۳۷/۸ | ۲۱۷۱۴/۹ | ۱۳۹۴۰/۳ | ۱۵۳۳۷/۸ | ۱۳۷۴۷/۵ | ۸۱۴۴/۳ | ۱۵۸۷۸/۵ | ۱۵۷۷۶/۰ | ۸۹۶۳/۸ |

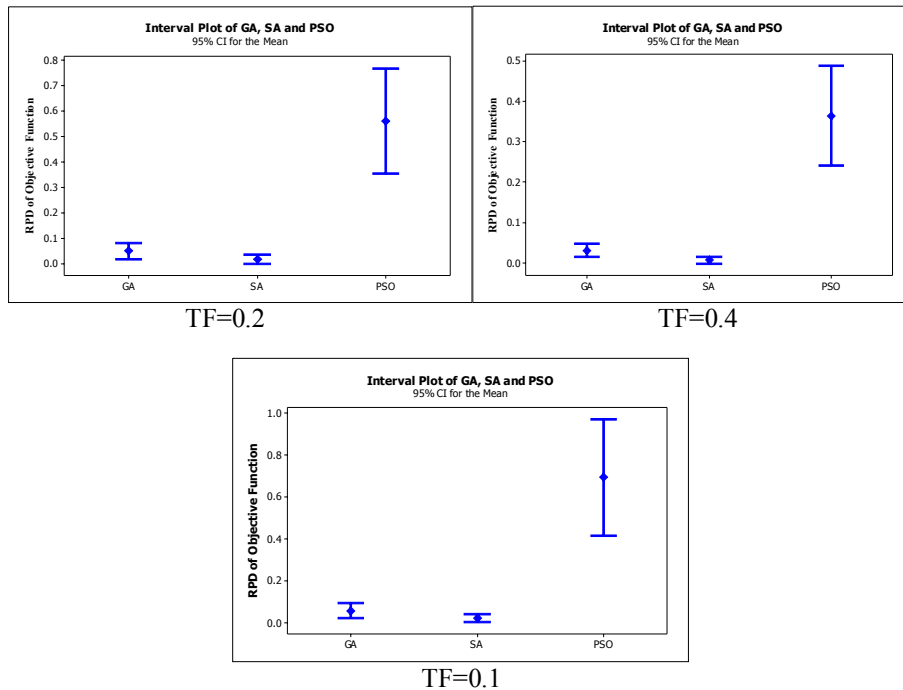
جهت مقایسه نتایج به دست آمده الگوریتمها از آزمون فرض ANOVA استفاده شد. در این راستا نتایج آزمونها در جدول ۲ ارائه شده است. سپس با استفاده از آزمون توکی^{۳۳} الگوریتمها دوبه دو مقایسه شدند که جدول ۳ نشان دهنده آن است.

جدول ۲ نتایج آزمون ANOVA بر شاخص‌های مقایسه اثر بخشی

| سطح معناداری (P) | آماره F | مربع میانگین | مجموع مربعات | درجه آزادی | نوع خطا | |
|------------------|---------|--------------|--------------|------------|-----------|--------|
| ۰,۰۰۰ | ۲۲/۸۰ | ۴/۵۱ | ۹/۰۳۷ | ۲ | بین گروه | TF=0.1 |
| | | ۰/۱۹۸ | ۱۸/۴۳۳ | ۹۳ | درون گروه | |
| | | | ۲۷/۴۷ | ۹۶ | کل | |
| ۰,۰۰۰ | ۲۶/۲۹ | ۲/۹۴۵ | ۵/۹۴۵ | ۲ | بین گروه | TF=0.2 |
| | | ۰/۱۱۲ | ۱۰/۴۱۷ | ۹۳ | درون گروه | |
| | | | ۱۶/۳۰۷ | ۹۶ | کل | |
| ۰,۰۰۰ | ۳۲/۲۲ | ۱/۲۷۹ | ۲/۵۵۹ | ۲ | بین گروه | TF=0.4 |
| | | ۰/۰۳۹ | ۳/۶۹۳ | ۹۳ | درون گروه | |
| | | | ۶/۲۵۳ | ۹۶ | کل | |
| ۰,۰۰۰ | ۶۸/۴۹ | ۸/۹۴۵ | ۱۶/۴۸۷ | ۲ | بین گروه | کل |
| | | ۰/۱۲۰ | ۳۴/۳۰۲ | ۲۸۵ | درون گروه | |
| | | | ۵۰/۷۹۰ | ۲۸۷ | کل | |

جدول ۳- مقایسه زوجی مقادیر تابع هدف بر اساس آزمون توکی

| | الگوریتم‌ها | حد پائین | حد بالا | تفاوت معنا دار (٪۹۵) |
|--------|-------------|----------|---------|----------------------|
| TF=۰/۸ | GA و SA | -۰/۲۹۹۲ | ۰/۲۳۱۲ | نه |
| | GA و PSO | ۰/۳۶۸۰ | ۰/۸۹۸۴ | بله |
| | SA و PSO | ۰/۴۰۲۰ | ۰/۹۳۲۴ | بله |
| TF=۰/۲ | GA و SA | -۰/۲۳۲۲ | ۰/۱۶۶۶ | نه |
| | GA و PSO | ۰/۳۰۸۹ | ۰/۷۰۷۷ | بله |
| | SA و PSO | ۰/۳۴۱۷ | ۰/۷۴۰۵ | بله |
| TF=۰/۴ | GA و SA | -۰/۱۴۲۸ | ۰/۰۹۴۷ | نه |
| | GA و PSO | ۰/۲۱۵۰ | ۰/۴۵۲۵ | بله |
| | SA و PSO | ۰/۲۳۹۱ | ۰/۴۷۶۵ | بله |
| کل | GA و SA | -۰/۱۴۷۵ | ۰/۰۸۶۹ | نه |
| | GA و PSO | ۰/۳۷۴۵ | ۰/۶۰۸۹ | بله |
| | SA و PSO | ۰/۴۰۴۸ | ۰/۶۳۹۲ | بله |



شکل ۱ میانگین و نمودار فاصله‌ای تابع هدف

اگرچه برتری ضعیفی بین نتایج روش SA نسبت به روش GA قابل مشاهده است، ولی همان‌طور که در جدول بالا نتایج مقایسه‌های آماری نشان داده شده است، در کلیه TF های مختلف، تفاوت معناداری بین اثربخشی نتایج GA و SA وجود ندارد. همچنین نتایج حاکی از وجود تفاوت معناداری بین هر دو الگوریتم با روش PSO در کلیه مسائل است. در مجموع می‌توان بیان کرد که الگوریتم‌های گسسته از قبیل GA و SA نتایج مناسب‌تری نسبت به روش‌های پیوسته مانند PSO در این مسئله زمان‌بندی تعریف شده، نشان داده‌اند. قابل‌ذکر است که در مسائل با ابعاد کوچک جواب‌ها به یکدیگر بسیار نزدیک‌اند، اما در مسائل بزرگ تفاوت قابل مشاهده است.

همچنین جهت بررسی کارایی الگوریتم‌ها، زمان محاسبه (CPU time) هر یک به دست آمده و در جدول ۴ نشان داده شده است. همچنین نتایج آزمون ANOVA جهت مقایسه کارایی الگوریتم‌ها، در جدول ۵ (میزان معنی‌داری آزمون $P < 0.05$) و شکل ۲ ارائه شده است. در ضمن، از آزمون توکی برای مقایسه دوبه‌دو زمان حل توسط الگوریتم‌ها نیز استفاده شد.

جدول ۴ زمان حل مسائل توسط الگوریتمها (ثانیه)

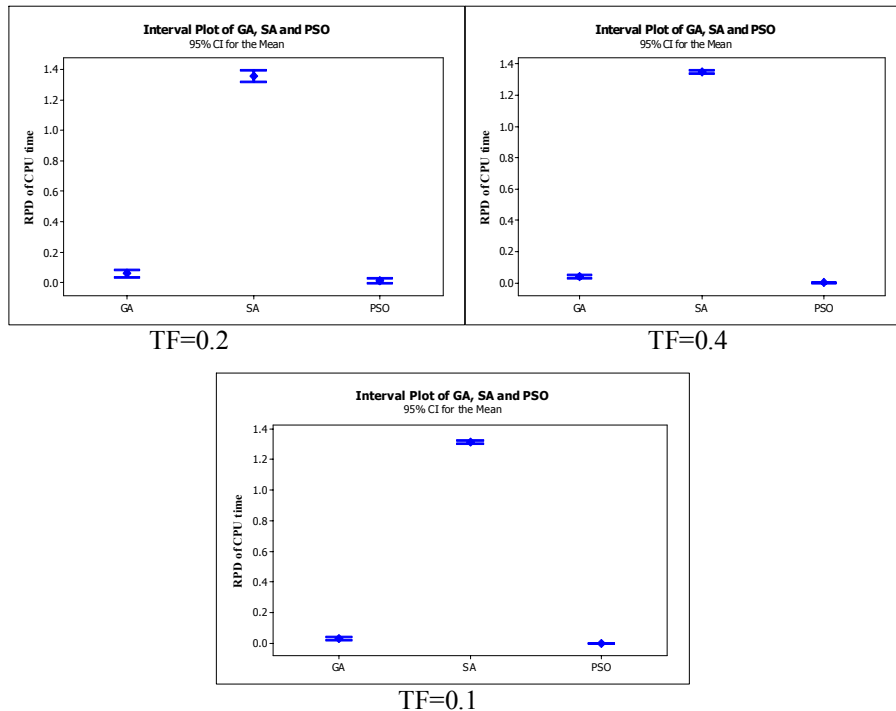
| m | n | PSO | | | SA | | | GA | | |
|----|----|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| | | TF=۰/۴ | TF=۰/۲ | TF=۰/۱ | TF=۰/۴ | TF=۰/۲ | TF=۰/۱ | TF=۰/۴ | TF=۰/۲ | TF=۰/۱ |
| ۵ | ۵ | ۸/۴۶ | ۸/۷۹ | ۸/۸۵ | ۱۹/۰۳ | ۱۹/۹۳ | ۱۹/۸۷ | ۸.۳۳ | ۸/۸۷ | ۸/۷۰ |
| | ۱۰ | ۸/۴۸ | ۸/۷۶ | ۹/۰۳ | ۱۹/۲۴ | ۱۹/۹۴ | ۲۰/۵۹ | ۸/۴۲ | ۸/۷۹ | ۸/۹۴ |
| | ۱۵ | ۸/۶۵ | ۸/۷۹ | ۹/۰۴ | ۱۹/۳۷ | ۲۰/۰۹ | ۲۰/۴۹ | ۸/۴۸ | ۸/۹۱ | ۹/۰۴ |
| | ۲۰ | ۸/۶۵ | ۸/۸۵ | ۹/۱۸ | ۱۹/۴۷ | ۲۰/۱۷ | ۲۰/۹۸ | ۸/۵۱ | ۸/۹۰ | ۹/۲۲ |
| ۱۰ | ۵ | ۸/۶۲ | ۸/۸۱ | ۸/۹۷ | ۱۹/۹۲ | ۲۰/۵۳ | ۲۰/۷۸ | ۹/۰۹ | ۹/۳۶ | ۹/۵۵ |
| | ۱۰ | ۸/۵۷ | ۸/۹۴ | ۹/۱۲ | ۲۰/۰۹ | ۲۰/۹۷ | ۲۱/۰۱ | ۹/۰۳ | ۹/۵۸ | ۹/۵۵ |
| | ۱۵ | ۸/۶۴ | ۹/۱۰ | ۹/۲۴ | ۲۰/۵۰ | ۲۱/۳۲ | ۲۱/۳۸ | ۹/۱۹ | ۹/۶۷ | ۹/۶۸ |
| | ۲۰ | ۸/۸۲ | ۹/۲۵ | ۹/۳۷ | ۲۰/۸۳ | ۲۱/۶۹ | ۲۱/۶۶ | ۹/۳۱ | ۹/۸۳ | ۹/۷۶ |
| ۱۵ | ۵ | ۸/۵۲ | ۸/۹۴ | ۹/۱۰ | ۲۰/۱۴ | ۲۰/۹۵ | ۲۱/۰۰ | ۹/۱۵ | ۹/۶۳ | ۹/۶۴ |
| | ۱۰ | ۸/۷۲ | ۹/۹۹ | ۹/۳۴ | ۲۰/۶۹ | ۲۱/۴۷ | ۲۱/۶۱ | ۹/۳۴ | ۹/۷۷ | ۹/۸۲ |
| | ۱۵ | ۹/۰۰ | ۹/۶۵ | ۹/۶۲ | ۲۱/۲۶ | ۲۴/۹۱ | ۲۲/۲۵ | ۹/۴۸ | ۱۰/۸۷ | ۹/۹۸ |
| | ۲۰ | ۹/۱۷ | ۱۰/۳۴ | ۹/۷۹ | ۲۱/۶۹ | ۲۳/۴۳ | ۲۲/۶۲ | ۹/۶۴ | ۱۰/۲۷ | ۱۰/۱۳ |
| ۲۰ | ۵ | ۸/۶۰ | ۱۰/۲۳ | ۹/۱۸ | ۲۰/۴۶ | ۲۲/۹۰ | ۲۱/۹۵ | ۹/۲۹ | ۱۰/۹۰ | ۹/۷۹ |
| | ۱۰ | ۸/۹۱ | ۹/۸۸ | ۹/۴۹ | ۲۱/۲۵ | ۲۴/۰۴ | ۲۲/۱۲ | ۹/۵۵ | ۱۱/۴۱ | ۱۰/۰۳ |
| | ۱۵ | ۹/۲۱ | ۱۰/۶۵ | ۹/۹۷ | ۲۱/۹۳ | ۲۳/۶۸ | ۲۲/۷۱ | ۹/۸۲ | ۱۰/۶۴ | ۱۰/۱۳ |
| | ۲۰ | ۹/۵۱ | ۱۰/۱۳ | ۱۰/۱۵ | ۲۲/۵۶ | ۲۴/۹۵ | ۲۳/۳۹ | ۹/۸۹ | ۱۱/۶۳ | ۱۰/۴۰ |
| ۲۵ | ۵ | ۸/۷۸ | ۹/۷۴ | ۹/۳۷ | ۲۰/۸۷ | ۲۱/۸۷ | ۲۱/۸۷ | ۹/۴۴ | ۹.۹۵ | ۹/۹۶ |
| | ۱۰ | ۹/۲۲ | ۱۰/۱۹ | ۹/۵۴ | ۲۱/۶۷ | ۲۶/۱۱ | ۲۲/۴۹ | ۹/۶۸ | ۱۲/۸۳ | ۱۰/۲۰ |
| | ۱۵ | ۹/۶۴ | ۹/۹۵ | ۱۰/۰۵ | ۲۲/۴۵ | ۲۳/۸۱ | ۲۳/۳۰ | ۹/۹۳ | ۱۰/۷۹ | ۱۰/۲۸ |
| | ۲۰ | ۱۰/۰۱ | ۱۱/۵۱ | ۱۰/۴۳ | ۲۳/۳۲ | ۲۴/۶۱ | ۲۴/۱۹ | ۱۰/۱۷ | ۱۰/۶۰ | ۱۰/۶۲ |
| ۳۰ | ۵ | ۸/۹۴ | ۹/۳۴ | ۹/۵۷ | ۲۱/۱۲ | ۲۲/۶۵ | ۲۱/۹۱ | ۹/۵۵ | ۱۰/۸۳ | ۹/۹۴ |
| | ۱۰ | ۹/۴۶ | ۱۰/۶۲ | ۱۰/۰۱ | ۲۲/۱۰ | ۲۳/۵۶ | ۲۲/۹۳ | ۹/۷۹ | ۱۰/۵۹ | ۱۰/۲۵ |
| | ۱۵ | ۹/۹۰ | ۱۰/۴۴ | ۱۰/۴۵ | ۲۳/۱۵ | ۲۵/۱۴ | ۲۳/۹۰ | ۱۰/۱۳ | ۱۱/۵۵ | ۱۰/۵۵ |
| | ۲۰ | ۱۰/۳۵ | ۱۲/۸۶ | ۱۰/۹۳ | ۲۴/۰۱ | ۲۵/۱۷ | ۲۴/۸۹ | ۱۰/۴۰ | ۱۰/۶۴ | ۱۰/۸۵ |
| ۳۵ | ۵ | ۹/۱۱ | ۹/۸۱ | ۹/۶۰ | ۲۱/۴۱ | ۲۲/۹۳ | ۲۲/۳۵ | ۹/۶۲ | ۱۰/۲ | ۱۰/۱۳ |
| | ۱۰ | ۹/۶۰ | ۱۰/۷۶ | ۱۰/۰۶ | ۲۲/۴۲ | ۲۳/۴۸ | ۲۳/۳۷ | ۹/۹۰ | ۱۰/۵۵ | ۱۰/۴۲ |
| | ۱۵ | ۹/۹۸ | ۱۰/۸۲ | ۱۰/۵۸ | ۲۳/۶۰ | ۲۷/۴۳ | ۲۴/۶۲ | ۱۰/۲۸ | ۱۱/۰۵ | ۱۰/۷۹ |
| | ۲۰ | ۱۰/۶۱ | ۱۰/۸۹ | ۱۱/۰۹ | ۲۴/۷۳ | ۲۷/۲۵ | ۲۵/۶۹ | ۱۰/۵۹ | ۱۲/۶۳ | ۱۱/۱۲ |
| ۴۰ | ۵ | ۹/۲۹ | ۹/۱۹ | ۹/۷۵ | ۲۱/۹۵ | ۲۲/۵۱ | ۲۲/۸۱ | ۹/۹۵ | ۱۰/۰۲ | ۱۰/۲۰ |
| | ۱۰ | ۹/۷۹ | ۹/۸۵ | ۱۰/۱۶ | ۲۳/۱۵ | ۲۴/۳۷ | ۲۴/۰۳ | ۱۰/۱۷ | ۱۰/۷۲ | ۱۰/۶۴ |
| | ۱۵ | ۱۰/۳۵ | ۱۱/۰۲ | ۱۰/۷۴ | ۲۴/۱۹ | ۲۵/۰۶ | ۲۴/۹۹ | ۱۰/۴۴ | ۱۰/۸۱ | ۱۰/۸۸ |
| | ۲۰ | ۱۰/۹۹ | ۱۱/۱۰ | ۱۱/۴۳ | ۲۵/۵۲ | ۲۶/۲۱ | ۲۶/۵۲ | ۱۰/۸۳ | ۱۱/۱۴ | ۱۱/۳۸ |

جدول ۵ نتایج آزمون در رابطه با زمان حل

| سطح معناداری (P) | آماره F | مربع میانگین | مجموع مربعات | درجه آزادی | نوع خطا | |
|------------------|----------|--------------|--------------|------------|-----------|--------|
| ۰/۰۰۰ | ۴۴۷۵۷/۴ | ۱۷/۹۹ | ۲۵/۹۹ | ۲ | بین گروه | TF=۰,۱ |
| | | ۰/۰۰۰۴ | ۰/۰۳۷ | ۹۳ | درون گروه | |
| | | | ۳۶/۰۳ | ۹۶ | کل | |
| ۰/۰۰۰ | ۳۳۷۷/۲ | ۱۸,۶۱ | ۳۷/۲ | ۲ | بین گروه | TF=۰,۲ |
| | | ۰/۰۰۵۵ | ۰/۵۱ | ۹۳ | درون گروه | |
| | | | ۳۷/۷۳ | ۹۶ | کل | |
| ۰/۰۰۰ | ۳۳۷۸۱/۱ | ۱۸/۷۲ | ۳۷/۴۵ | ۲ | بین گروه | TF=۰,۴ |
| | | ۰/۰۰۰۵ | ۰/۰۵۱ | ۹۳ | درون گروه | |
| | | | ۳۷/۵۱ | ۹۶ | کل | |
| ۰/۰۰۰ | ۲۴۴۵۱/۰۷ | ۵۵/۳۳ | ۱۱۰/۶۶ | ۲ | بین گروه | کل |
| | | ۰/۰۰۲۳ | | ۲۸۵ | درون گروه | |
| | | | | ۲۸۷ | کل | |

جدول ۶ مقایسات زوجی الگوریتم‌ها در رابطه با زمان حل

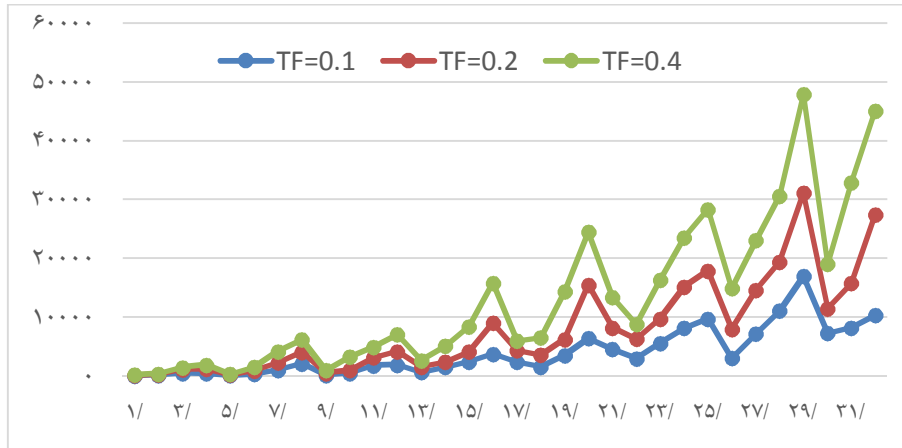
| | الگوریتم‌ها | حد پائین | حد بالا | تفاوت معنا دار (٪۹۵) |
|--------|-------------|----------|---------|----------------------|
| TF=۰/۱ | GA و SA | ۱/۲۷۱۵ | ۱/۲۹۵۴ | بله |
| | GA و PSO | -۰/۰۴۲۶ | -۰/۰۱۸۷ | بله |
| | SA و PSO | -۱/۳۲۶۰ | -۱/۳۰۲۱ | بله |
| TF=۰/۲ | GA و SA | ۱/۲۵۱۹ | ۱/۳۴۰۳ | بله |
| | GA و PSO | -۰/۰۹۲۴ | -۰/۰۰۴۰ | بله |
| | SA و PSO | -۱/۳۸۸۵ | -۱/۳۰۰۰ | بله |
| TF=۰/۴ | GA و SA | ۱/۲۹۱۹ | ۱/۳۲۰۰ | بله |
| | GA و PSO | -۰/۰۵۱۵ | -۰/۰۲۳۴ | بله |
| | SA و PSO | -۱/۳۵۷۴ | -۱/۳۲۹۴ | بله |
| کل | GA و SA | ۱/۲۷۹۱ | ۱/۳۱۱۲ | بله |
| | GA و PSO | -۰/۰۵۴۸ | -۰/۰۲۲۷ | بله |
| | SA و PSO | -۱/۳۵۰۰ | -۱/۳۱۷۸ | بله |



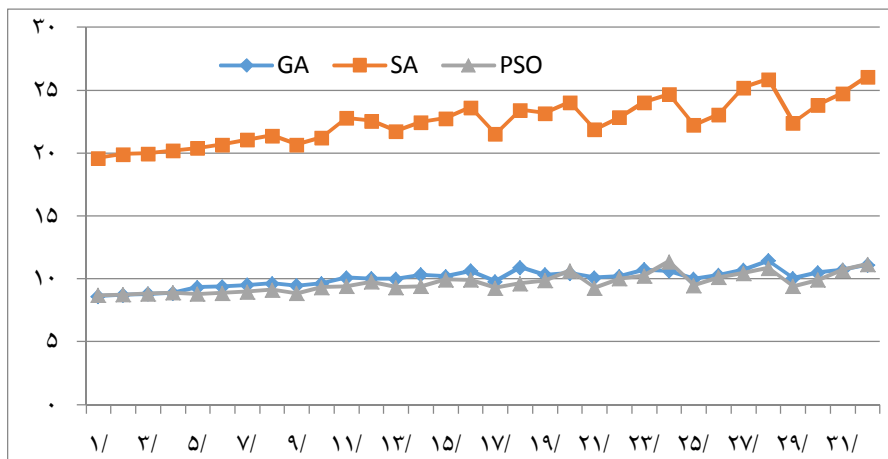
شکل ۲ میانگین و نمودار فاصله‌ای زمان حل مسائل (CPU time)

همان‌طور که در جداول بالا نشان داده شده است، طبق نتایج آماری، الگوریتم PSO نسبت به روش‌های SA و GA زمان حل کمتری را صرف می‌کند و همچنین روش GA نسبت به روش SA زمان کمتری را می‌طلبد. از این جهت، ابتدا PSO و سپس GA قرار می‌گیرد. در مورد زمان حل نیز تفاوت در مسائل کوچک، اندک ولی در مسائل بزرگ، زمان دو روش PSO و GA به یکدیگر نزدیک، اما با SA فاصله دارند.

شکل ۳ نشان‌دهنده روند جواب‌ها با TF های مختلف است. پاسخ‌های به‌دست‌آمده حاکی از آن است که میانگین تابع هدف با سه روش حل مسئله مورد بررسی، یعنی GA، SA و PSO، با افزایش میزان TF افزایش یافته است؛ به عبارتی میزان فشردگی موعد تحویل بر مقدار دیرکرد مؤثر است.



شکل ۳ روند پاسخ‌های به‌دست‌آمده از میانگین سه روش SA، GA و PSO با توجه به مقادیر مختلف TF



شکل ۴ روند میانگین زمان حل مسائل توسط الگوریتم‌های مورداستفاده

همان‌طور که در شکل ۴ نشان داده شده است، پاسخ‌های الگوریتم PSO نسبت به الگوریتم GA و SA زمان حل کمتری را جهت حل مسائل صرف می‌کند. در این شکل،

میانگین زمان حل سه روش در مسائل با TF های مختلف با یکدیگر مقایسه شده‌اند. زمان حل دو روش PSO و GA در برخی از موارد بسیار به هم نزدیک‌اند.

۵- نتیجه‌گیری

در این مقاله، زمان‌بندی کارها در یک مسئله جریان کارگاهی با در نظر گرفتن تابع هدف حداقل کردن مجموع دیرکردهای مرحله‌ای، معرفی شد. اگرچه مصادیقی مانند پروژه‌های تحقیقاتی و خدماتی که دارای موعدهای تحویل میانی هستند، برای چنین تابع هدفی وجود دارد، اما کمتر موردتوجه قرار گرفته است. قابل‌ذکر است که دستیابی به جواب بهینه و دقیق برای مسائل جریان کارگاهی با هدف حداقل کردن دیرکرد نهایی نیز امکان‌پذیر نیست. لذا برای حل مسئله تعریف‌شده در این مقاله که به دلیل داشتن موعدهای تحویل میانی از پیچیدگی بیشتری برخوردار است، از روش‌های فراابتکاری شبیه‌سازی تبرید، الگوریتم ژنتیک و ازدحام ذرات استفاده شد. جهت انجام آزمایش‌ها، ۹۶ نوع مسئله با ترکیب‌های مختلف تعداد ماشین (منبع)، تعداد کار (فعالیت) و میزان فشردگی موعد تحویل ایجاد شدند. این مسائل، مبنایی برای محاسبات مقدار مجموع دیرکرد قرار گرفتند. مقایسه الگوریتم‌ها بر اساس مقدار تابع هدف و مدت‌زمان حل مسائل صورت گرفت. نتایج نشان از برتری الگوریتم‌های شبیه‌سازی تبرید و ژنتیک از نظر مقدار دیرکرد دارد. از نظر زمان حل، تفاوت معناداری بین الگوریتم ژنتیک و شبیه‌سازی تبرید دیده شد؛ از این رو، در مجموع با توجه به یکسان بودن نتایج اثربخشی الگوریتم‌ها و وجود تفاوت چشم‌گیر در زمان حل مسائل بین الگوریتم ژنتیک و شبیه‌سازی تبرید، الگوریتم ژنتیک می‌تواند به‌عنوان روش برتر معرفی شود. این نتایج با تحقیقات قبلی همخوانی دارد [8;9]؛ اگرچه هیچ‌یک از آن‌ها با این تابع هدف حل نشده‌اند. به‌طورکلی، با بزرگ شدن ابعاد مسئله و افزایش فشردگی موعد تحویل، انتخاب روش مناسب اهمیت بیشتری پیدا می‌کند. شایان‌ذکر است که این مسئله را می‌توان برای مواردی که کارها به‌صورت سری انجام شده و هر مرحله دارای موعد است، در نظر گرفت. همچنین می‌توان این مفهوم را در مورد زنجیره‌های تأمین واجد ویژگی‌های فوق‌الذکر بسط داد. همچنین استفاده از روش‌های ابتکاری برای حل مسئله و مقایسه نتایج با الگوریتم‌های فراابتکاری می‌تواند در پژوهش‌های آتی مدنظر قرار گیرد.

تقدیر و تشکر

نویسنده بر خود لازم می‌داند که از یاری آقای احمد جعفریان قدردانی نماید.

۶- پی‌نوشت‌ها

1. Flow Shop
2. Makespan
3. Tardiness
4. Simulated Annealing
5. Genetic Algorithms
6. Particle Swarm Optimization
7. John Holland
8. Evolutionary Algorithms
9. Offspring
10. Reproduction Operator
11. Crossover Operator
12. Mutation Operator
13. Selection
14. Permutation
15. Swap
16. Particle Swarm Optimization
17. Eberhart
18. Kennedy
19. Particle
20. Swarm
21. gbest
22. Relative Percentage Deviation (RPD)
23. Tukey Test

۷- منابع

- [1] Deming L and Ping GX, "Variable neighborhood search for minimizing tardiness objectives on flow shop with batch processing machines," *Int J Prod Res.*, vol. 49, pp. 519-529, 2011.
- [2] Baker K.R and Schrage L.E, "Sequencing with earliness and tardiness penalties: a review," *Operations Research*, vol. 38, pp. 22-36, 1990.
- [3] Kanet J, "Minimizing the average deviation of job completion times about a common due date," *Naval Research Logistics Quart*, vol. 28, pp. 643-651, 1981.

- [4] Yeh W and Allahverdi A, "A branch and bound algorithm for the three machine flow shop scheduling problem with bi-criteria of make-span and total flow time," *International Transaction in Operations Research*, vol. 11(30), pp. 323-327, 2004.
- [5] Anghinolfi D and Paolucci M, "Parallel machine total tardiness Scheduling With a new hybrid met heuristic approach," *Computers & Operations Research*, vol. 34, pp. 3471-3490, 2007.
- [6] Sari cicek i and Celik C, "Two meta-heuristics for Parallel machine Scheduling with job splitting to minimize total tardiness," *Applied Mathematical Modelling*, vol. 35, pp. 4117-4126, 2011.
- [7] Demirel T, Ozkir V, Demirel NC and Tasdelen B, "A genetic algorithm approach for minimizing total tardiness in parallel machine scheduling problems.," *Proceedings of the World Congress on Engineering*, July 6-8 2011.
- [8] Vallada E, Ruiz R and Minella G, "Minimizing total tardiness in the m-machine flow shop problem: a review and evaluation of heuristics and meta heuristics," *Comput Oper Res*, vol. 35, pp. 1350-1373, 2008.
- [9] Karimi N and Davoudpour H, "A high Performing meta heuristic for multi-objective Flow shop Scheduling Problem," *Computers & Operations Research*, vol. 52, pp. 149-156, 2014.
- [10] Ghassemi Tari F and Olfat L, "Heuristic rules for tardiness Problem in Flow Shop With intermediate due. Int. J Adv manut Technology," *J Adv manut Technology*, vol. 71, pp. 381-393, 2014.
- [11] Brucker P., S. Knust and G. and Wang, "Complexity result for flow-shop problems with a single server," *European Journal of Operational Research*, vol. 165, pp. 398-407, 2005.
- [12] Elmi, A, Solimanpur, M and & Topaloglu, S, "A simulated annealing algorithm for the job shop cell scheduling problem with intercellular moves and reentrant parts," *Computers & Industrial Engineering*, 2011.

- [13] Gaafar, L and & Masoud, S, "Genetic algorithms and simulated annealing for scheduling in agile manufacturing.," *International Journal of Production Research*, vol. 43(14), pp. 3069-3085, 2005.
- [14] Rabiee, M, Zandieh, M and & Jafarian, A, "Scheduling of a no-wait two-machine flow shop with sequence-dependent setup times and probable rework using robust meta-heuristics," 2012.
- [15] Ghafoori S. and M. taghizadeh yazdi, "Proposing a Multi- Objective Mathematical Model for RCPSP and Solving it with Firefly and Simulated Annealing algorithms," *Journal of Modern Researches in Decision Making*, vol. vol 1 no4 (in Persian), pp.117-142,2017.
- [16] Lian, Z, Gu, X and & Jiao, B, "A novel particle swarm optimization algorithm for permutation flow-shop scheduling to minimize makespan," *Chaos, Solitons & Fractals*, vol. 35(5), pp. 851-861, 2008.
- [17] Tasgetiren, M.F, Liang, Y.-C, Sevkli, M and & Gency, "A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem," *European Journal of Operational Research*, vol. 177(3), p. 1930, 2007.
- [18] Chang, P.C, Chen, S.H and Fan, C.Y, "A hybrid electromagnetism-like algorithm for single machines scheduling problem," *Expert Syst. Appl*, vol. 36 (2), pp. 1259-1267, 2009.
- [19] Holland, J.H, "Adaptation in natural and artificial systems, University of Michigan press," *Ann Arbor, MI*, vol. 1(97), p. 5, 1975.
- [20] Yang, Y, Cui, Z and & Cheng, J, "An Improved Genetic Algorithm for Multiple-Depot Vehicle Routing Problem with Time Window," *Journal of Soochow University (Engineering Science Edition)*, vol. 26(2), pp. 20-23, 2006.
- [21] Chen, C.-L, Neppalli, R.V and & Aljaber, N, "Genetic algorithms applied to the continuous flow shop problem.," *Computers & Industrial Engineering*, vol. 30(4), pp. 919-929, 1996.

- [22] Gen M., F. Altiparmak and L & Lin, "A genetic algorithm for two- stage transportation problem using priority- based encoding or spectrum,". vol 28 (3),. pp 337-354, 2006.
- [23] Vahdani, B and & Zandieh, M, "Scheduling trucks in cross-docking systems: Robust meta-heuristics," *Computers & Industrial Engineering*, vol. 58(1), pp. 12-24, 2010.
- [24] Sivanandam, S and & Deepa, S, "Introduction to genetic algorithms: Springer Science & Business Media.," 2007.
- [25] Kirkpatrick, S, Gelatt, C.D and & Vecchi, M.P, "Optimization by simulated annealing," *science*, vol. 220(4598), p. 671, 1983.
- [26] Golden, B.L and & Skiscim, C.C, "Using simulated annealing to solve routing and location problems," *Naval Research Logistics Quarterly*, vol. 33(2), pp. 261-279, 1986.
- [27] Tarantilis, C and & Kiranoudis, C, "A meta-heuristic algorithm for the efficient distribution of perishable foods," *Journal of food engineering*, vol. 50(1), pp. 1-9, 2001.
- [28] Kennedy, J and & Eberhart, R.C, "Particle swarm optimization. In: P.o.I.C.o.N. Networks," *Perth, Australia, : IEEE.*, vol. 4, pp. 1942-1948, 1995.
- [29] Eberhart, R.C, Shi, Y and & Kennedy, J, "Swarm intelligence. San Mateo," *CA: Morgan Kaufmann division of Academic Press*, 2011.
- [30] Shi, Y and & Eberhart, R.C, "(1999). Empirical study of particle swarm optimization," *In: (Vol. 3): IEEE.*
- [31] Govindan, K, Jafarian, A, Khodaverdi, R and & Marid, "Two-echelon multiple-vehicle location-routing problem with time windows for optimization of sustainable supply chain network of perishable food," *International Journal of Production Economic*, 2013.