مهندسی صنایع و مدیریت شریف، (تابستان ۱۳۹۷)
دوره‌ی ۱-۳۴، شماره‌ی ۱/۱، ص. ۱۲۳-۱۳۸

Original Article

یک روش جدید مبتنی بر الگوریتم‌های فراابتکاری برای زمان‌بندی وسایل نقلیه در مسائل بارانداز تقاطعی

علی محتشمی (دانشیار)
گروه مدیریت صنعتی، واحد قزوین، دانشگاه آزاد اسلامی، قزوین، ایران

این مقاله یک روش جدید مبتنی بر الگوریتم‌های فراابتکاری برای زمان‌بندی وسایل نقلیه در مسائل بارانداز تقاطعی با درنظرگرفتن ذخیره‌ی موقت ارائه می‌کند. به‌عبارت دیگر، اگر وسیله‌ی نقلیه‌ی مورد نظر خروجی، در مقطعی از زمان برای بارگیری بعضی محصولات مورد نیاز، وجود نداشته باشد محصولات به‌طور موقت در یک فضای انبارش نگه‌داری می‌شوند تا زمانی‌که وسیله‌ی نقلیه‌ی مورد نظر به جایگاه خروجی وارد شود. روش پیشنهادی این مقاله در پی کمینه‌سازی زمان عملیات برای تخلیه و بارگیری کالاها در بارانداز تقاطعی است. برای این منظور در این مقاله فرض شده است که هردوی وسایل نقلیه‌ی ورودی و خروجی می‌توانند به‌طور مکرر به جایگاه‌های تخلیه و بارگیری وارد و خارج شوند. نتایج محاسباتی نشان می‌دهد که روش پیشنهادی این مقاله بهتر از روش معروف مورد مقایسه عمل می‌کند.

واژگان کلیدی: مدیریت زنجیره‌ی تأمین، بارانداز تقاطعی، الگوریتم‌های فراابتکاری، زمان‌بندی.

mohtashami@qiau.ac.ir

۱. مقدمه

بارانداز تقاطعی یک روش انبارداری است که شامل حرکت کالاها به‌طور مستقیم از جایگاه دریافت به جایگاه ارسال با حداقل زمان می‌شود. بارانداز تقاطعی می‌تواند به‌طور مؤثر باعث کاهش هزینه‌های حمل و نقل بدون افزایش موجودی‌ها شود. این کاهش هزینه‌ها در بارانداز تقاطعی هم‌زمان با مطلوب نگه‌داشتن سطح خدمت‌دهی به مشتریان صورت می‌گیرد.

بارانداز تقاطعی مانند یک پایگاه انتقال عمل می‌کند به‌طوری‌که محصولات را از یک تأمین‌کننده برای چند مقصد دریافت می‌کند، کالاها را با کالاهای تأمین‌کنندگان دیگر برای یک مقصد مشترک یکپارچه و ارسال می‌کند.[۱] درواقع، به‌علت یکپارچه‌کردن ارسال‌های مختلف به یک مقصد، بارانداز تقاطعی منجر به ارسال‌های کمتر با ظرفیت کامل وسایل حمل‌ونقل به‌ازای هر مقصد می‌شود. معمولاً، زمان نگه‌داری کالا در بارانداز تقاطعی کمتر از ۲۴ ساعت و بعضی مواقع کمتر از یک ساعت است.[۱و۲]

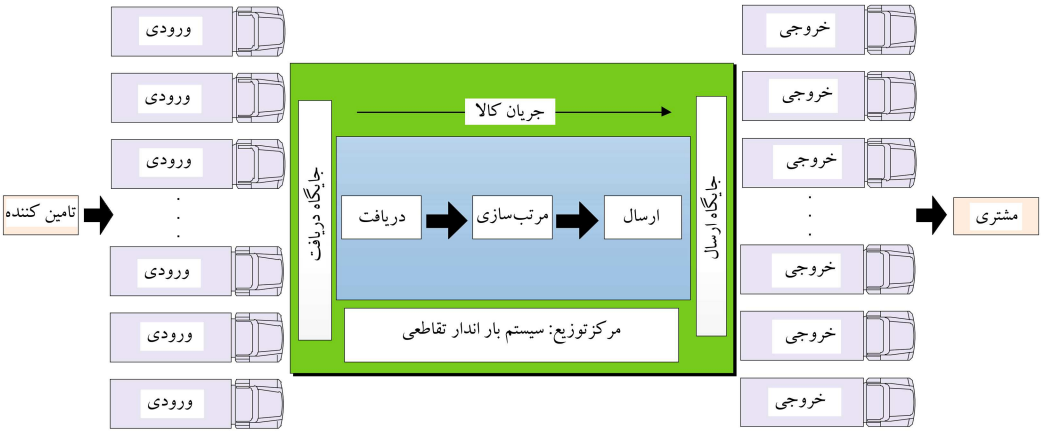
از یک دیدگاه کلی، در یک مرکز توزیع سنتی پنج مرحله عملیات بر روی محصولات انجام می‌شود که عبارت‌اند از: دریافت، ذخیره‌سازی، مرتب‌سازی، بازیابی و ارسال. تمام این مراحل هزینه‌هایی را به سیستم تحمیل می‌کند و هر رویکردی که منجر به کاهش هزینه‌ها در این عملیات شود، ارزشمند است. چهارچوب کلی سیستم‌های بارانداز تقاطعی به‌صورت شکل ۱ است. سیستم‌های بارانداز تقاطعی به‌صورت کلی ابتدا محصولات را از طریق وسایل نقلیه‌ی ورودی دریافت می‌کنند، سپس محصولات براساس مقصد مرتب می‌شوند و در نهایت کالاها از طریق وسایل نقلیه‌ی خروجی ارسال می‌شوند. یکی از پیامدهای استفاده از سیستم‌های بارانداز تقاطعی این است که دو مرحله‌ی انبارش و بازیابی براساس هم‌زمان‌کردن جریان کالا بین وسایل نقلیه‌ی ورودی و خروجی کاهش پیدا می‌کنند و حتی حذف می‌شوند.

۲. پژوهش‌های پیشین

محققان مختلفی تاکنون به مطالعه و تحقیق در زمینه‌ی سیستم‌های بارانداز تقاطعی پرداخته‌اند و سعی کرده‌اند با ارائه‌ی روش‌هایی به بهبود کارایی سیستم‌های بارانداز تقاطعی کمک کنند. برای این منظور تحقیقات مختلفی در حوزه‌های طراحی سیستم‌ها، مسیریابی وسایل نقلیه، زمان‌بندی وسایل نقلیه، برنامه‌ریزی توزیع و زمان‌بندی ارسال انجام شده است. از آنجایی‌که زمینه‌ی مقاله‌ی حاضر مربوط به زمان‌بندی وسایل نقلیه است، در ادامه به بررسی پژوهش‌ها در این حوزه پرداخته شده است.

محتشمی و همکارانش یک مدل ریاضی چندهدفه برای مسیریابی وسایل نقلیه در سیستم‌های بارانداز تقاطعی ارائه و از الگوریتم‌های فراابتکاری چندهدفه برای حل مسئله استفاده کردند.[۳] یو و اگبلو[۱] به‌منظور زمان‌بندی وسایل نقلیه در بارانداز تقاطعی با هدف کمینه‌سازی زمان عملیات به ارائه‌ی سه رویکرد حل پرداختند. آنها رویکردهای ارائه‌شده را با این فرض ارائه کردند که وسایل نقلیه‌ی ورودی و خروجی مجاز به ورودها و خروج‌های مکرر به جایگاه دریافت و ارسال برای تخلیه/بارگیری نیستند. به‌عبارت دیگر آنها فرض کردند که وقتی یک وسیله‌ی نقلیه وارد جایگاه

تاریخ: دریافت ۱۳۹۴/۴/۲۲، اصلاحیه ۱۳۹۵/۴/۲۶، پذیرش ۱۳۹۵/۵/۲۷.
DOI:10.24200/J65.2018.5541



شکل ۱. چهارچوب بارانداز تقاطعی.

دریافت یا ارسال می‌شود، در جایگاه توقف می‌کند تا زمانی‌که کل کالاهایش را تخلیه/بارگیری کند. در رویکرد اول ارائه‌شده توسط آنها، یک مدل ریاضی با هدف کمینه‌سازی زمان عملیات ارائه شد. در رویکرد دوم از شمارش کامل جواب‌ها برای تولید تمام جواب‌های ممکن استفاده شد. رویکرد شمارش کامل در مقاله‌ی ایشان برای ارائه‌ی مبنایی برای بررسی عملکرد دو روش دیگر استفاده شد.

آنها بیان کردند که به‌علت پیچیدگی محاسباتی، هردوی روش‌های ریاضی و رویکرد شمارش کامل برای حل مسائل مقیاس متوسط و بزرگ کارا نیستند. بنابراین، در رویکرد سوم از چند روش ابتکاری برای حل مسئله استفاده کردند.

وحدانی و زندیه[۴] مسئله‌ی بارانداز تقاطعی را با مفروضات یو و اگبلو[۱] درنظر گرفته‌اند و از پنج الگوریتم فرابتکاری برای زمان‌بندی وسایل نقلیه استفاده کردند. آنها نشان دادند که روش‌های فرابتکاری از روش‌های ابتکاری معرفی‌شده توسط یو و اگبلو[۱] بهتر عمل می‌کند. کونور و گولیاس[۵] به مسئله‌ی زمان‌بندی وسایل نقلیه در بارانداز تقاطعی پرداختند با این فرض که زمان ورود کامیون‌ها مشخص نیست و برای حل مسئله‌ی بارانداز تقاطعی روشی ارائه کردند که به کمینه‌سازی متوسط هزینه منجر می‌شود. مکنون و بابتیسته[۶] یک الگوریتم برنامه‌ریزی پویا توأم با الگوریتم تکاملی و رویکرد ابتکاری برای افزایش کارایی عملیات بارانداز تقاطعی ارائه کردند. بویسن و همکاران[۷] یک روش برنامه‌ریزی پویا برای بهینه‌سازی توالی عملیات کامیون‌های ورودی و خروجی در بارانداز تقاطعی ارائه کردند. سدیکوف[۸] رویکردی برای کاهش انبارش در بارانداز تقاطعی ارائه کرد. بویسن[۹] یک روش تبرید شبیه‌سازی‌شده و یک روش برنامه‌ریزی پویا برای برنامه‌ریزی وسایل نقلیه در سیستم‌های بارانداز تقاطعی با شرط مجاز نبودن انبار موقت ارائه کرد. این سیاست به این معنی است که پس از تخلیه‌ی کالا از کامیون ورودی، محصول باید به‌طور مستقیم درکامیون خروجی بارگیری شود. لی و همکارانش[۱۰] به تخصیص جایگاه‌ها به کامیون‌ها و توالی عملیات کامیون‌ها در سیستم‌های چند جایگاهی با استفاده از برنامه‌ریزی مختلط عدد صحیح پرداختند. جو و کیم[۱۱] یک الگوریتم ژنتیک و یک الگوریتم خودتکاملی برای مسئله‌ی زمان‌بندی کامیون‌ها با سه نوع کامیون پرداختند: کامیون‌های ورودی، کامیون‌های خروجی و کامیون‌های ترکیبی. کامیون‌های ترکیبی نقش هردوی کامیون‌های ورودی و خروجی را ایفا می‌کنند. لی و همکارانش[۱۲] مسئله‌ی بارانداز تقاطعی را به‌صورت یک مسئله‌ی دو مرحله‌یی ماشین‌های موازی با زمان‌های زودکرد و دیرکرد در نظرگرفتند. ایشان دو رویکرد مبتنی بر الگوریتم ژنتیک برای رسیدن به جواب ارائه کردند. لیاو و همکارانش[۱۳] به بررسی مسئله‌ی بارانداز تقاطعی با هدف کمینه‌سازی زمان عملیات پرداختند. آنها دو الگوریتم تکاملی برای حل مسئله ارائه کردند. چن و سانگ[۱۴] یک مسئله‌ی زمان‌بندی بارانداز تقاطعی ترکیبی را بررسی کردند که در آن یک معیار تقدم باید بین کارهای متوالی لحاظ شود. آنها برای مسائل کوچک برنامه‌ریزی عدد صحیح مختلط ارائه کردند و برای مسائل متوسط و بزرگ، چهار روش ابتکاری ارائه کردند. آلپان[۱۵] مسئله‌ی بارانداز تقاطعی را به‌صورت چند جایگاهی بررسی کرد و برای حل مسئله روشی با هدف کمینه‌سازی هزینه‌های کل ارائه کرد. برای این منظور او یک رویکرد برنامه‌ریزی پویا ارائه کرد. محتشمی[۱۶] یک روش جدید مبتنی بر الگوریتم‌های فرابتکاری برای زمان‌بندی کامیون‌ها در بارانداز متقاطع با درنظرگرفتن موجودی موقت ارائه کرد. محتشمی[۱۷] یک روش زمان‌بندی وسایل نقلیه با درنظرگرفتن الگوی ثابت تخلیه‌ی کامیون‌های ورودی و الگوی تکرارشونده‌ی بارگیری کامیون‌های خروجی ارائه کرد. محتشمی و فلاحیان[۱۸] یک روش جدید برای زمان‌بندی وسایل نقلیه در زنجیره‌ی تأمین با درنظرگرفتن بارانداز متقاطع ارائه کردند و برای حل مدل ارائه‌شده از الگوریتم‌های فرابتکاری استفاده کردند. لادیر و آلپان[۱۹] به بررسی مطالعات صورت گرفته در حوزه‌ی بارانداز متقاطع پرداخته‌اند و تحقیقات انجام‌شده در این زمینه را از منظر اجرا در سیستم‌های زنجیره‌ی تأمین بررسی و تحلیل کرده‌اند.

این مطالعات هر یک به ارائه‌ی روشی برای حل مسئله‌ی بارانداز تقاطعی با اهداف و مفروضات مختلف پرداخته‌اند. از آنجایی‌که یک هدف مهم در سیستم‌های بارانداز تقاطعی تحویل هرچه سریع‌تر کالاها به مشتریان است، بسیاری از تحقیقات سعی در آن داشته‌اند که با ارائه‌ی روش‌هایی به کاهش زمان عملیات در بارانداز تقاطعی بپردازند. زمان عملیات در بارانداز از لحظه‌ی تخلیه‌ی اولین کالا توسط اولین کامیون ورودی تا لحظه‌ی بارگیری آخرین کالا توسط آخرین کامیون خروجی را شامل می‌شود.

هدف این مقاله ارائه‌ی روشی است که منجر به کمینه‌سازی زمان عملیات در بارانداز تقاطعی شود. برای این منظور، در این مقاله برای تخلیه و بارگیری وسایل نقلیه از راهبرد ورود و خروج‌های تکرارشونده به جایگاه ورود و خروج استفاده می‌شود. به‌عبارت دیگر در استراتژی مذکور اجازه‌ی ورود و خروج‌های مکرر به کامیون‌های ورودی و خروجی به جایگاه تخلیه و بارگیری داده می‌شود. این بدان معناست که یک کامیون ورودی می‌تواند به جایگاه ورود وارد شود و بخشی از کالاهای خود را تخلیه کند و از جایگاه خارج شود و پس از مدتی مجدداً به جایگاه تخلیه وارد شود و بخشی دیگر از کالاها را تخلیه کند. این استراتژی برای کامیون‌های خروجی نیز به همین منوال است. یعنی یک کامیون خروجی می‌تواند به جایگاه خروج وارد شود و بخشی از کالاهای مورد نیاز خود را بارگیری کند و پس از طی مدتی مجدداً به جایگاه بارگیری

وارد شود و بخشی دیگر از کالاهای مورد نیاز خود را بارگیری کند. این استراتژی برخلاف برخی از تحقیقات پیشین است[۱،۴،۸و۱۱] که فرض می‌کنند کامیون‌های ورودی و خروجی تنها یک‌بار می‌توانند به جایگاه ورود و خروج وارد شوند و در این یک‌بار باید کل کالاهای خود را تخلیه/بارگیری کنند. در واقع استراتژی ورودها و خروج‌های مکرر یک استراتژی آزادتر نسبت به ورودها و خروج‌های مفرد است و می‌تواند منجر به کاهش زمان عملیات شود. یکی دیگر از وجوه تمایز این مطالعه نسبت به مطالعات پیشین، تخلیه و بارگیری یک نوع کالا به‌صورت بسته‌بندی است که دید واقع بینانه‌تری نسبت به سیستم‌های بارانداز تقاطعی واقعی ایجاد می‌کند. تحقیقاتی وجود دارند که مانند مقاله‌ی حاضر ورودی‌ها و خروجی‌های تکرارشونده را درنظرمی‌گیرند،[۲۰-۲۲] اما در این تحقیقات فرض می‌شود که یک کامیون ورودی/خروجی می‌تواند تعداد مورد نیاز از یک نوع کالا را (با هر مقداری) تخلیه/بارگیری کند که این امر تا حدی دور از واقعیت است. به بیان دیگر، در این تحقیقات فرض شده که مثلاً اگر یک کامیون ورودی شامل a واحد کالای نوع ۱ و b واحد کالای نوع ۲ باشد، این امکان وجود دارد که بار اول a_1 واحد ($a_1 < a$) از کالای نوع ۱ تخلیه شود، بار دوم b_1 واحد ($b_1 < b$) از کالای نوع ۲ تخلیه شود، بار سوم a_2 واحد دیگر ($a_2 < a$) از کالای نوع ۱ تخلیه شود و به همین صورت تا تمام کالاها تخلیه شوند. همچنان‌که مشاهده می‌شود با درنظرگرفتن ملاحظات مربوط به بسته‌بندی کالاها در واقعیت این نحوه‌ی تخلیه تا حدی و برخی مواقع عملی نیست. زیرا در بسیاری مواقع، کالاها در کامیون به‌صورت بسته‌بندی و درون پالت‌ها قرار داده می‌شوند و بازگشایی بسته‌بندی و جداسازی تعدادی از کالاها در داخل کامیون امکان‌پذیر نیست؛ اما تخلیه‌ی کامل یک پالت بسته‌بندی شده امکان‌پذیر است. مثال فوق یک مثال ساده برای بیان این مسئله است اما در مسائل واقعی معمولاً تنوع کالاها بسیار بیشتر از دو نوع است و این مسئله نیز بارزتر است. این توضیحات برای بارگیری کالاها در کامیون‌های خروجی نیز صادق است و بارگیری بخشی از یک نوع کالا تا حدی و برخی مواقع عملی نیست. بنابراین در این مقاله فرض می‌شود که کامیون‌های ورودی و خروجی می‌توانند به‌صورت مکرر به جایگاه‌های تخلیه و بارگیری وارد شوند و عملیات تخلیه و بارگیری را انجام دهند، اما در هر بار ورود به جایگاه باید بسته‌هایی با تعداد مشخصی از کالا را تخلیه و بارگیری کنند به‌طوری که تعداد کالاهای درون بسته می‌تواند از ۱ به بالا باشد. به‌عبارت دیگر ممکن است برای یک نوع کالا، تمام مقادیر کالا در یک بسته قرار داشته باشد که در این صورت باید تمام کالا تخلیه شود یا ممکن است کالا در بسته‌هایی با تعداد کمتر بسته‌بندی شده باشد که در این صورت تعدادی از بسته‌ها قابل تخلیه یا بارگیری است.

ادامه‌ی مقاله به این صورت سازمان‌دهی شده است: بخش ۳ مدل ریاضی و روش پیشنهادی برای حل مسئله را ارائه می‌کند، بخش ۴ مسائل عددی و نتایج محاسباتی را نشان می‌دهد و بخش ۵ به نتیجه‌گیری مقاله می‌پردازد.

۳. مدل ریاضی و روش پیشنهادی حل

سیستم‌های بارانداز تقاطعی با مفروضات مختلفی قابل بررسی و مطالعه هستند و بسته به تسهیلات، شرایط عملیات و سیاست‌ها می‌توان مدل‌های مختلفی را برای این سیستم‌ها در نظر گرفت. مشخصات و مفروضات سیستم بارانداز تقاطعی که در این مقاله به آن پرداخته شده است مطابق زیر است:

- در این مدل فرض شده است که یک فضای انبارش موقت درکنار جایگاه ارسال وجود دارد.
- کامیون‌های ورودی و خروجی می‌توانند برای انجام عملیات تخلیه و بارگیری بارها به جایگاه وارد و خارج شوند.
- تمام کامیون‌های ورودی و خروجی در زمان صفر موجود است.
- تمام کالاهای دریافت‌شده باید ارسال شوند.
- تخلیه یا بارگیری بخشی از یک نوع کالا با هر مقداری مجاز نیست و صرفاً تخلیه و بارگیری مقادیر مشخص (مطابق با مشخصات بسته‌بندی) مجاز است.
- فقط یک کالا می‌تواند در یک لحظه تخلیه/بارگیری شود و تخلیه و بارگیری موازی کالاهای مختلف مجاز نیست.
- زمان تعویض کامیون‌ها برای تمام کامیون‌های ورودی و خروجی یکسان است.
- یک جایگاه دریافت و ارسال در ترمینال وجود دارد و جایگاه‌ها مستقل از یکدیگرند.
- زمان تخلیه و بارگیری برای انواع کالاها برابر است و زمان لازم برای تخلیه یا بارگیری یک کالا، یک واحد زمانی است.
- زمان انتقال کالا از جایگاه دریافت به جایگاه ارسال معین و قطعی است.

۱.۳. مدل ریاضی پیشنهادی

این بخش به ارائه‌ی مدل ریاضی پیشنهادی برای حل مسئله‌ی بارانداز متقاطع با مفروضات ذکر شده می‌پردازد. پارامترها و متغیرهای به‌کار رفته در مدل به‌صورت زیر است.

۱.۱.۳. پارامترها

i: اندیس شماره‌ی وسایل نقلیه‌ی ورودی؛

j: اندیس شماره‌ی توالی تخلیه‌ی وسایل نقلیه‌ی ورودی؛

k: اندیس شماره‌ی نوع محصول در وسایل نقلیه‌ی ورودی؛

r: اندیس شماره‌ی وسایل نقلیه‌ی خروجی؛

s: اندیس شماره‌ی توالی بارگیری وسایل نقلیه‌ی خروجی؛

t: اندیس شماره‌ی نوع محصول در وسایل نقلیه‌ی خروجی؛

I: تعداد وسایل نقلیه‌ی ورودی؛

J: تعداد کل بسته‌ها در وسایل نقلیه‌ی ورودی؛

U: تعداد نوع محصول؛

R: تعداد وسایل نقلیه‌ی خروجی؛

S: تعداد کل بسته‌ها در وسایل نقلیه‌ی خروجی؛

α_{ik}: تعداد کالای نوع k در وسیله‌ی نقلیه‌ی ورودی نوع i که باید تخلیه شود؛

β_{rt}: تعداد کالای نوع t که باید در وسیله‌ی نقلیه‌ی خروجی نوع r بارگیری شود؛

D: زمان تعویض وسایل نقلیه؛

V: زمان انتقال کالاها از جایگاه دریافت به جایگاه ارسال؛

M: یک عدد بسیار بزرگ.

۲.۱.۳. متغیرهای پیوسته

T: زمان عملیات؛

s^x_{ijk}: زمان شروع تخلیه‌ی وسیله‌ی نقلیه‌ی ورودی i در توالی j برای تخلیه‌ی کالای k؛

f^x_{ijk}: زمان پایان تخلیه وسیله‌ی نقلیه‌ی ورودی i در توالی j برای تخلیه‌ی کالای k؛

s^y_{rst}: زمان شروع بارگیری وسیله‌ی نقلیه‌ی خروجی r در توالی s برای بارگیری کالای t؛

f^y_{rst}: زمان پایان بارگیری وسیله‌ی نقلیه‌ی خروجی r در توالی s برای بارگیری کالای t؛

γ_{ijrs}: تعداد کالایی که از وسیله‌ی نقلیه‌ی ورودی i در توالی j به وسیله‌ی نقلیه‌ی خروجی r در توالی s ارسال می‌شود.

۳.۱.۳. متغیرهای صفر ـ یک

x_{ijk}: در صورتی‌که وسیله‌ی نقلیه‌ی ورودی i در توالی j کالای k را تخلیه کند برابر ۱، در غیر این صورت برابر صفر است؛

y_{rst}: در صورتی‌که وسیله‌ی نقلیه‌ی خروجی r در توالی s کالای t را تخلیه کندبرابر ۱، در غیر این صورت برابر صفر است؛

$p_{jj'}$: در صورتی‌که وسیله‌ی نقلیه‌ی ورودی در توالی j به وسیله‌ی نقلیه‌ی ورودی در توالی j' تقدم داشته باشد برابر ۱، در غیر این صورت برابر صفر است؛

$q_{ss'}$: در صورتی‌که وسیله‌ی نقلیه‌ی خروجی در توالی s به وسیله‌ی نقلیه‌ی خروجی در توالی s' تقدم داشته باشد برابر ۱، در غیر این صورت برابر صفر است؛

v_{ijrs}: در صورتی‌که از وسیله‌ی نقلیه‌ی ورودی i در توالی j به وسیله‌ی نقلیه‌ی خروجی r در توالی s کالا فرستاده شود برابر ۱، در غیر این صورت برابر صفر است.

مدل ریاضی پیشنهادی به‌صورت زیر است.

$$\min : T \quad (1)$$

s.t.

$$T \geq f^y_{rst}, \quad r = 1,...,R;\ s = 1,...,S;\ t = 1,...,U \quad (2)$$

$$\sum_{i=1}^{I}\sum_{k=1}^{U} x_{ijk} = 1, \quad j = 1,...,J \quad (3)$$

$$\sum_{r=1}^{R}\sum_{t=1}^{U} y_{rst} = 1, \quad s = 1,...,S \quad (4)$$

$$\sum_{j=1}^{J} x_{ijk} = 1, \quad i = 1,...,I;\ k = 1,...,U \quad (5)$$

$$\sum_{s=1}^{S} y_{rst} = 1, \quad r = 1,...,R;\ t = 1,...,U \quad (6)$$

$$s^x_{i1k} = 0, \quad i = 1,...,I;\ k = 1,...,U \quad (7)$$

$$s^x_{ij'k} \geq f^x_{ljk} - M(1 - p_{jj'}), \quad i = 1,...,I;\ l = 1,...,I;\ j = 1,...,J;\ j' = 1,...,J;\ k = 1,...,U \text{ where } i = l \quad (8)$$

$$s^x_{ijk} \geq f^x_{lj'k} - M p_{jj'}, \quad i = 1,...,I;\ l = 1,...,I;\ j = 1,...,J;\ j' = 1,...,J;\ k = 1,...,U \text{ where } i = l \quad (9)$$

$$s^x_{ij'k} \geq f^x_{ljk} + D - M(1 - p_{jj'}), \quad i = 1,...,I;\ l = 1,...,I;\ j = 1,...,J;\ j' = 1,...,J;\ k = 1,...,U \text{ where } i \neq l \quad (10)$$

$$s^x_{ijk} \geq f^x_{lj'k} + D - M p_{jj'}, \quad i = 1,...,I;\ l = 1,...,I;\ j = 1,...,J;\ j' = 1,...,J;\ k = 1,...,U \text{ where } i \neq l \quad (11)$$

$$f^x_{ijk} \geq s^x_{ijk} + \alpha_{ik}, \quad i = 1,...,I;\ j = 1,...,J;\ k = 1,...,U \quad (12)$$

$$s^y_{r1t} = 0, \quad r = 1,...,R;\ t = 1,...,U \quad (13)$$

$$s^y_{rs't} \geq f^y_{lst} - M(1 - q_{ss'}), \quad r = 1,...,R;\ l = 1,...,R;\ s = 1,...,S;\ s' = 1,...,S;\ t = 1,...,U \text{ where } r = l \quad (14)$$

$$s^y_{rst} \geq f^y_{ls't} - M q_{ss'}, \quad r = 1,...,R;\ l = 1,...,R;\ s = 1,...,S;\ s' = 1,...,S;\ t = 1,...,U \text{ where } r = l \quad (15)$$

$$s^y_{rs't} \geq f^y_{lst} + D - M(1 - q_{ss'}), \quad r = 1,...,R;\ l = 1,...,R;\ s = 1,...,S;\ s' = 1,...,S;\ t = 1,...,U \text{ where } r \neq l \quad (16)$$

$$s^y_{rst} \geq f^y_{ls't} + D - M q_{ss'}, \quad r = 1,...,R;\ l = 1,...,R;\ s = 1,...,S;\ s' = 1,...,S;\ t = 1,...,U \text{ where } r \neq l \quad (17)$$

$$f^y_{rst} \geq s^x_{ijk} + v + \gamma_{ijrs} - M(1 - v_{ijrs}), \quad i = 1,...,I;\ j = 1,...,J;\ k = 1,...,U;\ r = 1,...,R;\ s = 1,...,S;\ t = 1,...,U \quad (18)$$

$$f^y_{rst} \geq s^y_{rst} + \sum_{t=1}^{U} \beta_{rt}, \quad r = 1,...,R;\ s = 1,...,S;\ t = 1,...,U \quad (19)$$

$$\gamma_{ijrs} \leq M v_{ijrs}, \quad i = 1,...,I;\ j = 1,...,J;\ r = 1,...,R;\ s = 1,...,S \quad (20)$$

$$\gamma_{ijrs} \leq \alpha_{ik}, \quad i = 1,...,I;\ j = 1,...,J;\ r = 1,...,R;\ s = 1,...,S,\ k = 1,...,U \quad (21)$$

$$\gamma_{ijrs} \leq \beta_{rt}, \quad i = 1,...,I;\ j = 1,...,J;\ r = 1,...,R;\ s = 1,...,S,\ t = 1,...,U \quad (22)$$

$$s^x_{ijk} \leq M x_{ijk}, \quad i = 1,...,I;\ j = 1,...,J;\ k = 1,...,U \quad (23)$$

$$f^x_{ijk} \leq M x_{ijk}, \quad i = 1,...,I;\ j = 1,...,J;\ k = 1,...,U \quad (24)$$

$$s^y_{rst} \leq M y_{rst}, \quad r = 1,...,R;\ s = 1,...,S;\ t = 1,...,U \quad (25)$$

$$f^y_{rst} \leq M y_{rst}, \quad r = 1,...,R;\ s = 1,...,S;\ t = 1,...,U \quad (26)$$

$$\text{all variables} \geq 0 \quad (27)$$

رابطه‌ی ۱ تابع هدف مسئله را نشان می‌دهد که زمان عملیات را کمینه می‌کند.

محدودیت ۲ تضمین می‌کند که زمان عملیات بزرگتر یا مساوی زمان پایان بارگیری تمام وسایل نقلیه است. محدودیت ۳ تضمین می‌کند که در توالی jام فقط یک وسیله‌ی نقلیه برای تخلیه وجود داشته باشد. محدودیت ۴ تضمین می‌کند که در توالی sام فقط یک وسیله‌ی نقلیه برای بارگیری وجود داشته باشد. محدودیت ۵ تضمین می‌کند وسیله‌ی نقلیه‌ی i کالای نوع k را فقط در یک توالی تخلیه کند. محدودیت ۶ تضمین می‌کند وسیله‌ی نقلیه‌ی r کالای نوع t را فقط در یک توالی بارگیری کند. محدودیت ۷ زمان شروع وسیله‌ی نقلیه‌یی که اولین تخلیه را انجام می‌دهد برابر صفر قرار می‌دهد. محدودیت ۸ تضمین می‌کند که زمان شروع تخلیه‌ی یک وسیله‌ی نقلیه در توالی j' (توالی بعدی) بزرگتر یا مساوی زمان شروع تخلیه‌ی وسیله‌ی نقلیه در توالی j (توالی قبلی) است اگر نوع وسیله‌ی نقلیه در هر دو توالی یکسان باشد ($i = l$). چنانچه وسیله‌ی نقلیه در توالی j به وسیله‌ی نقلیه در توالی j' تقدم داشته باشد، چون $p_{jj'}$ برابر ۱ می‌شود، محدودیت ۸ تبدیل به یک محدودیت مؤثر و محدودیت ۹ تبدیل به یک محدودیت زائد می‌شود و در صورتی‌که وسیله‌ی نقلیه در توالی j' به وسیله‌ی نقلیه در توالی j تقدم داشته باشد، محدودیت ۸ تبدیل به یک محدودیت زائد و محدودیت ۹ تبدیل به یک محدودیت مؤثر می‌شود. اگر وسیله‌ی نقلیه در دو توالی پی‌درپی یکسان نباشد ($i \neq l$)، محدویت‌های ۱۰ و ۱۱ تضمین می‌کنند که زمان شروع تخلیه در توالی بعدی بزرگتر یا مساوی زمان پایان توالی قبلی به‌علاوه‌ی زمان جابه‌جایی وسایل نقلیه است (وقتی وسایل نقلیه در دو توالی پی‌درپی یکسان نباشند، به این معنی است که وسایل نقلیه جابه‌جا شده‌اند). محدودیت ۱۲ تضمین می‌کند که زمان پایان تخلیه‌ی یک وسیله‌ی نقلیه بزرگتر یا مساوی زمان شروع تخلیه به‌علاوه‌ی α_{ik} (تعداد کالای نوع k در وسیله‌ی نقلیه ورودی نوع i که باید تخلیه شود) است. محدودیت ۱۳ زمان شروع بارگیری اولین وسیله‌ی نقلیه در توالی وسایل نقلیه‌ی خروجی را برابر صفر قرار می‌دهد. محدودیت ۱۴ تضمین می‌کند که زمان شروع بارگیری یک وسیله‌ی نقلیه در توالی s' (توالی بعدی) بزرگتر یا مساوی زمان شروع بارگیری وسیله‌ی نقلیه در توالی s (توالی قبلی) است اگر نوع وسیله‌ی نقلیه در هر دو توالی یکسان باشد ($r = l$). چنانچه وسیله‌ی نقلیه در توالی s به وسیله‌ی نقلیه در توالی s' تقدم داشته باشد، چون $q_{ss'}$ برابر ۱ می‌شود، محدودیت ۱۴ تبدیل به یک محدودیت مؤثر و محدودیت ۱۵ تبدیل به یک محدودیت زائد می‌شود و در صورتی‌که وسیله‌ی نقلیه در توالی s' به وسیله‌ی نقلیه در توالی s تقدم داشته باشد، محدودیت ۱۴ تبدیل به یک محدودیت زائد و محدودیت ۱۵ تبدیل به یک محدودیت مؤثر می‌شود. اگر وسیله‌ی نقلیه در دو توالی پی‌درپی یکسان نباشد ($r \neq l$)، محدویت‌های ۱۶ و ۱۷ تضمین می‌کنند که زمان شروع بارگیری در توالی بعدی بزرگتر یا مساوی زمان پایان توالی قبلی به‌علاوه‌ی زمان جابه‌جایی وسایل نقلیه است (وقتی وسایل نقلیه در دو توالی پی‌درپی یکسان نباشد، به این معنی است که وسایل نقلیه جابه‌جا شده‌اند). محدودیت ۱۸ تضمین می‌کند که زمان پایان بارگیری وسیله‌ی نقلیه‌ی r در توالی s بزرگتر یا مساوی زمان شروع تخلیه‌ی وسیله‌ی نقلیه‌ی i در توالی j به‌علاوه‌ی زمان انتقال کالا از جایگاه دریافت به ارسال (v) به علاوه γ_{ijrs} (تعداد کالایی که از وسیله‌ی نقلیه‌ی ورودی i در توالی j به وسیله‌ی نقلیه‌ی خروجی r در توالی s ارسال می‌شود) است. طبق مفروضات بخش ۳، زمان تخلیه و بارگیری برای تمام انواع کالاها برابر است و زمان لازم برای تخلیه یا بارگیری یک کالا، یک واحد زمانی است. محدودیت ۱۹ تضمین می‌کند که زمان پایان بارگیری وسیله‌ی نقلیه‌ی r در توالی s بزرگتر یا مساوی زمان شروع بارگیری وسیله‌ی نقلیه‌ی r در توالی s به‌علاوه‌ی مجموع تعداد کل کالاهایی است که باید بارگیری کند. محدودیت ۲۰ تضمین می‌کند که اگر کالایی از وسیله‌ی نقلیه‌ی ورودی i در توالی j به وسیله‌ی نقلیه‌ی r در توالی s ارسال شود (یعنی مقدار v_{ijrs} برابر ۱ شود)، متغیر γ_{ijrs} بتواند مقدار بگیرد و در غیر این صورت متغیر γ_{ijrs} برابر صفر خواهد بود. محدودیت ۲۱ تضمین می‌کند که مقدار کالای ارسال‌شده از وسیله‌ی نقلیه‌ی ورودی i در توالی j به وسیله‌ی نقلیه‌ی خروجی r در توالی s کوچکتر یا مساوی α_{ik} (تعداد کالای نوع k در وسیله‌ی نقلیه‌ی ورودی نوع i که باید تخلیه شود) باشد. محدودیت ۲۲ تضمین می‌کند که مقدار کالای ارسال‌شده از وسیله‌ی نقلیه‌ی ورودی i در توالی j به وسیله‌ی نقلیه‌ی خروجی r در توالی s کوچکتر یا مساوی β_{rt} (تعداد کالای نوع t که باید در وسیله‌ی نقلیه‌ی خروجی نوع r بارگیری شود) باشد. محدودیت‌های ۲۳ و ۲۴ تضمین می‌کنند که چنانچه وسیله‌ی نقلیه‌ی ورودی i برای تخلیه‌ی کالای k در توالی j وجود نداشته باشد ($x_{ijk} = 0$)، زمان شروع و پایان تخلیه برای این متغیر صفر در نظرگرفته شود، و در غیر این صورت مقداری بزرگتر یا مساوی صفر. محدودیت‌های ۲۵ و ۲۶ تضمین می‌کنند که چنانچه وسیله‌ی نقلیه‌ی خروجی r برای بارگیری کالای t در توالی s وجود نداشته باشد ($y_{rst} = 0$)، زمان شروع و پایان بارگیری برای این متغیر صفر درنظرگرفته شود، و در غیراین‌صورت مقداری بزرگتر یا مساوی صفر. محدودیت ۲۷ تمام متغیرها را بزرگتر یا مساوی صفر در نظر می‌گیرد.

۲.۳. روش حل پیشنهادی

روش پیشنهادی مبتنی بر الگوریتم ژنتیک پویاست. الگوریتم ژنتیک یکی از الگوریتم‌های تکاملی است که از مشخصه‌هایی مانند جمعیت، کروموزوم، ژن، تولید مجدد و برخی عملگرها برای انجام عملیات جست‌وجو استفاده می‌کند. ابتدا یک جمعیت اولیه شامل کروموزوم‌ها و ژن‌ها به‌صورت تصادفی تولید می‌شود، و سپس برای دست‌یابی به جمعیت‌های بعدی (نسل فرزندان)، عملیات مختلفی نظیر تقاطع و جهش بر روی جمعیت اولیه انجام می‌شود. این مراحل در هر تکرار (نسل) انجام می‌شود تا بهترین جواب ممکن براساس معیارهای توقف حاصل شود.

در الگوریتم ژنتیک سنتی عملگرهای تقاطع و جهش به‌صورت ایستا عمل می‌کنند، بدین معنی که احتمال تقاطع و جهش در طول تکرارها ثابت باقی می‌ماند، اما الگوریتم ژنتیک پیشنهادی این مقاله احتمال تقاطع و جهش را در طول تکرارها کاهش می‌دهد.

این امر به الگوریتم این امکان را می‌دهد که در تکرارهای اولیه فضای گسترده‌یی را جست‌وجو کند و همچنان که الگوریتم در نسل‌ها پیش می‌رود و به جواب بهینه نزدیکتر می‌شود، با کاهش احتمال تقاطع و جهش فضای جست‌وجو محدودتر شود.

از آنجایی‌که روش پیشنهادی این مقاله بر اساس الگوریتم ژنتیک است، بخش ۱.۲.۳. به ارائه‌ی ساختار جواب، که در الگوریتم ژنتیک به نام کروموزوم مطرح است، می‌پردازد.

۱.۲.۳. ساختار کروموزوم

ساختار جواب‌ها در روش پیشنهادی، یک ساختار سه‌سطحی مطابق شکل ۲ است. سطح اول نشان‌دهنده‌ی شماره‌ی کامیون است، سطح دوم نشان‌دهنده‌ی شماره‌ی محصول (نوع محصول) و سطح سوم نشان‌دهنده‌ی تعداد کالایی است که باید تخلیه/بارگیری شود. همچنان‌که مشاهده می‌شود کروموزم پیشنهادی از دو بخش تشکیل شده است: بخش اول شامل اطلاعات کامیون‌های ورودی و بخش دوم شامل اطلاعات کامیون‌های خروجی است.

همچنان‌که مشاهده می‌شود، در این کروموزم کامیون‌های ورودی قابلیت ورود و خروج‌های چندباره را به جایگاه دریافت و ارسال دارند. در این شکل سه کامیون ورودی و سه کامیون خروجی برای تخلیه‌و بارگیری ۴ نوع کالا مجموعاً به تعداد ۱۸۴

کامیون#	۱	۳	۲	۲	۱	۲	۱	۲	۳	۳	۱	۱	۱	۳	۲	۱	۲	۱	۱	۲	۲	۲	۳	۳	۳	۲	۱	۱	۳	۱	۲	۲
محصول#	۲	۴	۱	۳	۴	۱	۲	۱	۳	۳	۲	۱	۱	۳	۱	۲	۱	۴	۴	۴	۳	۳	۲	۱	۴	۴	۱	۲	۲	۱	۲	۳
تعداد#	۱۵	۲۰	۱۰	۲۲	۲۹	۱۵	۲۳	۱۹	۳۱	۱۰	۱۲	۲۵	۱۸	۲۰	۱۵	۱۴	۲۰	۱۰	۱۶	۱۴	۱۰	۱۲	۲۵	۱۸	۲۰	۱۰	۱۶	۱۴	۱۰	۲۰	۱۴	۱۰

کامیون های خروجی — کامیون های ورودی

شکل ۲. ساختار جواب زمان‌بندی تخلیه/بارگیری کامیون‌ها.

واحد کالا وجود دارد. به‌طور جزئی‌تر، توالی عملیات کامیون‌های ورودی نشان می‌دهد که اولین کامیون برای تخلیه‌ی کامیون شماره‌ی ۲ بوده که محصول ۳ را به میزان ۱۰ واحد تخلیه می‌کند، سپس این کامیون محصول ۱ و ۲ را به‌ترتیب به میزان ۱۴ و ۱۶ واحد تخلیه می‌کند. در موقعیت چهارم، پنجم و ششم کروموزوم (از چپ به راست)، کامیون ۳ قرار دارد که به‌ترتیب کالاهای ۲، ۱ و ۴ را به میزان‌های به‌ترتیب ۱۰، ۲۰ و ۱۴ تخلیه می‌کند. در موقعیت هفتم، کامیون ۲ مجدداً به جایگاه تخلیه وارد می‌شود و کالای ۴ را به میزان ۱۵ واحد تخلیه می‌کند. این روند به‌همین صورت ادامه پیدا می‌کند تا در موقعیت دوازدهم، کامیون ورودی ۱ کالای شماره ۱ را به میزان ۱۰ واحد تخلیه می‌کند. روند توضیح داده شده برای کامیون‌های خروجی نیز دقیقاً به‌همین منوال است. یعنی اولین کامیونی که عملیات بارگیری را آغاز می‌کند کامیون شماره ۳ است که کالای شماره‌ی ۳ را به میزان ۳۱ واحد بارگیری می‌کند. سپس کامیون ۳ کالای شماره ۱ را به میزان ۱۹ واحد بارگیری می‌کند و از جایگاه خارج می‌شود تا کامیون ۲ کالای شماره ۲ را به میزان ۲۳ واحد بارگیری نماید. همچنان که مشاهده می‌شود، کامیون شماره ۳ در موقعیت بیستم کروموزوم مجدداً به جایگاه باز می‌گردد تا کالای شماره ۴ را به میزان ۲۰ واحد بارگیری کند. برای سادگی مطلب، در این مثال فرض شده است که هر نوع کالا برای کامیون‌های ورودی و خروجی در یک بسته‌بندی (پالت) قرار دارد.

۲.۲.۳. الگوریتم پیشنهادی تولید جمعیت اولیه

اولین گام در پیاده‌سازی الگوریتم ژنتیک، تولید نسل (جمعیت) اولیه است که در الگوریتم ژنتیک سنتی به‌صورت تصادفی انجام می‌شود، اما در این بخش یک الگوریتم ابتکاری تصادفی برای تولید جمعیت اولیه ارائه می‌شود. با الگوریتم ابتکاری تصادفی سرعت رسیدن به جواب بهینه/نزدیک بهینه بیشتر می‌شود و همچنین کیفیت جواب نهایی (به لحاظ برازندگی) بالاتر خواهد بود. چنانچه برای تولید کروموزوم‌ها از روش کاملاً تصادفی استفاده کنیم، باید در هر ژن به‌صورت تصادفی کامیون‌ها و محصولات وارد شوند، اما این امر تا حدی از سرعت و کیفیت جواب‌ها می‌کاهد. به‌همین دلیل در شکل ۳ یک الگوریتم ابتکاری تصادفی برای تولید جمعیت اولیه ارائه شده است.

خط ۱ و ۲۰ حلقه‌ی اصلی الگوریتم است که باعث می‌شود الگوریتم برای تمام اندازه‌ی جمعیت (popsize) اجرا شود. خط ۳ یک کامیون ورودی را به تصادف انتخاب می‌کند و خط ۴ این کامیون انتخاب شده را از فهرست کامیون‌های ورودی حذف می‌کند. خط ۶ یک نوع از محصولات را انتخاب می‌کند که تعداد این محصول در کامیون انتخاب‌شده بیشتر از صفر باشد. خط ۷ محصول انتخاب‌شده را از فهرست محتویات کامیون انتخاب‌شده حذف می‌کند. خط ۸ کامیون انتخاب شده، محصول انتخاب شده، و تعداد محصول انتخاب‌شده در کامیون انتخاب‌شده را در ژن اول کروموزوم قرار می‌دهد و باعث می‌شود یکی از کامیون‌ها یکی از محصولات را تخلیه

```
1   for i=1 to popsize
2    do
3     choose a random inbound truck
4     update the list of inbound trucks
5     do
6      choose a random valid product type
7      update the contents of chosen inbound truck
8      gene ← chosen inbound truck, chosen product type, related quantity
9     loop until (all valid product types are allocated)
10   loop until (all inbound trucks are allocated)
11   do
12    choose a random outbound truck
13    update the list of outbound trucks
14    do
15     choose a random valid product type
16     update the contents of chosen outbound truck
17     gene ← chosen outbound truck, chosen product type, related quantity
18    loop until (all valid product types are allocated)
19   loop until (all outbound trucks are allocated)
20  next i
```

شکل ۳. الگوریتم پیشنهادی تولید جمعیت اولیه.

کند. حلقه‌ی ۹-۵ تا زمانی تکرار می‌شود که کل محصولات کامیون ورودی انتخاب شده تخلیه شوند. حلقه‌ی ۱۰-۲ تا زمانی تکرار می‌شود که کل کامیون‌های ورودی برای تخلیه انتخاب شوند. دلیل حذف کامیون انتخاب‌شده از فهرست کامیون‌های ورودی در خط ۴ این است که در دفعات بعدی تکرار حلقه، کامیون‌هایی که قبلاً انتخاب شده بودند و به ژن‌ها تخصیص یافته‌اند، مجدداً انتخاب نشوند. خط ۷ الگوریتم نیز شرایط مشابه خط ۴ را اعمال می‌کند. این خط باعث می‌شود که محصول انتخاب‌شده از فهرست محتویات کامیون انتخاب‌شده حذف شود، تا محصول انتخاب‌شده برای یک کامیون، مجدداً در کروموزوم قرار نگیرد.

مراحل فوق برای کامیون‌های خروجی نیز به همین صورت و مطابق با حلقه‌ی ۱۹-۱۱ است. یعنی هر بار یک کامیون خروجی به تصادف انتخاب و یک نوع از محصولات آن نیز به تصادف انتخاب می‌شود و کامیون انتخاب‌شده، محصول انتخاب‌شده را به تعداد مشخص بارگیری می‌کند.

همچنان که مشاهده می‌شود، الگوریتم پیشنهادی تولید جمعیت اولیه یک الگوریتم ابتکاری است و کاملاً تصادفی نیست به‌طوری‌که این امر به بهبود سرعت و کیفیت جواب‌های الگوریتم کمک می‌کند. درصورتی‌که بخواهیم الگوریتم تولید جمعیت اولیه کاملاً تصادفی باشد، باید در هر ژن یک کامیون تصادفی و یک محصول تصادفی قرار دهیم، اما این امر باعث می‌شود تعداد جابه‌جایی کامیون‌ها در جایگاه افزایش یابد زیرا ممکن است یک کامیون انتخاب شود و یک نوع محصول را تخلیه کند و سپس کامیون دیگری وارد جایگاه شود و محصول دیگری را تخلیه کند و به همین صورت تا کل کامیون‌ها بارهای خود را تخلیه کنند. الگوریتم پیشنهادی این بخش تولید جواب‌ها را به‌صورت کاملاً تصادفی انجام نمی‌دهد؛ زیرا هر بار یک کامیون ورودی را به‌صورت تصادفی انتخاب می‌کند و محصولات

کامیون را به‌صورت تصادفی تخلیه می‌کند، اما کامیون انتخاب‌شده عوض نمی‌شود تا زمانی‌که کل محصولات کامیون تخلیه شود. این امر باعث می‌شود جابه‌جایی کامیون‌ها در جواب‌های اولیه در حداقل تعداد ممکن باشد. اما از آنجایی‌که این امکان وجود دارد که با جابه‌جایی توالی تخلیه یا بارگیری، جواب‌های بهتری حاصل شود، جواب‌های تولیدشده‌ی اولیه در مراحل بعدی توسط عملگرهای تقاطع و جهش بهبود می‌یابند.

۳.۲.۳. الگوریتم پیشنهادی محاسبه‌ی برازندگی جواب‌ها

در این بخش، جواب‌های تولیدشده‌ی اولیه ارزیابی می‌شوند تا میزان برازندگی هر یک مشخص شود. شاخص برازندگی جواب‌ها در این الگوریتم زمان عملیات صرف‌شده در پایانه است. زمان عملیات از لحظه‌ی تخلیه‌ی اولین محصول توسط اولین کامیون ورودی شروع می‌شود و تا لحظه‌ی بارگیری آخرین محصول توسط آخرین کامیون خروجی ادامه می‌یابد. شبه کد محاسبه‌ی برازندگی جواب‌ها مطابق شکل ۴ است.

حلقه‌ی اصلی الگوریتم یعنی حلقه‌ی ۳۸-۱ تضمین می‌کند که برازندگی برای تمام کروموزوم‌ها محاسبه شود. حلقه‌ی ۱۳-۲ زمان شروع و پایان عملیات تخلیه برای کامیون‌های ورودی را محاسبه می‌کند. همچنان‌که مشاهده می‌شود در این الگوریتم i برای شماره‌ی جواب‌ها (کروموزوم‌ها) و j برای شماره‌ی ژن‌ها استفاده شده است. خط ۲ تضمین می‌کند که زمان تخلیه برای تمام ژن‌ها از ژن شماره ۱ تا ژن شماره IL محاسبه شود به‌طوری‌که IL نشان‌دهنده‌ی طول کروموزوم برای کامیون‌های ورودی است. مثلاً IL برای کروموزوم شکل ۲ برابر ۱۲ است. زمان شروع تخلیه برای اولین کامیون در اولین ژن برابر صفر در نظر گرفته شده است که این امر توسط خط ۳ و ۴ انجام می‌شود به‌طوری‌که $st(i,j)$ نشان‌دهنده‌ی زمان شروع تخلیه‌ی کامیون ورودی در ژن j کروموزوم i است. زمان پایان تخلیه‌ی کامیون ورودی در ژن اول کروموزوم i که با $ft(i,j)$ نشان داده می‌شود توسط خط ۱۲ انجام می‌شود و برابر است با زمان شروع تخلیه به‌علاوه $Q(i,j)$ به‌طوری‌که $Q(i,j)$ نشان‌دهنده‌ی تعداد کالایی است که کامیون ورودی کروموزوم i از طریق ژن j تخلیه می‌کند.

در صورتی‌که ژن تحت محاسبه اولین ژن توالی نباشد، زمان شروع تخلیه برابر صفر نخواهد بود که این امر توسط خطوط ۵ الی ۱۱ بررسی و زمان شروع تخلیه محاسبه می‌شود. خط ۵ تأیید می‌کند که ژن مورد بررسی ژن اول در توالی نیست. خط ۶ بررسی می‌کند که آیا کامیون ورودی تحت بررسی (CSI) در ژن j مطابق کامیون برنامه‌ریزی شده در ژن قبلی (PSI) هست یا خیر. در صورت مثبت بودن این عبارت، زمان شروع تخلیه‌ی کامیون فعلی (کامیون تحت بررسی در ژن j) برابر زمان پایان تخلیه‌ی کامیون برنامه‌ریزی‌شده در ژن قبلی است به‌طوری‌که $ft(i,j-1)$ نشان‌دهنده زمان پایان تخلیه‌ی کامیون ورودی در ژن $j-1$ کروموزوم i است. در واقع این توضیحات نشان می‌دهد چنانچه یک کامیون ورودی کالاهای مختلفی را در دو ژن متوالی توالی تخلیه کند، زمان شروع تخلیه‌ی ژن بعدی برابر زمان پایان تخلیه ژن قبلی است؛ زیرا در این حالت کامیون عوض نشده است و صرفاً کالای مورد تخلیه عوض شده است. برای مثال سه ژن اول شکل ۲ این حالت را نشان می‌دهد که در هر سه‌ژن کامیون برنامه‌ریزی شده، کامیون ۲ است که کالاهای مختلفی را تخلیه می‌کند. بنابراین برای مثال زمان شروع تخلیه در ژن دوم برابر زمان پایان تخلیه در ژن اول است. در صورتی‌که CSI مخالف PSI باشد که خط ۸ آن را تأیید می‌کند، زمان شروع تخلیه در ژن j برابر زمان پایان تخلیه در ژن $j-1$ به‌علاوه changeover است به‌طوری‌که changeover نشان‌دهنده‌ی زمان تعویض کامیون است و این امر توسط خط ۹ انجام می‌شود. برای مثال این حالت در ژن ۴ شکل ۲ قابل مشاهده

```
1   for i=1 to popsize
2    for j=1 to IL
3     if j=1 then
4      st (i,j) ← 0
5     else
6      if CSI=PSI then
7       st (i,j) ← ft (i,j-1)
8      else
9       st (i,j) ← ft (i,j-1) + changeover
10     end if
11    end if
12    ft (i,j) ← st (i,j) + Q (i,j)
13   next j
14   for j=IL+1 to IL+OL
15    if j=IL+1 then
16     st (i,j) ← 0
17    else
18     if CSI=PSI then
19      st (i,j) ← ft (i,j-1)
20     else
21      st (i,j) ← ft (i,j-1) + changeover
22     end if
23    end if
24    θ1 ← st (i,j) + Q (i,j)
25    counter ← 1
26    for l=1 to IL
27     if pt (i,l) = pt (i,j) then
28      if Q (i,l)>0 and Q (i,j)>0 then
29       θ2^counter ← st (i,l) + min{Q (i,l), Q (i,j)} + v
30       update Q (i,l) and Q (i,j)
31       counter ← counter + 1
32      end if
33     end if
34    next l
35    θ2 ← max {θ2^counter}
36    ft (i,j) ← max {θ1, θ2}
37   next j
38  next i
```

شکل ۴. الگوریتم پیشنهادی محاسبه‌ی برازندگی جواب‌ها.

است که کامیون ۲ از جایگاه بیرون می‌رود و کامیون ۳ وارد جایگاه می‌شود. بنابراین زمان شروع تخلیه در ژن ۴ برابر زمان پایان تخلیه در ژن ۳ به‌علاوه‌ی زمان تعویض کامیون است. در صورتی که CSI مخالف PSI باشد، زمان پایان تخلیه دقیقاً مانند حالتی محاسبه می‌شود که CSI برابر PSI باشد است؛ زیرا زمان پایان تخلیه صرفاً به زمان شروع تخلیه بستگی دارد.

حلقه ۳۷-۱۴ زمان شروع و پایان عملیات بارگیری برای کامیون‌های خروجی را محاسبه می‌کند. خط ۱۴ تضمین می‌کند که زمان بارگیری برای تمام ژن‌ها از ژن شماره $IL+1$ تا ژن شماره $IL+OL$ محاسبه شود به‌طوری‌که OL نشان‌دهنده‌ی طول کروموزوم برای کامیون‌های خروجی است و $IL+OL$ نشان‌دهنده‌ی طول کل کروموزوم است. مثلاً OL برای کروموزوم شکل ۲ برابر ۹ است. زمان شروع بارگیری

برای اولین کامیون خروجی در اولین ژن توالی کامیون‌های خروجی برابر صفر در نظر گرفته شده است که این امر توسط خطوط ۱۵ و ۱۶ انجام می‌شود.

در صورتی‌که ژن تحت محاسبه‌ی اولین ژن توالی کامیون‌های خروجی نباشد، زمان شروع بارگیری برابر صفر نخواهد بود که این امر توسط خطوط ۱۷ الی ۲۳ بررسی و زمان شروع بارگیری محاسبه می‌شود. خط ۱۷ تأیید می‌کند که ژن مورد بررسی ژن اول در توالی کامیون‌های خروجی نیست. خط ۱۸ بررسی می‌کند که آیا کامیون خروجی تحت بررسی (CSI) در ژن j مطابق کامیون برنامه‌ریزی‌شده در ژن قبلی (PSI) هست یا خیر. در صورت مثبت بودن این عبارت، زمان شروع بارگیری کامیون فعلی (کامیون تحت بررسی در ژن j) برابر زمان پایان بارگیری کامیون برنامه‌ریزی‌شده در ژن قبلی است به‌طوری‌که $ft(i,j-1)$ نشان‌دهنده‌ی زمان پایان بارگیری کامیون خروجی در ژن $j-1$ کروموزوم i است. در واقع توضیحات فوق نشان می‌دهد چنانچه یک کامیون خروجی کالاهای مختلفی را در دو ژن متوالی توالی بارگیری کند، زمان شروع بارگیری در ژن بعدی برابر زمان پایان بارگیری در ژن قبلی است؛ زیرا در این حالت کامیون عوض نشده است و صرفاً کالای مورد بارگیری عوض شده است. در صورتی که CSI مخالف PSI باشد که خط ۲۰ آن را تأیید می‌کند، زمان شروع بارگیری در ژن j برابر زمان پایان بارگیری در ژن $j-1$ به علاوه changeover است به‌طوری‌که changeover نشان‌دهنده‌ی زمان تعویض کامیون است و این امر توسط خط ۲۱ انجام می‌شود.

محاسبه‌ی زمان پایان بارگیری در توالی کامیون‌های خروجی با کامیون‌های ورودی که پیشتر توضیح داده شد تفاوت دارد و با استفاده از خطوط ۲۴ الی ۳۶ انجام می‌شود. خط ۲۴ حاصل جمع زمان شروع بارگیری کامیون را با تعداد کالایی که کامیون در ژن j بارگیری می‌کند محاسبه می‌کند و در θ_1 قرار می‌دهد. در واقع θ_1 گزینه‌ی اول برای زمان پایان بارگیری کامیون است و به این معنی است که کالاهای مورد نیاز کامیون در انبار موقت موجود است و به‌محض ورود کامیون به جایگاه ارسال، عملیات بارگیری انجام می‌شود. بنابراین، گزینه‌ی اول برای زمان پایان بارگیری کامیون در ژن j برابر $st(i,j)$ به علاوه‌ی $Q(i,j)$ است. برای مثال θ_1 برای ژن شماره‌ی ۱۳ شکل ۲ برابر $31=31+0$ است. الگوریتم ارائه شده در خطوط ۲۶ الی ۳۴ امکان ارسال مستقیم کالا از کامیون‌های ورودی به خروجی را نشان می‌دهد. برای این منظور، خط ۲۶ تمام ژن‌های از ۱ تا IL را به‌منظور ارسال کالا به کامیون خروجی تحت برنامه‌ریزی بررسی می‌کند. کامیون ورودی در ژن شماره l درصورتی امکان ارسال مستقیم کالا به کامیون خروجی در ژن شماره j را دارد که نوع کالای کامیون ورودی در ژن l $(pt(i,l))$ با نوع کالای کامیون خروجی در ژن j $(pt(i,j))$ مطابقت داشته باشد (خط ۲۷) و هر دوی کالاهای ژن l و j بزرگتر از صفر باشند (خط ۲۸). در صورت مثبت بودن این دو شرط، گزینه‌ی دیگری برای زمان پایان بارگیری کامیون خروجی در ژن j به‌دست می‌آید که طبق خط ۲۹ عبارت است از زمان شروع عملیات تخلیه‌ی کامیون ورودی در ژن l $(st(i,l))$ به‌علاوه‌ی کمینه‌ی تعداد کالای موجود در ژن l و j $(\min\{Q(i,l),Q(i,j)\})$ به‌علاوه زمان انتقال کالا از جایگاه دریافت به جایگاه ارسال (v). با توجه به اینکه در این گام تعداد اولیه‌ی کالاهای ورودی و خروجی تغییر می‌کند، خط ۳۰ مقادیر باقی‌مانده را به روز می‌کند. حلقه‌ی ۳۴-۲۶ تکرار می‌شود تا کالاهای کامیون خروجی در ژن j به‌طور کامل تأمین شود. در هر بار تکرار حلقه در صورتی‌که کالایی ارسال شود، یک گزینه برای زمان پایان بارگیری محاسبه می‌شود که این مقدار در $\theta_2^{\text{counter}}$ قرار می‌گیرد. خط ۳۵ بیشینه‌ی مقادیر $\theta_2^{\text{counter}}$ را محاسبه می‌کند و در θ_2 قرار می‌دهد و خط ۳۶ بیشینه‌ی مقادیر θ_1 و θ_2 را محاسبه و عدد حاصل را به عنوان زمان پایان بارگیری کامیون خروجی ژن j لحاظ می‌کند.

۴.۲.۳. الگوریتم پیشنهادی عملگر تقاطع متناسب با مسئله

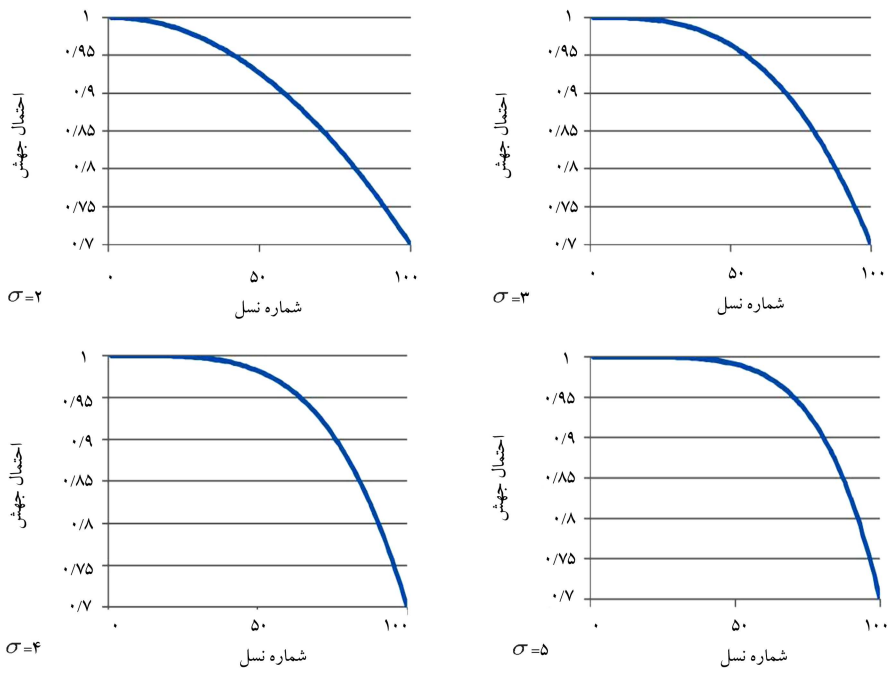
عملگر تقاطع در الگوریتم‌های ژنتیک مقادیر دو کروموزوم والد منتخب را عوض می‌کند و منجر به تولید کروموزوم‌های فرزند (نسل جدید) می‌شود. این عملگر به الگوریتم اجازه می‌دهد که جواب‌های جدید را جست‌وجو کند در حالی‌که کروموزوم‌های فرزند همچنان برخی از مشخصه‌های کروموزوم‌های والد را دارند. به‌منظور انجام عملیات تقاطع، نرخ تقاطع باید تعیین شود که این نرخ احتمالی است که بر اساس آن یک جفت کروموزوم تحت عملگر تقاطع قرار می‌گیرند. به‌علاوه، کروموزوم‌هایی که تحت عملگر تقاطع قرار نمی‌گیرند می‌توانند همچنان در گام‌های بعدی تحت عملگر جهش قرار بگیرند. شبه کد الگوریتم پیشنهادی عملگر تقاطع مطابق شکل ۵ است.

این یک الگوریتم تقاطع تک‌نقطه‌یی است که با احتمال P_c عمل تقاطع را انجام می‌دهد. برای این منظور خط ۳، دو کروموزوم را به‌صورت تصادفی بر اساس استراتژی رقابت انتخاب می‌کند. خط ۴ یک عدد تصادفی بین ۲ و طول توالی کامیون‌های ورودی منهای ۱ به‌عنوان نقطه‌ی تقاطع برای کامیون‌های ورودی (cross point۱) انتخاب می‌کند. انتخاب عدد ۲ و $IL-1$ بدین منظور است که در هر بار استفاده از عملگر تقاطع، حتما کروموزوم های جدید ایجاد شود. در صورتی‌که عدد تصادفی بین ۱ و IL انتخاب شود، با قرارگرفتن نقطه‌ی تقاطع بر روی ۱ یا IL، کروموزوم جدیدی ایجاد نخواهد شد. خط ۵ ژن‌های والد ۱ قبل از نقطه‌ی تقاطع را در کروموزوم فرزند ۱ کپی می‌کند. برای تکمیل ژن‌های بعد از نقطه‌ی تقاطع فرزند ۱، خط ۶ ژن‌های قابل انتقال را از والد ۲ به فرزند ۱ منتقل می‌کند. ژن‌های قابل انتقال ژن‌هایی هستند که در صورت اضافه‌شدن آنها از والد ۲ به فرزند ۱ ساختار اولیه‌ی کروموزوم تغییر نکند. به عبارت دیگر، اگر فرض کنیم در هر کامیون، هر نوع کالا در یک بسته قرار داشته باشد، در توالی کامیون‌های ورودی، هر کامیون فقط یکبار می‌تواند یک نوع کالا را تخلیه کند؛ بنابراین عملگر تقاطع باید به گونه‌یی باشد که در کروموزوم فرزند نیز یک کامیون برای تخلیه‌ی یک نوع کالا فقط یک‌بار وجود داشته باشد. در صورتی‌که یک نوع کالا در کامیونی در بیش از ۱ بسته قرار داده شده باشد، کامیون می‌تواند در توالی کامیون‌های ورودی به اندازه‌ی تعداد بسته‌ها برای کالای مورد نظر عملیات تخلیه را انجام دهد. خط ۷ ژن‌های والد ۲ قبل از نقطه‌ی تقاطع را در کروموزوم فرزند ۲ کپی می‌کند. برای تکمیل ژن‌های بعد از نقطه‌ی تقاطع فرزند ۲، خط ۸ ژن‌های قابل انتقال را از والد ۱ به فرزند ۲ منتقل می‌کند.

این عملیات باعث تولید دو کروموزوم فرزند می‌شود که توالی کامیون‌های ورودی را در خود جای داده است. عملیات مشابه آنچه برای کامیون‌های ورودی توضیح داده شد برای کامیون‌های خروجی نیز انجام می‌شود تا توالی کروموزوم‌های فرزند کامل شود. برای این منظور خط ۹ یک عدد تصادفی بین $IL+2$ و $IL+OL-1$ به‌عنوان نقطه‌ی تقاطع برای توالی کامیون‌های خروجی (cross point۲) انتخاب

```
1   ε ← a random number between [0,1]
2   if ε ≤ P_c then
3    select two parent chromosomes by tournament strategy
4    cross point_1 ← a random number between 2 and IL-1
5     copy the genes of parent #1 up to cross point_1 into offspring #1
6     copy the legitimate genes of parent #2 into offspring #1
7     copy the genes of parent #2 up to cross point_1 into offspring #2
8     copy the legitimate genes of parent #1 into offspring #2
9    cross point_2 ← a random number between IL+2 and IL+OL-1
10    copy the genes of parent #1 up to cross point_2 into offspring #1
11    copy the legitimate genes of parent #2 into offspring #1
12    copy the genes of parent #2 up to cross point_2 into offspring #2
13    copy the legitimate genes of parent #1 into offspring #2
14  end if
```

شکل ۵. الگوریتم پیشنهادی عملگر تقاطع.



شکل ۶. روند کاهش احتمال جهش در تکرارها.

می‌کند. خط ۱۰ ژن‌های والد ۱ قبل از نقطه‌ی تقاطع را در کروموزوم فرزند ۱ کپی می‌کند. برای تکمیل ژن‌های بعد از نقطه‌ی تقاطع فرزند ۱، خط ۱۱ ژن‌های قابل انتقال را از والد ۲ به فرزند ۱ منتقل می‌کند. خط ۱۲ ژن‌های والد ۲ قبل از نقطه‌ی تقاطع را در کروموزوم فرزند ۲ کپی می‌کند. برای تکمیل ژن‌های بعد از نقطه‌ی تقاطع فرزند ۲، خط ۱۳ ژن‌های قابل انتقال را از والد ۱ به فرزند ۲ منتقل می‌کند.

۵.۲.۳. الگوریتم پیشنهادی عملگر جهش متناسب با مسئله

عملگر جهش به‌طور تصادفی مقادیر برخی از ژن‌ها را در کروموزوم تغییر می‌دهد. این عملگر، یک عملگر حیاتی برای جلوگیری از دست‌یابی الگوریتم به جواب‌های محلی است.[۲۳] به‌منظور اجرای عملگر جهش، نرخ جهش باید تعیین شود به‌طوری‌که این نرخ احتمالی است که بر اساس آن یک کروموزوم تحت عملگر جهش قرار می‌گیرد. این مقاله یک الگوریتم پویای جهش پیشنهاد می‌کند به‌طوری‌که احتمال جهش در طول تکرارها (نسل‌ها) کاهش می‌یابد. همچنان‌که الگوریتم ژنتیک در طول نسل‌ها پیش می‌رود، به جواب بهینه نزدیکتر می‌شود. بنابراین، کاهش احتمال جهش در طول نسل‌ها می‌تواند از تغییرات نامطلوب در جواب‌ها جلوگیری کند. جهت اجرای جهش پویا، این مقاله رابطه‌ی ۲۸ را برای محاسبه‌ی احتمال جهش در هر تکرار (نسل) پیشنهاد می‌کند.

$$P_m^t = p_m^{\max} - \frac{(t-1)^{\sigma_m} \times (p_m^{\max} - p_m^{\min})}{(gen^{\max} - 1)^{\sigma_m}} \qquad (۲۸)$$

پارامتر p_m^t احتمال جهش در تکرار t را نشان می‌دهد؛ پارامتر $p_m^{\max}$ نشان‌دهنده‌ی بیشترین احتمال جهش و پارامتر $p_m^{\min}$ نشان‌دهنده‌ی کمترین احتمال جهش است؛ پارامتر σ_m یک عدد ثابت است. پارامتر $gen^{\max}$ نشان‌دهنده‌ی بیشترین تعداد نسل براساس شرایط توقف است.

پارامتر p_m^t احتمال جهش در تکرار t را به گونه‌یی محاسبه می‌کند که این احتمال از $p_m^{\max}$ در تکرار اول آغاز و به $p_m^{\min}$ در تکرار آخر ختم شود. ویژگی روش جهش پویای ارائه‌شده روند غیرخطی کاهشی با شیب فزاینده‌ی احتمال جهش در طول

```
1   ε ← a random number between [0,1]
2   if ε ≤ P_ml' then
3     p_l1 ← a random number between [1, bracket (IL/2)]
4     p_l2 ← a random number between [1, bracket (IL/2)]
5     p_21←p_l1+ bracket (IL/2)
6     p_22←p_l2+ bracket (IL/2)
7     swap "genes between [p_l1, p_l2]" for "genes between [p_21, p_22]"
8     p_l1 ← a random number between [IL+1, IL+bracket (OL/2)]
9     p_l2 ← a random number between [IL+1, IL+bracket (OL/2)]
10    p_21←p_l1+ bracket (OL/2)
11    p_22←p_l2+ bracket (OL/2)
12    swap "genes between [p_l1, p_l2]" for "genes between [p_21, p_22]"
13  end if
```

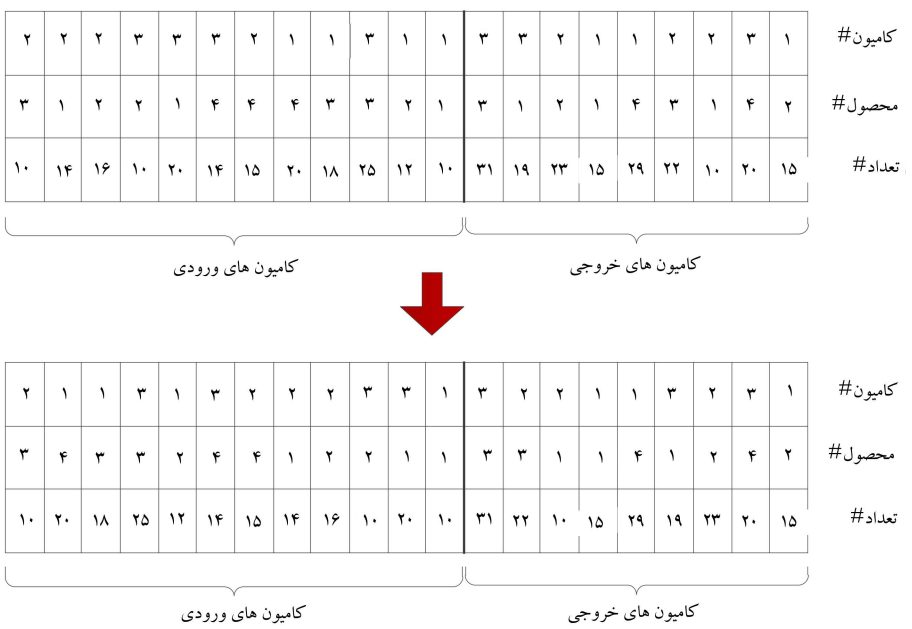
شکل ۷. الگوریتم پیشنهادی جهش معاوضه‌یی دسته‌یی.

تکرارها با استفاده از مقادیر بزرگتر از ۱ برای σ_m است. به عبارت دیگر، در تکرارهای اولیه احتمال جهش با نرخ پایین کاهش می‌یابد که این امر منجر به جست‌وجوی فضای گسترده‌یی از جواب‌ها می‌شود، اما همچنان‌که الگوریتم در تکرارها پیش می‌رود احتمال جهش با نرخ بیشتری کاهش می‌یابد که باعث کاهش فضای جست‌وجو می‌شود. شکل ۶ روند کاهش احتمال جهش را برای مقادیر مختلف σ_m نشان می‌دهد. در این شکل $p_m^{\max}$ و $p_m^{\min}$ به ترتیب برابر ۱ و ۰/۷ و بیشینه تعداد نسل برابر ۱۰۰ در نظر گرفته شده است. مشاهده می‌شود که با افزایش مقدار σ_m احتمال جهش در نسل‌های اولیه بیشتر و در نسل‌های بعدی با سرعت بیشتری کاهش می‌یابد. این مقاله دو روش مختلف جهش برای جست‌وجوی هرچه دقیق‌تر فضای جواب پیشنهاد می‌کند که عبارت‌اند از جهش معاوضه‌یی دسته‌یی و جهش معاوضه‌یی جداگانه که در ادامه به آنها پرداخته می‌شود.

جهش معاوضه‌یی دسته‌یی:

در این بخش به ارائه‌ی روش پیشنهادی جهش معاوضه‌یی دسته‌یی در قالب شکل ۷ پرداخته می‌شود.

الگوریتم پیشنهادی، جهش معاوضه‌یی دسته‌یی را در تکرار tام با احتمال p_{m}^{t}



شکل ۸. جهش معاوضه‌یی دسته‌یی.

انجام می‌دهد. برای این منظور، خطوط ۳ و ۴ دو عدد تصادفی بین ۱ و جزء صحیح $IL/۲$ تولید می‌کنند و مقادیر تولیدشده را در $p_{۱۱}$ و $p_{۱۲}$ قرار می‌دهند. در این صورت شماره‌ی دو ژن در نیمه‌ی اول توالی کامیون‌های ورودی به‌دست می‌آید و ژن‌های بین این دو ژن دسته اول نامیده می‌شود.

خطوط ۵ و ۶ به‌ترتیب حاصل جمع $p_{۱۱}+\text{bracket}(IL/۲)$ را در $p_{۲۱}$ و حاصل جمع $p_{۱۲}+\text{bracket}(IL/۲)$ را در $p_{۲۲}$ قرار می‌دهند. بر این اساس شماره‌ی دو ژن در نیمه‌ی دوم توالی کامیون‌های ورودی به‌دست می‌آید و ژن‌های بین این دو ژن دسته‌ی دوم نامیده می‌شود. خط ۷ موقعیت ژن‌های دسته اول و دوم را با یکدیگر تعویض می‌کند. به عبارت دیگر، خطوط ۳ الی ۷، توالی کامیون‌های ورودی را به دو نیم تقسیم می‌کنند. سپس یک دسته ژن به‌طور تصادفی در نیمه‌ی اول توالی کامیون‌های ورودی انتخاب و یک دسته ژن مطابق دسته‌ی اول (با تعداد ژن برابر) در نیمه‌ی دوم توالی کامیون‌های ورودی انتخاب و دسته‌ها با یکدیگر معاوضه می‌شوند. در خطوط ۳ الی ۶ از جزء صحیح $(IL/۲)$ برای دو نیم‌کردن توالی کامیون‌های ورودی استفاده شده، چون این امکان وجود دارد که طول توالی کامیون‌های ورودی (IL) یک عدد فرد باشد.

خطوط ۸ الی ۱۲ جهش معاوضه‌یی دسته‌یی را برای توالی کامیون‌های خروجی اجرا می‌کنند. بدین ترتیب که دو عدد تصادفی بین $۱+IL$ و $IL+\text{bracket}(OL/۲)$ ایجاد شده و ژن‌های بین این دو عدد تشکیل دسته‌ی اول را می‌دهند. همچنین ژن‌های بین $p_{۱۱}+\text{bracket}(OL/۲)$ و $p_{۱۲}+\text{bracket}(OL/۲)$ دسته دوم را تشکیل می‌دهند و خط ۱۲ موقعیت ژن‌های دسته اول و دوم را با یکدیگر تعویض می‌کند. به‌عبارت دیگر خطوط ۸ الی ۱۲، توالی کامیون‌های خروجی را به دو نیم تقسیم می‌کنند.

سپس یک دسته ژن به‌طور تصادفی در نیمه‌ی اول توالی کامیون‌های خروجی انتخاب و یک دسته ژن مطابق دسته‌ی اول (با تعداد ژن برابر) در نیمه‌ی دوم توالی کامیون‌های خروجی انتخاب و دسته‌ها با یکدیگر معاوضه می‌شوند. در خطوط ۸ الی ۱۱ از جزء صحیح $(OL/۲)$ برای دو نیم‌کردن توالی کامیون‌های خروجی استفاده شده است چون این امکان وجود دارد که طول توالی کامیون‌های خروجی (OL)

```
1   ε ← a random number between [0,1]
2   if ε ≤ P_m2^t then
3     counter_1 ← 1
4     do
5       select two random genes between [1,IL]
6       swap the amounts of selected genes
7       counter_1 ← counter_1 + 1
8     loop while (counter_1 ≤ δ_1)
9     counter_2 ← 1
10    do
11      select two random genes between [IL+1,IL+OL]
12      swap the amounts of selected genes
13      counter_2 ← counter_2 + 1
14    loop while (counter_2 ≤ δ_2)
15  end if
```

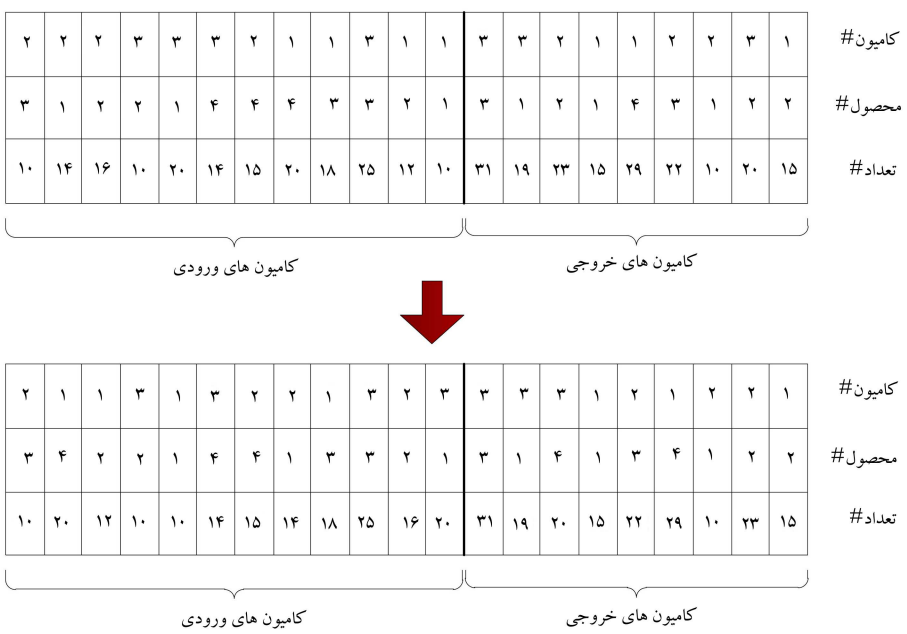
شکل ۹. الگوریتم پیشنهادی جهش معاوضه‌یی جداگانه.

یک عدد فرد باشد. شکل ۸ جهش معاوضه‌یی دسته‌یی را نشان می‌دهد.

جهش معاوضه‌یی جداگانه:

این بخش به ارائه‌ی روش پیشنهادی جهش معاوضه‌یی جداگانه در قالب شبه کد شکل ۹ می‌پردازد.

الگوریتم پیشنهادی جهش معاوضه‌یی جداگانه را در تکرار tام با احتمال $p^t_{m۲}$ انجام می‌دهد. همچنان که مشاهده می‌شود در روش جهش معاوضه‌یی دسته‌یی احتمال جهش با $p^t_{m۱}$ و در روش جهش معاوضه‌یی جداگانه احتمال جهش با $p^t_{m۲}$ نشان داده شده است. این تفاوت در علامت‌گذاری به این علت است که روش پیشنهادی جهش در این مقاله می‌تواند مقادیر مختلفی برای کمینه و بیشینه‌ی احتمال جهش برای دو نوع استراتژی مختلف جهش براساس رابطه‌ی ۱ در نظر بگیرد. به بیان دیگر، الگوریتم قادر است احتمال جهش معاوضه‌یی دسته‌یی را بر اساس رابطه‌ی ۱ از $p^{\min}_{m۱}$ تا $p^{\max}_{m۱}$ و احتمال جهش معاوضه‌یی جداگانه را بر اساس رابطه‌ی ۱ از $p^{\min}_{m۲}$ تا $p^{\max}_{m۲}$ در نظر بگیرد.



شکل ۱۰. جهش معاوضه‌یی جداگانه.

جهت اجرای جهش معاوضه‌یی جداگانه در توالی کامیون‌های ورودی، خط ۵ دو ژن را به تصادف بین ۱ و IL انتخاب می‌کند و خط ۶ مقادیر این دو ژن را با یکدیگر معاوضه می‌کند. حلقه‌ی ۸-۴ الگوریتم را قادر می‌سازد تا این عمل δ_1 بار انجام شود تا از این طریق هر بار مقادیر دو ژن در توالی کامیون‌های ورودی با یکدیگر معاوضه شود. برای اجرای جهش معاوضه‌یی جداگانه در توالی کامیون‌های خروجی، خط ۱۱ دو ژن را به تصادف بین ۱ و $IL+1$ و $IL+OL$ انتخاب می‌کند و خط ۱۲ مقادیر این دو ژن را با یکدیگر معاوضه می‌کند. حلقه‌ی ۱۴-۱۰ الگوریتم را قادر می‌سازد تا این عمل δ_2 بار انجام شود تا از این طریق هر بار مقادیر دو ژن در توالی کامیون‌های خروجی با یکدیگر معاوضه شود. شکل ۱۰ جهش معاوضه‌یی جداگانه را به ازای $\delta_1 = 3$ و $\delta_2 = 2$ نشان می‌دهد.

محاسبه‌ی برازندگی کروموزوم‌های فرزند و تولید مجدد:

در بخش ۴.۲.۳. و ۵.۲.۳. الگوریتم‌های پیشنهادی تقاطع و جهش معرفی شدند. با اجرای الگوریتم جهش، کروموزوم‌های جدیدی به نام کروموزوم‌های فرزند ایجاد می‌شوند که مشخصه‌هایشان تا حدی با مشخصه‌های کروموزوم‌های والد تفاوت دارد. بنابراین، پس از تولید کروموزوم‌های فرزند، نیاز است تا برازندگی هر یک تعیین شود. برای این منظور، الگوریتمی که در بخش ۳.۲.۳. برای محاسبه‌ی برازندگی جواب‌ها معرفی شد، برای محاسبه‌ی برازندگی کروموزوم‌های فرزند نیز به کار می‌رود.

پس از محاسبه‌ی برازندگی کروموزوم‌های فرزند، استراتژی رقابت و جایگزینی برای تولید نسل جدید به کار می‌رود. برای این منظور، هر بار α کروموزوم از جمعیت والد و فرزند به‌طور مستقل انتخاب می‌شود و با استفاده از استراتژی رقابت بهترین کروموزوم در جمعیت والد و فرزند انتخاب می‌شود. سپس مجدداً با استفاده از استراتژی رقابت، بهترین کروموزوم بین کروموزوم منتخب والد و کروموزوم منتخب فرزند انتخاب می‌شود و به نسل بعد منتقل می‌شود. عملیات فوق تکرار می‌شود تا زمانی‌که تعداد کروموزوم‌های نسل جدید به popsize برسد. بر اساس این رویکرد، نسل جدید شامل کروموزوم‌های منتخب از هر دوی جمعیت‌های والد و فرزند است. عملیات محاسبه‌ی برازندگی، تقاطع و جهش بر روی نسل جدید کروموزوم‌ها اجرا می‌شود تا زمانی‌که شرایط توقف حاصل شود. در روش پیشنهادی این مقاله از بیشینه‌ی تعداد نسل‌ها برای توقف الگوریتم استفاده شده است.

۴. پیاده‌سازی و نتایج عددی

این بخش به ارائه‌ی نتایج حل مسائل عددی برای بررسی عملکرد الگوریتم پیشنهادی می‌پردازد. به‌منظور بررسی صحت روش حل پیشنهادی، جواب‌های روش حل پیشنهادی مسائل عددی با مدل ریاضی ارائه‌شده در بخش ۱.۳. مقایسه شده است. به‌منظور بررسی کیفیت جواب‌های ارائه‌شده توسط الگوریتم پیشنهادی، نتایج حل مسائل عددی توسط الگوریتم پیشنهادی با روش شمارش کامل[۱] مقایسه شده است. دلیل مقایسه‌ی الگوریتم پیشنهادی با روش شمارش کامل این است که بین سه رویکرد ارائه شده در[۱] که شامل رویکردهای مدل بهینه‌سازی، شمارش کامل و روش‌های ابتکاری است، رویکرد شمارش کامل به‌طور قطع بهتر یا حداقل برابر با دو رویکرد دیگر عمل می‌کند؛ زیرا این رویکرد به بررسی کل جواب‌های ممکن می‌پردازد. بنابراین مبنای بسیار خوبی برای مقایسه است.

همچنان‌که در بخش ۱ ذکر شد، وحدانی و زندیه[۴] پنج روش فراابتکاری را برای حل مسئله‌ی بارانداز تقاطعی ارائه کردند و نشان دادند که روش‌های فراابتکاری ارائه شده بهتر از روش‌های ابتکاری[۱] عمل می‌کنند، اما ایشان روش‌های فراابتکاری را با رویکرد شمارش کامل[۱] مقایسه نکردند. در واقع، واضح است که رویکرد شمارش کامل به‌طور قطع بهتر یا حداقل برابر با روش‌های فراابتکاری[۴] و روش‌های ابتکاری[۱] عمل می‌کند. زیرا به‌بررسی کامل جواب‌های ممکن می‌پردازد. بنابراین، از آنجایی‌که در این بخش نشان داده می‌شود که روش پیشنهادی این مقاله بهتر از رویکرد شمارش کامل[۱] عمل می‌کند، می‌توان نتیجه گرفت که روش پیشنهادی بهتر از روش‌های فراابتکاری[۴] و روش‌های ابتکاری[۱] عمل می‌کند.

در تمام مسائل عددی این بخش، زمان انتقال کالاها از جایگاه دریافت تا ارسال (v) برابر ۱۰۰ و زمان جابه‌جایی وسایل نقلیه در جایگاه (changeover) برابر ۱۵

در نظرگرفته شده است. به‌علاوه زمان تخلیه و بارگیری برای تمام کالاها یکسان و برابر ۱ واحد زمانی برای یک کالا در نظرگرفته شده است.

۱.۴. پارامترهای الگوریتم

پارامترهای الگوریتم ژنتیک پیشنهادی بر اساس رویکرد تجربی به‌صورت آزمون و خطا تعیین شده است. در تمام مسائل عددی، اندازه‌ی جمعیت (popsize) برابر ۵۰۰، بیشینه تعداد نسل‌ها ($gen^{\max}$) برابر ۱۰۰۰، نرخ تقاطع (P_c) برابر ۰٫۹، بیشینه‌ی نرخ جهش معاوضه‌یی دسته‌یی برابر ۰٫۵، کمینه‌ی نرخ جهش معاوضه‌یی دسته‌یی برابر ۰٫۰۵، بیشینه‌ی نرخ جهش معاوضه‌یی جداگانه برابر ۰٫۶، کمینه‌ی نرخ جهش معاوضه‌یی جداگانه برابر ۰٫۱، تعداد کروموزوم‌ها برای استراتژی رقابت برابر ۱۰، تعداد دفعات اجرای جهش معاوضه‌یی جداگانه (δ_1 و δ_2) برابر ۳ در نظر گرفته شده است.

۲.۴. تحلیل نتایج

برای نشان‌دادن صحت و عملکرد مناسب الگوریتم پیشنهادی، ۲۰ مسئله با مقادیر تخلیه و بارگیری تصادفی ایجاد و توسط مدل ریاضی، الگوریتم پیشنهادی و روش شمارش کامل[۱] حل شده است. جدول ۱ نشان‌دهنده‌ی نتایج حل مسائل عددی توسط مدل ریاضی و روش حل پیشنهادی است و جدول ۲ نشان‌دهنده‌ی نتایج حل مسائل توسط روش حل پیشنهادی و روش شمارش کامل[۱] است.

همچنان که در جدول ۱ مشاهده می‌شود، در چهارده مسئله از بیست مسئله، زمان عملیات روش حل پیشنهادی برابر زمان عملیات مدل ریاضی است (مدل ریاضی با نرم افزار Lingo ۱۱ حل شده است) و در شش مسئله از بیست مسئله، زمان عملیات مدل ریاضی کمتر از روش حل پیشنهادی شده است. متوسط درصد انحراف زمان عملیات مدل ریاضی و روش پیشنهادی در این بیست مسئله برابر ۰٫۴۲٪ است که عددی بسیار پایین است. از طرف دیگر دو ستون آخر جدول که نشان‌دهنده‌ی زمان CPU به دقیقه‌اند نشان می‌دهند که زمان CPU مدل ریاضی به مراتب بالاتر از روش حل پیشنهادی است که دلیل آن NP-Hard بودن مدل ریاضی و تعداد متغیرها و محدودیت‌های بسیار زیاد در مدل ریاضی است که تا حد زیادی مدل ریاضی را ناکارامد می‌کند. لازم به توضیح است که NP-Hard بودن مدل‌های بارانداز متقاطع و عدم کارایی مدل‌های ریاضی برای این نوع مسائل توسط محققان دیگری یو و اگبلو[۱] نشان داده شده است. میانگین زمان CPU در روش حل پیشنهادی برابر ۵٫۴۷ دقیقه است که زمانی منطقی و کاراست و برای مدل ریاضی برابر ۸۶۵٫۶۹ دقیقه است. بنابراین، مشاهده می‌شود که روش حل پیشنهادی از کارایی بسیار بالایی در مقابل مدل ریاضی برخوردار است.

همچنان‌که در جدول ۲ مشاهده می‌شود، ۲۰ مسئله‌ی مختلف با تعداد متفاوت کامیون‌های ورودی، کامیون‌های خروجی، نوع محصول و کل محصولات در این بخش ارائه شده است. زمان عملیات حاصل از دو روش پیشنهادی و شمارش کامل[۱] به‌ترتیب در ستون‌های «زمان عملیات روش پیشنهادی» و «زمان عملیات شمارش کامل YE» ارائه شده است. همچنان‌که در این دو ستون مشاهده می‌شود، زمان عملیات روش پیشنهادی در تمام ۲۰ مسئله کمتر از روش شمارش کامل[۱] است. ستون‌های «انحراف زمان عملیات» و «درصد انحراف زمان عملیات»، به‌ترتیب میزان

جدول ۱. مقایسه‌ی نتایج مدل ریاضی و روش حل پیشنهادی.

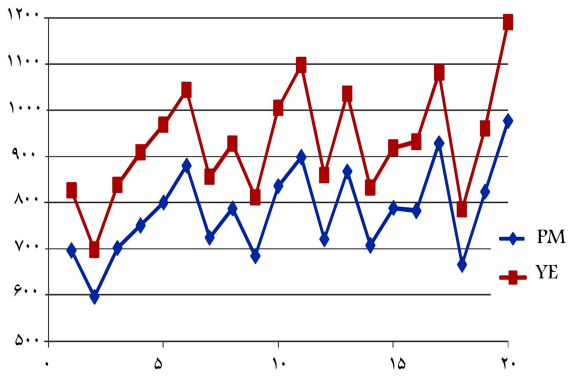
مسئله	تعداد کامیون‌های		تعداد نوع محصول	تعداد کل محصولات	زمان عملیات		انحراف زمان عملیات	درصد انحراف زمان عملیات	زمان CPU	
	ورودی	خروجی			روش پیشنهادی	مدل ریاضی			روش حل پیشنهادی	مدل ریاضی
۱	۴	۴	۵	۴۳۶	۶۹۶	۶۹۶	۰	۰٫۰۰	۳٫۳۳	۳۹۷٫۲۴
۲	۴	۴	۶	۳۴۴	۵۹۶	۵۸۵	۱۱	۱٫۸۸	۳٫۰۶	۳۷۷٫۲۴
۳	۴	۵	۵	۴۲۷	۷۰۲	۷۰۲	۰	۰٫۰۰	۴٫۲۲	۵۴۵٫۷۲
۴	۴	۵	۶	۴۶۷	۷۵۰	۷۵۰	۰	۰٫۰۰	۵٫۰۸	۶۳۸٫۵۹
۵	۴	۶	۶	۴۸۴	۸۰۰	۷۸۸	۱۲	۱٫۵۲	۵٫۱۷	۷۴۸٫۸۶
۶	۴	۶	۷	۵۴۴	۸۸۰	۸۸۰	۰	۰٫۰۰	۶٫۶۹	۹۶۳٫۸۶
۷	۵	۴	۶	۴۳۹	۷۲۴	۷۲۴	۰	۰٫۰۰	۴٫۹۴	۶۳۹٫۶۸
۸	۵	۴	۷	۴۷۹	۷۸۷	۷۸۷	۰	۰٫۰۰	۵٫۸۳	۷۲۲٫۱۸
۹	۵	۵	۴	۴۱۲	۶۸۴	۶۸۴	۰	۰٫۰۰	۳٫۹۷	۶۰۵٫۹۲
۱۰	۵	۵	۶	۵۱۱	۸۳۶	۸۲۱	۱۵	۱٫۸۳	۵٫۰۰	۷۴۷٫۹۳
۱۱	۵	۵	۷	۵۲۴	۸۹۸	۸۹۲	۶	۰٫۶۷	۶٫۹۴	۱۰۲۰٫۰۵
۱۲	۵	۶	۵	۴۲۱	۷۲۱	۷۲۱	۰	۰٫۰۰	۴٫۷۲	۷۸۸٫۹۱
۱۳	۵	۶	۶	۵۰۶	۸۶۸	۸۶۸	۰	۰٫۰۰	۶٫۵۳	۱۰۸۷٫۹۳
۱۴	۶	۴	۶	۴۰۷	۷۰۷	۷۰۷	۰	۰٫۰۰	۴٫۳۳	۶۴۰٫۶۰
۱۵	۶	۴	۷	۴۶۷	۷۸۸	۷۸۸	۰	۰٫۰۰	۵٫۸۳	۸۷۰٫۷۸
۱۶	۶	۵	۵	۴۳۳	۷۸۲	۷۷۶	۶	۰٫۷۷	۵٫۶۹	۹۵۵٫۸۷
۱۷	۶	۵	۷	۵۶۹	۹۲۸	۹۱۲	۱۶	۱٫۷۵	۸٫۱۷	۱۳۴۴٫۶۲
۱۸	۶	۶	۴	۳۷۷	۶۶۵	۶۶۵	۰	۰٫۰۰	۴٫۱۱	۸۵۷٫۲۳
۱۹	۶	۶	۵	۴۷۲	۸۲۳	۸۲۳	۰	۰٫۰۰	۶٫۱۹	۱۴۳۳٫۶۱
۲۰	۶	۶	۷	۵۹۲	۹۷۷	۹۷۷	۰	۰٫۰۰	۹٫۷۲	۱۹۲۷٫۰۴

جدول ۲. مقایسه‌ی نتایج روش حل پیشنهادی و روش شمارش کامل.[۱]

مسئله	تعداد کامیون‌های		تعداد نوع محصول	تعداد کل محصولات	زمان عملیات		انحراف زمان عملیات	درصد انحراف زمان عملیات	زمان CPU	
	ورودی	خروجی			روش پیشنهادی	شمارش کامل YE			روش حل پیشنهادی	شمارش کامل YE
۱	۴	۴	۵	۴۳۶	۶۹۶	۸۲۶	۱۳۰	۱۸٫۶۸	۳٫۳۳	۰٫۰۰۹
۲	۴	۴	۶	۳۴۴	۵۹۶	۶۹۷	۱۰۱	۱۶٫۹۵	۳٫۰۶	۰٫۰۰۹
۳	۴	۵	۵	۴۲۷	۷۰۲	۸۳۸	۱۳۶	۱۹٫۳۷	۴٫۲۲	۰٫۰۳۶
۴	۴	۵	۶	۴۶۷	۷۵۰	۹۰۸	۱۵۸	۲۱٫۰۷	۵٫۰۸	۰٫۰۴۲
۵	۴	۶	۶	۴۸۴	۸۰۰	۹۶۸	۱۶۸	۲۱٫۰۰	۵٫۱۷	۰٫۱۶۴
۶	۴	۶	۷	۵۴۴	۸۸۰	۱۰۴۳	۱۶۳	۱۸٫۵۲	۶٫۶۹	۰٫۱۶۹
۷	۵	۴	۶	۴۳۹	۷۲۴	۸۵۵	۱۳۱	۱۸٫۰۹	۴٫۹۴	۰٫۰۳۹
۸	۵	۴	۷	۴۷۹	۷۸۷	۹۲۷	۱۴۰	۱۷٫۷۹	۵٫۸۳	۰٫۰۴۴
۹	۵	۵	۴	۴۱۲	۶۸۴	۸۱۱	۱۲۷	۱۸٫۵۷	۳٫۹۷	۰٫۱۹۵
۱۰	۵	۵	۶	۵۱۱	۸۳۶	۱۰۰۴	۱۶۸	۲۰٫۱۰	۵٫۰۰	۰٫۱۹۵
۱۱	۵	۵	۷	۵۲۴	۸۹۸	۱۰۹۷	۱۹۹	۲۲٫۱۶	۶٫۹۴	۰٫۲۰۱
۱۲	۵	۶	۵	۴۲۱	۷۲۱	۸۵۹	۱۳۸	۱۹٫۱۴	۴٫۷۲	۰٫۸۷۴
۱۳	۵	۶	۶	۵۰۶	۸۶۸	۱۰۳۵	۱۶۷	۱۹٫۲۴	۶٫۵۳	۰٫۸۷۸
۱۴	۶	۴	۶	۴۰۷	۷۰۷	۸۳۲	۱۲۵	۱۷٫۶۸	۴٫۳۳	۰٫۱۶۰
۱۵	۶	۴	۷	۴۶۷	۷۸۸	۹۱۸	۱۳۰	۱۶٫۵۰	۵٫۸۳	۰٫۱۶۵
۱۶	۶	۵	۵	۴۳۳	۷۸۲	۹۳۱	۱۴۹	۱۹٫۰۵	۵٫۶۹	۰٫۸۵۰
۱۷	۶	۵	۷	۵۶۹	۹۲۸	۱۰۸۱	۱۵۳	۱۶٫۴۹	۸٫۱۷	۰٫۸۷۲
۱۸	۶	۶	۴	۳۷۷	۶۶۵	۷۸۵	۱۲۰	۱۸٫۰۵	۴٫۱۱	۱٫۰۰۶
۱۹	۶	۶	۵	۴۷۲	۸۲۳	۹۶۰	۱۳۷	۱۶٫۶۵	۶٫۱۹	۱٫۰۲۲
۲۰	۶	۶	۷	۵۹۲	۹۷۷	۱۱۹۰	۲۱۳	۲۱٫۸۰	۹٫۷۲	۱٫۰۶۱

جدول ۳. آمار توصیفی زمان عملیات و زمان CPU.

	کمینه	بیشینه	میانگین	انحراف استاندارد
زمان عملیات روش پیشنهادی	۵۹۶	۹۷۷	۷۸۰٫۶	۹۷٫۳۷۵
زمان عملیات شمارش کامل YE	۶۹۷	۱۱۹۰	۹۲۸٫۲۵	۱۲۱٫۲۸۸
انحراف زمان عملیات	۱۰۱	۲۱۳	۱۴۷٫۶۵	۲۶٫۸۱۷
درصد انحراف زمان عملیات	۱۶٫۴۹	۲۲٫۱۶	۱۸٫۸۴۵	۱٫۶۹۸
زمان CPU روش پیشنهادی	۳٫۰۶	۹٫۷۲	۵٫۴۷۶	۱٫۶۱۵
زمان CPU شمارش کامل YE	۰٫۰۰۹	۱٫۰۶۱	۰٫۳۹۹	۰٫۴۱۲



شکل ۱۱. زمان عملیات روش پیشنهادی و روش شمارش کامل.[۱]

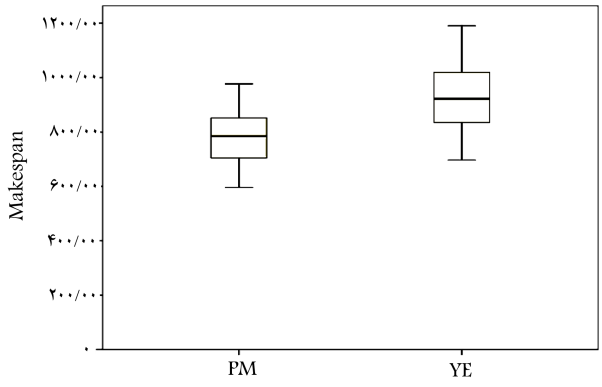
و درصد انحراف زمان عملیات بین دو روش را نشان می‌دهند. به‌علاوه ستون‌های «زمان CPU روش پیشنهادی» و «زمان CPU شمارش کامل YE»، زمان حل مسائل توسط دو روش را نشان می‌دهند. لازم به توضیح است که تمام مسائل با یک رایانه Core i5 CPU (۲٫۶ GHz) و ۴ گیگابایت RAM اجرا شده‌اند. زمان CPU روش شمارش کامل[۱] در این ۲۰ مسئله کمتر از روش پیشنهادی است، اما برای مسائل مقیاس بزرگ زمان CPU روش پیشنهادی به مراتب کمتر است، زیرا برای مثال مسئله‌یی با ۱۰ کامیون ورودی و ۱۰ کامیون خروجی با استفاده از روش شمارش کامل[۱] نیازمند بررسی $۱٫۳۱۶ \times ۱۰^{۱۳} = ۱۰! \times ۱۰!$ جواب است که بسیار زمان بر است و این امر باعث کارایی بسیار ضعیف این روش می‌شود.

جدول ۳ مقادیر کمینه، بیشینه، میانگین و انحراف استاندارد را برای زمان عملیات دو روش، میزان و درصد انحراف زمان عملیات دو روش، و زمان CPU دو روش را نشان می‌دهد. همچنان‌که مشاهده می‌شود کمینه‌ی درصد انحراف زمان عملیات دو روش برابر ۱۶٫۴۹، بیشینه‌ی درصد انحراف زمان عملیات دو روش برابر ۲۲٫۱۶ و میانگین درصد انحراف زمان عملیات دو روش برابر ۱۸٫۸۴ است که نشان‌دهنده‌ی برتری روش پیشنهادی نسبت به روش شمارش کامل[۱] است. شکل ۱۱ زمان عملیات روش پیشنهادی را که با PM علامت‌گذاری شده و روش شمارش کامل[۱] را که با YE علامت‌گذاری شده‌اند، نشان می‌دهد.

برای مقایسه‌ی آماری زمان عملیات روش پیشنهادی و روش شمارش کامل[۱]، از نمودار جعبه‌یی و آزمون T نمونه‌های مستقل استفاده شده است. شکل ۱۲ نمودار جعبه‌یی را برای زمان عملیات دو روش نشان می‌دهد. در این شکل محور

جدول ۴. مقایسه الگوریتم ژنتیک با الگوریتم بهینه‌سازی ازدحام ذرات.

مسئله	تعداد کامیون‌های		تعداد نوع محصول	تعداد کل محصولات	زمان عملیات		انحراف زمان عملیات	درصد انحراف زمان عملیات	زمان CPU	
	ورودی	خروجی			روش پیشنهادی الگوریتم ژنتیک	الگوریتم بهینه‌سازی ازدحام ذرات			روش پیشنهادی الگوریتم ژنتیک	الگوریتم بهینه‌سازی ذرات
۱	۴	۴	۵	۴۳۶	۶۹۶	۷۸۷	۹۱	۱۳٫۰۷	۳٫۳۳	۴٫۳۳
۲	۴	۴	۶	۳۴۴	۵۹۶	۶۴۵	۴۹	۸٫۲۲	۳٫۰۶	۴٫۰۶
۳	۴	۵	۵	۴۲۷	۷۰۲	۷۱۹	۱۷	۲٫۴۲	۴٫۲۲	۵٫۲۲
۴	۴	۵	۶	۴۶۷	۷۵۰	۷۸۶	۳۶	۴٫۸۰	۵٫۰۸	۶٫۰۸
۵	۴	۶	۶	۴۸۴	۸۰۰	۸۵۱	۵۱	۶٫۳۸	۵٫۱۷	۶٫۱۷
۶	۴	۶	۷	۵۴۴	۸۸۰	۸۹۰	۱۰	۱٫۱۴	۶٫۶۹	۷٫۶۹
۷	۵	۴	۶	۴۳۹	۷۲۴	۸۰۱	۷۷	۱۰٫۶۴	۴٫۹۴	۵٫۹۴
۸	۵	۴	۷	۴۷۹	۷۸۷	۸۶۰	۷۳	۹٫۲۸	۵٫۸۳	۶٫۸۳
۹	۵	۵	۴	۴۱۲	۶۸۴	۷۴۰	۵۶	۸٫۱۹	۳٫۹۷	۴٫۹۷
۱۰	۵	۵	۶	۵۱۱	۸۳۶	۸۵۵	۱۹	۲٫۲۷	۵	۵٫۳۷
۱۱	۵	۵	۷	۵۲۴	۸۹۸	۹۸۳	۸۵	۹٫۴۷	۶٫۹۴	۷٫۹۴
۱۲	۵	۶	۵	۴۲۱	۷۲۱	۷۹۱	۷۰	۹٫۷۱	۴٫۷۲	۵٫۷۲
۱۳	۵	۶	۶	۵۰۶	۸۶۸	۸۸۰	۱۲	۱٫۳۸	۶٫۵۳	۷٫۵۳
۱۴	۶	۴	۶	۴۰۷	۷۰۷	۷۲۵	۱۸	۲٫۵۵	۴٫۳۳	۵٫۳۳
۱۵	۶	۴	۷	۴۶۷	۷۸۸	۹۱۳	۱۲۵	۱۵٫۸۶	۵٫۸۳	۶٫۸۳
۱۶	۶	۵	۵	۴۳۳	۷۸۲	۷۹۵	۱۳	۱٫۶۶	۵٫۶۹	۶٫۶۹
۱۷	۶	۵	۷	۵۶۹	۹۲۸	۱۰۵۴	۱۲۶	۱۳٫۵۸	۸٫۱۷	۹٫۱۷
۱۸	۶	۶	۴	۳۷۷	۶۶۵	۷۰۹	۴۴	۶٫۶۲	۴٫۱۱	۵٫۱۱
۱۹	۶	۶	۵	۴۷۲	۸۲۳	۸۵۳	۳۰	۳٫۶۵	۶٫۱۹	۷٫۱۹
۲۰	۶	۶	۷	۵۹۲	۹۷۷	۱۱۰۱	۱۲۴	۱۲٫۶۹	۹٫۷۲	۱۰٫۷۲
۲۱	۱۰	۱۰	۸	۶۲۴	۱۲۳۷	۱۳۱۸	۸۱	۶٫۵۵	۱۰٫۱۴	۱۱٫۲۶
۲۲	۱۴	۱۰	۱۰	۶۱۰	۱۲۵۶	۱۳۲۴	۶۸	۵٫۴۱	۱۰٫۲۳	۱۱٫۳۷
۲۳	۱۰	۱۴	۸	۷۱۶	۱۲۴۱	۱۴۰۶	۱۶۵	۱۳٫۳۰	۱۰٫۷۶	۱۱٫۷۲
۲۴	۱۴	۱۴	۱۰	۷۴۳	۱۳۵۶	۱۵۱۲	۱۵۶	۱۱٫۵۰	۱۱٫۰۸	۱۲٫۱۹
۲۵	۱۸	۱۶	۱۲	۸۱۳	۱۴۰۱	۱۴۶۸	۶۷	۴٫۷۸	۱۱٫۳۵	۱۲٫۲۹
۲۶	۱۶	۱۸	۱۰	۸۰۲	۱۴۶۳	۱۵۴۰	۷۷	۵٫۲۶	۱۱٫۲۱	۱۲٫۲۸
۲۷	۱۸	۱۸	۱۲	۹۵۶	۱۴۹۵	۱۶۸۳	۱۸۸	۱۲٫۵۸	۱۱٫۸۴	۱۲٫۸۷
۲۸	۲۰	۱۸	۱۵	۸۶۱	۱۴۷۲	۱۵۳۹	۶۷	۴٫۵۵	۱۱٫۷۶	۱۲٫۶۳
۲۹	۱۸	۲۰	۱۵	۱۰۱۲	۱۵۴۸	۱۷۳۴	۱۸۶	۱۲٫۰۲	۱۲٫۲۱	۱۳٫۵۴
۳۰	۲۰	۲۰	۱۸	۱۰۳۵	۱۵۶۱	۱۷۲۱	۱۶۰	۱۰٫۲۵	۱۲٫۳۴	۱۳٫۷۸



شکل ۱۲. نمودار جعبه‌یی برای مقایسه زمان عملیات دو روش.

افقی نشان‌دهنده‌ی روش پیشنهادی و روش شمارش کامل[۱] است و محور عمودی نشان‌دهنده‌ی زمان عملیات است. همچنان که در شکل ۱۲ مشاهده می‌شود، توزیع زمان عملیات برای روش پیشنهادی تقریباً پایین‌تر از توزیع روش[۱] است. به‌علاوه، کل جعبه روش پیشنهادی که شامل ۵۰ درصد داده‌ها حول میانگین می‌شود تقریباً پایین‌تر از جعبه[۱] است. نتایج مقایسه‌ی میانگین‌های دو روش با استفاده از آزمون T نمونه‌های مستقل، مقدار آماره‌ی T را برابر ۴٫۲۴- و مقدار P-value را برابر ۰٫۰۰۰ نشان می‌دهد. بنابر مقادیر حاصل، میانگین زمان عملیات روش پیشنهادی در سطح اطمینان ۹۵ درصد کمتر از روش شمارش کامل[۱] است.

برای بررسی کیفیت جواب‌های ارائه‌شده توسط الگوریتم ژنتیک پیشنهادی در مقایسه با دیگر الگوریتم‌های فراابتکاری، سی مثال عددی ارائه شده و توسط الگوریتم فراابتکاری بهینه‌سازی ازدحام ذرات[۲۴] حل شده و نتایج در جدول ۴

نشان داده شده است. پارامترهای الگوریتم بهینه‌سازی ازدحام ذرات به‌صورت تجربی/آزمایشی تعیین شده است. بر این اساس برای این الگوریتم، اندازه‌ی جمعیت برابر ۱۰۰، فاکتور انقباض برابر ۰/۷، پارامتر فردی برابر ۱/۵، پارامتر جمعی برابر ۲، پارامتر اینرسی برابر ۰/۳ و تعداد تکرارها برابر ۲۰۰ در نظر گرفته شده است. همچنان که از نتایج جدول ۴ مشاهده می‌شود، الگوریتم ژنتیک پیشنهادی به جواب‌های بهتری نسبت به الگوریتم بهینه‌سازی ازدحام ذرات دست یافته است.

۵. نتیجه‌گیری

جریان فیزیکی در زنجیره‌ی تأمین یکی از مهم‌ترین مسائل در پیاده‌سازی مدیریت زنجیره‌ی تأمین است. در محیط رقابتی امروز، تولیدکننده‌ها و توزیع‌کننده‌ها سعی می‌کنند محصولات را به مشتریان سریع و با هزینه‌های پایین عرضه کنند. بنابراین، بسیاری از شرکت‌ها در پی یافتن روش‌های مناسب برای کنترل جریان مواد هستند. برای این منظور، استراتژی توزیعی به نام بارانداز تقاطعی یک استراتژی مناسب برای کاهش موجودی است. هدف این مقاله دست‌یابی به یک برنامه‌ی زمان‌بندی بهینه/نزدیک بهینه تخلیه و بارگیری برای وسایل نقلیه‌ی ورودی و خروجی در بارانداز تقاطعی به نحوی است که زمان عملیات کمینه شود. منظور از زمان عملیات، بازه‌ی زمانی بین زمان تخلیه‌ی اولین واحد کالا توسط اولین وسیله‌ی نقلیه‌ی ورودی تا بارگیری آخرین کالا توسط آخرین وسیله‌ی نقلیه‌ی خروجی است. برای این منظور، این مقاله یک روش جدید مبتنی بر الگوریتم ژنتیک برای زمان‌بندی تخلیه و بارگیری وسائل نقلیه‌ی ورودی و خروجی پیشنهاد می‌کند. روش پیشنهادی این مقاله به وسایل نقلیه‌ی ورودی و خروجی امکان تخلیه و بارگیری مکرر را می‌دهد. به‌عبارت دیگر در روش پیشنهادی این مقاله، یک وسیله‌ی نقلیه‌ی ورودی می‌تواند به جایگاه تخلیه وارد شود و بخشی از کالاهای خود را تخلیه کند و از جایگاه بیرون رود تا یک وسیله‌ی نقلیه‌ی ورودی دیگر عملیات تخلیه‌ی بخشی از کالاهای خود را انجام دهد و پس از مدتی مجدداً به جایگاه تخلیه بازگردد و بخشی دیگر از کالاهای خود را تخلیه کند. این عملیات برای وسایل نقلیه‌ی خروجی نیز برای بارگیری کالاها به همین صورت است.

برای دست‌یابی به جواب نزدیک بهینه، در این مقاله روش‌های پیشنهادی تولید جمعیت اولیه، محاسبه‌ی برازندگی کروموزوم‌ها، عملگرهای تقاطع و جهش ارائه شده است. الگوریتم جهش پیشنهادی در این مقاله یک الگوریتم پویاست؛ به‌نحوی‌که احتمال جهش در طول تکرارها (نسل‌ها) را کاهش می‌دهد. همچنان‌که الگوریتم ژنتیک در طول نسل‌ها پیش می‌رود، به جواب بهینه نزدیکتر می‌شود. بنابراین، کاهش احتمال جهش در طول نسل‌ها می‌تواند از تغییرات نامطلوب در جواب‌ها جلوگیری کند.

به‌منظور ارزیابی عملکرد روش پیشنهادی برای زمان‌بندی وسایل نقلیه‌ی ورودی و خروجی، بیست مسئله‌ی مختلف تولید و حل شده و جواب‌های روش پیشنهادی با جواب‌های روش شمارش کامل[۱] مقایسه شده است. نتایج مقایسه‌یی نشان‌دهنده‌ی برتری روش پیشنهادی در هر بیست مسئله است زیرا این روش منجر به ارائه‌ی جواب‌هایی با زمان عملیات کمتر می‌شود.

در این مقاله، پارامترهای الگوریتم ژنتیک به‌صورت آزمون و خطا تعیین شد؛ پیشنهاد می‌شود در مطالعات بعدی، پارامترها با روش‌هایی مانند طراحی آزمایش‌ها تعیین و نتایج آن با نتایج این مقاله مقایسه شود.

منابع (References)

1. Yu, W. and Egbelu, P.J. "Scheduling of inbound and outbound trucks in cross docking systems with temporary storage", *European Journal of Operational Research*, **184**, pp. 377-396 (2008).
2. Wen, M., Larsen, J., Clausen, J., Cordenau, J.F. and Laporte, G. "Vehicle routing with cross-docking", *Journal of the Operational Research Society*, **60**, pp. 1708-1718 (2009).
3. Mohtashami, A., Tavana, M., Santos-Arteaga, F. and Fallahian-Najafabadi, A. "A novel nulti-objective meta-heuristic model for solving cross-docking scheduling problems", *Applied Soft Computing*, **31**, pp. 30-47 (2015).
4. Vahdani, B. and Zandieh, M. "Scheduling trucks in cross-docking systems: Robust meta-heuristics", *Computers and Industrial Engineering*, **58**, pp. 12-24 (2010).
5. Konur, D. and Golias, M.M. "Cost-stable truck scheduling at a cross-dock facility with unknown truck arrivals: A meta-heuristic approach", *Transportation Research Part E: Logistics and Transportation Review*, **49**(1), pp. 71-91 (2013).
6. Maknoon, M.Y. and Baptiste, P. "Cross-docking: Increasing platform efficiency by sequencing incoming and outgoing semi-trailers", *International Journal of Logistics: research and Applications*, **12**, pp. 249-261 (2009).
7. Boysen, N., Fliedner, M. and Scholl, A. "Scheduling inbound and outbound trucks at cross docking terminals", *OR Spectrum*, **32**, pp. 135-161 (2010).
8. Sadykov, R. "Scheduling incoming and outgoing trucks at cross docking terminals to minimize the storage cost", *Annals of Operations Research*, **201**, pp. 423-440 (2012).
9. Boysen, N. "Truck scheduling at zero-inventory cross docking terminals", *Computers and Operations Research*, **37**, pp. 32-41 (2010).
10. Lee, K., Kim, B.S. and Joo, C.M. "Genetic algorithms for door-assigning and sequencing of trucks at distribution centers for the improvement of operational performance", *Expert Systems with Applications*, **39**, pp. 12975-12983 (2012).
11. Joo, C.M. and Kim, B.S. "Scheduling compound trucks in multi-door cross-docking terminals", *International Journal of Advanced Manufacturing Technology*, **64**, pp. 977-988 (2013).

12. Li, Y., Lim, A. and Rodrigues, B. "Cross docking: Just in time scheduling with time windows", *Journal of the Operational Research Society*, **55**(12), pp. 1342-1351 (2004).

13. Liao, T.W., Egbelu, P.J. and Chang, P.C. "Two hybrid differential evolution algorithms for optimal inbound and outbound truck sequencing in cross docking operations", *Applied Soft Computing*, **12**, pp. 3683-3697 (2012).

14. Chen, F. and Song, K.L. "Minimizing makespan in two-stage hybrid cross-docking scheduling problem", *Computers and Operations Research*, **36**(6), pp. 2066-2073 (2009).

15. Alpan, G., Larbi, R. and Penz, B. "A bounded dynamic programming approach to schedule operations in a cross docking platform", *Computers and Industrial Engineering*, **60**, pp. 385-396 (2011).

16. Mohtashami, A. "A novel dynamic genetic algorithm-based method for vehicle scheduling in cross docking systems with frequent unloading operation", *Computers and Industrial Engineering,* **90**, pp. 221-240 (2015).

17. Mohtashami, A. "Scheduling trucks in cross docking systems with temporary storage and repetitive pattern for shipping trucks", *Applied Soft Computing*, **36**, pp. 468-486 (2015).

18. Mohtashami, A. and Fallahian, A. "Scheduling trucks in supply chain with considering cross docking using meta-heuristics", *Industrial Management Studies,* **31**, pp. 55-84 (2013).

19. Ladier, A.L. and Alpan, G. "Cross-docking operations: Current research versus industry practice", *Omega*, **62**, pp. 145-162 (2016).

20. Vahdani, B., Soltani, R. and Zandieh, M. "Scheduling the truck holdover recurrent dock cross-dock problem using robust meta-heuristics", *International Journal of Advanced Manufacturing Technology*, **46**, pp. 769-783 (2010).

21. Madani-Isfahani, M., Tavakkoli-Moghaddam, R. and Naderi, B. "Multiple cross-docks scheduling using two meta-heuristic algorithms", *Computers and Industrial Engineering,* **74**, pp. 129-138 (2014).

22. Soltani, R. and Sadjadi, S.J. "Scheduling trucks in cross-docking systems: A robust meta-heuristics approach", *Transportation Research Part E,* **46**, pp. 650-666 (2010).

23. Mohtashami, A. "A new hybrid method for buffer sizing and machine allocation in unreliable production and assembly lines with general distribution time-dependent parameters", *International Journal of Advanced Manufacturing Technology*, **74**, pp. 1577-1593 (2014).

24. Eberhart, R. and Kennedy, J. "A new optimizer using particle swarm theory", In: *Proceeding of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, pp. 39-43 (1995).