

یک الگوریتم تفاضل تکاملی براساس رویه‌ی نمایش مبتنی بر اولویت برای بازطراحی زنجیره‌ی تأمین حلقه‌بسته با رویکرد بهینه‌سازی استوار فازی

محمدسعید اتابکی (دانشجوی دکتری)

محمد محمدی* (دانشیار)

گروه مهندسی صنایع، دانشکده فنی و مهندسی، دانشگاه خوارزمی

در این مقاله با در نظر گرفتن سیاست جریمه و پاداش، به ارائه‌ی یک مدل برنامه‌ریزی غیرخطی برای تعیین سطح بازیابی و مکان مراکز جمع‌آوری و بازطراحی شبکه‌ی حمل‌ونقل در یک زنجیره‌ی تأمین حلقه‌بسته می‌پردازیم. پارامترهای مدل به صورت اعداد فازی در نظر گرفته می‌شوند. همچنین برای افزایش تاب‌آوری تصمیم اتخاذی در برابر عدم قطعیت، از یک روش بهینه‌سازی استوار استفاده می‌شود. یک الگوریتم تفاضل تکاملی با پارامترهای تنظیم شده با روش پاسخ سطح برای حل مسائل با ابعاد واقعی به کار گرفته می‌شود. الگوریتم پیشنهادی بر اساس رویه‌ی نمایش مبتنی بر اولویت و یک فرایند رمزنگاری کارا طراحی می‌شود. برای ارزیابی عملکرد الگوریتم پیشنهادی از نرم‌افزار گمز در مسائل کوچک و مقایسه با الگوریتم ژنتیک در مسائل بزرگ و به کارگیری شاخص‌هایی چون بهترین جواب، میانگین جواب‌ها، میانگین درصد انحراف نسبی، و روش رتبه‌بندی شطرنج استفاده می‌شود. مقاله دربرگیرنده‌ی نوآوری‌هایی در مدل‌سازی، نمایش جواب، رمزنگاری، و ارزیابی عملکرد است.

واژگان کلیدی: زنجیره‌ی تأمین حلقه‌بسته، بازیابی، الگوریتم تفاضل تکاملی، روش پاسخ سطح، رتبه‌بندی شطرنج.

std_atabaki@khu.ac.ir
mohammadi@khu.ac.ir

۱. مقدمه

توزیع محصول را در نظر گرفته است و پیشنهادی درباره‌ی جمع‌آوری محصولات برگشتی، تعمیر و بازسازی، بازیافت، یا دفع آن‌ها ندارد. زنجیره‌ی تأمین معکوس در مقابل تعریف زنجیره‌ی تأمین مستقیم که تنها به جریان مواد و محصولات از سمت تأمین‌کننده به مشتری نهایی می‌پردازد، قرار دارد. بر اساس شورای اجرایی لجستیک معکوس آمریکا، لجستیک یا زنجیره‌ی معکوس به صورت زیر تعریف می‌شود:

فرایند برنامه‌ریزی، اجرا، و کنترل کارا و اثربخش جریان مواد، قطعات نیم‌ساخته، محصولات نهایی و اطلاعات مرتبط با آن‌ها از نقطه‌ی مصرف به نقطه‌ی مبدأ باهدف ایجاد ارزش مجدد یا دفع و انهدام مناسب.^[۲] شرکت‌هایی مانند زراکس، کداک و کاترپیلار در سال‌های اخیر با توجه به مفهوم لجستیک معکوس بر فعالیت‌های بازیابی و بازسازی تمرکز کرده‌اند و موفقیت‌های قابل توجهی در این زمینه به دست آورده‌اند.^[۳] ملاحظه‌ی هر دو جریان‌های مستقیم و معکوس به‌طور هم‌زمان منجر به تولید مفهوم زنجیره‌ی تأمین حلقه‌بسته (CLSC)^۱ می‌شود. در یک CLSC در

در سال‌های اخیر شرکت‌ها برای کاهش هزینه‌ها و متعاقباً افزایش سود، به مدیریت یکپارچه‌ی زنجیره تأمین روی آورده‌اند. مدیریت یکپارچه‌ی زنجیره‌ی تأمین با جلوگیری از زیربهنیگی ناشی از طراحی جداگانه در مقایسه با سیستم‌های سنتی به افزایش کارایی می‌انجامد. در سیستم‌های سنتی هر یک از طرف‌ها درصدد دستیابی به اهداف خود هستند که گاهی این اهداف می‌تواند متعارض با اهداف طرف‌های دیگر فعال در زنجیره باشد. این درحالی است که مدیریت یکپارچه‌ی زنجیره‌ی تأمین به دنبال افزایش سود کل زنجیره است. در گذشته، تلقی از زنجیره‌ی تأمین عمده‌تاً زنجیره‌ی تأمین مستقیم بود. بر اساس تعریف، زنجیره‌ی تأمین مستقیم شبکه‌ی یکپارچه از فرایندها و منابع است که مسئولیت تهیه‌ی مواد اولیه، تبدیل این مواد به محصولات میانی و نهایی، و توزیع محصولات نهایی به مشتریان را بر عهده دارد.^[۱] همان‌طور که مشخص است این تعریف تنها مسیر تولید و

* نویسنده مسئول

تاریخ: دریافت ۱۳۹۵/۲/۲۸، اصلاحیه ۱۳۹۵/۷/۲۱، پذیرش ۱۳۹۵/۸/۲۳.

DOI:10.24200/J65.2018.5547

۲. پیشینه‌ی پژوهش

برای بررسی جامع، پژوهش‌های پیشین در سه دسته مورد ارزیابی قرار می‌گیرد. دسته‌ی اول به بررسی الزامات قانونی در زنجیره‌های تأمین حلقه‌بسته می‌پردازد؛ در دسته دوم زنجیره‌های تأمین فازی استوار بررسی می‌شود و دسته‌ی سوم به روش‌های حل و الگوریتم تفاضل تکاملی می‌پردازد.

۱.۲. الزامات قانونی

الزامات قانونی یکی از سطوح کلان در ارزیابی فضای کسب‌وکار کشورهاست که تأثیر زیادی بر استراتژی‌های فعالان اقتصادی دارد. از آنجا که دولت‌ها یکی از ذی‌نفعان زنجیره‌های تأمین معکوس و حلقه‌بسته شناخته می‌شوند، پاداش مستقیم یا به صورت یارانه در کنار جریمه را به‌عنوان یکی از رویکردهای عمومی برای سمت‌وسو دادن به فعالیت‌های شرکت‌ها و بنگاه‌های اقتصادی در نظر می‌گیرند. مدل‌سازی الزامات قانونی و تأثیرات آن بر CLSC یکی از موضوعات تحقیقی است که پژوهشگران با سرعت فزاینده‌ی در حال بررسی آن هستند.^[۱۱] وبستر و میترا^[۱۲] به بررسی تأثیر یارانه‌ی بازسازی بر تولیدکننده و بازتولیدکننده پرداختند. پلامبیک و وانگ^[۱۳] به این نتیجه رسیدند که سیاست ایجاد هزینه روی دفع موجب انگیزش تولیدکنندگان به طراحی محصولات بازافت‌شدنی می‌شود. آکسن و همکاران^[۱۴] به توسعه‌ی مدل‌هایی برای راهنمایی دولت‌ها در پرداخت یارانه به تولیدکنندگان پرداختند. آتاسو و همکاران^[۱۵] به بحث پیرامون شرایط مؤثر دولت‌ها در قانون‌گذاری بازایی و بازسازی ضایعات تجهیزات الکترونیکی و الکتریکی پرداختند. رحمان و سابرامیان^[۱۶] در تحقیق خود به این نتیجه رسیدند که قوانین دولتی، مشوق‌ها، و تقاضای مشتریان از دلایل عمده‌ی تحریک شرکت‌ها برای بازایی قطعات کامپیوتری‌اند. ژاکوبس و سابرامیان^[۱۷] یک مدل دوسطحی برای بررسی تأثیر اختیار بازایی بر انگیزه‌ی اجرای طرح‌های بازایی توسعه دادند. بررسی‌های گوپندان و همکاران^[۱۸] نشان داد از دلایل اصلی پرداختن زنجیره‌های تأمین حلقه‌بسته به لجستیک معکوس کسب ارزش از محصولات برگشتی و مواد بازیافتی و معدوم‌سازی ایمن ضایعات طبق الزامات قانونی است. وانگ و همکاران^[۱۹] به بررسی اجرای سازوکار پاداش - جریمه توسط دولت‌ها برای ایجاد انگیزه برای فعالیت‌های بازافت پرداختند. آن‌ها مدل‌های CLSC را در حالت‌های تولیدکننده - پیشرو و جمع‌آوری کننده - پیشرو مورد کنکاش قرار دادند و اظهار داشتند که افزایش مقدار پاداش - جریمه موجب بهبود بازایی و کاهش قیمت محصول می‌شود.

۲.۲. زنجیره‌های تأمین فازی استوار

با توجه به این‌که در بسیاری از مسائل بهینه‌سازی تعیین قطعی پارامترها بسیار دشوار است محققان تلاش‌های زیادی برای غلبه بر عدم قطعیت پارامترها و در نتیجه نزدیک شدن به فضای واقعی داشته‌اند. دو رویکرد عمده برای غلبه بر عدم قطعیت به‌کاربردن روش‌های احتمالی و منطقی فازی است. اخیراً رویکرد جدیدی با عنوان بهینه‌سازی استوار نیز مورد توجه محققان قرار گرفته است. شروع توسعه‌ی بهینه‌سازی استوار به مقاله‌ی سوپستر^[۱۹] در سال ۱۹۷۳ برمی‌گردد که یک روش برنامه‌ریزی استوار بدینانه برای برخورد با مسائل برنامه‌ریزی خطی غیردقیق ارائه داد. پس از آن در حوزه‌ی برنامه‌ریزی ریاضی فازی، اونگوچی و ساکاو^[۲۰] روش‌هایی با رویکرد کمینه‌سازی بهینه‌ی تأسّف ارائه دادند. مولوی و همکاران^[۲۱] یک روش برنامه‌ریزی استوار متعطف بر پایه‌ی مدل برنامه‌ریزی تصادفی سناریومحور توسعه دادند که پایه‌ی

کنار توجه به جریان مواد و محصولات از تأمین‌کننده به مشتری، جریان برگشت محصولات از مشتریان و فعالیت‌های بازافت، تعمیر، بازسازی، و دفع ایمن نیز در نظر گرفته می‌شود. چنین سیستمی در صنایع شیشه،^[۲] تجهیزات الکتریکی و الکترونیکی،^[۵] باتری،^[۶] اتومبیل،^[۷] رایانه،^[۸] نساجی^[۹] و غیره می‌تواند کاربرد داشته باشد. از آنجا که با رشد فزاینده‌ی فناوری و کوتاه‌شدن عمر مفید محصولات، روزانه مقادیر زیادی آهن، مس، پارچه، پلاستیک، چوب و مواد اولیه‌ی دیگر در صنایع نام‌برده به‌خصوص در کشورهای درحال توسعه و جوامعی با فرهنگ مصرف‌گرایی، مصرف می‌شود، استفاده‌ی مجدد از این مواد می‌تواند انگیزه‌ی اصلی شرکت‌ها در یک سطح و دولت‌ها در سطح بالاتر برای بهره‌گیری از مزایای CLSC باشد. علاوه بر این، آلودگی محیط‌زیست موجب فشار بر صنایع یا تشویق آن‌ها توسط دولت‌ها برای جمع‌آوری و بازافت محصولات برگشتی می‌شود. جمع‌آوری و بازافت محصولات برگشتی هم از دیدگاه دفع ایمن ضایعات و هم از دیدگاه کاهش آلودگی به دلیل کاهش مصرف انرژی در تولید مواد اولیه، موجب آسیب کمتر به محیط‌زیست می‌شود. این دلایل باعث شده است شرکت‌ها، حتی در کشورهایی با اقتصاد مبتنی بر استخراج که تاکنون تنها به مدیریت زنجیره‌ی تأمین مستقیم می‌پرداختند به سرمایه‌گذاری برای ساخت مراکز بازافت و به‌کارگیری رویکرد زنجیره‌ی تأمین حلقه‌بسته متمایل شوند. در نظرگرفتن یک یا چند مرکز بازافت منجر به تغییر در ساختار زنجیره می‌شود و نیاز به بازطراحی شبکه لجستیک را ایجاب می‌کند. این بازطراحی شامل تعیین مکان مراکز بازافت و اصلاح شبکه‌ی حمل‌ونقل برای توزیع محصول و جمع‌آوری محصولات برگشتی است. نکته‌ی مهم در ساخت مراکز بازافت، کسب دانش و فناوری بازافت محصولات برگشتی است. دست‌یابی به فناوری بازافت نیاز به سرمایه‌گذاری نسبتاً بالایی دارد. این امر موجب صرف‌نظرکردن شرکت‌هایی با توان اقتصادی پایین از ایجاد مراکز بازافت می‌شود. به این دلیل دولت‌ها با تشویق‌ها یا جریمه‌هایی به جمع‌آوری و بازافت محصولات شرکت‌ها ورود می‌کنند. شرکت‌ها با بهبود کیفی و کمی بازافت می‌توانند یارانه بیشتری دریافت کنند یا از پرداخت مالیات و سایر عوارض معاف شوند.

یکی از روش‌های متداول برای بازطراحی شبکه‌ی زنجیره‌ی تأمین استفاده از مدل‌های برنامه‌ریزی ریاضی است. به دلیل وجود نرم‌افزارهای بهینه‌سازی مختلف که دست‌یابی به جواب بهینه‌ی کلی را تضمین می‌کنند، محققان سعی می‌کنند از مدل‌های برنامه‌ریزی خطی یا خطی آمیخته عدد صحیح استفاده کنند. از آنجا که مسائل دنیای واقعی اغلب پیچیده و مشکل هستند، نمی‌توانند فقط با به‌کارگیری رویکردهای برنامه‌ریزی خطی ساده مدل شوند.^[۱۹] استفاده از مدل‌های خطی گاهی موجب غفلت از برخی وجوه زنجیره‌ی تأمین می‌شود.

در این مقاله به ارائه‌ی یک مدل برنامه‌ریزی غیرخطی CLSC برای اخذ تصمیمات مکان مراکز بازافت، تعیین مقادیر حمل‌ونقل بین تسهیلات، و سطح بازافت می‌پردازیم. بیشتر مقالات گذشته در حوزه‌ی زنجیره‌ی تأمین حلقه‌بسته، با در نظرگرفتن پارامتری به‌عنوان درصد بازافت، به طراحی شبکه‌ی زنجیره‌ی تأمین از ابتدای تأسیس می‌پردازند. از آنجا که در کشور ما و سایر کشورها، شرکت‌های زیادی وجود دارند که مدیریت زنجیره‌ی تأمین مستقیم را اجرا کرده‌اند و اکنون با توجه به مزایای CLSC، قصد به‌کارگیری این رویکرد را دارند، در این مقاله به بازطراحی یک زنجیره‌ی تأمین با ورود مراکز بازافت می‌پردازیم.

برای نزدیک‌تر شدن به فضای واقعی غیرقطعی مسائل زنجیره‌ی تأمین، پارامترها به صورت فازی در نظر گرفته می‌شود. همچنین با به‌کارگیری رویکرد بهینه‌سازی استوار سعی می‌شود تصمیم اتخاذشده در برابر عدم قطعیت تاب‌آوری داشته باشد.

جست‌وجوی ممنوع (TS)^۵ و بهینه‌سازی ازدحام ذرات (PSO)^۶ نشان دادند.

۴.۲. شکاف تحقیقاتی و نوآوری تحقیق

نتایج بررسی تحقیقات پیشین حاکی از آن است که بیشتر تحقیقات در حوزه زنجیره تأمین به طراحی یک مدل زنجیره تأمین از ابتدا می‌پردازند درحالی‌که در دنیای واقعی ممکن است شرکتی پس از چندین سال مدیریت زنجیره تأمین مستقیم، قصد در نظر گرفتن محصول برگشتی را در زنجیره خود داشته باشد که در این صورت نیاز به بازطراحی شبکه‌ی لجستیک دارد. همچنین بررسی‌ها نشان می‌دهد در اکثر تحقیقات CLSC، درصد بازبایی به صورت یک پارامتر و نه متغیر مشخص است. همچنین تعداد تحقیقات اندکی با در نظر گرفتن عدم قطعیت فازی پارامترها، از روش بهینه‌سازی استوار بهره برده‌اند. علاوه بر این، به دلیل پیچیدگی مسائل زنجیره تأمین، ارائه‌ی روش حل کارا برای حل مسائل با ابعاد واقعی یکی از چالش‌های محققان است.

بر این اساس، نوآوری‌های تحقیق پیش‌رو شامل موارد زیر است:

- ارائه‌ی مدل برنامه‌ریزی غیرخطی برای تعیین سطح بازیافت، شبکه‌ی حمل‌ونقل و محل مراکز جمع‌آوری در یک زنجیره تأمین حلقه‌بسته؛
- در نظر گرفتن عدم قطعیت پارامترها با استفاده از منطق فازی و بهینه‌سازی استوار؛
- توسعه‌ی یک الگوریتم تفاضل تکاملی کارا با پارامترهای تنظیم‌شده برای حل مسائل با ابعاد واقعی؛
- ارائه‌ی یک روش نمایش جواب و رمزنگاری بر اساس روش مبتنی بر اولویت با کارایی بالاتر از روش استاندارد مبتنی بر اولویت؛
- به‌کارگیری روش رتبه‌بندی شطرنج برای ارزیابی جواب‌های ایجادشده به‌وسیله‌ی الگوریتم‌های فراابتکاری.

بر این اساس تفاوت تحقیق پیش‌رو با برخی تحقیقات مرتبط شامل موارد زیر است:

در تحقیق بیمون^[۲۳] تأثیر عوامل زیست‌محیطی بر طراحی زنجیره تأمین و دست‌یابی به زنجیره تأمین سبز، تفاوت زنجیره‌های تأمین سنتی و مدرن، چالش‌های توسعه‌ی مدل‌های زنجیره تأمین، شاخص‌های ارزیابی زنجیره‌های تأمین و مسائلی از این دست به‌طور کیفی مورد تحلیل قرار گرفته اما مدل‌سازی کمی انجام نگرفته است. در پژوهش کریک و دیگران^[۲۴] تأثیر ساختار محصول مانند تفکیک‌پذیری^۷ اجزای تشکیل‌دهنده‌ی آن بر طراحی زنجیره حلقه‌بسته مورد تحلیل و بررسی قرار گرفته است. همچنین شرایط یک زنجیره تأمین حلقه‌بسته بررسی شده است. با وجود این، تعیین بهینه‌ی سطح بازبایی مورد کنکاش قرار نگرفته، تأثیر عدم قطعیت بررسی نشده و الگوریتم‌های فراابتکاری موضوعیت نداشته است. متغیرها در تحقیقات جهرمی و دیگران^[۲۵] و پیشوایی و دیگران^[۲۶] بر طراحی شبکه‌ی حمل‌ونقل و تعیین محل تسهیلات متمرکز است، درحالی‌که در پژوهش حاضر یکی از متغیرهای اصلی، تعیین سطح بازبایی است و همچنین ساختار شبکه‌ها با هم متفاوت است. نوع عدم قطعیت در نظر گرفته شده برای این پژوهش با کارهای نام‌برده و همچنین نوع روش بهینه‌سازی استوار نیز متفاوت است. در تحقیقات مذکور از روش دقیق برای حل مسئله استفاده شده است، درحالی‌که در پژوهش پیش‌رو روش فراابتکاری برای حل مسائل با ابعاد بزرگ پیشنهاد شده است. تمرکز مقاله‌ی کانان و دیگران^[۲۷] بر طراحی شبکه‌ی حمل‌ونقل و تعیین محل تسهیلات است درحالی‌که در پژوهش حاضر سطح بازبافت جزء متغیرهای تصمیم است. همچنین در پژوهش مذکور پارامترها به صورت

بسیاری از روش‌های بعدی شد. بنتال و نمیرفسکی^[۲۳،۲۴] با توسعه‌ی روش‌های پیشین روشی برای مسائل برنامه‌ریزی خطی غیرقطعی با مجموعه‌های عدم قطعیت محدب ارائه دادند. پس‌از آن پیشوایی و همکاران^[۲۴] به توسعه‌ی کاربرد منطق فازی در فضای برنامه‌ریزی امکانی پرداختند. همچنین پیشوایی و خلف^[۲۵] به ارائه‌ی مدل‌های جدید بهینه‌سازی فازی استوار پرداختند.

با توسعه‌ی مدل‌های فازی استوار، محققان سعی کردند از آن‌ها در مسائل زنجیره تأمین نیز بهره‌مند شوند. پیشوایی و همکاران^[۲۴] یک مدل برنامه‌ریزی امکانی استوار برای یک زنجیره تأمین توسعه دادند. درحالی‌که هدف اول مدل آن‌ها کاهش هزینه‌های سیستم بود، هدف دوم در پی افزایش پایداری زنجیره از طریق بهینه‌سازی مسئولیت‌پذیری اجتماعی بود. زو^[۲۶] با در نظر گرفتن یک مدل زنجیره تأمین از منطق فازی و بهینه‌سازی خطی استوار برای غلبه بر عدم قطعیت تقاضای وابسته به قیمت استفاده کرد. بای و لیو^[۲۷] یک مدل بهینه‌سازی استوار برای یک زنجیره تأمین سه سطحی با پارامترهای فازی برای بهینه‌سازی ارزش در معرض ریسک^۲ به کار بردند. قدرتمنا و همکاران^[۲۸] از دو رویکرد برنامه‌ریزی آرمانی فازی و استوار برای حل مسئله‌ی چندهدفه‌ی مکان‌یابی - تخصیص هاب در فضای زنجیره تأمین استفاده کردند. وحدانی و محمدی^[۲۹] یک مدل بهینه‌سازی دوهدفه برای طراحی شبکه‌ی زنجیره تأمین حلقه‌بسته تحت عدم قطعیت توسعه دادند و برای حل مدل از یک رویکرد ترکیبی بر پایه‌ی برنامه‌ریزی بازه‌ی، برنامه‌ریزی احتمالی، بهینه‌سازی استوار، و برنامه‌ریزی چندهدفه‌ی فازی استفاده کردند. طلائی و همکاران^[۳۰] یک مدل فازی استوار برای کمی‌سازی هزینه‌های CLSC دو هدفه با رویکرد توجه به مسائل زیست‌محیطی ارائه دادند و با روش اسپیلون محدود آن را حل کردند.

۳.۲. الگوریتم تفاضل تکاملی در زنجیره تأمین

الگوریتم تفاضل تکاملی (DE)^۳ یک الگوریتم فراابتکاری مبتنی بر جمعیت است که اولین بار توسط پرایس و استرم^[۳۱] در سال ۱۹۹۵ برای حل مسائل بهینه‌سازی پیوسته ارائه شد و به دلیل کارایی بالای آن، به سرعت برای حل مسائل مختلف مانند زمان‌بندی^[۳۲] برنامه‌ریزی تولید^[۳۳] کنترل موجودی^[۳۴] و زنجیره تأمین به کار گرفته شد و نسخه‌های مختلفی از آن برای مسائل گسسته و پیوسته توسعه داده شده است. چان و همکاران^[۳۵] الگوریتم DE را برای تعیین اندازه‌ی سفارش/تولید باهدف کمی‌سازی هزینه‌ها در یک زنجیره تأمین شامل تأمین‌کننده و توزیع‌کننده به کار بردند. بابو و گوچارادیل^[۳۶] الگوریتم DE چندهدفه را برای حل زنجیره تأمین سه‌سطحی شامل تأمین‌کننده، تولیدکننده، و مشتری به کار گرفتند. دینگ و همکاران^[۳۷] با در نظر گرفتن یک زنجیره تأمین معکوس دوسطحی، الگوریتم DE را برای حل مدل غیرخطی آمیخته به کار بردند. زو و همکاران^[۳۸] با به‌کارگیری مفهوم مهاجرت عشایر، الگوریتم DE بهبودیافته‌ی برای حل مسائل زنجیره تأمین ارائه دادند. فاتریاس و شیمیزو^[۳۹] الگوریتم DE را برای تجزیه و تحلیل مسئله‌ی کنترل موجودی مرور دوره‌ی در یک زنجیره تأمین دوسطحی به کار گرفتند. طیار و همکاران^[۴۰] الگوریتم DE چندهدفه را برای ایجاد جواب‌های پارتو برای یک زنجیره تأمین با اهداف چندگانه به کار بردند. لیکنز و ونداله^[۴۱] الگوریتم DE استاندارد را برای حل مسئله‌ی اندازه‌ی انباشته در یک مدل زنجیره تأمین غیرخطی عدد صحیح با پارامترهای احتمالی برای بهبود زمان تحویل و هزینه‌های راه‌اندازی به کار بردند. بیلدیز و اولتان^[۴۲] یک الگوریتم بر پایه‌ی DE را برای بهینه‌سازی یک مدل زنجیره تأمین به‌کار بردند و کارایی و دقت بالاتر DE را در مقایسه با الگوریتم‌های ژنتیک (GA)^۴،

قطعی در نظر گرفته شده‌اند و الگوریتم ژنتیک برای حل مسائل با ابعاد واقعی پیشنهاد شده است اما رویه‌ی نمایش جواب ارائه نشده است. علاوه بر این روش‌های ارزیابی عملکرد الگوریتم فراابتکاری در این تحقیق متفاوت از مقاله کانان و دیگران^[۲۷] است.

۳. تعریف مسئله

شرکتی را در نظر بگیرید که با داشتن چندین مرکز تولید در مکان‌های مختلف به تولید محصولی برای ارائه به مشتریان خود می‌پردازد. با توجه به سیاست‌های دولت در جمع‌آوری، دفع ایمن و بازیافت محصولات دست‌دوم با هدف حفظ محیط‌زیست و کاهش منابع مصرفی و همچنین توجه مدیران شرکت به کاهش هزینه‌ها از طریق استفاده‌ی مجدد از محصولات برگشتی، شرکت درصدد عقد قرارداد با شرکتی دیگر برای احداث یک یا چند مرکز جمع‌آوری محصولات برگشتی است. این مراکز وظیفه‌ی جمع‌آوری محصول برگشتی، آزمایش، بازیافت، و دفع را بر عهده دارند. برای استفاده از مزیت‌های مدیریت یکپارچه، قرارداد با شرکت بازیافت برحسب شراکت در سود کل زنجیره است. شرکت بازیافت نیاز به سرمایه‌گذاری برای دست‌یابی به فناوری بازیافت دارد. هرچه قدر سرمایه‌گذاری بالاتر باشد، درصد بیشتری از محصولات برگشتی بازیافت می‌شود. اگر B درصد بازیافت محصول باشد، همانند مقاله‌ی وانگ و همکاران^[۱۱] هزینه‌ی سرمایه‌گذاری یک مرکز جمع‌آوری را به صورت $\lambda * B^t$ در نظر می‌گیریم که در آن λ پارامتر مقیاس هزینه‌ی ایجاد یک مرکز جمع‌آوری است. بر این اساس با افزایش سطح بازیافت، هزینه‌ی ایجاد یک مرکز جمع‌آوری به‌طور فزاینده‌ی افزایش می‌یابد. دولت سطح هدفی برابر β در نظر گرفته است و برای مقادیر درصد بیشتر از β پاداش و برای مقادیر کمتر جریمه در نظر گرفته است. شرکت در پی تعیین سطح بازیافت به گونه‌یی است که هزینه‌ی کل زنجیره کمینه شود. تعیین تعداد و مکان مراکز بازیافت از جمله تصمیمات راهبردی است که باید اتخاذ شود. همچنین تعیین مقادیر حمل‌ونقل محصول از مراکز تولید به مشتریان، محصول برگشتی از مشتریان به مراکز بازیافت و مواد بازیافتی از مراکز بازیافت به مراکز تولید به‌عنوان تصمیمات سطح فنی باید اتخاذ شود. بر این اساس فرض‌های زیر برای مدل‌سازی مسئله در نظر گرفته شده است:

- مکان مراکز تولید و مشتریان مشخص و ثابت است.
- مکان بالقوه مراکز جمع‌آوری مشخص است.
- تمام محصولات برگشتی از مشتریان باید جمع‌آوری شود.
- بیشینه‌ی تعداد مراکز جمع‌آوری قابل ساخت محدود است.
- کیفیت محصول ساخته‌شده از مواد بازیافتی و مواد نو یکسان است.

۴. مدل‌سازی

۱.۴. اندیس‌ها و پارامترها

i : اندیس مراکز تولید $i = 1, 2, \dots, I$ ؛
 j : اندیس مشتریان $j = 1, 2, \dots, J$ ؛
 k : اندیس مراکز جمع‌آوری $k = 1, 2, \dots, K$ ؛
 d_j : تقاضای مشتری j ؛

m_i : هزینه‌ی تولید یک واحد محصول در مرکز تولید i با به‌کارگیری مواد نو؛
 r_i : هزینه‌ی تولید یک واحد محصول در مرکز تولید i با به‌کارگیری مواد بازیافت؛
 v : پاداش یا جریمه در نظر گرفته‌شده برای یک درصد افزایش یا کاهش سطح بازیافتی

از سطح هدف‌گذاری شده؛
 f_i : هزینه‌ی ثابت تولید مرکز تولید i ؛
 β : سطح بازیافت هدف‌گذاری شده توسط سیاست‌گذار؛
 TII_{ij} : هزینه‌ی حمل یک واحد محصول از مرکز تولید i به مشتری j ؛
 TKI_{jk} : هزینه‌ی حمل یک واحد محصول برگشتی از مشتری j به مرکز جمع‌آوری k ؛
 TKI_{ki} : هزینه‌ی حمل یک واحد محصول 100 درصد بازیافت‌شده از مرکز جمع‌آوری k به مرکز تولید i ؛
 a_k : هزینه‌ی بازیافت کامل یک واحد محصول برگشتی در مرکز جمع‌آوری k ؛
 e_k : هزینه‌ی دفع یک واحد محصول برگشتی در مرکز جمع‌آوری k ؛
 λ : پارامتر مقیاس هزینه‌ی ایجاد یک مرکز جمع‌آوری؛
 c_k : هزینه‌ی ثابت مورد نیاز برای ساخت مرکز جمع‌آوری k ؛
 n : بیشینه‌ی تعداد مراکز جمع‌آوری قابل ساخت؛
 ca_i : بیشینه‌ی ظرفیت مرکز تولید i ؛
 cr_k : بیشینه‌ی ظرفیت مرکز جمع‌آوری k ؛
 α : درصد بازگشت محصول؛
 M : یک عدد بزرگ.

۲.۴. متغیرها

B : سطح بازیافت؛
 QIJ_{ij} : مقدار محصول انتقال‌یافته از مرکز تولید i به مشتری j ؛
 QJK_{jk} : مقدار محصول انتقال‌یافته از مشتری j به مرکز جمع‌آوری k ؛
 QKI_{ki} : مقدار محصول انتقال‌یافته از مرکز جمع‌آوری k به مرکز تولید i ؛
 X_k : برابر یک است اگر مرکز جمع‌آوری k احداث شود؛ در غیر این صورت صفر است؛
 Y_i : برابر یک است اگر در مرکز تولید i تولید انجام گیرد؛ در غیر این صورت صفر است.

۳.۴. مدل برنامه‌ریزی غیرخطی

بر اساس نمادهای فوق، مدل برنامه‌ریزی غیرخطی مسئله‌ی طراحی شبکه‌ی لجستیک مستقیم/معکوس ارائه می‌شود. تابع هدف شامل کمینه‌سازی مجموع روابط ۱ تا ۷ است.

$$\sum_j \sum_i TII_{ij} QIJ_{ij} + \sum_k \sum_j TJK_{jk} QJK_{jk} + \sum_i \sum_k TKI_{ki} QKI_{ki} B \quad (1)$$

$$\sum_k c_k X_k + \sum_k \lambda B^t X_k \quad (2)$$

$$\sum_j \sum_k a_k B QJK_{jk} \quad (3)$$

$$\sum_j \sum_k e_k (\lambda - B) QJK_{jk} \quad (4)$$

$$\sum_i \sum_k QKI_{ki} (Br_i + (\lambda - B) m_i) + \sum_i m_i \left(\sum_j QIJ_{ij} - \sum_k QKI_{ki} \right) \quad (5)$$

حالات ممکن پارامترهای دارای عدم قطعیت، شدنی باقی بماند. استواری بهینگی بدین مفهوم است که مقدار تابع هدف به ازای جواب استوار باید برای تمام (اکثریت قریب به اتفاق) حالات ممکن پارامترهای دارای عدم قطعیت، نزدیک به مقدار بهینه‌ی خود بوده یا به عبارت دیگر حداقل انحراف را از مقدار بهینه‌ی خود داشته باشد. بهینه‌سازی استوار به دنبال دست‌یابی به جواب‌های استوار برای یک مسئله‌ی بهینه‌سازی است و برای محافظت از تصمیم‌گیرنده در برابر ابهام پارامترها و عدم قطعیت به‌کار گرفته می‌شود. در این رویکرد تصمیم‌گیرنده به دنبال جواب‌های نزدیک به جواب بهینه است که با احتمال بالایی موجه باشند. این روش به تصمیم‌گیرنده اجازه می‌دهد بین عملکرد اقتصادی و قابلیت اطمینان سیستم توازن برقرار کند. [۲۸]

بر این اساس بهینه‌سازی استوار در مسائلی به‌کار گرفته می‌شود که نوعی عدم قطعیت در آن‌ها وجود داشته باشد. بر اساس تعریف مولا و همکاران [۲۹] عدم قطعیت در مسائل بهینه‌سازی به دو دسته‌ی کلی تقسیم می‌شود: انعطاف‌پذیری در اهداف و محدودیت‌ها (برنامه‌ریزی منعطف)؛ عدم قطعیت در داده‌ها (برنامه‌ریزی امکانی) که خود به دو نوع تصادفی^۱ و عدم قطعیت شناختی^{۱۱} طبقه‌بندی می‌شود. در این مقاله با در نظر گرفتن عدم قطعیت شناختی در داده‌ها، ضرایب تابع هدف، [۵۱،۵۰] تقاضا [۵۲،۵۱] و ظرفیت مراکز جمع‌آوری [۵۵-۵۲،۲۵] غیرقطعی و به صورت عدد فازی مثلی در نظر گرفته می‌شود. از آنجا که کارخانه‌ها موجود و در حال فعالیت هستند، ظرفیت آن‌ها مشخص و قطعی در نظر گرفته می‌شود. در این مقاله از رویکرد برنامه‌ریزی امکانی استوار ارائه‌شده توسط پیشوایی و خلف [۲۵] استفاده می‌شود. روش به‌کار گرفته‌شده برعکس روش‌های محافظه‌کارانه که سعی می‌کنند با هزینه‌ی بالا با عدم قطعیت مقابله کنند، به دنبال ایجاد تعادل منطقی بین استواری و هزینه‌های استواری است. بدین مفهوم که پارامترهای هزینه تعیین می‌کند که جواب نهایی به چه میزان عدم قطعیت را با صرف هزینه پوشش دهد.

مدل برنامه‌ریزی خطی زیر را با متغیرهای پیوسته و دودویی در نظر بگیرید:

$$\begin{aligned} & \text{Min } cx + fy \\ & \text{s.t. } Ax \geq d \\ & \quad Sx \leq Ny \\ & y = 0 \text{ or } 1, \quad x \geq 0 \end{aligned} \quad (16)$$

مدل ۱۶ می‌تواند یک مدل عمومی زنجیره‌ی تأمین باشد که در آن عبارت اول تابع هدف بیان‌گر هزینه‌های متغیر و عبارت دوم گویای هزینه‌های ثابت احداث تسهیلات است. محدودیت اول بیان‌گر محدودیت تقاضا و محدودیت دوم بیان‌گر محدودیت ظرفیت تسهیلات است. با فرض اینکه بردارهای c و f در تابع هدف و بردار d و ماتریس N در محدودیت‌ها دارای عدم قطعیت باشند و به صورت اعداد فازی مثلی باشند، بر اساس رویکرد ارائه‌شده توسط خیمز و همکاران [۵۶] و پیشوایی و ترابی [۵۷] مدل پایه برنامه‌ریزی امکانی آن به صورت زیر است:

$$\begin{aligned} & \text{Min } \left(\frac{c^{(1)} + c^{(2)} + c^{(3)}}{3} \right) x + \left(\frac{f^{(1)} + f^{(2)} + f^{(3)}}{3} \right) y \\ & \text{s.t. } Ax \geq \delta_1 \left(\frac{d^{(1)} + d^{(2)}}{2} \right) + (1 - \delta_1) \left(\frac{d^{(1)} + d^{(2)}}{2} \right) \\ & \quad Sx \leq \left[\delta_2 \left(\frac{N^{(1)} + N^{(2)}}{2} \right) + (1 - \delta_2) \left(\frac{N^{(1)} + N^{(2)}}{2} \right) \right] y \\ & y = 0 \text{ or } 1, \quad x \geq 0 \end{aligned} \quad (17)$$

همان‌طور که مشاهده می‌شود در این مدل از مقادیر انتظاری اعداد فازی در تابع

$$\sum_i f_i Y_i \quad (6)$$

$$100 * V(\beta - B) \quad (7)$$

رابطه‌ی ۱ بیان‌کننده هزینه‌های حمل‌ونقل (از مراکز تولید به مشتریان، از مشتریان به مراکز جمع‌آوری، از مراکز جمع‌آوری به مراکز تولید) است. رابطه‌ی ۲ هزینه‌ی ساخت مراکز جمع‌آوری با توجه به سطح باز یافت، رابطه‌ی ۳ هزینه‌ی باز یافت و رابطه‌ی ۴ هزینه‌ی دفع مواد اسقاط است. رابطه‌ی ۵ هزینه‌ی تولید با به‌کارگیری مواد نو و باز یافت، رابطه‌ی ۶ هزینه‌ی ثابت تولید و رابطه‌ی ۷ پاداش / جریمه مرتبط با سطح باز یافت را نشان می‌دهد. در رابطه‌ی ۸ از آنجا که کیفیت محصول ساخته‌شده از مواد باز یافتی و مواد نو یکسان است، هزینه‌ی تولید محصول بر اساس هزینه‌ی تولید با مواد نو، هزینه‌ی تولید با مواد باز یافتی و سطح باز یافت تعیین می‌شود. بر این اساس هزینه‌ی تولید برابر مجموع هزینه‌ی تولید با استفاده از ترکیب مواد نو و باز یافتی بر اساس سطح باز یافت و هزینه‌ی تولید با استفاده از مواد کاملاً نو است.

رابطه‌های ۸ تا ۱۵ محدودیت‌های مسئله را تعیین می‌کند.

$$\sum_i QIJ_{ij} \geq d_j \quad \forall j \quad (8)$$

$$\sum_i QKI_{ki} \leq \sum_j QJK_{jk} \quad \forall k \quad (9)$$

$$\sum_k QJK_{jk} = \alpha d_j \quad \forall j \quad (10)$$

$$\sum_j QIJ_{ij} \leq ca_i \quad \forall i \quad (11)$$

$$\sum_j QJK_{jk} \leq cr_k X_k \quad \forall k \quad (12)$$

$$\sum_j QIJ_{ij} \leq MY_i \quad (13)$$

$$\sum_k X_k \leq n \quad (14)$$

$$B \leq 1 \quad (15)$$

محدودیت ۸ تضمین‌کننده تأمین تقاضای مشتریان است. محدودیت ۹ مشخص‌کننده آن است که مقدار محصول انتقال یافته از یک مرکز جمع‌آوری به تمام مراکز تولید کمتر از مقدار محصول انتقال یافته از تمام مشتریان به آن مرکز جمع‌آوری است. رابطه ۱۰ جمع‌آوری تمام محصول برگشتی را تضمین می‌کند. رابطه ۱۱ بیان‌گر محدودیت ظرفیت مراکز تولید است. محدودیت ۱۲ بیان‌گر بیشینه ظرفیت مراکز جمع‌آوری است. محدودیت ۱۳ مشخص می‌کند که در صورتی که یک مرکز تولید، تولید داشته باشد باید به آن هزینه ثابت تعلق گیرد. محدودیت ۱۴ بیان‌گر بیشینه‌ی مراکز جمع‌آوری قابل احداث است. محدودیت ۱۵ مشخص می‌کند که سطح باز یابی نمی‌تواند بیشتر از ۱۰۰ درصد باشد.

۵. مدل فازی استوار

تصمیم استوار تصمیمی است که نسبت به عدم قطعیت محیط تاب بیاورد و عملکرد ناشی از آن حداقل نوسان را داشته باشد. یک جواب برای یک مسئله‌ی بهینه‌سازی، یک جواب استوار است اگر دارای استواری شدنی بودن^۸ و استواری بهینگی^۹ باشد. استواری شدنی بودن به این معناست که جواب باید برای تمام (اکثریت قریب به اتفاق)

$$\sum_j \sum_k \left(\frac{e_k^{(1)} + e_k^{(r)} + e_k^{(r)}}{3} \right) (\lambda - \pi) QJK_{jk} - \sum_j \sum_k \left(\frac{e_k^{(1)} + e_k^{(r)} + e_k^{(r)}}{3} \right) B (\lambda - \pi) QJK_{jk} \quad (22)$$

$$\sum_i \sum_k \left(\frac{r_i^{(1)} + r_i^{(r)} + r_i^{(r)}}{3} \right) B (\lambda - \pi) QKI_{ki} - \sum_i \sum_k \left(\frac{m_i^{(1)} + m_i^{(r)} + m_i^{(r)}}{3} \right) B (\lambda - \pi) QKI_{ki} + \sum_j \sum_i \left(\frac{m_i^{(1)} + m_i^{(r)} + m_i^{(r)}}{3} \right) (\lambda - \pi) QIJ_{ij} \quad (23)$$

$$\sum_i \left(\frac{f_i^{(1)} + f_i^{(r)} + f_i^{(r)}}{3} \right) (\lambda - \pi) Y_i \quad (24)$$

$$\frac{v}{\sigma_j} (\beta - B) \quad (25)$$

$$\pi \left[\sum_j \sum_i TIJ_{ij}^{(r)} QIJ_{ij} + \sum_k \sum_j TJK_{jk}^{(r)} QJK_{jk} \right] + \pi \left[\sum_i \sum_k TKI_{ki}^{(r)} QKI_{ki} B \right] + \pi \left[\sum_k c_k^{(r)} X_k + \sum_k \lambda^{(r)} B^T X_k + \sum_k \sum_j a_k^{(r)} QJK_{jk} \right] + \pi \left[\sum_j \sum_k e_k^{(r)} (\lambda - B) QJK_{jk} \right] + \pi \left[\sum_i \sum_k r_i^{(r)} QKI_{ki} B + \sum_i \sum_k m_i^{(r)} (\lambda - B) QKI_{ki} \right] + \pi \left[\sum_i m_i^{(r)} \left(\sum_j QIJ_{ij} - \sum_k QKI_{ki} \right) \right] + \pi \left[\sum_i f_i^{(r)} Y_i \right] \quad (26)$$

$$w_1 \left[\sum_j d_j^{(r)} - \delta_1 \left(\frac{\sum_j d_j^{(1)} + \sum_j d_j^{(r)}}{2} \right) \right] - w_1 \left[(\lambda - \delta_1) \left(\frac{\sum_j d_j^{(1)} + \sum_j d_j^{(r)}}{2} \right) \right] + w_2 \left[\delta_2 \left(\frac{\sum_k cr_k^{(1)} X_k + \sum_k cr_k^{(r)} X_k}{2} \right) - \sum_k cr_k^{(1)} X_k \right] + w_2 \left[(\lambda - \delta_2) \left(\frac{\sum_k cr_k^{(1)} X_k + \sum_k cr_k^{(r)} X_k}{2} \right) \right] \quad (27)$$

$$\sum_i QIJ_{ij} \geq \delta_1 \left(\frac{d_j^{(1)} + d_j^{(r)}}{2} \right) + (\lambda - \delta_1) \left(\frac{d_j^{(1)} + d_j^{(r)}}{2} \right) \quad \forall j \quad (28)$$

$$\sum_i QKI_{ki} \leq \sum_j QJK_{jk} \quad (29)$$

$$\sum_k QJK_{jk} = \alpha \left(\frac{d_j^{(1)} + 2d_j^{(r)} + d_j^{(r)}}{4} \right) \quad \forall j \quad (30)$$

هدف استفاده شده است. همچنین برای تعیین سطح اطمینان محدودیت‌های دارای پارامترهای غیرقطعی، از پارامترهای δ_1 و δ_2 استفاده شده که مقادیر آن‌ها باید توسط تصمیم‌گیرنده مشخص شود. یکی از محدودیت‌های این روش این است که با افزایش تعداد محدودیت‌ها تعیین مقادیر مناسب برای پارامترهای δ_1 و δ_2 مشکل می‌شود. محدودیت دیگر این است که مقدار انحراف تابع هدف نادیده گرفته شده است. برای غلبه بر ضعف‌های این مدل، مدل استوار توسط پیشوایی و خلف [۲۵] به صورت زیر پیشنهاد شده است:

$$\text{Min} \left(\frac{c^{(1)} + c^{(r)} + c^{(r)}}{3} \right) x + \left(\frac{f^{(1)} + f^{(r)} + f^{(r)}}{3} \right) y + \pi (c^{(r)} x + f^{(r)} y) - \pi \left[\left(\frac{c^{(1)} + c^{(r)} + c^{(r)}}{3} \right) x + \left(\frac{f^{(1)} + f^{(r)} + f^{(r)}}{3} \right) y \right] + w_1 \left[d^{(r)} - \delta_1 \left(\frac{d^{(1)} + d^{(r)}}{2} \right) - (\lambda - \delta_1) \left(\frac{d^{(1)} + d^{(r)}}{2} \right) \right] + w_2 \left[\delta_2 \left(\frac{N^{(1)} + N^{(r)}}{2} \right) + (\lambda - \delta_2) \left(\frac{N^{(1)} + N^{(r)}}{2} \right) - N^{(1)} \right] \text{ s.t. } Ax \geq \delta_1 \left(\frac{d^{(1)} + d^{(r)}}{2} \right) + (\lambda - \delta_1) \left(\frac{d^{(1)} + d^{(r)}}{2} \right) Sx \leq \left[\delta_2 \left(\frac{N^{(1)} + N^{(r)}}{2} \right) + (\lambda - \delta_2) \left(\frac{N^{(1)} + N^{(r)}}{2} \right) \right] y y = 0 \text{ or } 1, x \geq 0, 0 \leq \delta_1, \delta_1 \leq 1 \quad (18)$$

در مدل ۱۸ عبارات اول و دوم تابع هدف مشخص‌کننده‌ی میانگین عملکرد سیستم بر اساس مقادیر انتظاری پارامترهای غیرقطعی است. عبارت سوم تابع هدف تنظیم‌کننده‌ی درجه‌ی استواری بهینگی است که بیشینه‌ی انحراف از مقادیر انتظاری را کمینه می‌کند. عبارات چهارم و پنجم که تنظیم‌کننده‌ی استواری شدنی بودن است، برای تعیین فاصله بین بدبینانه‌ترین مقادیر و مقادیر انتخاب شده پارامترهای غیرقطعی در مدل گنجانده شده است. پارامترهای w_1 و w_2 بیان‌گر جریمه‌ی یک واحد انحراف محدودیت‌ها در نتیجه‌ی عدم دقت پارامترهای غیرقطعی است. با به‌کارگیری این رویکرد، معادل فازی استوار مدل زنجیره‌ی تأمین در نظر گرفته شده، به صورت زیر خواهد بود. مجموع روابط ۱۹ تا ۲۷ بیان‌گر تابع هدف است. عبارات ۲۸ تا ۳۵ محدودیت‌های مسئله را نشان می‌دهد.

$$\sum_j \sum_i \left(\frac{TIJ_{ij}^{(1)} + TIJ_{ij}^{(r)} + TIJ_{ij}^{(r)}}{3} \right) (\lambda - \pi) QIJ_{ij} + \sum_k \sum_j \left(\frac{TJK_{jk}^{(1)} + TJK_{jk}^{(r)} + TJK_{jk}^{(r)}}{3} \right) (\lambda - \pi) QJK_{jk} + \sum_i \sum_k \left(\frac{TKI_{ki}^{(1)} + TKI_{ki}^{(r)} + TKI_{ki}^{(r)}}{3} \right) (\lambda - \pi) QKI_{ki} B \quad (19)$$

$$\sum_k \left(\frac{c_k^{(1)} + c_k^{(r)} + c_k^{(r)}}{3} \right) (\lambda - \pi) X_k \quad (28)$$

$$+ \sum_k \left(\frac{\lambda^{(1)} + \lambda^{(r)} + \lambda^{(r)}}{3} \right) B^T (\lambda - \pi) X_k \quad (29)$$

$$\sum_j \sum_k \left(\frac{a_k^{(1)} + a_k^{(r)} + a_k^{(r)}}{3} \right) (\lambda - \pi) QJK_{jk} \quad (21)$$

می‌شود. برای افزایش کارایی روش مبتنی بر اولویت، در این مقاله طول رشته را برابر با $I + K$ در نظر می‌گیریم که موجب کاهش تعداد جواب‌های کل به $(I + K)!$ می‌شود. در نظر گرفتن رشته به طول $I + K$ برای مسئله‌ی زنجیره‌ی تأمین مدنظر، به یک رویه‌ی رمزنگاری^{۱۴} ابتکاری برای دست‌یابی به شبکه‌ی حمل‌ونقل نیاز دارد.

۱.۶. روش نمایش جواب

هر جواب، به صورت یک رشته اعداد (کروموزوم) به طول $I + K + ۳$ شامل سه زیررشته به طول ۳، I و K نمایش داده می‌شود. درحالی‌که سه عدد اول از زیررشته‌ی اول به ترتیب بیان‌گر B ، δ_1 و δ_2 است، از $I + K$ عدد بعدی (زیررشته‌های دوم و سوم) برای تعیین ساختار شبکه‌ی حمل‌ونقل استفاده می‌شود. شکل ۱ یک جواب با ۵ مرکز تولید و ۳ مرکز جمع‌آوری را نشان می‌دهد.

مقدارگره اول از زیررشته اول یک عدد تصادفی بین صفر و یک و مقدارگره‌های دوم و سوم این زیررشته یک عدد تصادفی بین ۰٫۵ و ۱ و یک است. مقدارگره‌های زیررشته‌های دوم و سوم یک جایگشت تصادفی از اعداد $I + K$ ، ۱، ۲، ... است.

۲.۶. رویه‌ی رمزنگاری

گام‌های مورد نیاز برای دست‌یابی به جواب متناظر با یک کروموزوم به صورت است:

۱. مقداربر متغیرهای B ، δ_1 و δ_2 را به ترتیب برابر مقدارگره‌های اول تا سوم زیررشته‌ی اول قرار دهید؛

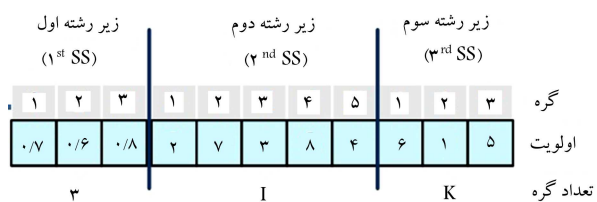
۲. زیررشته‌ی دوم را در نظر بگیرید و با اجرای الگوریتم ۱ (شکل ۲) تقاضای مشتریان را به مراکز تولید تخصیص دهید؛

۳. زیررشته‌ی سوم را در نظر بگیرید و با اجرای الگوریتم ۲ (شکل ۳) ضمن تعیین مراکز جمع‌آوری مورد نیاز، محصولات برگشتی مشتریان را به مراکز جمع‌آوری مشخص شده تخصیص دهید؛

۴. زیررشته‌های دوم و سوم را توأم در نظر بگیرید و با اجرای الگوریتم ۳ (شکل ۴) جریان محصول بین مراکز جمع‌آوری و مراکز تولید را مشخص کنید.

۳.۶. الگوریتم تفاضل تکاملی

تفاضل تکاملی یک الگوریتم فراابتکاری مبتنی بر جمعیت است که توسط استرم و پرایس^[۳۱] در سال ۱۹۹۵ ارائه شد. همانند GA، DE از عملگرهای تکاملی انتخاب، ترکیب و جهش بهره می‌برد. مهم‌ترین تفاوت DE با GA در این است که DE مستقیماً از اطلاعات جمعیت برای هدایت فرایند جست‌وجو استفاده می‌کند. در الگوریتم DE که ابتدا برای مسائل با متغیرهای پیوسته با کارایی بسیار بالا ارائه شد، ابتدا عملگر جهش، سپس ترکیب، و در انتها انتخاب اجرا می‌شود. در عملیات جهش، تفاضل موزون دو جواب تصادفی از جمعیت به جواب سوم اضافه



شکل ۱. نمونه‌ی کروموزوم با ۵ مرکز تولید و ۳ مرکز جمع‌آوری.

$$\sum_j QI J_{ij} \leq ca_i \quad \forall i \quad (31)$$

$$\sum_j QJ K_{jk} \leq \left[\delta_2 \left(\frac{cr_k^{(1)} + cr_k^{(2)}}{2} \right) + (1 - \delta_2) \left(\frac{cr_k^{(1)} + cr_k^{(2)}}{2} \right) \right] X_k \quad \forall k \quad (32)$$

$$\sum_j QI J_{ij} \leq MY_i \quad (33)$$

$$\sum_k X_k \leq n \quad (34)$$

$$B, \delta_1, \delta_2 \leq 1, \quad \delta_1, \delta_2 \geq 0.5 \quad (35)$$

با تعریف متغیر کمکی $(Y_i')X_k'$ و اضافه کردن محدودیت‌های زیر می‌توان متغیر دودویی $(Y_i)X_k$ را به متغیر پیوسته تبدیل و مدل را به غیرخطی پیوسته تبدیل کرد:

$$X_k + X_k' = 1 \quad (36)$$

$$X_k X_k' = 0 \quad (37)$$

$$X_k \geq 0, \quad X_k' \geq 0 \quad (38)$$

۶. رویکرد حل

از آنجاکه مدل توسعه داده شده در این پژوهش به عنوان زیرمجموعه‌ی بی از مسائل طراحی شبکه‌ی زنجیره‌ی تأمین چندمرحله‌ی بی یک مسئله‌ی NP-Hard شناخته می‌شود،^[۵۸] با افزایش اندازه مسئله، پیچیدگی آن افزایش می‌یابد و رویکردهای دقیق قادر به حل مسئله در زمان معمول نیستند.^[۵۹] همچنین غیرخطی بودن مدل موجب افزایش پیچیدگی حل آن می‌شود.^[۶۰] از این رو برای حل مدل ارائه شده در ابعاد بزرگ یک روش حل بر اساس الگوریتم فراابتکاری تفاضل تکاملی ارائه می‌شود. به دلیل این‌که عملکرد الگوریتم‌های تکاملی به شدت تحت تأثیر روش نمایش جواب^{۱۲} است، به ارائه‌ی یک روش نمایش جواب بر پایه‌ی رویه‌ی نمایش جواب مبتنی بر اولویت می‌پردازیم. رویه‌ی نمایش جواب مبتنی بر اولویت که متعلق به مجموعه‌ی روش‌های نمایش جواب مبتنی بر رأس است، توسط ژن و همکاران^[۶۱] برای غلبه بر محدودیت‌های روش عدد آزمونگر^{۱۳} پیشنهاد شده است. مهم‌ترین محدودیت اجرای رویه‌ی نمایش جواب عدد آزمونگر در مسائل زنجیره‌ی تأمین این است که با به‌کارگیری این روش ایجاد رشته‌ی بی از اعداد که بتواند معیار شدنی بودن را برآورده کند بسیار مشکل است.^[۶۱] در روش نمایش جواب مبتنی بر اولویت، یک عدد در یک رشته از اعداد دربرگیرنده‌ی دو نوع داده است. جایگاه عدد در رشته بیان‌گر مبدأ (یا مقصد) و بزرگی عدد تعیین‌کننده‌ی اولویت مبدأ (یا مقصد) برای انتخاب است. با به‌کارگیری نمایش جواب مبتنی بر اولویت برای یک شبکه‌ی زنجیره‌ی تأمین، یک رشته از اعداد شامل مجموعه اولویت‌هایی از مبادی و مقاصد به منظور حصول درخت حمل‌ونقل به‌کار گرفته می‌شود. در فرم استاندارد نمایش جواب مبتنی بر اولویت، طول یک رشته برای هر مرحله از شبکه‌ی زنجیره‌ی تأمین برابر تعداد کل مبادی (M) و مقاصد (N) است $(|M| + |N|)$. از آنجاکه شبکه‌ی CLSC در نظر گرفته شده دارای یک مرحله‌ی مستقیم و دو مرحله‌ی معکوس است به‌کارگیری فرم استاندارد نمایش جواب مبتنی بر اولویت موجب می‌شود طول رشته برابر $I + J + J + K + K + I$ شود که در آن I تعداد مراکز تولید، J تعداد مشتریان، و K تعداد مراکز جمع‌آوری است. در نتیجه تعداد جواب‌های کل برابر $(I + J)! \times (K + I)! \times (J + K)!$

Input: $2^{nd}SS, d_j, TJI_{ij}, ca_i$
Output: QIJ_{ij}
Step 0: $SS = 2^{nd}SS, Td = \sum_j d_j, QIJ_{ij} = 0 \forall i, j$
Step 1: $i^* = arg \max\{SS\}$ مبدأ با بالاترین اولویت را انتخاب کنید
Step 2: $j^* = arg \min\{TJI_{ij^*} | d_{j^*} > 0\}$ مقصد با کمترین هزینه حمل و نقل و با تقاضای مثبت را انتخاب کنید
Step 3: $QIJ_{i^*j^*} = \min\{d_{j^*}, ca_{i^*}\}$ مقصد انتخابی را به مبدأ انتخابی تخصیص دهید
 مقدار کل تقاضای برآورده نشده را به روز کنید $Td = Td - QIJ_{i^*j^*}$
 اگر کل تقاضا برآورده شده به الگوریتم خاتمه دهید *If Td = 0 then exit*
 ظرفیت مبدأ را به روز کنید $ca_{i^*} = ca_{i^*} - QIJ_{i^*j^*}$
 تقاضای مقصد را به روز کنید $d_{j^*} = d_{j^*} - QIJ_{i^*j^*}$
 اگر ظرفیت مبدأ انتخاب شده صفر است آن را از اولویت 1 *If ca_{i^*} = 0 then SS(i^*) = 0 and go to step 1*
 انتخاب خارج کرده و به گام یک بروید
 در غیر این صورت به گام ۲ بروید *Else go to step 2*

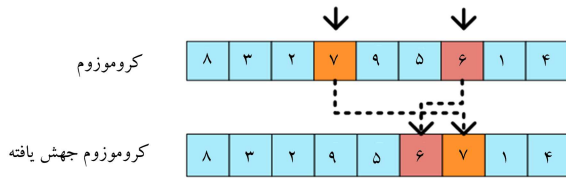
شکل ۲. الگوریتم ۱: تخصیص مشتریان به مراکز تولید.

Input: $3^{rd}SS, d_j, n, \alpha, TJK_{jk}, cr_k, \lambda$
Output: QJK_{jk}, X_k
Step 0: $SS = 3^{rd}SS, s_j = \alpha d_j, X_k = 0, Ts = \sum_j s_j, QJK_{jk} = 0 \forall j, k$
Step 1: $k^* = arg \max\{SS\}$ مقصد با بالاترین اولویت را انتخاب کنید
 تعیین کنید که مقصد انتخاب شده باز است $X_{k^*} = 1$
 اگر تعداد مقاصد باز بیشتر از سقف مجاز $2 * \lambda$ است *if $\sum_k X_k > n$ then consider a penalty equal to $2 * \lambda$*
 است جریمه در نظر بگیرید
Step 2: $j^* = arg \min\{TJK_{jk^*} | s_{j^*} > 0\}$ مبدأ با کمترین هزینه حمل و نقل و با عرضه مثبت را انتخاب کنید
Step 3: $QJK_{j^*k^*} = \min\{s_{j^*}, cr_{k^*}\}$ مبدأ انتخابی را به مقصد انتخابی تخصیص دهید
 مقدار کل عرضه جمع‌آوری نشده را به روز کنید $Ts = Ts - QJK_{j^*k^*}$
 اگر کل عرضه جمع‌آوری شده به الگوریتم خاتمه دهید *If Ts = 0 then exit*
 ظرفیت مقصد را به روز کنید $cr_{k^*} = cr_{k^*} - QJK_{j^*k^*}$
 عرضه مبدأ را به روز کنید $s_{j^*} = s_{j^*} - QJK_{j^*k^*}$
 اگر ظرفیت مقصد انتخاب شده صفر است آن را از اولویت 1 *If cr_{k^*} = 0 then SS(k^*) = 0 and go to step 1*
 انتخاب خارج کرده و به گام یک بروید
 در غیر این صورت به گام ۲ بروید *Else go to step 2*

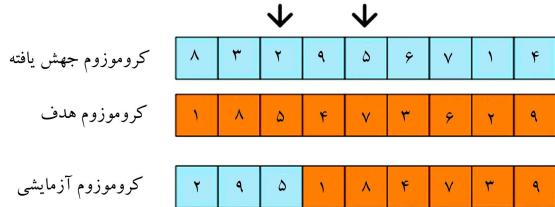
شکل ۳. الگوریتم ۲: تخصیص محصولات برگشتی به مراکز جمع‌آوری.

Input: $2^{nd}SS, 3^{rd}SS, TKI_{ki}, QIJ_{ij}, QJK_{jk}$
Output: QKI_{ki}
Step 0: $SS = 2^{nd}SS, I, s_k = \sum_j QJK_{jk}, o_i = \sum_j QIJ_{ij}, Ts = \sum_k s_k, QKI_{ki} = 0 \forall i, k$
Step 1: $A = arg \max\{SS\}$ جایگاه با بیشترین اولویت را تعیین کنید
 اگر جایگاه انتخاب شده مربوط به زیررشته دوم است آن را به عنوان مقصد انتخابی در نظر بگیرید *If $A \leq I$ then $i^* = A$*
 در غیر این صورت آن را به عنوان مبدأ انتخابی در نظر بگیرید *Else $k^* = A - I$ and go to step 4*
Step 2: $k^* = arg \min\{TKI_{ki^*} | s_{k^*} > 0\}$ مبدأ با کمترین هزینه حمل و نقل و با عرضه مثبت را انتخاب کنید
Step 3: $QKI_{k^*i^*} = \min\{s_{k^*}, o_{i^*}\}$ مبدأ انتخابی را به مقصد انتخابی تخصیص دهید
 مقدار کل عرضه جمع‌آوری نشده را به روز کنید $Ts = Ts - QKI_{k^*i^*}$
 اگر کل عرضه جمع‌آوری شده به الگوریتم خاتمه دهید *If Ts = 0 then exit*
 ظرفیت مقصد را به روز کنید $o_{i^*} = o_{i^*} - QKI_{k^*i^*}$
 عرضه مبدأ را به روز کنید $s_{k^*} = s_{k^*} - QKI_{k^*i^*}$
 اگر عرضه مبدأ انتخاب شده تمام شده است به گام ۲ بروید *If $s_{k^*} = 0$ then go to step 2*
 در غیر این صورت مقصد انتخاب شده را از اولویت خارج کرده و به گام یک بروید *Else $SS(i^*) = 0$ and go to step 1*
Step 4: $i^* = arg \min\{TKI_{ki^*} | o_{i^*} > 0\}$ مقصد با کمترین هزینه حمل و نقل و با ظرفیت مثبت را انتخاب کنید
Step 5: $QKI_{k^*i^*} = \min\{s_{k^*}, o_{i^*}\}$ مقصد انتخابی را به مبدأ انتخابی تخصیص دهید
 مقدار کل عرضه جمع‌آوری نشده را به روز کنید $Ts = Ts - QKI_{k^*i^*}$
 اگر کل عرضه جمع‌آوری شده به الگوریتم خاتمه دهید *If Ts = 0 then exit*
 ظرفیت مقصد را به روز کنید $o_{i^*} = o_{i^*} - QKI_{k^*i^*}$
 عرضه مبدأ را به روز کنید $s_{k^*} = s_{k^*} - QKI_{k^*i^*}$
 اگر عرضه مبدأ تمام شده است آن را از اولویت خارج کرده و به گام یک بروید *If $s_{k^*} = 0$ then SS(A) = 0 and go to step 1*
 در غیر این صورت گام ۴ بروید *Else go to step 4*

شکل ۴. الگوریتم ۳: تعیین جریان محصول بین مراکز جمع‌آوری و مراکز تولید.



شکل ۵. عملگر جهش جابه جایی.



یا



شکل ۶. عملگر ترکیب دو برش.

همچنین جهت تطابق بیشتر، F یک عدد حقیقی در بازه $[0, 1]$ در نظر گرفته می شود. در صورتی که مقادیر ژن های زیررشته ای اول از حد مجاز تجاوز کند، جریمه در نظر گرفته می شود.

عملگرهای جهش و ترکیب معرفی شده، بر روی زیررشته ای اول کروموزوم که دارای ژن هایی با مقادیر حقیقی است قابل پیاده سازی است. از آنجا که کروموزوم تعریف شده، دارای ژن هایی با مقادیر گسسته در زیررشته های دوم و سوم است، از عملگرهای جهش جابه جایی^{۱۵} و ترکیب دو برش^{۱۶} ارائه شده توسط پن و همکاران،^[۳۲] در کنار دو عملگر پیشین برای ایجاد جواب های جدید استفاده می شود. شکل های ۵ و ۶ چگونگی اجرای عملگرهای جابه جایی و دو برش را نشان می دهد.

در نهایت فرایند اجرای الگوریتم DE برای مسئله در نظر گرفته شده به صورت شکل ۷ است. همان طور که در شکل ۷ مشخص است، برای تعیین کیفیت جواب ها از رویه رمزنگاری پیشنهادی استفاده می شود. همچنین برای اجرای عملگرهای جهش و ترکیب، کروموزوم به دو بخش تقسیم می شود. عملگرهای جهش و ترکیب استاندارد DE بر روی زیررشته ای اول و عملگرهای جابه جایی و دو برش بر روی زیررشته های دوم و سوم اجرا می شود.

۷. تنظیم پارامترها

در این مقاله برای تنظیم دو پارامتر نرخ ترکیب و اندازهی جمعیت الگوریتم DE از روش پاسخ سطح (RSM)^{۱۷} استفاده می شود. روش پاسخ سطح یک روش مدل سازی تجربی است که برای تعیین شرایط بهینه ای متغیرهای یک فرایند به کار گرفته می شود. بر این اساس و پس از اجراهای مقدماتی الگوریتم، سطوح پارامترها به صورت جدول ۱ تعیین می شود.

در روش RSM برای هر متغیر وابسته مدلی تعریف می شود که آثار اصلی و متقابل پارامترها را بر روی هر متغیر جداگانه بیان می کند. مدل چندمنظوره به صورت زیر است:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_{11} X_1^2 + \beta_{22} X_2^2 + \beta_{33} X_3^2 + \beta_{12} X_1 X_2 + \beta_{13} X_1 X_3 + \beta_{23} X_2 X_3 \quad (42)$$

می شود و جواب حاصل، جواب جهش یافته نامیده می شود. سپس عملگر ترکیب برای ترکیب جواب جهش یافته و جواب هدف و ایجاد جواب آزمایشی به کار گرفته می شود. در نهایت عملگر انتخاب به مقایسه کیفیت دو جواب هدف و آزمایشی می پردازد و جواب بهتر به نسل بعد انتقال داده می شود. از آنجا که الگوریتم DE یک روش جست و جوی جواب رقابتی است که کارایی بالایی در بسیاری از توابع آزمون استاندارد و مسائل بهینه سازی دنیای واقعی داشته است،^[۶۲] در این مقاله از این الگوریتم برای حل مسئلهی زنجیره ای تأمین در ابعاد بزرگ استفاده می شود. الگوریتم DE برای مسئلهی مدنظر، دارای گام های زیر است:

- گام ۱. یک جمعیت اولیه ی تصادفی تولید کنید؛
- گام ۲. شماره ی نسل را برابر صفر بگیرید؛
- گام ۳. تا زمان رسیدن به شرط توقف مراحل زیر را اجرا کنید:
 - گام ۱.۳. به شماره ی جمعیت یکی اضافه کنید؛
 - گام ۲.۳. برای شماره ی i از ۱ تا اندازه ی جمعیت مراحل زیر را اجرا کنید:
 - گام ۱.۲.۳. عملگر جهش را به کار ببرید؛
 - گام ۲.۲.۳. عملگر ترکیب را برای ترکیب جواب جهش یافته و جواب i برای خلق جواب آزمایشی به کار ببرید؛
 - گام ۳.۳. عملگر انتخاب شامل دو مورد زیر را به کار ببرید:

تعیین کیفیت جواب های هدف (جواب i) و آزمایشی؛

پذیرش جواب بهتر از بین جواب های هدف و آزمایشی برای جمعیت جدید؛
 - گام ۴. جمعیت جدید را جایگزین جمعیت پیشین کنید و به گام ۳ بروید.

برای هر جواب (یا بردار) هدف $(i = 1, 2, \dots, N)$ ، پرایس و استرم^[۳۱] عملگر جهش را به صورت زیر معرفی کردند:

$$v_{i,G+1} = x_{r_1,G} + F \cdot (x_{r_2,G} - x_{r_3,G}) \quad (39)$$

که در آن r_1, r_2, r_3 اعداد تصادفی از مجموعه $\{1, 2, \dots, N\}$ و مخالف i هستند و $F > 0$. F یک عدد حقیقی و ثابت در بازه $[0, 2]$ است که به عنوان وزن برای تفاضل $(x_{r_2,G} - x_{r_3,G})$ عمل می کند. همچنین G بیانگر نسل موجود و $G + 1$ بیانگر نسل جدید است.

عملگر ترکیب برای ایجاد جواب آزمایشی $(j = 1, 2, \dots, D)$ به صورت زیر ارائه شد:

$$u_{j,i,G+1} = \begin{cases} v_{j,i,G+1} & \text{if } (randb(j) \leq CR) \text{ and } j = rndr \\ x_{j,i,G} & \text{if } (randb(j) > CR) \text{ and } j \neq rndr \end{cases} \quad (40)$$

که در آن $randb(j)$ یک عدد تصادفی در بازه $[0, 1]$ ، CR نرخ ترکیب و $rndr$ یک عدد تصادفی بین صفر و D است. D ، تعداد بعد (تعداد ژن) است. این عملگر تضمین می کند که جواب آزمایشی، حداقل در یک ژن با جواب جهش یافته تفاوت دارد.

عملگرهای جهش و ترکیب در نسخه ی اولیه DE به صورت فوق است. در این مقاله ما از بهترین عضو جامعه در تولید جواب های جهش یافته استفاده می کنیم (نسخه DE/best/1/bin). در نتیجه عملگر جهش به صورت زیر تغییر می یابد:

$$v_{i,G+1} = x_{best,G} + F \cdot (x_{r_1,G} - x_{r_2,G}) \quad (41)$$

جدول ۲. کلاس مسائل نمونه.

کلاس	I	J	K	n
۱	۳	۴	۲	۱
۲	۵	۱۵	۳	۲
۳	۸	۵۰	۵	۳
۴	۱۵	۹۰	۱۰	۵
۵	۲۵	۱۵۰	۲۰	۸
۶	۴۰	۲۵۰	۳۰	۱۵

جدول ۳. محدوده‌ی پارامترهای مسائل نمونه.

پارامتر	مقدار
α	۰٫۴
v	$U(۸, ۱۰, ۱۲, ۱۴, ۱۶)$
β	$U(۰٫۲۵, ۰٫۳۰, ۰٫۳۵, \dots, ۰٫۵۰)$
a_k	$U(۴, ۴٫۵, ۵, \dots, ۷)$
e_k	$U(۰٫۵, ۰٫۷, ۰٫۹, \dots, ۱٫۵)$
λ	$N(۵۰۰, ۳۰)$
c_k	$N(۱۵۰۰, ۵۰)$
f_i	$N(۵۰۰, ۲۰)$
d_j	$U(۳۰, ۳۵, ۴۰, \dots, ۷۰)$
m_i	$U(۶, ۶٫۵, ۷, \dots, ۱۵)$
r_i	$U(۲, ۲٫۲, ۲٫۴, \dots, ۵)$
$TJ_i j_j$	$U(۲, ۲٫۴, ۲٫۸, \dots, ۸)$
TKI_{ki}	$U(۳, ۳٫۵, ۴, \dots, ۹)$
TJK_{jk}	$U(۱, ۱٫۲, ۱٫۴, \dots, ۴)$
ca_i	$U\left(\frac{2 \sum_j d_j}{I} \pm 0.5 \frac{\sum_j d_j}{I}\right)$
cr_k	$U\left(\frac{\sum_j d_j}{n} \pm 0.2 \frac{\sum_j d_j}{K}\right)$

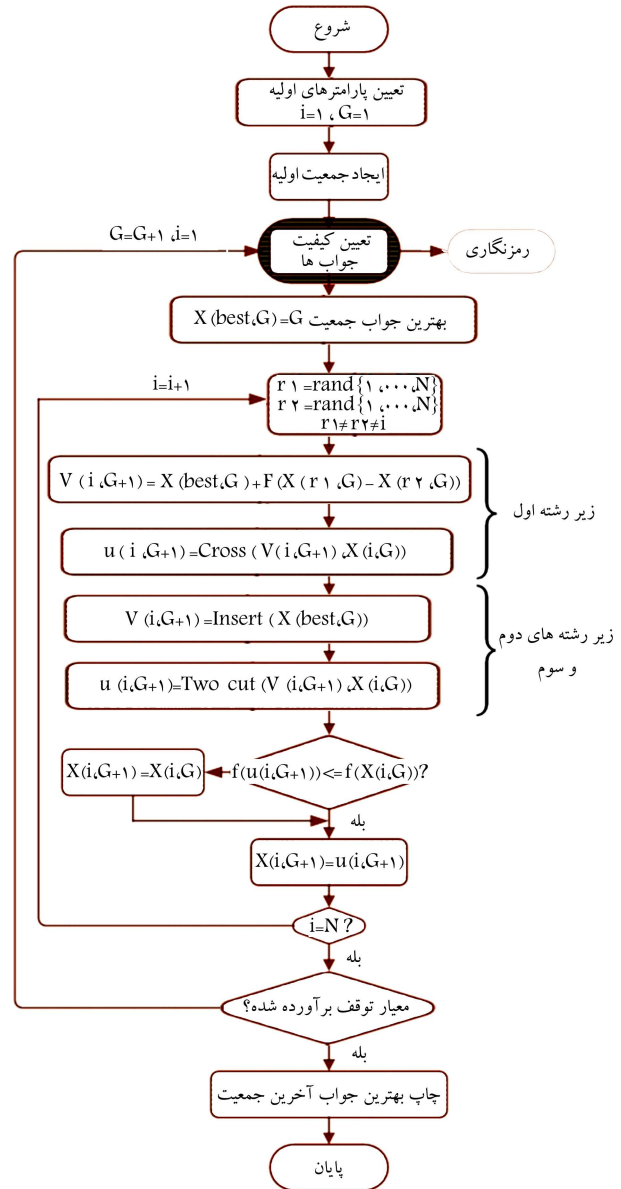
پارامترهای مسائل هر کلاس به صورت تصادفی و با به‌کارگیری توزیع‌های نرمال و یک‌نواخت مطابق جدول ۳ به دست می‌آید. حد بالا و پایین پارامترهای فازی با ضرب پارامتر در یک عدد تصافی از بازه‌های [۱٫۳ ۱٫۱] و [۰٫۹ ۰٫۷] به دست می‌آید.

برای تنظیم پارامترها ۳ نمونه مسئله از هر کلاس ایجاد می‌شود. به دلیل ماهیت تصادفی الگوریتم، هر مسئله ۶ بار حل شده و نتایج ذخیره می‌شود. به منظور اجرای الگوریتم از نرم‌افزار متلب نسخه ۷٫۸ که بر روی یک رایانه‌ی خانگی با پردازنده‌ی دو هسته‌ی ۵۳۰۰ نصب شده است، استفاده می‌شود. زمان اجرای الگوریتم به عنوان معیار توقف در نظر گرفته می‌شود.

از آنجا که هر کلاس از مسئله دارای ابعاد مخصوص به خود است، مقیاس تابع هدف در هر مسئله نمونه از نمونه‌ی دیگر متفاوت است و نمی‌تواند مستقیماً مورد استفاده قرار گیرد. برای غلبه بر این مشکل از روش بی‌مقیاس‌سازی درصد انحراف نسبی (RPD)^{۱۸} استفاده می‌شود. مقدار RPD از رابطه‌ی زیر به دست می‌آید:

$$RPD = \frac{Alg_{sol} - Min_{sol}}{Min_{sol}} \times 100 \quad (43)$$

که در آن Alg_{sol} جواب به دست آمده در هر بار اجرای یک نمونه مسئله و Min_{sol} بهترین جواب به دست آمده در بین ۶ بار تکرار یک نمونه مسئله است. پس از تبدیل مقادیر تابع هدف به RPD، میانگین RPDها برای هر مسئله‌ی نمونه محاسبه



شکل ۷. فلوچارت الگوریتم تفاضل تکاملی پیشنهادی.

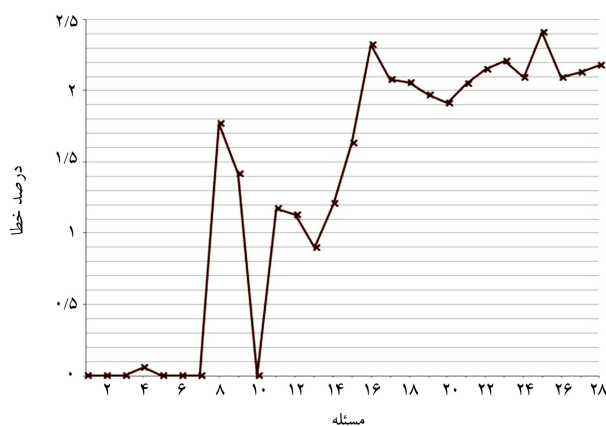
جدول ۱. پارامترها و سطوح در روش پاسخ سطح.

پارامتر(فاکتور) مستقل	نماد	کد و سطح مربوطه
نرخ ترکیب	CR	۰ ۱
اندازه جمعیت	N	۸۰ ۱۲۰ ۱۶۰

که در آن Y پاسخ پیش‌بینی شده، β ضریب ثابت، $\beta_1, \beta_2, \beta_3$ اثرات خطی، $\beta_{11}, \beta_{12}, \beta_{13}, \beta_{22}, \beta_{23}, \beta_{33}$ اثر مربعات و $\beta_{12}, \beta_{13}, \beta_{23}$ اثرات متقابل است. به منظور تجزیه و تحلیل از نرم‌افزار Design Expert ۷٫۰ استفاده می‌شود. بر این اساس تعداد آزمایش‌های مورد نیاز برابر ۱۱ است. به منظور افزایش دقت نتایج، مسئله‌ی زنجیره‌ی تأمین مورد نظر با پارامترها و اندازه‌های مختلف حل شده و نتایج بی‌مقیاس شده برای تنظیم پارامترها به کار گرفته می‌شود. جدول ۲ مشخصات کلی هر کلاس از مسئله را نشان می‌دهد.

جدول ۵. مقایسه‌ی نتایج DE و گمز.

درصد خطا	زمان (S)		تابع هدف		مستله	کلاس
	DE	گمز	میانگین DE	بهترین DE		
۰	۳	۷	۲۴۶۲۷	۲۴۶۲۷	۲۴۶۲۷	۱
۰	۳	۷	۲۶۹۳۲	۲۶۸۵۱	۲۶۸۵۱	۱
۰	۳	۸	۲۷۰۴۹	۲۷۰۴۹	۲۷۰۴۹	۱
۰٫۰۶	۳	۸	۲۷۵۸۳	۲۷۵۶۴	۲۷۵۴۷	۱
۰	۳	۷	۲۹۶۲۴	۲۹۶۲۴	۲۹۶۲۴	۱
۰	۳	۸	۳۰۱۱۲	۳۰۱۱۲	۳۰۱۱۲	۱
۰	۳	۷	۳۲۶۳۷	۳۲۵۴۶	۳۲۵۴۶	۱
۱٫۷۷	۶	۲۶۷	۸۱۵۷۵	۸۰۹۵۲	۷۹۵۴۶	۲
۱٫۴۲	۶	۲۷۲	۸۵۶۳۴	۸۴۶۳۲	۸۳۴۵۱	۲
۰	۶	۲۶۸	۹۰۲۱۴	۸۹۶۸۰	۸۹۶۸۰	۲
۱٫۱۷	۶	۲۶۴	۹۸۲۴۷	۹۶۷۴۵	۹۵۶۲۷	۲
۱٫۱۳	۶	۲۸۹	۱۰۱۵۴	۱۰۰۶۸۵	۹۹۵۵۷	۲
۰٫۹۰	۶	۲۹۱	۱۱۳۸۴۵	۱۱۱۲۳۶	۱۱۰۲۴۵	۲
۱٫۲۱	۶	۲۸۶	۱۱۹۶۷۴	۱۱۸۲۰۱	۱۱۶۷۸۹	۲
۱٫۶۳	۱۱	۱۲۳۵	۲۷۶۸۵۴	۲۷۲۲۱۵	۲۶۷۸۵۱	۳
۲٫۳۲	۱۱	۱۴۴۶	۲۸۳۵۴۱	۲۷۹۵۱۴	۲۷۳۱۷۴	۳
۲٫۰۸	۱۱	۱۳۶۸	۲۸۶۷۴۹	۲۸۲۶۱۲	۲۷۶۸۵۰	۳
۲٫۰۶	۱۱	۱۳۴۹	۳۰۰۵۱۳	۲۹۵۶۲۳	۲۸۹۶۵۴	۳
۱٫۹۷	۱۱	۱۴۸۵	۲۹۸۶۷۴	۲۹۵۹۴۱	۲۹۰۲۱۴	۳
۱٫۹۱	۱۱	۱۴۶۲	۳۰۵۱۴۳	۳۰۱۹۸۴	۲۹۶۳۱۲	۳
۲٫۰۵	۱۱	۱۳۹۹	۳۰۶۰۱۸	۳۰۵۸۴۷	۲۹۹۶۸۹	۳
۱۵٫۲	۲۵	۵۶۸۳	۵۹۰۵۶۲	۵۸۴۴۲۶	۵۷۲۱۲۵	۴
۲۱٫۲	۲۵	۵۴۳۲	۶۰۵۶۲۰	۵۹۹۲۶۷	۵۸۶۳۱۰	۴
۲۹٫۲	۲۵	۶۰۱۲	۶۱۴۸۸۱	۶۰۷۸۱۸	۵۹۴۲۱۱	۴
۴۱٫۲	۲۵	۵۸۶۳	۶۲۴۰۵۸	۶۱۵۷۲۱	۶۰۱۲۳۱	۴
۰۹٫۲	۲۵	۵۷۱۲	۶۳۹۰۱۶	۶۳۱۳۷۷	۶۱۸۴۵۱	۴
۳۳٫۲	۲۵	۶۱۷۳	۶۵۰۷۴۳	۶۴۳۰۵۲	۶۲۸۴۱۰	۴
۱۸٫۲	۲۵	۵۷۶۳	۶۷۵۴۶۴	۶۶۷۴۵۴	۶۵۳۲۱۴	۴



شکل ۸. درصد خطای الگوریتم تفاضل تکاملی.

می‌شود. در نهایت بهترین سطح هر یک از پارامترها توسط نرم‌افزار به صورت جدول ۴ تعیین شد.

طبق جدول ۴ بهترین مقادیر پارامترها برابر ۰٫۷۴۵ برای نرخ ترکیب و ۱۲۱ برای اندازه‌ی جمعیت است. معادله‌ی خط نهایی برای متغیر پاسخ به صورت زیر است:

$$Y = 0.718 - 0.012CR - 0.003N - 0.010 \times CR \times N + 0.042CR^2 + 0.057N^2 \quad (44)$$

۸. مثال‌های عددی

برای ارزیابی عملکرد روش حل ارائه شده از دو روش استفاده می‌شود. در روش اول ۲۸ مسئله شامل ۷ مسئله از هر یک از کلاس‌های ۱ تا ۴ به صورت تصادفی ایجاد شده و به وسیله‌ی نرم‌افزار گمز جواب بهینه‌ی آن‌ها به دست آمده و با جواب‌های حاصل از اجرای الگوریتم پیشنهادی مقایسه شده است. در روش دوم ۲۱ مسئله شامل ۷ مسئله از هر یک از کلاس‌های ۴ تا ۶ به صورت تصادفی ایجاد شده و جواب‌های حاصل از روش حل پیشنهادی با جواب‌های GA مقایسه شده است.

۱.۸. مقایسه با گمز

جدول ۵ خلاصه‌ی نتایج حل مسائل نمونه به وسیله گمز و رویکرد پیشنهادی را ارائه می‌کند. هر مسئله ۶ بار با رویکرد پیشنهادی حل شده و بهترین جواب و میانگین جواب‌ها گزارش شده است. در جدول ۵ درصد خطا به صورت زیر محاسبه شده است:

$$\% \text{خطا} = \frac{\text{جواب گمز - بهترین DE}}{\text{جواب گمز}} \times 100 \quad (45)$$

شکل ۸ درصد خطاها را به طور واضح‌تری نمایش می‌دهد. همان‌طور که در جدول ۵ و شکل ۸ مشخص است جواب‌های حاصل از DE انحراف کمی نسبت به جواب بهینه دارند. این در حالی است که این جواب‌ها در زمان محاسباتی بسیار کمتری در مقایسه با گمز حاصل شده است. ۶ مسئله از ۷ مسئله‌ی کلاس ۱ دارای درصد خطای صفر هستند. همچنین یکی از مسائل کلاس ۲ نیز درصد خطایی برابر صفر دارد. ۱۷ مسئله خطایی کمتر از ۲ درصد دارند و حداکثر خطا برابر ۲٫۴۱ درصد مربوط به مسئله‌ی ۴ از کلاس ۴ است. تمام مسائل کلاس ۴ خطایی بیشتر از ۲ درصد دارند. در ۴ مسئله از کلاس ۱ میانگین جواب‌های DE با بهترین جواب DE برابر شده است. این بدان معنی است که ۶ اجرای DE جواب بهینه ایجاد کرده است. همان‌طور که مشخص است هر چه ابعاد مسئله بزرگ‌تر می‌شود جواب‌های DE انحراف بیشتری از جواب بهینه پیدا می‌کنند. با این حال انحراف در حد قابل قبولی است و می‌توان در مسائل بزرگ‌تر به رویکرد حل پیشنهادی اعتماد کرد.

جدول ۴. سطوح بهینه‌ی پارامترها، متغیر پاسخ و مطلوبیت.

مطلوبیت	Y	N	CR
۰٫۹۰۴	۰٫۱۸۱	۰٫۰۴	۰٫۱۵

RPD جدول نشان می دهد که در برخی مسائل میانگین درصد انحراف نسبی DE از GA کمتر است و در برخی مسائل دیگر برعکس. با توجه به این که RPD یک شاخص پراکندگی است می توان نتیجه گرفت که اگرچه بهترین جواب DE از GA بهتر است، اما الگوریتم DE در شاخص RPD بر GA برتری ندارد.

۲.۲.۸. سیستم رتبه بندی شطرنج

برای ارزیابی بهتر الگوریتم های DE و GA از سیستم رتبه بندی شطرنج الگوریتم های تکاملی (CRS4EAs)^[۲۴] ارائه شده توسط ویزک و همکاران^[۵۰] در سال ۲۰۱۴ استفاده می کنیم. اگرچه آزمون های آماری مانند NHST^[۲۵] از اهمیت بالایی در مقایسه ی الگوریتم ها برخوردارند، اما گاهی به کارگیری این روش ها به درستی انجام نگرفته و تفسیرها اشتباه صورت می گیرد. به این دلیل در این مقاله از روش CRS4EAs استفاده می شود. از مزایای روش CRS4EAs نسبت به آزمون های NHST می توان به سهولت استفاده، حساسیت کمتر نسبت به داده های دورافتاده، و کارایی بالاتر در رتبه بندی الگوریتم جدید توسعه یافته اشاره کرد.^[۵۰] در CRS4EAs هر الگوریتم تکاملی به عنوان یک بازیکن شطرنج، یک مقایسه بین دو الگوریتم به عنوان یک بازی و اجرای تمام مقایسات دوتایی بین الگوریتم ها به عنوان یک تورنمنت شناخته می شود. هر الگوریتم در یک بازی سه خروجی ممکن دارد: برد (با امتیاز ۱)، باخت (با امتیاز صفر)، تساوی (با امتیاز ۰/۵). در میان سیستم های مختلف رتبه بندی در شطرنج مانند الو، چسمریکس، عدد ارزیابی آلمانی و غیره، CRS4EAs از سیستم گلیکو-۲ که توسط گلیکمن در سال ۲۰۱۲ به عنوان نسخه ارتقا یافته گلیکو ارائه شد بهره می گیرد^[۶۷] به طور خلاصه برای مقایسه ی الگوریتم i با J الگوریتم $(j = 1, 2, \dots, J)$ بر پایه ی N مسئله ی بهینه سازی $(n = 1, 2, \dots, N)$ و اجرای مستقل $(l = 1, 2, \dots, L)$ پس از یک تورنمنت، CRS4EAs به صورت زیر عمل می کند:^[۶۷، ۵۰]

۱. تعیین پارامترهای اولیه (R: رتبه ی بازیکن i ، RD: انحراف رتبه ی بازیکن i ، σ : نوسانات بازیکن i ، τ : ثابت سیستم، ε : محدوده ی نتیجه ی تساوی).

۲. انجام $M = J \times N \times L$ بازی و ذخیره ی نتایج.

۳. تعیین و ذخیره ی امتیازات بازیکن i در برابر هر یک از رقبا $(S_{j,n,l} = 0, 0.5, 1)$.

۴. تبدیل رتبه و انحراف رتبه ی بازیکن i و رقبا به مقیاس مورد قبول سیستم گلیکو-۲:

$$\mu_j = \frac{(R_j - 1500)}{1737178} \quad (46)$$

$$\phi_j = \frac{RD_j}{1737178} \quad (47)$$

۵. محاسبه ی پارامتر وزن رقبا:

$$g(\phi_j) = \left(1 + 2\phi_j^2/\pi^2\right)^{-0.5} \quad (48)$$

۶. محاسبه ی امتیاز انتظاری بازیکن i در مقابل رقبا:

$$E(\mu, \mu_j, \phi_j) = \left(1 + 10^{-g(\phi_j)(\mu - \mu_j)}\right)^{-1} \quad (49)$$

۷. محاسبه ی واریانس رتبه ی بازیکنان:

$$v = \left(\sum_{j=1}^J g(\phi_j)^2 E(\mu, \mu_j, \phi_j) (1 - E(\mu, \mu_j, \phi_j)) NL\right)^{-1} \quad (50)$$

۲.۲.۸. مقایسه با الگوریتم ژنتیک

به طور کلی هر الگوریتم ژنتیک برای حل یک مسئله دارای اجزای زیر است: نمایش جواب، تولید جواب اولیه، ارزیابی تطابق، فرایند انتخاب، عملگرهای عمومی (بازتولید، ترکیب و جهش) و مقادیر پارامترها (اندازه ی جمعیت، احتمال اجرای عملگرها).^[۶۳] مشخصات الگوریتم ژنتیک به کار گرفته شده در این مقاله در جدول ۶ ارائه شده است. در جدول ۶ به منظور تعیین پارامترها با به کارگیری اجراهای آزمایشی از روش پاسخ سطح استفاده شده است.

همان طور که جدول ۶ نشان می دهد برای انجام ترکیب و جهش از دو عملگر استفاده شده است. پس از انتخاب دو والد به وسیله ی عملگر ۴- تورنمنت^[۶۳]، ابتدا عملگر یکنواخت^[۶۴] برای ترکیب زیررشته های اول دو والد به کار رفته است و سپس عملگر نگاشت بخشی (PMX)^[۶۵] برای ترکیب زیررشته های دوم و سوم دو والد مذکور به کار گرفته می شود. پس از انجام عمل ترکیب، عملگر مبادله^[۶۶] بر روی زیررشته ی اول و عملگر انتقال^[۶۳] بر روی زیررشته های دوم و سوم برای ایجاد جهش در جواب به کار گرفته می شود.

برای مقایسه ی الگوریتم پیشنهادی با GA، ۲۱ مسئله شامل ۷ مسئله از هر یک از کلاس های ۴، ۵ و ۶ به طور تصادفی تولید و هر مسئله ۶ بار با الگوریتم پیشنهادی و GA حل می شود. برای مقایسه ی عملکرد دو الگوریتم از دو روش استفاده می شود. در روش اول بهترین جواب، میانگین جواب ها و میانگین درصد انحراف نسبی RPD دو الگوریتم با هم مقایسه می شود. در روش دوم از یک روش ارزیابی مبتنی بر رتبه بندی شطرنج استفاده می شود.

۱.۲.۸. بهترین جواب، میانگین جواب ها و RPD

با در نظر گرفتن زمان اجرای الگوریتم ها به عنوان معیار توقف، جدول ۷ نتایج حاصل از اجرای دو الگوریتم پیشنهادی و GA را برای ۲۱ مسئله ی نمونه نشان می دهد. در جدول ۷، RPD میانگین RPD های ۶ بار حل هر مسئله است.

همان طور که در جدول ۷ مشخص است تنها در مسائل ۳ و ۶ از کلاس ۴ و مسئله ۹ از کلاس ۵ بهترین جواب DE از GA بهتر است. این امر نشان می دهد که الگوریتم پیشنهادی نسبت به الگوریتم ژنتیک در تعداد اجراهای مساوی برای یک مسئله و در زمان برابر در بیشتر اوقات جواب بهتری ایجاد می کند. همچنین جدول ۷ بیان گر این است که در مسائل ۳ و ۶ از کلاس ۴، مسائل ۹ و ۱۲ از کلاس ۵ و مسئله ۱۶ از کلاس ۶ میانگین جواب های GA از DE بهتر است. ستون

جدول ۶. مشخصات الگوریتم ژنتیک مورد استفاده.

مشخصه GA	شرح
نمایش جواب	روش نمایش مبتنی بر اولویت پیشنهادی
جواب اولیه	تصادفی
ارزیابی تطابق	تابع هدف
عملگر انتخاب	۴- تورنمنت ^[۶۳]
بازتولید	انتقال بهترین جواب نسل قبلی به نسل جدید
ترکیب	نگاشت بخشی (PMX) ^[۶۵] و یکنواخت
جهش	انتقال و مبادله
اندازه ی جمعیت	۹۲
رخ جهش	۰/۱۶

جدول ۷. مقایسه‌ی DE و GA.

زمان (S)	GA			DE			کلاس	مسئله
	RPD	میانگین	بهترین	RPD	میانگین	بهترین		
۲۵	۱,۱۸	۵۴۵۹۱۳	۵۳۹۵۴۶	۱,۲۱	۵۴۳۲۳۶	۵۳۶۷۴۱	۴	۱
۲۵	۱,۱۶	۵۵۷۷۹۱	۵۵۱۳۹۵	۱,۰۸	۵۵۳۸۲۸	۵۴۷۹۱۱	۴	۲
۲۵	۱,۱۴	۵۹۳۵۳۲	۵۸۶۸۴۲	۱,۱۹	۵۹۴۹۶۰	۵۸۷۹۶۳	۴	۳
۲۵	۱,۱۱	۶۱۱۲۴۱	۶۰۴۵۳۱	۱,۱۴	۶۰۸۰۹۲	۶۰۱۲۳۸	۴	۴
۲۵	۱,۰۹	۶۱۸۲۰۸	۶۱۱۵۴۲	۰,۹۹	۶۱۷۰۳۱	۶۱۰۹۸۲	۴	۵
۲۵	۱,۱۷	۶۳۰۸۰۹	۶۲۳۵۱۴	۱,۱۵	۶۳۱۷۳۳	۶۲۴۵۵۱	۴	۶
۲۵	۱,۰۹	۶۴۸۹۱۶	۶۳۸۹۵۱	۱,۲۳	۶۴۳۲۶۳	۶۳۵۴۴۷	۴	۷
۴۵	۱,۷۵	۹۲۱۲۶۴	۹۰۵۴۱۹	۱,۶۸	۹۱۷۰۲۲	۹۰۱۸۷۱	۵	۸
۴۵	۱,۹۰	۹۵۵۶۳۵	۹۳۷۸۱۶	۱,۹۵	۹۵۶۷۱۵	۹۳۸۴۱۶	۵	۹
۴۵	۱,۸۹	۹۶۷۵۶۹	۹۴۹۶۲۱	۲,۰۵	۹۶۵۶۰۷	۹۴۶۲۱۰	۵	۱۰
۴۵	۱,۹۹	۹۹۲۸۹۱	۹۷۳۵۱۸	۱,۹۴	۹۸۷۲۰۹	۹۶۸۴۲۲	۵	۱۱
۴۵	۱,۰۶	۹۹۳۳۶۷	۹۸۲۹۴۸	۱,۸۵	۹۹۶۰۹۴	۹۷۸۰۰۱	۵	۱۲
۴۵	۱,۸۶	۱۰۲۰۷۸۵	۱۰۰۲۱۴۵	۲,۱۰	۱۰۱۷۱۷۲	۹۹۶۲۵۱	۵	۱۳
۴۵	۱,۹۴	۱۰۴۴۰۳۱	۱۰۲۴۱۶۲	۱,۸۴	۱۰۴۲۴۹۱	۱۰۲۳۶۵۶	۵	۱۴
۶۰	۲,۴۵	۱۶۷۱۹۷۸	۱۶۳۱۹۹۴	۲,۳۸	۱۶۶۳۸۲۲	۱۶۲۵۱۴۴	۶	۱۵
۶۰	۲,۴۹	۱۶۹۲۴۱۴	۱۶۵۱۲۹۷	۲,۶۴	۱۶۹۳۴۰۸	۱۶۴۹۸۵۲	۶	۱۶
۶۰	۲,۸۶	۱۸۱۴۶۶۴	۱۷۶۴۲۰۸	۲,۷۵	۱۸۰۷۹۴۲	۱۷۵۹۵۵۴	۶	۱۷
۶۰	۲,۶۷	۱۸۶۲۵۶۷	۱۸۱۴۱۳۰	۲,۴۵	۱۸۵۴۳۵۷	۱۸۱۰۰۱۲	۶	۱۸
۶۰	۲,۷۴	۱۹۰۰۳۵۵	۱۸۴۹۶۷۴	۲,۶۴	۱۸۹۴۹۹۲	۱۸۴۶۲۵۱	۶	۱۹
۶۰	۲,۳۷	۱۹۰۹۰۴۸	۱۸۶۴۸۵۱	۲,۴۳	۱۹۰۸۴۲۲	۱۸۶۳۱۴۸	۶	۲۰
۶۰	۲,۶۱	۱۹۸۲۴۰۹	۱۹۳۱۹۸۴	۲,۳۱	۱۹۶۹۹۳۲	۱۹۲۵۴۵۴	۶	۲۱

۱۲. تبدیل رتبه و انحراف رتبه به مقادیر اصلی:

۸. محاسبه‌ی بهبود ایجادشده در رتبه:

$$R' = ۱۷۳,۷۱۷۸\mu' + ۱۵۰۰ \quad (۵۶)$$

$$RD' = ۱۷۳,۷۱۷۸\phi' \quad (۵۷)$$

$$\Delta = \nu \sum_{l=1}^L \sum_{n=1}^N \sum_{j=1}^J g(\phi_j) (S_{j,n,l} - E(\mu, \mu_j, \phi_j)) \quad (۵۱)$$

۱۳. تحلیل نتایج، رتبه بالاتر یک الگوریتم بیانگر عملکرد بهتر آن است.

۹. یافتن نوسانات جدید رتبه (σ') با به‌کارگیری الگوریتم Illinois. (گام‌های الگوریتم Illinois قابل مشاهده است.) تابع مورد استفاده به صورت زیر است:

پس از اجرای مراحل فوق برای تمام الگوریتم‌ها، فاصله‌ی اطمینان ۹۹,۷٪ به صورت $[R - 3RDR + RD]$ محاسبه می‌شود. اگر فاصله‌ی اطمینان دو الگوریتم باهم همپوشانی نداشته باشد، عملکرد دو الگوریتم به‌طور معنی‌داری متفاوت خواهد بود.

$$f(x) = \frac{e^x (\Delta^r - \phi^r - \nu - e^x)}{2(\phi^r + \nu + e^x)^r} - \frac{x - \ln(\sigma^r)}{r^2} \quad (۵۲)$$

برای به‌کارگیری روش CRS4EAs در این مقاله، ۲۱ مسئله از کلاس‌های ۴ تا ۶ هر یک ۶ بار با الگوریتم‌های DE و GA حل شد. طبق پیشنهاد گل‌کیم [۶۷] رتبه‌ی R، انحراف رتبه‌ی RD و نوسانات رتبه‌ی σ به ترتیب برابر ۱۵۰۰، ۳۵۰ و ۶۰۰، $\varepsilon_4 = ۱۰۰۰$ و $\varepsilon_5 = ۱۷۰۰$ در نظر گرفته شد. این بدان مفهوم است که مثلاً برای یک مسئله از کلاس ۴ اگر تفاوت جواب حاصل شده از اجرای دو الگوریتم کمتر از ۶۰۰ باشد نتیجه‌ی آن بازی مساوی در نظر گرفته می‌شود. جدول ۸ نتایج بازی‌ها را نشان می‌دهد.

از الگوریتم Illinois برای یافتن صفرهای تابع استفاده می‌شود. با فرض این که x یکی از نقاط صفر این تابع باشد، σ' برابر $e^{x/r}$ خواهد بود.

$$\phi^* = (\phi^r + \sigma'^r)^{r/5} \quad (۵۳)$$

۱۱. به‌روزرسانی رتبه و انحراف رتبه‌ی بازیکن i :

$$\phi^i = \left(\frac{1}{\phi^{*r}} + \frac{1}{\nu} \right)^{-r/5} \quad (۵۴)$$

همان‌طور که جدول ۸ نشان می‌دهد الگوریتم DE از مجموع ۱۲۶ بازی در مقابل GA، ۷۷ برد، ۱۱ تساوی و ۳۸ باخت داشته است. این نشان از برتری الگوریتم

$$\mu^i = \mu + \phi^{i,r} \sum_{l=1}^L \sum_{n=1}^N \sum_{j=1}^J g(\phi_j) (S_{j,n,l} - E(\mu, \mu_j, \phi_j)) \quad (۵۵)$$

جدول ۸. نتایج بازی‌های رودرویی DE و GA.

الگوریتم	نتیجه	DE	GA	مجموع
DE	برد	-	۷۷	۷۷
	باخت	-	۳۸	۳۸
	تساوی	-	۱۱	۱۱
GA	برد	۳۸	-	۳۸
	باخت	۷۷	-	۷۷
	تساوی	۱۱	-	۱۱

جدول ۹. رتبه‌ها، انحراف رتبه‌ها، و فواصل اطمینان.

الگوریتم	R	RD	RI
DE	۱۶۵۸	۴۵٫۸۶	[۱۷۹۵ - ۱۵۲۰]
GA	۱۳۴۲	۴۵٫۸۶	[۱۴۷۹ - ۱۲۰۴]

DE نسبت به GA دارد. برای تعیین معنی‌دار بودن این برتری روش CRS4EAs را به‌کار گرفته شد. رتبه‌ها، انحراف رتبه‌ها، و فواصل اطمینان ۹۹٫۷٪ در جدول ۹ ارائه شده است.

همان‌طور که در جدول ۹ مشخص است رتبه‌ی الگوریتم DE با ۱۸۵ واحد بهبود از ۱۵۰۰ اولیه به ۱۶۸۵ ارتقا پیدا کرده است. این در حالی است که رتبه‌ی GA به ۱۳۴۲ تنزل پیدا کرده است. از آنجایی که فواصل اطمینان دو الگوریتم همپوشانی ندارند، نتیجه می‌شود الگوریتم DE به‌طور معنی‌داری عملکرد بهتری نسبت به GA دارد.

۹. نتیجه‌گیری

توجه به مسائل زیست‌محیطی و استفاده‌ی مجدد از منابع مصرفی، موجب افزایش اهمیت مدیریت زنجیره‌ی تأمین معکوس شده است. از آنجاکه بسیاری از شرکت‌های در حال فعالیت، زنجیره‌ی تأمین خود را به‌صورت مستقیم طراحی کرده‌اند، ورود مراکز جمع‌آوری به زنجیره‌ی تأمین برای دست‌یابی به مزایای CLSC به بازطراحی زنجیره نیاز دارد. از طرفی گنجاندن مراکز جمع‌آوری در شبکه‌ی موجود، نیاز به سرمایه‌گذاری و کسب فناوری بازیافت دارد که بسیاری از شرکت‌های کوچک توان اقتصادی این کار را ندارند. از این‌رو، دولت‌ها با اعمال فشار و تدوین مشوق‌ها به

این امر ورود می‌کنند. بر این اساس در این مقاله به ارائه‌ی یک مدل CLSC برای بازطراحی شبکه‌ی حمل‌ونقل، تعیین محل مراکز جمع‌آوری و تعیین سطح بازیابی پرداختیم. مدل ارائه‌شده یک مدل غیرخطی است که با توجه به هزینه‌های سیستم و مقدار پاداش یا جریمه‌ی سیاست‌گذار، سطح فعالیت بازیابی را مشخص می‌کند. برای واقعی‌تر ساختن مدل، پارامترها غیرقطعی و به‌صورت اعداد فازی مثلثی در نظر گرفته شد. همچنین به‌منظور افزایش قابلیت اطمینان تصمیم، از یک روش بهینه‌سازی استوار بهره‌گرفته شد. با توجه به عدم کارایی نرم‌افزار گمز در حل مدل در ابعاد بزرگ، از الگوریتم فراابتکاری تفاضل تکاملی استفاده شد. در الگوریتم DE پیشنهادی از یک روش نمایش جواب مبتنی بر اولویت استفاده شد. روش نمایش مبتنی بر اولویت به‌گونه‌ی به‌کار گرفته شد که کارایی بالاتری نسبت به روش استاندارد مبتنی بر اولویت داشته باشد. جواب متناظر با هر کروموزوم با یک روش رمزنگاری چندمرحله‌ی استخراج شد. همچنین به‌دلیل ماهیت ساختاری کروموزوم از دو عملگر جهش و دو عملگر ترکیب در فرایند الگوریتم DE استفاده شد. برای بهبود عملکرد الگوریتم DE، با استفاده از طراحی آزمایش‌ها به روش پاسخ سطح، پارامترها تنظیم شد. مقایسه‌ی جواب‌های ایجادشده توسط DE با جواب‌های دقیق گمز در مسائل کوچک، حاکی از انحراف کم جواب‌های DE داشت. ۲۸ مسئله از ۴ کلاس مختلف توسط گمز و DE حل شد که میانگین خطای DE برابر ۱٫۳۳٪، محاسبه شد. میانگین خطای DE در مسائل پیچیده‌ترین کلاس برابر ۲٫۲۳٪ حاصل شد. در ۶ مسئله از ۷ مسئله در ساده‌ترین کلاس، بهترین جواب DE برابر جواب بهینه بود و در ۴ مسئله از این مسائل تمام اجزای DE منجر به دست‌یابی به جواب بهینه شد. بیشترین زمان حل یک مسئله از پیچیده‌ترین کلاس برای DE، ۲۵ ثانیه ثبت شد در حالی که گمز نیاز به ۶۱۷۳ ثانیه زمان برای حل همان مسئله داشت. اگرچه مقایسه‌ی عملکرد DE با رویکرد دقیق گمز نشان از عملکرد خوب الگوریتم پیشنهادی داشت، برای ارزیابی بیشتر در مسائل با ابعاد بزرگ از الگوریتم ژنتیک استفاده شد. حل ۲۱ مسئله از ۳ کلاس مختلف نشان داد در ۸۵٪ مسائل، الگوریتم پیشنهادی نسبت به GA در تعداد اجزای مساوی و در زمان برابر جواب بهتری ایجاد می‌کند. همچنین در ۷۶٪ مسائل، میانگین جواب‌های DE وضعیت بهتری از GA داشت. با وجود این شاخص میانگین درصد انحراف نسبی نشان از برتری الگوریتم DE بر GA در پراکندگی جواب‌ها نداشت. برای افزایش اعتبار مقایسه‌ی دو الگوریتم، از یک سیستم رتبه‌بندی شطرنج براساس سیستم گلیکو-۲ استفاده شد. نتایج ۱۲۶ بازی نشان داد در ۶۹٪ مواقع، DE مغلوب GA نشده است. همچنین نتایج بیان‌گر این بود که الگوریتم DE در سطح اطمینان ۹۹٫۷٪ به‌طور معنی‌داری عملکرد بهتری نسبت به GA دارد.

پانوشته‌ها

1. closed loop supply chain
2. value-at-risk
3. differential evolution
4. genetic algorithm
5. tabu search
6. particle swarm optimization
7. modularity

8. feasibility robustness
9. optimality robustness
10. randomness
11. epistemic uncertainty
12. encoding
13. Prüfer number
14. Decoding
15. insert
16. two cut
17. response surface methodology

18. relative percent deviation
19. Tournament
20. uniform
21. partially-mapped crossover
22. swap
23. shift
24. chess rating system for evolutionary algorithms
25. null hypothesis significance testing

منابع (References)

1. Goetschalckx, M., *Supply Chain Engineering*, Springer Science & Business Media (2011).
2. Rogers, D.S., Tibben-Lembke, R.S. and Council, R.L.E., *Going Backwards: Reverse Logistics Trends and Practices*, Reverse Logistics Executive Council Pittsburgh (1999).
3. Üster, H., Easwaran, G., Akçali, E. and Cetinkaya, S. "Benders decomposition with alternative multiple cuts for a multi-product closed-loop supply chain network design model", *Naval Research Logistics (NRL)*, **54**(8), pp. 890-907 (2007).
4. González-Torre, P.L. and Adenso-Díaz, B. "Reverse logistics practices in the glass sector in Spain and Belgium", *International Business Review*, **15**(5), pp. 527-546 (2006).
5. Krikke, H. "Impact of closed-loop network configurations on carbon footprints: A case study in copiers", *Resources, Conservation and Recycling*, **55**(12), pp. 1196-1205 (2011).
6. Biehl, M., Prater, E. and Realff, M.J. "Assessing performance and uncertainty in developing carpet reverse logistics systems", *Computers & Operations Research*, **34**(2), pp. 443-463 (2007).
7. Olugu, E.U. and Wong, K.Y. "An expert fuzzy rule-based system for closed-loop supply chain performance assessment in the automotive industry", *Expert Systems with Applications*, **39**(1), pp. 375-384 (2012).
8. Kusumastuti, R.D., Piplani, R. and Lim, G.H. "Re-designing closed-loop service network at a computer manufacturer: A case study", *International Journal of Production Economics*, **111**(2), pp. 244-260 (2008).
9. Morana, R. and Seuring, S. "Organizing a closed loop supply chain for used textiles—The ECOLOG case study", *Strategy and Organization in Supply Chains*, **37**(2), pp. 369-384 (2003).
10. Govindan, K., Soleimani, H. and Kannan, D. "Reverse logistics and closed-loop supply chain: A comprehensive review to explore the future", *European Journal of Operational Research*, **240**(3), pp. 603-626 (2015).
11. Wang, W., Zhang, Y., Zhang, K., Bai, T. and Shang, J. "Reward-penalty mechanism for closed-loop supply chains under responsibility-sharing and different power structures", *International Journal of Production Economics*, **170**, pp. 178-190 (2015).
12. Webster, S. and Mitra, S. "Competitive strategy in remanufacturing and the impact of take-back laws", *Journal of Operations Management*, **25**(6), pp. 1123-1140 (2007).
13. Plambeck, E. and Wang, Q. "Effects of e-waste regulation on new product introduction", *Management Science*, **55**(3), pp. 333-347 (2009).
14. Aksen, D., Aras, N. and Karaarslan, A.G. "Design and analysis of government subsidized collection systems for incentive-dependent returns", *International Journal of Production Economics*, **119**(2), pp. 308-327 (2009).
15. Atasu, A., Van Wassenhove, L.N. and Sarvary, M. "Efficient take-back legislation", *Production and Operations Management*, **18**(3), pp. 243-258 (2009).
16. Rahman, S. and Subramanian, N. "Factors for implementing end-of-life computer recycling operations in reverse supply chains", *International Journal of Production Economics*, **140**(1), pp. 239-248 (2012).
17. Jacobs, B.W. and Subramanian, R. "Sharing responsibility for product recovery across the supply chain", *Production and Operations Management*, **21**(1), pp. 85-100 (2012).
18. Govindan, K., Popiuc, M.N. and Diabat, A. "Overview of coordination contracts within forward and reverse supply chains", *Journal of Cleaner Production*, **47**, pp. 319-334 (2013).
19. Soyster, A. "Convex programming with set-inclusive constraints and applications to inexact linear programming", *Operations Research*, **21**, pp. 1154-1157 (1973).
20. Inuiguchi, M. and Sakawa, M. "Minimax regret solution to linear programming problems with an interval objective function", *European Journal of Operational Research*, **86**, pp. 526-536 (1995).
21. Mulvey, J., Vanderbei, R. and Zenios, S. "Robust optimization of large-scale systems", *Operations Research*, **43**, pp. 264-81 (1995).
22. Ben-Tal, A. and Nemirovski, A. "Robust convex optimization", *Mathematics of Operations Research*, **2**, pp. 769-805 (1998).
23. Ben-Tal, A. and Nemirovski, A. "Robust solutions of linear programming problems contaminated with uncertain data", *Mathematical Programming*, **88**, pp. 411-424 (2000).
24. Pishvaei, M.S., Razmi, J. and Torabi, S.A. "Robust possibilistic programming for socially responsible supply chain network design: A new approach", *Fuzzy Sets and Systems*, **206**, pp. 1-20 (2012).
25. Pishvaei, M.S. and Khalaf, M.F. "Novel robust fuzzy mathematical programming methods", *Applied Mathematical Modelling*, **40**(1), pp. 407-418 (2016).
26. Xu, J. "Multi-objective operation model for supply chain with uncertain prices based on fuzzy sets and robust optimization", *Advances Information Sciences and Service Sciences*, **5**(2), pp. 320-327 (2013).
27. Bai, X. and Liu, Y. "Robust optimization of supply chain network design in fuzzy decision system", *Journal of Intelligent Manufacturing*, **27**(6), pp. 1131-1119 (2014).
28. Ghodrattnama, A., Tavakkoli-Moghaddam, R. and Azaron, A. "Robust and fuzzy goal programming optimization approaches for a novel multi-objective hub location-allocation problem: A supply chain overview", *Applied Soft Computing*, **37**, pp. 255-276 (2015).

29. Vahdani, B. and Mohammadi, M. "A bi-objective interval-stochastic robust optimization model for designing closed loop supply chain network with multi-priority queuing system", *International Journal of Production Economics*, **170**, pp. 67-87 (2015).
30. Talaei, M., Moghaddam, B.F., Pishvae, M.S., Bozorgi-Amiri, A. and Gholamnejad, S. "A robust fuzzy optimization model for carbon-efficient closed-loop supply chain network design problem: A numerical illustration in electronics industry", *Journal of Cleaner Production*, **113**, pp. 662-673 (2015).
31. Storn, R. and Price, K. "Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces", *Journal of Global Optimization archive*, **11**(4), pp. 341-359 (1995).
32. Pan, Q.-K., Tasgetiren, M.F. and Liang, Y.-C. "A discrete differential evolution algorithm for the permutation flowshop scheduling problem", *Computers & Industrial Engineering*, **55**(4), pp. 795-816 (2008).
33. Yeh, M.F. "Applying differential evolution to aggregate production planning", *Universal Journal of Industrial and Business Management*, **2**(7), pp. 164-172 (2014).
34. Prasertwattana, K. and Shimizu, Y. "Optimization of material ordering and inventory control of supply chain through an incentive scheme using differential evolution", *Journal of Japan Industrial Management Association*, **59**(4), pp. 283-289 (2008).
35. Chan, F., Routroy, S. and Kodali, R. "Differential evolution algorithm for supply chain inventory planning", *Journal of Manufacturing Technology Management*, **16**(1), pp. 7-17 (2005).
36. Babu, B. and Gujarathi, A.M. "Multi-objective differential evolution (MODE) for optimization of supply chain planning and management", *Congress on Evolutionary Computation*, pp. 2732-2739 (2007).
37. Ding, S.-B., Huang, W.-L. and Zhang, Z.-G. "Supply Chain Network Design Optimization Based on Differential Evolution Algorithm", *Logistics Technology*, **11**, pp. 26-31 (2008).
38. Xu, J.M., Xiong, J.Z., Chen, Y. and Hu, G.W. "Supply chain optimization using migration differential evolution ensemble", *International Conference on Intelligent Computing and Intelligent Systems*, pp. 755-759 (2010).
39. Fatrias, D. and Shimizu, Y. "Multi-objective analysis of periodic review inventory problem with coordinated replenishment in two-echelon supply chain system through differential evolution", *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, **4**(3), pp. 637-650 (2010).
40. Gholizadeh Tayyar, S., Roy, D. and Ghaderi, S. "Economic, environmental and social responsible supply chain design using differential evolution multi objective algorithm", *International Conference on Industrial Engineering and Engineering Management*, pp. 1617-1621 (2013).
41. Lieckens, K. and Vandaele, N. "Differential evolution to solve the lot size problem in stochastic supply chain management systems", *Annals of Operations Research*, **224**, pp. 1-25 (2015).
42. Yıldız, Y.E. and Altun, O. "Chemotaxis differential evolution optimization algorithm (CDEOA) for minimization of supply chain cost with embedded risk", *International Conference on Image Processing, Production and Computer Science*, Istanbul, Turkey, pp. 31-37 (2015).
43. Beamon, B.M. "Designing the green supply chain", *Logistics Information Management*, **12**(4), pp. 332-342 (1999).
44. Krikke, H., le Blanc, I. and van de Velde, S. "Product modularity and the design of closed-loop supply chains", *California Management Review*, **46**(2), pp. 23-39 (2004).
45. Zareian-Jahromi, H., Fallahnezhad, M.S., Sadeghieh, A. and Ahmadi-Yazdi, A. "A robust multi objective optimization model for sustainable closed-loop supply chain network design", *Journal Of Industrial Engineering Research In Production Systems*, **2**(3), pp. 93-111 (2014).
46. Pishvae, M.S., Rabbani, M. and Torabi, S.A. "A robust optimization approach to closed-loop supply chain network design under uncertainty", *Applied Mathematical Modelling*, **35**(2), pp. 637-649 (2011).
47. Kannan, G., Sasikumar, P. and Devika, K. "A genetic algorithm approach for solving a closed loop supply chain model: A case of battery recycling", *Applied Mathematical Modelling*, **34**(3), pp. 655-670 (2010).
48. Mohseni, S., Pishvae, M.S. and Sahebi, H. "Robust design and planning of microalgae biomass-to-biodiesel supply chain: A case study in Iran", *Energy*, **111**, pp. 736-755 (2016).
49. Mula, J., Poler, R. and Garcia, J. "MRP with flexible constraints: A fuzzy mathematical programming approach", *Fuzzy Sets and Systems*, **157**(1), pp. 74-97 (2006).
50. Saha, A., Kar, S. and Maiti, M. "Multi-item fuzzy-stochastic supply chain models for long-term contracts with a profit sharing scheme", *Applied Mathematical Modelling*, **39**(10), pp. 2815-2828 (2015).
51. Gholamian, N., Mahdavi, I., Tavakkoli-Moghaddam, R. and Mahdavi-Amiri, N. "Comprehensive fuzzy multi-objective multi-product multi-site aggregate production planning decisions in a supply chain under uncertainty", *Applied Soft Computing*, **37**, pp. 585-607 (2015).
52. Pramanik, S., Jana, D.K., Mondal, S.K., and Maiti, M. "A fixed-charge transportation problem in two-stage supply chain network in Gaussian type-2 fuzzy environments", *Information Sciences*, **325**, pp. 190-214 (2015).
53. Talaei, M., Moghaddam, B.F., Pishvae, M.S., Bozorgi-Amiri, A. and Gholamnejad, S. "A robust fuzzy optimization model for carbon-efficient closed-loop supply chain network design problem: A numerical illustration in electronics industry", *Journal of Cleaner Production*, **113**, pp. 662-673 (2016).
54. Afrouzy, Z.A., Nasser, S.H., Mahdavi, I. and Paydar, M.M. "A fuzzy stochastic multi-objective optimization model to configure a supply chain considering new product development" *Applied Mathematical Modelling*, **40**(17-18), pp. 7545-7570 (2016).
55. Azadeh, A., Raoofi, Z. and Zarrin, M. "A multi-objective fuzzy linear programming model for optimization of natural gas supply chain through a greenhouse gas reduction

- approach”, *Journal of Natural Gas Science and Engineering*, **26**, pp. 702-710 (2015).
56. Jimenez, M., Arenas, M., Bilbao, A. and Rodriguez, M.V. “Linear programming with fuzzy parameters: An interactive method resolution”, *European Journal of Operational Research*, **177**, pp. 1599-1609 (2007).
 57. Pishvaei, M.S. and Torabi, S.A. “A possibilistic programming approach for closed-loop supply chain network design under uncertainty”, *Fuzzy Sets Syst*, **161**, pp. 2668-2683 (2010).
 58. Cheng, R. and Gen, M., *Genetic Algorithms and Engineering Design*, Wiley, New York (1997).
 59. Kuk, G. “Effectiveness of vendor-managed inventory in the electronics industry: Determinants and outcomes”, *Information & Management*, **41**, pp. 645-654 (2004).
 60. Nachiappan, S.P. and Jawahar, N. “A genetic algorithm for optimal operating parameters of VMI system in a two-echelon supply chain”, *European Journal of Operational Research*, **182**, pp. 1433-1452 (2007).
 61. Gen, M., Altiparmak, F. and Lin, L. “A genetic algorithm for two-stage transportation problem using priority-based encoding”, *OR Spectrum*, **28**(3), pp. 337-354 (2006).
 62. Das, S. and Suganthan, P.N. “Differential evolution: A survey of the state-of-the-art”, *Evolutionary Computation*, *IEEE Transactions on*, **15**(1), pp. 4-31 (2011).
 63. Atabaki, M.S. and Mohammadi M. “A genetic algorithm for integrated lot sizing and supplier selection with defective items and storage and supplier capacity constraints”, *International Journal of Operational Research*, **28**(2), pp. 183-200 (2017).
 64. Miller, B. and Goldberg, D. “Genetic algorithms, tournament selection, and the effects of noise”, *Complex Systems*, **9**, pp. 193-212 (1995).
 65. Goldberg, D.E. and Lingle, R. “Alleles, loci, and the traveling salesman problem”, *International Conference on Genetic Algorithms and Their Applications*, pp. 154-159 (1985).
 66. Veček, N., Mernik, M. and Črepinšek, M. “A chess rating system for evolutionary algorithms: A new method for the comparison and ranking of evolutionary algorithms”, *Information Sciences*, **277**, pp. 656-679 (2014).
 67. Vecek, N., Crepinsek, M., Mernik, M. and Hrcic, D. “A comparison between different chess rating systems for ranking evolutionary algorithms”, *Conference on Computer Science and Information Systems*, pp. 511-518 (2014).
 68. Dowell, M. and Jarratt, P. “A modified regula falsi method for computing the root of an equation”, *BIT Numerical Mathematics*, **11**(2), pp. 168-174 (1971).
 69. Atabaki, M.S., Mohammadi M. and Naderi B. “Hybrid genetic algorithm and invasive weed optimization via priority based encoding for location-allocation decisions in a three-stage supply chain” *Asia-Pacific Journal of Operational Research*, **34**(02), pp. 1-44 (2017).