

توسعه و حل یک مدل دو هدفه جهت تکمیل توأم موجودی چند کالا با محدودیت فضای انبار

ام البنین یوسفی^{۱*}، میربهادر قلی آریانزاد^۲، سید جعفر سجادی^۲

۱- دانشجوی دکتری مهندسی صنایع، دانشگاه علم و صنعت ایران و عضو هیأت علمی دانشگاه صنعتی مالک اشتر

۲- استاد دانشکده صنایع، دانشگاه علم و صنعت ایران

چکیده

در این تحقیق، یک مدل دو هدفه تکمیل موجودی همزمان چند کالا با فرض محدودیت یکی از منابع، توسعه یافته و حل شده است. مدل پیشنهادی دارای یک محدودیت فضای انبار بوده و به دنبال بهینه کردن دو تابع هدف می باشد. این دو تابع عبارتند از کمینه سازی مجموع هزینه نگهداری و سفارش دهی سالیانه و کمینه سازی کل سرمایه سالانه درگیر در موجودی. سپس یک الگوریتم ژنتیک چند هدفه برای حل آن توسعه یافته است. به منظور بررسی کارایی الگوریتم، عملکرد آن در اجرای ۱۶۰۰ مسأله که به صورت تصادفی ایجاد شده اند و با پارامترهایی که از ادبیات موضوع استخراج شده اند، بررسی شده است. یافته نشان دهنده آن است که الگوریتم پیشنهادی توانایی خوبی در ارائه جوابهای بهینه پارتویی دارد. در پایان، کاربرد رویکرد حل مسأله و نتایج حاصل از الگوریتم پیشنهادی در مورد یک مسأله خاص که به صورت تصادفی تولید شده، نشان داده شده است.

واژه های کلیدی: مسأله تکمیل توأم موجودی چند کالا، بهینه سازی چند هدفه، محدودیت فضای انبار، الگوریتم ژنتیک

مقدمه

در عمل مدل‌های موجودی مختلفی وجود دارد که با چندین محصول در ارتباطند، هدف این مدل‌ها، اغلب مینیم کردن هزینه کل بوده در حالیکه تقاضا نیز برآورده شود. هزینه کل، معمولاً از دو جزء تشکیل می‌شود:

الف) هزینه راه اندازی یا سفارش دهی^۱

این هزینه، برای محصولاتی که ساخته می‌شوند در واقع هزینه آماده سازی ماشین آلات و تجهیزات، قبل از تولید است و برای محصولاتی که به یک تأمین کننده سفارش داده می‌شوند شامل هزینه آماده سازی و دریافت سفارش و نیز هزینه حمل و نقل است.

ب) هزینه نگهداری^۲

این هزینه، همان هزینه نگهداری موجودی است که شامل هزینه درگیر در موجودی، مالیات، بیمه و نظایر آن می‌باشد.

مهمترین مسأله در مسائل چند محصولی عبارت است از تصمیم گیری روی مقادیر بهینه محصولات مختلف در یکی از حالات زیر: (گویال^۳، ۱۹۷۴)

۱- هنگامی که چندین کالا به یک تأمین کننده سفارش داده می‌شوند.

۲- هنگامی که چندین محصول از تجهیزات حمل و نقل مشترک استفاده می‌کنند.

۳- هنگامی که یک محصول بعد از تولید انبوه یا دسته ای در مقادیر مختلف بسته بندی شود.

علیرغم هزینه های مرتبط با موجودیها، داشتن موجودی در هر کارخانه امری غیر قابل اجتناب است. مسأله مهم این است که هزینه های روبرو شدن با کمبود کالا و مواد اولیه و قطعات یدکی، مشکلات توقف تولید و از دست دادن فرصت فروش کالا و کسر اعتبار شرکت را در بر خواهد داشت. در مواردی این اشکالات می تواند به مراتب بیشتر از هزینه های ذخیره موجودی باشد. هدف اصلی امور برنامه ریزی و کنترل موجودیها آن است که با تجزیه و تحلیل شرایط و هزینه ها، مناسب ترین سیاستها را برای سفارش و نگهداری موجودی در کارخانه اتخاذ نماید. از جمله مهمترین تقسیمات مدل‌های موجود در مباحث کنترل موجودی مدل‌های تک کالایی و مدل های چند کالایی است. در مدل‌های تک کالایی همواره وضعیت موجودی و سیاستهای سفارشات کالاها به صورت منفرد و مستقل از سایر کالاهای موجود در انبار بررسی می‌شوند. در عمل، بسیاری از موارد دیده می‌شود که بین کالاهای مختلف موجود در انبار و مورد استفاده سازمان می‌تواند انواع وابستگی ها وجود داشته باشد، در نتیجه معمولاً سیاستهای اتخاذ شده برای یک کالا بر سایر کالاهای انبار تأثیرگذار خواهند بود. اثرات متقابل کالاها بر یکدیگر می‌تواند به عوامل زیادی مرتبط باشد به عنوان مثال:

✓ تأثیرات متقابل به علت محدودیتها

✓ تأثیرات متقابل در هزینه های سفارش دهی

✓ تأثیرات متقابل در قیمت‌های خرید

✓ تأثیرات متقابل در محدودیتهای تولید

1 Set up or ordering cost

2 Holding cost

3 Goyal

زمان سیکل پایه گفته می شود. هر محصول، ضریب عدد صحیحی از زمان سیکل پایه را داراست و در سیکلهای سفارشی که مضرب صحیح از این ضریب است سفارش داده می شود. گروهها در استراتژی IGS، بصورت غیر مستقیم، با محصولاتی که دارای یک ضریب صحیح یکسان هستند، تشکیل می شود. آرکین^۷ و همکاران(۱۹۸۹)، ثابت کردند که مسأله JRP، یک مسأله با پیچیدگی سخت^۸ است، بنابراین الگوریتمی که در یک زمان چند جمله ای بتواند مسأله JRP مخصوصاً با ابعاد بزرگ را حل کند وجود ندارد و لذا در مقالات بسیاری، با استفاده از الگوریتمهای فراابتکاری، روشهای حل تقریبی برای مسأله توسعه داده شده است. برای اولین بار گویال(۱۹۷۴) با محاسبه یک حد بالا و پایین روی زمان سیکل پایه یک الگوریتم محاسباتی، برای مشخص کردن کلیه می نیم های محلی و در نتیجه می نیم کل ارائه داد. بنابراین رویکرد گویال منجر به یافتن جواب بهینه برای مسأله JRP می شد ولی ممکن بود از لحاظ محاسباتی برای مسائل بزرگ به جواب نرسد. سیلور^۹ (۱۹۷۶) یک الگوریتم کارا برای برای حل JRP، ارائه داد که در سال ۱۹۷۹ توسط گویال بهبود داده شده، سپس توسط کسپی و روزنبلات^{۱۰} (۱۹۸۳) توسعه بیشتری یافت. این الگوریتم شاید معروفترین روش هیوریستیک در حل JRP بوده و به نام الگوریتم RAND معروف است. این الگوریتم بر اساس محاسبه m مقدار مساوی برای زمان سیکل پایه از طریق حد پائین و بالای آن یعنی

هزینه ترتیب دادن یک سفارش از یک تأمین کننده برای تعدادی محصول متفاوت از دو جزء زیر تشکیل می شود:

الف) یک هزینه کلی سفارش دهی^۱ که از تعداد محصولات مختلف در یک سفارش مستقل است.

ب) یک هزینه جزئی سفارش دهی^۲ که به تعداد محصولات مختلف در یک سفارش بستگی دارد.

مسأله فوق به «مسأله تکمیل همزمان موجودی^۳» یا JRP معروف است. به علت وجود هزینه کلی سفارش دهی، بکارگیری سفارش دهی گروهی ممکن است منجر به صرفه جویی معنی داری در هزینه ها شود. این صرفه جویی بطور قابل توجهی وقتی تقاضای بین اقلام بطور نزدیکی به هم مرتبط هستند، افزایش می یابد (تسا و هوانگ^۴، ۲۰۰۹). بعلاوه صرفه جویی حاصل از محل سفارش دهی گروهی هر چه هزینه کلی سفارش دهی بزرگتر باشد، معنی دارتر است.

استراتژیهای حل JRP می تواند به دو نوع تقسیم شود:

۱- استراتژی گروهبندی مستقیم^۵ (DGS)

۲- استراتژی گروهبندی غیرمستقیم^۶ (IGS)

در استراتژی DGS، محصولات به تعدادی گروه از قبل مشخص شده، تقسیم می شوند و محصولات داخل هر گروه، با یک زمان سیکل مشترک سفارش داده می شوند. در استراتژی IGS، سفارش دهی در فواصل زمانی منظم انجام می شود که معمولاً به آن

1 Major ordering cost

2 Minor ordering cost

3 Joint Replenishment Problem (JRP)

4 Tsai and Huang

5 Direct Grouping Strategy (DGS)

6 Indirect Grouping Strategy (IGS)

7 Arkin

8 NP-hard

9 Silver

10 Kaspi and Rossenblatt

✓ در نظر گرفتن تخفیف های مقداری کلی
 ✓ در نظر گرفتن قیمت هر واحد محصولات
 به صورت کاهشی یا افزایشی نسبت به زمان
 ✓ بکارگیری JRP بعنوان یک زیر مسأله در
 سایر مسائل
 ✓ در نظر گرفتن وابستگی هزینه های سفارش
 دهی به ترکیب موجود در کالاهایی که سفارش
 داده می شوند.
 ✓ در نظر گرفتن محدودیتهایی در مدل نظیر
 ظرفیتهای انبار و حمل و نقل و بودجه
 ✓ در نظر گرفتن تقاضای نا معین مشتری و
 همچنین تخمین های نا دقیق واحد هزینه
 نگهداری
 در زمینه توسعه مدل JRP کلاسیک در شرایط
 خاص که تحقیق حاضر نیز به آن متعلق است، موارد
 زیر از ادبیات موضوع قابل استخراج می باشد، کلین
 و ونچورا^۳ (۱۹۹۵)، جهت ایجاد یک سیاست
 سفارش دهی ساده تر برای پیاده سازی در صنعت
 مدلی را ارائه دادند که در آن زمانهای تکمیل
 موجودی تنها به ابتدای پروده های زمانی گسسته
 محدود شد. مون و چا^۴ (۲۰۰۵)، دو الگوریتم برای
 حل JRP، وقتی تأمین کننده تخفیف های مقداری
 کلی پیشنهاد می کند، توسعه دادند. نویسندگان دو
 قضیه اثبات کردند که سپس در توسعه دو الگوریتم
 برای پیدا کردن جواب بکار بردند. خوجا و
 همکاران^۵ (۲۰۰۵) فرض کردند که هزینه هر واحد
 محصولات در یک افق زمانی محدود و متناسب با
 یک آهنگ پیوسته نسبت به زمان کاهشی یا افزایشی

روی فاصله، $[T_{min} \quad T_{max}]$ و سپس بکارگیری
 الگوریتم سیلور جهت بهبود الگوریتم برای هر مقدار
 از زمان سیکل پایه می باشد.

بررسی تحقیقات انجام شده

تحقیقاتی که تا کنون روی مسأله JRP انجام شده
 است را می توان به دو دسته کلی زیر تقسیم کرد:

✓ ارائه روشهای مختلف جهت حل مدل JRP
 کلاسیک

✓ توسعه مدل های جدید از روی مسأله JRP
 کلاسیک و ارائه روشهای مختلف جهت حل آنها
 که در این مورد عمدتاً موارد زیر در تحقیقات
 گذشته به چشم می خورد:

الف- توسعه « مدل تکمیل موجودی توأم چند
 کالا در حالت احتمالی^۱ » یا SJRP که در آن تقاضای
 تقاضای محصولات به صورت احتمالی در نظر
 گرفته می شود و هدف مینیم کردن هزینه مورد
 انتظار در هر واحد زمان می باشد.

ب- توسعه « مدل تکمیل موجودی توأم چند کالا
 در حالت پویا^۲ » یا DJRP که در آن تقاضای قطعی
 است ولی نسبت به زمان یکنواخت نیست و هدف
 عبارتست از مینیم کردن هزینه کل در افق برنامه
 ریزی که شامل چند پرپود می باشد.

ج- توسعه یکی از مدل های JRP یا SJRP یا DJRP
 با در نظر گرفتن شرایط خاص بعنوان مثال:

✓ در نظر گرفتن زیر مجموعه ای از پرپودهای
 زمانی گسسته جهت تکمیل موجودی

3 Klein and Ventura

4 Moon and Cha

5 Koudja et al

1 Stochastic demand joint replenishment
 problem(SJRP)

2 Dynamic demand joint replenishment problem
 (DJRP)

دقیق واحد هزینه نگهداری پرداختند. سو^۷ (۲۰۰۹)، مسأله JRP را برای مدلسازی تصمیمات موجودی برای یک کارخانه مرکزی و شعبه های آن در یک سیستم تولید بهنگام بکار گرفت. لی^۸ (۲۰۰۴) یک مدل JRP وقتی چندین خریدار وجود دارند مدلسازی کرده و با توسعه ای از الگوریتم RAND آنرا حل نمود. اکثر مدل‌های موجودی چندین مفهوم هزینه ای و نیز احتیاجات سطح سرویس دهی به مشتری را در داخل یک تابع هدف تکی جایگزین می کنند، سپس تصمیمات بهینه در مورد اینکه چقدر سفارش داده شود و کی سفارش داده شود، توسط روشهای سنتی بهینه سازی آغاز می شود، در حالیکه موارد زیادی در مسائل مربوط به سیستم های واقعی کنترل موجودی وجود دارد که تصمیم گیرندگان مایل به بهینه سازی بیش از یک هدف که بعضاً ممکن است با یکدیگر در تضاد نیز باشند هستند. در مورد مسأله JRP، تا کنون به حالتی که بیش از یک هدف در مدل در نظر گرفته شده و در نتیجه بر کارایی کاربردی و عملی آن بیفزاید، پرداخته نشده است. در مقاله حاضر به مدلسازی مسأله JRP با در نظر گرفتن دو هدف پرداخته شده است و از طرفی جهت کاربردی تر شدن مدل، محدودیت فضای انبار نیز که در عمل اکثر سیستمهای کنترل موجودی با آن مواجه هستند نیز به مدل اضافه شده است. هدف دیگر مقاله حاضر، ارائه یک رویکردی کارا برای حل مسأله پیشنهادی است که به این منظور به توسعه الگوریتم ژنتیک چند هدفه که کارایی خوبی برای بهینه سازی مسائل چند هدفه از خود نشان داده است پرداخته و

است و یک الگوریتم برای حل JRP ارائه دادند. سیاجادی^۱ و همکاران (۲۰۰۵)، مدلی را فرموله کرد که در آنها JRP بعنوان یک زیر مسأله بکار رفت، آنها مدل بهینه سازی تصمیمات موجودی یک فروشنده که بطور همزمان چندین قطعه را برای تولید محصول نهایی بکار می برد، را توسعه دادند. چان^۲ و همکاران (۲۰۰۶)، یک رویکرد حل برای برنامه ریزی تحویل از یک تأمین کننده به چندین خریدار توسعه دادند که خریداران JRP را برای تکمیل موجودی خود از تأمین کننده بکار می برند و در ادامه مسأله برنامه ریزی تحویل تحت ۴ هدف مختلف فرموله شد. هوک^۳ (۲۰۰۶)، یک مدل توسعه یافته از JRP برای در برگرفتن موارد عملی با در نظر گرفتن ظرفیتهای انبار و حمل و نقل و محدودیت بودجه ارائه داده است و یک رویکرد ساده برای محاسبه حد پایین مناسب برای سیکل مشترک پایه ارائه داد. السن^۴ (۲۰۰۸)، حالتی از مسأله JRP را در نظر گرفت که در آن هزینه های جزئی سفارش دهی به یکدیگر وابسته اند. وابستگی خطی هزینه های جزئی سفارش دهی در JRP، هنگامی رخ می دهد که هزینه وارد کردن یک آیتم در یک سفارش به اینکه کدامیک از کارهای دیگر در سفارش هستند، بستگی دارد و در ادامه، یک الگوریتم تکاملی^۵ (EA) برای حل چنین مسأله ای ارائه داد. در مقاله ای که وانگ^۶ و همکاران (۲۰۰۸) ارائه دادند به مطالعه ای از مسأله JRP تحت تقاضاهای نامعین مشتری و تخمین نا

1 Siadjadi et al
2 Chan et al
3 Hoque
4 Olsen
5 Evolutionary Algorithms
6 Wang et al

7 Hsu
8 Li

«مقدار سفارش اقتصادی^۱» یا همان EOQ است که این فرضیات عبارتند از:

- ۱- وجود تقاضای قطعی و یکنواخت
- ۲- عدم وجود تخفیف
- ۳- خطی بودن هزینه نگهداری
- ۴- عدم مجاز بودن کمبود
- علاوه بر آن فرضیات زیر نیز اضافه می شوند:
- ۵- استفاده از استراتژی IGS
- ۶- محدودیت فضای انبار

پارامترهای مدل

n : تعداد محصولات
 i : اندیس محصولات $i = 1, \dots, n$
 D_i : تقاضای سالیانه محصول i ام
 h_i : هزینه نگهداری سالیانه برای محصول i ام
 S : هزینه کلی سفارش دهی در هر بار سفارش
 s_i : هزینه جزئی سفارش دهی که در صورت سفارش محصول i ام سفارش داده شود پرداخت می شود
 c_i : هزینه خرید یک واحد محصول i ام
 V : ماکزیمم فضای انبار
 v_i : فضای لازم برای انبار کردن یک واحد کالای i ام

متغیرهای تصمیم

Q_i : میزان سفارش محصول i ام (واحد کالا)
 T : زمان بین دو سفارش متوالی یا زمان سیکل پایه

نتایج حاصل نشان داده شده است. روش انجام تحقیق حاضر، روش اسنادی و مطالعات کتابخانه ای یا مراجعه به منابع در دسترس در خصوص موضوعات مرتبط با موضوع تحقیق می باشد. همچنین استفاده از نرم افزارهای محاسباتی و برنامه نویسی پیچیده و بهره گیری از امکانات کامپیوتر مناسبت در اجرای مدل توسعه داده شده ضروری خواهد بود. در ادامه مطالب ارائه شده در مقاله حاضر در ۵ بخش ارائه می شود. در قسمت سوم به معرفی مدل پیشنهادی می پردازیم. به معرفی الگوریتم ژنتیک چند هدفه در بخش چهارم پرداخته می شود. توسعه الگوریتم ژنتیک چند هدفه برای حل مدل پیشنهادی در قسمت پنجم انجام می شود. در قسمت ششم نتایج حاصل از حل مسأله با روش پیشنهادی ارائه می شود و در قسمت هفتم نیز به جمع بندی، نتیجه گیری و ارائه پیشنهاد برای ادامه تحقیق در مورد این موضوع می پردازیم.

معرفی مدل دو هدفه تکمیل توأم موجودی چند کالا با وجود محدودیت فضا

در این قسمت به معرفی فرضیات، پارامترها و متغیرهای تصمیم در مدل پیشنهادی پرداخته و سپس مدل نهایی ارائه می شود.

فرضیات

فرضیات مدل، شبیه فرضیات در مدل JRP کلاسیک می باشد که آن هم شبیه فرضیات مدل

$$T \geq 0, k_i \geq 1, k_i \text{ is integer } \forall i = 1, 2, \dots, n$$

توسعه الگوریتم ژنتیک چند هدفه برای حل مدل پیشنهادی

بهینه سازی مسائل چند هدفه

یک دسته از سخت ترین و در عین حال پرکاربردترین مدلها در مسائل بهینه سازی، مسائل چند هدفه هستند. در یک مسأله بهینه سازی چند هدفه، معمولاً یک جواب بهینه منحصر به فرد وجود دارد ولی در مسائل بهینه سازی چند هدفه، توابع هدف ممکن است با یکدیگر در تعارض باشند، بنابراین پیدا کردن یک جواب بهینه بطوریکه بصورت همزمان کلیه اهداف را بهینه کند معمولاً امکان پذیر نیست. یکی از روشهای حل اینگونه مسائل ادغام کردن اهداف مختلف در یک تابع هدف است. در چنین مواردی جواب بهینه با استفاده از متدهایی نظیر تابع مطلوبیت و یا مجموع وزین توابع هدف بدست می آید، و لیکن در این روشها نیز انتخاب اوزان مناسب برای توابع هدف و یا انتخاب یک تابع مطلوبیت مناسب یک چالش محسوب می شود. از جمله روشهای دیگر حل مسائل چند هدفه، روشهایی هستند که هدفشان بدست آوردن جوابهای بهینه پارتویی و یا زیر مجموعه ای کارا از آن می باشد. مجموعه جوابهای پارتویی³ شامل جوابهایی می شود که توسط هیچ جواب دیگری از مجموعه جوابهای امکان پذیر مسأله چیره⁴ نمی شوند. یک جواب چیره ناپذیر جوابی است که بهبود در یک تابع هدف آن

T_i : فاصله زمانی بین دو سفارش متوالی محصول

ام i

k_i : ضریب عدد صحیح سفارش دهی محصول i

ام

متغیرهای تصمیم اصلی عبارتند از T و k_i ، بطوریکه مطابق روابط ۱ و ۲ متغیرهای Q_i و T_i از روی آنها قابل بدست آوردن هستند.

(۱)

$$T_i = k_i T$$

(۲)

$$Q_i = T_i D_i = T k_i D_i$$

توابع هدف

الف) هزینه کل سفارش دهی^۱ (TCS) و

نگهداری موجودی^۲ (TCH) در واحد زمان (f_1)

ب) هزینه کل سرمایه درگیر در موجودی (f_2)

مدل پیشنهادی

(۳)

$$\begin{aligned} \text{a) minimize } f_1(T, K_{1,n}) &= TCH + TCS \\ &= \frac{\sum_{i=1}^n k_i D_i h_i}{2} T + \frac{S + \sum_{i=1}^n \frac{S_i}{k_i}}{T} \end{aligned} \quad (4)$$

$$\text{b) minimize } f_2(T, K_{1,n}) = T \sum_{i=1}^n c_i k_i D_i \quad (5)$$

$$\text{subject to : } \sum_{i=1}^n k_i D_i v_i T \leq V$$

3 Pare to solution
4 Dominate

1 Total cost of set up
2 Total cost of holding

- ۱- توابع ادغامی^۴ که مجموع وزین توابع هدف را بعنوان یک تابع هدف کلی در نظر می گیرند.
- ۲- روشهای مبتنی بر جمعیت^۵، به عنوان یک مثال کلاسیک از این دسته^۶ VEGA^۶ را می توان نام برد.
- ۳- روشهای مبتنی بر جوابهای پارتویی^۷، یک مثال کلاسیک از این دسته^۸ MOGA^۸ است.

الگوریتم ژنتیک چند هدفه (MOGA)

در مکانیسم انتخاب در MOGA رتبه هر کروموزوم در جمعیت برابر است با تعداد جوابهایی که آنرا چیره می کنند. کل جوابهایی که چیره ناپذیرند دارای رتبه برابر هستند و در نتیجه شانس برابری برای انتخاب در تکرار بعد و تولید جمعیت جدید را دارند. MOGA به منظور ایجاد پراکندگی لازم در فضای جواب از رویکرد اشتراک برانندگی^۹ برای بدست آوردن جوابهایی که بطور یکنواخت در مجموعه جوابهای پارتو توزیع شده اند، استفاده می کند. در این تکنیک جوابهایی که در مناطقی قرار دارند که چگالی جوابهای یکسان در آنها زیاد است، برانندگیشان کاهش می یابد.

برای توسعه هر الگوریتم ژنتیک اجزاء اصلی و مهم الگوریتم بایستی به دقت تعریف شوند که در ادامه به معرفی آنها می پردازیم:

منجر به بدتر شدن حداقل یک تابع هدف دیگر می شود. با استفاده از چنین روشهایی و ارائه مجموعه جوابهای چیره ناپذیر، این فرد تصمیم گیرنده است که با توجه به تبادله^۱ بین اهداف به انتخاب جواب مطلوب خود می پردازد. (کناک^۲ و همکاران، ۲۰۰۶)

الگوریتمهای تکاملی چند هدفه^۳

در طی دهه گذشته الگوریتمهای مبتنی بر جمعیت مانند الگوریتمهای ژنتیک در بهینه سازی مسائل چند هدفه کاربرد زیادی پیدا کرده اند، چرا که این الگوریتمها می توانند مجموعه جوابهای پارتویی را با یک بار اجرای الگوریتم بدست آورند، در حالیکه سایر روشهای بهینه سازی سنتی با چند بار اجرای متوالی و جدا گانه به این مجموعه دست پیدا می کنند. (کناک و همکاران، ۲۰۰۶)

در یک الگوریتم تکاملی چند هدفه که یک توسعه از الگوریتم تکاملی است به دو سؤال اساسی بایستی پاسخ داده شود:

- ۱- مکانیسم انتخاب کروموزومها به نحوی که جوابهای چیره ناپذیر شانس بیشتری برای انتخاب و ایجاد نسل بعد را داشته باشند.
- ۲- مکانیسم ایجاد تنوع در جوابهای تولید شده به نحوی که حتی المقدور کل فضای جواب جستجو شده و تا حد امکان کل مجموعه جوابهای پارتویی ارزیابی شود.

الگوریتمهای تکاملی با توجه به مکانیسمی که برای پاسخ به هر یک از دو پرسش فوق انتخاب می کنند، متعلق به یکی از سه دسته زیر هستند:

4 Aggregate functions
 5 Population-based approaches
 6 Vector Evaluated Genetic Algorithm
 7 Pareto-based approaches
 8 Multi-Objective Genetic Algorithms (MOGAs)
 9 Fitness sharing

1 Trade off
 2 Konack et al
 3 Multi-Objective Evolutionary Algorithms

حدود متغیرهای تصمیم

همانطور که قبلاً نیز اشاره شد در استراتژی IGS که ما نیز از آن استفاده می کنیم متغیرهای تصمیم عبارتند از، زمان سیکل پایه (T) و ضرایب عدد صحیح مربوط به هر کالا ($k_{i,s}$). حد بالا و پایین مقادیر T را بصورت زیر در نظر می گیریم (خوجا و همکاران، ۲۰۰۰):

(۶)

$$T_{max} = \left[\frac{2(S + \sum_{i=1}^n S_i)}{\sum_{i=1}^n D_i h_i} \right]^{1/2} \quad (۷)$$

$$T_{min} = \min_{1 \leq i \leq n} \sqrt{\frac{2S_i}{h_i D_i}}$$

حد پایین برای مقادیر ($k_{i,s}$) که مسلماً عبارتست از $k_i(\min) = 1$ برای بدست آوردن حد بالا هم ابتدا مقادیر $T_i(\min)$ از فرمول EOQ بصورت مستقل برای هر کالا بدست می آید و سپس T_{min} به صورت زیر بدست می آید (خوجا و همکاران، ۲۰۰۰).

(۸)

$$T_i(\min) = \left(\frac{2(S + s_i)}{h_i} \right)^{1/2} \quad (۹)$$

$$k_i(\max) = \left\lceil \frac{T_i(\min)}{T_{min}} \right\rceil$$

بطوریکه در رابطه فوق منظور از نماد [a] اولین عدد صحیح بزرگتر از a است.

تابع هدف دوم ما بر روی روابط فوق بی تأثیر است ولی محدودیت مدل، T_{max} را بصورت زیر تغییر می کند:

(۱۰)

$$T_{max} = \min \left(\left[\frac{2(S + \sum_{i=1}^n S_i)}{\sum_{i=1}^n D_i h_i} \right]^{1/2}, \frac{V}{\sum_{i=1}^n D_i v_i} \right)$$

نمایش^۱

یکی از مهمترین مسائل در الگوریتم ژنتیک، نحوه باز نمایی یا کد کردن جوابها می باشد. در مدل پیشنهادی ما، یک کروموزوم یا یک جواب حاوی (n+1) ژن می باشد، بطوریکه n ژن اول که مقادیر عدد صحیح بوده حاوی مقادیر ($k_{i,s}$) می باشند و یک ژن آخر حاوی مقدار مربوط به زمان سیکل پایه بوده و متغیری پیوسته می باشد.

جمعیت اولیه^۲

جمعیت اولیه بصورت تصادفی ایجاد می شود. برای ایجاد هر عضو جمعیت، ابتدا n عدد تصادفی عدد صحیح بین $k_i(\min)$ و $k_i(\max)$ تولید می شود و سپس یک عدد تصادفی پیوسته بین T_{min} و T_{max} ایجاد می شود. قاعدتاً از بین جوابهای تولید شده جوابهایی قابل قبول هستند که در محدودیت مدل صدق کنند.

استراتژی انتخاب^۳

برای انتخاب والدین^۴ از میان جمعیت برای ایجاد ایجاد نسل بعدفرزندان^۵ از مکانیسم چرخ رولت^۶ استفاده می شود. در این مکانیسم انتخاب کروموزومها

- 1 Representation
- 2 Initial population
- 3 Selection strategy
- 4 Parents
- 5 Offsprings
- 6 Roulette-wheel selection mechanism

σ_{share} آستانه تشابه نام دارد و می تواند بین ۰ تا ۱ تغییر کند.

$dz(x,y)$ فاصله اقلیدسی بین دو جواب x,y در فضای نرمال سازی شده توابع هدف بوده و بین صفر و یک می باشد و مطابق رابطه زیر بدست می آید:

(۱۴)

$$dz(x,y) = \sqrt{\left(\frac{f1(x) - f1(y)}{f1_{max} - f1_{min}}\right)^2 + \left(\frac{f2(x) - f2(y)}{f2_{max} - f2_{min}}\right)^2}$$

۱- برای هر جواب $x \in p_t$ مقدار برازندگی اشتراکی یعنی $f^{(1)}(x,t)$ مطابق رابطه زیر بدست می آید:

(۱۵)

$$f^{(1)}(x,t) = \frac{f(x,t)}{nc(x,t)}$$

۲- مقادیر برازندگی با بکارگیری مقادیر برازندگی اشتراکی به صورت زیر نرمال سازی می شود:

(۱۶)

$$f^{(2)}(x,t) = \frac{f^{(1)}(x,t) n_{r(x,t)}}{\sum_{y \in p_t} f^{(1)}(y,t)} f(x,t)$$

تقاطع^۲ و جهش^۳

بعد از انتخاب والدین با مکانیزم چرخ رولت و بر اساس $f^{(2)}(x,t)$ که در رابطه ۱۶ ارائه شد، فرزندان با دو مکانیزم تقاطع و جهش تولید می شوند. متدهای مختلفی برای انجام این دو عملگر وجود دارند. در اینجا از متد نقطه تصادفی جهت عملگر تقاطع استفاده می شود، بطوریکه این عملگر با احتمال p_c عمل می کند و تعدادی کروموزوم تصادفی

احتمالی است، به طوریکه کروموزومهای با تابع برازندگی بیشتر شانس بیشتری برای انتخاب شدن دارند و در واقع احتمال انتخاب i امین کروموزوم با تابع برازندگی f_i عبارتست از $\frac{f_i}{\sum f_i}$.

تابع برازندگی^۱

میزان برازندگی هر کروموزوم مطابق مراحل زیر ارزیابی می شود:

۱- برای هر کروموزوم در هر تکرار $(x \in p_t)$ ، رتبه آن یعنی $r(x,t)$ مطابق رابطه زیر محاسبه می شود:

$$r(x,t) = 1 + nq(x,t)$$

$nq(x,t)$ عبارتست از تعداد جوابهایی که در تکرار t ، جواب x را چیره می کنند. و منظور از p_t ، تکرار t ام جمعیت می باشد.

۲- برای هر جواب بر اساس رتبه بدست آمده یک مقدار برازندگی مطابق رابطه زیر محاسبه می شود:

(۱۲)

$$f(x,t) = N - \sum_{k=1}^{r(x,t)} n_k - 0.5 * (n_{r(x,t)} - 1)$$

n_k عبارتست از تعداد جوابهای با رتبه k و لذا

$$n_{r(x,t)}$$
 عبارتست از تعداد جوابهای با رتبه $r(x,t)$

۱- برای هر جواب یک شمارنده تشابه یعنی $nc(x,t)$ مطابق رابطه زیر محاسبه می کنیم:

(۱۳)

$$nc(x,t) = \sum_{y \in p_t, r(y,t) = r(x,t)} \max\left\{\frac{\sigma_{share} - dz(x,y)}{\sigma_{share}}, 0\right\}$$

2 Crossover
3 Mutation

1 Fitness function

قدم ۳: برای هر عضو جمعیت مقدار تابع برازندگی $f^{(2)}(x,t)$ را با استفاده از رابطه ۱۶ محاسبه کنید.

قدم ۴: با استفاده از متد چرخ رولت و بر اساس مقادیر $f^{(2)}(x,t)$ والد‌ها را جهت تولید نسل بعد انتخاب کنید.

قدم ۵: با استفاده از عملگر تقاطع و جهش فرزندان نسل جدید را تولید کنید.

قدم ۶: با استفاده از مکانیزم چرخ رولت از مجموع والدین و فرزندان تولید شده به تعداد اندازه جمعیت کروموزوم انتخاب کنید.

قدم ۷: قرار دهید $t = t + 1$ و به قدم ۲ بروید.

نتایج حاصل از اجرای مدل

برای بررسی عملکرد الگوریتم پیشنهادی جهت حل مدل، الگوریتم پیشنهادی روی ۱۶۰۰ مسأله که هر یک بصورت تصادفی ایجاد شده اند، تست شده است. هر مسأله دارای ۶ پارامتر اصلی است که از مقاله خوجا و همکاران (۲۰۰۰) استخراج شده و در جدول ۱ خلاصه شده است، همچنین پارامترهای مورد نیاز الگوریتم نیز در جدول ۲ خلاصه شده است. این پارامترها با تکرار الگوریتم به تعداد دفعات زیاد و مطابق با بهترین نتایج حاصل انتخاب شده است.

از دو والد با مکان مشترک را انتخاب می کند و جای آنها را تعویض می کند. بعد از تولید فرزندان با این مکانیزم، عملگر جهش به کار گرفته می شود که در این عملگر هر کروموزوم با احتمال ضعیف برابر p_m تغییر می کند. در هر یک از دو عملگر فوق مقدار هر کروموزوم بین حدود پایین و بالایی که در قسمتهای قبل ارائه شد تغییر می کند و همچنین تنها جوابهایی که در محدودیت مسأله صدق کنند قابل قبول خواهند بود.

جایگزینی^۱

بعد از تولید فرزندان، مجدداً از مکانیزم چرخ رولت استفاده می شود و به تعداد اندازه جمعیت که از قبل تعریف شده است، از فرزندان جدید و نسل قبل به نسل بعد منتقل می شوند.

شرط توقف^۲

هر گاه تعداد جوابهای بهینه پارتویی در طی ۵۰ تکرار متوالی تغییر نکند، الگوریتم متوقف خواهد شد.

مراحل کلی الگوریتم ژنتیک چند هدفه

قدم ۱: با یک جمعیت اولیه تصادفی (p_0) شروع کنید ($t=0$)

قدم ۲: اگر شرط توقف برقرار است متوقف شوید، جواب نهایی عبارت است از p_t

جدول ۱. مقادیر پارامترهای هر مسأله

پارامتر	حدود مقادیر پارامتر
تفاضل (D)	[۱۰۰,۱۰۰۰۰۰۰]
هزینه کلی سفارش دهی (S)	۵,۱۰,۱۵,۲۰
هزینه سفارش دهی جزئی (s_i)	[۵,۵ .۰]
تعداد محصولات (n)	۱۰,۲۰,۳۰,۵۰
هزینه نگهداری	[۲,۳ .۰]
قیمت خرید هر قلم کالا	۱
فضای مورد نیاز جهت انبار هر قلم کالا	۱

جدول ۲. مقادیر پارامترهای الگوریتم

پارامتر	مقدار
سایز جمعیت	۱۰۰
احتمال انجام عملگر تقاطع	۰/۶
احتمال انجام عملگر جهش	۰/۲
	۱

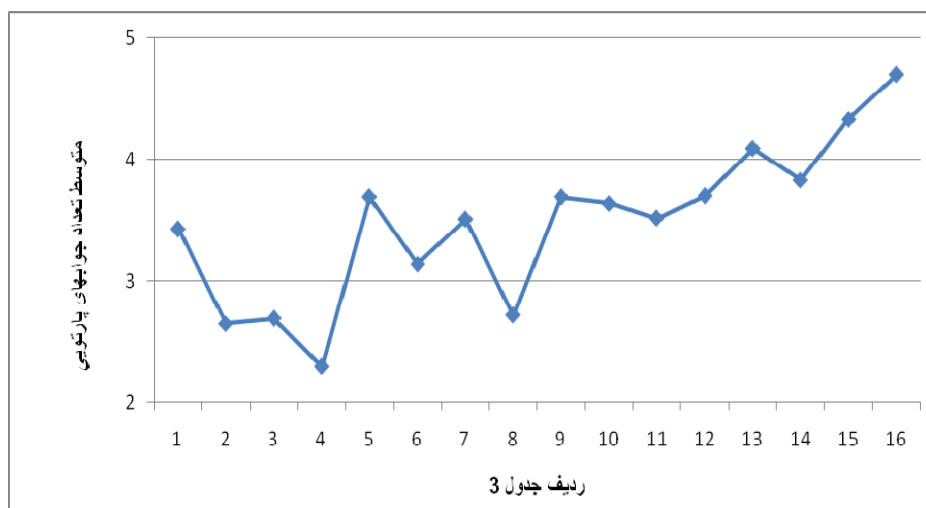


۱۰۰ مسأله محاسبه کرده و از مقادیر متناظر $f2$ آنها نیز میانگین گرفته و این مقادیر را در ستون ششم و هفتم جدول ارائه شده است. همین کار برعکس برای مقادیر می نیمم $f2$ و مقادیر متناظر $f1$ نیز انجام شده و نتایج در ستون هشتم و نهم جدول ارائه شده است. ستون دهم جدول تفاضل ستونهای ۸ و ۶ و ستون یازدهم تفاضل ستونهای ۹ و ۷ جدول می باشد هر چه مقادیر دو ستون آخر جدول بزرگتر باشد نشان دهنده این است که جوابهای پارتویی دارای پراکندگی بیشتری هستند. این مقادیر در قالب نمودار ۲ نمایش داده شده است، همانگونه که ملاحظه می کنید؛ اولاً با افزایش ابعاد مسأله این مقادیر زیادتیر شده و در واقع جوابهای پارتویی دارای تنوع بیشتری می شوند و ثانیاً در کل مقادیر مربوط به $f1$ نسبت به مقادیر $f2$ از پراکندگی بیشتری برخوردارند.

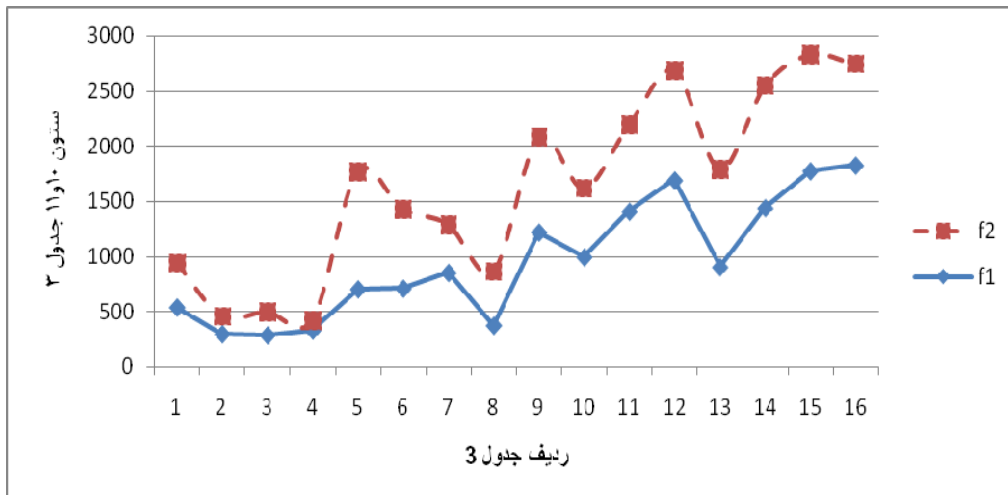
برای هر مقدار n و S که با توجه به جدول ۱ شامل ۱۶ مقدار می شود، ۱۰۰ مسأله تصادفی تولید شده که تفاوت آنها در مقادیر D_i و H_i می باشد، این مسائل با الگوریتم پیشنهادی که توسط نرم افزار ویژوال بیسیک ۲۰۰۷ کد شده، حل گردیده و نتایج حاصل در جدول ۳ خلاصه شده است. ستون پنجم این جدول نشان دهنده متوسط تعداد جوابهای بهینه پارتویی است که برای هر ردیف در ۱۰۰ مسأله تصادفی ایجاد شده، بدست می آید. روند تغییرات این مقادیر در نمودار ۱ نشان داده شده است، همانگونه که این نمودار نشان می دهد با افزایش ابعاد مسأله تعداد جوابهای پارتویی نیز افزایش یافته است. در ادامه در هر ردیف این جدول برای هر مسأله تولید شده از بین جوابهای بهینه پارتویی مقدار مینیمم $f1$ را انتخاب کرده و میانگین آن را روی هر

جدول ۳. نتایج حاصل اجرای ۱۶۰۰ مسأله تصادفی

رتبه	حداکثر فضای در دسترس (میلیون)	تعداد کل کالاها (N)	هزینه کلی سفارش دهی (S)	متوسط تعداد جوابهای بهینه پارتویی	مقایسه مینیمم f1 و مقادیر متناظر f2		مقایسه مینیمم f2 و مقادیر متناظر f1		متوسط f1 - متوسط مینیمم f2	متوسط f2 - متوسط مینیمم f1
					متوسط مینیمم f1	متوسط f2	متوسط f1	متوسط مینیمم f2		
۱	۴	۱۰	۵	۹۳.۲	۷,۳۹۵	۵,۱۸۰	۷,۹۳۷	۴,۷۸۱	۵۴۲	۳۹۹
۲	۶	۱۰	۱۰	۱۵.۲	۷,۸۳۳	۵,۱۰۹	۸,۱۳۳	۴,۹۵۳	۳۰۰	۱۵۷
۳	۹	۱۰	۱۵	۱۹.۲	۸,۶۷۸	۵,۷۳۱	۸,۹۶۷	۵,۵۲۷	۲۹۰	۲۰۴
۴	۱۲	۱۰	۲۰	۸.۱	۹,۲۴۰	۵,۸۴۳	۹,۵۷۲	۵,۷۵۸	۳۳۲	۸۵
۵	۱۰	۲۰	۵	۱۹.۳	۱۹,۰۴۹	۱۲,۹۷۱	۱۹,۷۵۵	۱۱,۹۱۰	۷۰۶	۱,۰۶۱
۶	۲۰	۲۰	۱۰	۶۴.۲	۱۹,۲۵۷	۱۲,۵۶۱	۱۹,۹۶۸	۱۱,۸۴۶	۷۱۱	۷۱۵
۷	۳۰	۲۰	۱۵	۰.۱.۳	۲۰,۷۱۷	۱۳,۰۲۲	۲۱,۵۷۲	۱۲,۵۹۰	۸۵۵	۴۳۲
۸	۴۰	۲۰	۲۰	۲۲.۲	۲۱,۹۳۲	۱۳,۶۶۲	۲۲,۳۱۰	۱۳,۱۷۵	۳۷۸	۴۸۷
۹	۳۰	۳۰	۵	۱۹.۳	۳۲,۵۸۹	۲۰,۹۲۵	۳۳,۸۱۰	۲۰,۰۶۳	۱,۲۲۱	۸۶۲
۱۰	۴۵	۳۰	۱۰	۱۴.۳	۳۷,۲۶۹	۲۳,۲۱۸	۳۸,۲۶۵	۲۲,۵۹۷	۹۹۶	۶۲۱
۱۱	۶۸	۳۰	۱۵	۰.۲.۳	۳۹,۰۲۷	۲۵,۴۲۹	۴۰,۴۳۶	۲۴,۶۴۴	۱,۴۱۰	۷۸۶
۱۲	۹۰	۳۰	۲۰	۲.۳	۴۳,۰۱۷	۲۷,۸۹۰	۴۴,۷۰۸	۲۶,۸۹۸	۱,۶۹۱	۹۹۳
۱۳	۵۰	۵۰	۵	۵۹.۳	۴۶,۱۴۶	۲۹,۷۱۰	۴۷,۰۵۲	۲۸,۸۲۶	۹۰۶	۸۸۴
۱۴	۱۰۰	۵۰	۱۰	۳۳.۳	۵۲,۸۸۱	۳۴,۷۶۴	۵۴,۳۲۳	۳۳,۶۵۷	۱,۴۴۲	۱,۱۰۷
۱۵	۱۵۰	۵۰	۱۵	۸۳.۳	۵۶,۰۲۴	۳۶,۷۹۹	۵۷,۷۹۹	۳۵,۷۴۳	۱,۷۷۵	۱,۰۵۶
۱۶	۲۰۰	۵۰	۲۰	۲.۴	۵۶,۱۵۰	۴۰,۳۴۲	۵۷,۹۷۶	۳۹,۴۲۴	۱,۸۲۶	۹۱۸
		متوسط		۳	۲۹,۸۲۵	۱۹,۵۷۲	۳۰,۷۸۷	۱۸,۸۹۹	۹۶۱	۶۷۳



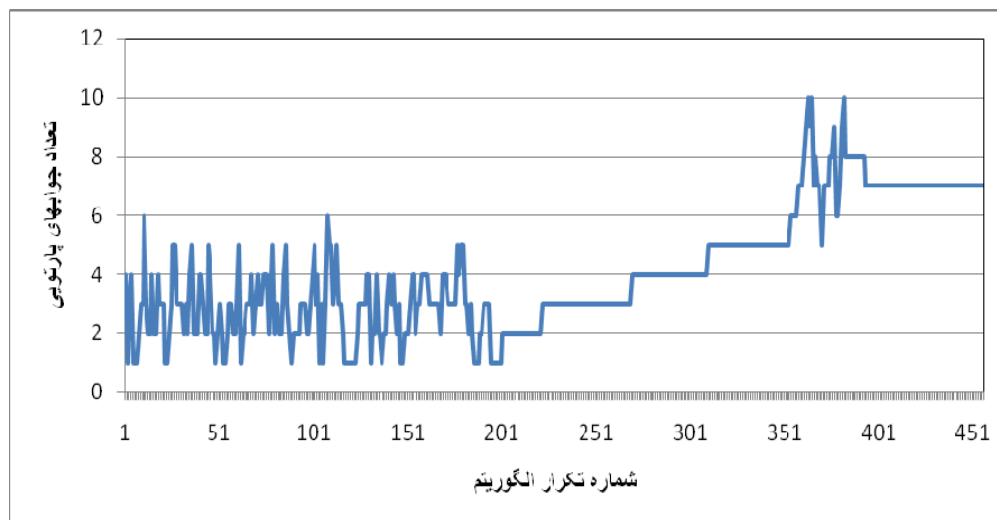
نمودار ۱. متوسط تعداد جوابهای پارتویی برای مقادیر مختلف n و S



نمودار ۲. میزان پراکندگی جوابهای پارتویی

مسئله طی تکرارهای متوالی الگوریتم به آن دست پیدا کرده در نمودار ۳ ارائه شده است. همانگونه که ملاحظه می شود و در شرط توقف الگوریتم نیز ذکر شد در طی ۵۰ تکرار آخر الگوریتم این تعداد بدون تغییر مانده است.

در ادامه نتایج حاصل از اجرای یک مسئله تصادفی ارائه می شود. در این مسئله پارامترها عبارتند از: $Shares = 1, N=30, S=15$. الگوریتم پیشنهادی طی ۴۵۷ تکرار این مسئله را حل کرده است. روند تغییرات تعداد جوابهای پارتویی که



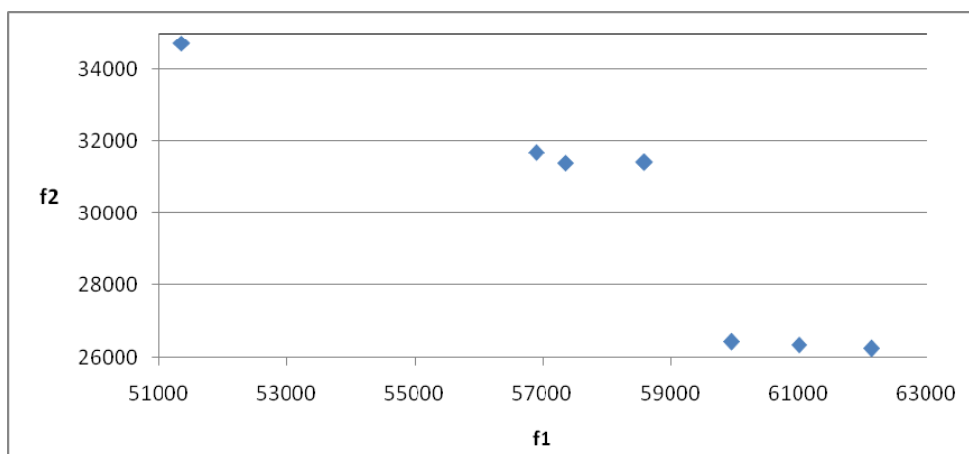
نمودار ۳. روند تغییر تعداد جوابهای بهینه پارتویی در طی تکرارهای الگوریتم

همانگونه که ملاحظه می کنید هیچ یک از این جوابها دیگری را مغلوب نمی کند، یا به عبارت ساده تر

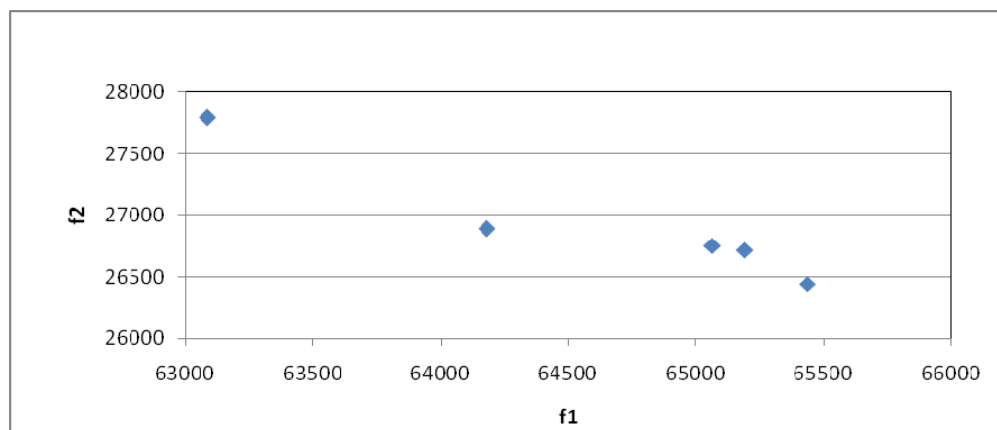
جوابهای نهایی این مسئله که شامل ۷ جواب بهینه پارتویی می شود در نمودار ۴ ارائه شده است.

طی تکرارهای متوالی الگوریتم نیز به ترتیب در نمودارهای ۶ و ۷ آمده است، همانگونه که ملاحظه می کنید شیب بهبود در مورد هر دو نمودار در ابتدای الگوریتم تندتر از تکرارهای پایانی است که نشان دهنده عملکرد منطقی الگوریتم پیشنهادی است. لازم به ذکر است که برای حل این مسأله از پارامترهای جدول ۲ استفاده شده است، ولی برای حل این مسأله یا هر مسأله دلخواه دیگری میتوان با ترکیبات مختلف پارامترها مسأله را حل کرد تا بهترین ترکیب پارامترها برای هر مسأله بدست آید.

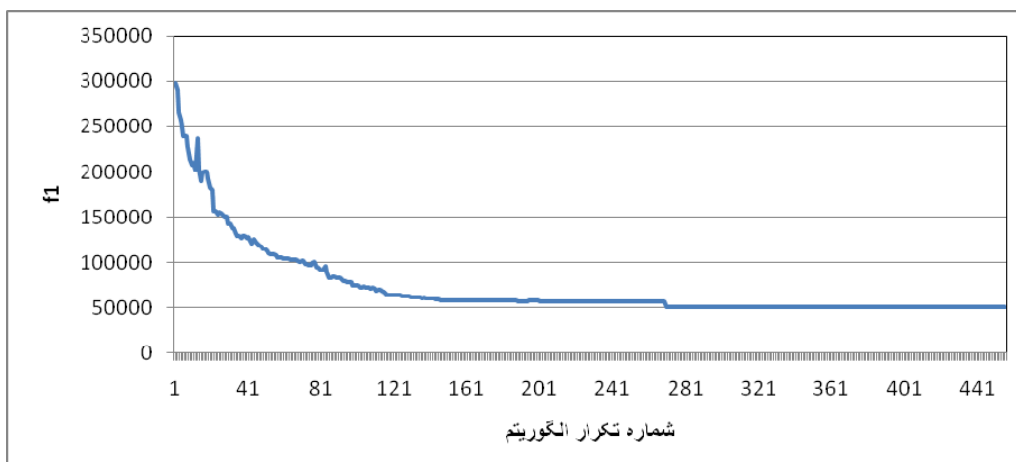
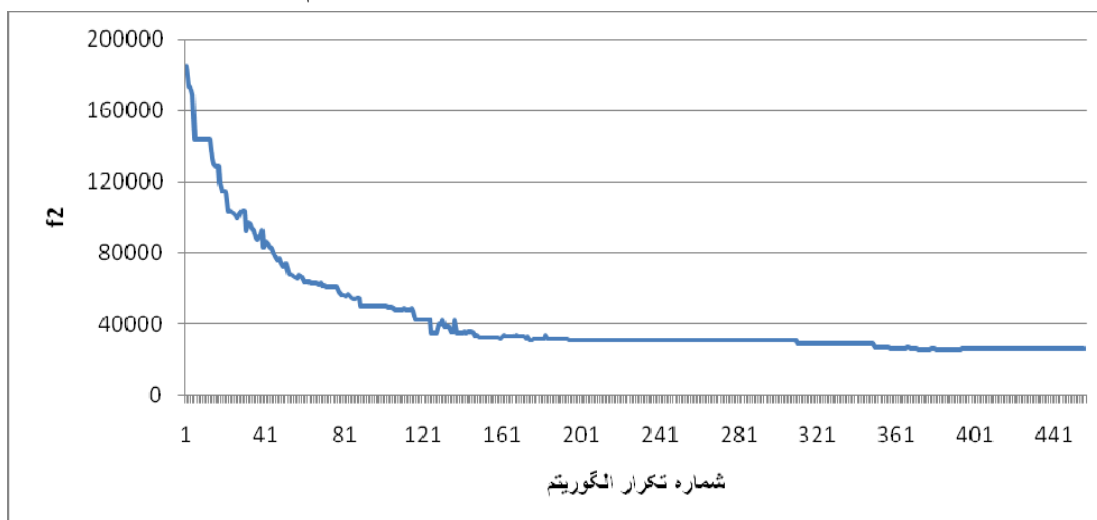
جوابی که هر دو تابع هدفش از دو تابع هدف سایر جوابها بهتر باشد وجود ندارد و نهایتاً این تصمیم گیرنده است که با توجه به تبدلات بین اهداف یکی از این جوابهای پارتویی را به عنوان جواب رضایت بخش خود انتخاب می کند. لازم به ذکر است که در هر تکرار الگوریتم تعدادی جواب پارتویی بدست می آید که ممکن است لیست جوابهای پارتویی که تا آن تکرار بدست آمده را تغییر دهد. بعنوان نمونه در نمودار ۵ جوابهای پارتویی تکرار آخر که شامل ۵ جواب می باشد ارائه شده است. روند بهبود $f1$ و $f2$



نمودار ۴. جوابهای بهینه پارتویی نهایی



نمودار ۵. جوابهای بهینه پارتویی تکرار آخر الگوریتم

نمودار ۶. روند بهبود $f1$ در طی تکرارهای الگوریتمنمودار ۷. روند بهبود $f2$ در طی تکرارهای الگوریتم

۷- نتیجه گیری و پیشنهادات

معمولاً در بسیاری از مدل‌های کنترل موجودی، به علت ارتباطات بین کالاها بعنوان مثال استفاده از فضای انبار مشترک یا بودجه مشترک و یا مشترک بودن قسمتی از هزینه های سفارش دهی، سیستم های کنترل موجودی به صورت مستقل برای کالاهای مختلف عمل نمی کنند و اصطلاحاً بایستی از سیستم هایی که سفارشات کالاها را بصورت توأم در نظر می گیرد، استفاده نمود. یکی از این نوع سیستم ها مدل تکمیل موجودی توأم کالاها (JRP) است. در تحقیق

حاضر حالت خاصی از این مدل که در عمل می تواند مفید باشد ولی تا بحال بدان پرداخته نشده توسعه داده شده و حل گردیده است. در مدل پیشنهادی برای مسأله JRP کلاسیک دو تابع هدف در نظر گرفته شده و همچنین محدودیت فضای انبار که معمولاً در عمل با آن مواجهیم نیز به مدل اضافه شده است. بعنوان تحقیقات آتی می توان به این مدل توابع هدف یا محدودیت های دیگری را نیز اضافه کرده و یا جایگزین توابع هدف و محدودیت موجود نمود. از آنجاییکه مدل پیشنهادی با الگوریتمهای

- Chan, C. K. , Yuk-on Li, L. , To Ng, C. , Kin-sion Cheung, B. , & Langevin, A. (2006). Scheduling of multi-buyer joint replenishments. *International Journal of Production Economics*, 102,132-142.
- Goyal, S. K. , & Belton, A. S. (1979) . A simple method of determining order quantities in joint replenishments under deterministic demand. *Management Science* , 25, 604- 614.
- Goyal, S. K . (1974). Determination of optimum packaging frequency of items jointly replenished. *Management Science*, 21,436-443.
- Hoque, M. A. (2006). An optimal solution technique for the joint replenishment problem with storage and transport capacities and budget constraints. *European Journal of Operational Research*, 175, 1033-1042.
- Hsu, S. L. (2009). Optimal joint replenishment decisions for a central factory with multiple satellite factories. *Expert Systems With Applications*, 36,2494-2502.
- Kaspi, M. , & Rosenblatt, M. J. (1983). An improvement of silver's algorithm for the joint replenishment problem. *IIE Transactions*, 15,264-269.
- Khouja, M. , Goyal, S . (2008). A review of the joint replenishment problem literature:1989-2005. *European Journal of Operational Research* ,186 ,1-16.
- Khouja, M. , Park, S. , & Saydam, C. (2005). Joint replenishment problem under continuous unit cost change . *International Journal of Production Research*, 43,311-326.
- Khouja, M. , Michalewicz, M. , & Satoskar ,S. (2000). A comparison between genetic algorithms and the RAND method for solving the joint replenishment problem. *Production Planning and Control*, 11,556-564.
- Klein, C. M. ,& Ventura, J. A. (1995). An optimal method for a deterministic joint replenishment inventory policy in discrete time. *The Journal of the Operational Research Society*, 46,643-657.
- Konak ,A. ,Coit, D. W. ,& Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: a tutorial. *Reliability Engineering & System Safety*,91,992-1007.

ریاضی دقیق قابل حل نیست، جهت حل به توسعه یکی از الگوریتمهای متاهیوریستیک که کارایی خوبی در حل مسائل چند هدفه از خود نشان داده است یعنی الگوریتم ژنتیک چند هدفه، پرداخته ایم. برای یک مسأله نمونه تعداد جوابهای بهینه پارتویی نمایش داده شده، همچنین روند کاهش دو تابع هدف در طی تکرارهای متوالی الگوریتم را نمایش داده و نشان داده ایم که میزان کاهش در تکرارهای ابتدایی الگوریتم بیشتر است. به منظور بررسی عملکرد کلی الگوریتم، ۱۶۰۰ مسأله به صورت تصادفی تولید کرده و حل شده است. روند تغییر تعداد جوابهای بهینه پارتویی با افزایش ابعاد مسأله نمایش داده و نشان داده شده است که این تعداد با افزایش ابعاد مسأله افزایش می یابد. همچنین نشان داده ایم که پراکندگی جوابهای بهینه پارتویی نیز با افزایش ابعاد مسأله افزایش می یابد. در زمینه توسعه الگوریتم کاراتر جهت حل مدل برای تحقیقات آتی می توان به توسعه الگوریتمهای هیوریستیک موجود در حل JRP کلاسیک مانند الگوریتم RAND پرداخت. به نظر می رسد ترکیب الگوریتم RAND و ژنتیک چند معیاره می تواند هم کیفیت جوابها را بهبود می دهد و هم زمان رسیدن به جواب نهایی را کاهش دهد و لذا توسعه چنین الگوریتمی هم می تواند زمینه ای برای تحقیقات آتی باشد.

فهرست منابع

- Arkin, E. , Joneja, D. , & Roundy, R (1989). Computational complexity of uncapacitated multi echelon production planning problems. *Operations Research Letters* , 8, 61-66.

- Production Planning and Control , 16 ,255-262.
- Silver, E. (1976). A simple method of determining order quantities in joint replenishments under deterministic demand. *Management Science*, 22, 1351-1361.
- Tsai, C. Yu. , Tsai, C. Ya. ,& Huang, P. W. (2009). An association clustering algorithm for can order policies in the joint replenishment problem. *International Journal of Production Economics*, 117 ,30-41.
- Li,Q. (2004). Solving the multi-buyer joint replenishment problem with the RAND method. *Computer and Industrial Engineering* ,466,755-762.
- Moon, I. K. , & Cha, B. C. (2006). The joint replenishment problem with resource restriction. *European Journal of Operational Research* , 173, 190-198.
- Olsen, A. (2008). Inventory replenishment with interdependent ordering: An evolutionary algorithm solution. *International Journal of Production Economics* ,113,359-369.
- Siajadi, H. , Ibrahim, R. N. ,Lochert, P. B. , & Chan, W. M. (2005). Joint replenishment policy in inventory production systems.