بسم الله الرحمن الرحیم

دانشکده ریاضی و علوم کامپیوتر

پایان‌نامه برای دریافت درجه کارشناسی ارشد در رشته علوم تصمیم و مهندسی دانش

کاهش ضایعات برش یک بعدی با رعایت کمینگی الگوها

استاد راهنما

دکتر مهدی زعفرانیه

استاد مشاور

دکتر محمود امین طوسی

پژوهشگر:

محمد حسین شاهمنصوری

تیر ۱۳۹۶

فرم ۱۱۳-ت

باسمه تعالی

فرم ارزشیابی و صورتجلسه دفاع از پایان‌نامه کارشناسی ارشد

جلسه دفاع از پایان‌نامه آقای /خانم **محمد حسین شاهمنصوری** دانشجوی رشته علوم تصمیم و مهندسی دانش به شماره دانشجویی ۹۳۱۳۱۳۷۰۶۴ با عنوان:

کاهش ضایعات برش یک بعدی با رعایت کمینگی الگوها

در مورخه .... در دانشکده ریاضی و علوم کامپیوتر تشکیل و توسط هیات داوران مورد ارزشیابی قرار گرفت و نمره .... برابر درجه .... برای آن تعیین گردید .

به‌این ترتیب از این تاریخ آقای/ خانم محمد حسین شاهمنصوری به عنوان کارشناس ارشد در رشته مذکور شناخته می‌شود .

موارد ارزشیابی	موارد	حداکثر نمره	نمره کسب شده
۱- کیفیت نگارش	رعایت اصول نگارش انسجام در تنظیم بخشهای مختلف ،کیفیت تصاویر ، جداول و اشکال ،تنظیم فهرست ها ، منابع و ماخذ	٤	
۲- کیفیت علمی	بررسی تاریخچه و سابقه تجربی و نظری موضوع انسجام منطقی دربخش های مختلف پایان نامه ،ابتکار و نوآوری ،اهمیت و ارزش علمی پایان نامه ،استفاده از منابع معتبر و جدید ،کیفیت تجزیه و تحلیل یافته ها و نتیجه گیری ،روشن بودن روش کار ،هدف ها و فرضیه های تحقیق ،جدید بودن روش تحقیق	۱۰	
۳-کیفیت ارایه در جلسه دفاع	تسلط بر موضوع و بیان واضح وتفهیم آن ،توانایی در پاسخگویی به سوالات مطرح شده در جلسه ، رعایت زمان ارایه , روش ارائه	٤	
٤-ارزشیابی گزارشات	گزارش های دوره ای پیشرفت کار(حداقل ٤مورد)	۱	
۵- خروجی پایان نامه	مقاله مستخرج از پایان نامه: این نمره به صورت زیر اختصاص می یابد ۱) چکیده کنفرانسی هر مورد ۰/۲۵نمره تا سقف۰/۵ نمره ۲) مقاله کامل در مجموع مقالات همایشهای معتبر یا مقاله در مجلات علمی-ترویجی معتبر پذیرفته شده یا چاپ شده هر مورد ۰/۵ نمره تا سقف ۱نمره ۳)مقاله پذیرفته شده یا چاپ شده در مجلات علمی پژوهشی معتبر ۱ نمره ٤)مقاله ارسال شدهبه مجلات علمی پژوهشی معتبرهر مورد ۲۵/. نمره تا سقف ۰/۵ نمره ۵) دستگاه ساخته شده دارای گواهی ثبت اختراع یا به سفارش سازمان ها تا سقف ۱ نمره ٦) دستگاه ساخته شده کاربردی که به تایید رئیس دانشکده رسیده باشد تا سقف ۰/۵ نمره	۱	
جمع			

درجه معادل کسب شده: (از ۱۹تا ۲۰ عالی☐ از۱۸تا ۱۸/۹۹بسیارخوب ☐ از ۱٦تا ۱۷/۹۹خوب ☐ از ١٤ تا ۱۵/۹۹قابل قبول☐ کمتر از ١٤ غیر قابل قبول☐)

مشخصات هیات دوران

ردیف	نام و نام خانوادگی	سمت	مرتبه علمی	محل کار	امضــــــــــــا
۱	دکتر مهدی زعفرانیه	استاد راهنما	استادیار	دانشگاه حکیم سبزواری	
۲	دکتر محمود امین طوسی	استاد مشاور	استادیار	دانشگاه حکیم سبزواری	
۳	دکتر مینا مسعودی فر	استاد داور	استادیار	دانشگاه حکیم سبزواری	
۴	دکتر علی اکبر استاجی	نماینده تحصیلات تکمیلی	استادیار	دانشگاه حکیم سبزواری	

امضا

مدیر گروه

امضا

رئیس دانشکده

سوگند نامه دانش آموختگان دانشگاه حکیم سبزواری

به نام خداوند جان و خرد   کزین برتر اندیشه بر نگذرد

اینک که به خواست آفریدگار پاک، کوشش خویش و بهره گیری از دانش استادان و سرمایه‌های مادی و معنوی این مرز و بوم، توشه‌ای از دانش و خرد گردآورده‌ام، در پیشگاه خداوند بزرگ سوگند یاد می‌کنم که در به کارگیری دانش خویش، همواره بر راه راست ودرست گام بردارم. خداوند بزرگ، شما شاهدان، دانشجویان و دیگر حاضران را به عنوان داورانی امین گواه می گیرم که ازهمه دانش و توان خود برای گسترش مرزهای دانش بهره گیرم و از هیچ کوششی برای تبدیل جهان به جایی بهتر برای زیستن، دریغ نورزم. پیمان می‌بندم که همواره کرامت انسانی را در نظر داشته باشم و همنوعان خود را در هر زمان و مکان تا سر حد امکان یاری دهم. سوگند می خورم که در به کارگیری دانش خویش به کاری که با راه و رسم انسانی، آیین پرهیزگاری، شرافت و اصول اخلاقی برخاسته از ادیان بزرگ الهی، به ویژه دین مبین اسلام، مباینت دارد دست نیازم. همچنین در سایه اصول جهان شمول انسانی و اسلامی، پیمان می‌بندم از هیچ کوششی برای آبادانی و سرافرازی میهن و هم میهنانم فروگذاری نکنم و خداوند بزرگ را به یاری طلبم تا همواره در پیشگاه او و در برابر وجدان بیدار خویش و ملت سرافراز، بر این پیمان تا ابد استوار بمانم.

نام و نام خانوادگی: محمد حسین شاهمنصوری

تاریخ و امضا:

تأییدیه‌ی صحت و اصالت نتایج

باسمه تعالی

اینجانب محمد حسین شاهمنصوری به شماره دانشجویی ۹۳۱۳۱۳۷۰۶۴ دانشجوی رشته علوم تصمیم و مهندسی دانش مقطع تحصیلی کارشناسی ارشد تأیید می‌نمایم که کلیه‌ی نتایج این پایان‌نامه حاصل کار اینجانب و بدون هرگونه دخل و تصرف است و موارد نسخه برداری شده از آثار دیگران را با ذکر کامل مشخصات منبع ذکر کرده‌ام. درصورت اثبات خلاف مندرجات فوق، به تشخیص دانشگاه مطابق با ضوابط و مقررات حاکم (قانون حمایت از حقوق مؤلفان و مصنفان و قانون ترجمه و تکثیر کتب و نشریات و آثار صوتی، ضوابط و مقررات آموزشی، پژوهشی و انضباطی ...) با اینجانب رفتار خواهد شد و حق هرگونه اعتراض درخصوص احقاق حقوق مکتسب و تشخیص و تعیین تخلف و مجازات را از خویش سلب می‌نمایم. در ضمن، مسؤولیت هرگونه پاسخگویی به اشخاص اعم از حقیقی و حقوقی و مراجع ذی صلاح (اعم از اداری و قضایی) به عهده ی اینجانب خواهد بود و دانشگاه هیچ گونه مسؤولیتی در این خصوص نخواهد داشت.

نام و نام خانوادگی: محمد حسین شاهمنصوری

تاریخ و امضا:

مجوز بهره برداری از پایان‌نامه

بهره برداری از این پایان‌نامه در چهارچوب مقررات کتابخانه و با توجه به محدودیتی که توسط استاد راهنما به شرح زیر تعیین می‌شود، بلامانع است:

☐ بهره برداری از این پایان‌نامه برای همگان بلامانع است.

☐ بهره برداری از این پایان‌نامه با اخذ مجوز از استاد راهنما، بلامانع است.

☐ بهره برداری از این پایان‌نامه تا تاریخ .... ممنوع است.

استاد راهنما: دکتر مهدی زعفرانیه

تاریخ و امضا:

تقدیم به:

همسر

و

پدر و مادرم

سپاس خداوندگار حکیم را که با لطف بی کران خود، آدمی را زیور عقل آراست.

در آغاز وظیفه خود می دانم از زحمات بی دریغ استاد راهنمای خود، جناب آقای دکتر زعفرانیه، صمیمانه تشکر و قدردانی کنم که قطعاً بدون راهنمایی های ارزنده ایشان، این مجموعه به انجام نمی رسید.

از جناب آقای دکتر امین طوسی که زحمت مطالعه و مشاوره این رساله را تقبل فرمودند و در آماده سازی این رساله، به نحو احسن اینجانب را مورد راهنمایی قرار دادند، کمال امتنان را دارم.

همچنین لازم می دانم از گروه پارسی‌لاتک در پاسخگویی به مشکلات کاربران کمال قدردانی را داشته باشم.

در پایان، بوسه می زنم بر دستان خداوندگاران مهر و مهربانی، پدر و مادر عزیزم و بعد از خدا، ستایش می‌کنم وجود مقدس شان را و تشکر می‌کنم از خانواده عزیزم به پاس عاطفه سرشار و گرمای امیدبخش وجودشان، که بهترین پشتیبان من بودند.

محمد حسین شاهمنصوری

تیر ۱۳۹۶

فهرست مطالب

فهرست جداول

فهرست تصاویر

فرم چکیده ی پایان‌نامه ی دوره ی تحصیلات تکمیلی

مدیریت تحصیلات تکمیلی

دانشگاه حکیم سبزواری

نام خانوادگی دانشجو:شاهمنصوری	نام:محمد حسین	ش. دانشجویی:۹۳۱۳۱۳۷۰۶۴
استاد راهنما: دکتر مهدی زعفرانیه		
استاد مشاور: دکتر محمود امین طوسی		
دانشکده ریاضی و علوم کامپیوتر	رشته: علوم تصمیم و مهندسی دانش	
مقطع: کارشناسی ارشد	تاریخ دفاع:تیر ۱۳۹۶	تعداد صفحات: ؟؟
عنوان پایان‌نامه: کاهش ضایعات برش یک بعدی با رعایت کمینگی الگوها		
کلید واژه‌ها:برش یک بعدی، کاهش ضایعات، تولید ستون، الگوریتم ژنتیک		

چکیده: مدیریت اقتصادی و تولید صحیح، از ارکان مهم در اداره و راهبری موثر يك سازمان است. مسئله کاهش ضایعات حاصل از برش در صنایع بسیاری مانند صنایع تولیدکننده بدنه اتومبیل، کاغذ، پوشاك، چرم، سنگ‌بری و یا در فعالیت‌هایی مانند حمل و نقل، انبارداری و بسته‌بندی کاربرد دارد. در این صنایع نیاز است اقلام از قطعات بزرگ‌تری برش داده شوند، از این‌رو نحوه برش و میزان ضایعات حاصله، بر تولید و هزینه‌ها، تاثیرگذار است. از این‌رو کم کردن ضایعات برش و فرصت استفاده مجدد از ضایعات بوجود آمده، از جمله دغدغه‌های اقتصادی مدیران اینگونه صنایع است.

این پایان‌نامه، به بحث در مورد کاهش ضایعات برش یک بعدی می‌پردازد به گونه‌ای که تعداد الگوهای مصرف شده نیز تا حد امکان کاهش یابد. برای این منظور از روش‌های مختلف بهینه‌سازی از جمله الگوریتم ژنتیک، روش سیمپلکس، برنامه‌ریزی پویا و روش تولید ستون و نیز ترکیب این روش‌ها با یکدیگر استفاده شده است.

نتایج نشان می‌دهد روش پیشنهادی مبتنی‌بر الگوریتم ژنتیک و روش تولید ستون بهبود قابل ملاحظه‌ای نسبت به روش های مشابه در کاهش ضایعات برش و تعداد الگوهای مصرفی داشته است.

پیش‌گفتار

محدود بودن منابع یکی از مهمترین عواملی است که بهینه‌سازی فرآیندهای تولید را توجیه می‌کند. تلاش در جهت کاهش مواد اولیه برای حجم معینی از تولید، منجر به افزایش بهره‌وری می‌گردد. نیاز به برش مواد اولیه برای تولید کالا در بسیاری از صنایع وجود دارد که با توجه به الگوی برش می‌تواند مقادیر ضایعات متفاوتی را بوجود آورد. شناسایی الگوهای برشی که کمترین ضایعات را تحمیل کند منجر به بوجود آمدن مساله کاهش ضایعات برش گردیده است که بسیاری از پژوهشگران برای حل آن تلاش نموده‌اند. در این پژوهش منابع اصلی که مورد مطالعه و بررسی قرار گرفته‌اند، به‌قرار زیر هستند:

- Haessler, Robert W and Sweeney, Paul E. Cutting stock problems and solution procedures. European Journal of Operational Research, 54(2):141–150, 1991.
- Kazunga, C, Mutambara, LHN, and Mapurisa, J. A column generation approach to a carpentry cutting stock problem: a case study for planks cutting in zimbabwe. African Journal of Educational Studies in Mathematics and Sciences, 9(1):49–59, 2011.
- Thomas, Jaya and Chaudhari, Narendra S. An integrated genetic algorithm approach to 1dcutting stock problem. International Journal of Operational Research, 27(1-2):23–46, 2016.

در این پژوهش ضمن مطالعه این تلاش‌ها، به ارائه راهکارهای ابتکاری برای حل مساله برش یک بعدی پرداخته‌ایم و عملکرد راهکارهای پیشنهادی با استفاده از داده‌های منابع ذکر شده با راهکارهای مطرح شده در این منابع مقایسه شده است.

این پایان‌نامه شامل ۴ فصل است: در فصل ۱ تعاریف و مفاهیم مورد نیاز مسائل برش موجودی مطرح شده است. در فصل ۲ الگوریتم ژنتیک به عنوان یکی از روش‌های حل مسئله برش موجودی مرور شده است. در فصل ۳ روش تولید ستون به عنوان یکی دیگر از روش‌های حل مسئله برش موجودی شرح داده شده است. در فصل ۴ روش‌های پیشنهادی برای حل مسئله برش موجودی یک بعدی به همراه الگوریتم و پیاده‌سازی آن در نرم‌افزار متلب شرح داده شده است.

فصل ۱

تعاریف و پیشینه تحقیق

۱-۱ مقدمه

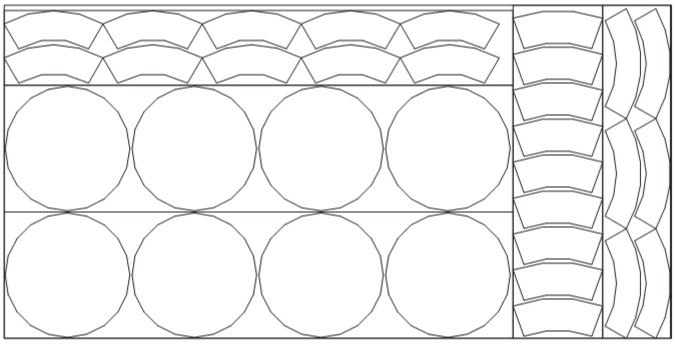
مدیریت اقتصاد و تولید صحیح، از ارکان مهم در اداره وراهبری موثر یک سازمان است. در فرآیند تولید بسیاری از صنایع نیاز است تا قطعات کوچکتری با برش اجسام بزرگ‌تر حاصل شوند و یا به طور معادل قطعات کوچک تری در یک جسم بزرگ‌تر جای داده شوند. در این عمل معمولا بخش‌هایی از جسم بزرگ‌تر به قطعاتی تبدیل می‌شوند که قابل استفاده در هیچ یک از محصولات تولیدی نبوده و بعنوان ضایعات و دورریز محسوب می‌شوند. کاهش چنین ضایعاتی نقش مهمی در کاهش هزینه‌ها داشته و بعنوان یکی از موضوعات علم تحقیق در عملیات و تحت عنوان مساله برش موجودی (CSP)[۲] توجه بسیاری از محققان را در نیم قرن گذشته به خود جلب کرده است [؟].

در عمل، قطعات کوچک‌تر به عنوان لیست سفارشات و قطعات بزرگ‌تر به عنوان موجودی شناخته می‌شوند. هدف اصلی، کمینه کردن ضایعات حاصل از برش به منظور تامین سفارشات از موجودی است. به همین دلیل این مسئله را مسئله کاهش ضایعات برش نیز می‌نامند. غالباً هدف مدیریت در برنامه‌ریزی، برش مواد خام به نحوی است که ضمن کمینه کردن ضایعات برش، کمترین رول‌های مصرفی را نیز داشته باشیم تا علاوه برکاهش هزینه مواد اولیه و نیز ضایعات به تولید مناسب‌تر، کاهش هزینه‌ها و روش‌های کمک به اقتصاد سازمان دست یافت. در کاربردهای واقعی ملاحظه یکسری اهداف فرعی دیگر از جمله کاهش تعداد الگوهای برش‌های مختلف (کاهش هزینه راه‌اندازی/تنظیم)، هزینه استفاده از یک مدل برش خاص و... نیز لازم است.

این مساله تقریبا در تمامی صنایع دارای کاربرد فراوانی است و در صورت بهبود نحوهء استفاده از مواداولیه، می‌توان صرفه‌جویی قابل ملاحظه‌ای در قیمت تمام شده محصولات بدست آورد. کاربرد این مدل در حالت یک بعدی، مانند برش میله و الوار، دو بعدی مانند برش شیشه، چوب ، فلز، پلاستیک ، پارچه، چرم، .... و سه بعدی مانند سنگ ، امکانپذیر است. حالت دوبعدی و سه‌بعدی می‌تواند از دید بارگذاری و حمل و نقل نیز مورد استفاده قرار گیرد. به دلیل اهمیت و کاربرد زیاد

[۲] Cutting Stock Problem

این مدل در صنایع مختلف، تلاش زیادی برای حل آن انجام شده است. برای مثال آقای کوی [۳] الگوریتمی ارائه می‌دهد که در آن تعدادی از سفارشات در یک نوار کنار هم قرار داده شده‌اند، سپس با قرار دادن تعدادی از این نوار ها در محور X یا Y الگوهای T شکل را بوجود می‌آورد.



شکل ۱-۱: نمایش الگوهای T شکل[؟]

در این الگوریتم، ممکن است بیش از یک ردیف از قطاع های مشابه در یک نوار قرار گیرد. هر نوار نیز در یکی از دو جهت افقی x یا عمودی y می‌تواند قرار گیرد. این روش از الگوریتم کوله‌پشتی و یک روش شمارش ضمنی برای تعیین ترکیب بهینه سطرهای شیار در یک نوار، تعداد نوار و جهت آنها در یک الگو استفاده می‌کند. این الگوریتم در یک مسئله واقعی برش کارخانه که ژنراتورهای الکتریکی بزرگ می سازد، استفاده شده است. نتایج نشان می‌دهد این الگوریتم از دو لحاظ زمان محاسباتی و مصرف مواد، بهینه است [؟].

الپ [۱] و همکاران به کاربرد مسئله برش موجودی در کارگاه تولیدی میماگ ماکینا [۲] پرداخته‌اند. در این کارگاه، هدف مسئله برش موجودی یک بعدی، برش تعداد میله‌ای دلخواه با طول‌های دلخواه از یک میله بزرگ‌تر است. شرکت میماگ از میله‌های برش داده شده در پروژه‌های عمرانی استفاده می‌کند. الگوریتم ارائه شده در این مقاله، مساله برش موجودی را به‌صورت برنامه ریزی عدد صحیح تبدیل کرده و به کمک کتابخانه های موجود برای برنامه ریزی عدد صحیح، حل می‌کند. [؟].

مرابیتو [۳] [؟] و همکاران برای حل مساله برش صفحات مستطیل شکل در صنعت تولید و برش تخته‌های فشرده در برزیل جهت تعیین بهترین الگو برای ماشین های برش، از روش‌های بهینه سازی موسوم به کاهش ضایعات برش استفاده کرده‌اند. دستگاه‌های برش در این صنعت شامل مجموعه‌ای از دیسک اره‌ها، دستگاه‌های حرکت‌دهنده و یا نگهدارنده صفحات، و در نهایت بارگیری و تخلیه در ایستگاه مورد نظر می‌باشند. این دستگاه شامل محدودیت‌های غیر معمول ازجمله مرزبندی روی انواع نمونه‌ها و تفاوت قائل شدن بین بزرگ‌ترین و کوچک‌ترین طول اقلام در الگوی برش و هم‌چنین محدودیت‌های معمولی مانند در دسترس بودن اره طولی و عرضی، متعامد و دستگاه گیوتینی دو مرحله‌ای بدون پیرایش است. برای این منظور از روش تولید ستون دو فازی استفاده شده است تا بتواند یک فرمول برای صنعت برش تخته های فشرده توصیف نماید. هر فاز از پردازش با یک برنامه‌ریزی عدد صحیح مدل شده است که توسط دو روش جایگزین پاسخ داده می‌شود. روش اول

[۳] Cui [۱] Alp [۲] Mimag Makina [۳] Morabito

براساس برنامه‌ریزی پویا و روش دوم براساس توسعه‌ای ساده از روش شمارش ضمنی ارائه شده توسط گیلمور و گوموری است [؟].

۱-۲ طبقه‌بندی مسائل برش موجودی

کاربردهای متنوع و زیاد مساله برش موجودی باعث شده است دایکوف [1] [؟] یک طبقه‌بندی برای مسائل برش موجودی و بسته‌بندی (مسائل بسته‌بندی به شدت مرتبط با مسائل برش موجودی هستند) ارائه دهد. او مسائل را با استفاده از ۴ مشخصه به شرح جدول ؟؟ طبقه‌بندی نموده است [؟].

جدول ۱-۱: چهار مشخصه طبقه بندی مسائل برش موجودی توسط دایکوف [؟]

ابعاد	نوع تخصیص	انواع موجودی	سفارشات
تعداد ابعاد (N)	استفاده از همه موجودی برای تولید سفارش‌ها (B)	یک جسم بزرگ(O)	سفارشات کم با ابعاد مختلف (F)
	تولید همه سفارشات با بخشی از موجودی (V)	تعداد زیادی موجودی یکسان (I)	سفارشات زیاد با ابعاد زیاد و متفاوت (M)
		موجودی‌های متفاوت(V)	سفارشات زیاد با ابعاد نسبتا کم(R) سفارشات زیاد و یکسان(C)

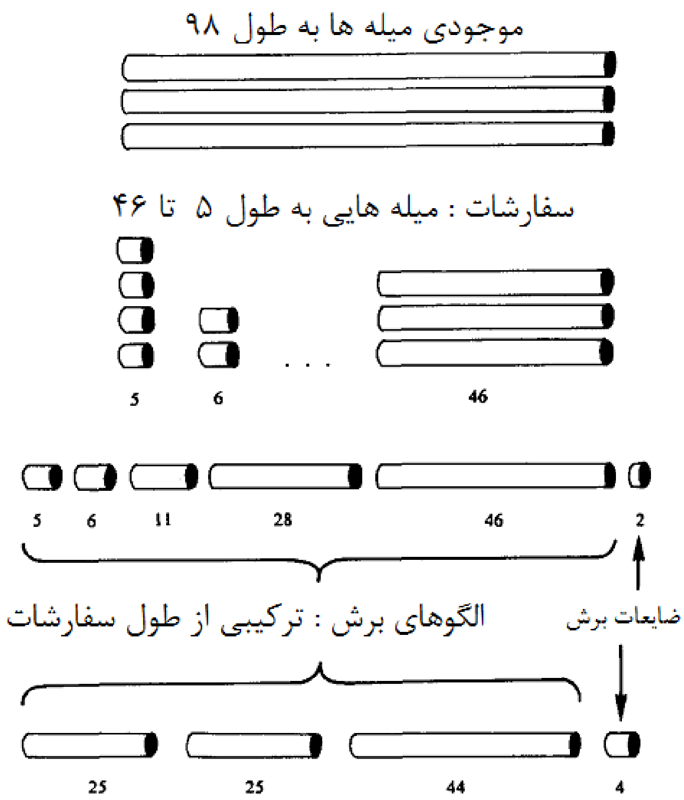
مسائل برش موجودی ابتدا با مسأله یک بعدی معرفی شده‌اند. به عنوان مثال در مسأله‌ای که هدف آن برش تعداد زیادی سفارش نسبتاً کوچک از چندین قطعه هم‌سان است، خروجی توپولوژی دایکوف به صورت $(۱/V/I/R)$ است، که ۱ به معنای یک بعدی بودن مساله، V به معنای تولید همه سفارشات با بخشی از موجودی، I به معنای تعداد زیادی موجودی یکسان و R به معنای سفارشات زیاد با ابعاد نسبتا کم می‌باشد.

حل مسائل برش موجودی دو بعدی سخت‌تر از مسائل یک بعدی است زیرا با پیچیدگی بیشتری برای تعیین الگوهای برش امکان‌پذیر مواجه می‌شویم. از این رو در مسائل دو بعدی تمرکز بیشتر بروی فرآیند تولید الگوها است [؟].

۱-۲-۱ مسائل برش موجودی یک بعدی

مسائل برش موجودی براساس ابعاد طبقه‌بندی می‌شوند. در مساله یک بعدی، برش‌های صورت گرفته تنها در یک بعد انجام می‌گردد و سایر ابعاد موجودی در نحوه برش بررسی نمی‌شوند. برای مثال در مقاله آقای هینکسمن [؟] ، در برش میله های

[1] Dyckhoff



شکل ۱-۲: نمایش موجودی، سفارشات، برش لوله ها و ضایعات[؟]

فولادی یکسان، سفارشات تنها در طول با یکدیگر متفاوت‌اند و سایر ابعاد مانند ضخامت و یا قطر با یکدیگر مشابهت دارند.

مثال ۱-۲-۱. با توجه به شکل ؟؟ موجودی، لوله هایی یکسان و به طول ۹۸ است که می خواهیم با ایجاد برش هایی به روی آن سفارشاتی به طول های ۵ تا ۴۶ تولید کنیم. به ترکیبی از طول های سفارش داده شده، الگوی برش گفته می‌شود. اگر مجموع طول سفارشات ترکیب شده، از طول موجودی بزرگتر شود، الگو نشدنی و درغیر این صورت الگو شدنی است. برای نمونه ترکیب سفارشاتی به‌طول‌های ۲۵و۴۴ و به صورت رابطه ؟؟ الگوی برش شدنی است ولی ترکیب همان سفارشات به صورت رابطه ؟؟ الگوی نشدنی بوجود می‌آورد.

$$۲۵ + ۲۵ + ۴۴ = ۹۴ \leq ۹۸ \qquad (۱-۱)$$

$$۲۵ + ۴۴ + ۴۴ = ۱۱۳ > ۹۸ \qquad (۲-۱)$$

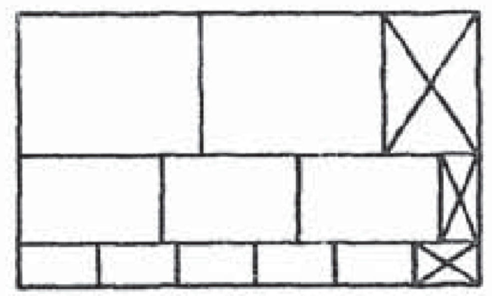
به مقدار باقیمانده پس از برش که غالبا کمتر از کوچکترین سفارش است ضایعات برش گفته می‌شود. بنابراین مقدار ضایعات الگوی برش ؟؟ برابر ۴ می‌شود.

مثال ۱-۲-۲. در این مثال، مقادیری از رول‌ها با عرض‌های مختلف و قطرهای یکسان باید از رول‌های موجودی با قطرها و عرض‌های استاندارد بریده شوند. بنابراین هدف، شناسایی الگوهای برش و میزان استفاده ازآن‌ها است؛ به‌طوریکه تقاضا برای

رول‌های سفارش داده شده، با کمترین ضایعات ممکن و دیگر فاکتورهای قابل کنترل برآورده شود. محدودیت امکان‌سنجی الگوی برش در این مساله این است که مجموع طول سفارشات، از طول موجودی تجاوز نکند [؟].

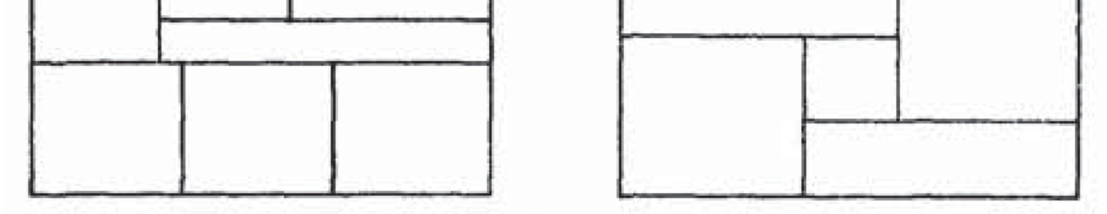
۱-۲-۲ مسائل برش موجودی دوبعدی مستطیل شکل

در حالت دو بعدی، موجودی به عنوان ورقه‌هایی مستطیل شکل نگهداری می‌شود و تقاضای مشتری مستطیل‌هایی با ابعاد کوچکتر است. ممکن است فرآیند برش شامل تعدادی مراحل مجزا باشد که در هرکدام تعدادی برش موازی در زیر ورقه‌های حاصل از مرحله قبلی، در زاویه سمت راست برش‌ها در مرحله قبلی، ایجاد می‌شود. این حالت برش دو بعدی دو مرحله‌ای نامیده می‌شود. شکل ؟؟ برش دو مرحله‌ای را شرح می‌دهد [؟].



شکل ۱-۳: الگوی برش دومرحله‌ای.X نشان‌دهنده نواحی ضایعات است.

در برش سه مرحله‌ای، ورقه‌های موجودی به بخش‌ها، بخش‌ها به نوارها و نوارها به قطعات سفارش داده شده، برش داده می‌شوند. اگر محدودیتی روی تعداد برشی که به‌منظور بدست آوردن مستطیل‌های سفارش داده شده می‌تواند ایجاد شود وجود نداشته باشد، و فقط یک محدودیت داشته باشیم که هر برش باید موازی با یک طرف ورق موجودی باشد، مساله به عنوان برش متعامد دوبعدی [۱] نامیده می‌شود. در مواردی مانند شیشه محدودیتی وجود دارد که هر برش باید بصورت برش گیوتینی [۲] ایجاد شود، یعنی باید عرض کامل ورقه‌ها یا برخی زیر ورقه‌هایی که از برش‌های قبلی تولید شده‌اند، ادامه داده شود. برش‌های متعامد گیوتینی و غیر گیوتینی در شکل ؟؟ شرح داده شده است [؟].

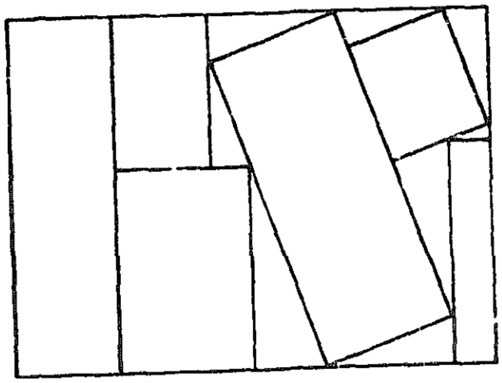


شکل ۱-۴: برش گیوتینی (چپ) و غیر گیوتینی(راست)

---

[۱] orthogonal 2-dimensional [۲] guillotine

دکانی [۳] [؟] معتقد است در برخی موارد با برش‌های غیر متعامد می‌توان از موجودی کمتری برای تولید سفارشات، استفاده نمود. مثالی از برش‌های نامتعامد در شکل ؟؟ شرح داده شده است [؟] .



شکل ۱-۵: برش نامتعامد

۱-۳ فرمول‌بندی مسائل برش موجودی

اولین مدل رياضي مسئله برش توسط، کانتروویچ [۱] در سال ۱۹۳۹ ارائه گردید. اگرچه این مدل در سال ۱۹۶۰منتشر شد، با تحقیق و بررسي‌هاي علمي صورت گرفته در مورد اين مسئله فرمول‌ها و روش‌هاي حل متفاوتي براي مسائل برش ارائه شده است [؟، ؟، ؟]. کانترویچ براي مسئله برش يك بعدي با تقاضا، مدلي را بيان مي‌کند که براي حل آن، بايد تمامي الگوهاي برش یک میله به قطعات مورد نياز شناسايي شوند، که اين امر تنها براي مسائل بسيار کوچك عملي خواهد بود.

لذا گیلمور گموری [۲] [؟] مسئله برش موجودی را بعنوان یک مسئله برنامه عدد صحیح فرمول‌بندی کردند که در آن متغیرهای تصمیم گیری تعداد دفعاتی که هر الگوی برش استفاده می‌شود، هستند. در مسئله برش موجودی هنگامی که تعداد اقلام مورد تقاضا افزایش می‌یابد، تعداد الگوهای ممکن و متغیرهای تصمیم گیری بصورت نمایی افزایش می‌یابد. بنابراین، گیلمور و گموری ابتدا الگوهای برش مفید را بوسیله حل یک مساله کمکی، تولید می‌کنند سپس آنها برنامه‌ریزی خطی ساده شده، را حل می‌کنند و فرآیند گردکردن را برای یک راه‌حل غیرصحیح معمول اعمال می‌کنند [؟]. این رویکرد اکتشافی اغلب به یک راه‌حل صحیح بهینه منتج می‌شود. این الگوریتم هم‌چنین به عنوان الگوریتم تولید ستون شناخته شده است.

با استفاده از این روش آنها مسأله برش دو بعدی را به عنوان مسأله برش یک بعدی دو مرحله‌ای در نظر گرفتند که در آن، اولین برش، موازی با یکی از اضلاع و سپس هر برش، عمود بر برش اول است. روش حل آنها بر مبنای برنامه‌ریزی عدد صحیح با تکنیک ایجاد ستون است. هر ستون، یک الگوی برش شدنی را ارائه می‌کند؛ الگویی که از طریق حل یک مسأله

[۳] De Cani [۱] Kantorvich [۲] Gilmore and Gomory

کوله‌پشتی دو بعدی ایجاد می‌شود [؟].

۱-۳-۱ فرمول‌بندی استاندارد مسئله برش موجودی

فرمول استاندارد برای مساله برش با لیستی از m سفارش آغاز می‌شود، هر سفارش به تعداد مورد نیاز است. سپس لیستی از تمام ترکیبات ممکن برش‌ها (لگو) ساخته می‌شود، برای هر الگو متغیر صحیح مثبت x_j نشان می‌دهد چه تعداد از الگو j مورد استفاده قرار گرفته است. فرمول‌بندی مساله برنامه‌ریزی خطی عدد صحیح با فرمول ؟؟ تعریف می‌شود. در این رابطه a_{ij} نشان‌دهنده تعداد دفعاتی است که سفارش i در الگوی j ظاهر می‌شود و C_j هزینه(اغلب ضایعات) الگوی j و d_i حداقل میزان سفارش i است.

$$Z = \min \sum_{j=۱}^{n} C_j x_j \tag{۱-۳}$$

$$s.t. \quad \sum_{j} a_{ij} x_j \geq d_i \qquad \forall i \in [۱..m], \tag{۱-۴}$$

$$x_j \geq ۰ \quad and \quad integer,$$

محدودیت مقدار در فرمول بالا محدودیت مینیمم است (یعنی با توجه به محدودیت ؟؟ حداقل مقدار داده شده از هر سفارش باید تولید شود). فرمول کلی‌تر این مساله دارای محدودیت دو طرفه (کران بالا و پایین) است (و در این حالت ممکن است کوچک‌ترین قطعه برش ما از کوچک‌ترین قطعه ضایعات کمتر باشد) . بنابراین، در فرمول بالا می‌توان محدودیت ؟؟ را جایگزین محدودیت ؟؟ کرد.

$$dl_i \leq \sum_{j=۱}^{n} a_{ij} x_j \leq DU_i \quad , \forall i \in [۱..m] \tag{۱-۵}$$

رابطه ؟؟ علاوه بر مسائل برش یک بعدی، برای مسائل با بیش از یک بعد و یا مسائل بیشینه سازی ارزش تولید براساس سفارشات با ارزش‌های متفاوت، قابل استفاده است. در این رابطه dl_i و DU_i کران‌های پایینی و بالایی برای سفارش i و مجوز مقدار تولید آن سفارش درون یک محدوده مشخص هستند. برای اینکه عناصر a_{ij} الگوی برش قابل اعمال در طول موجودی را تشکیل دهند، باید محدودیت زیر برآورده شود.

$$\sum_{i} a_{ij} l_i \leq L, \tag{۱-۶}$$

$$a_{ij} \geq ۰ \quad and \quad integer$$

به عبارت دیگر، در ایجاد الگوی برش، ترکیب سفارشات همیشه کوچکتر و یا مساوی از طول موجودی است.

$$C_j = L - \sum_i a_{ij} l_i \quad C_j \geq ۰ \tag{۷-۱}$$

L طول موجودی استفاده شده، l_i اندازه سفارش i و C_j هزینه(اغلب ضایعات) مربوط به الگوی j است.

۱-۳-۲ فرمول‌بندی مسئله برش موجودی به همراه هزینه راه‌اندازی

هدف اصلی در فرمول‌بندی مساله ضایعات برش کمینه کردن ضایعات برش است. در اغلب کاربردهای صنعتی، لازم است علاوه برضایعات سایر فاکتورها مانند هزینه راه اندازی، هزینه موجودی های متعدد غیر همسان،هزینه حمل و نقل و ... نیز ملاحظه شوند. بنابراین کنترل تعداد الگوهای استفاده شده به منظور برآورده کردن تقاضاهای سفارش داده شده، نقش بسیار مهمی در بدست آوردن جوابهای بهینه دارد [؟]. زیرا هر الگوی برش به راه‌اندازی اضافی روی ماشین برش احتیاج دارد. این امر بخصوص زمانی‌که راه‌اندازی الگو روی ماشین برش زمان‌بر و یا هزینه‌بر است، حائز اهمیت است. در اغلب موارد، لازم است موازنه‌ای بین مصرف مواد اولیه و تعداد راه‌اندازی‌ها به‌منظور رسیدن به طرح‌های تولیدی بهتر پیدا شود. در این مساله هزینه تولید شامل هزینه مواد و هزینه راه‌اندازی است. هربار که یک الگوی برش جدید استفاده می‌شود، به‌منظور راه‌اندازی روی ماشین برش یک هزینه راه‌اندازی وارد می‌شود. این مساله، مساله برش موجودی به همراه هزینه راه‌اندازی (CSP-S) [۱] نامیده می‌شود [؟]. می‌توان CSP-S را با برنامه‌ریزی غیرخطی عدد صحیح به صورت زیر فرمول بندی کرد.

$$\min C_s \sum Y_j + C_p L \sum X_j \tag{۸-۱}$$

$$s.t.$$

$$\sum a_{ij} x_j \geq d_i \quad \forall i \in I \tag{۹-۱}$$

$$\sum a_{ij} l_i \leq L Y_j \quad \forall j \in J \tag{۱۰-۱}$$

$$x_j, a_{ij} \in \mathbb{Z}^+ \quad \forall i \in I, j \in J \tag{۱۱-۱}$$

$$Y_j \in \{۰, ۱\} \quad \forall j \in J \tag{۱۲-۱}$$

معادله ؟؟ تابع هدف است که مجموع هزینه راه‌اندازی و تامین مواد اولیه است. برای هر مورد i، محدودیت ؟؟ ما را مطمئن می سازد که تقاضا برآورده شده است. محدودیت‌های ؟؟ تضمین می‌کند طول کل موارد برش در الگو j از طول موجودی تجاوز نمی‌کند. این محدودیت‌ها هم‌چنین تعیین می‌کنند که آیا الگو j استفاده شده است یا نه. در نهایت، محدودیت‌های ؟؟ و ؟؟ مقدار صحیح و غیر منفی متغیرهای تصمیم گیری را نشان می‌دهند. در اینجا،مقدار Y_j برابر یک است اگر از الگوی j استفاده شود و در غیر این صورت برابر صفر است. Cs نشان‌دهنده هزینه راه‌اندازی و Cp نشان‌دهنده

[۱] Cutting Stock Problem with Setup Cost

هزینه قلم موجودی در واحد طول است. $\mathbb{Z}^+$ به معنی مجموعه ای از اعداد صحیح نامنفی است [؟].

۱-۳-۳ فرمول‌بندی مسائل برش موجودی با سایزهای موجودی متعدد

درصورتی که اندازه موجودی (منابع) متفاوت باشد، آنگاه حالت‌های پیچیده‌تری برای تولید سفارشات خواهیم داشت. یکی از زمینه‌های پرکاربرد در مسئله برش موجودی، با موجودی‌های با اندازه‌های متفاوت این است که این موجودی‌ها از نظر مکانی نیز با یکدیگر فاصله داشته باشند. در این صورت هزینه حمل و نقل نیز بر انتخاب موجودی مورد استفاده موثر خواهدبود. این مسئله به صورت زیر فرمول بندی می‌شود.

$$\begin{aligned} &\min \sum_k (D_k \sum_j C_{jk}\, x_{jk} + \sum_i \sum_j S_{ki}\, a_{ijk}\, x_{jk}) \\ s.t. \quad & dl_i \le \sum_k \sum_j a_{ijk}\, x_{jk} \le DU_i \quad \forall i \in [۱..m] \\ & \sum_j x_{jk} \le M_k \quad \forall k \in \mathbb{N},\ x_{jk} \ge ۰ \quad and \quad integer \end{aligned} \tag{۱-۱۳}$$

a_{ijk} تعداد سفارشات i ام که با الگوی j در موجودی k تولید می‌شود.

x_{jk} تعداد موجودی‌هایی با اندازه k که توسط الگوی j برش داده می‌شوند.

C_{jk} ضایعات حاصل از اعمال الگوی j بر موجودی با اندازه k .

D_k هزینه مالی حاصل از اتلاف یک واحد از موجودی به اندازه k است.

S_{ki} هزینه حمل و نقل سفارش i از موجودی به اندازه k است. فرض برآن است که تمام موجودی‌های با اندازه k در مکان k قرار دارند. اگر تمام موجودی‌ها در یک مکان قرار داشته باشند این مقدار برابر با صفر خواهد بود.

M_k حداکثر تعداد موجودی به اندازه k است.

یک نمونه ساده شده از این مساله که توسط [؟] توسعه یافته، بصورت زیر است:

$$\begin{aligned} &\min \sum_j \sum_k C_{jk} x_{jk} \\ s.t. \quad & \sum_j \sum_k a_{ijk}\, x_{jk} \ge R_i \quad \forall i \in [۱..m] \\ & x_{jk} \ge ۰. \end{aligned} \tag{۱-۱۴}$$

برای حالت کلی با هزینه‌های مختلف در ضایعات و حمل و نقل، C_{jk} می‌تواند به شرح زیر بیان شود:

$$C_{jk} = C_k + C_{pk}(\sum_i a_{ijk}\, l_i) + \sum_i S_{ki} a_{ijk} \tag{۱-۱۵}$$

l_i اندازه سفارش i ام .

C_k هزینه موجودی با اندازه k .

C_{pk} هزینه تولید مفید هر اینچ استفاده شده از رول k .

S_{ki} هزینه حمل و نقل سفارش i از محل موجودی k است.

در این وضعیت، قبل از یافتن الگوی برش بعدی (درصورت وجود) که باید به پایه LP وارد شود، هزینه های سایه [۱] باید تعیین گردند. مساله کوله‌پشتی زیر باید برای هر موجودی به‌طول k حل شود.

$$\begin{aligned} Z_k = \max \sum_i (U_i - C_{pk}W_i - S_{ki})a_{ijk} - C_k \\ s.t. \quad \sum_i W_i a_{ijk} \leq UW_k \\ a_{ijk} \geq ۰ \quad and \quad integer. \end{aligned} \tag{۱-۱۶}$$

الگویی که به ازای آن Z_k بزگترین مقدار مثبت به دست می‌آید، وارد راه حل می‌شود. [؟].

۱-۳-۴ مسئله کوله پشتی

مسئله کوله پشتی، مسئله‌ای در بهینه‌سازی ترکیبیاتی است. فرض کنید مجموعه‌ای از اشیا، که هر کدام داری وزن و ارزش خاصی هستند در اختیار دارید. می‌خواهیم تعدادی از اشیاء را بگونه‌ای انتخاب کنیم که وزن اشیا انتخاب شده کوچکتر یا مساوی مقدار از پیش تعیین شده، و ارزش آنها بیشینه شود. علت نامگذاری این مسئله، جهانگردی است که کوله‌پشتی ای با اندازهٔ محدود دارد و باید آن را با مفیدترین صورت ممکن از اشیا پر کند.

فرض کنید n جسم x_i داریم که از ۱ تا n شماره گذاری شده‌اند. جسم i ام ارزشی معادل v_i و وزنی برابر با w_i دارد. معمولاً فرض می‌شود که وزن‌ها و ارزش‌ها نامنفی اند. برای ساده‌تر شدن نمایش، بدون کم شدن از کلیت مسئله می‌توان فرض کرد اشیا به ترتیب صعودی بر حسب وزنشان مرتب شده‌اند. بیشترین وزنی که می‌توان در کوله‌پشتی حمل کرد، W است. بنابراین می‌توان مسئله کوله‌پشتی را به صورت ؟؟ فرمول بندی نمود:

[۱] Shadow Costs

$$K = \max \sum_{i}^{n} v_i x_i$$
$$s.t. \sum_{i}^{n} w_i x_i \leq W$$
$$x_i \geq ۰ \quad and \quad integer \tag{۱۷-۱}$$

در رابطه ؟؟ اگر $x_i \in [۰, ۱]$ باشد، مسئله را کوله پشتی صفر و یک می نامیم. در این حالت از هر کالا تنها یک عدد موجود است که می‌توان آن را انتخاب نمود ($x_i = ۱$) و یا اینکه آن را انتخاب نکرد ($x_i = ۰$).

اگر از هریک کالاهای موجود تعدادی محدودی را بتوان انتخاب نمود ($x_i \in [۰, ۱, ۲, ..., c_{x_i}], \quad c_{x_i} \geq ۰$) مسئله را کوله‌پشتی کراندار می نامند و در غیر این‌صورت کوله‌پشتی بیکران نامیده می‌شود که در این حالت هیچ محدودیتی روی تعداد اشیا وجود ندارد و از هر شی، به هر تعداد دلخواهی می‌توان انتخاب کرد. برای حل این مسئله می‌توان از روش‌های مختلفی استفاده کرد. ازجمله این روش‌ها میتوان به الگوریتم ژنتیک، روش حریصانه، برنامه ریزی پویا، روش شاخه و برش و ... اشاره نمود.

در روش حریصانه ابتدا ارزش هر واحد وزنی از کالاهای موجود را با تقسیم ارزش هر کالا را بر وزن آن ($\frac{v_i}{w_i}$) محاسبه می‌کنیم. سپس ارزش‌های وزنی محاسبه شده را به ترتیب نزولی(بزرگ به کوچک) مرتب می‌نماییم. در این مرحله باتوجه به مقدار موجودی از هر کالا و به ترتیب بیشترین ارزش وزنی، کوله را پر می کنیم تا زمانیکه مجموع وزن کالاهای قرار گرفته در کوله از گنجایش آن بیشتر نشده باشد. ازآنجاییکه کالاهای انتخاب شده را بر اساس بیشترین ارزش وزنی انتخاب کردیم بنابراین درپایان ارزش کوله پرشده بیشینه است.

در روش برنامه ریزی پویا روال کار به این صورت است که :ابتدا مسئله بزرگ را به یک سری از مسائل کوچکتر و قابل حل تر تقسیم می کنیم، سپس از مسائل کوچکتر شروع به حل نموده و حاصل را در یک مکان ذخیره می نماییم و به تدریج به حل مسئله اصلی (بزرگترین مسئله) می‌پردازیم. در مسئله کوله پشتی صفر و یک اگر فرض کنیم A یک زیر مجموعه بهینه از n قطعه موجود باشد، دو حالت امکان پذیر است:

حالت اول : A حاوی قطعه n ام هست که دراین صورت ارزش کل برابر است با $v_n + \sum_{i=۱}^{n-۱} v_i$ البته به شرط آنکه رابطه وزنی $W \geq W_n + \sum_{i=۱}^{n-۱} W_i$ برقرار باشد.

حالت دوم: A حاوی قطعه n ام نیست که در این حالت ارزش A با ارزش زیر مجموعه ای بهینه از $n - ۱$ قطعه اول برابر است. به عبارت دیگر اگر $K_{w,i}$ ارزش بهینه حاصل از انتخاب i قطعه اول باشد به گونه ای که وزن آن از w بیشتر نشود داریم :

$$K_(w, i) = \max\{K(W - w_i, i - ۱) + v_i, K(w, i - ۱)\}$$

فصل ۲

مروری بر الگوریتم ژنتیک

۲-۱ مقدمه

در تئوری محاسبات مسائل تصمیم‌گیری به دو دسته تصمیم‌پذیر و تصمیم‌ناپذیر تقسیم می‌شوند. یک مسئله تصمیم‌پذیر مسئله‌ای است که قابل حل باشد به این معنی که بتوان یک الگوریتم برای آن طراحی کرد، در غیر این صورت مسئله مورد نظر تصمیم‌ناپذیر خواهد بود. مسائل تصمیم‌پذیر به نوبه خود و با توجه به مرتبه زمانی حل خود به دسته‌های متفاوتی تقسیم می‌شوند. دسته‌ای از آنها مسائلی هستند که برای آنها الگوریتمی با مرتبه زمانی چند جمله‌ای موجود است. این مسائل به کلاس مرتبه زمانی چند جمله‌ای تعلق دارند. از طرف دیگر مسائلی وجود دارند که اثبات شده است الگوریتمی با مرتبه زمانی چند جمله‌ای برای آنها وجود ندارد. همچنین مسائلی وجود دارند که تعلق یا عدم تعلق آنها به کلاس مرتبه زمانی چند جمله‌ای اثبات نشده است. تمام مسائل NP-Complete و برخی از مسائل NP-Hard از این دسته مسائل هستند.

تمام مسائل NP-Complete به طور موثری قابل کاهش به یکدیگر هستند. بنابراین اگر یکی از آنها با یک مرتبه زمانی چند جمله‌ای بر حسب اندازه مسئله حل شود، تمامی آنها با این مرتبه زمانی قابل حل خواهند بود [؟]. کلاس NP-Hard شامل بسیاری از مسائل بهینه‌سازی است. امروزه بسیاری از مسائل بهینه‌سازی ترکیبی [۲] که اغلب ازجمله مسائل با درجه غیر چندجمله‌ای (NP-Hard) [۳] هستند، به‌صورت تقریبی با کامپیوترهای موجود قابل حل است. ازجمله راه‌حل‌های موجود در برخورد با اینگونه مسائل، استفاده از الگوریتم‌های تقریبی است. این الگوریتم‌ها تضمینی نمی‌دهند که جواب به دست آمده بهینه باشد و تنها با صرف زمان بسیار می‌توان جواب نسبتاً دقیقی به‌دست آورد و در حقیقت بسته به زمان صرف شده، دقت جواب تغییر می‌کند. الگوریتم‌های تقریبی نیز شامل الگوریتم‌های ابتکاری و فراابتکاری هستند.

---

[۲] combinatorial optimization [۳] non-deterministic polynomial-time hard

۲-۱-۱ الگوریتم‌های ابتکاری و فراابتکاری

معیارها، روش‌ها و یا اصولی برای تصمیم گیری بین چند راه‌حل و انتخاب اثربخش‌ترین آن‌ها برای دستیابی به اهداف مورد نظر را الگوریتم‌های ابتکاری گویند؛ این الگوریتم‌ها نتیجه برقراری اعتدال بین دو نیاز هستند: نیازبه ساخت معیارهای ساده و توانایی تمایز درست بین انتخاب‌های خوب و بد. خاصیت روش‌های ابتکاری خوب این است که ابزار ساده‌ای برای تشخیص خط مشی‌های بهتر ارائه می‌دهند و اگرچه به صورت شرط لازم، تشخیص خط مشی‌های اثربخش را تضمین نمی‌کنند اما به صورت شرط کافی این تضمین را فراهم می‌آورند. بیشتر مسائل پیچیده نیازمند ارزیابی تعداد انبوهی از حالات ممکن برای تعیین یک جواب دقیق هستند. زمان لازم برای یافتن یک جواب دقیق اغلب بسیار طولانی است. الگوریتم‌های ابتکاری با استفاده از روش‌های نیازمند ارزیابی‌های کمتر و ارائه جواب‌هایی در محدودیت‌های زمانی قابل قبول، دارای نقشی اثربخش در حل مسائل خواهند بود. در حالت کلی ۳ دسته از الگوریتم‌های ابتکاری قابل تشخیص است [؟]:

- الگوریتم‌هایی که بر ویژگی‌های ساختاری مسئله و ساختار جواب متمرکز می‌شوند و با استفاده از آنها الگوریتم‌های سازنده یا جستجوی محلی را تعریف می‌کنند.

- الگوریتم‌هایی که بر هدایت اکتشافی یک الگوریتم سازنده یا جستجوی محلی متمرکز می‌شوند به گونه‌ای که آن الگوریتم بتواند بر شرایط حساس (مانند فرار از بهینه محلی) غلبه کند.

- الگوریتم‌هایی که بر ترکیب یک چارچوب یا مفهوم ابتکاری یا گونه‌هایی از برنامه ریزی ریاضی (معمولا روش‌های دقیق) متمرکز می‌شوند.

الگوریتم‌های ابتکاری نوع اول خیلی خوب می‌توانندعمل کنند(گاهی اوقات تا حد بهینگی)، اما ممکن است در جواب‌هایی باکیفیت پائین (بهینه‌های محلی) گیر کنند. برای بهبود این الگوریتم‌ها از اواسط دهه هفتاد موج تازه‌ای از رویکردها آغاز گردید. این رویکردها شامل الگوریتم‌هایی است که صریحا یا به طور ضمنی تقابل بین ایجاد تنوع جستجو (وقتی علائمی وجود دارد که جستجو به سمت مناطقی فضای جستجو میرود که شامل پاسخ مساله نیست) و تشدید جستجو (با هدف یافتن بهترین جواب در منطقه مورد بررسی) را مدیریت می‌کنند. این الگوریتم‌ها، فراابتکاری نامیده می‌شوند [؟].

الگوریتم های فراابتکاری یک الگوریتم فراابتکاری یک چارچوب الگوریتمی است که می‌تواند با تغییرات کم، برای مسائی بهینه‌سازی مختلف به‌کار رود. استفاده از الگوریتم‌های فراابتکاری به طور قابل ملاحظه‌ای توانایی یافتن جواب‌هایی با کیفیت بالا را برای مسائل بهینه‌سازی ترکیبی افزایش می‌دهد. به‌عبارت دیگر، یک الگوریتم فراابتکاری روشی ابتکاری است که قادر به جستجوی فضای جواب، برای یافتن جواب‌های باکیفیت بالا است [؟، ؟] . روش‌های فراابتکاری دارای ویژگی‌های زیر هستند [؟، ؟]:

- این روش ها، همگی تاحدودی احتمالی هستند، چنین رویکردی این امکان را می‌دهد که از قرار گرفتن الگوریتم در بهینه محلی جلوگیری شود.

- این روش‌ها معمولا برای حل مسائل گسسته ارائه شده‌اند، اما امکان کاربرد در مسائل پیوسته را دارند.

- این روشها معمولا از مفاهیم زیست شناسی، رفتارشناسی جانوری (مانند روش الگوریتم کلونی موچه‌ها [1] ، الگوریتم زنبور عسل [2] ، الگوریتم پرواز پرندگان [3]) و فیزیک (روش شبیه سازی تبریدی [4])الهام گرفته شده‌اند.
- از معایب مشترک در این روش‌ها، دشوار بودن تنظیم و تطبیق پارامترها و ارزیابی دقت جواب بدست آمده است.

۲-۱-۲ روش شمارش کامل

یک روش برای حل مسائل بهینه‌سازی، شمارش کامل است که درآن تمامی جواب‌های شدنی درنظر گرفته شود و توابع هدف مربوط به آن محاسبه شود و درنهایت، بهترین جواب انتخاب گردد. روشن است که شیوه شمارش کامل، نهایتاً به جواب دقیق مسأله منتهی می‌شود؛ اما در عمل به دلیل زیاد بودن تعداد جواب‌های شدنی، استفاده از آن در بسیاری از مسائل غیرممکن است. با توجه به مشکلات مربوط به روش شمارش کامل، همواره بر ایجاد روش‌های مؤثرتر و کاراتر تأکید شده است. در این زمینه، الگوریتم‌های مختلفی به وجود آمده است که مشهورترین نمونه آنها، روش سیمپلکس برای حل برنامه‌های خطی و روش شاخه و کران برای حل برنامه‌های خطی با متغیرهای صحیح است. برای مسائلی با ابعاد بزرگ، روش سیمپلکس از کارایی بسیار خوبی برخوردار است، ولی روش شاخه و کران کارایی خود را از دست می‌دهد و عملکرد بهتری از شمارش کامل نخواهد داشت. به دلایل فوق، اخیراً تمرکز بیشتری بر روشهای ابتکاری یا فرا ابتکاری [1] یا جستجوی تصادفی [2] صورت گرفته است. در این فصل به بیان الگوریتم ژنتیک و ساختار و نحوه عملکرد آن در حل مسائل پرداخته خواهد شد.

۲-۲ الگوریتم ژنتیک

الگوریتم ژنتیک برای بهینه‌سازی، جستجو و یادگیری ماشین [3] مورد استفاده قرار می‌گیرد. اساس این الگوریتم قانون تکامل داروین (بقا بهترین) است که می‌گوید: موجودات ضعیف‌تر از بین می‌روند و موجودات قوی‌تر باقی می‌مانند. در واقع، قانون انتخاب طبیعی برای بقا می‌گوید که هرچه امکان تطبیق موجود بیشتر باشد بقای موجود امکان‌پذیرتر است و احتمال تولید مثل بیشتری، برایش وجود دارد. این قانون بر اساس پیوند بین رشته‌ها و عملکرد ساختمان‌های رمزگشایی شده آنها است. الگوریتم ژنتیک به دلیل تقلید نمودن از طبیعت دارای تفاوت‌های عمده‌ای با روش‌های جستجو و بهینه‌سازی متداول دارد که برخی از این تفاوت‌ها به‌صورت خلاصه عبارتند از:

- الگوریتم ژنتیک با رشته‌های بیتی از مجموعه جواب‌ها کارمی‌کند نه با خود آنها.
- الگوریتم ژنتیک برای راهنمایی جهت جستجو، انتخاب تصادفی [4] انجام می‌دهد .

الگوریتم‌های ژنتیکی را می‌توان یک روش بهینه‌سازی تصادفی جهت‌دار دانست که به تدریج به سمت نقطه بهینه حرکت می‌کند. در مورد ویژگی‌های الگوریتم ژنتیک در مقایسه با دیگر روش‌های بهینه‌سازی می‌توان گفت که الگوریتمی ست که

---

[1] Ant colony optimization algorithms [2] Artificial bee colony algorithm [3] Bird Flocks [4] Simulated annealing [1] Metaheuristic [2] Random Method [3] machine learning [4] Random

بدون داشتن هیچ گونه اطلاعی از مسئله و هیچ گونه محدودیتی بر نوع متغیرهای آن برای هرگونه مسئله‌ای قابل اعمال است و دارای کارآیی اثبات شده‌ای در یافتن بهینه کلی [۵] است. توانایی این روش در حل مسائل پیچیده بهینه‌سازی است زیرا در پیاده‌سازی آن محدودیت‌هایی مانند مشتق‌پذیری و ... لازم نیست. در مسائل پیچیده بهینه‌سازی روش‌های کلاسیک یا قابل اعمال نیستند و یا دریافتن بهینه کلی قابل اطمینان نیستند [؟].

- الگوریتم ژنتیک در هر تکرار، همه فضای جستجو را کاووش نمی‌کند بلکه چند نقطه از فضای جستجو را در نظر می‌گیرد؛ هرچقدر پراکندگی این نقاط در فضای جستجو یکنواخت تر باشد، شانس اینکه به یک ماکزیمم محلی همگرا شود کاهش می‌یابد.

در مکانیزم جستجو گرچه مقدار تابع هدف در تمام فضای جواب محاسبه نمی‌شود ولی مقدار محاسبه شده تابع هدف برای هر نقطه، در متوسط‌گیری آماری تابع هدف برای آن و یا در کلیه زیر فضاهایی که آن نقطه به آنها وابسته بوده دخالت داده می‌شود و این زیر فضاها به‌طور موازی از نظر تابع هدف متوسط‌گیری آماری می‌شوند. این مکانیزم را توازی ضمنی [۱] می‌گویند. این روند باعث می‌شود که جستجوی فضا به نواحی از آن که متوسط آماری تابع هدف در آنها زیاد بوده و امکان وجود نقطه بهینه مطلق در آنها بیشتر است تمایل پیدا کند. چون در این روش برخلاف روش‌های تک‌مسیری فضای جواب به طور همه جانبه جستجو می‌شود، امکان کمتری برای همگرایی به یک نقطه بهینه محلی وجود خواهد داشت.

- الگوریتم ژنتیک از قواعد انتقال احتمالی استفاده می‌کند [؟]. مکانیزم‌های تصادفی که روی انتخاب و حذف رشته‌ها عمل می‌کنند به‌گونه‌ای هستند که رشته‌هایی که عدد برازندگی بیشتری دارند، احتمال بیشتری برای ترکیب و تولید رشته‌های جدید داشته و در مرحله جایگزینی نسبت به دیگر رشته‌ها مقاوم‌تر هستند. بدین لحاظ جمعیت دنباله‌ها در یک رقابت بر اساس تابع هدف در طی نسل‌های مختلف، کامل شده و متوسط مقدار تابع هدف در جمعیت رشته‌ها افزایش می‌یابد. بنابراین در هر مرحله از اجرای الگوریتم ژنتیکی، یک دسته از نقاط فضای جستجو مورد پردازش‌های تصادفی قرار میگیرند که در آخر براساس این که تابع هدف در هر یک از نقاط چه مقدار باشد، احتمال شرکت نمودن آنها در مرحله بعد تعیین می گردد [؟، ؟].

بقای جمعیت در الگوریتم ژنتیک، براساس میزان مطلوبیت آن در محیط تعریف می‌شود. به روش‌های مختلفی انتخاب جمعیت انجام می‌شود؛ وظیفه روش انتخاب، پیدا کردن کاندیداهایی است که ترکیب آن‌ها سبب بهبود راه‌حل‌ها در نسل بعد می‌شود [؟]. افرادی با قابلیت های برتر، شانس ازدواج وتولید مثل بیشتری را خواهند یافت. بنابراین بعد از چند نسل فرزندانی با کارایی بهتر بوجود می‌آیند.

در الگوریتم ژنتیک هر فرد از جمعیت به‌صورت یک کروموزوم معرفی می‌شود. کروموزوم ها در طول چندین نسل کامل‌تر می‌شوند. در هر نسل کروموزوم ها ارزیابی می‌شوند و متناسب با ارزش خود امکان بقا و تکثیر می‌یابند. تولید نسل در بحث الگوریتم ژنتیک با عملگرهای برش [۲] و جهش [۳] صورت می‌گیرد. والدین برتر بر اساس یک تابع برازندگی [۴] انتخاب می‌شوند.

---

[۵]Global Optimum [۱]Implicit Parallelism [۲]Cross Over [۳]Mutation operators [۴]Fitness function

0	1	1	0	1	0	0	1
g0	g1	g2	g3	g4	g5	g6	g7

شکل ۲-۱: شکل یک کروموزوم

۲-۲-۱ ساختار الگوریتم ژنتیک

به طور کلی، الگوریتم های ژنتیک از کروموزوم، تابع برازندگی، عملگرهای ژنتیکی، عملگرهای انتخاب، عملگر برش، عملگر جهش و شرط خاتمه تشکیل می‌شوند.

- کروموزوم

در الگوریتم‌های ژنتیک، هر کروموزوم نشان دهنده یک نقطه در فضای جستجو و یک راه حل ممکن برای مسئله مورد نظر است. خود کروموزوم ها (راه‌حل‌ها) از تعداد ثابتی ژن [1] (متغیر) تشکیل می‌شوند. برای نمایش کروموزوم‌ها، معمولاً از کدگذاریهای دودویی (رشته های بیتی) مانند شکل ؟؟ استفاده می‌شود.

- جمعیت

مجموعهای از کروموزوم‌ها یک جمعیت را تشکیل می‌دهند. با تاثیر عملگرهای ژنتیکی بر روی هر جمعیت، جمعیت جدیدی با همان تعداد کروموزوم تشکیل می شود.

- تابع برازندگی [2]

به‌منظور حل هر مسئله با استفاده از الگوریتمهای ژنتیک، ابتدا باید یک تابع برازندگی برای آن مسئله ابداع شود. برای هر کروموزوم، این تابع عددی غیر منفی را برمیگرداند که نشان دهنده شایستگی یا توانایی فردی آن کروموزوم است.

تابع برازندگی از اِعمال تبدیل مناسب بر روی تابع هدف یعنی تابعی که قرار است بهینه شود به دست می‌آید. این تابع هر رشته را با یک مقدار عددی ارزیابی می‌کند که کیفیت آن را مشخص می‌نماید. هر چه کیفیت رشته جواب بالاتر باشد مقدار برازندگی جواب بیشتر است و احتمال مشارکت برای تولید نسل بعدی نیز افزایش خواهد یافت. بسته به اینکه مسأله مورد نظر بیشینه‌سازی یا کمینه‌سازی باشد برازندگیِ بیشتر مترادف با بیشینه یا کمینه بودن تابع هدف خواهد بود، از آنجایی که الگوریتم ژنتیک با جستجو در فضای کروموزوم‌ها به دنبال بیشینه تابع برازندگی است [؟] ، باید مسائل کمینه‌سازی به بیشینه‌سازی تبدیل شود. چندین روش برای تبدیل تابع هدف برازندگی وجود دارد. ساده‌ترین روش مساوی قرار دادن تابع برازندگی با تابع هدف است. این روش در مسائلی که تابع هدف بایستی بیشینه شود مناسب است. برای تبدیل مسائل کمینه‌سازی به بیشینه‌سازی روش‌های مختلفی وجود دارد، با فرض

[1] Gene [2] merit function

اینکه مقدار تابع هدف معادل فرد i ام ϕ_i باشد، ساده‌ترین روش، کم کردن ϕ_i از یک مقدار ثابت C است بطوریکه به ازای تمام نسل‌ها $\phi_i < C$ باشد. در این‌صورت ϕ_i مقدار کمینه است اگر به ازای آن f_i بیشینه باشد.

$$f_i = C - \phi_i$$

در صورتی که نتوانیم بزرگترین مقدار تابع هدف را به‌صورت تقریبی بیابیم، می‌توانیم در هر نسل $\phi_{\max}$ و $\phi_{\min}$ یافته و برازندگی را به صورت زیر محاسبه کنیم.

$$f_i = (\phi_{\max} + \phi_{\min}) - \phi_i$$

روش دیگر، استفاده از تابع نمائی $f_i = e^{-\phi_i}$ است.

۲-۲-۲ عملگرهای ژنتیکی

در الگوریتم‌های ژنتیک، در طی مرحله تولید مثل [۱] ازعملگرهای ژنتیکی استفاده می‌شود. با تاثیر این عملگرها بر روی یک جمعیت، نسل [۲] بعدی آن جمعیت تولید می‌شود. عملگرهای انتخاب [۳]، برش و جهش معمولاً بیشترین کاربرد را در الگوریتم‌های ژنتیکی دارند. در اینجا، هر یک از عملگرهای فوق به صورت جداگانه معرفی می‌شود:

۲-۲-۲-۱ عملگر انتخاب

این عملگر از بین کروموزومهای موجود در یک جمعیت، تعدادی کروموزوم را برای تولید مثل انتخاب می‌کند. کروموزوم‌های برازنده تر شانس بیشتری دارند تا برای تولید مثل انتخاب شوند [؟].

در مرحله انتخاب، یک جفت از کروموزوم‌ها برگزیده می‌شوند تا با هم ترکیب شوند، عملگر انتخاب رابط بین دو نسل است و بعضی از اعضای نسل کنونی را به نسل آینده منتقل می‌کند، بعد از انتخاب، عملگرهای ژنتیک روی دو عضو برگزیده اعمال می‌شوند، معیار انتخاب اعضاء ارزش تطابق آنهاست اما روند انتخاب حالتی تصادفی دارد. شاید انتخاب مستقیم و ترتیبی به این شکل که بهترین اعضاء دوبه‌دو انتخاب شوند در نگاه اول روش مناسبی به نظر برسد اما باید به نکته‌ای توجه داشت؛ در الگوریتم ژنتیک ما با انتخاب ژن‌ها روبه‌رو هستیم، یک عضو با تطابق پایین اگرچه در نسل خودش عضو مناسبی نیست اما ممکن است شامل ژن‌های خوب باشد و اگر شانس انتخاب شدنش صفر باشد، این ژن‌های خوب نمی‌توانند به نسل‌های بعد منتقل شوند. پس روش انتخاب باید به گونه‌ای

[۱] Reproduction [۲] Generation [۳] Selection

باشد که به این عضو نیز شانس انتخاب شدن داده شود. راه‌حل مناسب، طراحی روش انتخاب به گونه‌ای است که احتمال انتخاب شدن اعضای با تطابق بالاتر بیشتر باشد. انتخاب باید به گونه‌ای صورت گیرد که تا حد ممکن هر نسل جدید نسبت به نسل قبلی‌اش تطابق میانگین بهتری داشته باشد.

دو روش متداول انتخاب عبارت اند از انتخاب چرخ رولت[۱] و نخبه سالاری[۲] که در ادامه به آن می‌پردازیم.

۱. انتخاب چرخ رولت

انتخاب چرخ رولت که اولین بار توسط «هولند» [۳] پیشنهاد شد یکی از مناسب‌ترین انتخاب‌های تصادفی بوده که ایدهٔ آن، احتمال انتخاب است. احتمال انتخاب متناظر با هر کروموزوم، براساس برازندگیِ آن محاسبه شده که اگر مقدار برازندگی کروموزوم k ام باشد، احتمال بقای متناظر با آن کروموزوم عبارت است از:

$$p_k = \frac{f_k}{\sum_i^n f_i} \tag{۱-۲}$$

رابطه ؟؟ احتمال انتخاب در روش چرخ رولت است.

اکنون کروموزوم‌ها را براساس p_k مرتب کرده و q_k که همان مقادیر تجمعی p_k است، به صورت زیر به دست می‌آید:

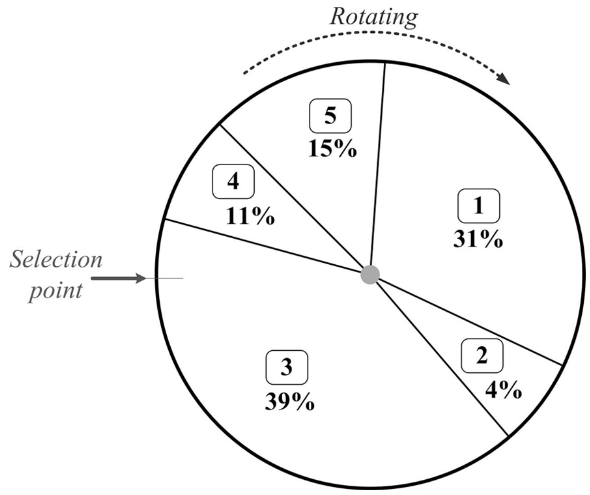
$$q_k = \sum_i^k p_i \tag{۲-۲}$$

چرخ رولت برای انتخاب هر کروموزوم یک عدد تصادفی بین یک و صفر تولید کرده سپس با توجه به بازه‌ای که عدد مذکور در آن قرار می‌گیرد، کروموزوم مربوطه انتخاب می‌شود. البته روش پیاده‌سازیِ چرخ رولت به این شکل است که ما یک دایره را در نظر گرفته و آن را به تعداد کروموزوم‌ها طوری تقسیم می‌کنیم که هر بخش متناظر با مقدار برازندگی کروموزوم مربوط باشد، اکنون چرخ را چرخانده و هر کجا که چرخ متوقف شد به شاخص چرخ نگاه کرده، کروموزوم مربوط به آن بخش انتخاب می‌گردد. شکل ؟؟ نشان می‌دهد که هرچه مقدار برازندگی کروموزومی بیشتر باشد، فضای بیشتری از چرخ رولت را به خود اختصاص می‌دهد؛ بنابراین احتمال انتخاب آن نیز بیشتر می‌شود.

انتخاب چرخ رولت، روشی است که نسبت مقدار تطابق، اعضاء را انتخاب می‌کند. این روش یک چرخ رولت را شبیه‌سازی می‌کند تا تعیین کند کدام اعضاء شانس باز تولید را دارند. هر عضو به نسبت تطابقش، تعدادی از بخش‌های چرخ رولت را به خود اختصاص می‌دهد. سپس در هر مرحله انتخاب یک عضو برگزیده می‌شود و روند آنقدر تکرار می‌شود تا به اندازه کافی، جفت برای تشکیل نسل بعد انتخاب گردد. این روش انتخاب را می‌توان به صورت زیر بیان کرد:

---

[۱] Roulette wheel selection [۲] Elitist Selection [۳] John Henry Holland



شکل ۲-۲: نحوه انتخاب کوروموزوم برتر توسط چرخ رولت

برداری مانند v در نظر بگیرید:

$$v = [۱, ..., M]$$

M تعداد عناصر بردار است. اگر تعداد اعضای مجموعه N باشد، هر عضو i دارای تطابقی مانند f_i است. هر عضو i به نسبت f_i ، p_i بار در v تکرار می‌شود. هر چه f_i بیشتر باشد، عضو مکان‌های بیشتری را به خود اختصاص می‌دهد. پس از تشکیل بردار v یک مقدار تصادفی $۱ \leq r \leq M$ انتخاب می‌شود. این مقدار به مکانی در بردار اشاره می‌کند که آن مکان خود معرّف عضوی از اعضای جمعیت است.

مثال ۲-۲-۱. فرض کنیم مجموعه ای با چهار نوع عنصر مختلف ($n = ۴$) داشته باشیم به‌صورتیکه تعداد اعضای مشابه هر یک از این عناصر به شرح زیر باشد :

$$f_۱ = ۱۰, \quad f_۲ = ۱۰, \quad f_۳ = ۱۵, \quad f_۴ = ۲۵$$

حال می‌خواهیم بدانیم که عدد $r = ۳۲$ (یک عدد تصادفی)منتسب به کدام عنصر است .

برای این منظور، میدانیم که مقدار مجموع اعضای عناصر عبارت است از:

$$\sum_{i=۱}^{n} f_i = ۶۰$$

بردار v را برداری با ۶۰ عنصر در نظر می‌گیریم. این بردار به صورت زیر پر می‌شود:

به عضو ۱، ۱۰ مکان، به عضو ۲، ۱۰ مکان، به عضو ۳، ۱۵مکان و به عضو ۴، ۲۵ مکان اختصاص می‌یابد.

اکنون r بین ۱ تا ۶۰ به تصادف انتخاب می‌شود. فرض کنید r=۳۲ بنابراین بازه سوم انتخاب می‌شود.

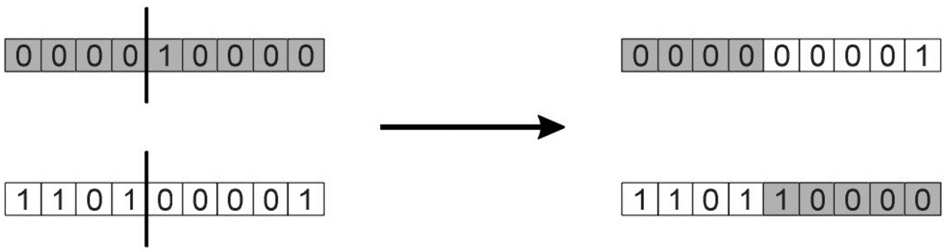
۲. انتخاب نخبه گرایی

ایدهٔ نخبه‌سالاری یا نخبه گرایی، ویژگی تازه‌ای به پروسهٔ انتخاب اضافه می‌کند، در نخبه سالاری، بهترین عضو هر جمعیت، زنده می‌ماند و در جمعیت بعد حضور دارد، به عبارت دیگر عضوی که بالاترین تطابق را دارد به طور خودکار به جمعیت جدید منتقل می‌شود. این روش ابتدا در سال ۱۹۷۵ توسط «کندی جونز» معرفی شد. اعمال نخبه سالاری در الگوریتم ژنتیک، معمولاً باعث بهبود کارایی آن می‌شود.

۲-۲-۲-۲ انتخاب عملگر برش

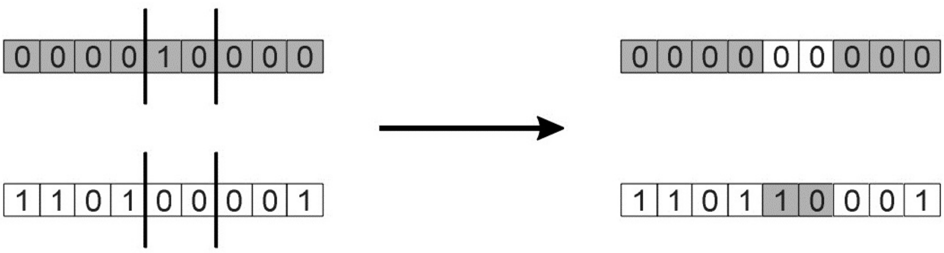
عملگر برش بر روی یک زوج کروموزوم از نسل مولد عمل کرده و یک زوج کروموزوم جدید تولید می‌کند. این عملگر زیرتوالی‌های دو کروموزوم را باهم مبادله می‌کند [؟]. عملگرهای برش متعددی از قبیل، برش تک‌نقطه‌ای [۱] و برش دو نقطه‌ای [۲] وجود دارد.

در برش تک‌نقطه‌ای، یک موقعیت تصادفی بین دو ژن در نظر گرفته می‌شود. سپس تمامی ژن‌های طرف راست یا طرف چپ این موقعیت در کروموزوم‌های والد با یکدیگر جابجا میشوند تا کروموزوم‌های جدید بدست آیند. در شکل ؟؟ برش تک نقطه‌ای نشان داده شده است. در برش دو نقطه‌ای، دو موقعیت به صورت تصادفی انتخاب



شکل ۲-۳: برش تک نقطه‌ای[؟]

می‌شود و تمامی ژن‌های بین این دو موقعیت در کروموزوم‌های والد با یکدیگر جابجا می‌شوند. در شکل ؟؟ برش دو نقطه‌ای نشان داده شده است.



شکل ۲-۴: برش دو نقطه‌ای[؟]

---

[۱] One-point Crossover [۲] Two-point Crossover

لازم به ذکر است که برش معمولاً برروی همه زوج کروموزوم‌های انتخاب شده برای ترکیب به کار برده نمی شود. معمولاً احتمال برش برای هر زوج کروموزوم بین ۰.۶ تا ۰.۹۵ در نظر گرفته می‌شود که به این عدد نرخ برش [۳] یا احتمال برش [۴] گفته میشود و با Pc نمایش داده میشود. در صورتی که بر روی یک زوج کروموزوم عمل برش صورت نگیرد، فرزندان با تکرار نمودن والدین تولید می‌شوند.

۲-۲-۲-۳ عملگر جهش

شکل ۲-۵: یک کروموزوم قبل و بعد از اعمال عملگر جهش [؟]

پس از اتمام عمل برش، عملگر جهش بر روی کروموزوم‌ها اثر داده می‌شود. این عملگر برخی بیت‌ها از یک کروموزوم را به طور تصادفی انتخاب نموده و سپس محتوای آن‌ها را تغییر می‌دهد. اگر ژن از جنس اعداد دودویی باشد، آن را به وارونش تبدیل می‌کند و چنانچه متعلق به یک مجموعه باشد، مقدار یا عنصر دیگری از آن مجموعه را به جای آن ژن قرار می‌دهد. در شکل ؟؟ چگونگی جهش یافتن ژن سوم، چهارم و هشتم از یک کروموزوم نشان داده شده است.

به طور کلی جهش از قرار گرفتن الگوریتم ژنتیک در اکسترمم‌های محلی جلوگیری می‌کند. در شکل ؟؟ نقاط اکسترمم محلی مشخص گردیده است. دراین شکل نقطه $LocalMax1$ را می توان به عنوان نطقه بیشینه سراسری هم تعیین کرد. کاربرد جهش آنجایی دیده می‌شود که فضای جستجوی ما در یکی از تکرارها اطراف نقاط بیشینه های محلی ۲ و یا ۳ باشد. در این صورت فرآیند الگوریتم ژنتیک ما را به سمت این بیشینه های محلی هدایت می کند تا زمانی که با یک جهش تصادفی فضای جستجو را به اطراف نقطه $LocalMax1$ تغییر دهد، در این صورت الگوریتم ژنتیک دراین مرحله به بیشینه سراسری می‌رسد. استفاده از جهش نباید زیاد صورت گیرد زیرا در این صورت الگوریتم ژنتیک به جستوی کاملا تصادفی تبدیل خواهد شد.

شکل ۲-۶: مثالی از اکسترمم های محلی و سراسری

[۳] Crossover Rate [۴] Crossover Probability

جهش می‌تواند در هر موقعیت بیتی در یک رشته با یک احتمال رخ دهد [؟]. احتمال انجام عمل جهش برروی هر کروموزوم را نرخ جهش [1] یا احتمال جهش [2] میگویند. معمولاً این عدد را بسیار کوچک (مثلاً ۰.۰۰۱) در نظر می‌گیرند [؟] .پس از اتمام عمل جهش، کروموزوم‌های تولید شده به عنوان نسل جدید شناخته شده و برای دور بعد اجرای الگوریتم ارسال می‌شوند.

۲-۲-۳ چارچوب کلی الگوریتم ژنتیک

نحوه عملکرد الگوریتم ژنتیک به طور کلی در الگوریتم ؟؟ آورده شده است.

الگوریتم ۲-۱ چارچوب کلی الگوریتم ژنتیک [؟].

---

۱. تولید یک جمعیت اولیه به صورت تصادفی از n کروموزوم l بیتی .

۲. ارزیابی تابع برازندگی برای هر کروموزوم در جمعیت.

۳. تکرار مراحل زیر تا زمانیکه n فرزند ایجاد شود

(آ) یک جفت کروموزوم والد از جمعیت فعلی انتخاب می‌شود، احتمال انتخاب تابعی افزایشی از برازش است. انتخاب "همراه با جایگزین" انجام می‌شود، به این معنی که یک کروموزوم می‌تواند بیش از یک بار به عنوان والد انتخاب شود.

(ب) روی جفت انتخابی با احتمال p_c ("احتمال برش" یا "نرخ برش")، برش صورت می‌گیرد. برش در نقطه‌ای که به صورت تصادفی انتخاب (انتخاب با احتمال یکنواخت) می‌شود، انجام خواهد شد تا دو فرزند جدید شکل گیرند. درصورتیکه عمل برش انجام نشود، دو فرزند دقیقا رونوشت پدر و مادر خود هستند.

(ج) در اینجا نرخ برش احتمال این که روی والدین در بیش از یک نقطه برش صورت گیرد، تعریف می‌شود. "برش چند نقطه‌ای" در نسخه های الگوریتم ژنتیک که نرخ برش برای یک جفت والد، تعداد نقاطی که در آن برش صورت می‌گیرد است، هم وجود دارد.

(د) در هر مکان دو فرزند با احتمال p_m (احتمال جهش یا نرخ جهش) جهش صورت می‌گیرد، و کروموزوم‌های حاصل در جمعیت جدید قرار داده می‌شوند.
اگر n فرد باشد، یک عضو جدید جمعیت می تواند به صورت تصادفی دور انداخته شود.

۴. جایگزین جمعیت جدید با جمعیت فعلی .

۵. بازگشت به مرحله ۲ .

---

[1] Mutation Rate [2] Mutation Probability

۲-۲-۳-۱ شرط پایان اجرای الگوریتم ژنتیک

برای اینکه تشخیص دهیم چه موقع از اجرا الگوریتم متوقف شود، از شیوه‌های مختلفی می‌توان استفاده کرد. به‌عنوان نمونه می‌توان همگرا شدن کل جمعیت را در نظر گرفت و یا اینکه فاصله ارزیابی (برازندگی) بهترین فرد جمعیت از متوسط ارزیابی‌ها را در نظر گرفت که در این حالت باید از حد مشخصی کوچکتر باشد، یا مقدار تابع هدف از حد مشخصی بیشتر باشد یا می‌توان تعداد مشخصی از نسل‌ها را به‌عنوان شرط خاتمه در نظر گرفت. برخی از شروط خاتمه به شرح زیر است.

۱. به دست آوردن جواب نهایی مورد نظر بعد از چند تکرار و قابل قبول بودن جواب به ازای خطای خاص.

۲. اگر با پیشروی الگوریتم هیچ نوع بهبودی مشاهده نشود؛ خواه الگوریتم جواب دلخواه را پیدا کرده باشد و یا اینکه در کمینه محلی متوقف شده باشد.

۳. اگر مقدار میانگین تابع هدف به ازای تعدادی تکرار به مقدار خاصی رسیده باشد.

۴. الگوریتم به تعداد ثابتی از نسل‌ها رسیده باشد.

۵. بیشترین درجه برازش فرزندان حاصل شود یا دیگر نتایج بهتری حاصل نشود.

۶. ترکیبی از موارد بالا.

۲-۳ پیشینه حل مساله برش موجودی با استفاده از روش ژنتیک

در کشورها و شهرها، مقادیر زیادی از زباله ساخت و ساز در هر سال تولید می‌شود. زباله ساخت و ساز بخش قابل توجهی از جریان زباله‌های شهری به‌شمار می‌آید. برش موجودی‌های یک بعدی یکی از عوامل عمده زباله‌های ساخت و ساز است. برش موجودی‌های یک بعدی بمنظور کسب طول های مورد نیاز پروژه، منجر به تولید ضایعات می‌شود که از علل اصلی زباله های موجودی یک بعدی هستند. گرچه بخشی از این زباله‌ها مانند زباله فولاد قابل بازیافت هستند، کاهش تولید زباله‌ها می‌تواند استعمال مواد موجودی را افزایش دهد و در نتیجه پتانسیل سود شرکت افزایش یابد. تکنیک‌های بهینه‌سازی سنتی (مثل برنامه‌ریزی خطی و برنامه‌ریزی عدد صحیح) برای حل مساله برش موجودی یک بعدی دارای اشکالاتی هستند. شاهین [۱] و سالم [۲] [؟] مدل الگوریتم ژنتیک را برای حل مساله برش موجودی یک بعدی ارائه دادند. سه مورد مطالعاتی واقعی یک کارگاه فولاد محلی در فارگو، داکوتای شمالی، مورد بررسی قرار گرفته است.برای این سه مورد مطالعاتی، راه‌حل‌هایی (برنامه‌های برش) با استفاده از الگوریتم ژنتیک و با برنامه‌های برش فعلی کارگاه مقایسه شده است. نتایج مقایسه پتانسیل بالای ذخیره‌سازی مواد موجودی که با استفاده از راه‌حل‌های پیشنهادی می‌تواند بدست آید، نشان می‌دهد [؟].

جراشیا [۳] و همکاران [؟] روشی برای حل مساله برش تیرهای ساختمانی حاصل در یک شرکت محلی فلزکاری ارائه دادند. مساله مربوطه به کلاس مسائل برش موجودی یک بعدی با منابع با سایزهای متعدد تعلق دارد. در روش ارائه شده تولید

[۱] Shahin [۲] Salem [۳] Gracia

بیش از حد و کمتر از حد تیرها اداره می‌شود و شامل استفاده مجدد از بقایا در فرآیند بهینه‌سازی است.در کنار الگوریتم‌های ژنتیک، روش شامل دیگر الگوریتم‌هایی جدید پالایش که براساس استراتژی‌های مختلف جستجو و خوشه بندی‌های هستند، نیز است. علاوه براین، رمزگذاری جدید، با تعداد متغییر ژن برای الگوهای برش به‌منظور ایجاد امکان استفاده از عملگرهای ژنتیکی توسعه داده شده است. روش بصورت تجربی روی مجموعه‌ای از نمونه‌های مشابه با شرکت فلزکاری محلی تست شده است. بطور خاص، نتایج مقایسه نشان می‌دهد که روش ارائه شده بطور قابل ملاحظه‌ای عملکرد روش‌های اکتشافی گذشته را بهبود داده است [؟].

توماس [۱] و چادهاری [۲] [؟] روش جدیدی مبتنی بر ژنتیک به‌منظور حل مساله برش موجودی یک بعدی ارائه دادند. روش آنان شامل تکنیک‌های تولید ستون برای ایجاد ثبات و سرعت بخشیدن به فرآیند حل مساله است. این سرعت بخشیدن از طریق تحمیل تابع جریمه بر مقادیر برازندگی برای تکامل جمعیت بهتر، بدست می‌آید. توانایی الگوریتم ژنتیک از طریق رفتار پویا در برش و عملگرهای جهش افزایش یافته است. پویایی کمک می‌کند نرخ همگرایی راه‌حل برای گسترش وسیع و کنترل‌های رفتار تصادفی به سطح قابل قبولی بهبود یابد. روش ارائه شده سبب کاهش ۶۰ درصدی تولید ضایعات شده است. مقایسه روش ارائه شده با رویکردهای مشابه موجود مبتنی بر برنامه‌ریزی خطی ترکیبی و دیگر روش‌های موجود و روش‌های فراابتکاری اخیر، شدنی بودن و اعتبار الگوریتم ارائه شده را ثابت می‌کند. هم‌چنین در این پژوهش کارایی و کاربرد روش پیشنهادی در مسائل صنعتی ثابت شده است [؟].

برانداوو [۳] [؟] در حل مساله برش موجودی یک بعدی هزینه راه اندازی را در الگوریتم ژنتیک تک‌هدفه [۴] مورد مطالعه و بررسی قرار دادند. الگوریتم ژنتیک تک‌هدفه از دو نوع جمعیت تشکیل شده است: راه‌حل ها و الگوها. برای نمونه، توانایی ساخت الگوهای برش صرفنظر از راه‌حل‌ها را داراست. جمعیت الگوها که بطور تصادفی تولید می‌شوند ایستا هستند و دراین شرایط هیچ گونه تکاملی ندارند. نتایج بدست آمده از الگوریتم ژنتیک تک‌هدفه با دیگر روش‌های موجود مقایسه و قابلیت آن در یافتن راه‌حل‌های عملی و دارای مزیت رقابتی تایید شده است. نتایج محاسباتی نشان می‌دهد که روش جدید امیدبخش و پیشرو در مسائل برش موجودی یک بعدی است [؟].

جوانشیر [۵] [؟] و همکاران، در مسئله برش تک بعدی بر مبنای اقلام معیاری را به‌منظور سنجش پراکندگی ضایعات برش مورد استفاده قرار دادند. در این مقاله مقدار پراکندگی ضایعات که یک پارامتر کیفی است، به یک پارامتر کمی تبدیل شده است، سپس با استفاده از الگوریتم ژنتیک و ارائه الگوریتمی در این خصوص، مسئله پراکندگی ضایعات برش تک بعدی، در ابعاد بزرگ، مورد بررسی و حل قرار گرفته است. نتایج محاسباتی روش پیشنهادی بر روی مسائل نمونه، کارایی بالای الگوریتم را در رده‌های مختلف مسائل نشان می‌دهد [؟].

---

[۱] Thomas [۲] Chaudhari [۳] Brandao [۴] singleGA [۵] Javanshir

فصل ۳

روش تولید ستون

۳-۱ مقدمه

برای حل مسائلی با مقیاس بزرگ، ابتدا بایستی این مسائل را با روشی مناسب به یک یا چند مسئله کوچکتر با اندازه قابل کنترل و هماهنگ تبدیل کرد. روش‌هایی نظیر تجزیه ولف-دانتزیک [۲] ، تجزیه بندرز [۳] و آزادسازی که در مسائل برنامه‌ریزی خطی و به عنوان کاربرد اصل تجزیه مورد استفاده قرار می‌گیرند، برای همین منظور توسعه داده شده‌اند. کاربرد اصل تجزیه روشی نظام‌مند، برای حل برنامه‌های خطی بزرگ مقیاس و یا برنامه‌های خطی‌ای است که محدودیت‌هایی آسان دارند. به طور کلی محدودیت‌ها به دو مجموعه محدودیت‌های سخت (محدودیت‌های پیچیده) و محدودیت‌های آسان ،افراز می‌شوند. درصورت وجود محدودیت‌های آسان، کارایی روش تجزیه ارتقا می‌یابد.

استراتژی روش تجزیه روی دو برنامه خطی جدا پیاده‌سازی می‌شود:

- مسئله برنامه‌ریزی خطی با مجموعه محدودیت‌های سخت
- مسئله برنامه‌ریزی خطی با مجموعه محدودیت‌های آسان

با تبادل اطلاعات بین دو برنامه خطی، یک جواب بهینه برای مسئله اصلی حاصل می‌شود. برنامه خطی روی محدودیت های سخت را مسئله ارشد [۴] و برنامه خطی روی محدودیت‌های آسان را مسئله فرعی [۵] می‌نامیم. مسئله ارشد با انتقال مجموعه ضرائب هزینه‌ی اصلاح شده به مسئله فرعی، ستون (یا ستون های) جدیدی بر اساس این ضرائب هزینه، از مسئله فرعی دریافت می کند. این فرایند تا رسیدن به جواب بهینه بطور مداوم تکرار می‌شود. به این دلیل، چنین روشی به روش تولید ستون نیز مشهور است.

---

[۲] Dantzig-Wolf Decomposition [۳] Benders Decomposition [۴] master problem [۵] sub problem

۲-۳ مبنای الگوریتم تجزیه

تولید ستون [۶] ، روشی کلاسیک برای حل مسائل برنامه‌ریزی خطی از طریق اضافه کردن مکرر متغیرهای لازم به مساله اولیه است. به طور معمول تنها بخش کوچکی از متغیرها برای بهبود بهینگی مورد نیاز است. مبانی اگوریتم تجزیه یا به عبارت دقیق‌تر، الگوریتم تجزیه‌ی دانتزیك ـ ولف غالباً توانایی حل مسائل بسیار بزرگ را دارد. هرگاه ماتریس ضرایب یک مسأله، دارای ساختاری خاص به صورت مسائل چند بخشی باشد، کارایی این الگوریتم بسیار بهبود می‌یابد. این نوع مسائل دارای بخش‌های (بلوك‌های) مستقلی هستند که با تعدادی محدودیت به هم پیوند داده می‌شوند. الگوریتم‌های تجزیه بر مبنای این چهار مفهوم بنیان شده‌اند:

۱. الگوریتم سیمپلکس(سیمپلکس تجدید نظر شده)

۲. نمایش مجموعه جواب‌های شدنی و جهت‌های گوشه‌ای

۳. روش کاهش محدودیت‌ها

۴. روش تولید ستون

که در ادامه به توضیح آن‌ها خواهیم پرداخت.

۳-۲-۱ روش سیمپلکس

روش سیمپلکس یک فن کلی برای حل مسائل برنامه‌ریزی خطی است که در آن ابتدا مدل را وارد جدول سیمپلکس کرده و سپس یک سری اعمال سطری مقدماتی بر روی جدول مذکور اجرا می‌گردد. در روش سیمپلکس همواره از یک گوشه به گوشه‌ای بهتر حرکت کرده تا بهترین گوشه پیدا شود. برای این منظور باید مسأله را به فرم استاندارد تبدیل نماییم.

۳-۲-۱-۱ فرم استاندارد

اولین قدم در حل یک مدل برنامه‌ریزی خطی در روش سیمپلکس تبدیل مدل به فرم استاندارد است. ویژگی‌های فرم استاندارد:

۱. تابع هدف حداکثر سازی

در صورتی که تابع هدف به صورت min تعریف شده باشد بایستی به فرم max تبدیل شود. برای این منظور می توان تابع را در یک منفی ضرب نمود.

۲. محدودیت هابه فرم مساوی

---

[۶] Column Generation

در صورتی که محدودیت نامساوی وجود داشته باشد با استفاده از متغیر های کمکی بایستی محدودیت را به فرم مساوی تبدیل نمود.

متغیر های کمکی

- نامعادلات محدودیت ها با استفاده از متغیر های کمکی به معادله با علامت مساوی تبدیل می‌شوند.
- این متغیرهای کمکی را در معادلات با علامت کوچکتر مساوی «متغیر کمبود» و برای نامعادلات باعلامت بزرگتر مساوی «متغیر مازاد» می‌نامند.

متغیر های کمبود

- متغیرهای کمبود بیانگر منابع مصرف نشده یا موجودی منابع هستند.

  متغیر های کمبود هیچ سهمی در ایجاد سود ندارند.

  متغیرهای کمبود همواره نامنفی هستند.

مثال ۳-۲-۱. مسئله‌ای با نامساوی های داده شده به عنوان محدودیت را به صورت زیر فرض میکنیم:

$$\begin{aligned} \max Z =& ۴۰x_۱ + ۵۰x_۲ \\ s.t. \quad & x_۱ + ۲x_۲ \leq ۴۰ \\ & ۴x_۱ + ۳x_۲ \leq ۲۰ \\ & x_۱, x_۲ \geq ۰ \end{aligned} \tag{۳-۱}$$

می‌خواهیم نامعادلات موجود در محدودیت‌های مسئله ؟؟ را با افزودن متغیرهای کمکی به‌صورت مساوی تبدیل کنیم که نتیجه آن در ؟؟ مشاهده می‌شود.

$$\begin{aligned} x_۱ + ۲x_۲ + s_۱ = ۴۰ \\ ۴x_۱ + ۳x_۲ + s_۲ = ۱۲۰ \\ x_۱, x_۲, s_۱, s_۲ \geq ۰ \end{aligned} \tag{۳-۲}$$

۳-۲-۱-۲ جدول سیمپلکس

در روش سیمپلکس پس از استاندارد سازی، اطلاعات مربوط به مدل وارد می شوند که به آنها جدول‌های سیمپلکس می‌گوییم. در این روش هر جدول معادل یک گوشه موجه است. پس از اینکه جدول نظیر یک گوشه ساخته شد، سطر صفر آن جدول بررسی می شود. در این صورت دو حالت پیش می‌آید:

الف) مقادیر سطر صفر همگی غیر منفی هستند (بزرگتر یا مساوی صفر).

در این صورت گوشه نظیر این جدول گوشه بهینه است و براساس اطلاعات این جدول، مختصات گوشه بهینه و مقادیر آن استخراج شده و محاسبات به اتمام می‌رسد.

ب) در سطر صفر مقادیر منفی وجود دارد.

در این صورت گوشه مورد بررسی بهینه نیست و باید به گوشه مجاور بعدی برویم (جدول بعدی: در اینجا منظور از گوشه مجاور بعدی جدولی است که با جدول فعلی تنها در یک متغیر اساسی تفاوت داشته باشد) پس از رفتن به گوشه بعدی مجددا مقادیر سطر صفر را بررسی می‌کنیم. اگر حالت الف پیش آمد، به جواب بهینه رسیده‌ایم و در غیر این صورت، به گوشه بعدی می‌رویم و آنقدر این عمل را تکرار می‌کنیم تا حالت الف پیش آید و به جواب بهینه برسیم.

نکته ۱) برای جابجایی از هر جدول به جدول بعدی باید مشخص نماییم کدام متغیر باید به پایه وارد شود و کدام متغیر باید از پایه خارج شود. به این عمل محورگیری و یا تعیین متغیرهای ورودی و خروجی گفته می‌شود.

نکته۲) اهمیت روش سیمپلکس در این است که اولا برای مسائل بیش از دو بعد که تجسم هندسی آنها مشکل است کاربرد دارد و ثانیا نیازی به بررسی همه گوشه‌های مدل نیست. با شروع از جدول مبدا، از هرگوشه به گوشه بعدی وضعیت تابع هدف بهتر می‌شود، تا جاییکه به جواب بهینه برسیم (مقادیر سطر صفر همگی غیر منفی بشوند).

الگوریتم ۳-۱ خلاصه روش سیمپلکس اصلاح شده

۱: توجه کنید B^{-1} از چه ستون‌هایی باید خوانده شود. برای شروع $B^{-1} = I$

۲: برای جدول فعلی $c_B B^{-1}$ را محاسبه کنیم

۳: کلیه متغیرهای غیرپایه‌ای جدول فعلی را قیمت‌گذاری کنیم.اگر قیمت کلیه متغیرهای غیرپایه‌ای نامنفی است، پایه فعلی بهینه است. اگر پایه فعلی بهینه نیست، یک متغیر غیر پایه‌ای را که دارای منفی‌ترین ضریب در سطر صفر است را وارد پایه می‌کنیم و آن را x_k می‌نامیم.

۴: برای تعیین اینکه x_k متغیر پایه‌ای چه سطری خواهد بود، ستون نظیر x_k را در جدول فعلی محاسبه می‌کنیم $(B^{-1}a_k)$ و همینطور سمت راست جدول فعلی را محاسبه می‌کنیم $(B^{-1}b)$ اکنون می‌توانیم با استفاده از آزمون نسبت مشخص کنیم که x_k متغیر پایه‌ای کدام سطر خواهد بود. بنابراین در این مرحله مجموعه متغیرهای پایه‌ای جدول جدید را می‌شناسیم.

۵: با استفاده از ستون x_k و اعمال سطری مقدماتی B^{-1} جدید را به دست می‌آوریم و سپس به مرحله ۱ برمی‌گردیم.

۳-۲-۲ سیمپلکس اصلاح شده

در روش سیمپلکس میتوانیم از روی جدول اولیه،با عملیات پاشنه گردی، جدول بهینه را به‌دست آورد و مجموعه متغیرهای پایه را مشخص نمود. (اگر مسئله دارای m معادله و n متغیر باشد، به‌طور خاص تعداد $n - m$ متغیری که در سطر هدف دارای مقدار صفر هستند را متغیرهای غیر اساسی یا غیرپایه و m متغیر دیگر که دارای جواب‌های منحصر به‌فرد باشند را

متغیرهای پایه گویند). البته این نکته را باید مورد توجه قرار داد که اگر تعداد متغیر های پایه خیلی کمتر از متغیر های غیر پایه‌ای باشد عمل پاشنه گردی برای رسیدن به حل بهینه فقط در تعداد اندکی از ستون‌های ماتریس ضرایب A انجام می‌پذیرد. ازآنجایی‌که سایر ستون ها به وضوح به کارگرفته نخواهند شد واضح است که روی تعداد زیادی از عناصر، عملیات بیهوده انجام گرفته است. روش سیمپلکس اصلاح شده عبارت از طراحی چارچوبی است که در آن عملیات لازم روش سیمپلکس را طوری به نظم درمی‌آورد که از عملیات اضافی پرهیز شود [؟]. برای اینکه نشان دهیم چگونه میتوان جدول نظیر مجموعه ای دلخواه از متغیر های پایه B [۱] را به‌دست آورد، ابتدا به توصیف نماد گذاری مرتبط با جدول اولیه در سیمپلکس می‌پردازیم و سپس با یک مثال روش سیمپلکس اصلاح شده را شرح می‌دهیم.

B مجموعه ای دلخواه از متغیرهای پایه‌ای

b بردار سمت راست متناظر قیود اولین جدول

a_j ستون متناظر با x_j در قیود مسئله اصلی

B یک ماتریس $m \times m$ که j امین ستون آن، متناظر B_j در قیود مسئله اصلی است.

c_j ضریب x_j در تابع هدف

c_B بردار سطری $m \times ۱$ که j امین درایه آن ضریب تابع هدف متناظر با متغیر پایه ای B_j است.

u_i بردار ستونی $۱ \times m$ که iامین درایه آن، یک و بقیه درایه های آن صفر است.

در جدول سیمپلکس فرمول بندی های زیر را داریم:

۱. $B^{-۱}a_j$ = ستون نظیر X_j در جدول B
۲. $c_B B^{-۱}a_j - c_j$ = ضریب x_j در سطر صفر= $z_j - c_j$
۳. $B^{-۱}b$ = سمت راست قیود در جدول B
۴. $c_B B^{-۱}u_i$ = ضریب متغیر کمبود در s_i در سطر صفر جدول B
۵. $c_B B^{-۱}(-u_i)$ ضریب متغیر مازاد e_i در سطر صفر جدول B
۶. $M + c_B B^{-۱}u_i$ =ضریب متغیر مصنوعی a_i در سطر صفر جدول B (در مسائل ماکزیمم سازی)
۷. $c_B B^{-۱}b$ سمت راست سطر صفر جدول B

اگر $B, \ B^{-۱}$ و جدول اولیه را در اختیار داشته باشیم ، قادریم به کمک روابط ؟؟ تا ؟؟ هر قسمت از جدول سیمپلکس را برای هر مجموعه از متغیرهای پایه ای B محاسبه کنیم. برای توضیح بیشتر به حل یک مسئله می‌پردازیم. [؟]

مثال ۳-۲-۲. مسئله زیر را در نظر می‌گیریم:

[۱] Basic Value

$$\begin{aligned} \max z = & 60x_1 + 30x_2 + 20x_3 \\ s.t. \quad & 8x_1 + 6x_2 + x_3 \le 48 \\ & 4x_1 + 2x_2 + 1/5x_3 \le 20 \\ & 2x_1 + 1/5x_2 + 0/5x_3 \le 8 \end{aligned} \tag{۳-۳}$$

بعد از افزودن متغیرهای کمکی s_1,s_2,s_3 جدول اولیه (جدول صفر) عبارت است از :

$$\begin{aligned} \max z = & 60x_1 + 30x_2 + 20x_3 \\ s.t. \quad & 8x_1 + 6x_2 + x_3 + s_1 = 48 \\ & 4x_1 + 2x_2 + 1/5x_3 + s_2 = 20 \\ & 2x_1 + 1/5x_2 + 0/5x_3 + s_3 = 8 \end{aligned} \tag{۴-۳}$$

بدون توجه به اینکه چه تعداد عملیات انجام شده است، B^{-1} نظیر جدول ؟؟ ماتریسی است 3×3 که j امین ستون آن ستون نظیر s_j در این جدول است. بنابراین در جدول اولیه $B(0)$ مجموعه متغیرهای پایه‌ای، عبارتند از :

$NB(0) = \{x_1x_2x_3\}$ $B(0) = \{s_1s_2s_3\}$

فرض کنیم B_i ستون‌هایی از مسئله LP اولیه باشد که متناظر متغیرهای پایه‌ای جدول است آنگاه

$$\mathbf{B}_0^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{۵-۳}$$

اکنون می‌توان متغیرهای غیرپایه‌ای وارد شونده به پایه را با محاسبه ضرایب متغیرهای غیر پایه‌ای در سطر صفر فعلی تعیین کرد.این فرآیند اغلب به عنوان قیمت گذاری متغیرهای غیرپایه‌ای شناخته می‌شود. روابط ؟؟ تا ؟؟ که قیمت متغیرهای غیرپایه‌ای تا وقتی که $c_BB_0^{-1}$ محاسبه گردد امکان پذیر نیست. از آنجایی که

$$\mathbf{c}_B = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \tag{۶-۳}$$

داریم

$$\mathbf{c}_B B_{\circ}^{-1} = \begin{bmatrix} \circ & \circ & \circ \end{bmatrix} \begin{bmatrix} ۱ & \circ & \circ \\ \circ & ۱ & \circ \\ \circ & \circ & ۱ \end{bmatrix} = \begin{bmatrix} \circ & \circ & \circ \end{bmatrix} \tag{۷-۳}$$

اکنون با استفاده از ؟؟ متغیرهای پایه‌ای را قیمت گذاری می‌کنیم.

$$\bar{c}_۱ = \begin{bmatrix} \circ & \circ & \circ \end{bmatrix} \begin{bmatrix} ۸ \\ ۴ \\ ۲ \end{bmatrix} - ۶\circ = -۶\circ$$

$$\bar{c}_۲ = \begin{bmatrix} \circ & \circ & \circ \end{bmatrix} \begin{bmatrix} ۶ \\ ۲ \\ ۱/۵ \end{bmatrix} - ۳\circ = -۳\circ$$

$$\bar{c}_۳ = \begin{bmatrix} \circ & \circ & \circ \end{bmatrix} \begin{bmatrix} ۱ \\ ۱/۵ \\ \circ/۵ \end{bmatrix} - ۲\circ = -۲\circ$$

از آنجایی که $x_۱$ منفی‌ترین ضریب را در سطر صفر داراست، لذا $x_۱$ وارد پایه می‌شود. در ادامه تنهاچیزی که برای جدول جدید نیاز داریم این است که مجموعه جدید از متغیرهای پایه‌ای، $B(۱)$ و $B_۱^{-۱}$ را بشناسیم.برای تعیین $B(۱)$ باید بدانیم که $x_۱$ متغیر پایه‌ای چه سطری خواهد شد. اکنون ستون نظیر $x_۱$ و همچنین سمت راست جدول فعلی را محاسبه می کنیم. از ؟؟ داریم :

$$\text{ستون نظیر } x_۱ \text{ در جدول فعلی} = \begin{bmatrix} ۱ & \circ & \circ \\ \circ & ۱ & \circ \\ \circ & \circ & ۱ \end{bmatrix} \begin{bmatrix} ۸ \\ ۴ \\ ۲ \end{bmatrix} = \begin{bmatrix} ۸ \\ ۴ \\ ۲ \end{bmatrix}$$

و از (۳) داریم

$$\text{سمت راست جدول فعلی} = \begin{bmatrix} ۱ & \circ & \circ \\ \circ & ۱ & \circ \\ \circ & \circ & ۱ \end{bmatrix} \begin{bmatrix} ۴۸ \\ ۲\circ \\ ۸ \end{bmatrix} = \begin{bmatrix} ۴۸ \\ ۲\circ \\ ۸ \end{bmatrix}$$

اکنون با استفاده از آزمون نسبت تعیین می‌کنیم که $x_۱$ متغیر پایه ای کدام سطر خواهد بود. نسبت برای سطر یک $۶ = \frac{۴۸}{۸}$ ،برای سطر دو $۵ = \frac{۲۰}{۴}$ و برای سطر سوم $۴ = \frac{۸}{۲}$ است. بنابراین $x_۱$ متغیرپایه‌ای سطر سوم خواهد بود. به این ترتیب در جدول جدید(جدول یک) داریم : $B(۱) = \{x_۱,s_۲,s_۳\}$ و $NB(۱) = \{s_۳,x_۲,x_۳\}$ $B^{-۱}$ جدید، ستونهای نظیر $S_۱,S_۲,S_۳$ درجدول جدید خواهد بود. برای تعیین $B^{-۱}$ ، به ستون نظیر متغیر وارد شونده $x_۱$ در جدول صفر توجه کنید. با توجه به این ستون مشاهده می کنیم که برای تبدیل جدول صفر به جدول یک باید اعمال سطری و مقدماتی زیر را اجرا گردد.

- ضرب سطر صفر جدول صفر در $\frac{۱}{۲}$
- تعویض سطر ۱ جدول صفر با ((۴-) برابر سطر ۳ جدول صفر + سطر یک جدول صفر)
- تعویض سطر ۲ جدول صفر با ((۲-) برابر سطر ۳ جدول صفر + سطر دو جدول صفر) با بکارکیری اعمال مذکور روی $B_۰^{-۱}$ داریم :

$$B_۱^{-۱} = \begin{bmatrix} ۱ & ۰ & -۴ \\ ۰ & ۱ & -۲ \\ ۰ & ۰ & ۰/۵ \end{bmatrix}$$

$$c_B B^{-۱} = \begin{bmatrix} ۰ & ۰ & ۶۰ \end{bmatrix} \begin{bmatrix} ۱ & ۰ & -۴ \\ ۰ & ۱ & -۲ \\ ۰ & ۰ & ۰/۵ \end{bmatrix} = \begin{bmatrix} ۰ & ۰ & ۳۰ \end{bmatrix}$$

اکنون با استفاده از ؟؟ و ؟؟ ، متغیرهای غیرپایه‌ای متغیر غیر پایه‌ای جدول یک را قیمت گذاری می‌کنیم :

$$\bar{c}_۲ = \begin{bmatrix} ۰ & ۰ & ۳۰ \end{bmatrix} \begin{bmatrix} ۶ \\ ۲ \\ ۱/۵ \end{bmatrix} - ۳۰ = ۱۵$$

$$\bar{c}_۳ = \begin{bmatrix} ۰ & ۰ & ۳۰ \end{bmatrix} \begin{bmatrix} ۱ \\ ۱/۵ \\ ۰/۵ \end{bmatrix} - ۲۰ = -۵$$

$$\text{ضریب } s_۳ \text{ در سطر صفر} = \begin{bmatrix} ۰ & ۰ & ۳۰ \end{bmatrix} \begin{bmatrix} ۰ \\ ۰ \\ ۱ \end{bmatrix} - ۰ = ۳۰$$

از آنجا که $x_۳$ تنها متغیر دارای ضریب منفی در سطر صفر است، پس $x_۳$ وارد پایه می‌شود. برای اینکه مجموعه جدیدی از

متغیرهای پایه‌ای، $B(۲)$ و $B_۲^{-۱}$ نظیر آنها را تعیین کنیم، لازم است ببینیم $x_۳$ متغیر پایه‌ای په سطری خواهد شد. داریم:

$$\text{ستون نظیر } x_۳ \text{ در جدول یک} = B_۱^{-۱}a_۳ = \begin{bmatrix} ۱ & ۰ & -۴ \\ ۰ & ۱ & -۲ \\ ۰ & ۰ & ۰{,}۵ \end{bmatrix} \begin{bmatrix} ۱ \\ ۱/۵ \\ ۰{,}۵ \end{bmatrix} = \begin{bmatrix} -۱ \\ ۰/۵ \\ ۰/۲۵ \end{bmatrix}$$

$$\text{سمت راست جدول یک} = B_۱^{-۱}b = \begin{bmatrix} ۱ & ۰ & -۴ \\ ۰ & ۱ & -۲ \\ ۰ & ۰ & ۰/۵ \end{bmatrix} \begin{bmatrix} ۴۸ \\ ۲۰ \\ ۸ \end{bmatrix} = \begin{bmatrix} ۱۶ \\ ۴ \\ ۴ \end{bmatrix}$$

برای تعیین سطری که $X_۳$ متغیر پایه‌ای آن خواهد شد، از آزمون نسبت استفاده می کنیم که در نتیجه آن نسبت برای سطر یک $\frac{۱۶}{-۱} = -۱۶$ ،برای سطر دو $\frac{۴}{۰/۵} = ۸$ و برای سطر سوم $\frac{۴}{۰/۲۵} = ۱۶$ است. بنابراین $x_۳$ متغیرپایه‌ای سطر دوم خواهد بود. در محاسبه $B_۲^{-۱}$ برای تبدیل $x_۳$ به یک متغیر پایه ای باید یک سری اعمال سطری مقدماتی را روی جدول یک اجرا نمود که در نتیجه آن داریم :

$$B_۲^{-۱} = \begin{bmatrix} ۱ & ۲ & -۸ \\ ۰ & ۲ & -۴ \\ ۰ & -۰/۵ & ۱/۵ \end{bmatrix}$$

با قیمت گذاری متغیرهای غیرپایه‌ای جدول دوم را قیمت گذاری می‌کنیم. داریم :

$$c_BB_۲^{-۱} = \begin{bmatrix} ۰ & ۲۰ & ۶۰ \end{bmatrix} \begin{bmatrix} ۱ & ۲ & -۸ \\ ۰ & ۲ & -۴ \\ ۰ & -۰/۵ & ۱/۵ \end{bmatrix} = \begin{bmatrix} ۰ & ۱۰ & ۱۰ \end{bmatrix}$$

$$\bar{c}_۲ = \begin{bmatrix} ۰ & ۱۰ & ۱۰ \end{bmatrix} \begin{bmatrix} ۶ \\ ۲ \\ ۱/۵ \end{bmatrix} - ۳۰ = ۵$$

$$\text{ضریب } s_۲ \text{ در سطر صفر} = \begin{bmatrix} ۰ & ۱۰ & ۱۰ \end{bmatrix} \begin{bmatrix} ۰ \\ ۱ \\ ۰ \end{bmatrix} - ۰ = ۱۰$$

$$\text{ضریب } s_3 \text{ در سطر صفر} = \begin{bmatrix} 0 & 10 & 10 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} - 0 = 10$$

$$\text{سمت راست جدول ۲} = B_2^{-1}b = \begin{bmatrix} 1 & 2 & -8 \\ 0 & 2 & -4 \\ 0 & -0/5 & 1/5 \end{bmatrix} \begin{bmatrix} 48 \\ 20 \\ 8 \end{bmatrix} = \begin{bmatrix} 24 \\ 8 \\ 2 \end{bmatrix}$$

ازآنجایی که $B(2) = \{s_1, x_3, x_1\}$ پاسخ بهینه این مسئله عبارت است از:

$s_1 = 24, x_3 = 8, x_1 = 2, x_2 = s_2 = s_3 = 0$ بنابراین مقدار بهینه z از رابطه ؟؟ محاسبه می‌گردد:

$$c_B B_2^{-1} b = \begin{bmatrix} 0 & 10 & 10 \end{bmatrix} \begin{bmatrix} 48 \\ 20 \\ 8 \end{bmatrix} = 280$$

۳-۲-۳ شکل ضربی وارون ماتریس پایه

قسمت عمده‌ای از محاسبات روش سیمپلکس اصلاح شده در ارتباط با تغییرات B^{-1} از یک جدول به جدول بعدی است. در این بخش، روشی موثر برای محاسبه B^{-1} ارائه می دهیم[؟].

فرض کنید یک مسئله LP با m قید را حل می‌کنیم. می‌دانیم که x_k، باید متغیر پایه‌ای سطر r ام شود(x_k باید وارد پایه شود). اگر ستون نظیر x_k در جدول جدید بصورت زیر باشد:

$$\begin{bmatrix} a_{1k} \\ a_{2k} \\ \vdots \\ a_{mk} \end{bmatrix}$$

ماتریس $E_{m,m}$ را بصورت زیر تعریف می کنیم:

$$\mathbf{E} = \begin{bmatrix} ۱ & ۰ & \cdots & -\frac{\bar{a}_{۱k}}{\bar{a}_{rk}} & \cdots & ۰ & ۰ \\ ۰ & ۱ & \cdots & -\frac{\bar{a}_{۲k}}{\bar{a}_{rk}} & \cdots & ۰ & ۰ \\ \vdots & \vdots & & & & \vdots & \vdots \\ ۰ & ۰ & \cdots & \frac{۱}{\bar{a}_{rk}} & \cdots & ۰ & ۰ \\ \vdots & \vdots & & & & \vdots & \vdots \\ ۰ & ۰ & \cdots & -\frac{\bar{a}_{m-۱,k}}{\bar{a}_{rk}} & \cdots & ۱ & ۰ \\ ۰ & ۰ & \cdots & -\frac{\bar{a}_{mk}}{\bar{a}_{rk}} & \cdots & ۰ & ۱ \end{bmatrix}$$

به طور خلاصه ماتریس E همان ماتریس I_m است که ستون r آن با بردار ستونی زیر تعویض شده است:

$$\begin{bmatrix} -\frac{\bar{a}_{۱k}}{\bar{a}_{rk}} \\ -\frac{\bar{a}_{۲k}}{\bar{a}_{rk}} \\ \vdots \\ \frac{۱}{\bar{a}_{rk}} \\ \vdots \\ -\frac{\bar{a}_{m-۱,k}}{\bar{a}_{rk}} \\ -\frac{\bar{a}_{mk}}{\bar{a}_{rk}} \end{bmatrix}$$

تعریف: یک ماتریس (مانند E) را که فقط در یک ستون با ماتریس همانی تفاوت دارد، ماتریس مقدماتی نامند. اکنون نشان می‌دهیم که

$$B^{-۱}\ \text{جدول جدید} = E\,(B^{-۱}\ \text{جدول فعلی}) \tag{۳-۸}$$

برای اینکه درستی این رابطه را بررسی کنیم، به اعمال سطری مقدماتش که جدول فعلی را به جدول جدید تبدیل می‌کنند، توجه می‌کنیم. داریم:

$$\text{سطر } r \text{ ماتریس } B^{-۱}\ \text{جدید} = (\frac{۱}{\bar{a}_{rk}})\,(\text{سطر } r \text{ ماتریس } B^{-۱}\ \text{فعلی}) \tag{۳-۹}$$

و برای $i\neq r$ داریم:

$$\text{سطر } i \text{ ماتریس } B^{-1} \text{ جدید} = (\text{سطر } i \text{ ماتریس } B^{-1} \text{ فعلی}) - \frac{\bar{a}_{ik}}{\bar{a}_{rk}} (\text{سطر } r \text{ ماتریس } B^{-1} \text{ فعلی}) \qquad (۳-۱۰)$$

می‌دانیم که

$$\text{سطر } i \text{ ماتریس } ((B^{-1} \text{ فعلی}) \, E) = (\text{سطر } i \text{ ماتریس } B^{-1} \text{ فعلی})(\text{ماتریس } E) \qquad (۳-۱۱)$$

از ترکیب رابطه ؟؟ با تعریف E داریم:

$$\text{سطر } r \text{ ماتریس } ((B^{-1} \text{ فعلی}) \, E) = \left(\frac{1}{\bar{a}_{rk}}\right) (\text{سطر } r \text{ ماتریس } B^{-1} \text{ فعلی})$$

و برای $i\neq r$ داریم:

$$\text{سطر } i \text{ ماتریس } ((B^{-1} \text{ فعلی}) \, E) = (\text{سطر } i \text{ ماتریس } B^{-1} \text{ فعلی}) - \left(\frac{\bar{a}_{ik}}{\bar{a}_{rk}}\right) (\text{سطر } r \text{ ماتریس } B^{-1} \text{ فعلی})$$

بنابراین ؟؟ با روابط ؟؟ و ؟؟ سازگار است. لذا می‌توان با استفاده از ؟؟، ماتریس B^{-1} جدید را از روی ماتریس B^{-1} فعلی به دست آورد.

اگر جدول اولیه را جدول صفر و E_i را ماتریس مقدماتی نظیر iامین جدول سیمپلکس بنامیم، داریم:

$$B_1^{-1} = E_0 B_0^{-1} = E_0 \qquad (B_0^{-1} = I_m)$$

آنگاه

$$B_2^{-1} = E_1 B_1^{-1} = E_1 E_0$$

وبه طور کلی

$$B_k^{-1} = E_{k-1} E_{k-2} \ldots E_1 E_0$$

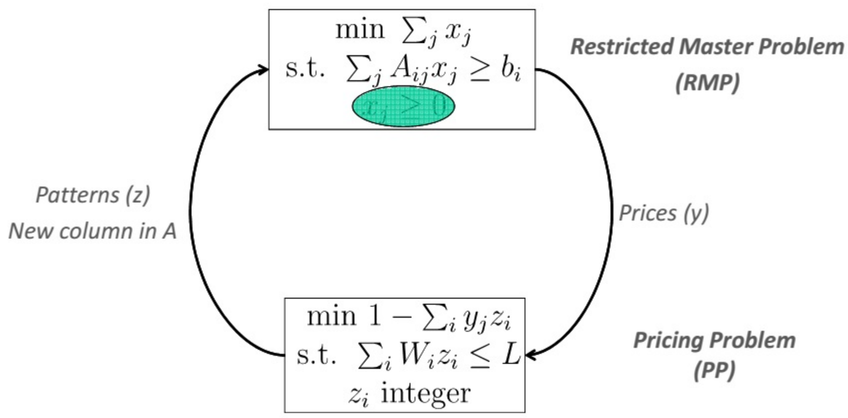
را شکل ضربی وارون نامند. اغلب نرم افزارهای حل مسئله برنامه ریزی خطی از روش سیمپلکس اصلاح شده همراه با محاسبه متوالی B^{-1} از طریق شکل ضربی وارون استفاده می‌کنند.

۳-۲-۴ روش تجزیه

اصل تجزیه روشی سیستماتیک برای حل برنامه های خطی بزرگ مقیاس یا برنامه های خطی دارای محدودیت های با ساختار ویژه است. محدودیت ها به دو دسته تقسیم می شوند: محدودیت های مشترک (محدودیت پیچیده) و محدودیت ها دارای

ساختار ویژه (محدودیت های مستقل) است. البته لازم نیست که هر برنامه خطی الزاماً دارای محدودیت‌هایی مستقل باشند؛ با این حال، محدودیت‌های مستقل، در صورت وجود، باعث افزایش بهره وری می شود.

استراتژی روش تجزیه این است که بر روی دو برنامه خطی جداگانه کار کند: یکی بر روی مجموعه ای از محدودیت های مشترک ودیگری بر روی مجموعه‌ای از محدودیت های مستقل. اطلاعات بین دو برنامه خطی تبادل می‌شود تا زمانیکه یک راه حل بهینه برای مسئله اصلی به‌دست آید. برنامه خطی بر روی محدودیت های مشترک مسئله اصلی نامیده می‌شود و برنامه خطی بر روی محدودیت های مستقل، زیرمسئله نامیده می‌شود. مسئله اصلی به طور منظم مجموعه ای از ضرائب هزینه (ارزش‌ها) را به زیرمسئله می‌دهد و از زیرمسئله یک ستون جدید براساس این ضرایب هزینه دریافت می کند. به همین دلیل به این روش، تولید ستون می‌گویند.



شکل ۳-۱: تبادل اطلاعات بین مسئله اصلی و زیر مسئله در روش تجزیه

در ابتدا فرض می‌کنیم محدودیت های مستقل کراندار هستند. در طول فرآیند تجزیه فرض می‌کنیم محدودیت‌ها تقلیل یافته‌اند و همچنین تا حد امکان مسئله اصلی را به چندین زیرمسئله گسترش و روابط بین آنها را نشان دهیم. برنامه خطی زیر را در نظر بگیرید، جایی که$\mathbb{X}$ مجموعه ای چندوجهی است که محدودیت های مستقل را نشان می‌دهد، A یک ماتریس $m \times n$ است، و bیک بردار است:

(۱۲-۳)

$$\begin{aligned} &\min \sum cx \\ &s.t. \quad Ax = b \\ &x \in \mathbb{X} \end{aligned}$$

قضیه ۳-۲-۳. قضیه نمایش

فرض کنیم که $\mathbb{X} = \{x : Ax \leq b, \quad x \geq ۰\}$ یک مجموعه چندوجهی ناتهی باشد، هنگامی‌که نقاط اکسترمم

این مجموعه ناتهی و محدود باشد به‌صورت $x_1, x_2, ...x_t$ نشان داده می‌شوند.بنابراین مجموعه جهت‌های گوشه‌ای تهی است اگر و فقط اگر $\mathbb{X}$ کراندار باشد؛ اگر $\mathbb{X}$ کراندار نباشد، دراین هنگام مجموعه جهت‌های گوشه‌ای ناتهی و دارای عناصر محدودی است که آن را به صورت $d_1, d_2, ...d_l$ نمایش می‌دهیم. باوجود این $\bar{X} \in \mathbb{X}$ اگر و فقط اگر بتوان آن را با یک ترکیب محدب از $x_1, x_2, ...x_t$ بعلاوه ترکیب خطی نامنفی از $d_1, d_2, ...d_l$ آن را نشان داد:

$$\bar{X} = \sum_{j=1}^{t} \lambda_j x_j + \sum_{j=1}^{l} \alpha_j d_j \tag{۱۳-۳}$$
$$\sum_{j=1}^{t} \lambda_j = 1$$
$$\lambda_j \geq 0 \quad , j = 1, 2, ...t$$
$$d_j \geq 0 \quad , j = 1, 2, ...l$$

فرض می‌کنیم $\mathbb{X}$ ناتهی و کراندار باشد؛ باتوجه به قضیه نمایش از آنجاییکه $\mathbb{X}$ یک مجموعه چند وجهی است، لذا هر کدام از $x \in \mathbb{X}$ را می توان برحسب ترکیب حدی از تعدادی از نقاط اکسترمم $\mathbb{X}$ به‌دست آورد. این رابطه را به صورت زیر می‌توان نمایش داد.

$$x = \sum_{j=1}^{t} \lambda_j x_j \tag{۱۴-۳}$$
$$\sum_{j=1}^{t} \lambda_j = 1$$
$$\lambda_j \geq 0 \quad , j = 1, 2, ...t$$

اکنون می‌توانیم x های به‌دست آمده از رابطه ؟؟ را در رابطه ؟؟ جایگزین کنیم:

$$\min Z = \sum_{j=1}^{t} (cx_j)\lambda_j \tag{۱۵-۳}$$
$$\sum_{j=1}^{t} (Ax_j)\lambda_j = b \tag{۱۶-۳}$$
$$\sum_{j=1}^{t} \lambda_j = 1 \tag{۱۷-۳}$$
$$\lambda_j \geq 0 \quad , j = 1, ..., t$$

فرض می‌کنیم که t تعداد تقاط اکسترمم‌های $\mathbb{X}$ باشد که معمولا هم بسیار بزرگ است فلذا تلاش برای به‌دست آوردن همه

این نقاط اکسترمم و پاسخ‌های مسئله کاری بسیار دشوار است.بنابراین باید با روش‌هایی مانند تجزیه (تولید ستون)که نیاز به محاسبه همه اکسترمم ها نداشته باشد این مشکل را حل نمود. می‌توان معادلات فوق را بصورت زیر بازنویسی کرد:

$$\max Z = c[\sum_{j=۱}^{t} \lambda_j x_j] \tag{۱۸-۳}$$

$$A[\sum_{j=۱}^{t} \lambda_j x_j] = b$$

$$\sum_{j=۱}^{t} \lambda_j = ۱$$

$$\lambda_j \geq ۰ \quad , j = ۱, ..., t$$

اکنون این مسأله، تبدیل به یک مسأله‌ی برنامه‌ریزی خطی با متغیرهای تصمیم λ_j شده است و x_j مقادیری ثابت از نقاط گوشه‌ای می‌باشد. برای سادگی عملیات، با تغییر متغیر $g_j = cx_j$ و $h_j = Ax_j$ مسأله اصلی به صورت زیر در می‌آید:

$$\max Z = \sum_{j=۱}^{t} g_j \lambda_j \tag{۱۹-۳}$$

$$\sum_{j=۱}^{t} h_j \lambda_j = b \tag{۲۰-۳}$$

$$\sum_{j=۱}^{t} \lambda_j = ۱ \tag{۲۱-۳}$$

$$\lambda_j \geq ۰ \quad , j = ۱, ..., t$$

ستون مورد نیاز، ستونی است که در معادله‌ی تابع هدف مسأله، در هر تکرار بزرگترین ضریب مثبت (منفی‌ترین مقدار در سطر صفر جداول سیمپلکس) را داشته باشد. به طور کلی ضرایب متغیرها در معادله‌ی تابع هدف (به جز ضرایب متغیرهای اساسی شروع مسأله) با عبارت $C_j - Z_j$ مشخص می‌شوند که مقدار آن برای متغیر تصمیم λ_j بااستفاده از رابطه $c_B B^{-۱} a_j - c_j$ به‌صورت زیر بدست می‌آید:

$$C_j - Z_j = g_j - Y \begin{bmatrix} h^j \\ ۱ \end{bmatrix} \tag{۲۲-۳}$$

که Y بردار قیمت‌های سایه (یا ضرایب سیمپلکس) و $\begin{bmatrix} h^j \\ ۱ \end{bmatrix}$ ضریب ستون مسأله‌ی تبدیل شده بر حسب λ_j است. بردار Y را می‌توان به دو قسمت تقسیم کرد: ضرایب سیمپلکس مربوط به معادلات ؟؟ و ضرایب سیمپلکس محدودیت ؟؟ که به

ترتیب با بردار $Y_\circ$ و $Y_۱$ نشان داده می‌شوند.

با استفاده از نمادهای تعریف شده، معادله ؟؟ به صورت زیر تغییر می‌یابد:

$$C_j - Z_j = cx_j - [Y_\circ Y_۱] \begin{bmatrix} Ax_j \\ ۱ \end{bmatrix} \qquad (۲۳\text{-}۳)$$
$$= cx_j - Y_\circ Ax_j - Y_۱$$

از آنجا که هدف، یافتن متغیری با بزرگترین ضریب در تابع هدف به منظور تعیین متغیر ورودی(λ_j) است، باید $C_j - Z_j$ بیشینه گردد. اما از آنجا که $Y_۱$ مقداری است ثابت، این مقدار موقتاً از معادله ؟؟ حذف می‌گردد. پس

$$\max Z' = cx_j - Y_\circ Ax_j, \quad j = ۱, ..., t \qquad (۲۴\text{-}۳)$$

اما با یادآوری این نکته که x_j نقاط گوشه‌ای مجموعه‌ی محدب $\mathbb{X}$ است که در آن $x \geq \circ$ و $Dx = b_۱$، این مجموعه محدودیت‌ها را باید با تابع هدف فوق در نظر گرفت

$$\max Z' = cx - Y_\circ Ax \qquad (۲۵\text{-}۳)$$
$$s.t. \quad Dx = b_۱ \quad x \geq \circ\,,$$

بنابر این به منظور پیدا کردن متغیر و ستون ورودی برای مسأله‌ی اصلی، مسأله‌ی فرعی ؟؟ حل خواهد شد.

۳-۲-۵ گام‌های الگوریتم تجزیه (روش تولید ستون)

گام۱. یافتن یك جواب موجه ابتدایی برای مسأله اصلی

گام۲. انتخاب متغیر ورودی

گام۳.حل مسأله فرعی

مسأله فرعی را به منظور تعیین مثبت‌ترین ضریب متغیر غیر اساسی ورودی، حل می‌کنیم. در مسأله فرعی برای بدست آوردن $Y_\circ$از رابطه‌ی $Y = (Y_\circ, Y_۱) = c_B B^{-۱}$ استفاده می‌نماییم.

بانوجه به دوگان متغیرهای دوگان مطابق معادلات ؟؟ و ؟؟ ضریب متغیر ورودی (λ_j) در تابع هدف مسأله از رابطه‌ی cx_j به‌دست می‌آید.

گام۴. بررسی شرط توقف

اگر مقدار $C_j - Z_j$ از دستور زیر، صفر باشد به جواب بهینه رسیده‌ایم. (مقدار متغیرهای اساسی جواب بهینه از

رابطه‌ی $\begin{bmatrix} b_0 \\ 1 \end{bmatrix} b = B^{-1}$ به‌دست می‌آید)، در غیر این صورت به گام بعد می‌رویم. $C_j - Z_j$ ضریب متغیر λ_i در هر تکرار و برابر با. $C_j - Z_j = Z' - Y_1$ است.

باتوجه به مطالب گفته شده اکنون می توانیم الگوریتمی برای حل مسئله برش موجودی یک بعدی با روش تولید ستون ارائه نماییم :

الگوریتم ۳-۲ خلاصه روش تولید ستون برای 1D-CSP

۱: مسئله را به فرم LP تعریف می‌کنیم
۲: متغیرهای سطر هدف را به دوقسمت پایه و غیر پایه تقسیم می کنیم
۳: با استفاده از ستون‌های نظیر متغیرهای پایه، ماتریس پایه در جدول فعلی (B) را تعیین نموده و معکوس آن را محاسبه می‌کنیم (B^{-1}).
۴: با استفاده از رابطه $c_B B^{-1}$ ضرایب متغیرهای پایه در سطر هدف را بدست می‌آوریم.
۵: برای تعیین وجود متغیر غیر پایه که قابلیت ورود به پایه را داشته باشد، متغیرهای غیرپایه‌ای را به کمک حل یک مسئله کوله‌پشتی $c_B B^{-1} a_j - c_j$ ارزش‌گذاری می‌نماییم. اگر پاسخ مسئله کوله پشتی کمتر یا مساوی یک شد جدول فعلی بهینه است و متغیرغیرپایه‌ای جدیدی برای ورود به پایه وجود نخواهد داشت.
۶: با ضرب جواب مسئله کوله پشتی در B^{-1} ستون تولید شده جدید را به‌دست می‌آوریم.
۷: سمت راست جدول فعلی را به‌دست می‌آوریم.
۸: با استفاده از آزمون نسبت سطری که متغیر غیرپایه وارد آن می‌شود به‌دست می‌آوریم.
۹: با استفاده از قائده شکل ضربی وارون و ستون جدید، B^{-1} جدول جدید را تولید می‌کنیم و به گام ۲ برمی‌گردیم.

۳-۲-۶ روش تولید ستون در حل مسئله صفحات برش

روش سیمپلکس اصلاح شده نسبت به الگوریتم سیمپلکس به محاسبات کمتری نیاز دارد. در این بخش، روش تولید ستون را که گیلمور و گومری ابداع کرده‌اند، بررسی می‌کنیم [؟]. در مسائل LP که تعداد زیادی متغیر وجود دارد، روش ایجاد ستون می‌تواند کارایی الگوریتم سیمپلکس اصلاح شده را افزایش دهد. علاوه بر این روش تولید ستون، یکی از مولفات مهم الگوریتم تجزیه دانتزیک–ولف نیز است. به‌منظور تشریح مفهوم تولید ستون، یک نمونه ساده از مسئله کلاسیک برش موجودی را حل می‌کنیم.

مثال ۳-۲-۴. شرکت وودکو [1] چوب به قطعات ۳،۵ و ۹ فوتی می‌فروشد. مشتریان شرکت، ۲۵ قطعه ۳ فوتی، ۲۰ قطعه ۵ فوتی، ۱۵ قطعه ۱ فوتی درخواست کرده‌اند. شرکت باید تقاضای مشتریان را از برش قطعات ۱۷ فوتی چوب تامین کند.

ابتدا یک مسئله LP می‌نویسیم که مقدار ضایعات را حداقل کند. سپس این مسئله را به روش تولید ستون حل می‌کنیم. ابتدا شرکت باید مشخص کند چگونه قطعات ۱۷ فوتی چوب را برش دهد. برای مثال اگر یک قطعه ۱۷ فوتی را به سه قطعه

[1] Woodco

۵ فوتی برش دهد، آنگاه $۲ = ۱۵ - ۱۷$ فوت ضایعات این تصمیم خواهد بود. بسیاری از الگوها می‌توانند کنار گذاشته شوند. مثلا عاقلانه نخواهد بود اگر از یک قطعه ۱۷ فوتی فقط دو قطعه ۹ و ۵ فوتی را با ۳ فوت ضایعات (۳=۵-۹-۱۷) در نظر بگیریم و بهتر است آن را به سه صورت ۹،۵و ۳ فوتی مورد بررسی قرار دهیم که ضایعاتی ندارد. به طور کلی هر برش را که ضایعات آن بزرگتر از سه فوت یا مساوی آن باشد، می توان بررسی کرد؛ زیرا می‌توان ضایعات را برای تولید قطعات کوچکتر به‌کار گرفت. جدول ؟؟ ، لیستی از روش‌های مناسب یا معقول برش را ارائه می‌دهد.

جدول ۳-۱: روش‌های مختلف برش قطعه‌ها.

طرح برش	تعداد قطعه ۳ فوتی	تعداد قطعه ۵ فوتی	تعداد قطعه ۹ فوتی	مقدار ضایعات (فوت)
۱	۵	۰	۰	۲
۲	۴	۱	۰	۰
۳	۲	۲	۰	۱
۴	۲	۰	۱	۲
۵	۱	۱	۱	۰
۶	۰	۳	۰	۲

x_i را به صورت زیر تعریف می‌کنیم:

$x_i =$ **تعداد قطعات ۱۷ فوتی چوب که با طرح i برش داده شوند**

از آنجاکه

$$۳۱۰ = ۹ \times ۱۵ + ۵ \times ۲۰ + ۳ \times ۲۵ = \text{فوت کل تقاضای مشتریان}$$

$$۱۷(x_۱ + x_۲ + x_۳ + x_۴ + x_۵ + x_۶) = \text{طول کل قطعات برش داده شده}$$

بنابراین

$$۱۷x_۱ + ۱۷x_۲ + ۱۷x_۳ + ۱۷x_۴ + ۱۷x_۵ + ۱۷x_۶ - ۳۱۰ = \text{ضایعات شرکت}$$

به این ترتیب، هدف شرکت عبارت است از:

$$\min z = ۱۷x_۱ + ۱۷x_۲ + ۱۷x_۳ + ۱۷x_۴ + ۱۷x_۵ + ۱۷x_۶ - ۳۱۰$$

که معادل است با کمینه ساختن

$$۱۷(x_۱ + x_۲ + x_۳ + x_۴ + x_۵ + x_۶)$$

و یا

$$x_۱ + x_۲ + x_۳ + x_۴ + x_۵ + x_۶$$

بنابراین تابع هدف مسئله

$$\min z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6$$

یعنی شرکت می تواند ضایعات را از طریق تعداد قطعات ۱۷ فوتی که برش می‌دهد کمینه سازد.

قیود این مسئله عبارتند از:

قید ۱: حداقل ۲۵ قطعه ۳ فوتی لازم است.

قید ۲ : حداقل ۲۰ قطعه ۵ فوتی لازم است.

قید ۳: حداقل ۱۵ قطعه ۹ فوتی لازم است.

از آنجا که تعداد قطعات ۳ فوتی برابر

$$5x_1 + 4x_2 + 2x_3 + 2x_4 + x_5$$

است، پس قید یک عبارت است از:

$$5x_1 + 4x_2 + 2x_3 + 2x_4 + x_5 \geq 25$$

به طور مشابه قید (۲)

$$x_2 + 2x_3 + x_5 + 3x_6 \geq 20$$

و قید (۳) عبارت است از:

$$x_4 + x_5 \geq 15$$

توجه کنید که ضریب x_i باید عددی صحیح فرض شود. علی رغم این حقیقت، در مسائل بزرگ مقیاس یک جواب نزدیک به بهینه را می‌توان از حل مسئله، به منزله یک مسئله LP ویا گرد کردن مقادیر کوچک به سمت بالا به‌دست آورد. البته این کار ممکن است به بهینه‌ترین پاسخ عدد صحیح مسئله منجر نشود، ولی معمولا یک جواب نسبتا خوب ارائه می‌دهد. به این دلیل، ما روی حالت LP مسئله تمرکز می‌کنیم. بنابراین، داریم:

$$\begin{aligned}
\min z = & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \\
s.t. \quad & 5x_1 + 4x_2 + 2x_3 + 2x_4 + x_5 \geq 25 \\
& x_2 + 2x_3 + x_5 + 3x_6 \geq 20 \\
& x_4 + x_5 \geq 15 \\
& x_1, x_2, x_3, x_4, x_5, x_6 \in z^+
\end{aligned} \tag{۳-۲۶}$$

توجه کنید که $x_۱$ فقط در قید اول وجود دارد (زیرا فقط طرح یک قطعه ۳ فوتی تولید می‌کند) و به طور مشابه $x_۶$ فقط در قید دوم وجود دارد. لذا می‌توان از $x_۱$ و $x_۶$، به عنوان متغیر پایه‌ای در قیود اول و دوم استفاده کرد. متاسفانه قید سوم فاقد یک متغیر پایه‌ای بدیهی است. برای اجتناب از افزودن متغیر مصنوعی، الگوی برش ۷ را تعریف می‌کنیم. این طرح فقط قطعه ۹ فوتی تولید می‌کند. هم‌چنین $x_۷$ را به‌منزله تعداد برش با الگوی ۷ در نظر می‌گیریم. ستون نظیر $x_۷$ در مسئله CSP عبارت است از:

$$\text{ستون نظیر } x_۷ \text{ در مسئله LP} = \begin{bmatrix} ۰ \\ ۰ \\ ۱ \end{bmatrix}$$

و یک جمله $x_۷$ با ضریب یک به تابع هدف افزوده خواهد شد. اکنون $B = \{x_۱, x_۶, x_۷\}$ اولین پایه مسئله LP ؟؟ است. اگر جدول نظیر این پایه را جدول صفر بنامیم، داریم:

$$\mathbf{B}_۰ = \begin{bmatrix} ۵ & ۰ & ۰ \\ ۰ & ۳ & ۰ \\ ۰ & ۰ & ۱ \end{bmatrix} \quad \Longrightarrow \quad \mathbf{B}_۰^{-۱} = \begin{bmatrix} \frac{۱}{۵} & ۰ & ۰ \\ ۰ & \frac{۱}{۳} & ۰ \\ ۰ & ۰ & ۱ \end{bmatrix}$$

$$\mathbf{c_B B}_۰^{-۱} = \begin{bmatrix} ۱ & ۱ & ۱ \end{bmatrix} \begin{bmatrix} \frac{۱}{۵} & ۰ & ۰ \\ ۰ & \frac{۱}{۳} & ۰ \\ ۰ & ۰ & ۱ \end{bmatrix} = \begin{bmatrix} \frac{۱}{۵} & \frac{۱}{۳} & ۱ \end{bmatrix}$$

اکنون اگر متغیرهای غیرپایه‌ای را قیمت‌گذاری کنیم، می‌توانیم متغیر وارد شونده را تعیین نماییم. در مسائل بزرگ مقیاس که هزاران متغییر وجود دارد، قیمت‌گذاری همه متغیرهای غیرپایه‌ای، کاری بس دشوار و خسته‌کننده است. در چنین وضعیتی، روش تولید ستون نقشی مهم و کلیدی بازی می‌کند. از آنجا که یک مسئله کمینه سازی را حل می‌کنیم، پس به متغیری با قیمت مثبت (ضریب مثبت در سطر صفر جدول فعلی) نیاز داریم. در مسئله برش، هر ستون یا متغیر، بیانگر طرحی از برش است؛ یعنی یک متغیر با ۳ عدد $a_۳$،$a_۵$ و $a_۱$ مشخص می‌شود که a_i تعداد قطعات i فوتی است که از برش یک قطعه ۱۷ فوتی بر طبق طرح داده شده به دست می‌آید. برای مثال متغیر $x_۲$ با $a_۳ = ۴$ و $a_۵ = ۱$ مشخص می‌شود. ایده اصلی در روش تولید ستون آن است که به روشی موثر، ستونی را بیابیم که قیمتی مناسب (مثبت در مسئله کمینه سازی و منفی در مسئله بیشینه سازی)برای ورود به پایه داشته باشد. بااستفاده از رابطه ؟؟ $(c_B B^{-۱} a_j - c_j)$ برای پایه فعلی قیمت گذاری ترکیبی از $a_۳$،$a_۵$ و $a_۱$ عبارت است از:

$$\mathbf{c_B B_0^{-1}} \begin{bmatrix} a_3 \\ a_5 \\ a_9 \end{bmatrix} - 1 = \frac{1}{5}a_3 + \frac{1}{3}a_5 + a_9 - 1 \tag{۲۷-۳}$$

توجه کنید که در رابطه ؟؟ a_3,a_5 و a_9 باید طوری انتخاب شوند که بیش از ۱۷ فوت چوب استفاده نکنند درحالی که بیشترین قیمت را برای ورود به پایه ایجاد کنند. همچنین می دانیم که a_3,a_5 و a_9 باید صحیح و نامنفی باشند. این حالت مشابه مسئله کوله‌پشتی است که حجم کوله درآن برابر ۱۷ فوت و ارزش کالاهای موجود برابر $\frac{1}{5}$ و $\frac{1}{3}$ و ۱ است. اکنون ترکیبی که دارای بهترین قیمت است از حل مسئله کوله پشتی زیر به‌دست می‌آوریم :

$$\begin{aligned} \max z =& \frac{1}{5}a_3 + \frac{1}{3}a_5 + a_9 - 1 \\ s.t. \quad & 3a_3 + 5a_5 + 9a_9 \leq 17 \\ & a_3, a_5, a_9 \in z^+ \end{aligned} \tag{۲۸-۳}$$

مسئله ؟؟ یک مسئله کوله پشتی بدون محدودیت صفر-یک روی متغیر ها است که می‌توان با روش‌های مختلفی (مانند شاخه و برش، الگوریتم ژنتیک ، روش حریصانه) آن را حل نمود.

جواب بهینه مسئله ؟؟ عبارت است از: $a_3, a_5, a_9 = 1$ و $z = \frac{8}{15}$ لذا قیمت $x_5 = \frac{8}{15}$ و x_5 وارد پایه می‌شود که برای این منظور بایستی سمت راست جدول فعلی و ستون نظیر x_5 را محاسبه کنیم.

$$\text{ستون } x_5 \text{ در جدول فعلی} = B_0^{-1}a_5 = \begin{bmatrix} \frac{1}{5} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{5} \\ \frac{1}{3} \\ 1 \end{bmatrix}$$

$$\text{سمت راست جدول فعلی} = B_0^{-1}b = \begin{bmatrix} \frac{1}{5} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 25 \\ 20 \\ 15 \end{bmatrix} = \begin{bmatrix} 5 \\ \frac{20}{3} \\ 15 \end{bmatrix}$$

آزمون نسبت نشان می‌دهد که x_5 متغیر پایه ای سطر سوم است. پس $B(1) = \{x_1,x_6,x_5\}$ با استفاده از شکل ضربی وارون داریم :

$$B_1^{-1} = E_0 B_0^{-1} = \begin{bmatrix} 1 & 0 & -\frac{1}{5} \\ 0 & 1 & -\frac{1}{3} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{5} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{5} & 0 & -\frac{1}{5} \\ 0 & \frac{1}{3} & -\frac{1}{3} \\ 0 & 0 & 1 \end{bmatrix}$$

$$c_B B_1^{-1} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{5} & 0 & -\frac{1}{5} \\ 0 & \frac{1}{3} & -\frac{1}{3} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{5} & \frac{1}{3} & \frac{7}{15} \end{bmatrix}$$

با قیمت‌های سایه‌ای جدید ($c_B B_1^{-1}$) می‌توان با روش تولید ستون مشخص نمود که آیا ترکیبی برای ورود به پایه وجود دارد یا خیر. برای قیمت های سایه فعلی، قیمت ترکیبی از a_3, a_5 و a_9 عبارت است از :

$$\begin{bmatrix} \frac{1}{5} & \frac{1}{3} & \frac{7}{15} \end{bmatrix} \begin{bmatrix} a_3 \\ a_5 \\ a_9 \end{bmatrix} - 1 = \frac{1}{5}a_3 + \frac{1}{3}a_5 + \frac{7}{15}a_9 - 1$$

بنابراین برای جدول فعلی روش تولید ستون به مسئله زیر منتهی می‌شود :

$$\begin{aligned} \max z = & \frac{1}{5}a_3 + \frac{1}{3}a_5 + \frac{7}{15}a_9 - 1 \\ s.t. \quad & 3a_3 + 5a_5 + 9a_9 \leq 17 \\ & a_3, a_5, a_9 \geq 0; a_3, a_5, a_9, integer \end{aligned} \tag{۳-۲۹}$$

با حل مسئله کوله پشتی ؟؟ باروش برنامه‌ریزی‌پویا داریم : $a_3 = 4, a_5 = 1, a_9 = 0$ بنابراین x_2 وارد پایه می‌شود.

ستون نظیر x_2 در جدول فعلی $= \begin{bmatrix} \frac{4}{5} \\ \frac{1}{3} \\ 0 \end{bmatrix}$ سمت راست جدول فعلی $= \begin{bmatrix} 2 \\ \frac{5}{3} \\ 15 \end{bmatrix}$

آزمون نسبت نشان می‌دهد که x_2 متغیر پایه ای سطر یک خواهد بود. بنابراین $B(2) = x_2, x_6, x_5,$ با استفاده از شکل ضربی وارون داریم :

$$E_1 = \begin{bmatrix} \frac{5}{4} & 0 & 0 \\ -\frac{5}{12} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$B_{۲}^{-۱} = E_{۱}B_{۱}^{۱} = \begin{bmatrix} \frac{۵}{۴} & ۰ & ۰ \\ -\frac{۵}{۱۲} & ۱ & ۰ \\ ۰ & ۰ & ۱ \end{bmatrix} \begin{bmatrix} \frac{۱}{۵} & ۰ & -\frac{۱}{۵} \\ ۰ & \frac{۱}{۳} & -\frac{۱}{۳} \\ ۰ & ۰ & ۱ \end{bmatrix} = \begin{bmatrix} \frac{۱}{۴} & ۰ & -\frac{۱}{۴} \\ -\frac{۱}{۱۲} & \frac{۱}{۳} & -\frac{۱}{۴} \\ ۰ & ۰ & ۱ \end{bmatrix}$$

مقادیر جدید قیمت سایه‌ای عبارتنداز :

$$c_B B_{۲}^{-۱} = \begin{bmatrix} ۱ & ۱ & ۱ \end{bmatrix} \begin{bmatrix} \frac{۱}{۴} & ۰ & -\frac{۱}{۴} \\ -\frac{۱}{۱۲} & \frac{۱}{۳} & -\frac{۱}{۴} \\ ۰ & ۰ & ۱ \end{bmatrix} = \begin{bmatrix} \frac{۱}{۶} & \frac{۱}{۳} & \frac{۱}{۲} \end{bmatrix}$$

با توجه به قیمت سایه‌ای جدید مسئله تولید ستون به‌صورت زیر تغریف می‌شود:

$$\begin{aligned} \max z =& \frac{۱}{۶}a_{۳} + \frac{۱}{۳}a_{۵} + \frac{۱}{۲}a_{۹} - ۱ \\ s.t. \quad & ۳a_{۳} + ۵a_{۵} + ۹a_{۹} \leq ۱۷ \\ & a_{۳}, a_{۵}, a_{۹} \geq ۰; a_{۳}, a_{۵}, a_{۹} \text{عدد صحیح} \end{aligned} \tag{۳۰-۳}$$

با حل مسئله کوله پشتی ؟؟ داریم: $a_{۳} = ۴, a_{۵} = ۱, a_{۹} = ۰$ بنابراین $z = ۰$ می‌شود بنابراین هیچ متغیری نمی‌تواند وارد پایه شود. این به آن معناست که جواب پایه فعلی جواب بهینه مسئله است. برای تعیین مقدار متغیرهای پایه‌ای سمت راست جدول فعلی را محاسبه می‌کنیم:

$$\text{سمت راست جدول فعلی} = B_{۲}^{-۱}b = \begin{bmatrix} \frac{۱}{۴} & ۰ & -\frac{۱}{۴} \\ -\frac{۱}{۱۲} & \frac{۱}{۳} & -\frac{۱}{۴} \\ ۰ & ۰ & ۱ \end{bmatrix} \begin{bmatrix} ۲۵ \\ ۲۰ \\ ۱۵ \end{bmatrix} = \begin{bmatrix} \frac{۵}{۲} \\ \frac{۵}{۶} \\ ۱۵ \end{bmatrix}$$

بنابراین جواب بهینه مسئله عبارت است از : $x_{۵} = ۱۵, x_{۶} = \frac{۵}{۶}, x_{۲} = \frac{۵}{۲}$ که می‌توانیم با گرد کردن به یک جواب صحیح به صورت $x_{۵} = ۱۵, x_{۶} = ۱, x_{۲} = ۲$ برسیم.در روش تولید ستون تنها با داشتن یک جواب پایه‌ای شدنی و بدون لیست کردن فهرستی از سایر پاسخ های شدنی می توانیم به جواب بهینه برسیم چراکه درهر تکرار یک ترکیب خوب که پاسخ z را بهبود ببخشد با حل یک مسئله IP به‌دست می‌آید.

فصل ۴

معرفی روش‌های ابتکاری

۴-۱ مقدمه

توسعه پایدار، توسعه‌ای است که نیازهاي کنونی جهان را بی آنکه توانایی نسل هاي آتی را در رفع نیازهای خود به مخاطره اندازد، تأمین می نماید [؟] . امروزه، توسعه پایدار در کشورهاي در حال توسعه (مانند ایران) به یکی از الزامات و ضروریات منطقی تبدیل شده است. از سویی دیگر در سال هاي اخیر، کمبود منابع جنگلی و چوبی در کشور، حقیقتی تلخ و انکارناپذیر است که کاهش ظرفیت تولید و حتی تعطیلی تعدادي از کارخانه هاي مرتبط با صنایع چوب را به دنبال داشته است. بر اساس مطالعات انجام شده، ارزش محیط زیستی و اکولوژیکی جنگل ها تا ۴۰۰ برابر ارزش تولید چوب آنها برآورد شده است. آمار و گزارش هاي سازمان جنگل ها نشان میدهد که بهره برداری از جنگل هاي شمال طی ده سال گذشته از بیش از دو میلیون مترمکعب به کمتر از یک میلیون مترمکعب رسیده و این به معنی کاهش ۵۰ درصدی در عرضه چوب به بازار مصرف است. در تولید محصولات چوبی سالیانه مقدار زیادی ضایعات چوبی ایجاد می شود (به طور مثال این مقدار در امریکا حدود ۶۳ میلیون تن در سال ۲۰۰۲ بوده است) که استفاده از این ضایعات در تولید، باعث پایین آمدن قیمت تمام شده محصول می‌شود [؟] .

کشور ایران با توجه به کمبود منابع جنگلی و همچنین طرح‌های صیانت از جنگلها، با مشکل جدي تامین مواد اولیه چوبی جهت تولید فراورده‌هاي مختلف چوبی مواجه می‌باشد. از اینرو تلاش برای ارائه راه‌کارهای عملیاتی جهت کاهش ضایعات حاصل از برش در کنار صنایع تبدیلی (بازیافت) امری ضروری است. چراکه صنایع تبدیلی (مانند ساخت تخته های نوپان) با صرف هزینه، ضایعات را در نهایت به مواد اولیه‌ای برای تولید تبدیل می‌کنند که آن هم شامل ضایعات برش می‌گردد. پس منطقی‌تر آن است که تا حد امکان از تولید ضایعات جلوگیری بعمل آید تا هزینه های کمتری برای بازیافت آن پرداخته شود. به همین منظور در این پژوهش با طرح یک مسئله برش در یک کارگاه نجاری، تاثیرات استفاده از روش های (ابتکاری) مطرح شده به روی کاهش ضایعات حاصل از برش مورد بررسی قرار گرفته است.

در این فصل چند روش حل برای مسئله برش موجودی ارائه شده است. به‌منظور بررسی و مقایسه عملکرد این روش‌ها، مسئله کارگاه نجاری برگرفته از مرجع [؟] مورد مطالعه قرار گرفته است. در مقاله [؟] برای حل مسئله کارگاه نجاری، از روش تولید ستون استفاده شده است. روش‌های ابتکاری ارائه شده در این پژوهش عبارتند از:

۱. روشی ابتکاری برای به دست آوردن جواب تقریبی در حل مسئله نجاری

۲. حل صفحات برش با استفاده از روش تولید ستون و برنامه ریزی پویا

۳. حل صفحات برش با استفاده از روش مبتنی بر تولید ستون و روش حریصانه

۴. حل صفحات برش با استفاده از ترکیب روش مبتنی بر تولید ستون و الگوریتم ابتکاری پیشنهادی

۵. حل صفحات برش با استفاده از روش مبتنی بر الگوریتم ژنتیک

۶. حل صفحات برش با استفاده از روش مبتنی بر تولید ستون و ترکیب با روش مبتنی بر الگوریتم ژنتیک

   (آ) بدون در نظر گرفتن مقدار جریمه در هزینه

   (ب) با احتساب مقدار جریمه در هزینه

۴-۲ مسئله نجاری

مثال ۴-۲-۱. در کارگاه نجاری مشابه بسیاری از صنایع دیگر با مسئله برش موجودی با هدف کاهش ضایعات مواجه هستیم. از اینرو در این پژوهش به‌منظور مقایسه روش‌های حل ارائه شده، مسئله کارگاه نجاری برگرفته از مرجع [؟] مورد بررسی قرار می‌گیرد.

در این مسئله فرض شده ۲۰ میله به طول L= ۴۲۵ سانتی‌متر وجود دارد که می‌خواهیم با استفاده از آنها سفارشات زیر را تولید کنیم:

۲۵ میله ۷۵= $l_۱$ سانتی‌متری

۲۰ میله ۱۲۵ = $l_۲$ سانتی‌متری

۱۵ میله ۲۲۵ = $l_۳$ سانتی‌متری

و نیز مایل هستیم که این سفارشات با کمترین هزینه و ضایعات تولید شوند [؟].

۳-۴ فرمول بندی ریاضی مسئله نجاری

به‌منظور تامین سفارشات به الگوهای برش که به آن‌ها x_j می‌گوییم، نیازمندیم. این الگو ها از ترکیب و کنار هم قراردادن تعدادی از سفارشات در کنار یکدیگر تولید می‌شوند. شرط شدنی بودن الگوی برش این است که مجموع سفارشات تعیین شده در آن کمتر یا مساوی طول موجودی باشد .الگوهای بسیاری برای برش وجود دارد اما بررسی همه آنها لازم نیست. برای مثال عاقلانه نیست در مسئله نجاری الگوهایی که ضایعات بزرگتر از ۷۵ سانتی متر (مقدار کوچکترین سفارش) داشته باشند استفاده شوند. برای نمونه جدول الگوهای ؟؟ را در نظر بگیرید.

جدول ۴-۱: الگوها و ضایعات آن [؟]

شماره الگو	تعداد برش ۷۵سانتی متر	تعداد برش ۱۲۵سانتی متر	تعداد برش ۲۲۵سانتی متر	ضایعات برش
۱	۵	۰	۰	۵۰
۲	۴	۱	۰	۰
۳	۲	۲	۰	۲۵
۴	۲	۰	۱	۵۰
۵	۱	۱	۱	۰
۶	۰	۳	۰	۵۰

همانطور که مشاهده می‌شود، مجموع سفارشات تولید شده در هر کدام از الگوهای بالا از طول مرجع کمتر است. فرمول برنامه‌ریزی عدد صحیح این مسئله به صورت زیر نوشته می‌شود.

$$
\begin{aligned}
&\min \sum_{j=1}^{n} C_j x_j \\
s.t. \quad &\sum_{j=1}^{n} a_{ij} x_j \geqslant d_i; \quad \forall i = 1, ..., m \\
&x \in Z^{+}
\end{aligned}
\tag{۴-۱}
$$

a_{ij} تعداد تولید سفارش i در الگوی j است. C_j هزینه(اغلب ضایعات) الگوی j است. x_j تعداد الگوی j ام است . d_i تعداد سفارشات i ام است. در جدول ؟؟ اگر کمینه بودن ضایعات تنها ملاک تصمیم گیری باشد، ساده‌ترین راه‌حل استفاده از الگوی پنجم و به تعداد بیشترین سفارش است. در این حالت ضایعات نداریم اما اضافه تولیدی به اندازه ۲۸۷۵ سانتی‌متر خواهیم داشت که بیش از ۶ میله ۴۲۵ سانتی‌متری را شامل می‌شود. درصورتی که اضافه تولید مصرف نشود خود به عنوان ضایعات محسوب می‌گردد؛ زیرا اضافه تولید نیاز به هزینه‌های حمل و نقل، انبارداری،بازاریابی و ... دارد که درصورتی که پیش‌بینی‌های لازم درخصوص آن اندیشیده نشده باشد می‌تواند منجر به زیان‌دهی تولید گردد.

۴-۴ روش‌های ابتکاری حل مسئله نجاری

۴-۴-۱ روشی ابتکاری برای به دست آوردن جواب تقریبی در حل مسئله نجاری

در این روش با استفاده از روش سیمپلکس و تغییر در کران بالای مسئله برای یافتن پاسخی مناسب تلاش می‌کنیم. البته زمانی می‌توان از این روش استفاده کرد که الگوها از قبل تعیین شده‌اند و هدف انتخاب بهترین الگو از میان الگوهای موجود است. در این روش در هر تکرار مقادیری از سفارشات را تولید می‌کنیم. اگر میزان تولید با سفارش برابر بود مسئله حل شده است در غیراین‌صورت، باقیمانده موجودی و مقدار سفارشات تولید نشده، محاسبه و دوباره مسئله با داده‌های جدید حل می‌شود. شرط خاتمه تولید شدن تمام سفارشات است.

۴-۴-۱-۱ تعریف تابع هدف و محدودیت ها

کمینه کردن ضایعات یکی از مهمترین اهداف حل مسئله برش موجودی است، ازآنجایی‌که هر الگو فقط به روی یک موجودی اعمال می گردد، بنابراین اگر تعداد الگوها کاهش پیدا کند، مقدار موجودی مصرف شده برای تولید سفارشات نیز کاهش می‌یابد. هم‌چنین مجموع ضایعات حاصل از برش با استفاده از الگو j ام با تعداد استفاده از الگوی j ارتباط مستقیم دارد. بنابراین تابع هدف با توجه به رابطه ؟؟ به صورت زیر تعریف می‌شود:

$$\begin{aligned}
&\min Z = (۵۰x_۱ + ۰x_۲ + ۲۵x_۳ + ۵۰x_۴ + ۰x_۵ + ۵۰x_۶) \qquad (۴\text{-}۲)\\
&s.t. \quad -(۵x_۱ + ۴x_۲ + ۲x_۳ + ۲x_۴ + ۱x_۵ + \quad) \leq -۲۵\\
&-(\quad +۱x_۲ + ۲x_۳ + \quad +۱x_۵ + ۳x_۶) \leq -۲۰\\
&-(\quad + \quad + \quad +۱x_۴ + ۱x_۵ + \quad) \leq -۱۵\\
&-(۳۷۵x_۱ + ۴۲۵x_۲ + ۴۰۰x_۳ + ۳۷۵x_۴ + ۴۲۵x_۵ + ۳۷۵x_۶)...\\
&\leq -(۲۵ \times ۷۵ + ۲۰ \times ۱۲۵ + ۱۵ \times ۲۲۵) = -۷۷۵۰\\
&۳۷۵x_۱ + ۴۲۵x_۲ + ۴۰۰x_۳ + ۳۷۵x_۴ + ۴۲۵x_۵ + ۳۷۵x_۶ \leq ۲۰ \times ۴۲۵ = ۸۵۰۰\\
&x_۱, x_۲, x_۳, x_۴, x_۵, x_۶ \in \mathbb{Z}^+
\end{aligned}$$

c_i ها میزان ضایعات تولید شده توسط الگو j را نشان می‌دهند. x_i بیانگر تعداد استفاده شده از الگویi است.برای نمونه در الگوی اول، ۵ برش ۷۵ سانتی متری وجود دارد. لذا مجموع برش داده شده با این الگو برابر است با

$$c_i = L - \sum_{j=۱}^{n} a_{ij} \times l_i \, , i = ۱, ..., m \qquad (۴\text{-}۳)$$

$$c_1 = ۴۲۵ - \begin{bmatrix} ۷۵ & ۱۲۵ & ۲۲۵ \end{bmatrix} \begin{bmatrix} ۵ \\ ۰ \\ ۰ \end{bmatrix}$$

$$c_1 = ۴۲۵ - (۷۵ \times ۵ + ۱۲۵ \times ۰ + ۲۲۵ \times ۰)$$

$$c_1 = ۴۲۵ - ۳۷۵ = ۵۰$$

با توجه به حاصل عبارت ؟؟ میزان ضایعات به‌دست آمده از الگوی اول برابر با ۵۰ سانتی‌متر است. به‌طور مشابه برای سایر الگوهای مطرح شده می‌توانیم رابطه ؟؟ را محاسبه نماییم. بنابراین مقدار C به‌صورت زیر به‌دست می‌آید.

$$\mathbf{C} = \begin{bmatrix} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \end{bmatrix} \tag{۴-۴}$$
$$\mathbf{C} = \begin{bmatrix} ۵۰ & ۰ & ۲۵ & ۵۰ & ۰ & ۵۰ \end{bmatrix}$$

ورودی‌های تابع در واقع همان الگوهای ما هستند. لذا ماتریسA را می‌توان به صورت ماتریس ؟؟ تعریف کرد. درواقع ماتریسA ماتریس ضرائب متغیرهای غیر پایه است.

$$\mathbf{A} = \begin{bmatrix} -۵ & -۴ & -۲ & -۲ & -۱ & ۰ \\ ۰ & -۱ & -۲ & ۰ & -۱ & ۳ \\ ۰ & ۰ & ۰ & -۱ & -۱ & ۰ \\ -۳۷۵ & -۴۲۵ & -۴۰۰ & -۳۷۵ & -۴۲۵ & -۳۷۵ \\ ۳۷۵ & ۴۲۵ & ۴۰۰ & ۳۷۵ & ۴۲۵ & ۳۷۵ \end{bmatrix} \tag{۵-۴}$$

ماتریسA در الگوریتم پیشنهادی از سه قسمت تشکیل شده است. قسمت اول مربوط به سه سطر اول ماتریس است که هر سطر متناظر با یکی از سفارشات مسئله است.به عبارت دیگر هر عنصر در سطر i بیانگر مقدار تولید سفارش i ام در ستون j است (ستون j معرف الگوی j است) به‌شرط آنکه، مجموع عناصر در هر سطر بزرگتر یا مساوی حجم سفارش متناظر با آن سطرباشد. قسمت دوم مربوط به سطر چهارم است که نشان‌دهنده حداقل موجودی لازم برای تولید سفارشات در مسئله نجاری است. به‌عبارت دیگر حداقل موجودی مورد نیازبرای تولید تمام سفارشات، با جمع موجودی مصرف شده براساس تعداد هر الگوی انتخاب شده باید برابر باشد. سومین قسمت، سطر پنجم است که بیانگر مقدار موجودی است یعنی اینکه مقدار تولید نباید از میزان کل موجودی فراتر رود. همچنین عناصر در هر ستون به عنوان الگوهای برش، باید به

گونه ای انتخاب شوند که مجموع آنها کوچکتر یا مساوی از طول موجودی گردد. به عبارت دیگر باید $c_i \leq L$ باشد.

- نکته : مجموع برش‌های صورت گرفته با استفاده از الگوهای برش، باید بزرگتر-مساوی از مجموع سفارشات و کوچکتر-مساوی از کل موجودی خام (برش نخورده) باشد.فرض کنیم تعداد کل موجودی برابر با num و اندازه طول هر سفارش برابر l_i باشد در این صورت داریم:

$$\sum_i l_i \times d_i \leq \sum_j x_j (\sum_i a_{ij} \times l_i) \leq \sum num \times L \qquad (۶-۴)$$

ماتریسb به‌صورت $b = [d_۱; d_۲; d_۳; z; s]$ تعریف می‌شود. در این تعریف d_i ها مطابق آنچه در ؟؟ شرح داده شد، بیانگر مقدار مورد نیاز از سفارش i ام است. همچنین z مقدار موجودی باقیمانده پس از اعمال برش توسط الگوهای برش است و s موجودی لازم برای مقدار سفارشات تولید نشده است. بنابراین b به‌صورت زیر نشان داده می‌شود.

$$b = [-۲۵; -۲۰; -۱۵; -۸۵۰۰; ۷۷۵۰]$$

۴-۱-۴-۲ پیاده‌سازی

روش پیشنهادی به شرح الگوریتم ؟؟ در نرم افزار متلب پیاده‌سازی شده است.

۴-۱-۴-۳ تحلیل نتایج محاسباتی

همانطور که در جدول ؟؟ ملاحظه می‌شود مسئله تنها با دو الگوی ۵ و ۲ با میزان ضایعات صفر حل شده است. هم‌چنین در مجموع ۲۰ میله ۴۲۵ سانتی‌متری استفاده شده است. این درحالی‌ست که میزان تولید قطعه ۷۵ سانتی‌متری یک واحد و قطعه ۲۲۵ سانتی‌متری سه واحد افزایش داشته است.

۴-۴-۲ حل مسئله نجاری با استفاده از روش تولید ستون و برنامه ریزی پویا

با توجه به آنچه در فصل سوم بخش ؟؟ دررابطه‌با روش تولید ستون شرح داده شده است، به حل مسئله نجاری می‌پردازیم. در این پژوهش از روش برنامه ریزی پویا برای حل مسئله کوله پشتی صفر و یک استفاده شده است که نتایج استفاده از ۱۹ میله و ۳ الگو و میزان ضایعات ۵۰ را نشان می‌دهد.

الگوریتم ۴-۱ چارچوب کلی الگوریتم ابتکاری پیشنهادی [؟].

۱. مقدار کران پایین x_i را برابر با صفر در نظر می‌گیریم زیرا تعداد استفاده از هر الگو نامنفی است.

۲. تا زمانیکه تمامی سفارشات تولید نشده است ، الگوریتم را تکرار کنید.

(آ) ماتریسb به‌صورت $b = [d_۱; d_۲; d_۳; z; s]$ تعریف می‌شود که در طول تکرار حلقه با تغییرات مقادیر d_i ، s و z تغییر می‌کنند.

(ب) با برنامه‌ریزی‌خطی مسئله بدون در نظر گرفتن کران حل می‌شود. در این گام پاسخ‌های بدست آمده به سمت پایین گرد می‌شوند، زیرا ممکن است مجموع برش‌های بدست آمده از الگوی برنامه‌ریزی خطی کوچکتر مساوی مقدار موجودی شود.

(ج) اگر در پاسخ دستور مرحله ؟؟ اعداد منفی وجود داشته باشد مقدار صفر جایگزین می شود. زیرا پاسخ منفی پاسخی نشدنی و غلط است.

(د) به‌صورت تصادفی یک واحد به یکی از پاسخ‌های حاصل از مرحله ؟؟ اضافه می‌شود و سپس به عنوان کران بالا در تکرارهای بعدی در گام ؟؟ قرار داده می‌شود. مجدداً برنامه ریزی‌خطی با استفاده از کران بالای تعریف شده به تعداد از پیش تعریف شده، تکرار می‌شود.

(ه) مقدار موجودی استفاده شده و سفارشات باقیمانده برای تکرار بعدی محاسبه می‌شود.

(و) اگرسفارشی به‌طور کامل تولید شود، برای مرحله بعد مقدار آن صفر در نظر گرفته می‌شود در غیر این صورت باید میزان سفارش باقیمانده برای تکرار بعد محاسبه گردد.

برای این منظور، مقدار سفارشات تولید شده از مقدار تقاضای اولیه کم می‌شود.

$$d(۱)_i = d(۰)_i - \sum_i \sum_j a_{ij} \times x_j \qquad (۴-۷)$$

از آنجایی که در هر تکرار مسئله برای سفارشات باقیمانده حل می‌شود لذا پاسخ نهایی مجموع پاسخ‌های تمامی تکرارهاست.

۳. در پایان پاسخ‌های به دست آمده از هر تکرار با هم مقایسه و جوابی که دارای کمترین باقی مانده است به‌عنوان پاسخ نهایی پذیرفته می‌شود.

۴-۴-۳ حل مسئله نجاری با استفاده از روش مبتنی‌بر تولید ستون و روش حریصانه

همانطورکه در فصل ۳ اشاره شد در روش تولید ستون برای حل زیر مسئله از الگوریتم‌های متنوعی می‌توان استفاده نمود. در اینجا برای حل زیر مسئله کوله‌پشتی از روش حریصانه استفاده می‌کنیم. مسئله بخش ؟؟ را در نظر بگیرید. با این تفاوت که در اینجا باید الگوهای برش را تولید کنیم.

جدول ۴-۲: نتایج حل مسئله نجاری با استفاده از روش ابتکاری پیشنهادی

عنوان	تعداد
مجموع الگوهای استفاده شده	۲۰
تعداد استفاده شده از الگوی ۶	۰
تعداد استفاده شده از الگوی ۵	۱۸
تعداد استفاده شده از الگوی ۴	۰
تعداد استفاده شده از الگوی ۳	۰
تعداد استفاده شده از الگوی ۲	۲
تعداد استفاده شده از الگوی ۱	۰
تعداد تولید شده از میله ۲۲۵ سانتی‌متری	۱۸
تعداد تولید شده از میله ۱۲۵ سانتی‌متری	۲۰
تعداد تولید شده از میله ۷۵ سانتی‌متری	۲۶
باقیمانده	۰

۴-۴-۳-۱ گام اول : ساخت الگوی اولیه

ابتدا باید مسئله اولیه را به گونه ای تعریف کرد که یک جواب شدنی داشته باشد. به‌این‌منظور مسئله اولیه به صورت زیر تعریف می‌شود.

$$f = \min \sum_{i=1}^{n} C_i x_i \quad x \geqslant 0, \text{ عدد صحیح} \tag{۴-۸}$$

x_i تعداد الگوی i ام

$$\mathbf{B} = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix} = L/l_i \tag{۴-۹}$$

$$\mathbf{b} = \begin{bmatrix} 25 & 20 & 15 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

به‌منظور شروع از یک پاسخ شدنی، طول موجودی بر طول هر سفارش تقسیم صحیح می‌شود(اگر حاصل تقسیم

اعشاری شد به سمت پایین گرد می‌کنیم). به این علت این پاسخ شدنی است که مجموع حاصل ضرب هر سطر در بردار سفارشات از طول مرجع عبور نمی‌کند و نیز سطر و یا ستون صفر در پاسخ نداریم.

۴-۴-۳-۲ گام دوم : دوگان مسئله اولیه

در برنامه‌ریزی خطی برای هر مسئله، مسئله ای بنام دوگان [1] تعریـف می شود که خود، یک مسئله برنامه‌ریزی خطی است و رابطه‌ای تنگاتنگ با مسئله اصلی دارد. بـا داشـتن جدول بهینه‌ی یکی از مسائل، جواب بهینه‌ی مسئله‌ی دیگر را می‌توان بدست آورد. گاه حل مسأله دوگان که تعداد محدویت‌های کمتری نسبت به مسئله اولیه داشته باشد، راحت‌تر از حـل مسئله اولیـه است. در این مرحله باید دوگان مسئله اولیه حل شود. لذا ابتدا دوگان مسأله اولیه بدست آورده می‌شود. با استفاده

جدول ۴-۳: تبدیل دوگان.

	مسئله بیشینه سازی	مسئله کمینه سازی	
متغیرها	$y \geq 0$ $y \leq 0$ $free$	$x \geq$ $x \leq$ $=$	محدودیت‌ها
محدودیت‌ها	$y \leq$ $y \geq$ $=$	$x \geq 0$ $x \leq 0$ $free$	متغیرها

از روابط جدول ؟؟ که برای دوگان گیری از یک مسأله صادق است (اگر محدودیتی در مسئله اول به صورت تساوی باشد، متغیر مربوط به آن محدودیت در مسئله دوگان، آزاد در علامت خواهد بود) می‌توان روابط زیر را به دست آورد:

$$\max z = 25y_1 + 20y_2 + 15y_3 \tag{۴-۱۰}$$

$$\begin{aligned} s.t. \quad 5y_1 &\leq 1 \\ 3y_2 &\leq 1 \\ 1y_3 &\leq 1 \\ y_1,\ y_2,\ y_3 &\geq 0 \end{aligned} \tag{۴-۱۱}$$

$$Duality_B = B = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix} = L/l_i \tag{۴-۱۲}$$

[1] Dual Simplex

$$Duality_F = b' = \begin{bmatrix} ۲۵ & ۲۰ & ۱۵ \end{bmatrix} \quad (۱۳-۴)$$

$$Duality_b = c' = \begin{bmatrix} ۱ \\ ۱ \\ ۱ \end{bmatrix} \quad (۱۴-۴)$$

درنتیجه حاصل دوگان برابر است با

$$y_۱ = ۰/۲, \quad y_۲ = ۰/۳۳, \quad y_۳ = ۱$$

بنابراین بردار π برابر است با

$$\pi = \begin{bmatrix} ۰/۲ & ۰/۳۳ & ۱ \end{bmatrix}$$

۴-۳-۴-۳ گام سوم : تولید زیر مسئله

مطابق آنچه در فصل سوم شرح داده شده است،اگر حالا یک متغیر غیر پایه را خارج کنیم به ما نشان می‌دهد که کدام متغیر باید وارد پایه شود .اگرچه در یک مسئله برش خیلی بزرگ ممکن است هزاران متغیر داشته باشیم،در چنین وضعیتی روش تولید ستون برای کم کردن استفاده از نقاط گوشه‌ای (نقاط گوشه ای معادل با نقاط شدنی هستند) استفاده می‌شود. از آنجایی که به دنبال حل مسائل کمینه سازی هستیم بنابراین باید به دنبال ستون مثبت تر باشیم (یعنی ضریب سطر صفر مثبت باشد). در مسئله برش موجودی هر ستون یا متغیر، نماینده یک ترکیب برای برش موجودی است. در مسئله نجاری هر متغیر با سه عدد نشان داده می‌شود، مثلاً $a_۱, a_۲, a_۳$ که a_i شماره مسئله iام است که از میله موجودی برش خورده . ایده تولید ستون جستجوی بهینه برای ستونی است که باید خارج شود (در مسئله min مثبت و مسئله max منفی). برای مسئله نجاری یک ترکیب به صورت $\sum a_i$ متغیر خروجی را به صورت ؟؟ مشخص می کند. توجه شود a_i ها باید طوری انخاب شوند که کمتر از طول موجودی شوند و نیز a_i باید صحیح و غیر منفی باشند به طور خلاصه برای هر ترکیب a_i باید داشته باشیم

$$\sum a_i \leq L$$

اکنون با استفاده از حاصل به دست آمده در گام دوم (π) و حل مسئله کوله پشتی ؟؟ میتوانیم متغیر خروجی از پایه را تعیین کنیم.

$$\min(۱-\sum \pi * a_i) \implies \max \sum (\pi * a_i) \tag{۱۵-۴}$$

$$f_{\max_d} = -\pi'_i \tag{۱۶-۴}$$

$$\mathbf{B_{max_d}} = \begin{bmatrix} L/l_۱ & L/l_۲ & L/l_۳ \\ ۱ & ۰ & ۰ \\ ۰ & ۱ & ۰ \\ ۰ & ۰ & ۱ \\ -۱ & ۰ & ۰ \\ ۰ & -۱ & ۰ \\ ۰ & ۰ & -۱ \end{bmatrix} \tag{۱۷-۴}$$

$$\mathbf{b_{max_d}} = \begin{bmatrix} L & L/l_۱ & L/l_۲ & L/l_۳ & ۰ & ۰ & ۰ \end{bmatrix} \tag{۱۸-۴}$$

لذا با توجه به روابط ؟؟ تا ؟؟ داریم:

$$\mathbf{f_{max_d}} = \begin{bmatrix} ۰/۲ & ۰/۳۳ & ۱ \end{bmatrix}$$

$$\mathbf{B_{max_d}} = \begin{bmatrix} ۵ & ۳ & ۱ \\ ۱ & ۰ & ۰ \\ ۰ & ۱ & ۰ \\ ۰ & ۰ & ۱ \\ -۱ & ۰ & ۰ \\ ۰ & -۱ & ۰ \\ ۰ & ۰ & -۱ \end{bmatrix}$$

$$\mathbf{b_{max_d}} = \begin{bmatrix} ۴۲۵ & ۵ & ۳ & ۱ & ۰ & ۰ & ۰ \end{bmatrix}$$

۴-۴-۳-۴ گام چهارم : حل زیر مسئله

مسئله ؟؟ یک مسئله کوله‌پشتی است و برای حل آن می‌توان از الگوریتم‌های متنوعی استفاده کرد. در این مسئله ظرفیت کوله به اندازه طول میله مرجع و ارزش هر شی در آن معادل حاصل حل مسئله دوگان است. مقدار هر شی L/L_i است.

نتیجه حل زیر مسئله تعیین‌کننده شرط خاتمه است. به اینصورت که اگر نتیجه عبارت

$$۱ - \sum \pi * a_i \geq ۰ \qquad (۱۹-۴)$$

شود شرط توقف برقرار شده است و ماتریس B همان ماتریس الگوهای مورد نیاز برای حل مسئله است. در غیر این صورت باید بردار a_i را به عنوان ستون جدید به ماتریس B اضافه کرد و از گام دوم مجدداً اقدام به حل مسئله نمود. اگر با استفاده از برنامه‌ریزی پویا مسئله کوله‌پشتی را حل شود حاصل ۰٫۵۳− می‌گردد. از آنجایی که مقدار زیر مسئله در اولین تکرار منفی است باید مجدداً از گام اول مسئله حل شود. البته در اینصورت یک ستون که همان بردار a_i است، به آن اضافه می‌گردد. بنابراین ماتریسB برابر است با:

$$\mathbf{B} = \begin{bmatrix} ۵ & ۰ & ۰ & ۱ \\ ۰ & ۳ & ۰ & ۱ \\ ۰ & ۰ & ۱ & ۱ \end{bmatrix}$$

با استفاده از ماتریس B تولید شده در تکرا ر بعدی مقدار زیر مسئله نامنفی می‌شود بنابراین ماتریس B همان ماتریسی است که با استفاده از آن مرحله اول شروع شده است. نتایج حاصل در جدول ؟؟ شرح داده شده است.

جدول ۴-۴: .نتایج حل مسئله نجاری با استفاده از روش تولید ستون و برنامه‌ریزی پویا

الگوی ۱	الگوی ۲	الگوی ۳	الگوی ۴	
۵ ۰ ۰	۰ ۳ ۰	۰ ۰ ۱	۱ ۱ ۱	$x_۱ x_۲ x_۳$
۲	۲	۰	۱۵	تعداد الگوی استفاده شده
$۲ \times [۵\ ۰\ ۰]$	$۲ \times [۰\ ۳\ ۰]$	$۰ \times [۰\ ۰\ ۱]$	$۱۵ \times [۱\ ۱\ ۱]$	$x_۱, x_۲, x_۳ = ۲۵, ۲۱, ۱۵$

همانطورکه در جدول ؟؟ مشاهده می‌شود الگوی ۱ و ۲ هر کدام ۲ بار و الگوی چهارم ۱۵ بار استفاده شده است. با توجه به جدول ؟؟ میزان ضایعات برابر با ۲۰۰ سانتی‌متر (۲۰۰= ۲×۵۰ + ۲×۵۰ + ۰×۱۵) است. هم‌چنین در مجموع ۱۹ میله ۴۲۵ سانتی‌متری استفاده شده است و فقط ۱ قطعه ۱۲۵ سانتی‌متری مازاد تولید شده است.

۴-۴-۴ حل مسئله نجاری با استفاده از ترکیب روش مبتنی بر تولید ستون و روش ابتکاری پیشنهادی :

پیاده سازی این روش مشابه روش مبتنی بر تولید ستون و روش حریصانه است با این تفاوت که در ؟؟ برای حل زیر مسئله از روش شرح داده شده در بخش ؟؟ استفاده می‌شود. همان‌طور که قبلا اشاره شد شرط استفاده از روش ؟؟ داشتن الگوهای برش است. ترکیب روش ابتکاری پیشنهادی با روش مبتنی بر تولید ستون این امکان را فراهم می‌آورد، زیرا روش مبتنی بر تولید ستون از ماتریس اولیه (L/l_i) شروع به تولید ستون می‌کند و در هر مرحله زیر مسئله تولید شده به تابع نوشته شده بر مبنای روش ؟؟ یا روش ابتکاری پیشنهادی ،داده می‌شود تا در نهایت بهینه‌ترین تعداد استفاده از هر الگو مشخص شود. در این تابع ابتدا مسئله بدون کران و با استفاده از تابع حل مسائل برنامه ریزی خطی حل می‌شود. میانگین بردار بدست آمده به‌صورت تصادفی به یکی از عناصر بردار C اضافه و به عنوان کران بالا در تکرار بعدی استفاده می‌گردد. کران پایین بردار صفر در نظر گرفته می‌شود زیرا حداقل مقدار ممکن برای تعداد استفاده از یک الگو برابر با صفر است. این عمل به تعداد از پیش تعیین شده‌ای (مثلاً ۵ بار) تکرار می‌شود و در نهایت پاسخی که بیش‌ترین مقدار تابع هدف را بر میگرداند نتایج حاصل در جدول ؟؟ شرح داده شده است.

جدول ۴-۵: نتایج حل مسئله نجاری با استفاده از ترکیب روش مبتنی بر تولید ستون و الگوریتم ابتکاری پیشنهادی .

الگوی ۱	الگوی ۲	الگوی ۳	الگوی ۴	الگوی ۵	
۵ ۰ ۰	۰ ۳ ۰	۰ ۰ ۱	۱ ۱ ۱	۲ ۲ ۰	$x_۱x_۲x_۳$
۱	۰	۰	۱۵	۳	تعداد الگوی استفاده شده
۱ × [۵ ۰ ۰]	۲ × [۰ ۳ ۰]	۰ × [۰ ۰ ۱]	۱۵ × [۱ ۱ ۱]	۳ × [۲ ۲ ۰]	$x_۱, x_۲, x_۳ =$ ۲۶, ۲۱, ۱۵

همانطورکه در جدول ؟؟ مشاهده می‌شود در این روش از ۱۹ الگو استفاده شده است و با توجه به جدول ؟؟ میزان مجموع ضایعات تولید شده برابر ۱۲۵ سانتی‌متر (۳ × ۲۵ + ۰ × ۱۵ +۰+۰+ ۱ × ۵۰) است. هم‌چنین باتوجه‌به اینکه هر الگو یک میله موجودی را برش می دهد لذا ۱۹ میله موجودی استفاده شده است.

۵-۴-۴ حل مسئله نجاری با استفاده از روش مبتنی بر تولید ستون و روش مبتنی بر الگوریتم ژنتیک :

در پیاده‌سازی عدد صحیح روش مبتنی بر تولید ستون فرآیند بهینه‌سازی و تولید ستون به ماتریس ورودی اولیه بستگی دارد به‌گونه‌ای‌که بعضاً با تغییرات جزئی در ماتریس ورودی اولیه پاسخ بهتر و یا بدتر می‌شود. برای مثال در مسئله نجاری با استفاده از روش مبتنی بر تولید ستون به ازای دو ماتریس ورودی اولیه مختلف، پاسخ‌ها به شرح جدول ؟؟ است.

همانطور که در جدول ؟؟ مشاهده می‌شود تفاوت قابل توجهی در خروجی مسئله نجاری با استفاده از روش مبتنی بر

جدول ۴-۶: نتایج حل مسئله نجاری با استفاده از روش مبتنی بر تولید ستون و دو ماتریس ورودی اولیه مختلف
.

	مسئله اول	مسئله دوم
ماتریس B_{init}	$\mathbf{B}_i = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\mathbf{B}_i = \begin{bmatrix} 2 & 3 & 0 \\ 2 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$
ماتریس B_{final}	$\mathbf{B}_f = \begin{bmatrix} 5 & 0 & 0 & 1 \\ 0 & 3 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$	$\mathbf{B}_f = \begin{bmatrix} 2 & 3 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix}$
تعداد استفاده شده از هر الگو	۱۵ ، ۰ ، ۲ ، ۲	۱۳ ، ۰ ، ۰ ، ۲ ، ۳
ضایعات برش	۲۰۰	۲۵

تولید ستون به ازای دو ماتریس ورودی اولیه مختلف، وجود دارد. این امر بیانگر اهمیت انتخاب ماتریس اولیه در روش مبتنی بر تولید ستون است. از طرفی امکان بررسی تمام ماتریس‌های شدنی اولیه که در محدودیت‌های مسئله صدق کند وجود ندارد بنابراین به عنوان راه‌حل ترکیب روش مبتنی بر تولید ستون با روشی مبتنی بر الگوریتم ژنتیک را مطرح می‌کنیم. چراکه الگوریتم ژنتیک با تولید فرزندان از والدینی که ضایعات کمتری دارند نسلی را تولید می‌کند که ضایعات تولیدشده و تعداد الگوی مصرفی را کاهش دهند. بنابراین برای یافتن حالت بهینه دیگر نیاز به تولید و بررسی تمامی حالت‌ها نیست. در ادامه ترکیب روش مبتنی بر الگوریتم ژنتیک با روش مبتنی بر تولید ستون را برای مسئله نجاری شرح می‌دهیم.

گام اول : داده‌های اولیه مسئله

در این مرحله داده‌های مسئله اولیه را تعیین می‌کنیم. این داده‌ها شامل طول موجودی، طول و تعداد میله‌های سفارش داده شده، تعداد تکرار روش مبتنی بر الگوریتم ژنتیک و تعداد فرزندان تولید شده در هر مرحله هستند.

گام دوم : تولید دو ماتریس والد برای شروع مسئله

برای اینکه ماتریس‌های تولید شده در مسئله صدق کنند باید شروط زیر را دارا باشند:

۱. هر ستون ماتریس اولیه، به عنوان یک الگوی برش معرفی می‌شود. از آنجایی که در این روش نیازی به شناسایی تمامی الگوها در ابتدا ندارد، بنابراین میتوانیم الگوهای دلخواهی رابرای شروع به عنوان ماتریس اولیه در نظر بگیریم.

۲. هر سطر ماتریس اولیه تعداد سفارشات تولید شده توسط هر الگو است.

۳. ماتریس اولیه، نباید سطر و یا ستون تمام صفر داشته باشد : زیرا در این صورت سفارشات متناظر با سطر صفر هرگز تولید نمی‌شوند بنابراین ورودی نشدنی است و نیز الگوی متناظر با ستون صفر هرگز در تولید سفارشات اثری نخواهد داشت.

۴. حاصل جمع سفارشات تولید شده در هر الگو نباید از طول موجودی بیشتر شود.

- ماتریس اولیه، نباید ستون تکراری داشته باشد : زیرا ستون تکراری در روند حل مسئله باعث اضافه بار پردازشی و اتلاف زمان می‌گردد.

به منظور دستیابی به شرط‌های فوق دو رابطه برای تولید دو ماتریس اولیه معرفی می‌کنیم:

اول : یک ماتریس $n \times n$ صفر در نظر می‌گیریم که n در آن برابر با تعداد نوع سفارشات است. سپس قطر اصلی این ماتریس با حاصل تقسیم صحیح طول میله مرجع به طول هر سفارش جایگزین می‌شود، به عبارت دیگر داریم:

$$\mathbf{B} = \begin{bmatrix} \lfloor \frac{L}{Li} \rfloor & ۰ & ۰ \\ ۰ & \lfloor \frac{L}{Li} \rfloor & ۰ \\ ۰ & ۰ & \lfloor \frac{L}{Li} \rfloor \end{bmatrix}$$

به این ترتیب یک جواب شدنی برای مسئله تولید می‌شود. برای مسئله نجاری مطابق با تعاریف بالا داریم:

$$\mathbf{B} = \begin{bmatrix} \lfloor \frac{۴۲۵}{۷۵} \rfloor & ۰ & ۰ \\ ۰ & \lfloor \frac{۴۲۵}{۱۲۵} \rfloor & ۰ \\ ۰ & ۰ & \lfloor \frac{۴۲۵}{۲۲۵} \rfloor \end{bmatrix}$$

که می‌شود

$$\mathbf{B} = \begin{bmatrix} ۵ & ۰ & ۰ \\ ۰ & ۳ & ۰ \\ ۰ & ۰ & ۱ \end{bmatrix}$$

دوم : برای اینکه یک والد دیگر هم داشته باشیم می‌توانیم از ماتریس همانی $n \times n$ استفاده کنیم.

$$\mathbf{B}_{sec} = \begin{bmatrix} ۱ & ۰ & ۰ \\ ۰ & ۱ & ۰ \\ ۰ & ۰ & ۱ \end{bmatrix}$$

گام سوم : تولید فرزندان:

در این پژوهش ما برای تولید تعداد مشخصی فرزند تابعی [۱] نوشتیم که در آن B و B_{sec} به عنوان دو والد اولیه به آن داده می شود. فراخوانی می‌شود.

گام سوم - بخش ۱ : هم اندازه کردن دو ماتریس

---

[۱] پیوست؟؟

B و B_{sec} از آنجایی که تمایل داریم فرزندانی که منجر به پاسخ‌های بهتر می‌شوند را به عنوان والدین تکرار بعدی نگه داریم، ممکن است در طول فرآیند حل مسئله اندازه ماتریس‌های خروجی متفاوت شود. زیرا در روش ارائه شده برخلاف روش تولید ستون که ستون جدید را جایگزین یکی از ستون های B می‌کند، در صورتی که الگوی جدیدی بوجود آید، آن را به انتهای B اضافه می‌کنیم و اگر هم الگویی استفاده نشود آن را از B حذف می‌نماییم. در تکرار بعدی فرض می‌کنبم مسئله جدیدی را می‌خواهیم حل نماییم که ماتریس اولیه آن یک الگو بیشتر(و یا کمتر) از مسئله تکرار قبلی دارد. بنابراین مزیت این روش نسبت به روش تولید ستون این است که تعداد الگوها در ماتریس B قابل تغییر است. از آنجایی که مسئله برای تمام فرزندان تولید شده در هر مرحله تکرار می‌شود، و احتمال تغییر اندازه در ماتریس‌های خروجی وجود دارد ممکن است، دو ماتریس منتخب به‌عنوان والدین تکرار بعدی هم اندازه نباشند. از این‌رو می‌توان از بردار صفر برای هم اندازه کردن دو ماتریس استفاده نمود.

گام سوم - بخش ۲ : برش

اگر بخواهیم روی دو بردار برش داشته باشیم راه‌کارهای متنوعی نظیر برش تک نقطه ای وجود دارد. همانطور که در بخش ؟؟ برش تک نقطه‌ای شرح داده شده است، در حالت ماتریسی نیز می‌توان از آن استفاده کرد،البته با این تفاوت که در ماتریس بجای نقطه باید یک ستون در نظر گرفته شود. به‌این‌منظور ابتدا یک عدد تصادفی (بین ۱ تا تعداد ستون های ماتریس) انتخاب می‌شود سپس ستون متناظر با عدد تصادفی و ستون‌های سمت راست ستون مورد نظر در دو ماتریس با هم جابجا می‌شوند. با فرض اینکه عدد تصادفی ۲ باشد بعد از عمل برش تک نقطه‌ای خواهیم داشت:

$$\mathbf{B} = \begin{bmatrix} ۵ & ۰ & ۰ \\ ۰ & ۱ & ۰ \\ ۰ & ۰ & ۱ \end{bmatrix} \tag{۲۰-۴}$$

$$\mathbf{B}_{sec} = \begin{bmatrix} ۱ & ۰ & ۰ \\ ۰ & ۳ & ۰ \\ ۰ & ۰ & ۱ \end{bmatrix} \tag{۲۱-۴}$$

گام سوم - بخش ۳ : جهش

مطابق آنچه در بخش ؟؟ شرح داده شده است، جهشی که در این روش در نظر گرفته‌ایم به شرح زیر است:

۱. ابتدا از هریک از ماتریس‌های ؟؟ و ؟؟ یک عنصر به‌صورت تصادفی انتخاب می‌شود که به آن $rand$ می‌گوییم.

۲. باتوجه به نحوه ذخیره‌سازی ماتریس‌ها، مختصات$rand$ (سطر i و ستون j) را به‌دست می‌آوریم.

۳. هر عنصر متعلق به یک ستون و هر ستون یک الگوی برش است، بنابراین مقدار موجودی که توسط این الگو برش

داده شده است، برابر با حاصل‌ضرب الگوی برش انتخاب شده در بردار سفارشات (l_i) است. باتوجه به رابطه ؟؟ مقدار باقیمانده الگوی مشخص شده را به‌دست می‌آوریم و آن را C_{rand} می‌نامیم.

۴. حداکثر مقداری که می‌توان از سفارش انتخاب شده توسط C_{rand} تولید کرد، ازطریق تقسیم مقدار C_{rand} به طول سفارشی که توسط سطر $rand$ مشخص شده است و آن را l_{rand} مینامیم. محاسبه می‌شود : $\left\lfloor \frac{C_{rand}}{l_{rand}} \right\rfloor$

۵. برای اینکه مقدار مناسبی برای جهش عنصر انتخاب شده بدست آید، مقدار آن عنصر با مقدار $\left\lfloor \frac{C_{rand}}{l_{rand}} \right\rfloor$ جمع می‌شود و سپس یک مقدار از صفر تا حاصل‌جمع، به صورت تصادفی انتخاب شده و به آن اختصاص می‌یابد.

۶. این عمل به ترتیب برای تمام برش‌های تعریف شده در الگوی انتخاب شده تا زمانی که دیگر سطری برای بررسی وجود نداشته باشد، تکرار و در هر مرحله میزان باقیمانده مجددا محاسبه می‌شود .

در پایان مرحله پنجم اولین فرزند تولید می‌شود. برای نمونه یکی از فرزندان تولید شده توسط پنج مرحله شرح داده شده برابر است با :

$$\mathbf{B} = \begin{bmatrix} ۱ & ۰ & ۱ \\ ۰ & ۳ & ۰ \\ ۰ & ۰ & ۱ \end{bmatrix}$$

تمام مراحل بالا عیناً برای والد B_{sec} تکرار می‌شود.

گام چهارم :حل مسئله تولید ستون با استفاده از فرزندان تولید شده

در طی گام‌های قبل، با استفاده از روش مبتنی بر الگوریتم ژنتیک ماتریس‌های شدنی به عنوان ماتریس اولیه برای روش مبتنی بر تولید ستون ایجاد شدند. حال مطابق آنچه در روش مبتنی بر تولید ستون که در بخش ؟؟ شرح داده شده است اقدام به حل مسئله می‌کنیم. نتایج به‌دست آمده براساس سه شرط زیر (هزینه‌های تولید) با هم مقایسه و دو ماتریس که منجر به بهینه‌ترین پاسخ می‌شوند به عنوان والدین مرحله بعد ذخیره می‌گردند.

شرط اول : میزان ضایعات تولید شده، کمتر باشد.

شرط دوم : تعداد الگوها، کمتر باشد.

شرط سوم : مقدار موجودی مصرف شده، کمتر باشد.

نکته مهم این است که اگر نسبت اهمیت هزینه‌های تولید به یکدیگر، درنظر گرفته نشود، نتایج حاصل ممکن است به دلیل وجود ترتیب در برآوردن ۳ شرط فوق، از جواب بهینه دور شوند. بنابراین اگر در مسئله کاهش ضایعات برش، هزینه‌های تولید با یکدیگر جمع شوند، آنگاه پاسخ‌های به دست آمده کمترین هزینه تولید را نشان می‌دهند، بنابراین به هرکدام از هزینه‌های ذکر شده یک مقدار جریمه اختصاص داده می‌شود. این مقدار با توجه به صنعتی که مسئله برای آن تعریف شده است می‌تواند متفاوت باشد زیرا ممکن است در یک صنعت هزینه تعویض الگوی برش بیش از هزینه یک یا چند واحد از موجودی‌ها باشد ازاین‌رو میزان جریمه در نظر گرفته شده برای هزینه تعویض از جریمه موجودی مصرفی بیشتر خواهد بود.

درنهایت حاصل‌جمع هزینه‌ها با احتساب جریمه اختصاص یافته به آن‌ها محاسبه و با هم مقایسه می‌شوند. کمترین حاصل بهینه‌ترین پاسخ است.

گام پنجم : بررسی شرط خاتمه الگوریتم. تعداد تکرار الگوریتم مقداری ثابت و از پیش تعریف شده است.

گام ششم : حل مسئله با الگوهای بهینه (ماتریس B نهایی که شامل الگوهایی است که با آن کمترین ضایعات را داریم)

در این مرحله شرط خاتمه برقرار شده است و پاسخ روش مبتنی بر تولید ستون بهینه‌ترین الگوهایی است که با تعداد تکرار مشخص قابل دستیابی بوده است. در این مرحله با داشتن ماتریس الگوهای نهایی و با برنامه‌ریزی‌خطی در متلب مسئله اولیه را حل می‌شود.

نتایج حاصل از حل مسئله بخش ؟؟ با استفاده از ترکیب روش مبتنی بر تولید ستون و روش مبتنی بر الگوریتم ژنتیک بدون لحاظ جریمه در ارزش هزینه های تولید در جدول ؟؟ نشان داده شده است.

جدول ۴-۷: نتایج حل مسئله نجاری با استفاده از ترکیب روش مبتنی بر تولید ستون و روش مبتنی بر الگوریتم ژنتیک .

تعداد میله‌های مصرف شده	ضایعات برش	تعداد الگوهای استفاده شده	الگوی های برش
۱۹	۵۰	$x_۱ = ۳, x_۲ = ۱۵, x_۳ = ۱$	$\mathbf{B}_{final} = \begin{bmatrix} ۴ & ۱ & ۰ \\ ۱ & ۱ & ۳ \\ ۰ & ۱ & ۰ \end{bmatrix}$

همانطورکه در جدول ؟؟ مشاهده می‌شود در این روش از ۳ الگو استفاده شده است. هم‌چنین با استفاده از ۱۹ میله ۴۲۵ سانتی‌متری، ۲۷ میله ۷۵ سانتی‌متری،۲۱ میله ۱۲۵ سانتی‌متری و همچنین ۱۵ میله ۲۲۵ سانتی‌متری تولید شده است و فقط ۵۰ سانتی‌متر ضایعات تولید شده است.

حال اگر برای هزینه‌های تولید جریمه های ۱ در نظر گرفته شود، خواهیم داشت:

هزینه تولید =

(جریمه × ارزش ضایعات تولید شده) + (جریمه × موجودی استفاده شده) + (جریمه × تعداد الگوی تعریف شده)

$$\sum C_i \times p_{C_i} + \sum l_i \times C_p + \sum Yi \times C_s \qquad (۲۲-۴)$$

$$(۱۹ \times ۱) + (۵۰ \times ۱) + (۳ \times ۱) = ۷۲$$

حال مسئله را با در نظر گرفتن هزینه تولید حل می‌کنیم. نتایج حاصل در جدول ؟؟ نشان داده شده است.

با توجه به رابطه ؟؟ داریم :

$$(۲۱ * ۱) + (۰ * ۱) + (۲ * ۱) = ۲۳$$

جدول ۴-۸: نتایج حل مسئله با روش مبتنی بر تولید ستون و روش مبتنی بر الگوریتم ژنتیک با احتساب هزینه تولید .

تعداد میله‌های مصرف شده	ضایعات برش	تعداد الگوهای مصرف شده	الگوی های برش
۲۱	۰	$x_۱ = ۱۸, x_۲ = ۳$	$\mathbf{B}_{final} = \begin{bmatrix} ۱ & ۴ \\ ۱ & ۱ \\ ۱ & ۰ \end{bmatrix}$

بنابراین علی‌رغم اینکه ۲ واحد بیش‌تر از مقدار موجودی‌ها مصرف شده است ولی به دلیل کاهش ضایعات و نیز کاهش تعداد الگوها، نتیجه به دست آمده بهبود یافته است. به عبارت دیگر میزان هزینه تولید به مقدار ۶۸ درصد کاهش یافته است.

بمنظور ارزیابی عملکرد این روش داده های استفاده شده در مقاله [؟] با فرضیات آورده شده در شکل ؟؟ مورد آزمایش قرار گرفت. نتایج حاصل در جدول ؟؟ آورده شده است هم‌چنین راه‌حل بدست آورده شده توسط مقاله [؟] در شکل ؟؟ آورده شده است. با مقایسه نتایج مشاهده می‌شود، در هر دو روش با استفاده از ۱۷ واحد موجودی سفارشات تولید شده است اما روش ارائه شده در بخش ؟؟ نسبت به پاسخ ارائه شده در مقاله [؟] دارای باقیمانده کمتری (باقیمانده = صفر) است و هم‌چنین ۶ الگو کمتر مصرف نموده است که در هزینه تولید بسیار اثرگذار خواهد بود.

$Type_i$	*Width*	*Demand Quantity*	*Available Stock Length*
1	435	5	1900
2	415	11	
3	385	13	
4	365	8	
5	335	3	
6	340	8	
7	320	7	
8	300	4	
9	260	6	

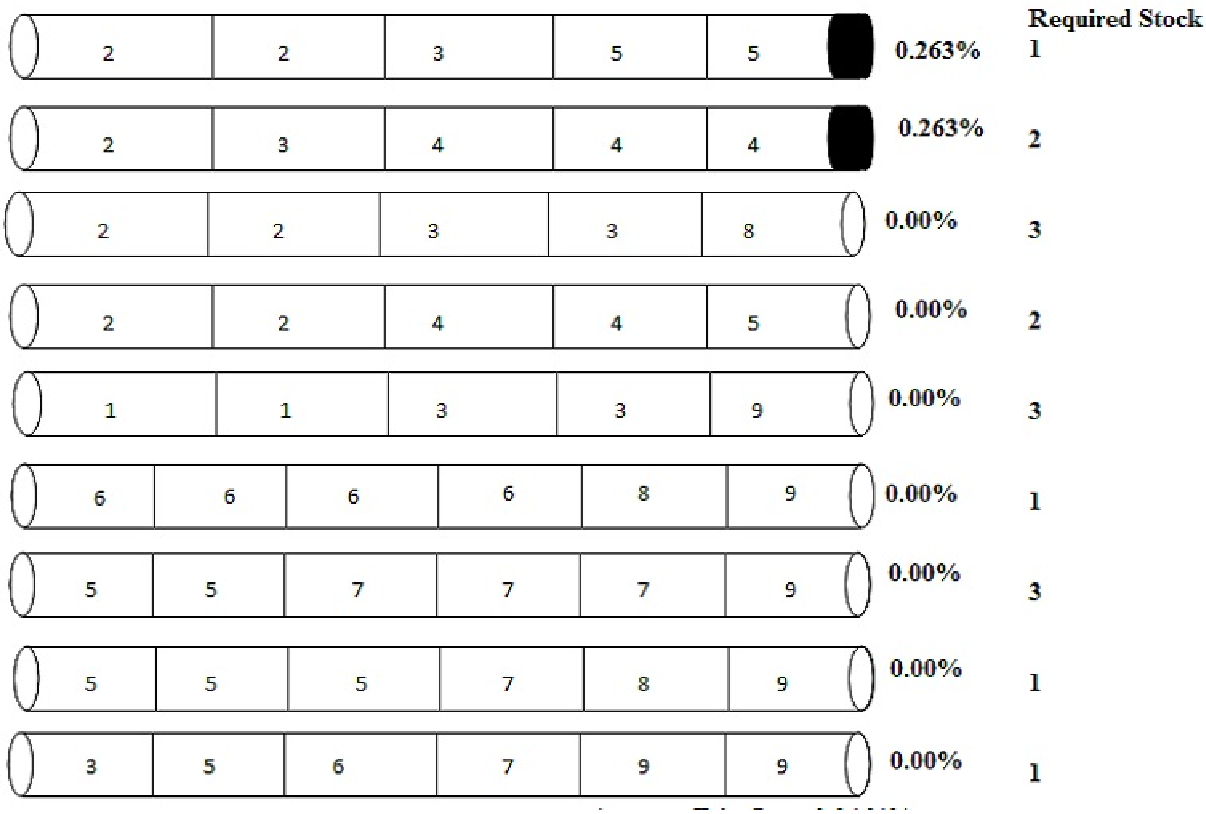
شکل ۴-۱: فرضیات مسئله در مقاله [؟]

۴-۴-۶ حل مسئله نجاری با استفاده از روش مبتنی بر الگوریتم ژنتیک

۵ مرحله ابتدایی روش حل مسئله نجاری با استفاده از الگوریتم ژنتیک مشابه روش حل با استفاده از ترکیب روش مبتنی بر تولید ستون و روش مبتنی بر الگوریتم ژنتیک است. زیرا در این ۵ مرحله به تولید والدین اولیه و فرزندان مورد نیاز از آنها پرداخته می‌شود. اما در مرحله ششم مقدار میله‌های برش داده شده و نیز تعداد سفارشات تولید شده برای هر فرزند محاسبه می‌شود. در هر مرحله مقدار میله برش داده شده و نیز ضایعات بوجود آمده توسط هر الگو (فرزند) با مقدار کمینه موجود (که در ابتدا عددی بزرگ فرض شده است) مقایسه می‌شود. درصورتی‌که مقدار ضایعات و تعداد الگوی مصرف شده در هر

جدول ۴-۹: نتایج حل مسئله مقاله [؟] با استفاده از ترکیب روش مبتنی بر تولید ستون و روش مبتنی بر الگوریتم ژنتیک با احتساب هزینه تولید .

الگوی های برش	تعداد الگوهای استفاده شده	ضایعات برش	تعداد میله‌های مصرف شده
$\mathbf{B}_{final} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 3 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 2 & 0 & 0 \end{bmatrix}$	$x_1 = 8, x_2 = 1, x_3 = 8$	۰	۱۷



شکل ۴-۲: پاسخ ارائه شده در مقاله [؟]

تکرار برای هر فرزند از مقدار کمینه کمتر باشد، به عنوان کمینه در نظر گرفته می‌شود. این مراحل تا برقرار شدن شرط خاتمه تکرار می‌گردد. شرط خاتمه در این روش این است که تمامی فرزندان بررسی شوند یا میزان کمینه مورد نظر حاصل شود.

با استفاده از جدول ۱-۷ مشخص می‌شود که با ۱۹ میله مرجع (۴۲۵ سانتی‌متری) توانسته ایم ۲۶ میله ۷۵ سانتی‌متری ، ۲۱ میله ۱۲۵ سانتی‌متری و همچنین ۱۵ میله ۲۲۵ سانتی‌متری را با ضایعاتی به مقدار ۱۲۵ سانتی‌متری تولید نماییم .

جدول ۴-۱۰: نتایج حل مسئله با استفاده از روش مبتنی بر تولید ستون و روش مبتنی بر الگوریتم ژنتیک .

موجودی مصرف شده	ضایعات برش	تعداد الگوهای استفاده شده	الگوی های برش
۱۹	۱۲۵	$x_1 = 0, x_2 = 15, x_3 = 3, x_4 = 1$	$\mathbf{B}_{final} = \begin{bmatrix} 0 & 1 & 2 & 5 \\ 0 & 1 & 2 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$

جدول ۴-۱۱: مقایسه روش‌های ارائه شده در این فصل برای یک مسئله نجاری

روش حل	میزان مصرف موجودی	تعداد الگو	میزان ضایعات
روش اول : الگوریتم ابتکاری	۲۰	۲	۰
روش دوم : حل مسئله نجاری با استفاده از روش تولید ستون و برنامه ریزی پویا	۱۹	۳	۵۰
روش سوم : روش تولید ستون و روش حریصانه	۱۹	۳	۲۰۰
روش چهارم : روش مبتنی بر تولید ستون و ترکیب با روش ابتکاری	۱۹	۳	۱۲۵
روش پنجم : روش مبتنی بر الگوریتم ژنتیک	۱۹	۴	۱۲۵
روش ششم : روش میتنی بر تولید ستون و ترکیب با روش مبتنی بر الگوریتم ژنتیک بدون جریمه	۱۹	۳	۵۰
روش هفتم : روش مبتنی بر تولید ستون و ترکیب با روش مبتنی بر الگوریتم ژنتیک با جریمه	۲۱	۲	۰

به منظور ارزیابی دقیق‌تر ترکیب روش مبتنی بر تولید ستون با روش مبتنی بر الگوریتم ژنتیک عملکرد این روش برای یک نمونه مسئله با مقیاس بزرگ مورد بررسی قرار گرفته است.

مثال ۴-۴-۱. فرضیات مساله در جدول ؟؟ آورده شده است. شرح این مسئله مشابه مسئله نجاری است که تنوع سفارشات در آن افزایش یافته است.همچنین با استفاده از اندازه‌های مختلف موجودی، تاثیر افزایش مقدار آن در کاهش تعداد الگوها و ضایعات مورد مطالعه قرار گرفته است.

نتایج حاصل در جدول ؟؟ و ؟؟ نمایش داده شده است. همانطورکه مشاهده می‌شود با افزایش مقدار موجودی، زمان پردازش، مجموع ضایعات و موجودی‌های مصرف شده کاهش قابل ملاحظه‌ای می‌یابند. همچنین مشاهده می‌شود اختصاص جریمه برای ضایعات و الگوهای مصرفی نتایج را بهبود می‌بخشد.

جدول ۴-۱۲: فرضیات مساله مقیاس بزرگ

شماره	طول میله‌های سفارش داده شده	تعداد مورد تقاضا
۱	۵	۲۰۰
۲	۱۰	۱۴۰
۳	۲۵	۵۸
۴	۳۰	۹۱
۵	۹	۳۰۰
۶	۱۱	۲۵۰
۷	۲	۱۹۰
۸	۶	۷۵
۹	۳۲	۸۰
۱۰	۱۸	۱۹۵
۱۱	۴	۲۴۰
۱۲	۳۹	۳۱۰
۱۳	۲۶	۳۲۰
۱۴	۵۱	۱۶۱
۱۵	۱۲	۱۵۰
۱۶	۱۴	۲۹۰
۱۷	۳۱	۳۵۰
۱۸	۲۷	۷۰
۱۹	۴۳	۸۰
۲۰	۵۷	۱۲۵
۲۱	۱۶	۲۲۵
۲۲	۴۰	۷۵
۲۳	۲۳	۱۵۴
۲۴	۱۹	۲۳۰
۲۵	۳۸	۲۶۰
۲۶	۴۴	۳۳۰

۴-۴-۷ نتیجه گیری

در این پژوهش تلاش شد در حل مسئله کاهش ضایعات برش، علاوه بر کاهش ضایعات تعداد الگوهای مورد استفاده (تعداد منابع استفاده شده) نیز کاهش یابند. هم‌چنین میزان زمان مصرف شده برای پردازش به عنوان یک محدودیت در نظر گرفته نشده است و فرض کردیم که ارزش کالاهای تولید شده با یکدیگر مساوی هستند. از این‌رو به‌دلیل زیاد بودن تعداد متغیرها در سطر تابع هدف نسبت به محدودیت‌های مسئله از روش تولید ستون به عنوان یک راه حل پایه برای مسائل برش موجودی

جدول ۴-۱۳: مقایسه تاثیر تغییر مقدار موجودی در حل مسئله مقیاس بزرگ، با ترکیب روش مبتنی بر تولید ستون و الگوریتم ژنتیک با جریمه

طول موجودی	تعداد الگوهای استفاده شده	تعداد میله‌های مصرف شده	میزان ضایعات	زمان پردازش
۸۰	۱۱	۱۷۴۳	۵۱۲	۴۸۳.۸
۱۰۰	۱۱	۱۶۴۴	۵۸۰	۵۰۸.۸
۱۵۰	۷	۱۰۰۷	۰	۶۸۴.۲
۲۰۰	۷	۷۷۴	۰	۴۱۹.۸
۳۰۰	۷	۴۹۳	۱۲۵	۶۱۲.۹
۵۰۰	۵	۳۴۳	۰	۶۸۶.۹
۱۰۰۰	۵	۲۵۴	۴	۶۸۶.۶

جدول ۴-۱۴: مقایسه تاثیر تغییر مقدار موجودی در حل مسئله مقیاس بزرگ، با ترکیب روش مبتنی بر تولید ستون و الگوریتم ژنتیک بدون جریمه

طول موجودی	تعداد الگوهای استفاده شده	تعداد میله‌های مصرف شده	میزان ضایعات	زمان پردازش
۸۰	۲۰	۱۵۳۱	۱۸۷۶	۵۰۷.۷
۱۰۰	۱۸	۱۲۸۷	۶۴۸	۵۷۰.۵
۱۵۰	۱۷	۸۲۵	۱۰۱۶	۳۴۸.۸
۲۰۰	۲۰	۶۱۲	۸۹۳	۳۵۵.۳
۳۰۰	۱۵	۴۶۱	۴۲۸	۳۶۰.۶
۵۰۰	۱۴	۳۰۲	۶۵۹	۳۲۶.۲
۱۰۰۰	۱۷	۱۳۷	۵۹۷	۵۳۶.۰

استفاده کردیم.

از آنجایی که روش تولید ستون از جمله روش‌های تجزیه محسوب می‌شود، به دلیل کاهش پردازش کلیه ابعاد مسئله از سرعت بالاتری نسبت به روش‌های معمول برخوردار است؛ اما روش تولید ستون دارای این اشکال است که با تغییر متغییرهای پایه در سطر صفر (در مسائل با ابعاد بزرگ) پاسخ نهایی (جواب بهینه) می‌تواند تغییر نماید. دلیل این تغییر پاسخ‌های متفاوت موجود در حل زیر مسئله در این روش است. همانطور که در فصل سوم اشاره شد، زیر مسئله تولید شده یک مسئله کوله پشتی است که از حل آن متغیری که ستون نظیر آن قادر به وارد شدن به پایه است، قابل شناسایی است. نکته قابل توجه این است که در یک مسئله کوله پشتی هم‌زمان می‌توان بیش از یک راه حل یافت که مقدار تابع هدف کوله را بهینه نمایند. بنابراین باتوجه به اینکه الگوهای ایجاد شده از پاسخ‌های زیر مسئله، در ادامه فرآیند روش تولیدستون تاثیرگذار است، لذا می‌تواند منجر به تغییر جواب بهینه گردد. لازم به ذکر است که به دلیل ماهیت مسئله کوله پشتی، تولید تمامی حالات آن از درجه نمایی است و به سادگی امکان‌پذیر نخواهد بود.

تغییر متغیرهای پایه در سطر صفر منجر به تغییر در زیر مسئله و نتایج حاصل از آن در طی فرآیند حل به روش تولید ستون می‌گردد. بنابراین ما تلاش نمودیم با استفاده از روشی مبتنی بر الگوریتم ژنتیک حالت‌های مختلفی از متغیرهای پایه

و ستون‌های نظیر آن را تولید کنیم تا در طی تکرارهای متوالی مسئله به سمت جواب بهینه هدایت شود.

در روش مبتنی بر تولید ستون ارائه شده در این پژوهش تلاش گردید تا این محدودیت برداشته شود؛ بنابراین در پاسخ‌های به دست آمده از این روش شاهد کاهش تعداد الگوهای مورد استفاده در برش با رعایت کاهش ضایعات هستیم. نتایج به دست آمده از روش های شرح داده شده در این فصل، در جدول ؟؟ با هم مقایسه شده‌اند. همانطورکه مشاهده می‌شود کمترین میزان ضایعات مربوط به روش ؟؟ و روش ؟؟ با احتساب جریمه است. باید توجه داشت که روش اول تنها در مواردی کاربرد دارد که هدف یافتن بهترین الگوها (که کمترین میزان ضایعات را تحمیل می‌کنند) از بین الگوهای ازپیش تعیین شده است و این امر در مسائل بزرگ مقیاس که به راحتی نمی‌توان تمامی الگوهای تقریبا بهینه را به دست آورد کارایی ندارد. لذا بهترین روش از بین روش‌های اشاره شده در جدول ؟؟ ترکیب روش مبتنی بر تولید ستون با روش مبتنی بر الگوریتم ژنتیک است زیرا در این روش میزان ضایعات تحمیل شده از همه کمتر بوده و نیز تعداد منبع استفاده شده در آن کمتر مساوی سایر روش‌ها است. هم‌چنین در صورت لحاظ جریمه برای هزینه‌ها در روش ترکیب روش مبتنی بر تولید ستون با روش مبتنی بر الگوریتم ژنتیک در مسئله با سفارشات زیاد نیز، هزینه تولید به میزان قابل توجهی کاهش می‌یابد .

فهرست منابع

[1] Kazunga, C, Mutambara, LHN, and Mapurisa, J. A column generation approach to a carpentry cutting stock problem: a case study for planks cutting in zimbabwe. *African Journal of Educational Studies in Mathematics and Sciences*, 9(1):49–59, 2011.

[2] Haessler, Robert W and Sweeney, Paul E. Cutting stock problems and solution procedures. *European Journal of Operational Research*, 54(2):141–150, 1991.

[3] Thomas, Jaya and Chaudhari, Narendra S. An integrated genetic algorithm approach to 1d-cutting stock problem. *International Journal of Operational Research*, 27(1-2):23–46, 2016.

[4] Cui, Yaodong. A cutting stock problem and its solution in the manufacturing industry of large electric generators. *Computers & operations research*, 32(7):1709–1721, 2005.

[5] Dyckhoff, Harald. A typology of cutting and packing problems. *European Journal of Operational Research*, 44(2):145–159, 1990.

[6] Andrew, Alex M. Introduction to evolutionary computing, by ae eiben and je smith (natural computing series), springer, berlin, 2003, hardback, xv+ 299 pp., isbn 3-540-40184-9 (£ 30.00), 2004.

[۷] مصلحي، قاسم و رضايي، عليرضا. ارايه الگوريتمي براي مسئله برش دوبعدي با تقاضا. استقلال، ۲۳(۲):۵۹-۷۵، اسفند ۱۳۸۳.

[8] Alp, Seda, Ertek, Gurdal, and Birbil, S Ilker. Application of the cutting stock problem to a construction company: a case study. in *International Symposium on Intelligent Manufacturing Systems, 5to, Sakarya, Turquia*, 2006.

[9] Morabito, Reinaldo and Garcia, Valdir. The cutting stock problem in a hardboard industry: A case study. *Computers & Operations Research*, 25(6):469–485, 1998.

[10] Hinxman, AI. The trim-loss and assortment problems: A survey. *European Journal of Operational Research*, 5(1):8–18, 1980.

[11] De Cani, Phillip. A note on the two-dimensional rectangular cutting-stock problem. *Journal of the Operational Research Society*, 29(7):703–706, 1978.

[12] Kantorovich, Leonid V. Mathematical methods of organizing and planning production. *Management Science*, 6(4):366–422, 1960.

[13] Gilmore, PC and Gomory, RE. The theory and computation of knapsack functions. *Operations Research*, 14(6):1045–1074, 1966.

[۱۴] جوانشیر، حسن، فرد، محمد تقی تقوی، و اسلامی، فروغ. الگوریتمی کارا به منظور مدیریت سرمایه و هزینه در صنایع برش. استقلال، ۲۱(۱):۹۳-۱۰۴، بهار ۱۳۸۹.

[15] Gilmore, Paul C and Gomory, Ralph E. A linear programming approach to the cutting stock problem-part ii. *Operations research*, 11(6):863–888, 1963.

[16] Mobasher, Azadeh and Ekici, Ali. Solution approaches for the cutting stock problem with setup cost. *Computers & operations research*, 40(1):225–235, 2013.

[17] Gilmore, PC and Gomory, Ralph E. Multistage cutting stock problems of two and more dimensions. *Operations research*, 13(1):94–120, 1965.

[18] Dov, AG BEGED. Some computational aspects of the m paper mills and p printers paper trim problem. *J. Bus. Admin*, 1:15–34, 1970.

[19] Michael R. Garey, David S. Johnson. *COMPUTERS AND INTRACTABILITY A Guide to the Theory of NP-Completeness*. BELL LABORATORIES MURRAY HILL, NEW JERSEY, 2006.

[۲۰] سلجوقی، سپیده. مروری بر الگوریتمهای فراابتکاری با ارائه یک روش دسته۲بندی. اولین کنفرانس ملی الگوریتم های فراابتکاری و کاربردهای آن در علوم و مهندسی، مازندران، (۱)، مرداد ۱۳۹۳.

[21] Dorigo, M and Stutzle, T. Ant colony optimization‖ mit press. *Cambridge, MA*, 2004.

[۲۲] مدانلوجویباری، علیرضا و حمیدرضا محمدپور. مروری بر الگوریتمهای فراابتکاری و بررسی قابلیتهای آنها. اولین کنفرانس ملی الگوریتم های فراابتکاری و کاربردهای آن در علوم و مهندسی، مازندران، (۱)، مرداد ۱۳۹۳.

[23] Dréo, Johann, Pétrowski, Alain, Siarry, Patrick, and Taillard, Eric. *Metaheuristics for hard optimization: methods and case studies*. Springer Science & Business Media, 2006.

[24] Fogel, David B. What is evolutionary computation? *IEEE Spectrum*, 37(2):26–28, 2000.

[۲۵] رضائی، علی رضا و سجاد، رنجبران. آموزش کاربردي الگوریتم ژنتیك در نرم افزار *MATLAB*. تهران، آذر ۱۳۸۶.

[26] Mitchell, Melanie. *An introduction to genetic algorithms*. MIT press, 1998.

[27] Goldberg, DE. Genetic algorithms in search, optimization, and machine learning, addison-wesley, reading, ma, 1989. *K. Ohno, K. Esfarjani, and Y Kawazoe: Computational Materi*.

[28] Gonçalves, José Fernando. A hybrid genetic algorithm-heuristic for a two-dimensional orthogonal packing problem. *European Journal of Operational Research*, 183(3):1212–1229, 2007.

[29] Mitchell, Melanie. Genetic algorithms: An overview. *Complexity*, 1(1):31–39, 1995.

[30] Shahin, Adham A and Salem, Ossama M. Using genetic algorithms in solving the one-dimensional cutting stock problem in the construction industry. *Canadian journal of civil engineering*, 31(2):321–332, 2004.

[31] Gracia, Carlos, Andrés, Carlos, and Gracia, Luis. A hybrid approach based on genetic algorithms to solve the problem of cutting structural beams in a metalwork company. *Journal of Heuristics*, 19(2):253–273, 2013.

[32] Brandão, Julliany Sales, Coelho, Alessandra Martins, do Carmo, Felipe, and Vasconcelos, João Flávio. Study of different setup costs in singlega to solve a one-dimensional cutting stock problem. *GSTF Journal on Computing (JoC)*, 2(1), 2014.

[۳۳] آریانژاد، میربهادرقلی و سجادی، سیدجعفر. برنامه‌ریزی خطی. انتشارات دانشگاه علم و صنعت ایران، ۱۳۸۵.

[34] Winston, Wayne L and Goldberg, Jeffrey B. *Operations research: applications and algorithms*, vol. 3. Duxbury press Boston, 2004.

[35] Taha, Hamdy A. *Operations research: an introduction*. Macmillan,, 1992.

[36] Kushwaha, SPS, Mukhopadhyay, Suchismita, Prasad, V Hari, and Kumar, Suresh. Sustainable development planning in pathri rao sub-watershed using geospatial techniques. *1st Iranian Conference on supplying raw material and development of wood And paper industry*, 2010.

[37] Abolfathi, M. and Hamedinejad. Fluctuations in the production of timber and development of wood industry in last half century. *Current science*, pp. 1479–1486, 2008.

پیوست آ

کد های برنامه در متلب

آ-۱ کد روش حل مساله با استفاده از روش تولید ستون و روش مبتنی بر الگوریتم ژنتیک

برنامه آ-۱: کد روش حل مساله با استفاده از روش تولید ستون و روش مبتنی بر الگوریتم ژنتیک در MATLAB

```
clear
clc

global L ;
L=1900;              % Length of stock rod
global l1;
global l2;
global l3;
global l4;
global l5;
global l6;
global l7;
global l8
global l9;

% Lenght of order items
l1=435;  l2=415;  l3=385;  l4=365;   l5=335;
l6=340;  l7=320;  l8=300;  l9=260;
```

```
% number of order items
n1=5;  n2=11; n3=13; n4=8;  n5=3;
n6=8;  n7=7;  n8=4;  n9=6;

% The objective function at the first time => x1+x2+x3+..+x10
f=[ 1 ; 1 ;1 ;1 ;1 ;1 ;1 ;1 ;1];

% identity matrix is The first parent
A_Initialize=eye(length(f))
A_Initialize=[
     1     0     0     0     0     0     0     0     0
     0     1     0     0     0     0     0     0     0
     0     0     1     0     0     0     0     0     0
     0     0     0     1     0     0     0     0     0
     0     0     0     0     1     0     0     0     0
     0     0     0     0     0     1     0     0     0
     0     0     0     0     0     0     1     0     0
     0     0     0     0     0     0     0     1     0
     0     0     0     0     0     0     0     0     1
     ];

% B_Initialize = [ L/Li] is The second parent
B_Initialize=[
     4     0     0     0     0     0     0     0     0
     0     4     0     0     0     0     0     0     0
     0     0     4     0     0     0     0     0     0
     0     0     0     5     0     0     0     0     0
     0     0     0     0     5     0     0     0     0
     0     0     0     0     0     5     0     0     0
     0     0     0     0     0     0     5     0     0
     0     0     0     0     0     0     0     6     0
     0     0     0     0     0     0     0     0     7
     ]

abi=1;
for tekrar=1:80

```

```
% number of child
rept=100;
% minimum waste for starting algorithm
min_waste=10000;
% minimum number of ROD for starting algorithm
x_min=10000;
csp_s=10000000;

%************ (  child generation function  ) ************
child= Child_Generation3(rept,A_Initialize,B_Initialize);
for e=1:rept
    % The objective function at the first time => x1+x2+x3+...+X10
    f=[ 1 ; 1 ;1 ;1; 1; 1; 1; 1; 1 ];
    c=[];

    %************ (((      Column Generation     ))) ************
      % use generated children for cutting Pattern
      A_Initialize=child{e};
    [row,col]=size(A_Initialize);
    c(1:col)=1;
    lb=zeros(1,col);
    b=[n1;n2;n3;n4;n5;n6;n7;n8;n9];
    Result_D=-1;
    Db=c';

    % termination condition of c_g
    while Result_D<0

        %****** ( Colum Generation  >>> duallity ) ******
        Df=b';
        Da=A_Initialize';
        p=linprog(-Df,Da,Db,[],[],zeros[length(Da)]);
        if isempty(p)==1
            p=0;
        end
        weights =[l1,l2,l3,l4,l5,l6,l7,l8,l9];
        values = p'; % Values of items
        capacity = L;
```

```
%************(Column Generation  >>>
%            solve knapsack problem which new Column find
%            )***********

[best amount] = knapsack(weights, values, capacity);
items = find(amount);
max_subproblem=amount;      % amount is ai

%************( Column Generation  >>> insert Column
%            if termination condition of the algorithm is false
%            )************

% 1-sum(ai*p) <==> subproblem
Result_D=round(1-(max_subproblem*p),2);
if Result_D<0
    A_incr_col=horzcat(A_Initialize,max_subproblem');
    A_Initialize=A_incr_col;
    Db= vertcat(Db,[1]);
else
    % The termination condition of Column Generation algorithm is true
    lb(1:length(A_Initialize))=0;
    x=round(linprog(Db,-A_Initialize,-b,[],[],lb),2);
    x=ceil(x) ;         % number used of patterns generated
    num_rods_used = sum(x) ;     % number used of rods reference
    A_final = A_Initialize ;    % Patterns of cutting rods

    %**********(   calculation Cutting_waste   )**********
    Sum_Cut=0;
    for CW=1:length(x)

       % cutted amount
        Sum_Cut=Sum_Cut+(x(CW)*(sum([l1,l2,l3,l4,l5,l6,l7,l8,l9] * A_Initialize(:,CW))));
    end
```

```
                %cutting waste amount
               Cutting_waste(e) = (sum(x)*L)-Sum_Cut;
               if   ( Cutting_waste(e)/10) + (num_rods_used /17)+( length(find(x))/10) <= csp_s

                   % mimimum function for find best parents
                   min_waste=Cutting_waste(e);
                   x_min=x;
                   csp_s = (Cutting_waste(e)/10) +(num_rods_used /17) +(length(find(x))/10);
                   A_init_min=A_Initialize;
                   result_csp_s{abi}=csp_s;
                   Cutting_waste_comp{abi}=min_waste;
                   AB_init{abi}=A_final(:,find(x,length(find(x)))); % child{e}%A_init_min;
                   abi=abi+1;
                   num_rods_used_min=num_rods_used;
               end
           end
       end
   end
   if length(AB_init) >1  %two matrices should be same size

       A_Initialize=AB_init{end};                        %A_init_min;
       B_Initialize=AB_init{end-1};                      %B_init_min;
       diff_size=abs(length(A_Initialize) - length(B_Initialize));
       [sr,sc]=size(A_Initialize);
       temp_matrix=zeros(sr,diff_size);
       if length(A_Initialize) < length(B_Initialize)
           A_Initialize= horzcat(A_Initialize,temp_matrix);
       else
           B_Initialize= horzcat(B_Initialize,temp_matrix);
       end

   else
       A_Initialize=AB_init{end};                 %A_init_min;
       B_Initialize=AB_init{end};                 %B_init_min;
```

```
    end
end
min_waste      % minimum genereted waste
x_min'         % minimum number of used stock rod
A_init_min     % the best cutting pattern
sum(x_min)     %  minimum number of cutting pattern
```

آ-۲ برنامه تولید جمعیت

برنامه آ-۲: برنامه تولید جمعیت MATLAB

```
function A_child = Child_Generation(cnt, A_Fisrtly,A_Secondly )
global L;                                   % Length of stock rod
global l1;
global l2;
global l3;
global l4;
global l5;
global l6;
global l7;
global l8
global l9;
li=[l1 l2 l3 l4 l5 l6 l7 l8 l9 ];
A_child={};
num=0;
% ******** ( make same size tow input matrices ******** )
[AFR,AFC] = size(A_Fisrtly);
[ASR,ASC] = size (A_Secondly);
sub_col = AFC-ASC;
if sub_col~= 0
    resizable=zeros(AFR,abs(sub_col));
    for g=1:abs(sub_col)
        rand_resize = randi(AFR);
        resizable(rand_resize,g) = floor(L/li(rand_resize));
    end
    if sub_col >0
        A_Secondly= horzcat(A_Secondly,resizable);
    end
```

```

    if sub_col < 0
        A_Fisrtly= horzcat(A_Fisrtly,resizable);
    end
end
% *********************** ( CrossOver *********************** )
%this section describes used Crossover method in this study
[row_cnt,col_cnt]=size(A_Fisrtly);
while num < cnt
    A=A_Fisrtly
    B=A_Secondly
    [row_cntA,col_cntA]=size(A);
    [row_cntB,col_cntB]=size(B);
    break_point_cnt=randi(col_cntA-1);
    temp_A=A;
    for k1=1:(break_point_cnt)
        break_point=randi(col_cntA);
        mat_temp=A(:,break_point);
        A(:,break_point)=B(:,break_point);
        B(:,break_point)=mat_temp;
    end
% *********************** ( Mutation )***********************
    r3=randi(numel(A));
    fcol=ceil(r3/col_cnt);
    frow= r3-((fcol-1)*col_cnt);

   A(:,fcol)=0;
    for u=1:row_cntA+5
        uu=randi(row_cntA);
        my_mod=L-li*A(:,fcol);
        max_rand=floor(my_mod/li(uu));
        max_rand=A(uu,fcol)+max_rand
        A(uu,fcol)=randi(max_rand+1)-1;
    end

    r3=randi(numel(B));
    fcolB=ceil(r3/col_cnt);
    frowB= r3-((fcolB-1)*col_cnt);
```

```
    B(:,fcolB)=0;
    for u=1:row_cntB+5
        uu=randi(row_cntB);
        my_mod=L-li*B(:,fcolB);
        max_rand=floor(my_mod/li(uu))
        max_rand=B(uu,fcolB)+max_rand;
        B(uu,fcolB)=randi(max_rand+1)-1;
    end
% ************************ ( There shouldn't be zero row or column
% If there is to be replaced with valid values)***********************
    [siA,sjA]=size(A);
    [siB,sjB]=size(B);
    nz=0;
    for repB=1:siB
        if B(repB,:)==0
            rand_resize = zeros(AFR,1);
            rand_resize(repB) =floor( L/li(repB));
            B= horzcat(B,rand_resize);
        end
    end
    for repA=1:siA
       if A(repA,:)==0
            rand_resize = zeros(AFR,1);
            rand_resize(repA) = floor( L/li(repA));
            A= horzcat(A,rand_resize);
        end
    end
    [siA,sjA]=size(A);
    [siB,sjB]=size(B);
    for repB=1:sjB
        if B(:,repB)==0
            rand_zero_row=randi(AFR);
            B(rand_zero_row,repB)= floor( L/li(rand_zero_row));
        end
    end
    for repA=1:sjA
        if A(:,repA)==0
            rand_zero_row=randi(AFR);
```

```
            A(rand_zero_row,repA)= floor( L/li(rand_zero_row));
                        end
    end
    % if nz==0
    [A, ia, ic] = unique(A','rows')
    [B, ia, ic] = unique(B','rows')
    A_child{1,num+1}=B';
    A_child{1,num+2}=A';
    num=num+2;
    % end
end
num
```

واژه‌نامه فارسی به انگلیسی

احتمالی . . . . Probabilistic
کروموزوم . . . . Chromosome
برش، تقاطع، همبری . . . . Crossover
نخبه گرایی، نخبه سالاری، بهینه گزینی . . . . Elitism
شایستگی، برازش . . . . Fitness
تابع برازش، تعیین میزان شایستگی یک راه‌حل . . . . Fitness – Function
جهش . . . . Mutation
فضای اولیه جواب‌ها با همه محیط آن که نشان دهنده رفتار فیزیکی واقعی است. . . . . Phenotype

واژه‌نامه انگلیسی به فارسی

کروموزوم .... Chromosome
برش، تقاطع، همبری .... Crossover
نخبه گرایی، نخبه سالاری، بهینه گزینی .... Elitism
شایستگی، برازش .... Fitness
تابع برازش، تعیین میزان شایستگی یک راه‌حل .... Fitness – Function
جهش .... Mutation
احتمالی .... Probabilistic
فضای اولیه جواب‌ها با همه محیط آن که نشان دهنده رفتار فیزیکی واقعی است. .... Phenotype

Hakim Sabzevari University

An Outline of MSc. Thesis

Surname:Shahmansoori	Name:Mohammad Hosein	Student No.:9313137064
Supervisor: Dr. Mehdi Zaferanieh		
Advisor: Dr. Mahmood Amintoosi		
Faculty of Mathematics and Computer Science		
Program: Applied Mathematics Field:Operational Research		
Title of thesis: Cutting Stock Problem		
Keywords:Column Generation, Genetic Algorithm, Cutting Stock Problem		
Abstract:		

Hakim Sabzevari University

Faculty of Mathematics and Computer Science

A Thesis Submitted in Partial Fulfilment of the Requirement for the Degree of Master of Science in Applied Mathematics

Cutting Stock Problem

Supervisor:

Dr. Mehdi Zaferanieh

Advisor:

Dr. Mahmood Amintoosi

By:

Mohammad Hosein Shahmansoori

June 2017