

بهینه سازی روش جایگذاری الگوها با در نظر گرفتن چندین دسته الگو

وچرخش قطعات

پدرام پیوندی^۱ - محمد امانی تهران^۲ - مسعود لطیفی^۳

^۱ دانشجوی دکترا، ^۲ استادیار، ^۳ دانشیار - دانشکده مهندسی نساجی، دانشگاه صنعتی امیرکبیر

چکیده

بهینه سازی چیدمان قطعات بریده شده دارای کاربردهای زیادی در صنایع برش ورق فلزی، چوببری، شیشه، چرم، کاغذ، کشتی سازی، صنایع فضایی و پوشاک می باشد به دلیل اهمیت کاهش ضایعات، روشهای زیادی برای حل این مشکل ارائه شده است. یکی از بهترین روشها استفاده از الگوریتم ژنتیک می باشد. هدف اصلی در این گونه مسائل قرارگیری قطعات با دورریز کمینه وبدون داشتن تداخل بروی سطح می باشد.

عوامل زیادی در دستیابی به بهینه سازی چیدمان قطعات بریده شده با راندمان بالا مؤثرند که از آن جمله می توان به دو عامل مهم تعداد مجموعه قطعات انتخابی و در نظر گرفتن امکان چرخش برای قطعات اشاره کرد.

در این پژوهش به این دو عامل توجه شده است. نتایج نشانگر این است که استفاده از تابع جریمه در تعیین برازندگی کروموزمها و همچنین وجود یک تابع برازندگی متغیر با تعداد تولید نسلها، در دستیابی به بهینه سازی با راندمان بالا دارای نقش کلیدی می باشد.

در این پژوهش نتیجه حاصل برای بهینه سازی یک دسته الگوهای غیر منظم با جمعیت اولیه ۵۰۰ و تعداد تولید نسل ۳۵۰ دارای راندمانی برابر با ۷۳٫۹٪ شد. و همچنین نتیجه حاصل برای بهینه سازی دودسته الگوهای غیر منظم با جمعیت اولیه ۱۰۰۰ و تعداد تولید نسل ۳۵۰ دارای راندمانی برابر با ۷۱٫۶٪ شد.

کلمات کلیدی: بهینه سازی - الگوریتم ژنتیک - الگو - چرخش

Optimization of Pattern Layout Considering Multi- Set Pattern & Pieces Rotation

P.Peivandi¹, M.Amani-Tehran² & ³M.Latifi

¹Phd. Student, ²Assistant Professor, ³Associate Professor

Department of Textile Engineering, Amirkabir University of Technology

Abstract

Packing problems arise in a wide variety of application Areas such as sheet metal, lumber, glass, leather, textile, and paper industries. Because of importance of minimizing scrap losses (cutting stock problem), many methods have been represented. One of the best methods is application of Genetic Algorithm. The main purpose in this case, is minimizing scrap losses without overlapping of pieces.

Many factors effects to achieve the qualified optimization. Two of the most important factors that we consider in this project are the number of sets and pieces' rotation. The results show that using penalty function to determine fitness of chromosomes and using dynamic fitness function are very important to achieve the qualified optimization.

The result of one non-fit set pattern with 500 initial population and 350 generation is 73.9% efficiency.

The result of two non-fit set pattern with 1000 initial population and 350 generation is 71.6 % efficiency.

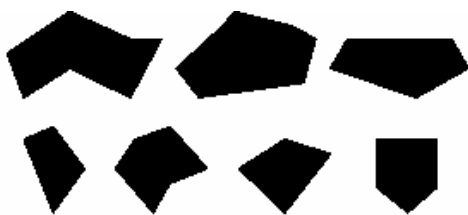
Key words: Cutting Stock-Cutting-Optimization-Genetic Algorithm-Rotation

۱- مقدمه

قبلی اشکال فقط امکان جابجای در راستای محور های مختصات را، برای دستیابی به چیدمان بهینه داشتند. در این مقاله سعی در تکمیل کارهای قبلی و همچنین در نظر گرفتن امکان چرخش برای الگوها با هدف دستیابی به چیدمان بهینه تر شده است.

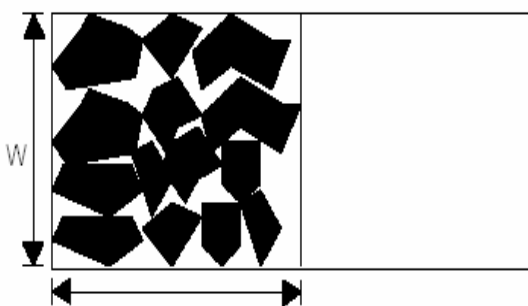
۲- طرح مسئله

در شکل (۱) نمونه ای از الگوهای سفارش داده شده، که می تواند یک پیراهن باشد. نشان داده شده است.



شکل (۱) نمونه ای از الگوهای سفارش داده شده. [۱]

اگر دو پیراهن مدنظر باشد چیدمان آن در طول پارچه‌ای با عرض ثابت مانند شکل (۲) خواهد بود. که در آن الگوها می توانند بین تا ۳۶۰ درجه چرخش داشته باشند. در این تحقیق هر الگو به صورت یک شکل چند ضلعی مدل شد و الگوهایی که دارای محیطی منحنی شکل بودند به چند ضلعی تقریب زده شدند.



شکل (۲) چیدمان الگوها در عرض ثابت [۱]

برای n الگوی داده شده در شکل (۲) هر شکل در حالت اولیه دارای زاویه می باشد اما ممکن است این زاویه در جواب نهایی تا ۳۶۰ درجه تغییر کرده باشد. ساختار جواب نهایی به صورت زیر است:

$$S = [(f_1, x_1, y_1, o_1), (f_2, x_2, y_2, o_2), \dots, (f_n, x_n, y_n, o_n)]$$

که در آن S جواب نهایی و f الگو و X طول و Y عرض از مبدا هر الگو و O میزان چرخش هر الگو می باشد.

جایگذاری الگوهای لباس بر روی پارچه به صورتی که باعث کمترین دور ریز در پارچه شود بسیار مورد توجه صنعت پوشاک می باشد. در طی بیست سال گذشته شماری از مدل‌های جایگذاری، از مدل‌های نیمه خودکار گرفته تا مدل‌های جایگذاری خودکار رایانه‌ای، پیشرفت داده شده‌اند. در روش‌های نیمه خودکار، جایگذاری با بهره از یک ابزار گرافیکی محاوره ای انجام می‌گیرد. کاربر می‌تواند با بهره از یک ابزار مناسب ورودی، الگوها را بر روی نمایش‌گر جابجا کرده و یا بچرخاند. در طول روند جایگذاری، نرم‌افزار به گونه‌ی پیوسته خطاهای پدیدآمده مانند روی هم افتادگی را آزمایش کرده و اندازه‌ی دور ریز را محاسبه می‌نماید. در بیشتر روش‌های پیشرفت یافته برای جایگذاری، زمان محاسبه بسیار بالا است. در چند سال گذشته شماری از تقریب‌های موفق جایگذاری، با بهره‌گیری از الگوریتم‌های اکتشافی و به سبب توانایی کاربردی آن‌ها در طراحی و تولید رایانه‌ای پیشرفت داده شده‌اند.

به گونه‌ی کلی، جایگذاری الگوهای دوبعدی به دو دسته جایگذاری الگوهای چهارگوش و نامنظم تقسیم می‌گردد. بیشتر روش‌های جایگذاری گزارش شده در مقاله‌های چند سال گذشته، بر جایگذاری الگوهای چهارگوش تمرکز دارند.

مسئله‌ی مهمی که کنترل جایگذاری را دشوار می‌کند، زمان محاسباتی مورد نیاز برای بدست آوردن پاسخ بهینه است که به گونه‌ی توانی با افزایش شمار الگوها افزایش می‌یابد. مسایل جایگذاری به عنوان مسایل دشوار شناخته شده‌اند و در نتیجه به گونه‌ی دقیق نمی‌توانند در زمان کوتاه پاسخ داده شوند. افزون بر این، محدودیت‌های بسیاری در مسایل واقعی وجود دارد که به طور معمول دشوارند و به سادگی نمی‌توان آن‌ها را به گونه‌ی ریاضی در آورد. به این دلایل، برخی از پژوهشگران تاکنون روش‌های جستجوی اکتشافی گوناگونی را بررسی نموده‌اند که از روش بهینه‌سازی بهره نمی‌گیرد ولی داده‌های اکتشافی بدست آمده از آزمایش‌ها و روش‌های نزدیک بهینه را به کار می‌گیرد. یکی از کاربردی‌ترین روش‌های جستجوی اکتشافی که در این تحقیق نیز از آن استفاده شد، الگوریتم ژنتیک می باشد.

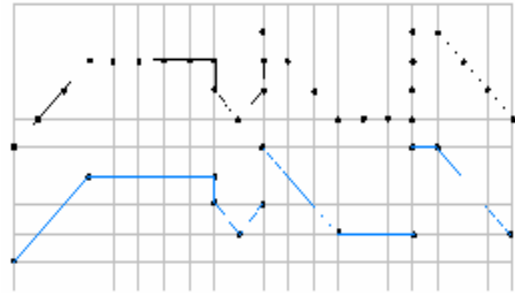
الگوریتم ژنتیک [۱] متعلق به دسته ای از روش‌های جستجوی احتمالی می باشد که از روی پدیده تکامل در طبیعت الگو برداری شده‌اند. در قدم اول از این روش برای بهینه سازی اشکال ساده هندسی استفاده شد [۲، ۳]. در مرحله بعدی سعی در بهینه سازی الگوهای غیر منظم شد [۴]. در کلیه کارهای

۳- روش کار

۳-۱ چندگوش سازی

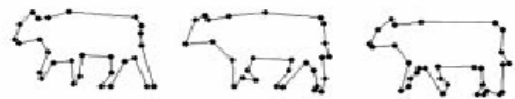
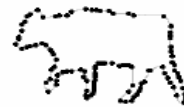
مرحله مقدماتی در بکارگیری الگوریتم جایگذاری، تبدیل الگوها و در صورت لزوم صفحه به صورت چندگوش می باشد. چندگوش سازی عبارت از تقریب زدن لبه های خمیده اشکال با خط راست می باشد. به این ترتیب ضمن تبدیل شکل به مجموعه ای از بردارها، تعداد بردارهای لازم برای مشخص کردن شکل نیز کاهش می یابد.

روش مورد استفاده، روش خطی سازی ناحیه ای (Piecwise Linear) می باشد. در این روش فاصله نقاط پشت سر هم محاسبه شده و به جای آن خطی که اولین نقطه را به آخرین نقطه متصل می کند جایگزین می شود به طوری که اختلاف طولی خط جدید و مجموع طول نقاط پیوسته از حد معینی (خطا) بیشتر نشود. شکل (۳)



شکل (۳) روش Piecwise Linear [۵]

در شکل (۴) نمونه های از چند گوش سازی با روش خطی سازی ناحیه ای با خطاهای مختلف نشان داده شده است.



شکل (۴) نمونه های از خطی سازی ناحیه ای Piecwise Linear

[۵]

۳-۲ الگوریتم ژنتیک

الگوریتم ژنتیک یک روش الهام گرفته از طبیعت است که برای بهینه سازی می تواند به کار برده شود. این الگوریتم بر پایه فرایندهای ژنتیکی ارگانیسم های بیولوژیک عمل می کند که در چندین نسل، جمعیت بر اساس اصول گزینش و زنده ماندن برانده ترین که نخستین بار چارلز داروین آن را مطرح نمود،

رشد می کند. با پیروی از این فرآیند، الگوریتم ژنتیک می تواند پاسخ هایی پدید آورده و آن ها را بهبود بخشد.

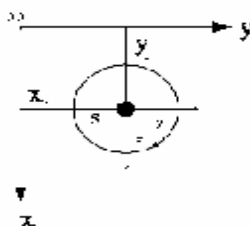
در طبیعت، جانداران برای زنده ماندن با هم رقابت می کنند و آن هایی که از توانایی بالاتری برخوردارند امکان تولیدمثل بیشتری خواهند داشت. به گفته ی دیگر، ژنهایی که دارای توانایی سازگاری و بهینه شدن بالایی هستند، در هر نسل گسترش یافته و گونه ی خود را افزایش می دهند. ترکیب ویژگی های خوب از پدر و مادرهای گوناگون، گاهی می تواند فرزندان با برانده ترین با لا پدید آورد که این برانده ترین، بسیار بیشتر از برانده ترین تک تک پدر و مادرها است.

الگوریتم ژنتیک با جمعیتی از افراد که هر کدام از آن ها بیانگر مناسب بودن پاسخ مسئله است، نمایش داده می شود. افراد دارای برانده ترین با لا و در فرآیند تولید مثل با دیگر افراد جمعیت، شانس بیشتری برای تولید دوباره دارند و افراد دارای برانده ترین پایین، شانس کمتری برای گزینش در این عمل پیدا می کنند و از این رو از میان می روند. جمعیت تازه، از گزینش بهترین افراد نسل کنونی و جفت گیری آن ها با هم پدید می آید. این نسل تازه، دارای سهم بیشتری از ویژگی های افراد مناسب نسل گذشته است، با این روش، در چندین نسل ویژگی های خوب توسط جمعیت گسترش یافته و با ویژگی های خوب دیگر آمیخته می شود. با این کار بیشترین فضای خوش آتیه از فضای جستجو بررسی می گردد. اگر الگوریتم ژنتیک به خوبی پایه ریزی شده باشد، جمعیت به پاسخ بهینه رسیده و در آن همگرا می شود.

۳-۳ کدگذاری و جمعیت اولیه

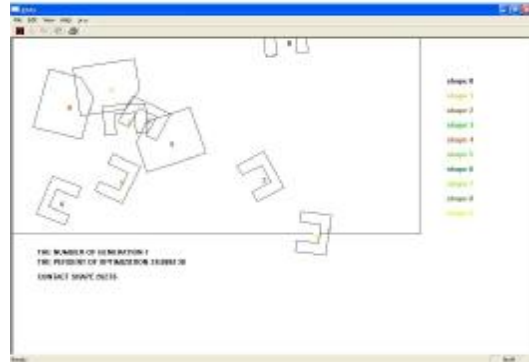
ابتدا فاصله نقاط روی محیط هر یک از اشکال نسبت به مرکز شکل محاسبه می شود و در نهایت مختصات مرکز شکل نسبت به مختصات اصلی (گوشه چپ بالا) محاسبه می شود. اینکار باعث می شود که محاسبات چرخش شکل نسبت به مرکز شکل به راحتی صورت گیرد (شکل ۵).

در نهایت مختصات شکل به صورت (θ_i, X_i, Y_i) نشان داده می شود. و می توان از یک آرایه سه بعدی برای ذخیره مختصات اشکال استفاده نمود.



شکل (۵) نحوه کدگذاری

جمعیت اولیه با قرار دادن تصادفی الگوها بر روی صفحه ایجاد می شود. جایگذاری تصادفی شامل طول و عرض از مبدا و همچنین زاویه الگوها است. در شکل (۶) نمونه ای از یک جایگذاری (کروموزم) از جمعیت اولیه نمایش داده شده است.



شکل (۶) نمونه ای از یک (کروموزم) از جمعیت اولیه

۴-۳ تابع برازندگی

در کنار عمل کدگذاری، تابع برازندگی نقش مهمی در اجرای الگوریتم ژنتیک ایفا می کند. تحقیقات انجام شده بر روی اثر عوامل دیگر (مانند نحوه و درصد دورگه شدن، اندازه جمعیت و نحوه و درصد جهش) که هیچ یک به اندازه عمل کدگذاری و تابع برازندگی، بحرانی و حساس نیستند.

بطور کلی، تعریف تابع برازندگی با قاعده و مناسب، نیاز تمامی مسائل است و متأسفانه برای بسیاری از آن‌ها، امکان ایجاد چنین توابعی ممکن نیست. معهداً، در صورت امکان اجرای مطلوب الگوریتم ژنتیک، ساختن تابع‌های برازندگی باید به گونه‌ای انجام شود که از ایجاد نقاط بیشینه محلی زیاد و یا نقطه‌ی بیشینه کلی جدا افتاده و تنها، جلوگیری کند.

قانون کلی در ساختن تابع برازندگی، قابلیت انعکاس مؤثر مقدار واقعی کروموزم (افراد) است. این امر برای بسیاری از مسائل موضوعی واضح است. به عنوان مثال اگر هدف، طراحی یک نازل با بیشترین جریان باشد، تابع برازندگی تعیین کننده‌ی مقدار جریانی است که از درون آن در واحد زمان عبور می نماید. محاسبه آن، کار سهل و آسانی نیست ولی مورد و چگونگی محاسبه مشخص بوده و دانش آن در کتب مرجع قابل یافتن است.

مقدار تابع برازندگی کروموزم همیشه یک مقدار مفید برای راهنمایی الگوریتم ژنتیک در فضای جستجو نیست. در بهینه‌سازی ترکیبی که علاوه بر تابع هدف محدودیت‌های زیادی نیز وجود دارند، اغلب نقاط در فضای جستجو، کروموزم‌های بی‌اعتباری را معرفی می کنند که دارای مقدار برازندگی صفر هستند. در این حالت برای مؤثر واقع شدن الگوریتم ژنتیک باید به گونه‌ای تابع برازندگی ساخته شود که

کروموزم بی‌اعتبار، هدایت کننده‌ی ما به طرف کروموزم‌های معتبر شود. بایستی محل کروموزم‌های معتبر دانسته شود تا نقاط نزدیک به آن‌ها بتوانند مقادیر برازندگی مطلوبی را ایجاد کنند و نقاط دورتر مقادیر برازندگی ضعیفی را بدهند. ولی در صورت نشناختن کروموزم‌های معتبر، این عمل ممکن نیست.

یک روش برای نزدیک شدن به جواب، استفاده از «تابع جریمه» است. این تابع می تواند بعنوان معیاری برای غیر معتبر بودن کروموزم بکار رود. تابع جریمه مناسب می تواند بعنوان هزینه اعمال محدودیت‌های موجود برای تابع هدف در نظر گرفته شود. به عبارتی، آن تابعی است که بیانگر مقدار هزینه تخصیصی به یک کروموزم بی‌اعتبار است تا آن را به یک کروموزم معتبر و جزو دسته جوابهای ممکن در فضای جستجو، تبدیل کند.

این تابع از دو قسمت تشکیل یافته است:

الف) تابع هدف: که در آن حداقل کردن مقدار دوریز (ضایعات)، هدف اصلی است به صورت زیر تعریف می شود.

$$o(s) = \sum_{j=1}^n (x_j \cdot A_j + y_j)$$

$X(j)$ و $Y(j)$ مختصات برداری اشکال (الگوها) و $A(j)$ مساحت آنها است. برای هر ترکیبی از اشکال، تابع هدف نسبت به خط منتهی‌الیه چپ محاسبه می گردد. مساحت یک الگو قابل مقایسه با جرم ذرات در سیستم فیزیکی است. ضریب اختصاص داده شده به مساحت لحظه‌ای اشکال (X_j, A_j) در تابع هدف، قابل مقایسه با عامل شتاب در ذرات سیستم فیزیکی است. بهنگام کاهش مساحت‌های لحظه‌ای اشکال، کل تابع هدف نیز کاهش می یابد این عمل باعث کشاندن الگوها به سمت کناره چپ می گردد و فاصله میانی الگوها را نسبت به هم کاهش می دهد.

ب) تابع جریمه:

توابع جریمه برپایه میزان تجاوز از محدوده‌ی فضای معتبر و تعریف شده از جوابهای ممکن، به تابع برازندگی افزوده می شوند، بگونه‌ای که با طراحی یک تابع جریمه مناسب، الگوریتم ژنتیک برای همگراشدن در یک جواب بهینه فرعی یا حتی جواب بهینه نهایی، آماده می شود. تنها محدودیت در نظر گرفته شده، عدم همپوشانی اشکال الگوها برای رسیدن به استقرار بهینه است.

$$p(s) = \sum_{j=1}^{n-1} \sum_{i=j+1}^n \text{OverlapArea}(ij)$$

در نهایت تابع برازندگی به صورت زیر تعریف می شود:

$$fitnessfunction = \frac{k}{a.o(s) + b.p(s)}$$

که در آن a, b, k مقادیر ثابتی هستند که به صورت تجربی تعیین می شوند.

۵-۳ تولید نسل جدید

حدود ۱۰٪ از نسل جدید مستقیماً از بهترین کروموزمهای نسل قبلی کپی می شوند و حدود ۲۰٪ نیز با استفاده از چرخ رولت [۶] انتخاب می گردند. عمل دورگه شدن به وسیله انتخاب یک نقطه به طور تصادفی برای دو کروموزم و تعویض ژنهای دو کروموزم از آن نقطه به بعد صورت می گیرد. که بین ۳۰٪ تا ۴۰٪ کروموزمهای نسل جدید با این روش تولید می شوند.

عملگر جهش نیز مختصات چند شکل را در کروموزمی که به صورت تصادفی با روش رولت انتخاب شده است را به صورت راندوم تغییر می دهد. که تعداد اشکال جابه جا شده با تعداد نسل تولید شده ارتباط دارد. بین ۲٪ تا ۱۰٪ نسل جدید به وسیله این عملگر تولید می شوند.

عملگر جابجایی (swap) نیز مکان دو شکل را در کروموزمی که به صورت تصادفی با روش رولت انتخاب شده اند با هم تعویض می کند. دو شکل نیز به صورت تصادفی انتخاب می شوند.

این عملگر بر روی ۱۰٪ جمعیت اعمال می شود.

جهش ظاهری (Phenotype mutation) برای بهترین کروموزم در هر نسل بکار می رود. که با اعمال تغییرات کوچکی در طول و عرض مختصات بردارهای اشکال و زاویه آنها، فضای موجود را بیشتر جستجو می کند. بعد از هر تغییر، جواب حاصل ارزیابی می شود و در صورت رسیدن به شرایط و جواب بهتر، آن حالت حفظ می گردد.

این عملگر بر روی هر یک از مختصات بردارها تغییرات لازم را با افزودن و کاستن مقادیر واحد بصورت مرحله به مرحله انجام می دهد. اگر مقدار برازندگی بهتر از مقدار اصلی باشد، این تغییر جایگزین مقادیر قبلی شده و تغییرات مختصات ادامه می یابد. عملگر جهش ظاهری باعث تسریع در پوشش فضای جستجو برای یافتن بهینه نهایی می شود اما در عین حال باعث افزایش عملیات محاسباتی لازم نیز می شود، به همین دلیل این عملگر بر روی نسلهای پایانی اعمال می شود. در شکل (۷) ساختمان برنامه نشان داده شده است.



شکل (۷) ساختمان برنامه

۴- آزمایشات

۴-۱ بررسی تأثیر بزرگی جمعیت نسلهای

فرضیات:

-تعداد اشکال ۱۰ عدد.

-به تمام اشکال امکان چرخش داده شد.

-تعداد تولید نسل ها ۳۵۰.

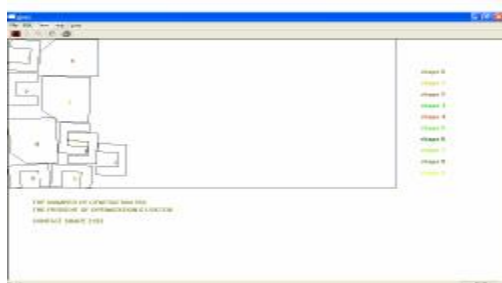
۲-۴ بررسی تغییر در تعداد کروموزمها برای انتخاب در نسل اول

الف: -نسل اول از میان ۵۰۰۰۰ کروموزم تولید شده بصورت تصادفی، انتخاب شد.



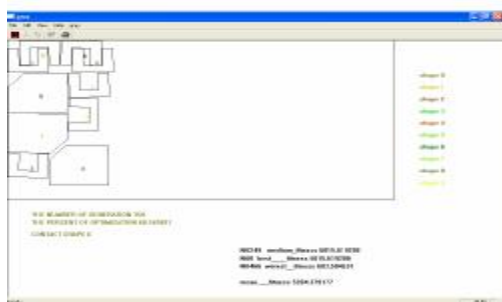
شکل (۱۱) بهترین کروموزم در نسل سی صدوینجاهم با انتخاب نسل اول از میان ۵۰۰۰۰ کروموزم.

ب: -نسل اول بصورت تصادفی انتخاب شد.



شکل (۱۲) بهترین کروموزم در نسل سی صدوینجاهم با انتخاب نسل اول تصادفی.

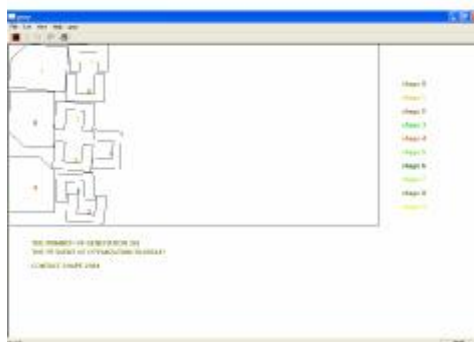
ج: -نسل اول از میان ۱۰۰۰۰ کروموزم تولید شده بصورت تصادفی، انتخاب شد.



شکل (۱۳) بهترین کروموزم در نسل سی صدوینجاهم با انتخاب نسل اول از میان ۱۰۰۰۰ کروموزم.

همانطور که از شکلها مشخص است انتخاب نسل اول از میان ۱۰۰۰۰ کروموزم نتیجه بهتری نسبت به بقیه حالات دارد. اهمیت انتخاب نسل اول در تولید نسلهای کم مشخص می شود و در تولید نسلهای زیاد، در نهایت فرقی بین انتخاب نسل اول به صورت تصادفی یا غیر تصادفی وجود ندارد [۸] (شکل ۱۴).

-نسل اول از میان ۱۰۰۰۰ کروموزم تولید شده بصورت تصادفی، انتخاب شد.
الف: اجرای برنامه با ۵۰ کروموزم در هر نسل:



شکل (۸) بهترین کروموزم در نسل سی صدوینجاهم با ۵۰ کروموزم در هر نسل.

ب: اجرای برنامه با ۳۰۰ کروموزم در هر نسل:



شکل (۹) بهترین کروموزم در نسل سی صدوینجاهم با ۳۰۰ کروموزم در هر نسل.

د: اجرای برنامه با ۵۰۰ کروموزم در هر نسل:

همانطور که مشاهده می شود عملکرد سیستم بطور زیادی متأثر از تعداد کروموزمها در نسلها می باشد و اگر این تعداد از حدی کمتر باشد، سیستم الگوریتم ژنتیک به علت نبود تغییرات کارایی خود را از دست می دهد [۷].

حداقل کروموزمها در نظر گرفته شده در این برنامه ۵۰۰ کروموزم در نسل می باشد. باید به این نکته توجه داشت که کاهش یا افزایش تعداد کروموزمها در نسل به شدت بر روی سرعت اجرای برنامه تأثیر می گذارد.



شکل (۱۰) بهترین کروموزم در نسل سی صدوینجاهم با ۵۰۰ کروموزم در هر نسل.

۴-۴ تأثیر ثابت یا متغیر بودن محدوده اثر عملگرها

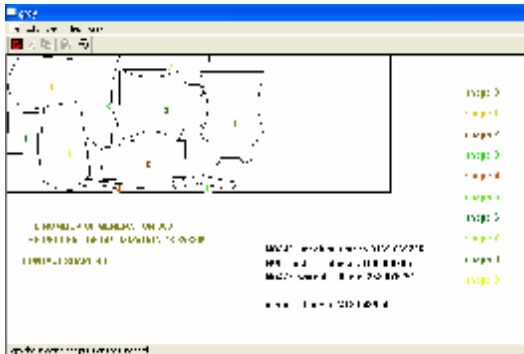
در نسله‌ها

الف: محدوده اثر عملگرها متغیر:

دورگه شدن از ۵٪ تا ۳۵٪ جهش از ۱۰٪ تا ۲۰٪ جابه جای اشکال از ۱۰٪ تا ۲۰٪ جهش پرشی از ۱۰٪ تا ۲۰٪ جهش زاویه ای از ۵٪ تا ۱۵٪ کپی کردن از ۲۵٪ تا ۱۰٪ با افزایش تولید نسله‌ها محدوده اثر عملگرها مطابق درصد‌های ذکر شده تغییر می‌کند. و نتیجه آن مانند شکل (۱۶) می‌باشد.

ب: محدوده اثر عملگرها ثابت:

دورگه شدن ۴۰٪ جهش ۲۰٪ جابه جای اشکال ۱۵٪ جهش پرشی ۱۰٪ جهش زاویه ای ۵٪ کپی کردن ۱۰٪ همانطور که از شکلهای (۱۶) و (۱۷) مشخص است ثابت یا متغیر بودن محدوده اثر عملگرها تأثیر چندانی در نتیجه نهایی ندارد. اما در هر صورت استفاده از محدوده اثر متغیر دارای نتیجه بهتری خواهد بود.



شکل (۱۷) نتیجه نهایی با محدوده اثر ثابت عملگرها

۴-۵ میزان تأثیر عملگرهای دورگه شدن و جهش

الف - درصد بالای دورگه شدن:

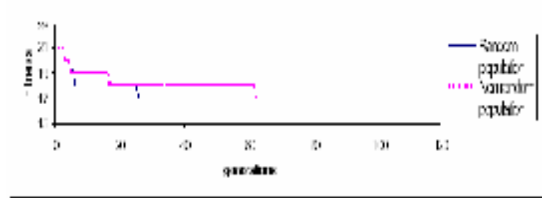
دورگه شدن ۹۰٪ جهش ۲٪ جابه جای اشکال ۱٪ جهش پرشی ۱٪ جهش زاویه ای ۱٪ کپی کردن ۵٪ نتیجه در شکل (۱۸) نشان داده شده است.

ب: درصد بالای جهش:

جهش ۸۷٪ دورگه شدن ۵٪ جابه جای اشکال ۱٪ جهش پرشی ۱٪ جهش زاویه ای ۱٪ کپی کردن ۵٪ در شکل (۱۹) نتیجه نشان داده شده است.



شکل (۱۸) نتیجه نهایی با ۹۰٪ دورگه شدن



شکل (۱۴) مقایسه نسل اول به صورت تصادفی یا انتخابی [۸]

همانطور که از شکل (۱۱) مشخص است از انتخاب نسل اول از میان تعداد کروموزومهای بسیار زیاد به علت یکنواختی (کروموزومها با برازندگی بالا و اختلاف کم در میزان برازندگی) موجب اختلال در عملکرد الگوریتم ژنتیک می‌شود.

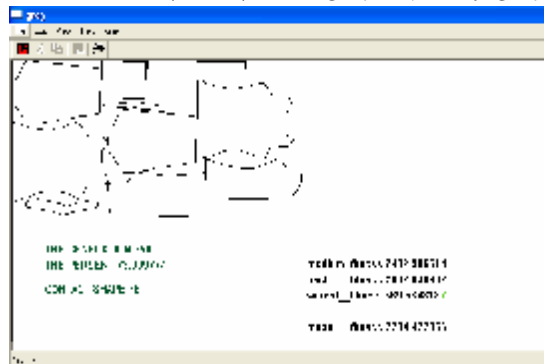
۴-۳ تأثیر ثابت یا متغیر بودن میزان تابع جریمه

الف: -میزان جریمه دور بودن از مبدا و جریمه تداخل اشکال ثابت فرض شد. (۱۵)



شکل (۱۵) ثابت بودن تابع جریمه

ب: -میزان جریمه دور بودن از مبدا و جریمه تداخل اشکال با افزایش تولید نسله‌ها، افزایش یافت. (شکل ۱۶)

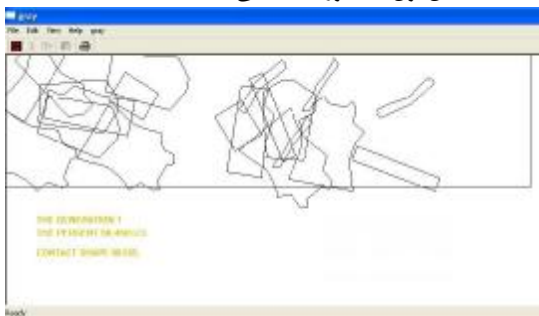


شکل (۱۶) متغیر بودن تابع جریمه (روند افزایشی)

همانطور که مشخص است افزایش تابع جریمه با توجه به شماره تولید نسل امکان بیشتری برای حذف کروموزومهای ضعیف فراهم می‌آورد و در شروع نیز امکان وجود تنوع بیشتری را در نسله‌های اولیه می‌دهد.

ب:

- تعداد اشکال ۲۰ عدد.
- به تمام اشکال امکان چرخش داده شد.
- تعداد کروموزم ها در هر نسل ۱۰۰۰ عدد.
- تعداد تولید نسل ها ۳۵۰.
- انتخاب نسل اول به صورت تصادفی



شکل (۲۲) بهترین کروموزم در نسل اول برای دودسته الگو با جمعیت ۱۰۰۰ کروموزم در هر نسل

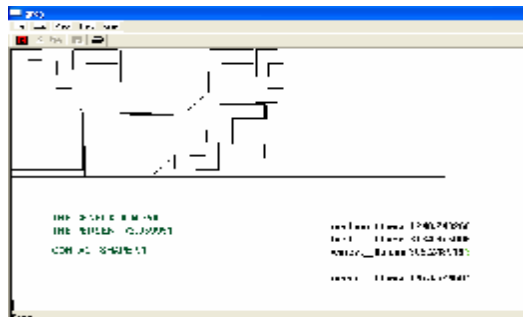


شکل (۲۳) نتیجه نهایی برای دودسته الگو با جمعیت ۱۰۰۰ کروموزم در هر نسل.

همانطور که از شکل (۲۱) مشخص است نتیجه نهایی حاصل از چیدن دو دسته الگو با جمعیت ۵۰۰ کروموزم در هر نسل دارای تداخل اشکال می باشد برای رفع مشکل جمعیت در هر نسل به ۱۰۰۰ کروموزم در هر نسل افزایش یافت که این افزایش همراه با افزایش زمان اجرا برنامه (حدود ۱۵ ساعت) بود. و نتیجه (شکل ۲۳) حاصل نشان دهنده کاهش میزان تداخل اشکال است. با افزایش میزان جریمه تداخل در تابع برازندگی مشکل تداخل اشکال بطور کلی برطرف شد (شکل ۲۴).



شکل (۲۴) نتیجه نهایی با افزایش میزان جریمه تداخل در تابع برازندگی

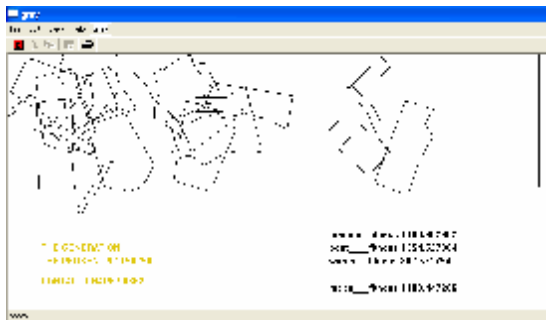


شکل (۱۹) نتیجه نهایی با ۸۹٪ جهش همانطور که از نتایج مشخص است کارایی عملگر جهش از دو رگه شدن بهتر می باشد. نتایج حاصل از مقایسه دو عملگر جهش و دو رگه شدن شامل موارد زیر است [۹]:
-عملگر دورگه شدن در جمعیت های زیاد باعث بالا رفتن سریع برازندگی می شود.
-عملگر جهش در جمعیت های کم و با تولید نسل زیاد باعث بالا رفتن سریع برازندگی می شود.

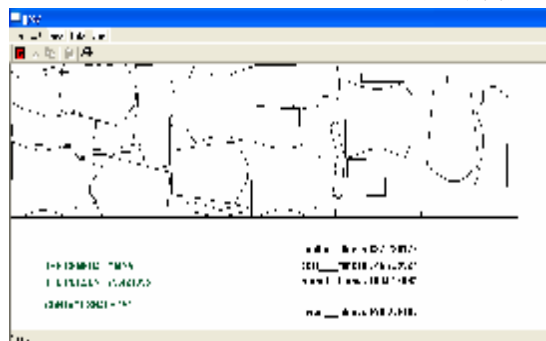
۴-۶ پهنه سازی چند دسته الگو:

الف:

- تعداد اشکال ۲۰ عدد.
- به تمام اشکال امکان چرخش داده شد.
- تعداد کروموزم ها در هر نسل ۵۰۰ عدد.
- تعداد تولید نسل ها ۳۵۰.
- انتخاب نسل اول به صورت تصادفی



شکل (۲۰) بهترین کروموزم در نسل اول برای دودسته الگو با جمعیت ۵۰۰ کروموزم در هر نسل.



شکل (۲۱) نتیجه نهایی برای دودسته الگو با جمعیت ۵۰۰ کروموزم در هر نسل.

۵- نتیجه گیری و پیشنهادات

بر خلاف کار قبلی [۴] در این پروژه از تابع جریمه در تعیین میزان برازندگی کمک گرفته شد. نتایج نشان داد که تابع جریمه در مسأله استقرار بهینه الگوها نقش مؤثری دارد. همچنین آزمایشات نشان داد که تابع برازندگی متغیر نسبت به تابع برازندگی ثابت دارای عملکرد بهتری می باشد. این امر با نتایج کارهای قبلی [۲] و [۳] نیز همخوانی دارد. در این تحقیق عامل چرخش نیز برای الگوها در نظر گرفته شد که امکان چیدمان بهتر را برای الگوها فراهم می آورد اما در عین حال فضای جستجو را به شدت افزایش می دهد، که دو عامل تعداد کروموزمها در جمعیت اولیه و تعداد تولید نسل، درپوش فضای جستجوی نقش اساسی دارند به گونه ای که با تعداد جمعیت اولیه کم و تعداد تولید نسل زیاد تقریباً به همان جوابی می رسیم که با تعداد جمعیت اولیه زیاد و تعداد تولید نسل کم، بنابراین حاصل ضرب این دو فاکتور باید بهینه شود. در این تحقیق برای بهینه چینی یک دسته الگو تعداد ۵۰۰ کروموزم برای هر نسل و تعداد تولید نسل برابر ۳۵۰ بار در نظر گرفته شد که این نتایج پس از بررسی تعداد کروموزمها در نسل و تعداد تولید نسلهای مختلف حاصل شد. بطور مثال با افزایش تعداد تولید نسلهای از ۳۵۰ به ۵۰۰ تغییر قابل توجهی در جواب نهایی حاصل نشد. همچنین نتایج نشان داد که تعیین نسل اول به صورت انتخابی یا تصادفی تأثیر چندانی در نتیجه بهینه چینی یک دسته الگو

ندارد. در بهینه چینی دو دسته الگو نتایج حاکی از نیاز به افزایش تعداد جمعیت اولیه بود. در این پژوهش نتیجه حاصل برای بهینه سازی یک دسته الگوهای غیر منظم با جمعیت اولیه ۵۰۰ و تعداد تولید نسل ۳۵۰ دارای راندمانی برابر با ۷۳،۹٪ شد (شکل ۱۶). و همچنین نتیجه حاصل برای بهینه سازی دودسته الگوهای غیر منظم با جمعیت اولیه ۱۰۰۰ و تعداد تولید نسل ۳۵۰ دارای راندمانی برابر با ۷۱،۶٪ شد (شکل ۲۴).

پیشنهاد می شود از روش الگوریتم ژنتیک موازی (PGA) [۸] برای بهینه چینی الگوها استفاده شود تا در زمان اجرای برنامه که در حال حاضر ۷ ساعت برای چیدن یک دسته الگو و بیش از ۱۵ ساعت برای چیدن دودسته الگو می باشد، به زمان قابل قبولی کاهش یابد و همچنین امکان استفاده از عملگرهای که به زمان و محاسبه زیادی دارند (مانند جهش ظاهری) فراهم شود. و امکان افزایش تعداد تولید نسلهای تا دستیابی به همگرایی نهایی حاصل شود.

منابع:

[۱]. S.Madarasmi. "Layout of Garment Patterns for Efficient Fabric Consumption", Department of Computer Engineering, Faculty of Engineering, King Mongkut's University of Technology Thonburi, 91 Pracha-Uthit Road, Bangkok 10140, Thailand , Tel. +66-2-470-9085, Fax.: +66-2-872-5050

[۲]. ص. اسماعیلی " الگوهای الکترونیک در طراحی الگوهای صنعتی نساجی، دانشگاه صنعتی امیرکبیر، ۱۳۷۸
[۳]. ح. عزیزی " الگوهای الکترونیک در طراحی الگوهای صنعتی نساجی، دانشگاه صنعتی امیرکبیر، ۱۳۷۹
[۴]. ج. جمشیدی. " الگوهای الکترونیک در طراحی الگوهای صنعتی نساجی، دانشگاه صنعتی امیرکبیر، ۱۳۸۰

[۵]. D.R..Magee,R.D. Boyle,"Building Shape Models from Image Sequences using Piecewise Linear Approximation". School of Computer Studies University of Leeds, Leeds LS2 9JT, UK, British Machine Vision Conference, drm@scs.leeds.ac.uk

[۶]. Joshi, S and Sudit, M., "Procedures for Solving Single Pass Strip Layout Problems," IIE Transactions, vol. 26, no. 1, pp.27-37, 1994.

[Y].S.Maouche, C.Bounsaythip "*OPTIMIZING TEXTILE SHAPE PLACEMENT BY TREE GENETIC ANNEALING*". Portland, Oregon, July 21-25 www.vtt.fi/tte/staff/bon/pub/scs96.pdf

[A]. A .Gmez , D.Fuente "*Resolution of strip-packing problems with genetic algorithms* "
University of Oviedo, Spain,1999.

[9]. Luke, S. and L. Spector. 1998. "*A Revised Comparison of Crossover and Mutation in Genetic Programming*". In Proceedings of the Third Annual Genetic Programming Conference (GP98). J. Koza et al, eds. 208-213. San Fransisco: Morgan Kaufmann