کاربرد الگوریتم ژنتیک برای حل مساله پوشش حداکثر

محسن صفاریان[1]

چکیده

مسایل پوشش حداکثر یکی از مهمترین مسایل مکانیابی هستند. از آنجایکه زمان حل آنها از یک تابع غیرچند جمله‌ای تبعیت می‌کنند لذا بزرگی ابعاد مسأله باعث افزایش زمان حل آنها می‌شود به این گونه مسایل، مسایل Np_Complete گفته می‌شود. روش‌های ابتکاری گوناگونی مانند الگوریتم لاگرانژ و گردی‌ادینگ برای حل آنها ارائه شده است. در چند سال اخیر الگوریتمهای ژنتیک کاربرد وسیعی در حل مسایل بهینه‌یابی پیدا نموده‌اند.

در این مقاله یک الگوریتم ژنتیک مناسب برای حل مدلهای Maximal Covering ارائه شده است. این الگوریتم را بر روی ۷۵ مسأله متفاوت اجرا نموده و نتایج آنرا با نتایج حاصل از دو الگوریتم لاگرانژ و گردی‌ادینگ بر روی همان مسایل مقایسه نموده‌ایم. همچنین نتایج حاصل ازاین الگوریتم را با نتایج حاصل از نرم‌افزار لینگو مقایسه کرده و از نظر میزان دقت و کارایی مورد بررسی و مطالعه قرار داده‌ایم. با توجه به بررسیهای انجام داده شده می‌توان گفت که الگوریتم طراحی شده دارای دقت و کارایی خوب و قابل قبولی می‌باشد.

کلمات کلیدی

الگوریتم ژنتیک ، مساله پوشش حداکثر ، مکانیابی

[1] عضو هیات علمی گروه مهندسی صنایع دانشگاه صنعتی بیرجند mo_saffarian@yahoo.com

۱. مقدمه

هرگاه بهینه‌سازی یک تابع یا یک فرایند مورد نظر باشد، طبیعتاً تکنیکها و روشهای گوناگونی جهت جستجو و بهینه سازی آن وجود دارد. همانگونه که می‌دانیم، اکثر روش‌های غیر تصادفی دارای این اشکال عمده هستند که به محض رسیدن به اولین نقطهٔ بهینه موضعی، متوقف شده و توانایی خروج از این نقطه و حرکت به سوی بهینه مطلق را ندارند. در این میان الگوریتم‌های تصادفی، بدلیل دارا بودن مکانیزم عملکرد ساده‌تر و در نتیجه راحتی اجراء بوسیلهٔ کامپیوتر مورد توجه بیشتری قرار گرفته‌اند. معمولاً برای هر دستهٔ خاص از مسائل بهینه‌سازی، الگوریتم‌های مخصوصی وجود داشته و الگوریتم‌های بهینه‌سازی عمومی بسیار نادر می‌باشند.

الگوریتم ژنتیک، یک گروه عمده از الگوریتم‌های تصادفی بوده و روشی برای بهینه‌سازی با جستجوی وسیع می‌باشد. این الگوریتم گر چه روشی بر مبنای جستجوی تصادفی است، اما ویژگیهای خاص آن موجب می‌شود نتوان آنرا جستجوی سادهٔ تصادفی قلمداد کرد. در این الگوریتم اطلاعات تاریخی از چگونگی تکامل شکلی کار، استخراج شده و در روند جستجو استفاده می‌شود.

از مزایای استفاده الگوریتم ژنتیک در مسایل بهینه‌سازی این است که قسمتهای مختلف فضای جستجو مورد کاوش قرار گرفته بنابراین امکان محدود شدن فضای جستجو وجود ندارد. از طرف دیگر در این روش محاسبات به طور دقیق انجام شده و هیچگونه تقریبی نظیر خطی‌سازی تابع هدف، گرد کردن نتایج و تغییر متغیرهای گسسته به پیوسته و بالعکس وجود ندارد، همچنین این روش بسیار ساده و قابل درک است.

۲. مدل مسایل پوشش حداکثر[۱](MCLP)

مدلهای موجود در زمینهٔ مکانیابی نمونه‌ای از مسایل برنامه‌ریزی ریاضی و OR می‌باشند. این دسته از مدلها جهت یافتن مکان مناسب برای نصب یا ایجاد تجهیزات، تأسیسات، و مراکز خدماتی بکار می‌روند. در عمل امکان اینکه تجهیزات موجود، کلیهٔ تقاضاها را تحت پوشش قرار دهد، وجود ندارد و این تجهیزات فقط جهت پوشش درصدی از تقاضاها کافی می‌باشد. در چنین حالتی انتظار می‌رود که تسهیلات در مکانهایی استقرار یابند که تعداد بیشتری از مشتری‌ها را پوشش دهند. **[۳][۴]**

مدل اصلی مسأله MC [۲] که اولین بار توسط چرچ و رول[۳] مطرح گردید، مدل MCLP نامیده می‌شود. در این مدل به مکانیابی P وسیله[۴] در یک شبکه برای پوشش دادن بیشترین مقدار تقاضا پرداخته می‌شود. شبکه‌ای را در نظر می‌گیریم که مقادیر تقاضا در گره‌های آن مشخص شده و فاصلهٔ (زمانی- مکانی) بین هر دو گروه نیز معین می‌باشد. در چنین شبکه‌ای فاصله بین دو گره با d_{ij} و تقاضای هر گره با h_i نمایش داده می‌شود. در مدل مذکور فرض می‌شود که تقاضاها فقط در گره‌های شبکه قرار دارند، البته این فرض از کلیت مسایل نخواهد کاست زیرا حتی اگر تقاضا در عمل به صورت متمرکز در گره‌ها نباشد(یعنی به صورت پراکنده در نقاط مختلف شبکه تعریف شود) می‌توان آنها را به صورت مجموعه‌ای از تعداد زیادی گره در نظر گرفت و همین مدل‌ها را برای آن‌ها بکار برد.

$$MAX \qquad Z = \sum_{i \in I} h_i Y_i$$

$$S.T \qquad Y_i \le \sum_{j \in J} a_{ij} X_j \qquad \forall i \in I$$

$$\sum_{j \in J} X_j = P$$

$$X_j = ۰, ۱ \qquad \forall j \in J$$

---

۱ Maximal covering location problems

۲ Maximal Covering

۳ Church & Revell

٤ Facility

$$Y_i = ۰, ۱ \qquad \forall i \in I$$

$$Y_i \le \sum_{j \in J} X_j \qquad \forall i \in I$$

در این مدل پارمترها به صورت ذیل تعریف می‌شوند:

I= مجموعه نقاطی که تقاضا بر آنها وجود دارد(گره‌ها)

J= مجموعه نقاط کاندید برای قرار گرفتن تجهیزات

h_i = i مقدار تقاضای موجود در گره

P= تعداد تجهیزات موجود برای مکان‌یابی

$$a_{ij} = \begin{cases} 1 & \text{اگر وسیله j تقاضای گره i را پوشش دهد یا i در شعاع پوشش j باشد} \\ 0 & \text{در غیر اینصورت} \end{cases}$$

S = شعاع پوشش

d_{ij} = فاصلهٔ گره i و j در روی شبکه

و متغیرهای تصمیم‌گیری مدل به صورت زیر تعریف می‌شوند:

$$X_j = \begin{cases} 1 & \text{اگر در نقطهٔ j وسیله‌ای قرار گیرد} \\ 0 & \text{در غیر اینصورت} \end{cases}$$

$$Y_i = \begin{cases} 1 & \text{اگر گره i توسط یک تجهیز یا بیشتر پوشش یابد} \\ 0 & \text{در غیر اینصورت} \end{cases}$$

با توجه به سایر پارامترها Y_i را می‌توان به این صورت تعریف نمود:

$$Y_i = \begin{cases} 1 & \sum_{j \in J} a_{ij} X_j \neq 0 \\ 0 & \sum_{j \in J} a_{ij} Xj = 0 \end{cases}$$

یعنی Y_j = ۰ خواهد بود اگر i در شعاع پوشش هیچ یک از نقاط کاندید نباشد و یا اینکه در آن نقاط که در شعاع پوشش i هستند وسیله‌ای قرار نگرفته باشد.

پارامترهای a_{ij} را به این صورت مقدار میدهیم که به ازای هر مکان کاندید j، فاصلهٔ آن مکان تا هریک از نقاط تقاضا را محاسبه کرده و هر فاصله‌ای که کمتر از شعاع پوشش (S) باشد، a_{ij} مربوط به آن را برابر یک قرار داده، در غیر این صورت به آن مقدار صفر اختصاص می‌دهیم. برای مقداردهی Y_i نیز $\sum_{j \in J} a_{ij} X_j$ را محاسبه کرده، در صورتی‌که مخالف صفر باشد Y_i را یک و درغیراین صورت آن را برابر صفر قرار می‌دهیم. به عبارت دیگر Y_i مربوط به نقطهi، در صورتی یک می‌باشد که حداقل یک وسیله، در مکانی که فاصله آن از نقطهٔ مزبور، کمتر از شعاع پوشش بوده، قرارگرفته باشد.

محدودیت اول بیان‌کننده این است که گره i ام زمانی پوشش یافته تلقی می‌شود که حداقل توسط یکی از تجهیزات پوشش یابد (در محدوده‌ی پوشش یکی از تجهیزات قرارگیرد) و محدودیت دوم نشان دهنده این است که ما تنها P تجهیز جهت استقرار در اختیار داریم. محدودیتهای سوم و چهارم باینری بودن دو متغیر تصمیم X_j, Y_i را بیان می‌کنند.

همانطور که ملاحظه شد، پارامتر a_{ij} را می‌توان به عنوان ورودی مسأله در نظر گرفته و آن را با توجه به فاصله نقاط تقاضا از نقاط کاندید و مقایسهٔ آن با شعاع پوشش مقداردهی نماییم.**[۳][۴]**

۳. ارائه الگوریتم ژنتیک طراحی شده

الگوریتم‌های ژنتیک از اجزای گوناگونی تشکیل می‌شود که در اینجا اجزاء تشکیل دهنده الگوریتم طراحی شده را به‌طور اجمال توضیح می‌دهیم:[۵]

الف) سیستم کدینگ[۱]

سیستم کدینگ به صورت دودویی انتخاب شده است.

ب) تولید جمعیت آغازین

جمعیت آغازین را به صورت تصادفی تولید می‌کنیم.

ج) عملگر تقاطعی[۲]

عملگر تقاطعی بکار رفته، عملگر تقاطعی دو نقطه برش می‌باشد.

د) عملگر جهشی[۳]

عملگر جهشی بکار رفته در الگوریتم به این صورت است که ابتدا در میان ژنهای صفر کروموزوم به دنبال ژنی هستیم که در صورت یک شدن، تعداد بیشتری از تقاضاها را پوشش دهد. پس از یافتن آن ژن، در میان ژن‌های با مقدار یک کروموزوم به دنبال ژنی می‌رویم که در صورت صفر شدن، تعداد کمتری از تقاضاها بدون پوشش باقی بماند. سپس صفر را تبدیل به یک و یک را تبدیل به صفر می‌کنیم. همانطور که می‌بینید در تعداد یک‌ها هیچ تغییری حاصل نمی شود؛ یعنی استراتژی اصلاح عملگر ژنتیک نیز اعمال شده است.[۶][۵]

ه) عمل تحول[۴]

در این قسمت از رویکرد انتخاب $(\mu+\lambda)$ استفاده شده است.

و) تابع برازش[۵]

تابع برازش را همان تابع هدف در نظر می گیریم.

ز) استراتژی برخورد با محدودیتها

استراتژی بکار رفته در هر چهار الگوریتم تلفیقی از استراتژی اصلاح عملگر ژنتیک[۶] و استراتژی ردی می‌باشد.[۱۰]

۴. تعیین پارامترهای مختلف الگوریتم

پس از انتخاب الگوریتم مورد نظر باید به تعیین پارامترهایی مانند تعداد نسل، اندازه جمعیت اولیه، مقدار عددی نرخ جهش و نرخ تقاطعی بپردازیم.هر یک از مراحل فوق را به‌طور مفصل توضیح می‌دهیم.

۱.۴. تعیین تعداد نسل

به منظور تعیین تعداد نسل مناسب، ابتدا ۱۵ مسأله مختلف که از نظر ابعاد آن با هم متفاوت هستند را در نظر گرفته و از هر کدام ۵ حالت گوناگون که در ضرایب تابع هدف و ماتریس A_{ij} متفاوت هستند را تولید می‌کنیم، در نتیجه ۷۵ مسأله مختلف در اختیار داریم، حال هر یک از آنها را با مشخصات زیر در نظر گرفته و اجرا می‌کنیم.

---

۱ Coding System

۲ Cross Over Operator

۳ Mutation Operator

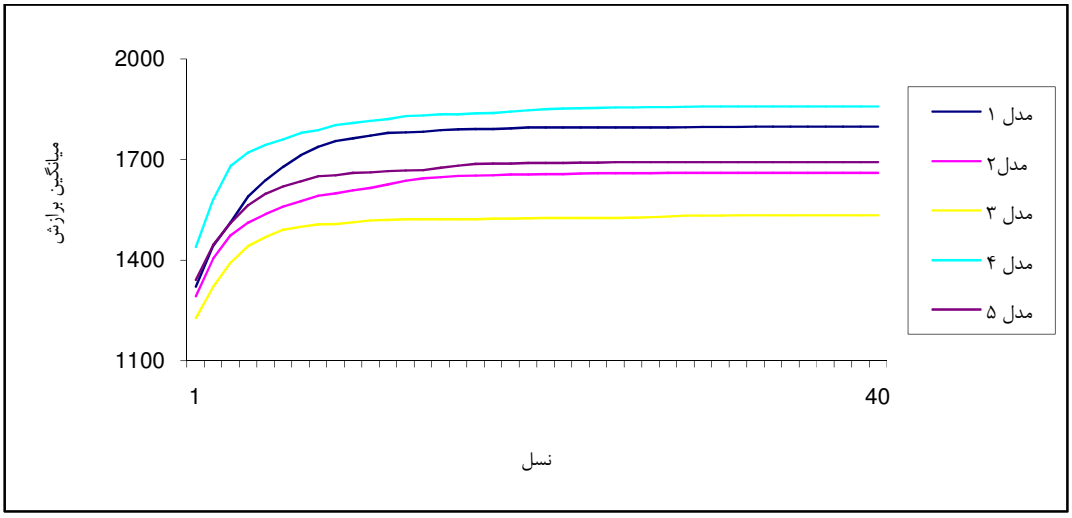
٤ Evolution Operation

٥ Fitness Function

٦ Modifying Genetic Operators Strategy

تعداد نسل = ۴۰ اندازه جمعیت = ۵۰ نرخ جهش = ۰/۹۵ نرخ تقاطع = ۰/۹۵

در هر اجرا متوسط مقدار برازش تابع هدف را به ازای ۵۰ کروموزوم نسل مربوطه را به‌دست می‌آوریم. پس از رسم نمودار مربوط به هر جدول، مشاهده می‌شود که پس از نسلهای ۱۵ به بعد تغییرات این مقدار بسیار ناچیز می‌باشد. لذا تعداد نسل را برای الگوریتم، ۲۰ نسل در نظر می‌گیریم. نمونه ای از آن در شکل ۱ آمده است.



شکل ۱: نحوه تاثیر افزایش نسل بر ثابت ماندن تابع برازش

۲.۴. تعیین اندازه جمعیت[۱]

برای تعیین اندازه جمعیت مناسب، ۶ مسأله مختلف را در نظر گرفته و با تغییر اندازه جمعیت، تأثیر آن بر کارایی مسأله را مورد بررسی قرار می‌دهیم، به این منظور اندازه جمعیت را ۱۰، ۱۵، ۲۰، ۲۵، ۳۰، ۴۰، ۵۰، ۷۵ و ۱۰۰ در نظر گرفته و برای هر یک از مسایل فوق آن‌را ۲۰ مرتبه اجرا کرده و جواب‌های آن‌را به‌دست می‌آوریم سپس میانگین ۲۰ جواب را محاسبه کرده و درصد خطای نسبی را برای آن مسأله محاسبه می‌کنیم. همانطور که در جدول ۱مشاهده می‌شود با افزایش اندازه جمعیت میزان درصد خطای نسبی کاهش یافته اما زمان اجرای برنامه افزایش می‌یابد. لذا با توجه به اینکه وقتی اندازه جمعیت اولیه ۵۰ است میزان خطای نسبی بسیار پایین و زمان اجرا نیز مناسب به نظر می‌رسد. بنابراین اندازه جمعیت اولیه را برابر ۵۰ در نظر می‌گیریم.

جدول(۱) تاثیر اندازه جمعیت بر میانگین خطای نسبی

اندازه جمعیت	۱۰	۱۵	۲۰	۲۵	۳۰	۴۰	۵۰	۷۵	۱۰۰
میانگین خطای نسبی	۵.۶۹	۳.۱۱	۱.۳۲	۱	۰.۲۸	۰.۱۷	۰.۰۱	۰	۰

۳.۴. تعیین نرخ جهش و نرخ تقاطع

به منظور تعیین نرخ جهش و نرخ تقاطع نیز ۴ مسأله مختلف در نظر گرفته و با تغییر دو پارامتر فوق اثر آن‌را بر روی جواب‌ها مورد بررسی قرار می‌دهیم. برای هر یک از دو پارامتر مذکور ۵ مقدار متفاوت ۰/۲، ۰/۴، ۰/۶، ۰/۸ و۰/۹۵ را اختیار کرده و ترکیب آنها را در حل مسأله به‌کار می‌بریم. مشاهده می‌شود که با افزایش نرخ جهش و نرخ تقاطع میزان خطای نسبی کاهش یافته ولی زمان افزایش می‌یابد، از آنجاییکه زمان اجرا در اولویت دوم قرار دارد بنابراین بهترین حالت برای دو پارامتر فوق این است که هر دو مقدار ۰/۹۵ را اختیار کنند.

---

۱ Population Size

با تعیین پارامترهای فوق الگوریتم مناسب طراحی شده است و باید آن‌را از نظر کارایی مورد آزمایش قرار دهیم به این منظور هر یک از مسایل را با استفاده از نرم‌افزار لینگو حل کرده و جواب بهینه را برای آن به‌دست می‌آوریم سپس آن‌را با مقدار حاصل از الگوریتم ژنتیک مقایسه می‌کنیم. همچنین الگوریتم ژنتیک فوق را با دو روش ابتکاری لاگرانژ[1] و گردی ادینگ [2]مقایسه می‌نماییم.[۸]

۵. مقایسه الگوریتم طراحی شده با نرم افزار لینگو

به منظور مقایسه الگوریتم طراحی شده با نرم افزار لینگو، ۷۵ مسأله مختلفی را که قبلا تولید نمودیم با لینگو حل کرده، زمان اجرا و جواب حاصل از آن‌را به‌دست می‌آوریم. حال هر یک از مسایل فوق را سه مرتبه با الگوریتم ژنتیک حل کرده، میانگین زمان اجرا و همچنین میانگین جواب حاصل از آن‌را محاسبه می‌کنیم.برای به‌دست آوردن متوسط درصد خطای نسبی از فرمول زیر استفاده می‌کنیم:

(۱)

$$\text{متوسط درصد خطای نسبی} = \frac{\sum_{i=1}^{3} \frac{(a\_b - a\_a_i)}{a\_b}}{3} \times 100\ \%$$

که در آن a_b جواب بهینه و a_a_i جواب حاصل از الگوریتم در اجرای i ام می‌باشد. پس از محاسبه عبارت فوق برای هر مسأله مشاهده می‌کنیم که بیشترین خطای نسبی حاصل از اجرای الگوریتم عبارت است از ۲/۰۵٪ و متوسط درصد خطای نسبی برای کلیه ۷۵ مسأله عبارت است از ۰/۱۰۵۸۶۶۶٪ که رقم بسیار ناچیزی بوده و می‌توان با اطمینان زیاد جواب حاصل از الگوریتم را به عنوان جواب بهینه در نظر گرفت. متوسط زمان اجرا نیز در طولانی‌ترین حالت عبارت است از ۸ ثانیه و ۴۴ صدم ثانیه و متوسط زمان اجرا برای تمام ۷۵ مسأله برابر است با ۲/۹۰۳۰۶۶۶ ثانیه،متوسط زمان اجرا برای ۷۵ مسأله فوق در نرم‌افزار لینگو برابر ۲/۹۰۶۶۶۶ می‌باشد که تقریبا برابر زمان اجرای الگوریتم ژنتیک می‌باشد.همانطور که ملاحظه کردید این الگوریتم دارای کارایی نسبتا خوبی بوده و اعتبار لازم را دارا می‌باشد.

۶. مقایسه الگوریتم ژنتیک با روش لاگرانژ

در اینجا الگوریتم طراحی شده را با الگوریتم لاگرانژ مقایسه می‌کنیم. به این منظور کلیه ۷۵ مسأله را به‌وسیله الگوریتم لاگرانژ سه مرتبه حل کرده، متوسط درصد خطای نسبی و متوسط زمان اجرای الگوریتم را به‌دست می‌آوریم. همانطور که مشاهده می‌شود در کلیه مسایل، متوسط درصد خطای نسبی و متوسط زمان اجرای الگوریتم لاگرانژ از الگوریتم ژنتیک طراحی شده بیشتر است.[۷]

مقدار میانگین کل درصد خطای نسبی برای ۷۵ مسأله حل شده به‌وسیله الگوریتم ژنتیک برابر ۰/۱۰۵۸۶۶۶٪ و برای الگوریتم لاگرانژ برابر ۲/۹۵۲۹۳۳۳٪ می‌باشد.یعنی خطای نسبی الگوریتم لاگرانژ بیشتر از الگوریتم ژنتیک می‌باشد.از نظر زمان اجرا نیز متوسط زمان اجرای مسایل فوق در الگوریتم ژنتیک برابر ۲/۹۰۳۰۶۶۶ ثانیه و در الگوریتم لاگرانژ برابر ۱۱/۵۵۰۶۶۶ می‌باشد که در نتیجه الگوریتم ژنتیک چه از نظر سرعت و چه از نظر دقت بهتر از الگوریتم لاگرانژ عمل می‌کند.

۷. مقایسه الگوریتم ژنتیک با روش گردی ادینگ

حال الگوریتم ژنتیک طراحی شده را با روش گردی ادینگ نیز مقایسه می‌کنیم.به این منظور تمامی ۷۵ مسأله قبل را ۳ مرتبه با این روش حل کرده و متوسط درصد خطای نسبی را برای هر کدام به‌دست می‌آوریم. همانطور که در جدول ۲ مشاهده می شود در اکثر موارد این مقدار نسبت به الگوریتم ژنتیک بالاتر بوده و خطای بیشتری دارد. مانند حالات قبل متوسط درصد خطای نسبی کل را برای گردی ادینگ محاسبه کرده و آن‌را به‌دست می‌آوریم. مشاهده می‌کنیم که متوسط درصد خطای نسبی در الگوریتم گردی ادینگ برابر ۱/۰۸۵۸۶۶۶٪ بوده که از مقدار ۰/۱۰۵۸۶۶۶٪ مربوط به الگوریتم ژنتیک بیشتر می‌باشد لذا میزان خطای روش گردی ادینگ از الگوریتم ژنتیک بیشتر است.[۹]

---

۱ Lagrangian

۲ Greedy Adding

جدول(۲) مقایسه الگوریتم ژنتیک، گردی ادینگ و لاگرانژ

ردیف	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵
تعداد نقاط تقاضا	۱۰	۲۰	۲۲	۲۵	۲۵	۳۵	۴۰	۴۰	۵۰	۵۰	۵۰	۵۶	۶۰	۶۰	۱۰۰
تعداد نقاط کاندید	۱۲	۲۰	۱۳	۱۵	۱۰۰	۴۰	۵۰	۶۵	۲۰	۴۰	۵۰	۱۹	۲۰	۴۰	۲۰
میانگین درصد خطای نسبی ژنتیک	۰	۰	۰	۰.۰۲	۰.۵۴	۰.۰۹	۰.۱۹	۰.۰۱	۰.۰۱	۰.۱۱	۰.۰۶	۰.۲۲	۰.۱	۰.۲۴	۰
میانگین درصد خطای نسبی گردی‌ادینگ	۱.۸۳	۰	۰.۸۵	۱.۵۵	۰.۷۷	۰.۷۶	۰.۵۶	۰.۹۵	۱.۴۱	۱.۷۴	۲.۲۳	۲.۴۵	۰.۶۵	۰.۳۸	۰.۱۶
میانگین درصد خطای نسبی لاگرانژ	۱.۲۹	۱.۹۹	۱.۹۱	۲.۰۱	۷.۶۳	۳.۴۵	۴.۶۶	۱.۷۶	۲.۸	۴.۴۲	۲.۳۸	۳.۴۲	۱.۲۷	۳.۲۳	۱.۶۹

۸. نتیجه‌گیری

همانطور که گفته شد الگوریتم ژنتیک با مجموعه‌ای از جواب‌ها شروع کرده و با انجام عملیات مناسب بر روی آنها این مجموعه را بهبود می‌دهد.به‌طوریکه در انتها با احتمال بالایی می‌توان مطمئن بود که جواب بهینه در نهای‌ت به‌دست خواهد آمد. با گذشت زمان و ایجاد نسلهای جدید میانگین مقدار برازش کروموزوم‌ها سیر صعودی داشته و در نسلهای پایانی شتاب این صعود کاهش می‌یابد؛ یعنی جواب‌های جدیدی وارد مجموعه جواب‌ها نمی‌شود. با اجرای الگوریتم بر روی ۷۵ مسأله متفاوت ملاحظه کردیم که متوسط درصد خطای نسبی الگوریتم برابر ۰/۱۰۵۸۶۶٪ بود که تقریبا قابل صرف نظر بوده و می‌توان مطمئن بود که الگوریتم جواب بهینه را به ما می‌دهد. بحث دیگری که در اینجا باید به آن توجه کرد این است که الگوریتم فوق هیچ محدودیتی روی ابعاد مسأله نداشته و فقط محدودیتهای سخت افزاری مطرح می‌باشد که این امر یک بحث طبیعی و عمومی بوده و با افزایش امکانات سخت افزاری می‌توان هر مسأله بزرگی را حل نمود.

با افزایش تعداد نسل یا اندازه جمعیت می‌توان احتمال رسیدن به جواب بهینه را افزایش داده و از بهینه بودن جواب اطمینان حاصل کرد.

۹. مراجع

[۱] AL-Sultan, K.S., Hussain, M.F., and Nizami, J.S., " *A Genetic algorithms for the set covering Problems*", Jornal of the Operational Research Society (۱۹۹۶) ۴۷: ۷۰۲-۷۰۹

[۲] Balakrishnan, J., and Cheng, C.H., P.C., " *Genetic Serarch and the Dynamic Layout Problem*", Computers & Operations research. (۲۰۰۰) ۲۷:۵۸۷-۵۹۳.

[۳] Chung, C.H., "*Recent Aplication of the Maximal Covering Location Planing Model*". Operational Research Society (۱۹۸۶) ۳۷:۷۳۵-۷۴۶.

[۴] Church, R.L., and Revelle, C.S., " *The Maximal Covering Location Problem*", Papers of the Regional science association (۱۹۷۴) ۳۲:۱۰۱ – ۱۱۸

[۵] Mitchall, M., "*An Introduction to Genetic Algorithms*", London, England,(۱۹۹۹)

[۶] Tavakokoli-moghadam, R., and Shayan, E., "*Facilities Layout Design by Genetic Algorithms*", Computers Industrial Engineering.(۱۹۹۸). ۳۵:۵۲۷-۵۳۰.

[۷] Haddadi, S.,"*Simple Lagrangian Heuristic for the Set Covering problems*", European Journal of Operational Research (۱۹۹۷) ۹۷: ۲۰۰- ۲۰۴.

[۸] Arakaki, R.G., and Lorena, L.A., "*A Constructive Genetic Algorithms for the Maximal Covering location Problems*", ۴ th Metaheuristics International Conference.(۲۰۰۱).

[۹] Donnell, R., " *Definitions Greedy Algorithm for Set_cover & Max Coverage*", (۲۰۰۸).

[۱۰] علی حسینی، احمد رضا." **بکارگیری الگوریتم ژنتیک برای حل مساله پوشش مجموعه**"، پایان نامه کارشناسی ارشد، دانشگاه تهران، دانشکده فنی و مهندسی، ۱۳۷۹