Archive of SID

مجله بین المللی علوم مهندسی، شماره ۴، جلد۱۶، زمستان ۱۳۸۴، صفحه ۱۲۴- ۱۱۵
دانشگاه علم وصنعت ایران

طراحی مدل توسعه‌یافته برای مسأله پوشش مجموعه و حل آن با الگوریتم ژنتیک (GA)

سید حسام‌الدین ذگردی، محمد مهدی سپهری، رضا توکلی مقدم و ابوالفضل سمیعی

چکیده: در این مقاله، یک مدل برنامه‌ریزی ریاضی برای مسأله پوشش مجموعه (SCP)، جهت نزدیک‌تر کردن مدل به بعضی از مسایل دنیای واقع، ارایه می‌شود. در این مدل قرار است m مشتری توسط n تسهیل پوشش یابند به نحوی که فاصله استقرار تسهیلات از یک حد مجاز کمتر نباشد. به دلیل غیر خطی بودن مدل مورد نظر، روش‌های سنتی بهینه‌سازی در مدت زمانی معین قادر به حل مسأله نمی‌باشند، لذا از یک الگوریتم ژنتیک به عنوان یک روش فراابتکاری برای حل مدل استفاده می‌شود. برای بررسی کارایی الگوریتم پیشنهادی، ۳۶ مسأله نمونه تصادفی در ۹ گروه با ابعاد مختلف مورد آزمایش قرار می‌گیرد و نتایج بدست آمده از الگوریتم ژنتیک با نرم‌افزار Lingo 6 مقایسه می‌گردد. نتایج نشان می‌دهد که الگوریتم طراحی شده دارای اعتبار بوده و کارایی لازم را در حل مسایل مذکور دارا می‌باشد.

واژه‌های کلیدی: مسأله پوشش مجموعه، الگوریتم ژنتیک، بهینه موضعی، بهینه کلی

۱. مقدمه

در مساله پوشش مجموعه[1] (SCP)، مجموعه‌ای از مشتری‌ها با تقاضای مشخص برای خدمات یا کالاها وجود دارند که باید توسط تعدادی تسهیل خدمت داده شوند و این کار با هدف کمینه کردن هزینه صورت می‌گیرد. اساسی‌ترین مدل ریاضی مطرح شده در مورد مسأله پوشش مجموعه را می‌توان بصورت زیرا ارایه نمود[1]:

$$\min \sum_{j=1}^{n} C_j X_j \qquad (۱)$$

s.t.

$$\sum_{j=1}^{n} a_{ij} X_j \geq 1 \qquad i = 1,2,\ldots,m$$

$$X_j = (0,1) \qquad j = 1,2,\ldots,n$$

C_j: هزینه استقرار تسهیلات در نقطه j ام

$$a_{ij} = \begin{cases} 1 & \text{اگر تسهیل مستقر شده در مرکز j بتواند مشتری i را پوشش دهد} \\ 0 & \text{در غیر این صورت} \end{cases}$$

$$X_{ij} = \begin{cases} 1 & \text{اگر تسهیل در مرکز j مستقر شود} \\ 0 & \text{در غیر این صورت} \end{cases}$$

در مدل فوق X_j متغییر تصمیم آن است، m مشتری و n تسهیل وجود دارد، C_j و a_{ij} نیز ورودی‌های مسأله هستند. SCP در حقیقت پوشش سطرهای یک ماتریس صفر ـ یک m سطری و n ستونی، بوسیله زیر مجموعه‌ای از ستون‌هاست بطوری‌که حداقل هزینه‌ها حاصل شود. تابع هدف مسأله هزینه مستقر کردن تسهیلات مورد نیاز را کمینه می‌کند. محدودیت (۲) تضمین می‌کند که هر مشتری بوسیله حداقل یک تسهیل پوشش داده می‌شود. محدودیت (۳) نیز نشان می‌دهد که مدل جزء مسایل برنامه‌ریزی خطی صفر - یک می‌باشد.

مسأله پوشش مجموعه در بسیاری از مسایل دنیای واقع مانند زمان-بندی کارکنان خطوط هوایی، استقرار ایستگاه‌های آتش‌نشانی، استقرار رادارها، استقرار مراکز پلیس، بیمارستان‌ها، کتابخانه‌ها،

مقاله در تاریخ ۱۳۸۲/۳/۲۷ دریافت شده و در تاریخ ۱۳۸۳/۳/۳۱ به تصویب نهایی رسیده است.

سید حسام‌الدین ذگردی دانشیار مهندسی صنایع دانشکده فنی و مهندسی دانشگاه تربیت مدرس، zegordi@modarres.ac.ir
محمد مهدی سپهری دانشیار مهندسی صنایع دانشکده فنی و مهندسی دانشگاه تربیت مدرس، mehdi.sepehri@modarress.ac.ir
رضا توکلی مقدم دانشیار مهندسی صنایع دانشکده فنی دانشگاه تهران، tavakoli@ut.ac.ir
ابوالفضل سمیعی کارشناس ارشد مهندسی صنایع، Samiei55@yahoo.com

[1] Set Covering Problem

www.SID.ir

Archive of SID

ادارات پستی، محل‌های تحویل کالا، طراحی و ساده‌سازی مدارهای سوئیچی، بالانس خط مونتاژ، مسیریابی وسایل نقلیه، زمان‌بندی کشتی‌ها، شبکه‌های دفاعی و تهاجمی، زمانبندی رانندگان اتوبوس، بازیابی اطلاعات، تولید، سرمایه‌گذاری، انتخاب پروژه، تخصیص ترافیک در سیستم‌های ارتباطی ماهواره‌ای، محاسبه حدود در برنامه‌ریزی عدد صحیح وغیره کاربرد دارد [2,3,4].

۱-۱. تحقیقات انجام شده

فیشر و کدیا [5] دو مسأله SCP و SPP[1] را بطور توأم مورد توجه قرار دادند و یک الگوریتم بهینه بر اساس ابتکاری همزاد[2] ارایه دادند. آنها مسایل SCP را تا ابعاد ۲۰۰ سطر و ۲۰۰۰ ستون مورد بررسی قرار دادند.

بیزلی و جرستن[2] الگوریتمی بهینه برای حل SCP ارایه کرده‌اند. الگوریتم آنها برای مسایلی با ابعاد ۴۰۰ سطر و ۴۰۰۰ ستون بکار می‌رود. ال - دارزی و میترا [6] از جمله افرادی هستند که در این زمینه الگوریتم دقیق ارایه داده‌اند. لوپز و لورنا [3] یک روش ابتکاری تحت عنوان ابتکاری با جانشین[3] (SH) ارایه کردند. الگوریتم آنها برای مسایلی تا ۱۰۰۰ سطر و ۱۲۰۰۰ ستون جواب‌هایی با کیفیت بالا حاصل می‌کرد. حدادی[7] یک روش ابتکاری لاگرانژی ساده برای SCP توسعه داد. این الگوریتم برای مسایلی با چگالی پایین و با ابعاد ۱۰۰۰ سطر و ۱۰۰۰۰ ستون کارا می‌باشد (چگالی برابر نسبت یک‌ها در ماتریس ضرایب محدودیت‌های مدل می‌باشد $\rho = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}$).

ژاکوبز و بروسکو [8] یک الگوریتم فوق ابتکاری بر اساس Simulated Annealing (SA) ارایه دادند. SA آنها در حل مسایلی با ابعاد ۱۰۰۰ سطر و ۱۰۰۰۰ ستون موفقیت قابل توجهی را بدست آورد.

السلطان و همکارانش[9]، بیزلی [10] و لورنا و لوپز [11] از الگوریتم ژنتیک[4] (GA) برای حل مدل استفاده نموده‌اند.

منفرگلیو [12]، هیفی و همکارانش [13] و اهلسن و همکاران [14] الگوریتم‌های فراابتکاری بر اساس شبکه‌های عصبی برای حل SCP طراحی کردند.

فرانسیس و فلوید[15] یک الگوریتم فرا ابتکاری به نام (HC)[5] را برای SCP بکار بردند که این یک الگوریتم دو مرحله‌ای می‌باشد. در مرحله اول راه حل‌های بهینه موضعی متنوعی ایجاد می‌شود و در مرحله دوم این جواب‌ها برای تعیین مسأله متمرکز شده بکار می‌رود، بطوریکه این مسأله به اندازه کافی کوچک است و می‌تواند بطور بهینه حل شود. سلار و همکارانش [4] یک الگوریتم ژنتیک موازی[6] برای حل مسأله پوشش مجموعه ارایه کردند.

این تحقیقات اهم کارهایی بودند که تاکنون بر روی مدل اصلی مسأله پوشش مجموعه صورت گرفته است. حالت خاصی از مدل اصلی یعنی آن هنگام که هزینه استقرار برای کلیه تسهیلات یکسان است نیز کاربرد فراوانی دارد. بعضی محققین کارهای خود را صرفا بر روی این حالت متمرکز کرده‌اند. المینانا و پاستور[6] یک روش ابتکاری با جانشین SH برای مسأله جانمایی SC ارایه دادند.

در تحقیقات انجام گرفته توسط محققین در زمینه مساله پوشش مجموعه، مسایلی با ابعاد مختلف مورد توجه قرار گرفته‌اند ولی آنها یا مدل اصلی پوشش مجموعه را مورد بررسی قرار داده‌اند یا حالت خاصی از مدل که هزینه استقرار را برای همه تسهیلات یکسان در نظر می‌گیرد. این دو حالت مدل دارای این خصوصیت هستند که نقاط بهینه برای استقرار تسهیلات را بدون هیچ توجهی به نقاطی که در حال حاضر در آنجا تسهیل وجود دارند، مشخص می‌کنند. در این مقاله مساله پوشش مجموعه را با توجه به نکته ذکر شده، توسعه داده و برای حل آن یک الگوریتم ژنتیک پیشنهاد داده می‌شود.

۲-۱. الگوریتم ژنتیک

الگوریتم ژنتیک عمومی‌ترین نوع شناخته شده الگوریتم‌های تکاملی است. الگوریتم ژنتیک، یکی از مهم‌ترین الگوریتم‌های ابتکاری می‌باشد که از آن برای بهینه‌سازی توابع مختلف استفاده می‌شود. در این الگوریتم اطلاعات گذشته با توجه به موروثی بودن الگوریتم استخراج شده و در روند جستجو مورد استفاده قرار می‌گیرد[16].

ابتدا توسط هالند[17] یک مفهوم از الگوریتم ژنتیک ارایه شد و سپس گلدبرگ[18] آن را توصیف کرد. الگوریتم‌های ژنتیک تکنیک‌های جستجوی تصادفی هستند که بر اساس انتخاب طبیعی و نسل‌شناسی طبیعی کار می‌کنند[19]. ساختار کلی یک الگوریتم ژنتیک را می‌توان به صورت ذیل تصور کرد:

ابتدا و پیش از هر چیز باید مکانیسمی برای تبدیل هر جواب مسأله به یک کروموزوم تعریف کرد. پس از آن یک مجموعه از کروموزوم‌ها که در حقیقت مجموعه‌ای از جواب‌های مسأله هستند به عنوان یک جمعیت آغازین[7] تهیه می‌گردند. این مجموعه که اندازه آن دلخواه است و توسط کاربر تعیین می‌شود، اغلب به صورت تصادفی ایجاد می‌گردد. بعد از این مرحله باید با بکارگیری اعمال ژنتیک[8] اقدام به ایجاد کروموزوم‌های جدید موسوم به اولاد[9]، نمود. این اعمال به دو گونه عمده تقاطعی و جهشی تقسیم‌بندی می‌شوند. همچنین برای گزینش کروموزوم‌هایی که باید نقش والدین[10] را بازی کنند، دو

[1] Set Partitioning Problem
[2] Dual Heuristic
[3] Surrogate Heuristic
[4] Genetic Algorithm
[5] Heuristic Concentration
[6] Parallel Genetic Algorithm
[7] Initial Population
[8] Genetic Operations
[9] Offspring
[10] Parents

www.SID.ir

Archive of SID

مفهوم نرخ تقاطعی[1] و نرخ جهشی[2] کاربری فراوانی دارند، که این دو نیز پیش از شروع الگوریتم توسط کاربر تعیین می‌شوند.

بعد از تولید یکسری کروموزوم جدید یا اولاد باید با استفاده از عمل تحول[3] اقدام به انتخاب برازنده‌ترین کروموزوم‌ها نمود. این عمل که در فرآیند انتخاب[4] نمود می‌یابد، گلچین کردن کروموزوم‌های برازنده در میان والدین و اولاد است بطوریکه تعداد کروموزوم‌های منتخب برابر با اندازه جمعیت اولیه باشد. فرآیند انتخاب مبتنی بر مقدار برازندگی[5] هر رشته است. در حقیقت فرآیند ارزیابی[6] محوری‌ترین بحث در فرآیند انتخاب است. تا بدین مرحله یک تکرار یا یک نسل[7] از الگوریتم طی شده است. الگوریتم بعد از طی چندین نسل به تدریج به سمت جواب بهینه همگرا می‌شود. شرط توقف مسأله نیز طی کردن تعداد معینی تکرار است که پیش از آغاز الگوریتم توسط کاربر تعیین می‌شود. به طور مثال هنگامی که تعداد نسل‌ها از ۲۰ تجاوز کند جمعیت بسیار شبیه هم شده و ادامه الگوریتم بهبودی در جواب‌ها حاصل نخواهد کرد. بنابراین قبل از اجرای الگوریتم شرطی گذاشته می‌شود که الگوریتم حداکثر تا ۲۰ نسل ادامه یابد.

۲. مدل پوشش مجموعه توسعه‌یافته

توسعه مسأله پوشش مجموعه، باعث نزدیک‌تر شدن مدل به بعضی از مسایل دنیای واقع شده است. بطوریکه باعث بوجود آمدن تغییراتی در تابع هدف و محدودیت‌های مدل اصلی شده است.

از مواردی که در دنیای واقع با آن برخورد می‌کنیم تعیین مکان مناسب نمایندگی‌های مجاز ایران‌خودرو و تعیین مکان داروخانه در سطح یک شهر می‌باشد. به طور مثال در مسأله تعیین مکان نمایندگی‌ها در سطح شهر، در مدل ارایه شده تابع هدف طوری در نظر گرفته شده است که نقاطی که در حال حاضر درآنجا نمایندگی وجود دارد از شانس بیشتری برای انتخاب شدن به عنوان جواب بهینه برخوردارند و به آنها ارزش بیشتری داده می‌شود و بین این نقاط با نقاطی که جدیداً می‌خواهند وارد سیستم شوند، تفاوت قایل می‌شود. همچنین حداقل فاصله بین نقاط جواب نیز که در مسأله تعیین مکان نمایندگی‌ها مطرح می‌شود، در محدودیت‌های مدل لحاظ شده است.

۲-۱. تعریف متغیر و پارامترها

در مدل مورد بحث X_j متغیر تصمیم است که یک متغیر عدد صحیح صفر و یک می‌باشد m مشتری و n تسهیل وجود دارد. d'_{ij}، a_{ij} و D نیز ورودی‌های مسأله هستند. کلیه متغیرها و پارامترها به صورت زیر تعریف می‌شوند:

a_{ij} ها به صورت صفر یا یک درنظر گرفته می‌شوند و با توجه به فاصله گره i از گره j (d_{ij}) و فاصله پوشش (d_c) تعیین می‌شوند. اگر فرض کنیم که یک تسهیل در گره j مستقر شود و یک مشتری در گره i وجود داشته باشد و رابطه $d_{ij} \le d_c$ برقرار باشد آنگاه $a_{ij} = 1$ (یعنی تسهیل مستقر شده در گره j می‌تواند مشتری موجود در گره i را پوشش دهد) و در غیر اینصورت $a_{ij} = 0$ خواهد بود.

به عنوان مثال اگر سه گره وجود داشته باشد، فاصله پوشش $d_c = 10$، و ماتریس فاصله بین این سه گره به صورت زیر باشد:

$$\left[d_{ij}\right] = \begin{bmatrix} 0 & 4 & 9 \\ 4 & 0 & 12 \\ 9 & 12 & 0 \end{bmatrix}$$

با توجه به نکات فوق، ضرایب a_{ij} در محدودیت‌ها به صورت زیر معین می‌شوند:

$$\left[a_{ij}\right] = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

$$a_{ij} = \begin{cases} 1 & \text{اگر تسهیل مستقر شده در مرکز j بتواند مشتری i را پوشش دهد} \\ 0 & \text{در غیر این صورت} \end{cases}$$

J_c: مجموعه نقاط موجود که در حال حاضر تسهیلات در آنها مستقر است

J_n: مجموعه نقاط مستقر برای استقرار تسهیلات

ε : یک عدد کوچک مثبت

d_{ij}: فاصله گره تقاضای i از گره کاندیدای تسهیل j

d'_{ij} : فاصله گره کاندیدای تسهیل i از گره کاندیدای تسهیل j

d_c : فاصله پوشش

D: حداقل فاصله مجاز بین محل استقرار تسهیلات

$$X_j = \begin{cases} 1 & \text{اگر تسهیل در مرکز j مستقر شود} \\ 0 & \text{در غیر این صورت} \end{cases}$$

۲-۲. مدل طراحی شده

$$\text{Min } q = \sum_{j \in J_c} X_j + (1+\varepsilon) \sum_{j \in J_n} X_j \qquad (۴)$$

s.t.

$$\sum_{j=1}^{n} a_{ij} X_j \ge 1 \qquad i = 1,2,\ldots,m \qquad (۵)$$

$$(d'_{ij} - D) X_i X_j \ge 0 \qquad i<j,\ i=1,2,\ldots,n \qquad (۶)$$

$$X_j = (0,1) \qquad j = 1,2,\ldots,n \qquad (۷)$$

که در مدل فوق X_j متغیر تصمیم است و m مشتری و n تسهیل وجود دارد. a_{ij} ,d'_{ij} ,D نیز ورودی‌های مسأله هستند. هزینه استقرار

[1] Crossover Rate
[2] Mutation Rate
[3] Evolution Operation
[4] Selection
[5] Fitness Value
[6] Evaluation
[7] Generation

www.SID.ir

Archive of SID

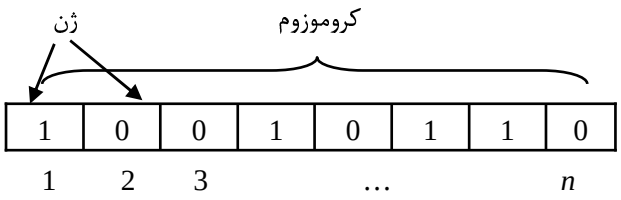
تسهیلات در نقاط مختلف با هم برابر فرض شده است، به عبارت دیگر در مدل C_jها با هم برابر بوده و می‌توان از نمایش آن در مدل صرفه‌نظر کرد. تابع هدف نیز از جنس هزینه می‌باشد به طوری که با توجه به نقاطی که در آنجا تسهیل مستقر می‌شود مجموع هزینه‌ها محاسبه می‌شود (به ازای استقرار تسهیل در هر نقطه هزینه ۱ به آن تعلق می‌گیرد). کلیه متغیرها و پارامترها در قسمت مربوطه تعریف شده‌اند. بطور کلی در مدل اصلی مسأله پوشش مجموعه، بدون در نظر گرفتن مکان‌هایی که در حال حاضر در آنجا تسهیلات مستقر می‌باشد، مکانهای بهینه نقاط استقرار تسهیلات را مشخص می‌کند. در اینجا مسأله طوری مدل می‌شود که از نقاطی که در حال حاضر در آنجا تسهیلات مستقر است، حداکثر استفاده بشود و همچنین هزینه استقرار تسهیلات جدید نیز حداقل شود بطوری‌که با استقرار آنها به تمامی تقاضاها پاسخ داده شود. ولی برای اینکه از تسهیلات موجود استفاده حداکثر بشود بایستی تابع هدف به نحوی تغییر کند تا بتواند اهمیتی برای نقاط فعلی موجود در سیستم برای استقرار تسهیلات قایل باشد. برای این منظور به نقاطی که جدیداً به سیستم وارد می‌شود، وزن به صورت عددی مثبت و کوچک اضافه می‌شود. به این ترتیب در شرایط برابر برای استقرار تسهیل جدید و یا وجود تسهیل کنونی، تسهیل کنونی ارجحیت داشته و به عنوان جواب مسأله نمایش داده می‌شود. همچنین یکی از محدودیت‌هایی که در بعضی از مسایل واقعی با آن برخورد می‌کنیم، محدودیت فاصله تسهیلات از یکدیگر می‌باشد. با این توضیح که فاصله تسهیلات از یکدیگر نباید از یک حد مجاز کمتر باشد که این محدودیت نیز به مدل اصلی اضافه شده است. محدودیت (۵) تضمین می‌کند که هر مشتری بوسیله حداقل یک تسهیل پوشش داده شود. محدودیت (۶) نیز باعث می‌شود فاصله تسهیلات استقرار یافته از یکدیگر حداقل D باشد همچنین این محدودیت که به صورت حاصلضرب دو متغیر می‌باشد باعث می‌شود مسأله جزء مسایل برنامه‌ریزی غیرخطی قرار بگیرد. محدودیت (۷) نیز نشان می‌دهد که مدل جزء مسایل برنامه‌ریزی غیر خطی صفر - یک می‌باشد. تعداد تسهیلات لازم و همچنین مکان استقرار آنها خروجی این مدل هستند بطوری‌که به کلیه تقاضای مشتریان پاسخگو بوده و فاصله بین آنها نیز از یک حد مجاز کمتر نخواهد بود.

۳. طراحی الگوریتم ژنتیک

با توجه به ماهیت مدل ارایه شده، الگوریتم ژنتیک مناسبی برای حل آن طراحی شده است. اجزای تشکیل دهنده الگوریتم طراحی شده در ذیل آمده است.

۳-۱. نحوه نمایش جواب

اولین گام در بکارگیری و پیاده‌سازی یک الگوریتم ژنتیک، نمایش جواب‌های مسأله بصورت یک کروموزوم است. متغیرهای تصمیم‌گیری در مدل بصورت متغیرهای صفر و یک تعریف شده‌اند، بطوریکه در صورت استقرار تسهیل در یک نقطه، متغیر مربوط به آن مقدار یک و در غیر اینصورت مقدار صفر را اتخاذ خواهد کرد. با توجه به این موضوع هر جواب مسأله بصورت رشته‌ای حاوی n مکان (ژن) در نظر گرفته شده است بطوری‌که n بیانگر تعداد مراکزی است که قابلیت استقرار تسهیلات در آنها وجود دارد. این سیستم کدینگ در واقع کدینگ رشته‌ای دودویی می‌باشد که هر ژن فقط می‌تواند مقادیر 0 یا 1 را بگیرد.



شکل ۱. نمایش مجموعه جواب‌های مسأله به صورت کروموزوم

۳-۲. ایجاد جمعیت اولیه

اولین مرحله بعد از تعیین یک سیستم کدینگ، ایجاد یک جمعیت اولیه از کروموزوم‌هاست. در الگوریتم طراحی شده، جمعیت‌های اولیه مختلفی آزمایش شده و در نهایت مناسب‌ترین جمعیت به تعداد ۵۰ انتخاب گردیده است. جمعیت اولیه به صورت تصادفی تولید می‌شود به طوری‌که بعد از تولید هر کروموزوم شدنی بودن آن نیز بررسی می‌شود.

۳-۳. برخورد با محدودیت‌ها

یکی از مسایل مهم در اجرای الگوریتم ژنتیک چگونگی سروکار داشتن با محدودیت‌هاست. زیرا در حین تولید جمعیت اولیه و همچنین هنگامی که عملگرها، کروموزوم‌های جدید تولید می‌کنند، در بسیاری موارد کروموزوم‌های تولیدی غیر موجه می‌باشند. در این الگوریتم در صورت تولید جواب ناشدنی، بلافاصله جواب حذف نمی‌شود بلکه سعی در اصلاح آنها دارد. همانطور که ذکر گردید ژن‌های این کروموزوم اعداد صفر و یک می‌باشند و ارضا نکردن محدودیت‌ها به معنای این است که باید مقدار بعضی از ژن‌ها در کروموزوم عوض شود یعنی اگر مقدار صفر دارند به یک و اگر مقدار یک دارند به صفر تغییر یابد.

۳-۴. تابع برازندگی

همانطور که می‌دانیم تابع برازش، شاخصی جهت ارزیابی کروموزوم‌ها، برای تعیین کروموزوم‌های مناسب‌تر می‌باشد. در این

www.SID.ir

Archive of SID

الگوریتم تابع برازش در واقع همان تابع هدف مدل ارایه شده است.

۳-۵. استراتژی انتخاب والدین

بعد از تولید هر نسل باید تعدادی والدین برای تولید نسل بعدی از بین کروموزوم‌های نسل جاری انتخاب شوند. این استراتژی باید به گونه‌ای باشد که شایسته‌ترین والدین برای تولید فرزندان نسل بعد انتخاب شوند. به این دلیل بعد از تولید هر نسل ابتدا برازندگی کروموزوم‌های آن نسل بصورت رابطه زیر نرمالیزه می‌شود:

$$Z_i = \frac{f_i - \mu}{\delta} \qquad (۸)$$

بطوری‌که f_i بیانگر برازندگی i امین کروموزوم نسل جاری و Z_i نرمال شده f_i، μ میانگین برازندگی نسل و δ برابر انحراف معیار برازندگی نسل می‌باشد. سپس نیمی از جمعیت که مقدار برازندگی نرمالیزه شده آنها کوچکتر یا مساوی صفر می‌باشد به عنوان والدین نسل بعدی انتخاب می‌شوند. زیرا در این مدل کمینه‌سازی مسأله مورد نظر است بنابراین کروموزوم‌های دارای برازش کمتر، مناسب‌تر خوانده می‌شوند.

۳-۶. عملگرهای ژنتیکی

همانطور که می‌دانیم یک بخش مهم در الگوریتم ژنتیک ایجاد کروموزوم‌های جدید موسوم به اولاد از طریق بعضی کروموزوم‌های قدیم موسوم به والدین است. این فرآیند مهم توسط اعمال ژنتیک صورت می‌گیرد. در این الگوریتم نیز از سه عملگر به نام‌های عملگر تقاطعی[1]، عملگر جهشی[2] و عملگر وارونگی[3] استفاده شده است. در ذیل نحوه عملکرد هر کدام از آنها در الگوریتم توضیح داده می‌شود.

۳-۶-۱. عملگر تقاطعی

عملگر تقاطعی در یک زمان بر روی دو کروموزوم پیاده می‌شود و اولاد بوسیله ترکیب ساختار دو کروموزوم ایجاد می‌گردد. در این الگوریتم از روش یک نقطه برش استفاده شده است. یعنی ابتدا در طول دو کروموزومی که به عنوان والدین در نظر گرفته شده‌اند یک نقطه تصادفی انتخاب می‌شود که این نقطه همان نقطه برش می‌باشد و کروموزوم‌ها از آن نقطه به دو قسمت تقسیم شده و عمل جابجایی بین آنها صورت می‌پذیرد. بدین ترتیب دو کروموزوم جدید حاصل می‌شود.

۳-۶-۲. جهش

عملگر جهشی هر زمان بر روی یک کروموزوم اعمال می‌شود. در این الگوریتم از عملگر جهشی یکنواخت استفاده شده است. یعنی پس از انتخاب کروموزوم مورد نظر، یک (یا چند) ژن از آن بصورت تصادفی انتخاب شده ومقدار آن اگر 1 باشد به 0 و اگر 0 باشد به 1 تغییر می‌یابد و با این تغییر کروموزوم فرزند حاصل می‌شود.

۳-۶-۳. معکوس

عملگر معکوس نیز همچون عملگر جهشی بر روی یک کروموزوم اعمال می‌شود. برای اجرای این عملگر ابتدا یک زیر رشته از کروموزوم به تصادف انتخاب شده، سپس مقادیر ژن‌های آن از ابتدا به انتها معکوس می‌شود.

۳-۷. معیار توقف

معیارهایی که جهت توقف الگوریتم درنظر گرفته شده‌اند عبارتند از:

- تعداد نسل‌ها؛ هرگاه تعداد نسل‌ها از مقدار مشخصی تجاوز کند آنگاه الگوریتم متوقف خواهد شد. در الگوریتم ارائه شده تعداد نسل‌ها برابر ۲۰ در نظر گرفته شده است.
- واریانس جمعیت؛ هرگاه واریانس برازندگی در هر نسل از مقدار تعیین شده کمتر باشد الگوریتم توقف خواهد کرد. در الگوریتم طراحی شده هرگاه واریانس برازندگی در یک نسل به ۰/۰۱ برسد، شرط توقف حاصل می‌گردد.

۴. طراحی آزمایشات

در این قسمت برای بررسی کارایی الگوریتم طراحی شده، مسایل مختلفی بطور تصادفی تولید شده و با استفاده از الگوریتم حل شده‌اند. با مقایسه جواب‌های حاصل از الگوریتم و جواب بهینه مسایل، خطای نسبی محاسبه می‌شود. توضیحات بیشتر در ذیل آمده است.

۴-۱. طراحی آزمایش

الگوریتم ژنتیک طراحی شده با زبان ویژوال بیسیک برنامه نویسی شده و برای انجام محاسبات از رایانه (پنتیوم III) با ۳۸۴ مگابایت حافظه جانبی و پردازشگر ۸۵۰ مگاهرتز استفاده شده است. برای بررسی کارایی الگوریتم طراحی شده و انجام مقایسات، ۳۶ مسأله بصورت تصادفی تولید شده‌اند. این مسایل به گونه‌ای هستند که هزینه استقرار تسهیلات در نقاط مختلف با هم برابر بوده و تسهیلات موجود شانس بیشتری برای انتخاب شدن به عنوان جواب بهینه خواهند داشت. فاصله استقرار تسهیلات دارای محدودیت می‌باشند به طوری که فاصله بین آنها از یک حد مجاز کمتر نمی‌باشد. این مسایل در ۹ گروه تقسیم‌بندی شده‌اند و در هر گروه نیز ۴ مسأله

[1] Crossover
[2] Mutation
[3] Inversion

www.SID.ir

Archive of SID

وجود دارند که هر کدام از این ۴ مسأله با عنوان‌های مسأله اول، دوم، سوم و چهارم در ستون اول جدول‌های ۱ تا ۹ نامگذاری شده‌اند. مسایل در این ۹ گروه از لحاظ تعداد نقاط کاندیدا برای استقرار تسهیلات (n) و همچنین تعداد نقاط تقاضا (m) با یکدیگر متفاوت هستند و در هر گروه نیز، مسایل از لحاظ ماتریس ضرایب a_{ij} ، مجموعه نقاط موجود که در حال حاضر تسهیلات در آنها مستقر است (J_e) ، یک عدد کوچک مثبت به عنوان جریمه (ε) و حداقل فاصله مجاز بین محل استقرار تسهیلات (D) با یکدیگر تفاوت دارند.

مسایل در این ۹ گروه طوری در نظر گرفته شده‌اند که هر سه حالت را که ممکن است تعداد نقاط تقاضا و تعداد نقاط کاندیدا برای استقرار تسهیلات، نسبت به یکدیگر داشته باشند را شامل می‌شود. در گروه‌های اول، دوم و ششم تعداد نقاط تقاضا بیشتر از تعداد نقاط کاندیدا و در گروه سوم، پنجم و هشتم تعداد آنها با هم برابر و در گروه‌های چهارم، هفتم و نهم تعداد نقاط کاندیدا بیشتر از تعداد نقاط تقاضا می‌باشد.

برای انجام مقایسات، پس از تولید مسأله، جواب بهینه که از جنس هزینه می‌باشد، برای آن بدست می‌آید. نرم‌افزار Lingo 6 قابلیت حل مسایل غیرخطی را دارا می‌باشد اما در مورد مدل ارایه شده جواب‌های حاصل از نرم‌افزار Lingo 6، در بسیاری موارد جواب بهینه موضعی می‌باشد. یعنی جواب بهینه کلی را بطور مستقیم نمی‌توان از این روش بدست آورد. بنابراین برای بدست آوردن جواب بهینه، ابتدا محدودیت‌هایی که باعث غیر خطی شدن مسأله می‌شوند از مدل حذف شده و پس از آن مسأله با استفاده از نرم‌افزار Lingo 6 حل می‌شود. بعد از بدست آمدن جواب بهینه، چنانچه جواب حاصل در محدودیت‌های مدل مورد نظر صدق کرد آن جواب در واقع جواب بهینه مدل نیز خواهد بود و در غیر این‌صورت آن را نمی‌توان به عنوان جواب بهینه مدل قبول کرد. روش بدست آوردن جواب بهینه برای این مسایل باعث می‌شود بررسی الگوریتم از لحاظ زمان حل مسایل امکان‌پذیر نباشد. بنابراین الگوریتم فقط از لحاظ کیفیت جواب‌ها مورد بررسی قرار گرفته است. بعد از بدست آوردن جواب بهینه برای مسأله، الگوریتم طراحی شده (تعداد جمعیت اولیه برابر ۵۰ و از تعداد نسل برابر ۲۰ و واریانس جمعیت ۰/۰۱ به عنوان شرط توقف استفاده شده است) را ۵ بار برای مسأله اجرا و متوسط مقدار تابع هدف که همان تابع برازش می‌باشد، محاسبه شده است. برای بدست آوردن متوسط درصد خطای نسبی از فرمول زیر استفاده می‌شود:

$$\text{متوسط درصد خطای نسبی} = \frac{\sum_{i=1}^{k} \frac{(a-b_i)}{a}}{k} \qquad (۹)$$

که در آن a جواب بهینه و b_i جواب حاصل از الگوریتم در اجرای i ام می‌باشد.

۲-۴. نتایج محاسباتی

پس از اجرای الگوریتم طراحی شده برای مسایل مختلف و محاسبه خطای نسبی مشاهده می‌شود که بیشترین خطای نسبی حاصل از اجرای الگوریتم عبارت است از۷/۳ درصد و متوسط خطای نسبی برای کلیه مسایل عبارت است از ۱/۸ درصد که این رقم نشان از مناسب بودن کیفیت جواب‌های حاصل از اجرای الگوریتم می‌باشد. پس ملاحظه می‌شود الگوریتم طراحی شده، برای حل مدل ارایه شده دارای کارایی مناسبی می‌باشد.

۵. نتیجه‌گیری

یکی از مسایل مهم در برنامه‌ریزی و تصمیم‌گیری مدیران، انتخاب محل مناسب برای استقرار تسهیلات در مراکز صنعتی، تجاری و خدماتی می‌باشد. در مدل پوشش مجموعه و همچنین مدل مورد بحث، که گسترش یافته مدل پوشش مجموعه می‌باشد، مجموعه‌ای از مشتری‌ها با تقاضای مشخص برای خدمات یا کالاها وجود دارند که باید توسط تعدادی تسهیل خدمت داده شوند و این کار با هدف حداقل کردن هزینه صورت می‌گیرد. در واقع این قسمت بین هر دو مدل مشترک می‌باشد. در مدل پوشش مجموعه مکان بهینه نقاط استقرار تسهیلات مشخص می‌شود اما این مکان‌ها مستقل از مکان‌هایی هستند که در حال حاضر در آنجا تسهیل مستقر است در حالیکه در بسیاری از مواقع لازم است به تسهیلات کنونی اهمیت بیشتری داده شود. برای این منظور تابع هدف به گونه‌ای تغییر داده شده است که این نقاط شانس بیشتری برای انتخاب شدن به عنوان جواب بهینه داشته باشند. از جمله محدودیت‌هایی که در مواجهه با بعضی از مسایل واقعی برخورد با آن اجتناب ناپذیر است، محدودیت در فاصله تسهیلات از یکدیگر می‌باشد بطوریکه فاصله استقرار تسهیلات از یکدیگر نباید از یک حد مجاز کمتر باشد که این مورد به صورت محدودیت‌هایی به مدل پوشش مجموعه اضافه شده است. در حقیقت این مدل اطمینان می‌دهد که هر یک از مشتری‌ها بوسیله حداقل یک تسهیل پوشش داده می‌شوند، حداقل فاصله بین تسهیلات رعایت شده و از تسهیلات کنونی نیز استفاده بهینه می‌شود. در سال‌های اخیر الگوریتم ژنتیک کاربرد وسیعی در حل مسایل بهینه‌یابی پیدا کرده و توانسته است توجه محققین را به خود جلب نماید. با توجه به نوع مسأله و معضلاتی که در حل مسایل غیرخطی وجود دارد و همچنین ساختار الگوریتم ژنتیک، این الگوریتم روش مناسب و مفیدی برای حل مسأله فوق می‌باشد.

www.SID.ir

Archive of SID

جدول ۱: مقایسه متوسط جواب‌های حاصل از پنج بار اجرای الگوریتم طراحی شده با جواب بهینه برای مسایل گروه اول (جواب‌ها از جنس هزینه می‌باشند)

مسأله	n	m	J_n	J_e	D	ε	جواب حاصل از نرم‌افزار Lingo 6	جواب بهینه	متوسط جواب حاصل از GA	متوسط درصد خطای نسبی
اول	۳۰	۴۰	۲۷	۳	۸	۰/۰۳۰	۶/۱۸	۴/۰۶	۴/۰۷	۰/۲
دوم	۳۰	۴۰	۲۶	۴	۱۵	۰/۰۴۰	۸/۲۴	۷/۲	۷/۲۱	۰/۱
سوم	۳۰	۴۰	۲۳	۷	۱۱	۰/۰۲۰	۱۰/۱۸	۹/۰۸	۹/۰۸	۰
چهارم	۳۰	۴۰	۲۵	۵	۱۳	۰/۰۳۵	۱۱/۲۴	۱۰/۰۷	۱۰/۰۷	۰

جدول۲: مقایسه متوسط جواب‌های حاصل از پنج بار اجرای الگوریتم طراحی شده با جواب بهینه برای مسایل گروه دوم

مسأله	n	m	J_n	J_e	D	ε	جواب حاصل از نرم‌افزار Lingo 6	جواب بهینه	متوسط جواب حاصل از GA	متوسط درصد خطای نسبی
اول	۴۰	۶۰	۳۳	۷	۷	۰/۰۳۰	۱۳/۳۰	۱۲/۲۴	۱۲/۲۷	۰/۲۴
دوم	۴۰	۶۰	۳۵	۵	۹	۰/۰۲۷	۱۳/۳۰	۱۳/۳۰	۱۳/۳۲	۰/۱۵
سوم	۴۰	۶۰	۳۶	۴	۱۱	۰/۰۲۵	۱۵/۳۰	۱۳/۲۷	۱۳/۳۰	۰/۲۲
چهارم	۴۰	۶۰	۳۲	۸	۱۳	۰/۰۲۰	۱۹/۳۰	۱۷/۲۰	۱۷/۲۲	۰/۱

جدول ۳: مقایسه متوسط جواب‌های حاصل از پنج بار اجرای الگوریتم طراحی شده با جواب بهینه برای مسایل گروه سوم

مسأله	n	m	J_n	J_e	D	ε	جواب حاصل از نرم‌افزار Lingo 6	جواب بهینه	متوسط جواب حاصل از GA	متوسط درصد خطای نسبی
اول	۵۰	۵۰	۴۴	۶	۱۲	۰/۰۲۲	۱۶/۳۰	۱۳/۲۴	۱۳/۲۵	۰/۰۷
دوم	۵۰	۵۰	۴۶	۴	۱۱	۰/۰۲۰	۱۵/۲۶	۱۴/۲۴	۱۴/۲۴	۰
سوم	۵۰	۵۰	۴۵	۵	۱۳	۰/۰۱۸	۱۸/۳۰	۱۷/۲۵	۱۷/۲۷	۰/۱۱
چهارم	۵۰	۵۰	۴۲	۸	۱۳	۰/۰۲۳	۲۴/۴۶	۲۰/۳۲	۲۰/۵۳	۱

www.SID.ir

Archive of SID

جدول ۴: مقایسه متوسط جواب‌های حاصل از پنج بار اجرای الگوریتم طراحی شده با جواب بهینه برای مسایل گروه چهارم

مسأله	n	m	J_n	J_e	D	ε	جواب حاصل از نرم‌افزار Lingo 6	جواب بهینه	متوسط جواب حاصل از GA	متوسط درصد خطای نسبی
اول	۶۰	۴۰	۵۵	۵	۱۲	۰/۰۱۵	۱۸/۲۲	۱۵/۲۱	۱۵/۸۷	۴/۳
دوم	۶۰	۴۰	۵۰	۱۰	۱۷	۰/۰۲۰	۱۹/۳۶	۱۸/۲۴	۱۸/۲۶	۰/۱
سوم	۶۰	۴۰	۵۲	۸	۱۶	۰/۰۱۹	۲۰/۳۰	۱۷/۲۱	۱۷/۴۲	۱/۲
چهارم	۶۰	۴۰	۵۳	۷	۱۱	۰/۰۱۷	۲۱/۳۲	۱۸/۲۲	۱۸/۲۳	۰/۰۵

جدول ۵: مقایسه متوسط جواب‌های حاصل از پنج بار اجرای الگوریتم طراحی شده با جواب بهینه برای مسایل گروه پنجم

مسأله	n	m	J_n	J_e	D	ε	جواب حاصل از نرم‌افزار Lingo 6	جواب بهینه	متوسط جواب حاصل از GA	متوسط درصد خطای نسبی
اول	۷۰	۷۰	۶۳	۷	۱۳	۰/۰۱۴	۲۵/۳۰	۲۳/۲۷	۲۳/۳۰	۰/۱
دوم	۷۰	۷۰	۶۲	۸	۱۰	۰/۰۱۶	۳۲/۴۸	۲۹/۳۴	۲۹/۳۹	۰/۱۷
سوم	۷۰	۷۰	۶۰	۱۰	۱۷	۰/۰۱۵	۳۹/۴۹	۳۶/۴۳	۳۶/۶۱	۰/۵
چهارم	۷۰	۷۰	۵۵	۱۵	۱۶	۰/۰۱۷	۴۱/۵۹	۳۹/۴۶	۴۰/۱۰	۱/۶

جدول ۶: مقایسه متوسط جواب‌های حاصل از پنج بار اجرای الگوریتم طراحی شده با جواب بهینه برای مسایل گروه ششم

مسأله	n	m	J_n	J_e	D	ε	جواب حاصل از نرم‌افزار Lingo 6	جواب بهینه	متوسط جواب حاصل از GA	متوسط درصد خطای نسبی
اول	۸۰	۱۰۰	۶۸	۱۲	۱۴	۰/۰۱۴	۱۸/۲۲	۱۷/۱۵	۱۸/۰۱	۵
دوم	۸۰	۱۰۰	۶۵	۱۵	۱۶	۰/۰۱۵	۳۳/۷۵	۳۰/۳۱	۳۲/۵۴	۷/۳
سوم	۸۰	۱۰۰	۷۰	۱۰	۱۳	۰/۰۱۳	۳۶/۴۰	۳۲/۳۵	۳۳/۷۰	۴/۱
چهارم	۸۰	۱۰۰	۷۲	۸	۱۵	۰/۰۱۲	۳۵/۴۰	۳۳/۳۲	۳۴/۸۸	۴/۷

www.SID.ir

Archive of SID

جدول ۷: مقایسه متوسط جواب‌های حاصل از پنج بار اجرای الگوریتم طراحی شده با جواب بهینه برای مسایل گروه هفتم

مسأله	n	m	J_n	J_e	D	ε	جواب حاصل از نرم‌افزار Lingo 6	جواب بهینه	متوسط جواب حاصل از GA	متوسط درصد خطای نسبی
اول	۱۰۰	۸۰	۸۸	۱۲	۱۱	۰/۰۱۱	۴۵/۴۳	۴۱/۳۶	۴۱/۸۸	۱/۲
دوم	۱۰۰	۸۰	۹۰	۱۰	۱۶	۰/۰۱۰	۳۸/۳۴	۳۴/۲۸	۳۵/۸۱	۴/۴
سوم	۱۰۰	۸۰	۸۲	۱۸	۱۵	۰/۰۱۲	۵۰/۴۸	۴۶/۴۱	۴۷/۷۹	۲/۹
چهارم	۱۰۰	۸۰	۸۵	**۱۵**	۱۴	۰/۰۰۹	۴۸/۳۶	۴۸/۳۲	۴۸/۳۶	۰/۰۸

جدول ۸: مقایسه متوسط جواب‌های حاصل از پنج بار اجرای الگوریتم طراحی شده با جواب بهینه برای مسایل گروه هشتم

مسأله	n	m	J_n	J_e	D	ε	جواب حاصل از نرم‌افزار Lingo 6	جواب بهینه	متوسط جواب حاصل از GA	متوسط درصد خطای نسبی
اول	۱۲۰	۱۲۰	۹۰	۳۰	۱۶	۰/۰۱۱	۷۳/۶۳	۷۰/۵۷	۷۱/۶۰	۱/۴
دوم	۱۲۰	۱۲۰	۱۰۵	۱۵	۱۵	۰/۰۰۹	۶۶/۵۲	۵۹/۴۴	۶۱/۴۷	۳/۴
سوم	۱۲۰	۱۲۰	۱۰۰	۲۰	۱۳	۰/۰۱۰	۶۸/۵۵	۶۵/۴۹	۶۶/۵۱	۱/۵
چهارم	۱۲۰	۱۲۰	۹۲	۲۸	۱۷	۰/۰۰۷	۷۵/۳۷	۷۳/۳۵	۷۳/۳۹	۰/۰۵

جدول ۹: مقایسه متوسط جواب‌های حاصل از پنج بار اجرای الگوریتم طراحی شده با جواب بهینه برای مسایل گروه نهم

مسأله	n	m	J_n	J_e	D	ε	جواب حاصل از نرم‌افزار Lingo 6	جواب بهینه	متوسط جواب حاصل از GA	متوسط درصد خطای نسبی
اول	۱۲۰	۱۲۰	۹۰	۳۰	۱۶	۰/۰۱۱	۷۳/۶۳	۷۰/۵۷	۷۱/۶۰	۱/۴
دوم	۱۲۰	۱۲۰	۱۰۵	۱۵	۱۵	۰/۰۰۹	۶۶/۵۲	۵۹/۴۴	۶۱/۴۷	۳/۴
سوم	۱۲۰	۱۲۰	۱۰۰	۲۰	۱۳	۰/۰۱۰	۶۸/۵۵	۶۵/۴۹	۶۶/۵۱	۱/۵
چهارم	۱۲۰	۱۲۰	۹۲	۲۸	۱۷	۰/۰۰۷	۷۵/۳۷	۷۳/۳۵	۷۳/۳۹	۰/۰۵

www.SID.ir

Archive of SID

کارآمدی الگوریتم به خوبی در اجراهای متعدد آن بر روی مسایل مختلف مشخص است. نگاهی به نتایج تست مسایل مختلف نشان می‌دهد الگوریتم در مواجهه با مسایلی که تا ۱۵۰ متغیر تصمیم دارند، عملکرد موفقی دارد. بطوریکه متوسط خطای الگوریتم برای مسایل حل شده ۱/۸ درصد است یعنی با اطمینان نسبتاً بالایی می‌توان از این الگوریتم استفاده نمود. این الگوریتم نیز مانند روش‌های ابتکاری دیگر توانایی اثبات اینکه جواب‌های حاصله بهینه هستند را ندارد ولی با افزودن تعداد دفعاتی که الگوریتم بر روی یک مسأله خاص اجرا می‌شود می‌توان تا حدود زیادی اطمینان داشت که جواب بهینه یا جوابی نزدیک به آن بدست می‌آید.

هرچند امکان اثبات کارایی الگوریتم از لحاظ زمان حل مسأله، وجود نداشت و الگوریتم از لحاظ زمان رسیدن به جواب مورد نظر بررسی نشد اما الگوریتم طراحی شده اهمیت خود را از دست نمی‌دهد. زیرا همانطور که قبلاً نیز اشاره شد مدل ارایه شده جزء مسایل برنامه‌ریزی غیر خطی صفر و یک می‌باشد که دارای محدودیت‌های بسیار زیادی نیز می‌باشد که حل آن را توسط روش‌های سنتی بسیار مشکل و گاهی اوقات غیر ممکن می‌سازد. بنابراین می‌توان از این الگوریتم برای حل اینگونه مسایل با در نظر گرفتن مقدار خطای قابل پیش‌بینی (متوسط ۱/۸ درصد)، استفاده کرد.

مراجع

[1] Francis, R.L., “Facility layout and location: an analytical approach”, Englewood Cliffs, N.J., Prentice Hall, 1992.

[2] Beasley, J.E. and Jornsten. K., “Enhancing an algorithm for set covering problem”, European Journal of Operation Research, 1992, Vol. 58, pp. 293-300.

[3] Lorena, L.A.N. and Lopes, F.B., “A surrogate heuristic for set covering problems”, European Journal of Operational Research, 1994, Vol. 79, pp. 138-150.

[4] Solar, M., Parada, V. and Urrutia, R., “A parallel genetic algorithm to solve the set-covering problem”, Computers and Operations Research, 2002, Vol. 29, pp. 1221-1235.

[5] Fisher, M.L. and Kedia, P., “Optimal solution of set covering/partitioning problems using dual heuristics”, Management Science, 1990, Vol. 36, pp. 674-688.

[6] Aliminana, M. and Pastor, J.T., “An adaptation of heuristic to the location set covering problem”, European Journal of Operational Research, 1997, Vol. 100, pp. 586-593.

[7] Haddadi, S., “Simple lagrangian heuristic for the set covering problem”, European Journal of Operational Research, 1997, Vol. 97, pp. 200-204.

[8] Jacobs, L.W. and Brusco, M.J., “A local-search heuristic for large set-covering problems”, Naval Research Logistics, 1995, Vol. 42, pp. 1129-1140.

[9] Al-Sultan, K.S., Hussain, M.F. and Nizami, J.S., “A genetic algorithm for the set covering problem”, Journal of Operational Research Society, 1996, Vol. 47, pp. 702-709.

[9] Beasley, J.E., “A genetic algorithm for the set covering problem”, European Journal of Operational Research, 1996, Vol. 94, pp. 392-404.

[10] Lorena, L. and Lopes, S., “Genetic algorithm applied to computationally difficult set covering problem”, Journal of Operational Research Society, 1997, Vol. 48, pp. 440-445.

[11] Monfroglio, A., “Hybrid heuristic algorithm for set covering”, Computers and Operations Research, 1998, Vol. 25, pp. 441-455.

[12] Hifi, M., Paschos, V.Th. and Zissimopoulos, V., “A neural network for the min set covering problem”, Chaos, Solitons and Fractals, 2000, Vol. 11, pp. 2079-2089.

[13] Ohlsson, M., Peterson, C. and Soderberg, B., “An efficient mean approach to the set covering problem”, European Journal of Operational Research, 2001, Vol. 133, pp. 583-595.

[14] Francis. J.V. and Floyd, E.W., “A heuristic concentration approach for weighted set covering problems”, Locator: Publication of Location Analysis, 2001, Vol. 2, pp. 1-14.

[15] Gen, M. and Cheng, R., “Genetic algorithms and engineering design”, New York, A Wiley-Interscience Publication, 1997.

[16] Holland, J.H., “Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence”, MIT Press, Cambridge, (2nd edition in 1992) 1975.

[17] Goldberg, D.E., “Genetic algorithms in search, optimization, and machine learning”, Addison-Wesley Publishing Co, 1989.

[18] Jaramillo, J.H., Bhadury, J. and Batta, R., “On the use of Genetic algorithm to solve location problems”, Computers and Operations Research, 2002, Vol. 29, pp. 761-779.

www.SID.ir