دانشکده ریاضی

گروه ریاضی کاربردی

مکان‌یابی مسیر مرکزی روی شبکه

نگارش

مریم رهبری

استاد راهنما

دکتر جعفر فتحعلی

استاد مشاور

دکتر نادر جعفری راد

پایان‌نامه ارشد جهت اخذ درجه کارشناسی ارشد

شهریور ۱۳۸۷

تقدیم به **عظمت آیة الکرسی**

این برگ سبز و این تحفه درویش را به آیین مهریان مهربان و به نشانه سپاس از همه مهرآوریها، پیچیده در حریر نور به دو باغبان مهربان و به دو سرو سایه افکن و سرفراز تقدیم می کنم :

پدرم به پاس همه رنجهایی که برای آبیاری ریشه های بوته امید خویش کشید

مادرم که از پرتو یاس گونه اش زندگی را چشیدم و نگاه خسته و لبهای دعاگویش بدرقه راهم گشت

آنانکه وجودم برایشان همه رنج بود و وجودشان برایم همه مهر

توانشان رفت تا به توانایی برسم

آنانکه فروغ نگاهشان گرمی کلامشان و روشنی رویشان سرمایه های جاودانی من است

آنانکه راستی قامتم در شکستگی قامتشان تجلی یافت در برابر وجودشان زانوی ادب بر زمین می نهم و با دلی مملو از عشق و محبت و خضوع بر قلبشان بوسه می زنم سرو وجودشان همیشه سرسبز و استوار باد.

و تقدیم به **برادرانم** که وجودشان گواهیست بر مهربانی خداوند.

آغاز علم شناسی خداست و سرانجام علم واگذار کردن کارها به اوست.

حضرت محمد (ص)

سپاس خداوندی را که سخنوران از ستودن او عاجزند و حساب‌گران از شمارش نعمت‌های او ناتوان و تلاش‌گران از ادای حق او درمانده‌اند. خدایی که افکار ژرف اندیش، ذات او را درک نمی‌کنند و دست غواصان دریای علوم به او نخواهد رسید. پروردگاری که برای صفات او حد و مرزی وجود ندارد و تعریف کاملی نمی‌توان یافت و برای خدا وقتی معین و سرآمدی مشخص نمی‌توان تعیین کرد. خلقت را آغاز کرد و موجودات را بیافرید، بدون نیاز به فکر و اندیشه‌ای، یا استفاده از تجربه‌ای، بی آن که حرکتی ایجاد کند و یا تصمیمی مضطرب در او راه داشته باشد. ستایش می‌کنم خداوند را، برای تکمیل نعمت‌های او و تسلیم بودن در برابر بزرگی او، و ایمن ماندن از نافرمانی او، در رفع نیازها از او یاری می‌طلبم؛ زیرا آن کس را که خدا هدایت کند، هرگز گمراه نگردد، و آن را که خدا دشمن دارد، هرگز نجات نیابد و هر آن کس را خداوند بی‌نیاز گرداند، نیازمند نخواهد شد. **پس ستایش خداوند، گران سنگ‌ترین چیز است، و برترین گنجی است که ارزش ذخیره شدن دارد.**

اکنون که با عنایات او نگارش این پایان‌نامه به اتمام رسیده است، فرصت را برای قدردانی از عزیزانی که در طی مراحل انجام این تحقیق همواره از یاری و همدلی ایشان بهره‌مند گشته‌ام، مغتنم می شمارم.

استاد ارجمندم جناب آقای **دکتر جعفر فتحعلی** که در طول انجام این پایان‌نامه از راهنمایی‌های ارزشمند ایشان بهره‌مند گشتم. بدون شک انجام این پایان‌نامه بدون رهنمودهای ایشان غیر ممکن می نمود. تلاش‌ها و زحمات بی‌شائبه ایشان جای تقدیر بسیار دارد. جناب آقای **دکتر نادر جعفری راد**، از زحمات ایشان نیز کمال تشکر و قدردانی را دارم. همچنین از آقایان **دکتر امید سلیمانی فرد و دکتر احمد نزاکتی** که قبول زحمت نموده و داوری پایان‌نامه را به عهده گرفتند تشکر می‌نمایم. و در نهایت آنچه نه انکارناپذیر است و نه فراموش‌شدنی محبت، بزرگواری و تلاش گوهران بی‌مانند زندگیم پدر و مادر و به خصوص برادر بزرگترم، میثم است که زمینه رشد وشکوفایی اندیشه‌ام را فراهم نمودند. ضمنا لازم میدانم از کلیه عزیزانی که اینجانب را در تهیه مقالات یاری نموده‌اند قدردانی نمایم. امید آن که مطالب این پایان‌نامه مورد استفاده علاقه‌مندان به تحقیق در زمینه مذکور قرار گیرد، انشاا..

به پاس نثار صداقت و صمیمیت.

مریم رهبری شهریورماه یکهزار و سیصد و هشتاد و هفت

اینجانب مریم رهبری تایید می نماییم که مطالب مندرج دراین پـایان نامه نتیجه تحقیقات خودم می باشد و در صورت استفاده از نتایج دیگران مرجع آن را ذکر نموده ام.

شهریور ۱۳۸۷

چکیده

مسائل مکان‌یابی از جمله مسائل تحقیق در عملیات هستند که در دهه‌های اخیر توجه بسیاری را به خود جلب کرده، و نیز کاربردهای فراوانی در دنیای واقعی دارند. یکی از مسائل اساسی در نظریه مکان‌یابی، مساله مکان‌یابی مسیر می باشد. در بعضی از مسائل مکان‌یابی به حالاتی برخورد می‌کنیم که اگر خواسته باشیم آنها را با مدل‌های مکان‌یابی نقطه‌ای، مکان‌یابی کنیم، با بی‌شمار نقطه مواجه خواهیم شد. در این گونه مسائل مدل‌های مکان‌یابی نقطه‌ای جوابگو نیستند. در نتیجه مدل‌های مکان‌یابی نقطه‌ای گسترش پیدا کردند و منجر به مکان‌یابی نوع خاصی از زیر گراف‌ها مثل درخت، مسیر و دور شده‌اند. مساله مکان‌یابی مسیر بر روی گراف

به مکان‌یابی سه نوع مسیر منجر می‌شود که عبارتند از مکان‌یابی مسیر مرکزی، مسیر میانه و مکان‌یابی مسیر بهینه پارتو. در این پایان‌نامه به مکان‌یابی مسیر مرکزی روی شبکه می‌پردازیم. مفهوم مسیر مرکزی اولین بار در سال ۱۹۷۷م توسط اسلیتر و هدتنیمی به طور جداگانه مطرح شد. در مساله مکان‌یابی مسیر مرکزی هدف پیدا کردن مسیری است به گونه‌ای که فاصله دورترین مشتری تا این مسیر کمترین مقدار شود. از کاربردهای این مساله می‌توان به مکان‌یابی لوله‌های آب، نفت و گاز و مکان‌یابی خطوط ریلی برون‌شهری و درون‌شهری اشاره کرد. در سال۱۹۸۱م هدتنیمی و در سال ۱۹۸۲م اسلیتر الگوریتم خطی برای مکان‌یابی مسیر مرکزی روی درخت ارائه کردند. در این پایان‌نامه مساله مسیر مرکزی بدون محدودیت، مساله مسیر مرکزی با محدودیت طول و الگوریتمی که اسلیتر برای مساله مسیر مرکزی روی درخت ارائه کرده است را بررسی خواهیم کرد. و چون مساله مسیر مرکزی روی شبکه‌ها مساله‌ای NP- سخت است، روش‌های ابتکاری برای حل این مساله ارائه می‌شود. ما در این پایان‌نامه الگوریتم ژنتیکی برای حل این مساله ارائه کرده و جهت بهبود نتایج، الگوریتم ترکیبی دیگری مرکب از الگوریتم ژنتیک و روش مورچه ارائه کرده‌ایم. و سپس نتایج این دو الگوریتم را با هم مقایسه خواهیم کرد. کلمات کلیدی: مکان‌یابی- مساله مسیر مرکزی- الگوریتم ژنتیک- روش مورچه

قسمتی از این پایان‌نامه در قالب مقاله‌ای تحت عنوان مکان‌یابی مسیر مرکزی با طول محدود روی شبکه با استفاده از الگوریتم ژنتیک در سی و نهمین کنفرانس ریاضی ایران (دانشگاه شهید باهنر کرمان) ارائه شده است. و قسمت دیگری از این پایان‌نامه در قالب مقاله‌ای تحت عنوان الگوریتم ترکیبی برای مساله مسیر مرکزی به مجله" Rairo Operation Research" که ISI می باشد ارسال شده است.

فهرست مطالب

فصل دوم

مکان‌یابی مسیر مرکزی روی درخت

فصل سوم

الگوریتم ژنتیک

فهرست اشکال

فهرست جداول

فصل اول

مسائل مکان‌یابی و مساله مسیر مرکزی

در این فصل ما ابتدا خلاصه‌ای از تاریخچه مسائل مکان‌یابی را بیان می‌کنیم، و سپس به بیان یک سری تعاریف مورد نیاز مسائل مکان‌یابی می‌پردازیم و در آخر فصل مساله مسیر مرکزی[1] را مورد بررسی قرار می‌دهیم. مطالب ارائه شده در این فصل برگرفته از مراجع [۲۵،۱] است.

مقدمه

مسائل مکان‌یابی[2] از جمله مسائلی هستند که امروزه کاربردهای فراوانی دارند و توجه بسیاری را به خود جلب کرده‌اند. ریشه یونانی مکان‌یابی کلمه topothesia می باشد. مسائل مکان‌یابی را در حالت کلی می‌توان به صورت مقابل مطرح کرد: یک مجموعه از مشتری‌ها که در یک ناحیه جغرافیایی قرار دارند و متقاضی دریافت سرویس می باشند.

براساس نوشته ریول[3] [۲] اولین کسی که بطور رسمی مساله مکان‌یابی را به کار برد، امپراتور کنستانتین در قرن چهاردهم میلادی بود. کنستانتین مساله مکان‌یابی نقاطی بر روی یک شبکه، که نشان دهنده مکان

---

[1] Path- center
[2] Location Problems
[3] ReVelle

پایگاه‌های ارتش رم بود، را حل کرد. پایگاه‌ها باید به گونه ای قرار می‌گرفتند که بتوانند در مقابل تهدیدات مهاجمان و شورشیان محلی از امپراتوری دفاع کنند.

پیدایش مساله مکان‌یابی را می توان به قرن هفدهم میلادی نسبت داد، زمانی که فرما[۴] مساله زیر را مطرح کرد: فرض کنید سه نقطه در صفحه داده شده است، نقطه چهارم را به گونه‌ای بیابید که مجموع فاصله‌های آن تا سه نقطه داده شده کمینه شود. توریچلی[۵] در سال ۱۶۴۰م این مساله را حل کرده است بدین دلیل نقطه بهینه را نقطه توریچلی و مساله را مساله فرما می نامند [۳]. مساله فرما، مساله اشتاینر***6*** نیز نامیده می‌شود، زیرا این مساله بر روی شبکه توسط اشتاینر در قرن نوزدهم میلادی مطرح شد[۴].

اولین تعریف مساله مکان‌یابی به صورت کاربردی در ۱۹۰۹م توسط وبر[۷] ارائه شد [۵]. اما مطالعات جدی بر روی این مساله از زمانی شروع شد که در ۱۹۶۴م حکیمی***8*** تابع هدف این مسائل را به صورت کمترین مجموع***9*** و مینیماکس***10*** مطرح کرد [۶]. او همچنین بررسی مسائل مکان‌یابی روی شبکه‌ها را انجام داد. از آن زمان به بعد مطالعات زیادی بر روی مسائل مکان‌یابی انجام شد. اولین طبقه‌بندی مدل‌های مختلف مکان‌یابی توسط هندلر***11*** و میرچندانی***12*** ارائه شد [۷]. پس از آن طبقه‌بندی‌های دیگری توسط هاماچر***13*** و نیکل***14*** نیز ارائه شد [۸]. همچنین افرادی مثل تانسل***15*** و همکاران [۹]، کراروپ***16*** و پروزن***17***[۱۰]، هانسن***18*** و همکاران[۱۱]، اون***19*** و دسکین***20*** [۱۲]، کاپنرا***21*** [۱۳]، اسکاپارا***22*** و اسکاتلا***23***[۱۴] لیستی

---

4 Fermat
5 Torricelli
6 Steiner problem
7 Weber
8 Hakimi
9 Minisum
10 Minimax
11 Handler
12 Mirchandani
13 Hamacher
14 Nickel
15 Tansel
16 Krarup
17 Pruzan
18 Hansen
19 Owen
20 Daskin
21 Cappanera
22 Scaparra
23 Scutella

از کارهای انجام شده در زمینه‌های مختلف مکان‌یابی را ارائه کرده‌اند. کارنت**24** و همکاران نیز کاربردهای مسائل مکان‌یابی را [۱۵] بیان نموده‌اند. در ادامه ما مساله مکان‌یابی را تعریف کرده وسپس به طور خاص به مساله مسیر مرکزی می‌پردازیم

تعاریف

یک گراف ساده زوجی به صورت $G=(V, E)$ است که در آن V مجموعه‌ایn عضوی موسوم به مجموعه رئوس گراف و E مجموعه‌ای‌از زیرمجموعه‌های دو عضوی V موسوم به مجموعه یال‌های گراف G است. یک گراف وزن‌دار گرافی است که با تخصیص اعدادی موسوم به وزن روی راس‌ها یا یال‌های یک گراف ساده حاصل می‌شود. فرض کنید $V=\{v_1,\ldots,v_n\}$ در این صورت وزن هر راس v_i را با w_i نشان می‌دهیم. همچنین اگر یال بین دو راس v_i و v_j را با e_{ij} نشان دهیم، آن گاه وزن یال e_{ij} را با w'_{ij} نشان می‌دهیم.

تعریف ۱-۲-۱. فاصله بین دو راس u,v در گراف G که با $d(u,v)$ نشان داده می‌شود برابر با مجموع وزنی یال‌های موجود در کوتاهترین مسیر بین u,v است. همچنین فاصله بین راس u تا زیرمجموعه S از راس‌های گراف G برابر است با کمترین فاصله‌ای که راس u نسبت به راس‌های S می‌تواند داشته باشد و با $d(u,S)$ نشان داده می‌شود. لذا

$$d(u,S)=\min\{d(u,w):w\in S\}. \qquad (۱-۱)$$

تعریف ۱-۲-۲. خروج از مرکز راس u[۲۵] در گراف G برابر است با بیشترین فاصله‌ای که راس u نسبت به بقیه رئوس می‌تواند داشته باشد و با نماد $e(u,G)$ نشان داده می‌شود. یعنی

$$e(u;G)=\max\{d(u,v):v\in V(G)\}. \qquad (۲-۱)$$

تعریف ۱-۲-۳. اگر S_n مجموعه‌ای شامل n راس از رئوس گراف G باشد، خروج از مرکز S_n که با $e(S_n,G)$ نشان داده می‌شود برابر است با بیشترین فاصله‌ای‌که رئوس گراف G می‌توانند نسبت به نزدیکترین راس مجموعه S_n داشته باشند. و به عبارت دیگر خواهیم داشت

---

24 Current
25 Eccentricity

$$e(s_n ;G)= \max\{d(v,s_n):v\in V(G)\}. \qquad (۳-۱)$$

تعریف ۱-۲-۴. یک راس مرکزی[۲۶] در گراف G راسی است که کمترین خروج از مرکز را در گراف G داشته باشد. همچنین مرکز گراف G شامل همه راس‌های مرکزی گراف G خواهد بود.

$$d'(s_n;G)= \sum_{v\in V(G)} d(v,s_n). \qquad (۴-۱)$$

تعریف ۱-۲-۵. مجموع فاصله‌ای‌که هر کدام از رئوس گراف G نسبت به رئوس S_n دارند را فاصله S_n تا گراف G می‌نامند. و به عبارت دیگر خواهیم داشت

در ادامه برای راحتی کار 'd را همان d در نظر می‌گیریم. یعنی داریم

$$d(s_n;G)= \sum_{v\in V(G)} d(v,s_n). \qquad (۵-۱)$$

تعریف ۱-۲-۶. راسی که کمترین مجموع فاصله را تا رئوس گراف G داشته باشد، راس میانه[۲۷] گوییم. میانه[۲۸] گراف G شامل همه راس‌های میانه گراف G خواهد بود.

تعریف ۱-۲-۷. با توجه به تعریف ۱-۲-۳ و ۱-۲-۵ خروج از مرکز مسیر P و فاصله مسیر P تا گراف G به صورت زیر تعریف می‌شود.

$$e(P;G)= \max\{d(v,P):v\in V(G)\}. \qquad (۶-۱)$$

$$d(P;G)= \sum_{v\in V(G)} d(v,P) \qquad (۷-۱)$$

و همچنین با توجه به تعریف ۱-۲-۱ فاصله هر راس تا مسیر P را به صورت زیر تعریف می‌شود

$$d\ (v,P)= \min\{d(v,w):w\in P\}. \qquad (۸-۱)$$

تعریف ۱-۲-۸. اگر برای هر زیرمجموعه n راسی از گراف G مثل S'_n داشته باشیم

$$e(s_n;G) \le e(s'_n;G)، \qquad (۹-۱)$$

---

[26] Center vertex
[27] Median vertex
[28] Median

آنگاه با توجه به تعریف ۱-۲-۴ مجموعه S_n را n- مرکز می‌نامند. و بطور مشابه اگر برای هر زیرمجموعه n راسی ازگراف G مثل S'_n داشته باشیم

$$d(s_n;G)\leq d(s'_n;G). \qquad (۱۰-۱)$$

آنگاه با توجه به تعریف ۱-۲-۶ مجموعه S_n را n - میانه می‌نامند.

تعریف ۱-۲-۹. به مجموع وزن یالهای مسیر P، طول مسیر P گویند. و با L(P) نشان داده می‌شود.

مسائل مکان‌یابی

۱-۳-۱ مدل های مکان‌یابی نقطه ای[۲۹]

۱-۳-۱-۱ مساله تک‌وسیله‌ای با کمترین مجموع[۳۰]

ساده‌ترین مساله مکان‌یابی همان مساله‌ای است که فرما مطرح کرد و ما آن را در بخش ۱-۱ بیان کردیم. تعمیمی از آن، که به مساله فرما - وبر معروف می‌باشد، به این صورت است که فرض کنید n نقطه $P_1,P_2,\ldots,P_n$ درصفحه موجودند و به ترتیب دارای وزن‌های $w_1,w_2,..,w_n$ هستند، می‌خواهیم نقطه‌ای مانند x را به گونه‌ای بیابیم که مجموع وزنی فاصله x تا نقاط موجود کمینه شود. یعنی اگر فاصله x تا P را به صورت d(x,P) نمایش دهیم آنگاه تابع هدف مساله به صورت زیر خواهد بود

$$\min \sum_{i=1}^{n} w_i\, d(x,P_i). \qquad (۱۱-۱)$$

مساله فوق به مساله تک‌وسیله‌ای با کمترین مجموع معروف است.

۱-۳-۱-۲ مساله تک‌وسیله‌ای مینیماکس[۳۱]

همچنین اگر هدف پیدا کردن x به گونه ای باشد که فاصله وزنی x تا دورترین نقطه موجود کمینه شود، آنگاه مساله را مساله تک‌وسیله‌ای مینیماکس گویند که تابع هدف این گونه مسائل به صورت زیر می‌باشد

$$\min \quad \max\, w_i\, d(x,P_i)\ ,\ i=1,2\ldots n\,. \qquad (۱۲-۱)$$

---

29 Point location models
30 Single Facility Minisum Problem
31 Single Facility Minimax Problem

۱-۳-۱-۳ مساله چندوسیله‌ای با کمترین مجموع [۳۲]

اگر به جای یک نقطه به دنبال چند نقطه باشیم یعنی بخواهیم چند مکان برای وسایل جدید به گونه‌ای بیابیم که مجموع وزنی فاصله‌ها کمینه شود، مساله را مساله چند وسیله‌ای با کمترین مجموع گوییم و در این حالت اگر $X=\{x_1,x_2,...,x_k\}$ مجموعه وسایل جدید باشد که باید مکان‌یابی شوند آنگاه می‌توان تابع هدف مسائل جدید را به صورت زیر نوشت. که در آن w_{ji} هزینه بین وسیله j ام و مکان (نقطه) i ام. و V_{jl} برابر است با هزینه بین وسیله j ام و وسیله l ام

$$\min_X F(X)=\sum_{j=1}^{k}\sum_{i=1}^{n} w_{ji}\, d(x_j,P_i)+\sum_{l,j=1}^{k} v_{jl}\, d(x_j,x_l). \qquad (۱۳-۱)$$

۱-۳-۱-۴ مساله چندوسیله‌ای مینیماکس [۳۳]

اگر خواسته باشیم فاصله وزنی وسایل جدید تا دورترین نقطه موجود را کمینه کنیم، آنگاه مساله را مساله چندوسیله‌ای مینیماکس گوییم. که طبق تعاریف بالا تابع هدف این گونه مسائل به صورت زیر خواهد بود

$$\min_X G(X)=\max_{i,j,l}\,(w_{ij}\, d(x_i,P_j),v_{lj}d(x_l,x_j))\text{؛} \qquad (۱۴-۱)$$

$$i=1,2,..,n \text{ ؛ } J,L=1,2,..,k.$$

۱-۳-۱-۵ مساله مکان‌یابی پیوسته

در بحث فوق اگر مکان وسایل جدید بتواند در هر نقطه‌ای از صفحه قرار گیرد آنگاه مساله را مساله مکان‌یابی پیوسته گویند. در این گونه مسائل اگر تابع فاصله یعنی $d(x,P)$ به صورت یکی از نرم‌های خطی، اقلیدسی و چبیشف باشد، راه‌حل‌هایی ارائه شده است [۱۶].

۱-۳-۱-۶ مساله مکان‌یابی گسسته

---

[32] Multi Facility Minisum Problem
[33] Multi Facility Minimax Problem

در حالت گسسته مکان وسایل جدید تنها می‌تواند در نقاط موجود قرار گیرند. و یک حالت خاص آن زمانی است که نقاط موجود روی شبکه قرار دارند و مکان وسایل جدید نیز باید نقطه‌ای از شبکه باشند. در این حالت تابع هدف به صورت کمترین مجموع و یا مینیماکس است.

۱-۳-۱-۶-۱ مساله مرکز[۳۴]

در مسائل مکان‌یابی گسسته مینیماکس، هدف یافتن راس u است، به قسمی که خروج از مرکز u مینیمم شود. به عبارت دیگر در مسائل مکان‌یابی مینیماکس هدف کمینه کردن بیشترین فاصله است. که این گونه مسائل را مساله مرکز گویند.

۱-۳-۱-۶-۲ مساله میانه [۳۵]

در مسائل مکان‌یابی گسسته با کمترین مجموع، هدف یافتن میانه گراف G است به عبارت دیگر هدف یافتن راس u است، به قسمی که مجموع فاصله همه راس‌ها تا u کمینه شود. که به این گونه مسائل مساله میانه گویند. الگوریتم‌های برای پیدا کردن مرکز و میانه درخت توسط هندلر [۱۷]، گلدمن[۳۶][۱۸] ارائه شده است.

۱-۳-۱-۷ مساله مکان‌یابی چند مرکز و چند میانه

در کارهای بعدی حکیمی[۱۹] و شیر[۳۷][۲۰] مفهوم مکان‌یابی یک وسیله را به پیدا کردن p وسیله جدید گسترش دادند. یعنی هدف پیدا کردن P- میانه و P- مرکز در گراف G است.

در دنیای واقعی برای مشخص کردن مکان بانک‌ها، انبارها، فروشگاه‌ها، مراکز اورژانس، مراکز امدادرسانی و ایستگاه‌های مترو و قطار می توانیم از مدل‌های مکان‌یابی که در بالا اشاره شد استفاده کنیم. برای مطالعه بیشتر می‌توانید به [۱۷،۱۸،۲۱،۲۲،۲۳] مراجعه کنید.

۱-۳-۲ مسائل مکان‌یابی مسیر

---

34 Center
35 Median
36 Goldman
37 Shier

در بعضی از مسائل مکان‌یابی به تجهیزاتی برخورد می‌کنیم، که اگر خواسته باشیم آنها را با مدل‌های مکان‌یابی نقطه ای، مکان‌یابی کنیم، با بی‌شمار نقطه مواجه خواهیم شد. در این گونه مسائل دیگر مدل‌های مکان‌یابی نقطه‌ای جوابگو نیستند. در نتیجه مدل‌های مکان‌یابی نقطه‌ای گسترش یافتند و منجر به مکان‌یابی نوع خاصی از زیر گراف‌ها مثل درخت، مسیر و دور شدند[۲۴،۲۵]. در این پایان‌نامه ما مکان‌یابی مسیر را بررسی خواهیم کرد. مساله مکان‌یابی مسیر بر روی درخت اولین بار توسط اسلیتر [۲۵،۲۶] مطرح شده است. از جمله مسائلی که در مکان‌یابی مسیر مورد توجه است، مکان‌یابی مسیر P با طول مشخص L می‌باشد که $L\geq 0$ فرض می‌شود.

۱-۳-۲-۱ انواع مسیر

الف- مسیر مرکزی. مسیر P در گراف G را یک مسیر مرکزی گوییم، اگر برای هر مسیر مثل P' در گراف G داشته باشیم

$$e(P;G) \leq e(P';G)\ , \qquad (۱۵-۱)$$

و اگر برای دو مسیر P و P' داشته باشیم

$$e(P;G) = e(P';G). \qquad (۱۶-۱)$$

آنگاه هر مسیر که طول کمتری داشته باشد، مسیر مرکزی خواهد بود.

$P'\subset P$ آنگاه داریم $e(P) < e(P')$. و اگر

ب- مسیر میانه [۳۸]. مسیر P در گراف G را یک مسیر میانه یا هسته[۳۹] گراف گوییم، اگر برای هر مسیر مثل P' در گراف G داشته باشیم

$$d(P;G) \leq d\left(P';G\right). \qquad (۱۷-۱)$$

اگر $P'\subset P$ آنگاه $d(P) < d(P')$.

[38] Path median
[39] Core

ج- مسیر بهینه پارتو[۴۰]. مسیر P در گراف G را یک مسیر بهینه پارتو گوییم، اگرمسیر P هم کمترین خروج از مرکز و هم کمترین فاصله را در گراف G داشته باشد. به عبارت دیگر اگر برای هر مسیر مثل 'P در گراف G داشته باشیم

$$d(P;G) \le d(P';G) \quad \text{and} \quad e(P;G) \le e(P';G). \qquad (۱۸-۱)$$

(و باید حداقل یکی از دو نامساوی بالا به صورت اکید برقرار باشد) آنگاه مسیر P را مسیر بهینه پارتو گوییم.

۱-۳-۲-۲ مسائل مکان‌یابی مسیر با کمترین مجموع

در مسائل مکان‌یابی مسیر با کمترین مجموع هدف پیدا کردن هسته گراف یا مسیر میانه گراف است. و یا به عبارت دیگر هدف مکان‌یابی مسیرP است، به گونه‌ای که مجموع فاصله همه راس‌های گراف تا این مسیر کمینه شود. تابع هدف این مساله به صورت زیر خواهد بود

$$\min \sum_{i=1}^{n} w_i \, d(v_i,P). \qquad (۱۹-۱)$$

اولین بار اسلیتر***41*** [۲۵] مساله پیدا کردن هسته درخت را مطرح کرد. مورگان و اسلیتر***42*** [۲۶] و بکر***43*** [۲۷] یک الگوریتم خطی برای پیدا کردن هسته درخت ارائه کرده‌اند.

از مسائل دیگری که می‌تواند مطرح شود پیدا کردن هسته گراف با طول معین یا هزینه مشخص است که پنگ و لئو***44*** [۲۸] و بکر و همکاران [۲۹] الگوریتمی برای این مساله ارائه کرده‌اند. الستراپ[۴۵] و همکاران نتایج خوبی برای پیدا کردن هسته گراف با طول معین روی درخت در سال ۱۹۹۷م ارائه کردند [۳۰]. الگوریتم موازی برای این مساله توسط ونگ ***46*** [۳۱] ارائه شده است.

---

40 Path-medi center or Parto-optimal path
41 Slater
42 Morgan and Slater
43 Becker
44 Peng and Lo
45 Alstrup
46 Wang

در تحقیقات اخیر که تمیر[۴۷] و همکاران [۳۲] و ونگ و همکاران [۳۳] ارائه داده اند شرایط جدیدی را به مساله بالا اضافه کردند. در این گونه مسائل هدف پیدا کردن مسیر میانه با طول محدود بر روی درخت است، به گونه‌ای که این مسیر از نقاط خاصی که از قبل مشخص شده است عبور کند.

همچنین ونگ و لین و همکاران [۳۴] مساله پیدا کردن مسیر میانه بر روی درخت تحت شرایطی که بعضی از تجهیزات از قبل وجود دارند را بررسی کردند.

۱-۳-۲-۳ مسائل مکان‌یابی مسیر مینیماکس

در مسائل مکان‌یابی مینیماکس هدف پیدا کردن مسیرP است، به گونه‌ای که فاصله دورترین راس تا این مسیر کمینه شود. به عبارت دیگر هدف پیدا کردن مسیر مرکزی روی گراف است. تابع هدف این مساله همانند۱-۲۰ خواهد بود

$$\min \max \; w_i \, d(v_i,P) \;,\; i=1,2,\ldots n. \qquad (۲۰-۱)$$

هدتنیمی[۴۸] و همکاران [۳۵] و اسلیتر [۲۵] الگوریتم‌های خطی برای مکان‌یابی مسیر مرکزی روی درخت ارائه کردند.

۱-۳-۲-۴ مساله مسیر مرکزی با طول محدود

در مساله مسیر مرکزی با طول محدود روی شبکه هدف پیدا کردن مسیرP با طول معین L بر روی شبکه N=(V,E) است، به گونه‌ای‌که فاصله دورترین راس تا مسیر P روی شبکه کمترین مقدار را داشته باشد. یعنی کمینه کردن تابع زیر

$$L(P) \leq L.\; F(P) = \max_{v_i \notin V(P)} d\big(v_i, V(P)\big) \text{ and} \qquad (۲۱-۱)$$

مینیکا[۴۹] [۳۶] مساله مکان‌یابی مسیر مرکزی با طول معین را بررسی کرده است. او هشت مساله بهینه‌سازی زیر را مطرح کرد:

---

47 Tamir at el
48 Hedetniemi
49 Minieka

۱. مساله پیدا کردن مسیر با طول معین با کمترین خروج از مرکز؛

۲. مساله پیدا کردن مسیر با طول معین با بیشترین خروج از مرکز؛

۳. مساله پیدا کردن مسیر با طول معین با کمترین مجموع فاصله؛

۴. مساله پیدا کردن مسیر با طول معین با بیشترین مجموع فاصله؛

۵. مساله پیدا کردن درخت با طول معین با کمترین خروج از مرکز؛

۶. مساله پیدا کردن درخت با طول معین با بیشترین خروج از مرکز؛

۷. مساله پیدا کردن درخت با طول معین با کمترین مجموع فاصله؛

۸. مساله پیدا کردن درخت با طول معین با بیشترین مجموع فاصله.

مینیکا [۳۶] برای مسائل اول تا هفتم بالا الگوریتم خطی و برای مساله هشتم الگوریتمی با پیچیدگی زمانی $O(2^P)$ ارائه کرده است که P تعداد برگ‌های درخت است. حکیمی و همکاران در سال ۱۹۹۳م ثابت کردند که این مساله NP- سخت است[۳۸].

مساله سوم در حقیقت مکان‌یابی مسیر میانه است و الگوریتم‌هایی که برای این مساله ارائه شده است در بخش قبل ذکر کردیم. برای مساله چهارم پنگ و لئو [۲۸] الگوریتمی ارائه کردند.

پنگ و تسونگ لو[۵۰] [۳۷] در سال ۱۹۹۴م برای همه مسائل مکان‌یابی بالا به غیر از مساله سوم و چهارم الگوریتم‌هایی موازی را ارائه کردند. آنها برای همه راس‌های شبکه وزن غیرمنفی در نظر گرفتند و در مساله آخر یعنی پیدا کردن درخت با بیشترین مجموع فاصله وزن هر راس شبکه را از مقادیر ثابتی کمتر گرفتند. آن‌ها تحت این شرایط توانستند این مساله را با پیچیدگی زمانی چند جمله‌ای‌حل کنند که نسبت به حالت قبل بهبود پیدا کرد [۳۷]. نتایجی که آنها به دست آوردند به شرح زیر است:

جدول ۱-۱ نتایج به دست آمده توسط پنگ و تسونگ لو[۳۷]

[50] Peng and Tsung Lo

	Path		Tree	
	Time	**# of processors**	**Time**	**# of processors**
کمترین خروج از مرکز	O(log n)	O(n)	O(n)	O(log n)
بیشترین خروج از مرکز	O(log n)	O(n/ log n)	O(n/ log n)	O(log n)
کمترین مجموع فاصله	$O(\log^2 n)$	O(n/log n)	O(n)	O(log n)
بیشترین مجموع فاصله	$O(\log^2 n)$	O(n/log n)	$O(n^3)$	$O(\log^3 n)$

برای مرور اجمالی بر مسائل مکان‌یابی مسیر بر روی درخت می‌توان به [۳۸،۳۹،۴۰] مراجعه کرد.

و برای مرور اجمالی بر مسائل مکان‌یابی زیر درخت‌ها و جنگل می‌توان به [۴۱] مراجعه کرد.

۱-۳-۲-۵ مسائل مکان‌یابی مسیر با کمترین مجموع و مینیماکس

همه مسائل مکان‌یابی مسیر تحت پوشش دو دسته بالا قرار نمی‌گیرند. در بعضی از مسائل اگر از هر کدام از معیارها به تنهایی استفاده کنیم ممکن است منجر به جواب قابل قبولی نشود. به عنوان مثال استفاده از معیار میانه به تنهایی ممکن است باعث شود که جواب‌ها قابل قبول برای بعضی از مشتریان نباشند زیرا ما در این مسائل مجموع فواصل را کمینه می‌کنیم به همین دلیل ممکن است که فاصله بعضی از مشتریان تا تجهیزات دور شود. و یا استفاده از معیار مرکز به تنهایی باعث شود که هزینه کل سیستم زیاد شود. به همین دلیل اگر در بعضی از مسائل از ترکیب این دو معیار استفاده کنیم به جواب بهتری می‌رسیم. در نتیجه دسته سوم شامل ترکیب دو دسته قبلی می باشد. که هدف در این مسائل مکان‌یابی مسیرهای پارتو است.

اورباخ و برمن[۵۱] [۴۲] مساله مکان‌یابی مسیر بهینه پارتو را مطرح کردند و الگوریتمی برای مکان‌یابی این مسیر روی درخت ارائه دادند. آنها سه مساله زیر را مطرح کرده اند:

پیدا کردن مسیری که کمترین مجموع فاصله را بین بقیه مسیرها داشته باشد و خروج از مرکز آن از یک مقدار معین کمتر باشد؛

پیدا کردن مسیری که خروج از مرکز آن بین بقیه مسیرها کمتر باشد و مجموع فاصله همه راس‌ها تا مسیر از یک مقدار مشخص بیشتر نباشد؛

پیدا کردن همه مسیرهای بهینه پارتو.

[51] Averbakh and Berman

آنها برای حل این سه مساله دو مرحله را در نظر گرفتند. فاز اول یافتن مجموعه M که شامل حداکثر n مسیر بهینه پارتو و غیر پارتو می‌شود(n تعداد راس‌های درخت است). و در فاز دوم مسیر بهینه‌ای‌که جواب مساله اول و دوم است را از بین مسیرهای مجموعه M پیدا می‌کنیم. آنها الگوریتمی را برای مکان‌یابی مسیر بهینه پارتو با پیچیدگی زمانی O(nlogn) ارائه دادند.

بکر و همکاران [۴۳] مساله مکان‌یابی مسیرهای پارتو را با طول محدود بر روی درخت بررسی کردند و الگوریتمی نیز برای آن ارائه دادند. پیچیدگی زمانی این الگوریتم $O(n\log^2 n)$ است.

رونالد[۵۲] و بکر و همکاران [۴۴] مساله مکان‌یابی مسیر سنت- دین[۵۳] را بر روی درخت مطرح کرده‌اند. که همان مساله دوم اورباخ و برمن [۴۲] است. در این مساله مکان‌یابی، هدف یافتن مسیری است که کمترین مجموع فاصله را بین همه مسیرهای شبکه داشته باشد و خروج از مرکز این مسیر از یک مقدار مشخصی کمتر باشد. رونالد و بکر برای این مساله مکان‌یابی الگوریتم خطی ارائه کردند [۴۴]. که از الگوریتم خطی که اورباخ و برمن ارائه کردند آسان‌تر است ولی پیچیدگی زمانی هر دو الگوریتم O(n) است.

مساله دیگری که رونالد و بکر [۴۴] ارائه کردند همان مساله بالا است با این شرط که طول مسیر باید از یک مقدار معینی چون L کمتر باشد که نویسنده برای این مساله الگوریتمی با پیچیدگی زمانی $O(n\log^2 n)$ ارائه کرده است.

---

52 Ronald

53 Cent –dian

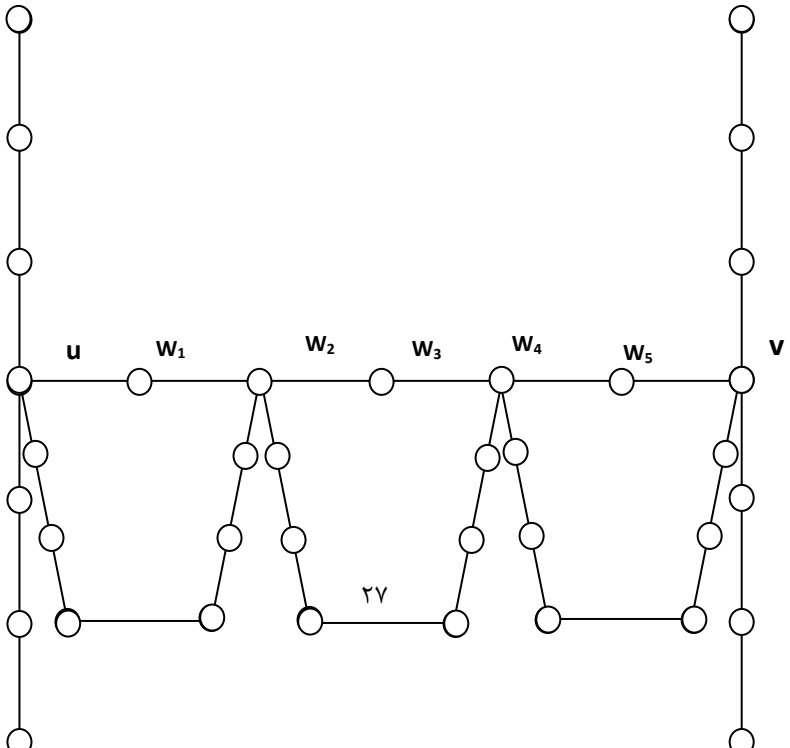
فصل دوم

مکان‌یابی مسیر مرکزی روی درخت

در این فصل به بررسی مساله مکان‌یابی مسیر مرکزی روی درخت می‌پردازیم و الگوریتمی را که اسلیتر برای حل این مساله ارائه داده است، بیان می‌کنیم. مطالب ارائه شده در این فصل برگرفته از مرجع [۲۵] می‌باشد.

۲-۱ مکان‌یابی مسیر مرکزی روی درخت

مفهوم مسیر مرکزی اولین بار توسط اسلیتر و هدتنیمی در سال ۱۹۷۷م به طور جداگانه مطرح شد و سرانجام هدتنیمی [۳۵] و اسلیتر [۲۵] الگوریتم‌های خطی برای این مساله بر روی درخت ارائه کردند. اسلیتر مسیر مرکزی را به گونه زیر تعریف کرده است[۲۵]: مسیر مرکزی روی گراف G مسیری است که کمترین خروج از مرکز را بین بقیه مسیرها داشته باشد و اگر دو مسیر خروج از مرکز یکسان داشتند، مسیری که کمترین طول را داشته باشد مسیر مرکزی خواهد شد. تعریف هدتنیمی همان تعریف اسلیتر است با این تفاوت که شرط کمترین طول را برای مسیر مرکزی در نظر نگرفته است.



شکل ۲-۱: گراف G

در شکل ۲-۱ [۲۵] طبق تعریف هدتنیمی بین u وv هشت مسیر وجود داردکه خروج از مرکز هر هشت مسیر ۳ است. و چون مسیر دیگری یافت نمی‌شود که خروج از مرکز آن کمتر از ۳ باشد پس این هشت مسیر، مسیر مرکزی هستند. اما طبق تعریف اسلیتر فقط مسیر $P=u,w_1,w_2,w_3,w_4,w_5,v$ مسیر مرکزی است.

۲-۲ الگوریتم مسیر مرکزی روی درخت

۲-۲-۱ قضایا

۲-۲-۱ -۱: قضیه جردن [۴۵]

مرکز هر درخت T یا شامل یک راس و یا شامل دو راس مجاور است.

اثبات

از استقرا روی تعداد راس‌ها استفاده می‌کنیم. فرض کنید n تعداد رئوس درخت T باشد. به ازای $n=1,2$ قضیه بدیهی است. برای $n>2$ فرض کنیم T یک درخت n– راسی دلخواه باشد. در درخت T تمام برگ‌ها را حذف می‌کنیم گراف حاصل را که یک درخت می‌باشد، T' می‌نامیم. برای هر راس u در درخت، هر راس در فاصله ماکسیمم از u یک برگ است. (زیرا مسیر میان راس v و دورترین راس x را در نظر می‌گیریم. اگر x یک برگ نباشد، آن گاه دارای همسایه دیگری است که از راس v دورتر است) چون همه برگ‌ها حذف شده‌اند و در مسیری که بین دو راس وجود دارد هیچ برگی وجود ندارد. در نتیجه برای هر $u\in V(T')$ داریم

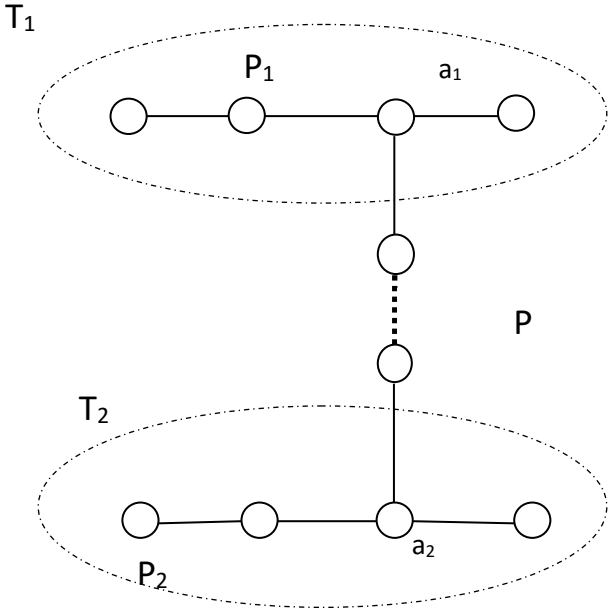
$$e_{T'}(u)=e_T(u)-1.$$

و همچنین خروج از مرکز یک برگ در T بزرگتر از خروج از مرکز همسایه‌اش در T است از این رو راس‌های که $e_T(u)$ را کمینه می‌کنند، همان راس‌هایی هستند که $e_T(u)$ را کمینه می‌کنند و بنابر فرض استقرا برای T' آنها متشکل از یک راس منفرد یا دو راس مجاور می‌باشند. پس مرکز درخت T نیز متشکل از یک راس یا دو راس مجاور است. □

قضیه۲-۲-۱-۲ : مسیرمرکزی درخت T منحصربه فرد است [۳۵].

اثبات. از برهان خلف استفاده می‌کنیم. فرض کنیم P_1 و P_2 دو مسیر مرکزی درخت T باشند. این دو مسیر یا راس مشترک با هم دارند یا هیچ راس مشترکی با هم ندارند.

الف- فرض کنید دو مسیر P_1 و P_2 هیچ راس مشترکی با هم نداشته باشند. (طبق شکل ۲-۲)



شکل ۲-۲: مولفه هایی که از حذف یال های مسیر P به دست آمده است.

از مسیر P_1 به P_2 یک مسیر مثل P با راس ابتدایی و انتهایی a_1 وa_2 در نظر می‌گیریم. و فرض کنید T^*جنگلی باشد که از حذف همه یال‌های مسیر P از T به دست آمده است.$i=1,2$ T_iمولفه‌هایی از T^* هستند که شامل P_i هستند.

راس$x_i \in V(T_i)$ را به گونه‌ای در نظر می‌گیریم که

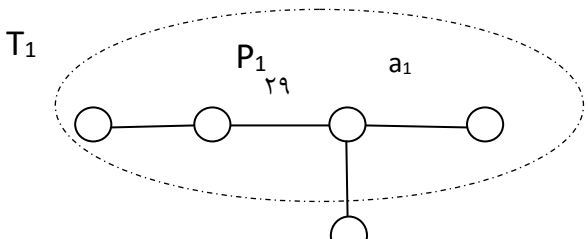
$$2, x \in V(T_i)\}. \ 1, = d(x , a_i)\}, i \ d\ (x_i , a_i)=max\ \{$$

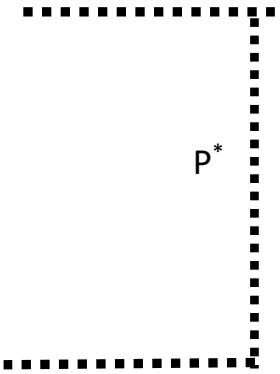
حال مسیر P^* را از x_1 به x_2 در نظر می‌گیریم. و ادعا می‌کنیم که

$$e(P^*) < e(P_1) = e(P_2). \qquad (۱-۲)$$

اگر$y \in V(T_1)$ طبق شکل ۲-۳ و با توجه به تعریف x_1,x_2 خواهیم داشت که

$$d(y,P^*) \leq d(y,a_1). \qquad (۲-۲)$$





شکل ۲-۳ : مولفه ای که از حذف یال های مسیر P به دست آمده همراه با مسیر P^*

پس در حالت کلی خواهیم داشت

$$d(y,P^*) \le d(y,a_1) \le d(x_1,a_1) < d(x_1,P_2) \le e(P_2). \qquad (۳-۲)$$

بطور مشابه اگر $y \in V(T_2)$ باشد آنگاه طبق شکل ۲-۳ و با توجه به تعریف x_1,x_2 خواهیم داشت

$$d(y,P^*) \le d(y,a_2) \le d(x_2,a_2) < d(x_2,P_1) \le e(P_1). \qquad (۴-۲)$$

تا اینجا اثبات کردیم که فاصله همه راس‌های T_1,T_2 تا مسیر P^* کمتر از $e(P_1),e(P_2)$ است.

راسی مثل z را از مسیر P در نظر می‌گیریم به طوری که در T_1,T_2 نباشد، آنگاه خواهیم داشت که

$$d(z,P^*) < d(z,P_i) \le e(P_i) \ , \quad i=1,2 \qquad (۵-۲)$$

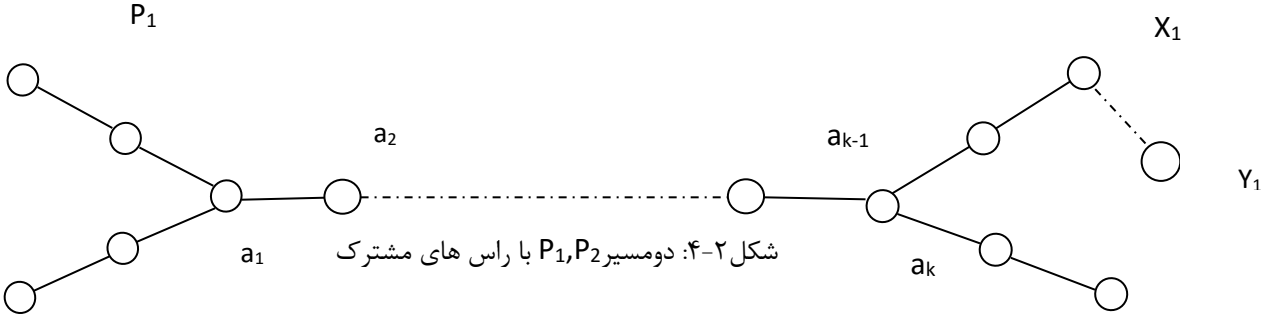
سر انجام برای همه راس‌های $z \in P$ داریم $d(z,P^*)=0$

بنابراین خواهیم داشت که

$$e(P^*) < e(P_1) = e(P_2) \qquad (۶-۲)$$

پس طبق رابطه ۲-۶ مسیری مثل P^* پیدا کردیم به گونه ای که خروج از مرکز آن کمتر از $e(P_1),e(P_2)$ است. و این خلاف فرض است. پس فرض خلف باطل است و $P_1=P_2$.

ب)حالتی را در نظر می‌گیریم که دو مسیر P_1,P_2 در راس‌های $a_1,\ldots a_k$ مشترک باشند (طبق شکل ۲-۴) و به دلیل اینکه شبکه مورد نظر درخت است و درخت فاقد دور است پس رئوس مشترک بین دو مسیر P_1,P_2 یعنی رئوس $a_1\ldots,a_k$ خود یک مسیر در T تشکیل می‌دهند.



شکل۲-۴: دومسیرP_1,P_2 با راس های مشترک

فرض می‌کنیم که x_1 راس پایانی مسیر P_1 باشد به قسمی که $x1\notin\{a_1,a_2..,a_k\}$.چون مسیر P_1 کمترین خروج از مرکز را دارد، پس راسی مثل $y_1\in V(T)$ وجود دارد به قسمی که کوتاهترین مسیر بین y_1 و مسیر P_1 از x_1 می‌گذرد و خواهیم داشت

$$d(y_1,x_1) = e(P_1). \qquad (۷-۲)$$

بنابراین با توجه به رابطه فوق داریم

$$e(P_2) \geq d(y_1,P_2) > d(y_1,x_1) = e(p_1). \qquad (۸-۲)$$

که متناقض با فرض است.

بنابراین راس پایانی مسیر P_1 متعلق به مسیر $a_1,a_2\ldots,a_k$ است. برای مسیر P_2 نیز به همین گونه اثبات می‌شود که راس پایانی P_2 متعلق به مسیر $a_1,a_2,\ldots,a_k$ است. بنابراین داریم $P_1=P_2$ □.

قضیه۲-۲-۱-۳ مسیر مرکزی هر درخت شامل مرکز درخت نیز می‌شود [۳۵].

اثبات. از برهان خلف استفاده می‌کنیم. فرض کنید u مرکز درخت و P مسیر مرکزی درخت باشد. فرض می‌کنیم که راس u جزء مسیر مرکزی نباشد، طبق خاصیت درخت که بین هر دو راس یک درخت یک مسیر منحصر به فرد وجود دارد، پس بین u و P مسیری مثل q وجود دارد. با حذف همه یال‌های مسیر q از درخت T جنگل T^* را تشکیل می‌دهیم. در این جنگل مولفه‌ای که شامل u باشد را T_u می‌نامیم. به دلیل اینکه u مرکز درخت است، پس راسی مثل x در T_u وجود دارد به قسمی که

$$d(x,u) \geq e(u) - 1. \quad (۹-۲)$$

$$e(P) \geq d(x,P) > d(x,u) \geq e(u)-1. \quad (۱۰-۲)$$

با توجه به رابطه فوق داریم

$$e(P) \geq e(u). \quad (۱۱-۲)$$

اما چون P مسیر مرکزی است داریم $e(P) \leq e(u)$ که مخالف با نامساوی (۲-۱۱) است. پس فرض خلف باطل است و مسیر مرکزی درخت شامل مرکز درخت نیز می‌شود. □

۲-۲-۲ نمادگذاری

فرض کنید $P=v_1,v_2,\ldots,v_k$ یک مسیر در درخت T باشد، آنگاه:

الف) برای $i=1,k$ درخت‌های T_1 و T_k به صورت زیر تعریف می‌شوند (که شامل راس v_1,v_k نیز می‌شود) و داریم $e_{12}= (v_1,v_2)$

$$T_1=T_k=T-(v_1,v_2)-(v_{k-1},v_k). \quad (۱۲-۲)$$

ب) برای $i=2,3,.,K-1$ درخت T_i به صورت زیر تعریف می‌شود (که شامل راس v_i نیز می‌شود)

$$T_i=T-(v_{i-1},v_i)-(v_i,v_{i+1}). \quad (۱۳-۲)$$

ج) برای $i=1,2,.,K$ می‌توان تعریف کرد

$$e_p(v_i)=e(v_i ,T_i). \quad (۱۴-۲)$$

۲-۲-۳ الگوریتم مسیر مرکزی روی درخت PCT [۵۴] [۲۵]

در این الگوریتم $w_i=1$ و $w'_{ij}=1$ فرض شده است. و $|V(T)| \geq 3$

گام صفر. مرکز درخت را مشخص کنید

گام اول. اگر مرکزدرخت Tشامل دو راس مجاور{u,v} باشد، قراردهید

$$u,v =P \text{ ، } 2=k$$

---

[54] Path center of a tree

و به گام سوم بروید.

گام دوم. اگر مرکز درخت فقط شامل راس {u} باشد، در این صورت اگر در سه مولفه از T-u راسی مانند v وجود داشته باشد به طوری که دارای خاصیت زیر باشد

$d(v,u)=e(u,T)$.

(یعنی راسی مثل v وجود داشته باشد که فاصله‌اش تا مرکز برابر خروج از مرکز راس مرکزی باشد.)

آنگاه قرار دهید: P= u و به گام پنجم بروید.

و اگر دقیقا دو مولفه از T-u شامل راسی باشند که خاصیت فوق را داشته باشد، در این صورت می‌توان مسیر مرکزی را از مرکز درخت گسترش داد و راس‌هایی که با u در این دو مولفه مجاور هستند جزء مسیر مرکزی قرار دهید. K=2 و به گام سوم بروید.

گام سوم. (بررسی شروط توقف الگوریتم) اگر راسی مثل i وجود داشته باشد که

$$e_p(v_i)=e_p(v_1)=e_p(v_k)\ ,\ 2\le i\le k-1 \qquad (۱۵-۲)$$

دراین صورت به گام پنجم بروید.

و یا اگر در هر کدام از گراف‌های T_1-v_1 و T_k-v_k بتوان دو مولفه یا بیشتر پیدا کرد به طوری که این مولفه‌ها شامل راسی باشند که در خاصیت زیر صدق کنند.

$$e_p(v_h)=d(v,v_h):v\in V(T). \qquad (۱۶-۲)$$

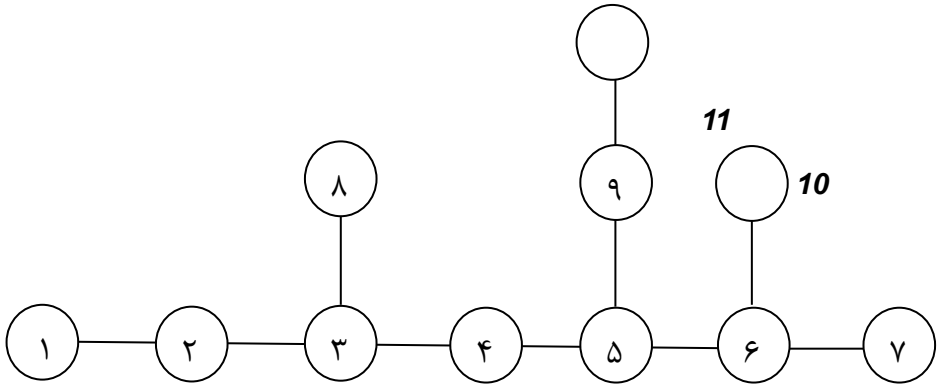
در این صورت نمی‌توان مسیر مرکزی را گسترش داد و به گام پنجم بروید.

گام چهارم. اگر در هر کدام از گراف T_1-v_1 و T_k-v_k دقیقا یک مولفه شامل راسی مثل v باشد به طوری که در رابطه فوق صدق کند در این صورت راس‌هایی که به v_1و v_h وصل هستند را جزء مسیر مرکزی قرار دهید و به k دو واحد اضافه کنید و به گام سوم بروید. (برای توسعه مسیر مرکزی حتما از هر طرف مسیر مرکزی اولیه باید فقط یک مولفه واجد شرایط وجود داشته باشد تا بتوانیم مسیر مرکزی را گسترش دهیم در غیر این صورت با یکتا بودن مسیر مرکزی متناقض خواهد شد).

گام پنجم. P مسیرمرکزی درخت T است.

۲-۳ مثال[۲۵]

درخت T در شکل ۲-۵ را در نظر بگیرید. وزن همه یال‌ها ۱ فرض شده است.



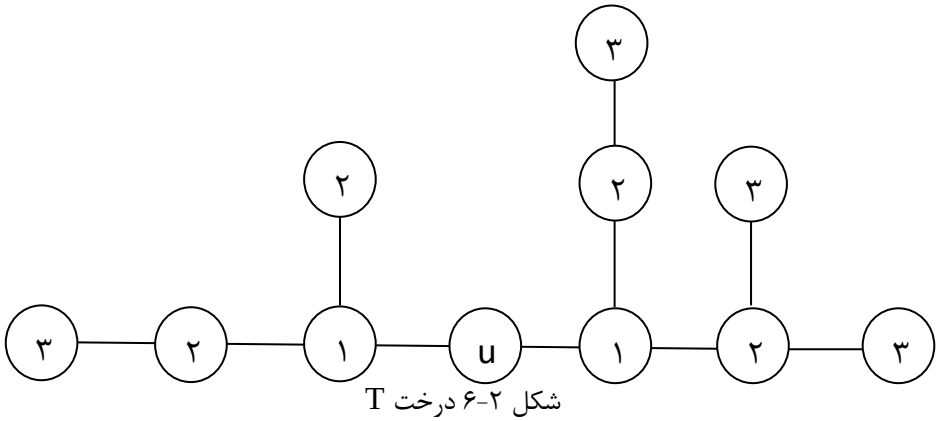
شکل ۲-۵ درخت T

گام صفر. برای اینکه مرکز درخت را مشخص کنیم باید خروج از مرکز همه راس‌ها را حساب کنیم.

e(1,G)=6	e(5,G)=4	e(9,G)=5
e(2,G)=5	e(6,G)=5	e(10,G)=6
e(3,G)=4	e(7,G)=6	e(11,G)=6
e(4,G)=3	e(8,G)=5	

$\text{Min}\{e_i\}=3$ Center vertex=4.

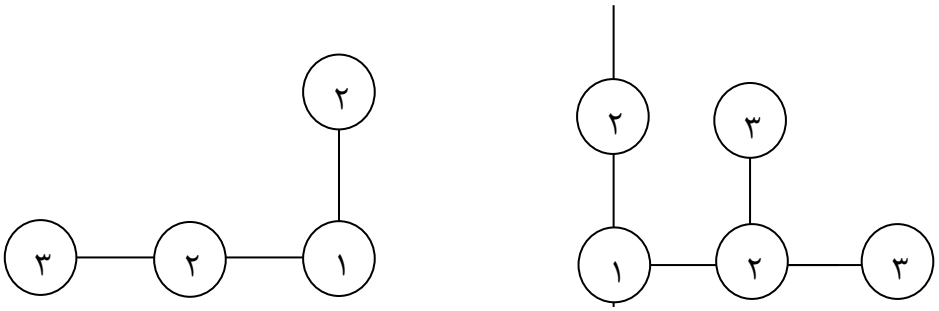
به دلیل اینکه کمترین خروج از مرکز عدد ۳ است پس مرکز گراف راس ۴ خواهد بود. در شکل ۲-۶ عدد داخل هر راس نشان‌دهنده فاصله هر راس تا مرکز گراف (u) است.



شکل ۲-۶ درخت T

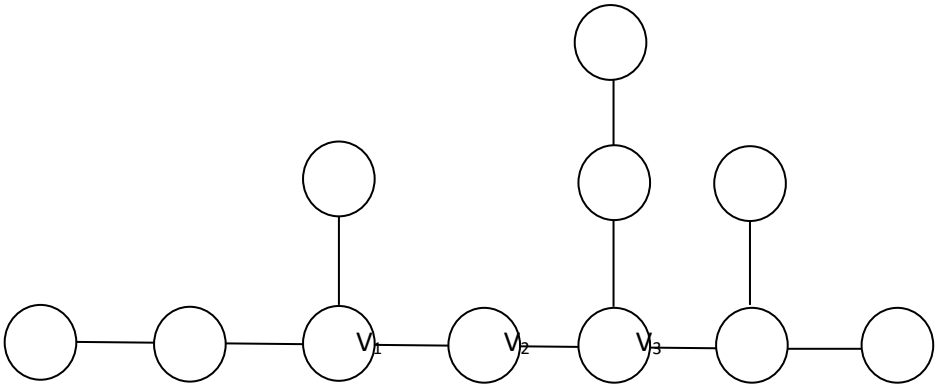
حال طبق الگوریتم چون مرکز گراف یک راس است پس وارد گام دوم می‌شویم.

گام دوم. در این گام باید T-u را بررسی کنیم. در شکل ۲-۷ عدد داخل هر راس نشان دهنده فاصله هر راس تا مرکز درخت T است.



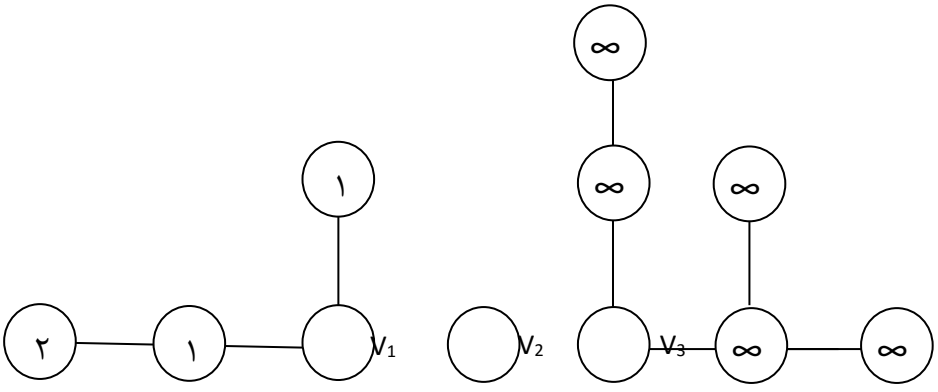
شکل ۲-۷ گراف T-u

چون $e(u,T)=3$ اســـت، پس باید در T-u به دنبال راس‌هایی در مولفه‌های مجزا باشـــیم به طوری که فاصـــله آن راس تا مرکز برابر ۳ باشـــد. چون تنها دو مولفه وجود دارد که دارای خاصـــیت فوق اســـت، پس می‌توان مسیر مرکزی را گسترش داد و قرار دهید $k=3$ و به گام سوم بروید.



شکل ۲-۸ مسیر مرکزی v_1,v_2,v_3

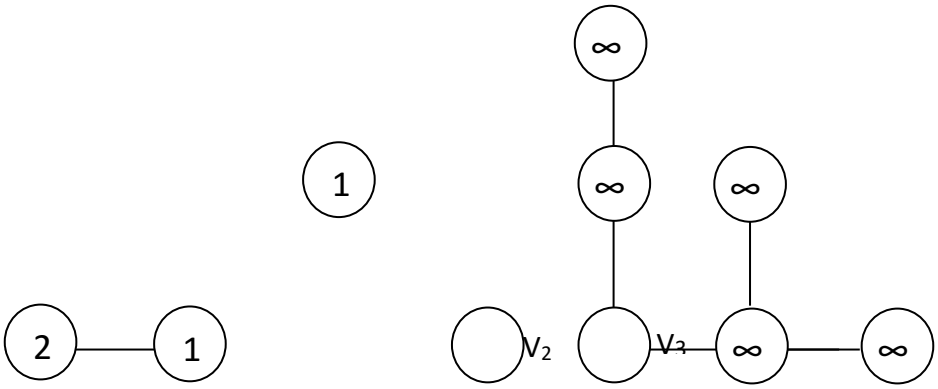
گام سوم. در گام سوم باید T_1-v_1 وT_3-v_3 را بررسی کنیم.



$T_1 = T-(v_1,v_2)-(v_2,v_3)$شکل۲-۹ گراف

در شکل ۲-۹ و ۲-۱۰ عدد داخل هر راس نشان دهنده فاصله هر راس تا راس V_1 در گراف T_1 است. با توجه به شکل ۲-۹ داریم

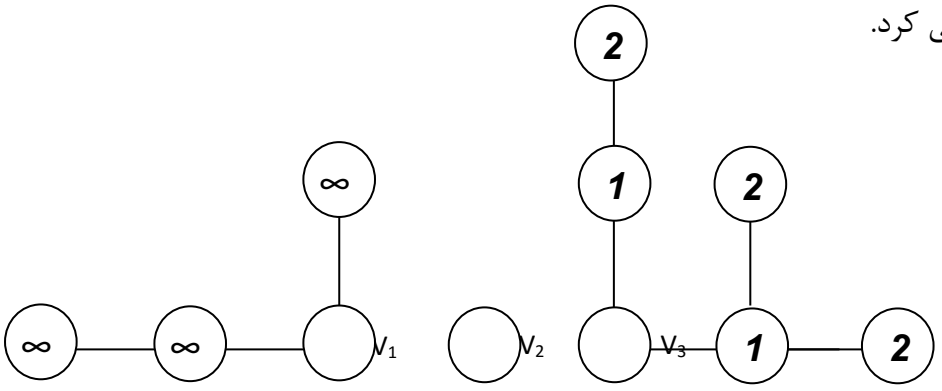
$e_p(v_1)=e(v_1,T_1)=2$.



شکل ۲-۱۰ گراف T_1 -v_1

با توجه به مقدار $e_p(v_1)$ در گراف T_1-v_1 به دنبال مولفه‌ای هستیم که شامل راسی باشد که فاصله‌اش تا راس v_1 در گراف T_1 برابر مقدار $e_p(v_1)$ باشد. با توجه به شکل ۲-۱۰ فقط یک مولفه وجود دارد که دارای خاصیت فوق است. پس از راس v_1 می توان مسیر مرکزی را گسترش داد.

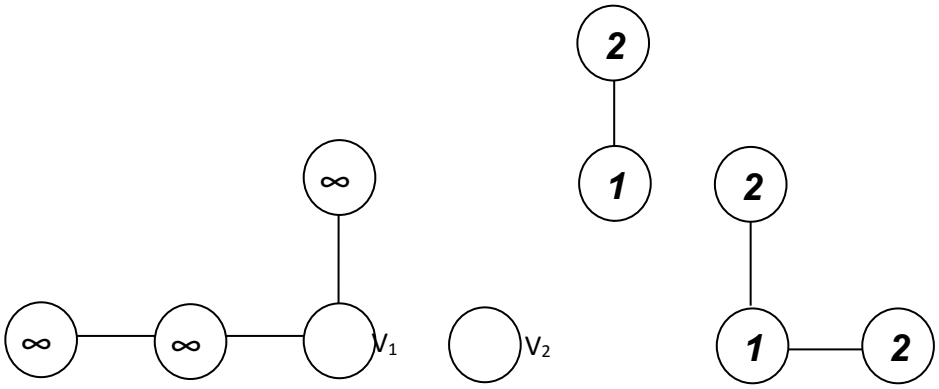
حال بررسی می‌کنیم که آیا می‌توان از راس v_3 نیز مسیر مرکزی را گسترش داد. برای اینکار باید گراف T_3-v_3 را بررسی کرد.



شکل ۲-۱۱ گراف $T_3 =T-(v_1,v_2)-(v_2,v_3)$

در شکل ۲-۱۱ عدد داخل هر راس نشان دهنده فاصله هر راس تا راس v_3 در گراف T_3 است. و با توجه به شکل ۲-۱۱ داریم

$e_p(v_3)=e(v_3,T_3)=2$.

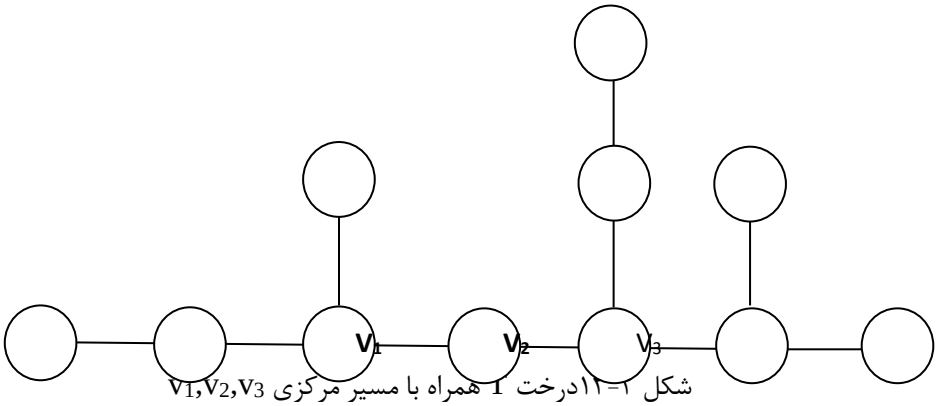


شکل ۲-۱۲گراف T_3-v_3

در گراف T_3-v_3 به دنبال مولفه‌ای هستیم که شامل راسی باشد که فاصله‌اش تا راس v_3 در گراف T_3 برابر مقدار $e_p(v_3)$ باشد. چون در گراف T_3-v_3 دو مولفه وجود دارد که دارای خاصیت فوق است پس طبق گام سوم نمی‌توان مسیر مرکزی را گسترش داد. پس به گام پنجم بروید.

گام پنجم. P مسیر مرکزی درخت T است.

$P = v_1,v_2,v_3$



شکل ۱-۱۱درخت T همراه با مسیر مرکزی v_1,v_2,v_3

فصل سوم

الگوریتم ژنتیک

برای مساله مسیر مرکزی روی درخت الگوریتم‌های خطی ارائه شده است. اما مساله مسیر مرکزی روی شبکه یک مساله NP- سخت[55] است[۳۸]. به همین دلیل ما در این پایان‌نامه الگوریتم ژنتیک[56] و الگوریتم

[55] Np-hard
[56] Genetic Algorithms(GA)

ترکیبی برای حل این مساله و مساله مسیر مرکزی با طول محدود روی شبکه پیشنهاد و نتایج این دو الگوریتم را بررسی و با هم مقایسه خواهیم کرد. در این فصل الگوریتم ژنتیک را برای این دو مساله را بیان خواهیم کرد.

۳-۱ الگوریتم ژنتیک چیست؟

الگوریتم ژنتیک یک الگوریتم جستجوی تصادفی است که از اصول انتخاب طبیعی داروین برای یافتن فرمول بهینه جهت پیش‌بینی یا تطبیق الگو استفاده می کند. الگوریتم ژنتیک در سال‌های اخیر به طور گسترده‌ای برای مسائل بهینه‌سازی مخصوصا در مورد مسائل مکان‌یابی به عنوان مثال مسائل میانه[۵۷] و مسائل مکان‌یابی واسطه[۵۸] به کار گرفته شده است [۴۶،۴۷،۴۸،۴۹].

۳-۲ ایده اصلی

در دهه هفتاد میلادی دانشمندی از دانشگاه میشیگان به نام جان هولند[۵۹] ایده استفاده از الگوریتم ژنتیک را در بهینه‌سازی رشته‌های مهندسی مطرح کرد. ایده اساسی این الگوریتم انتقال خصوصیات موروثی توسط ژن‌ها[۶۰] است. فرض کنید مجموعه خصوصیات انسان توسط کروموزوم‌های[۶۱] او به نسل بعدی منتقل می‌شوند. هر ژنی در این کروموزوم‌ها نماینده یک خصوصیت است. اگر همه این کروموزوم به نسل بعد انتقال یابند، تمامی خصوصیات نسل بعدی شبیه به خصوصیات نسل قبل خواهد بود. بدیهی است که در عمل چنین اتفاقی رخ نمی‌دهد. در واقع بصورت همزمان دو اتفاق برای کروموزوم‌ها می‌افتد. اتفاق اول ترکیب [۶۲] است. یعنی چسبیدن ابتدای یک کروموزوم به انتهای یک کروموزوم دیگر. این همان چیزی است که مثلا باعث می‌شود تا فرزند[۶۳] تعدادی از خصوصیات پدر و تعدادی از خصوصیات مادر را با هم به ارث ببرد و از شبیه شدن تام فرزند به تنها یکی از والدین جلوگیری می‌کند. اتفاق دوم جهش[۶۴] است. جهش به این صورت است

---

57 Median problem
58 Hub location problem
59 John Holland
60 Gens
61 Chromosom
62 Crossover
63 Child or offspring
64 Mutation

که بعضی ژن‌ها بصورت کاملا تصادفی تغییر می‌کنند. البته تعداد این گونه ژن‌ها بسیار کم می‌باشند. اما در هر حال این تغییر تصادفی بسیار مهم است. عمل ترکیب به تعداد بسیار بیشتری نسبت به عمل جهش رخ می‌دهد. برای مطالعه بیشتر می‌توانید به کتاب‌های گلدبرگ ***[۵۰]*** و ریوز ***[۵۱]*** مراجعه کنید. مطالب این بخش برگرفته از مرجع [۵۲] است.

۳-۳ چارچوب کلی الگوریتم ژنتیک

موتور الگوریتم ژنتیک در حالت کلی با یک مجموعه از جواب‌های ممکن اولیه (که به صورت تصادفی و یا به صورت ابتکاری تولید می‌شوند) و به آن اصطلاحا جمعیت اولیه[۶۵] گوییم ، شروع می‌شود. عناصر هر جمعیت را اصطلاحا نفر[۶۶] گوییم. الگوریتم ژنتیک با انتخاب تعدادی از زوج‌های جمعیت جاری به عنوان والدین[۶۷] و انجام عمل ترکیب بر روی آنها چند جواب ممکن یا فرزند از این والدین‌ها برای جمعیت جدید تولید می‌کند. علاوه بر این تعدادی از جواب‌ها با تطابق[۶۸] بسیار کم و یا با مقدار تابع هدف زیاد از جمعیت جاری خارج می‌شوند. این فرایند ادامه دارد تا اینکه شرط پایانی برقرار شود. در حین اجرای الگوریتم جهش‌ها و مهاجرت‌ها[۶۹] با تغییر بعضی از افراد و یا جایگزین کردن آنها با افراد بهتر صورت می‌گیرد. گاهی متناوبا از روش‌های بهینه محلی نیز برای بهبود جمعیت استفاده می‌شود. که اصطلاحا این الگوریتم‌ها را الگوریتم‌های ادغامی[۷۰] گوییم. جستجو در فضا را می‌توان به چند قسمت تقسیم کرد که اصطلاحا به این کار رقابت[۷۱] گوییم. یک رقابت یعنی اجرای الگوریتم ژنتیک با جمعیت‌های اولیه متفاوت و توقف آنها قبل از همگرایی و سپس ساختن یک جمعیت بهتر با انتخاب جواب‌های نهایی هر یک از این اجراها [۵۳].

۳-۴ روش اجرای الگوریتم

۳-۴-۱ تعیین نحوه نمایش یا کدبندی[۷۲] مساله

---

65 Initial Population
66 Individual
67 Parent
68 Fitness
69 Immigration
70 Hybrid Algorithms
71 Tournament
72 Coding

باید به هر جواب، یک نماد ژنتیکی[۷۳] اختصاص داد. عموماً کدگذاری‌ها به صورت ۲ تایی صفر و یک نشان داده می‌شوند، ولی روش‌های نمایش دیگری هم وجود دارند. نحوه نمایش، ارتباط مستقیم با مساله مورد نظر دارد و معمولا در رسیدن به جواب مناسب اثر بسیاری داشته و باید با دقت انتخاب شود.

۳-۴-۱-۱ انواع روش‌های نمایش

قبل از این که یک الگوریتم ژنتیک برای یک مساله اجرا شود، یک روش برای کدکردن ژن‌ها به زبان کامپیوتر باید به کار رود. یکی از روش‌های معمول کدکردن به صورت رشته‌های باینری است یعنی رشته‌های ۰و۱. یک راه حل مشابه دیگر کدکردن جواب‌ها در آرایه‌ای از اعداد صحیح یا اعشاری است. این راه حل در مقایسه با قبلی پیچیده‌تر و مشکل‌تر است. مثلاً این روش توسط استفان کرمر، برای حدس ساختار سه بعدی پروتئین موجود در آمینواسیدها استفاده شده است. همچنین الگوریتم‌های ژنتیکی که برای آموزش شبکه‌های عصبی استفاده می‌شوند، از این روش استفاده می‌کنند. سومین روش برای نمایش صفات در یک الگوریتم ژنتیک یک رشته از حروف است، که هر حرف نمایش دهنده یک خصوصیت از جواب‌ها است.

۳-۴-۲ تعریف میزان برازندگی

ارزیابی هر رشته از لحاظ میزان بهینگی بر اساس نحوه رفتار[۷۴] حاصل از آن و از طریق محاسبه تابع هدف و اختصاص عدد برازندگی به آن صورت می گیرد.

۳-۴-۳ تولید فرزند

یک گام مهم دیگر در الگوریتم، تولد است که در هر دوره اتفاق می‌افتد. برای هر فرد، یک جفت والد انتخاب می‌شود. انتخاب‌ها به گونه‌ای هستند که مناسب‌ترین عناصر انتخاب شوند تا حتی ضعیف‌ترین عناصر هم شانس انتخاب شدن داشته باشند. این کار باعث می‌شود که از نزدیک شدن به جواب محلی جلوگیری

---

73 Genotype
74 Phenotype

شود. چندین الگوی انتخاب وجود دارد: چرخ منگنه‌دار(رولت)[۷۵]، انتخاب مسابقه‌ای[۷۶] و غیره. بعد از عمل انتخاب والدین با توجه به عملگرهای ژنتیکی مثل ترکیب و جهش فرزند تولید می‌کنیم.

۳-۴-۳-۱ روش‌های انتخاب

روش‌های مختلفی برای انتخاب کروموزم‌ها در الگوریتم‌های ژنتیک وجود دارند. اما روش‌های لیست شده در زیر از معمول‌ترین روش‌ها هستند.

انتخاب نخبه‌گرایی[۷۷]. مناسب‌ترین عضو هر اجتماع انتخاب می‌شود.

انتخاب رولت. یک روش انتخاب است که در آن عنصری که عدد برازش (تناسب) بیشتری داشته باشد، انتخاب می‌شود.

انتخاب مقیاسی[۷۸]. به موازات افزایش متوسط عدد برازش جامعه، سنگینی انتخاب بیشتر و جزئی‌تر می‌شود. این روش وقتی کاربرد دارد که مجموعه دارای عناصری باشد که عدد برازش بزرگی دارند و فقط تفاوت‌های کوچکی، آن‌ها را از هم تفکیک می‌کند.

انتخاب مسابقه‌ای. یک زیرمجموعه از صفات یک جامعه انتخاب می‌شوند و اعضای آن مجموعه با هم رقابت می‌کنند و سرانجام فقط یک صفت از هر زیر گروه برای تولید انتخاب می‌شوند.

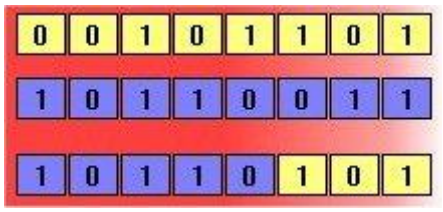
۳-۴-۴ روش‌های تغییر

وقتی با روش‌های انتخاب، کروموزوم‌ها انتخاب شدند، باید به طور تصادفی برای افزایش تناسبشان اصلاح شوند. دو راه حل اساسی برای این کار وجود دارد. اولین روش ترکیب نام دارد. دو کروموزوم برای معاوضه بخشی[۷۹] از کدشان انتخاب می‌شوند. این فرآیند براساس فرآیند ترکیب کروموزوم‌ها، در طول تولیدمثل موجودات زنده شبیه‌سازی شده است. اغلب روش‌های معمول ترکیب، شامل برش یک‌نقطه‌ای[۸۰] هستند، که

---

75 Roulette
76 Tournament
77 Elitist
78 Scaling
79 Segment
80 Single-point Crossover

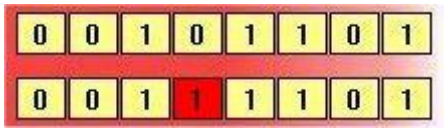
نقطه تعویض در جایی تصادفی بین ژن‌ها است. بخش اول قبل از نقطه، و بخش دوم بعد از نقطه ادامه پیدا می‌کند، که هر قسمت برگرفته از یک والد است، که ۵۰/۵۰ انتخاب شده‌اند.



۳-۱ :تاثیر عملگر ترکیب بر روی کروموزوم‌ها

کروموزوم را نشان می‌دهد که نقطه تعویض بین ۵ امین و ۶ امین مکان در کروموزوم قرار گرفته است. ایجاد یک کروموزوم جدید از پیوند این دو والد بدست می‌آید. معمولا عملگر ترکیب را به درصدی از جمعیت اعمال می‌کنند، که این درصد را با P_c نشان می‌دهند. و معمولا $P_c \in (0.5 , 1)$

دومین راه جهش نامیده می‌شود. به دلایل مختلف ممکن است تنوع و گوناگونی ژنی از بین برود و بازیابی آن با عملگر ترکیب غیرممکن شود. مثلا ممکن است در جمعیت تصادفی اولیه، همه رشته‌ها در یک ژن مخصوص مقدار یک داشته باشند ولی جواب بهینه در همان ژن مقدار صفر را داشته باشد. به این ترتیب هرگز به جواب بهینه نمی‌رسیم. بنابراین افزایش تنوع ژنی لازم و ضروری است. و این مهم با جهش صورت می‌گیرد. به عبارت دیگر اگر دست‌یابی به بعضی از نقاط فضای جستجو با عمل ترکیب ممکن نباشد، با جهش می‌توان به آن نقاط رسید. شکل زیر کروموزومی را نشان می‌دهد که ژن چهارم آن دچار جهش شده و صفر در آن مکان به یک تبدیل شده است.



۳-۲ :تاثیر عملگر جهش بر روی ژن چهارم

برای اعمال جهش:

۱- رشته‌ای از جمعیت را به تصادف انتخاب می‌کنیم؛

۲- یک ژن از آن رشته را به تصادف انتخاب می‌کنیم؛

۳- این ژن را تغییر می‌دهیم (مثلا در نمایش دودویی صفر به یک یا یک به صفر تغییر خواهند کرد).

نحوه اعمال جهش بر روی رشته بستگی به مساله و کدگذاری آن دارد. این عملگر روی درصدی از جمعیت و گاهی به صورت متناوب بر روی درصدی از جمعیت عمل می کند. این درصد را با P_m نمایش می‌دهیم. دو مقدار خوب برای این احتمال جهش یک بیت به صورت زیر پیشنهاد شده است:

$$P_m=\left(\sqrt{L}\times M\right)^{-1} \text{ یا } P_m=\frac{1}{L}$$

که در آن M، تعداد جمعیت و L طول کروموزم است. این فرآیندها باعث به وجود آمدن نسل جدیدی از کروموزوم‌هایی می شوند، که با نسل قبلی متفاوت هستند. این فرآیند تغییر تکرار می‌شود تا این‌که به آخرین مرحله برسیم.

۳-۴-۵ مشخص کردن شرط پایان

شرایط خاتمه الگوریتم‌های ژنتیک عبارتند از:

۱. به تعداد ثابتی از نسل‌ها برسیم.
۲. بودجه اختصاص داده شده تمام شود (زمان محاسبه/پول).
۳. یک فرد(فرزند تولید شده) پیدا شود که مینیمم (کمترین) ملاک را برآورده کند.
۴. بیشترین درجه برازش فرزندان حاصل شود. یا دیگر نتایج بهتری حاصل نشود.
۵. ترکیب‌هایی از شرایط بالا.

۳-۵ چند نمونه از کاربردهای الگوریتم ژنتیک

نرم‌افزار شناسایی چهره با استفاده از تصویر ثبت شده به همت مبتکران ایرانی طراحی و ساخته شد. در این روش، شناسایی چهره براساس فاصله اجزای چهره و ویژگی‌های محلی و هندسی صورت می‌گیرد که تغییرات ناشی از تغییرات نور و افزایش سن کمترین تأثیر را خواهند داشت.

همچنین با استفاده از الگویتم‌های ژنتیک گراف‌هایی برای چهره‌های جدید ساخته شده است و با استفاده از یک تابع تشابه، قابل مقایسه با یکدیگر هستند که این امر تأثیر بسیار زیادی در افزایش سرعت شناسایی خواهد داشت.

از الگوریتم ژنتیک برای حل مسائل زیر نیز استفاده شده است:

توپولوژی‌های شبکه‌های کامپیوتری توزیع شده؛

بهینه‌سازی ساختار ملکولی شیمیایی؛

مهندسی برق برای ساخت آنتن مخصوص؛

مهندسی نرم‌افزار؛

بازی‌های کامپیوتری؛

مهندسی مواد؛

مهندسی سیستم؛

رباتیک[۸۱]؛

تشخیص الگو و استخراج داده؛[۸۲]

حل مساله فروشنده دوره‌گرد؛

آموزش شبکه‌های عصبی مصنوعی؛

یاددهی رفتار به ربات‌ها با استفاده از الگوریتم ژنتیک؛

یادگیری قوانین فازی با استفاده از الگویتم‌های ژنتیک؛

برای کسب اطلاعات بیشتر و کامل‌تر به آدرس www.talkorigins.org مراجعه شود.

۳-۶ کاربرد در جهان واقعی

81 Robotics

82 Data mining

مثال‌های بسیاری وجود دارد که برای حل آن‌ها از الگوریتم ژنتیک استفاده شده است. از جمله می‌توان به مساله فروشنده دوره‌گرد[۸۳] مساله زمان‌بندی[۸۴] و مساله بسته‌بندی[۸۵] اشاره کرد. مساله‌ی جدول زمانی پرسنل و کارکنان با استفاده از الگوریتم ژنتیک به صورت موفقیت‌آمیزی حل شده است. از الگوریتم ژنتیک برای لیست کاری پرستاران در یکی از بیمارستان‌های بزرگ انگلستان نیز استفاده شده است.

۳-۶-۱ کاربرد الگوریتم ژنتیک در پرتودرمانی

همچنین در ده‌های اخیر الگوریتم ژنتیک به عنوان ابزاری قوی برای بهینه‌سازی مسائل پرتودرمانی بکار رفته است. ایزل[۸۶] ، الگوریتم ژنتیک را برای یافتن ترکیب بهینه‌ای از پرتوهای خارجی بکار برد [۵۴]. پس از آن لانگر[۸۷] از الگوریتم ژنتیک برای بهینه‌سازی تابع وزن پرتوها در درمان تومورهای بطنی استفاده کرد [۵۵]. ژیو و همکاران[۸۸] تکنیکی را برای بهینه‌سازی جهت و تابع وزن پرتوها با استفاده از الگوریتم ژنتیک ارائه کردند [۵۶]. از الگوریتم ژنتیک در بهینه‌سازی زاویه تابش پرتوهای پر انرژی نیز استفاده می‌شود. در حال حاضر در کلینیک‌های تخصصی پرتودرمانی برای تعیین زاویه تابش پرتوها از روش‌های تجربی استفاده می‌کنند. این در حال است که به طور قطع برای تعداد محدود و کم پرتوهای تابنده، زاویه تابش نقش مهمی را در عملکرد این روش تعیین می‌کند. بنابراین برای بهینه‌سازی تابع هدف، نیاز به یک الگوریتم انتخاب خودکار زاویه تابش ضروری می‌باشد. در این الگوریتم با توجه به پیچیدگی مساله و نامحدود بودن دامنه جستجو، بهینه‌سازی زاویه تابش و بهینه‌سازی نقشه شدت پرتوها به عنوان دو مساله جداگانه در نظر گرفته می‌شوند که برای بهینه‌سازی زاویه تابش و در بهینه‌سازی تابع وزن پرتوها به همراه ضریب اهمیت ارگان‌ها از الگوریتم ژنتیک استفاده می‌کنند [۵۷].

از الگوریتم ژنتیک برای برنامه‌ریزی فرآیند مبتنی بر کامپیوتر[۵۸] و برای انتخاب یک مجموعه دارایی از سهام بورس اوراق بهادار نیز استفاده می‌شود[۵۹].

---

83 Traveling salesman problem
84 Scheduling problem
85 Bin packing problem
86 Ezzell
87 Langer
88 Zhu at el

۳-۷ الگوریتم پیشنهادی برای مساله مسیر مرکزی

۳ -۷-۱ کدگذاری

در کدگذاری که ما برای مساله مکان‌یابی مسیر مرکزی روی شبکه با استفاده از الگوریتم ژنتیک انجام می‌دهیم، کروموزم‌ها دارای تعداد ژن‌های مختلف هستند که ترتیب قرار گرفتن آنها مهم است. هر کروموزوم نشان دهنده یک مسیر روی شبکه است. و هر ژن متناظر است با اندیس راسی که در مسیر مورد نظر قرار می‌گیرد. به عنوان مثال کروموزم {۱،۲،۶،۹} با کروموزم {۲،۶،۹،۱} متفاوت است زیرا کروموزم اولی معرف مسیری روی شبکه از راس v_1 تا راس v_9 است که از راس‌های v_2 و v_6 نیز می‌گذرد. ولی کروموزم دومی معرف مسیری روی شبکه از راس v_2 تا v_1 است که به ترتیب از راس‌های v_6 و v_9 می‌گذرد.

۳-۷-۲ تعیین مطلوبیت

مطلوبیت یک کروموزم مقدار تابع هدف مساله به ازای جواب متناظر با این کروموزم می‌باشد. که در این مساله همان خروج از مرکز مسیر می‌باشد.

۳-۷-۳ جمعیت اولیه

اندازه جمعیت و جمعیت اولیه دو عامل مهم در همگرایی الگوریتم ژنتیک هستند. انتخاب جمعیت‌های بزرگ موجب کند شدن سرعت همگرایی شده و انتخاب جمعیت‌های کوچک باعث می‌شود که الگوریتم نتواند کل فضای جواب را برای یافتن جواب بهتر جستجو کند. همچنین انتخاب یک جمعیت اولیه مناسب موجب سرعت دادن به همگرایی الگوریتم می‌شود. ما اندازه جمعیت را n در نظر گرفتیم، که n تعداد رئوس شبکه می‌باشد.(با بررسی‌های انجام شده این اندازه جمعیت مناسب به نظر می‌آید) به عبارت دیگر برای تولید جمعیت اولیه باید n مسیر تولید کنیم. برای ساختن هر مسیر ما از یک راس شبکه شروع می‌کنیم و بقیه

راس‌ها را به صورت تصادفی از بین رئوس مجاور انتخاب می‌کنیم. این روش را ادامه می‌دهیم تا جایی که راسی وجود نداشته باشد که بتوانیم به مسیر اضافه کنیم.

البته در ساختن مسیر به جای انتخاب راس‌ها به صورت تصادفی می‌توان از رئوسی استفاده کرد که دارای درجه بیشتر باشند که این روش مورد آزمایش قرار گرفت ولی به نتیجه بهتری منجر نشد زیرا این روش انتخاب باعث می‌شود که دائما یک سری از راس‌ها انتخاب شوند، در نتیجه تنوع ژنتیکی جمعیت کم می‌شود. و یا می‌توان در ساختن قسمتی از مسیر، ازکوتاهترین مسیر بین دو راس نیز استفاده کرد. که ما این روش را مورد آزمون قرار دادیم ولی به نتایج بهتری منجر نشد.(مدت اجرای الگوریتم نیز بالا می‌رود.)

۳-۷-۴ انتخاب والدین

برای تولید عضو جدید احتیاج به دو والد داریم که ما برای انتخاب والدین از بین جمعیت اولیه به گونه زیر عمل می‌کنیم: بهترین عضو جمعیت را به عنوان یکی از والدین انتخاب می‌کنیم. و والد دیگر را بطور تصادفی از بین جمعیت انتخاب می‌کنیم .البته روش‌های دیگری مثل انتخاب والدین به طور تصادفی و انتخاب اعضا با مطلوبیت بالاتر به عنوان والدین مورد آزمون قرار گرفت که به نتایج بهتری منجر نشد. همچنین انتخاب اعضای که مسیر بزرگتر دارند نیز به عنوان والدین مورد آزمون قرار گرفت که باز هم به نتایج بهتری منجر نشد.

۳-۷-۵ تولید عضو جدید

در الگوریتم ژنتیک معمولا کروموزم‌های والدین به دو قسمت شکسته شده و سپس با هم ترکیب می‌شوند تا دو عضو جدید تولید شود. ما در اینجا از یک روش دیگر استفاده می‌کنیم. ابتدا همه ژن‌های مشترک بین والدین را انتخاب می‌کنیم. ممکن است این ژن‌های مشترک با هم تشکیل مسیر ندهند یعنی جواب موجه نباشد. برای رفع این مشکل در این رشته بین هر دو زیرمسیر یک مسیر تولید می‌کنیم و به این رشته اضافه می‌کنیم. زیر مسیر جدید می‌تواند کوتاهترین مسیر بین آخرین راس زیرمسیر اول و راس ابتدایی زیرمسیر دوم باشد. این زیرمسیر در صورتی به رشته اصلی اضافه می‌شود که در رشته اصلی دور ایجاد نشود

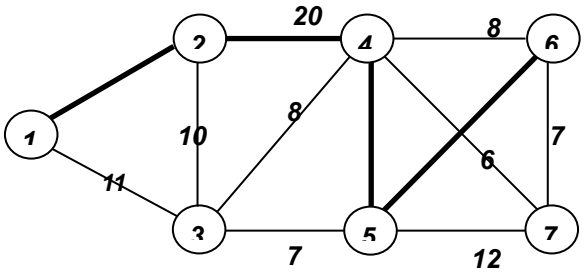
(به عبارت دیگر در صورت اضافه شدن زیرمسیر جدید هیچ راسی دو بار تکرار نشود). اگر در صورت اضافه شدن زیرمسیر به رشته اصلی دور ایجاد شد از کوتاهترین مسیر استفاده نمی‌کنیم.

در این صورت از زیرمسیری که بین این دو راس در یکی از والدین وجود دارد استفاده می کنیم این زیرمسیر جدید را به رشته اصلی اضافه می کنیم. برای انتخاب زیر مسیر، والدی را انتخاب می‌کنیم که تابع مطلوبیت بهتری داشته باشد. اگر در صورت اضافه کردن این زیرمسیر نیز دور ایجاد شد، در این صورت زیرمسیر دومی در رشته اصلی را حذف می‌کنیم. این روش را تا جای ادامه می‌دهیم که رشته اصلی به مسیر P_{new} تبدیل شود.

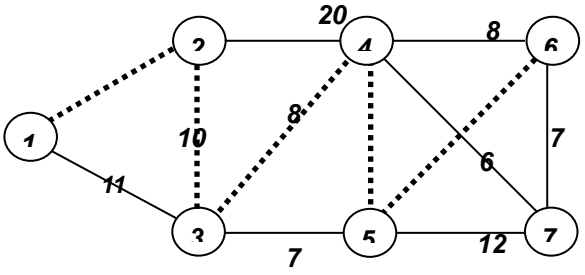
برای تولید زیرمسیر می‌توانیم به صورت تصادفی راس انتخاب کنیم تا زیر مسیر تولید شود اما تضمینی برای به دست آوردن زیر مسیر وجود ندارد. در نتیجه از این روش استفاده نمی‌کنیم.

۳-۷-۶ مثال

دو والد زیر را در نظر بگیرید. می‌خواهیم عضو جدیدی را تولید کنیم (در اشکال زیر بعضی یال‌ها بدون وزن هستند).



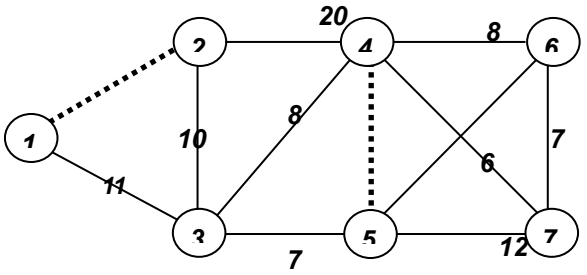
شکل۳-۳ والد اول(مسیرP_1 با خطوط مشکی مشخص شده است)



شکل ۳-۴ والد دوم (مسیرP_2 با خطوط نقطه چین مشخص شده است)

شکل ۳-۳ والد اول مسیرP_1= 1,2,4,5,6 را نشان می‌دهد که مقدار مطلوبیت آن هفت است یعنی $e(P_1)=7$. و شکل ۳-۴ والد دوم یعنی مسیر P_2= 1,2,3,4,5,6 را نشان می‌دهد که مقدار مطلوبیت این عضو برابر ۶ است، یعنی $e(P_2)=6$. حال برای تولید عضو جدید باید ژن‌های مشترک را مشخص کنیم.

S_{new})رشته اصلی=ژن‌های مشترک={1, 2, 4, 5}



شکل ۳-۵ ژن‌های مشترک دو والد(این ژن ها با نقطه چین مشخص شده است)

با توجه به شکل ۳-۵ بین راس دوم و راس چهارم مسیری وجود ندارد .در نتیجه باید بین این دو راس زیرمسیر ایجاد کنیم. زیرمسیرهای بین این دو راس شامل:

۱. مسیر P_1=2,4 که$L(P_1)$= 20

۲. مسیر P_2=2,3,4 که $L(P_2)$=18

۳. مسیر P_3=2,3,5,4 که $L(P_3)$=17

البته مسیرهای دیگری نیز بین این دو راس وجود دارد که به علت بالا بودن طول مسیر از ذکر آنها خودداری کردیم.

حال برای انتخاب زیرمسیر باید کوتاهترین مسیر را انتخاب کنیم. مسیر P_3=2,3,5,4 کوتاهترین مسیر است که باید این زیرمسیر را به رشته اصلی اضافه کنیم. در این صورت خواهیم داشت

S_{new}= {1, 2, 3, **5**, 4, **5**}

با توجه به S_{new} مشاهده می‌شود که راس ۵ دو بار تکرار شده است (یعنی دور ایجاد شده است) پس طبق روش گفته شده در بخش ۳-۷-۵ باید از زیرمسیری استفاده کرد که بین این دو راس در والد بهتر وجود دارد.

والد بهتر والد دوم است (مقدار مطلوبیت آن کمتر است). در والد دوم بین این دو راس مسیر P_2 وجود دارد. که باید این مسیر را به S_{new} اضافه کرد. در این صورت خواهیم داشت

S_{new}= {1, 2, 3, 4, 5}

با توجه به S_{new} ،مشاهده می‌شود که هیچ راسی دو بار تکرار نشد پس رشته اصلی تبدیل به P_{new} شده است. داریم

P_{new}= {1, 2, 3, 4, 5}

۳-۷-۷ گسترش مسیر

حال از انتهای مسیر، راس اضافه می‌کنیم تا مسیرP_{new} گسترش یابد. این راس‌ها از بین رئوسی که در والدین هستند انتخاب می‌شوند. بنابراین برای همه راس‌های مجاور راس آخر مسیر P_{new} تابع g_1 را محاسبه می‌کنیم. اگر v یکی از رئوس مجاور با راس آخر باشد. آنگاه خواهیم داشت که

$$g_1(v)=\alpha\frac{deg(v)}{d\ max}+(1-\alpha)\frac{f(v)}{F(P)} \qquad (۱-۳)$$

ما با امتحان کردن α های مختلف مثل0/25, 0/75, 0/5, α=0 به این نتیجه رسیدیم که به ازای α=0/5 جواب بهتری به دست می‌آید. در رابطه فوق نمادها به صورت زیرهستند

deg (v) درجه راس v است.

d max بیشترین درجه راس‌های شبکه است.

F (P) تابع مطلوبیت مسیر P_{new} است.

f (v) مقدار بهبود تابع هدف به ازای اضافه شدن راس v به مسیر P_{new} است.

چرا تابع g_1 به این گونه تعریف شده است؟ در انتخاب راس‌ها برای ادامه مسیر دو نکته بسیار مهم است اول اینکه راسی که انتخاب می‌شود بهترین بهبود را در تابع هدف ایجاد کند. ما برای لحاظ این نکته کسر $\frac{f(v)}{F(P)}$ را در نظر گرفتیم و دوم اینکه راسی که انتخاب می‌شود دارای درجه راس بیشتری باشد تا باعث شود که مسیر ادامه پیدا کند و تنوع ژنتیکی جمعیت بیشتر شود (در این مساله محدودیت هزینه را در نظر نگرفتیم). برای لحاظ کردن این نکته ما کسر $\frac{deg(v)}{d\ max}$ را در نظر گرفتیم. برای در نظر گرفتن مقدار تاثیرگذاری این دو

معیار از ضریب α استفاده کردیم. برای همه راس‌های مجاور با آخرین راس که در والدین وجود دارند، تابع g_1 را محاسبه می‌کنیم و با تابع احتمال زیر بهترین راس را انتخاب می‌کنیم.

$$p(v)=\frac{g_1(v)}{\sum_{v\in V'} g_1(v)} \qquad (۲-۳)$$

که 'V مجموعه راس‌های والدین هستند که می توان به مسیر P_{new} اضافه کرد. این روش اضافه کردن راس را ادامه می‌دهیم تا زمانی که راسی وجود نداشته باشد که بتوان به مسیر P_{new} اضافه کرد. و برای گسترش P_{new} همانند روش بالا از ابتدای مسیر نیز راس اضافه می‌کنیم.

چرا برای انتخاب راس‌ها از تابع احتمال p استفاده کردیم؟ما در انتخاب راس هم باید راس بهتر را انتخاب کنیم تا فرزندان بهتری تولید شوند و هم باید به نحوی راس‌ها را انتخاب کنیم که تنوع ژنتیکی جمعیت کم نشود. ما برای اینکه هر دو معیار رعایت شود از تابع احتمالp استفاده کردیم. البته روش‌های دیگری مانند انتخاب راس به صورت تصادفی و انتخاب راس با تابع هدف بهتر نیز مورد آزمون قرار گرفت که به نتیجه بهتری منجر نشد.

۳-۷-۸ جهش

انجام عمل جهش در الگوریتم ژنتیک موجب می شود که الگوریتم بتواند از گیرافتادن در نقاط بهینه موضعی فرار کند. برای اجرای عمل جهش، بعد از تولید عضو جدید از ابتدا و انتهای مسیر همانند روش بالا راس اضافه می‌کنیم. با این تفاوت که برای انتخاب راس‌ها برای اضافه کردن به مسیر به جای انتخاب راس‌ها از والدین از همه رئوس شبکه استفاده می‌کنیم.

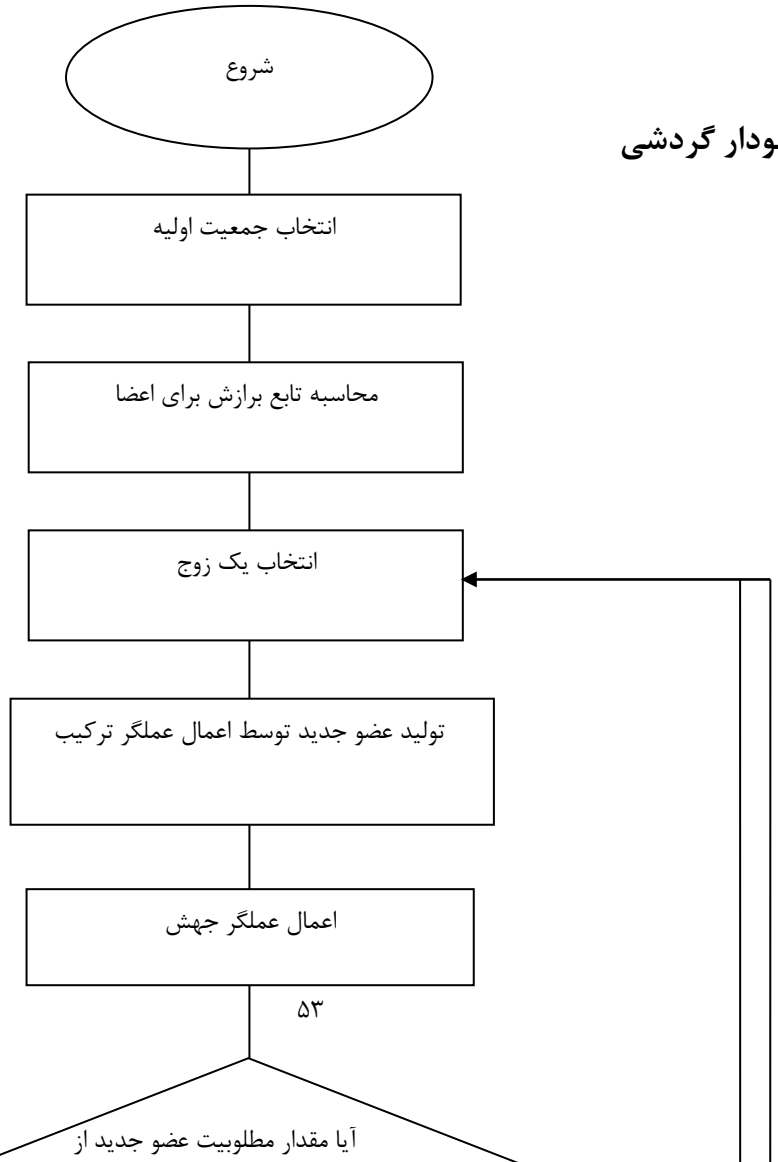
۳-۷-۹ جایگزینی

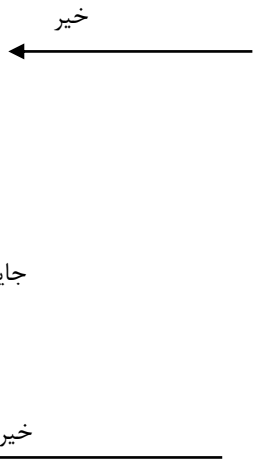
بعد از تولید یک عضو جدید و انجام عمل جهش باید این عضو را به جمعیت اضافه کنیم. اگر این عضو همانند هیچکدام از اعضای قبلی جمعیت نباشد و مقدار مطلوبیت آن از بدترین مقدار مطلوبیت جمعیت بهتر باشد آنگاه این عضو وارد جمعیت شده و عضوی از جمعیت که بدترین مقدار مطلوبیت را دارد از جمعیت خارج می‌شود.

۳-۷-۱۰ شرط توقف

شرط توقف می‌تواند رسیدن به حداکثر تعداد تکرار از ابتدای اجرای الگوریتم و یا بعد از آخرین بهبود باشد. همچنین یک حداکثر زمان اجرا را می‌توان به عنوان شرط توقف در نظر گرفت. ما رسیدن به n تکرار بعد از آخرین بهبود را به عنوان شرط پایانی در نظر گرفتیم.

۳-۷-۱۱ نمودار گردشی





مباحثی را که در بخش‌های قبل مطرح شد را می‌توان به صورت الگوریتم زیر بیان کرد.

۳-۸ الگوریتم ژنتیک برای مساله مسیر مرکزی(GAPCP[۸۹])

گام اول. جمعیت اولیه را با توجه به توضیحاتی که در بخش ۳-۷-۳ گفته شد، تولید کنید.

گام دوم. بهترین عضو و بدترین عضو جمعیت اولیه را پیدا کنید و مقدار تابع هدف هر کدام را (f_{best} ,) (f_{worst}) نیز محاسبه کنید (بهترین عضو، عضوی است که دارای کمترین تابع هدف با کمترین طول همراه با بیشترین تعداد راس باشد). و بدترین عضو، عضوی است که دارای بیشترین تابع هدف با بیشترین طول همراه با کمترین تعداد راس باشد.

قرار دهید :

$r := 0 , \ r_{best} := 0$

گام سوم. تا زمانیکه $r - r_{best} \le n$ مراحل زیر را انجام دهید:

۱) $r := r + 1$

[89] Genetic Algorithm for Path Center Problem

۲) دو عضو p_1,p_2 را از جمعیت انتخاب کنید به طوری که p_1 بهترین عضو جمعیت و p_2 به صورت تصادفی انتخاب شود.

۳) عضو جدید تولید کنید:

۳-۱ همه یال‌های مشترک دو عضو جدید را انتخاب کنید و S_{new} بنامید.

۳-۲ برای هر دو راس u و v که در S_{new} قرار دارند و به هم وصل نیستند مراحل زیر را انجام دهید:

۳-۲-۱ کوتاهترین مسیر بین دو راس u وv را پیدا کنید (P_{uv}). اگر با اضافه کردن P_{uv} به S_{new} دوری[۹۰] ایجاد نشد به (۳-۲) بروید.

(الف) از زیرمسیری که بین u وv در والد با تابع مطلوبیت بهتر وجود دارد استفاده کنید. و اگر در صورت اضافه شدن این زیر مسیر به S_{new} باز هم دوری ایجاد نشد به ۳-۲ بروید.

(ب) زیر مسیر دوم در S_{new} حذف کنید.

۳-۳ S_{new} را P_{new} بنامید.

۳-۴ x را راس پایانی P_{new} قرار دهید.

۳-۵ تا زمانیکه اضافه کردن راس دور ایجاد نکند مراحل زیر را انجام دهید:

۳-۵-۱ برای راس v ،که $v\in P_1 \cup P_2 \setminus P_{new}$ و به x وصل است $g_1(v)$ را محاسبه کنید و با استفاده از رابطه (۳-۲) بهترین راس را انتخاب کنید. و t بنامید.

۳-۵-۲ راس t را به P_{new} اضافه کنید.

۳-۵-۳ $x = t$

۴) انجام عمل جهش:

۴-۱ راس ابتدایی و انتهایی P_{new} را x_1,x_2 بنامید.

۴-۲ تا زمانیکه اضافه کردن راس دور ایجاد نکند مراحل زیر را انجام دهید:

۴-۲-۱ $x = x_1,x_2$.

---

[90] Cycle

۴-۲-۲ برای راس v، که $v \in G \setminus P_{new}$ و به x وصل باشد $g_1(v)$ را محاسبه کنید و با استفاده از تابع احتمال (۳-۲) راس t را انتخاب کنید.

۴-۲-۳ راس t را به P_{new} اضافه کنید.

۳-۴-۳ x = t

۵) تابع مطلوبیت را برای عضو جدید محاسبه کنید $F(P_{new})$. اگر عضو جدید در جمعیت وجود نداشت و $F(P_{new}) < f_{worst}$ مراحل زیر را انجام دهید:

۵-۱ عضو جدید را به جمعیت اضافه کنید و بدترین عضو را از جمعیت خارج کنید.

۵-۲ اگر $F(P_{new}) < f_{best}$ قرار دهید:

$f_{best} = F(P_{new})$, $r_{best} = r$.

گام چهارم. پایان

۳-۹ الگوریتم ژنتیک برای مساله مسیر مرکزی با طول محدودL (GAPCPBL[۹۱])

الگوریتم ژنتیک برای مساله مسیر مرکزی با طول محدودL همانند الگوریتم بخش ۳-۸ است با این تفاوت که در این الگوریتم از تابع g_2 به جای تابع g_1 استفاده می‌کنیم. تابع g_2 را به صورت زیر تعریف می‌کنیم:

اگر v یکی از رئوس مجاور با آخرین راس x باشد. آنگاه خواهیم داشت که

$$g_2(v)=\alpha\frac{f(v)}{F(P)}-(1-\alpha)\frac{L_{xv}}{L_2}. \qquad (۳-۳)$$

در تابع g_2 L_{xv} طول یال اضافه شده است و L_2 طول کل مسیر(P_{new}) است.

همانند تابع g_1 دو نکته در انتخاب راس‌ها بسیار مهم است اول اینکه راسی که انتخاب می‌شود بهترین بهبود را در تابع هدف ایجاد کند. ما برای لحاظ این نکته کسر $\frac{f(v)}{F(P)}$ را در نظر گرفتیم. در تابع g_1 محدودیت طولی نداشتیم و هدف این بود که مسیری پیدا کنیم که از همه راس‌های شبکه بگذرد. اما در تابع g_2 ما محدودیت طول داریم بنابراین کسر L_{xv}/L_2 را در نظر گرفتیم. و این کسر را منفی در نظر گرفتیم زیرا هر

[91] Genetic Algorithm for Path Center Problem with Bounded Length

چه طول کمتر باشد برای ما بهتر است. ما از کسر $L_{xv}/\max L$ ($\max L=\max w'_{ij}$) به جای کسر L_{xv}/L_2 نیز استفاده کردیم که به نتایج تقریبا مشابه رسیدیم. به ازای تابع g_3 نیز الگوریتم را تست کردیم که به نتایج بهتری منجر نشد.

$$g_3=\frac{f(v)}{F(P)}+\frac{\deg(v)}{d\max}-\frac{L_{xv}}{L_2}. \qquad (۴-۳)$$

برای در نظر گرفتن مقدار تاثیرگذاری این دو معیار درتابع g_2 از ضریب α استفاده کردیم. تابع g_2 را برای α های مختلف مثل 0/4,0/6,0/5,0/25,0/75 تست کردیم که به ازای α =0/5 جواب بهتری به دست آمد.

با توجه به توضیحات بالا الگوریتم ژنتیک برای مساله مسیر مرکزی با طول محدود به صورت زیر خواهد بود:

گام اول. برای ساختن جمعیت اولیه همانند بخش ۳-۷-۳ عمل می کنیم با این تفاوت که برای ساختن هر مسیر و اضافه کردن راس تا جایی می‌توانیم راس اضافه کنیم که طول مسیر از L بیشتر نشود.

گام دوم. بهترین عضو و بدترین عضو جمعیت اولیه را پیدا کنید و مقدار تابع هدف بهترین عضو f_{best} و مقدار تابع هدف بدترین عضو f_{worst} را محاسبه کنید. و قرار دهید :

$r:=0$, $r_{best}:=0$

گام سوم. تا زمانیکه $r-r_{best} \le n$ مراحل زیر را انجام دهید:

۱) $r:=r+1$

۲) دو عضو p_1,p_2 را از جمعیت انتخاب کنید به طوری که p_1 بهترین عضو جمعیت و p_2 به صورت تصادفی انتخاب شود.

۳) عضو جدید تولید کنید:

۳-۱ همه یال‌های مشترک دو عضو جدید را انتخاب کنید و S_{new} بنامید

۳-۲ برای هر دو راس u و v که در S_{new} قرار دارند و به هم وصل نیستند مراحل زیر را انجام دهید:

۳-۲-۱ کوتاهترین مسیر بین دو راس u وv را پیدا کنید (P_{uv}). اگر با اضافه کردن P_{uv} به S_{new} دوری ایجاد نشد به ۳-۲ بروید.

(الف) از زیرمسیری که بین u وv در والد با تابع مطلوبیت بهتر وجود دارد استفاده کنید. و اگر در صورت اضافه شدن این زیرمسیر به S_{new} باز هم دور ایجاد نشد به ۳-۲ بروید.

(ب) زیرمسیر دوم در S_{new} راحذف کنید.

۳-۳ S_{new} را P_{new} بنامید.

۳-۴ اگر طول P_{new} بیشتر از L بود کارهای زیر را انجام دهید:

۳-۴-۱ راس آخر مسیر P_{new} را حذف کنید

۳-۴-۲ بقیه مسیر را P_{new} بنامید.

۳-۵ x را راس پایانی P_{new} قرار دهید.

۳-۶ تا زمانیکه اضافه کردن راس دور ایجاد نکند و طول P_{new} بیشتر ازL نشود مراحل زیر را انجام دهید:

۳-۶-۱ برای راس v،که $v \in P_1 \cup P_2 \backslash P_{new}$ و به x وصل است $g_2(v)$ را محاسبه کنید

و با استفاده از تابع احتمال ۳-۵ راس t را انتخاب کنید.

$$p(v)=\frac{g_2(v)}{\sum_{v\in V'} g_2(v)} \qquad (۵-۳)$$

که 'V مجموعه راس‌های والدین هستند که می توان به مسیر P_{new} اضافه کرد.

۳-۶-۲ راس t را به P_{new} اضافه کنید.

۳-۶-۳ $x = t$

۴) انجام عمل جهش:

۴-۱ راس ابتدایی و انتهایی P_{new} را x_1,x_2 بنامید.

۴-۲ تا زمانیکه اضافه کردن راس دور ایجاد نکند و طول مسیر از L بیشتر نشود مراحل زیر را انجام دهید:

۴-۲-۱ $x = x_1,x_2$

۴-۲-۲ برای راس v، که $v \in G \backslash P_{new}$ و به x وصل است $g_2(v)$ را محاسبه کنید و با استفاده از تابع احتمال ۳-۵ راس t را انتخاب کنید.

۴-۲-۳ راس t را به P_{new} اضافه کنید.

$x = t$ ۳-۴-۳

۵) تابع مطلوبیت را برای عضو جدید محاسبه کنید ($F(P_{new})$. اگر عضو جدید در جمعیت وجود نداشت و $F(P_{new}) < f_{worst}$ مراحل زیر را انجام دهید:

۵-۱ عضو جدید را به جمعیت اضافه کنید و بدترین عضو را از جمعیت خارج کنید.

۵-۲ اگر $F(P_{new}) < f_{best}$ قرار دهید:

$f_{best} = F(P_{new})$, $r_{best} = r$.

گام چهارم. پایان

۳-۱۰ نتایج

ما الگوریتم ژنتیک برای مساله مسیر مرکزی و مساله مسیر مرکزی با طول محدود را توسط مسائل نمونه‌ای [60] تست کردیم. هر کدام از مسائل را پنج بار توسط الگوریتم اجرا کردیم. برای مساله مسیر مرکزی بهترین جواب را به عنوان جواب اصلی انتخاب کردیم. و برای مساله مسیر مرکزی با طول محدود میانه جواب‌ها را به عنوان جواب اصلی انتخاب کردیم. در این مسائل وزن همه راس‌ها برابر یک است ولی وزن یال‌ها برابر یک نیست.

برای تست الگوریتم مسیر مرکزی با طول محدود از دو طول متفاوت استفاده کردیم. در شبکه مربوط به مسائل نمونه‌ای حداقل یک سیکل مشاهده می‌شود. ما از طول این سیکل برای تست کردن الگوریتم استفاده کردیم. دومین طولی که در نظر گرفتیم، طول ۲۵۰۰ است که برای همه مسائل مشترک در نظر گرفتیم.

در جدول ۳-۱ ستون.چهارم، پنجم و ستون ششم تابع هدف، تعداد راس و طول جواب اولیه است که الگوریتم با آن آغاز می‌شود. ستون هفتم، هشتم و ستون نهم تابع هدف ، تعداد راس و طول بهترین جوابی است که با استفاده از الگوریتم ژنتیک به دست آوردیم. و ستون دهم تابع هدف بدترین جواب در جمعیت نهایی مساله است. ستون یازدهم تعداد تکرار بعد از آخرین بهبود است. و ستون آخر نشان دهنده زمان اجرا به ازای هر تکرار می‌باشد.

در جدول ۳-۲ ستون چهارم طول سیکل نمونه ای برای هر مساله است. ستون پنجم، ششم و ستون هفتم تابع هدف، تعداد راس و طول جواب اولیه است که الگوریتم با آن آغاز می‌شود. ستون هشتم، نهم و ستون دهم تابع هدف، تعداد راس و طول جواب نهایی (بهترین جواب) الگوریتم است. و ستون یازدهم تابع هدف بدترین جواب در جمعیت نهایی مساله است.

با توجه به نتایج به دست آمده در جدول ۳-۲ تقریبا برای همه مسائل جواب نهایی نسبت به جواب اولیه بهبود داشته است. فقط در مساله سوم با طول سیکل نمونه و مساله دوم با طول ۲۵۰۰ بهبودی در تابع هدف جواب نهایی نسبت به جواب اولیه مشاهده نشده است. اما تعداد راس ها نسبت به تعداد راس جواب اولیه بیشتر شده است. یعنی تعداد بیشتری از راس ها تحت پوشش قرار گرفتند.

جدول ۳-۱ نتایج الگوریتم ژنتیک برای مساله مسیر مرکزی

Test #	n	Edge NO.	Objective function	*Initial* Vertex NO	Length P	Objective function	*Best* Vertex NO	LengthP	*Worst* Objective function	Last improv.iter	Cpu / Iter(sec)
1	100	200	82	57	3208	60	54	3515	98	138	0.104
2	100	200	86	65	3532	68	71	4742	98	179	0.123
3	100	200	72	75	3864	72	75	2982	72	176	0.125
4	100	200	91	65	3630	91	65	3630	116	101	0.109
5	100	200	77	60	3168	77	60	3168	92	101	0.1
6	200	800	24	170	8571	24	170	8571	68	201	0.388
7	200	800	38	134	6711	38	168	7350	55	209	0.259
8	200	800	46	140	7111	38	125	4948	60	473	0.281
9	200	800	46	149	7506	38	143	5548	71	278	0.396
10	200	800	34	146	6835	34	154	8546	59	317	0.376

جدول ۳-۲ نتایج الگوریتم ژنتیک برای مساله مسیر مرکزی با طول محدود

Test #	n	Edge NO.	Length	Initial Objective function	Initial Vertex NO	Initial LengthP	best Objective function	best Vertex NO	best LengthP	worst Objective function	Last improv.iter	Cpu / Iter(sec)
1	100	200	5264	82	57	3208	76	77	3393	76	181	0.106
2	100	200	5313	86	65	3532	57	78	4329	94	145	0.135
3	100	200	5235	69	80	3982	69	82	2065	69	181	0.118
4	100	200	5294	91	65	3630	81	70	3838	81	222	0.116
5	100	200	5573	83	55	2870	73	67	3698	87	148	0.154
6	200	800	9854	24	170	8571	23	171	5705	43	397	0.418
7	200	800	10208	38	134	6711	23	191	6078	23	964	0.441
8	200	800	10062	31	156	3129	24	177	4595	34	501	0.167
9	200	800	10403	46	149	7506	17	187	9039	38	500	1.17
10	200	800	9882	34	146	6835	22	180	8503	22	837	0.394

Test #	n	Edge NO.	Length	Initial Objective function	Initial Vertex NO	Initial LengthP	best Objective function	best Vertex NO	best LengthP	worst Objective function	Last improv.iter	Cpu / Iter(sec)
1	100	200	2500	88	47	2482	79	50	2466	90	237	0.084
2	100	200	2500	96	52	2468	96	54	2493	96	366	0.094
3	100	200	2500	92	42	2435	87	46	2489	87	239	0.07
4	100	200	2500	98	45	2499	96	46	2455	106	237	0.487
5	100	200	2500	86	45	2347	85	49	2463	85	437	0.066
6	200	800	2500	66	51	2499	61	51	2481	79	298	0.129
7	200	800	2500	63	53	2486	57	62	2498	63	917	0.137
8	200	800	2500	72	55	2491	71	60	2450	88	238	0.201
9	200	800	2500	69	59	2499	63	76	2491	63	761	0.327
10	200	800	2500	56	60	2497	51	62	2487	51	594	0.139

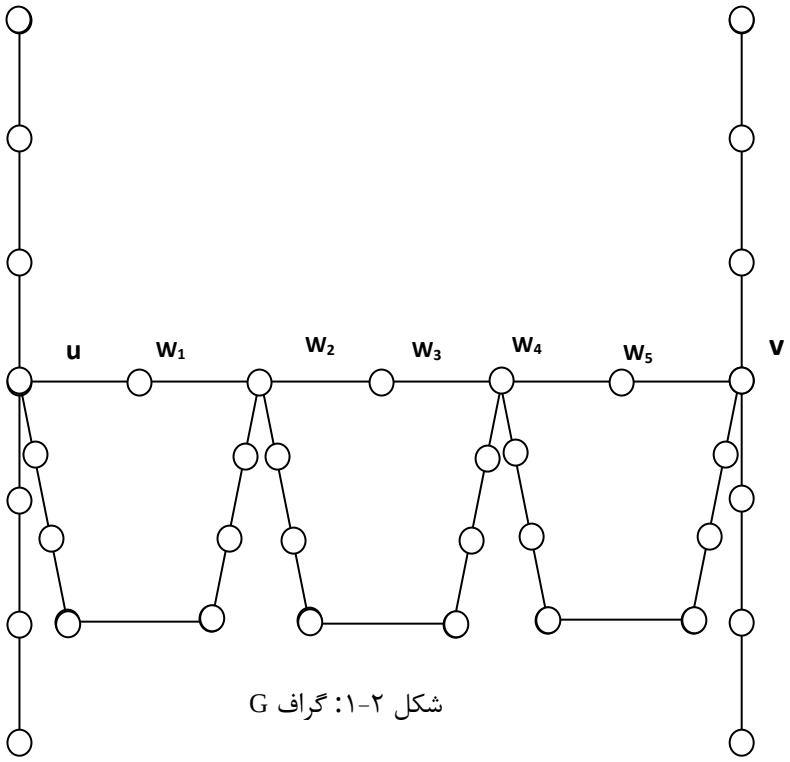
فصل دوم

مکان‌یابی مسیر مرکزی روی درخت

در این فصل به بررسی مساله مکان‌یابی مسیر مرکزی روی درخت می‌پردازیم و الگوریتمی را که اسلیتر برای حل این مساله ارائه داده است، بیان می‌کنیم. مطالب ارائه شده در این فصل برگرفته از مرجع [۲۵] می‌باشد.

۲-۱ مکان‌یابی مسیر مرکزی روی درخت

مفهوم مسیر مرکزی اولین بار توسط اسلیتر و هدتنیمی در سال ۱۹۷۷م به طور جداگانه مطرح شد و سرانجام هدتنیمی [۳۵] و اسلیتر [۲۵] الگوریتم‌های خطی برای این مساله بر روی درخت ارائه کردند. اسلیتر مسیر مرکزی را به گونه زیر تعریف کرده است[۲۵]: مسیر مرکزی روی گراف G مسیری است که کمترین خروج از مرکز را بین بقیه مسیرها داشته باشد و اگر دو مسیر خروج از مرکز یکسان داشتند، مسیری که کمترین طول را داشته باشد مسیر مرکزی خواهد شد. تعریف هدتنیمی همان تعریف اسلیتر است با این تفاوت که شرط کمترین طول را برای مسیر مرکزی در نظر نگرفته است.



شکل ۲-۱: گراف G

در شکل ۲-۱ [۲۵] طبق تعریف هدتنیمی بین u وv هشت مسیر وجود داردکه خروج از مرکز هر هشت مسیر ۳ است. و چون مسیر دیگری یافت نمی‌شود که خروج از مرکز آن کمتر از ۳ باشد پس این هشت مسیر، مسیر مرکزی هستند. اما طبق تعریف اسلیتر فقط مسیر$P=u,w_1,w_2,w_3,w_4,w_5,v$ مسیر مرکزی است.

۲-۲ الگوریتم مسیر مرکزی روی درخت

۲-۲-۱ قضایا

۲-۲-۱ -۱: قضیه جردن [۴۵]

مرکز هر درخت T یا شامل یک راس و یا شامل دو راس مجاور است.

اثبات

از استقرا روی تعداد راس‌ها استفاده می‌کنیم. فرض کنید n تعداد رئوس درخت T باشد. به ازای $n=1,2$ قضیه بدیهی است. برای $n>2$ فرض کنیم T یک درخت n– راسی دلخواه باشد. در درخت T تمام برگ‌ها را حذف می‌کنیم گراف حاصل را که یک درخت می‌باشد، T' می‌نامیم. برای هر راس u در درخت، هر راس در فاصله ماکسیمم از u یک برگ است. (زیرا مسیر میان راس v و دورترین راس x را در نظر می‌گیریم. اگر x یک برگ نباشد، آن گاه دارای همسایه دیگری است که از راس v دورتر است) چون همه برگ‌ها حذف شده‌اند و در مسیری که بین دو راس وجود دارد هیچ برگی وجود ندارد. در نتیجه برای هر $u\in V(T')$ داریم

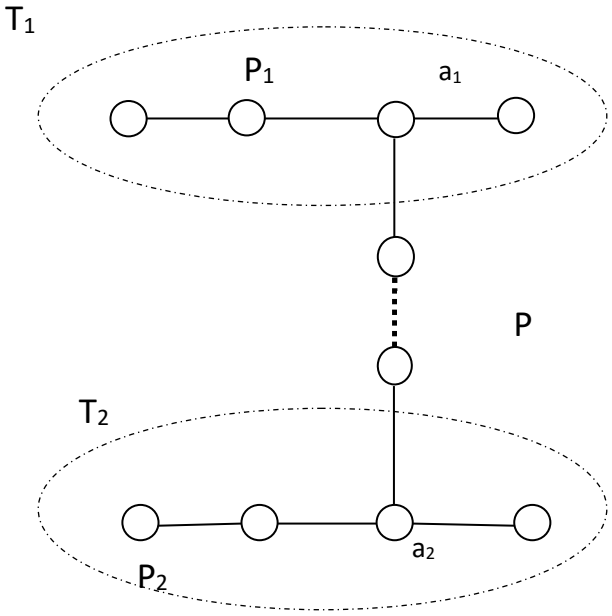
$$e_{T'}(u)=e_T(u)-1.$$

و همچنین خروج از مرکز یک برگ در T بزرگتر از خروج از مرکز همسایه‌اش در T است از این رو راس‌های که $e_T(u)$ را کمینه می‌کنند، همان راس‌هایی هستند که $e_T(u)$ را کمینه می‌کنند و بنابر فرض استقرا برای T' آنها متشکل از یک راس منفرد یا دو راس مجاور می‌باشند. پس مرکز درخت T نیز متشکل از یک راس یا دو راس مجاور است. □

قضیه۲-۲-۱-۲ : مسیرمرکزی درخت T منحصربه فرد است [۳۵].

اثبات. از برهان خلف استفاده می‌کنیم. فرض کنیم P_1 و P_2 دو مسیر مرکزی درخت T باشند. این دو مسیر یا راس مشترک با هم دارند یا هیچ راس مشترکی با هم ندارند.

الف- فرض کنید دو مسیر P_1 و P_2 هیچ راس مشترکی با هم نداشته باشند. (طبق شکل ۲-۲)



شکل ۲-۲: مولفه هایی که از حذف یال های مسیر P به دست آمده است.

از مسیر P_1 به P_2 یک مسیر مثل P با راس ابتدایی و انتهایی a_1 وa_2 در نظر می‌گیریم. و فرض کنید T^*جنگلی باشد که از حذف همه یال‌های مسیر P از T به دست آمده است.T_i i=1,2مولفه‌هایی از T^* هستند که شامل P_i هستند.

راس$x_i \in V(T_i)$ را به گونه‌ای در نظر می‌گیریم که

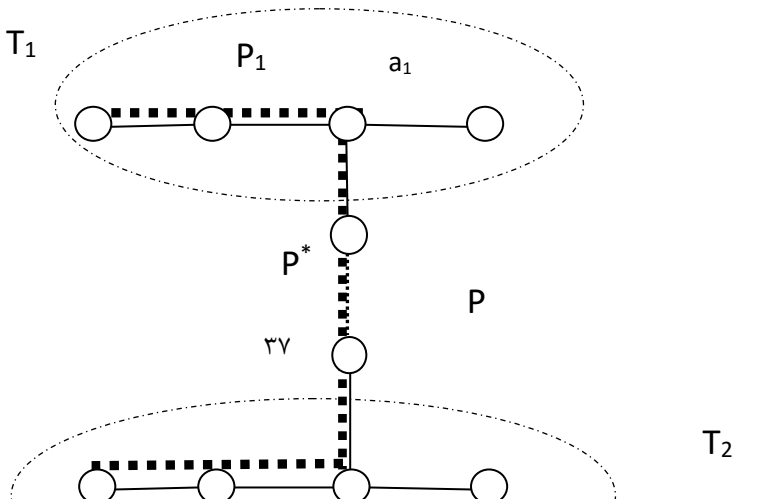
$$2, x \in V(T_i)\}. \; 1, = d(x , a_i)\}, \; i\, d\,(x_i , a_i)=\max\{$$

حال مسیر P^* را از x_1 به x_2 در نظر می‌گیریم. و ادعا می‌کنیم که

$$e(P^*) < e(P_1) = e(P_2). \qquad (۱-۲)$$

اگر$y \in V(T_1)$ طبق شکل ۲-۳ و با توجه به تعریف x_1,x_2 خواهیم داشت که

$$d(y,P^*) \leq d(y,a_1). \qquad (۲-۲)$$



شکل ۲-۳ : مولفه ای که از حذف یال های مسیر P به دست آمده همراه با مسیر P^*

پس در حالت کلی خواهیم داشت

$$d(y,P^*) \le d(y,a_1) \le d(x_1,a_1) < d(x_1,P_2) \le e(P_2). \quad (۳-۲)$$

بطور مشابه اگر $y \in V(T_2)$ باشد آنگاه طبق شکل ۲-۳ و با توجه به تعریف x_1,x_2 خواهیم داشت

$$d(y,P^*) \le d(y,a_2) \le d(x_2,a_2) < d(x_2,P_1) \le e(P_1). \quad (۴-۲)$$

تا اینجا اثبات کردیم که فاصله همه راس‌های T_1,T_2 تا مسیر P^* کمتر از $e(P_1),e(P_2)$ است.

راسی مثل z را از مسیر P در نظر می‌گیریم به طوری که در T_1,T_2 نباشد، آنگاه خواهیم داشت که

$$d(z,P^*) < d(z,P_i) \le e(P_i) \ , \ i=1,2 \quad (۵-۲)$$

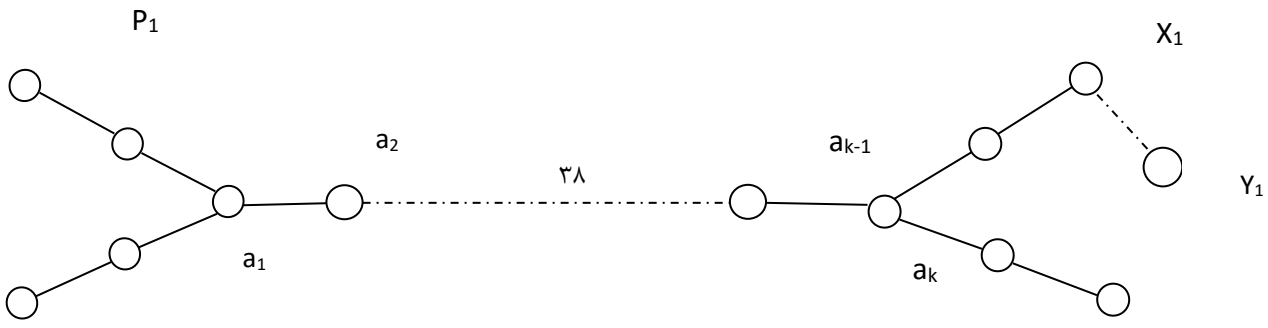
سر انجام برای همه راس‌های $z \in P$ داریم $d(z,P^*)=0$

بنابراین خواهیم داشت که

$$e(P^*) < e(P_1) = e(P_2) \quad (۶-۲)$$

پس طبق رابطه ۲-۶ مسیری مثل P^* پیدا کردیم به گونه ای که خروج از مرکز آن کمتر از $e(P_1),e(P_2)$ است. و این خلاف فرض است. پس فرض خلف باطل است و $P_1=P_2$.

ب)حالتی را در نظر می‌گیریم که دو مسیر P_1,P_2 در راس‌های $a_1,\dots a_k$ مشترک باشند (طبق شکل ۲-۴) و به دلیل اینکه شبکه مورد نظر درخت است و درخت فاقد دور است پس رئوس مشترک بین دو مسیر P_1,P_2 یعنی رئوس $a_1\dots,a_k$ خود یک مسیر در T تشکیل می‌دهند.



شکل۲-۴: دومسیرP_1,P_2 با راس های مشترک

فرض می‌کنیم که x_1 راس پایانی مسیر P_1 باشد به قسمی که $x1\notin\{a_1,a_2..,a_k\}$.چون مسیر P_1 کمترین خروج از مرکز را دارد، پس راسی مثل $y_1\in V(T)$ وجود دارد به قسمی که کوتاهترین مسیر بین y_1 و مسیر P_1 از x_1 می‌گذرد و خواهیم داشت

$$d(y_1,x_1) = e(P_1). \quad (۷-۲)$$

بنابراین با توجه به رابطه فوق داریم

$$e(P_2) \geq d(y_1,P_2) > d(y_1,x_1) = e(p_1). \quad (۸-۲)$$

که متناقض با فرض است.

بنابراین راس پایانی مسیر P_1 متعلق به مسیر $a_1,a_2\ldots,a_k$ است. برای مسیر P_2 نیز به همین گونه اثبات می‌شود که راس پایانی P_2 متعلق به مسیر $a_1,a_2,\ldots,a_k$ است. بنابراین داریم $P_1=P_2$ □.

قضیه۲-۲-۱-۳ مسیر مرکزی هر درخت شامل مرکز درخت نیز می‌شود [۳۵].

اثبات. از برهان خلف استفاده می‌کنیم. فرض کنید u مرکز درخت و P مسیر مرکزی درخت باشد. فرض می‌کنیم که راس u جزء مسیر مرکزی نباشد، طبق خاصیت درخت که بین هر دو راس یک درخت یک مسیر منحصر به فرد وجود دارد، پس بین u و P مسیری مثل q وجود دارد. با حذف همه یال‌های مسیر q از درخت T جنگل T^* را تشکیل می‌دهیم. در این جنگل مولفه‌ای که شامل u باشد را T_u می‌نامیم. به دلیل اینکه u مرکز درخت است، پس راسی مثل x در T_u وجود دارد به قسمی که

$$d(x,u) \geq e(u) -1. \quad (۹-۲)$$

$$e(P) \geq d(x,P) > d(x,u) \geq e(u)-1. \quad (۱۰-۲)$$

با توجه به رابطه فوق داریم

$$e(P) \geq e(u). \qquad (۱۱-۲)$$

اما چون P مسیر مرکزی است داریم $e(P) \leq e(u)$ که مخالف با نامساوی (۲-۱۱) است. پس فرض خلف باطل است و مسیر مرکزی درخت شامل مرکز درخت نیز می‌شود. □

۲-۲-۲ نمادگذاری

فرض کنید $P=v_1,v_2,\ldots,v_k$ یک مسیر در درخت T باشد، آنگاه:

الف) برای $i=1,k$ درخت‌های T_1 و T_k به صورت زیر تعریف می‌شوند (که شامل راس v_1,v_k نیز می‌شود) و داریم $e_{12}= (v_1,v_2)$

$$T_1=T_k=T-(v_1,v_2)-(v_{k-1},v_k). \qquad (۱۲-۲)$$

ب) برای $i=2,3,.,K-1$ درخت T_i به صورت زیر تعریف می‌شود (که شامل راس v_i نیز می‌شود)

$$T_i=T-(v_{i-1},v_i)-(v_i,v_{i+1}). \qquad (۱۳-۲)$$

ج) برای $i=1,2,.,K$ می‌توان تعریف کرد

$$e_p(v_i)=e(v_i ,T_i). \qquad (۱۴-۲)$$

۲-۲-۳ الگوریتم مسیر مرکزی روی درخت PCT [۹۲] [۲۵]

در این الگوریتم $w_i=1$ و $w'_{ij}=1$ فرض شده است. و $|V(T)| \geq 3$

گام صفر. مرکز درخت را مشخص کنید

گام اول. اگر مرکزدرخت Tشامل دو راس مجاور$\{u,v\}$ باشد، قراردهید

$$k=2 \text{ ، } P=u,v$$

و به گام سوم بروید.

گام دوم. اگر مرکز درخت فقط شامل راس $\{u\}$ باشد، در این صورت اگر در سه مولفه از T-u راسی مانند v وجود داشته باشد به طوری که دارای خاصیت زیر باشد

$$d(v,u)=e(u,T).$$

[92] Path center of a tree

(یعنی راسی مثل v وجود داشته باشد که فاصله‌اش تا مرکز برابر خروج از مرکز راس مرکزی باشد.)

آنگاه قرار دهید: P= u و به گام پنجم بروید.

و اگر دقیقا دو مولفه از T-u شامل راسی باشند که خاصیت فوق را داشته باشد، در این صورت می‌توان مسیر مرکزی را از مرکز درخت گسترش داد و راس‌هایی که با u در این دو مولفه مجاور هستند جزء مسیر مرکزی قرار دهید. K=2 و به گام سوم بروید.

گام سوم. (بررسی شروط توقف الگوریتم) اگر راسی مثل i وجود داشته باشد که

$$e_p(v_i)=e_p(v_1)=e_p(v_k) \ , \ 2\leq i\leq k-1 \qquad (۱۵-۲)$$

دراین صورت به گام پنجم بروید.

و یا اگر در هر کدام از گراف‌های T_1-v_1 و T_k-v_k بتوان دو مولفه یا بیشتر پیدا کرد به طوری که این مولفه‌ها شامل راسی باشند که در خاصیت زیر صدق کنند.

$$e_p(v_h)=d(v,v_h):v\in V(T). \qquad (۱۶-۲)$$

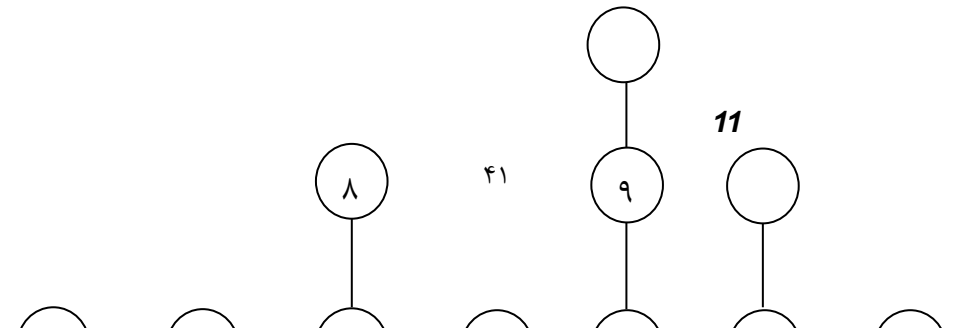
در این صورت نمی‌توان مسیر مرکزی را گسترش داد و به گام پنجم بروید.

گام چهارم. اگر در هر کدام از گراف T_1-v_1 و T_k-v_k دقیقا یک مولفه شامل راسی مثل v باشد به طوری که در رابطه فوق صدق کند در این صورت راس‌هایی که به v_1و v_h وصل هستند را جزء مسیر مرکزی قرار دهید و به k دو واحد اضافه کنید و به گام سوم بروید. (برای توسعه مسیر مرکزی حتما از هر طرف مسیر مرکزی اولیه باید فقط یک مولفه واجد شرایط وجود داشته باشد تا بتوانیم مسیر مرکزی را گسترش دهیم در غیر این صورت با یکتا بودن مسیر مرکزی متناقض خواهد شد).

گام پنجم. P مسیرمرکزی درخت T است.

۲-۳ مثال[۲۵]

درخت T در شکل ۲-۵ را در نظر بگیرید. وزن همه یال‌ها ۱ فرض شده است.



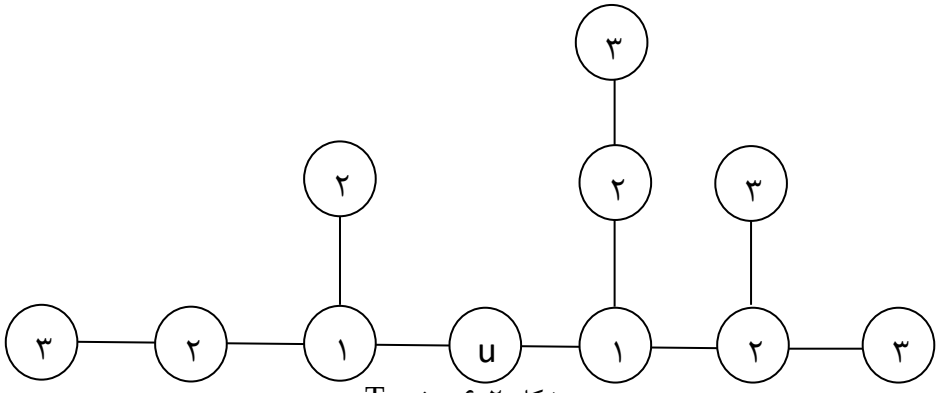
شکل ۲-۵ درخت T

گام صفر. برای اینکه مرکز درخت را مشخص کنیم باید خروج از مرکز همه راس‌ها را حساب کنیم.

e(1,G)=6	e(5,G)=4	e(9,G)=5
e(2,G)=5	e(6,G)=5	e(10,G)=6
e(3,G)=4	e(7,G)=6	e(11,G)=6
e(4,G)=3	e(8,G)=5	

$\text{Min}\{e_i\}=3$ Center vertex=4.

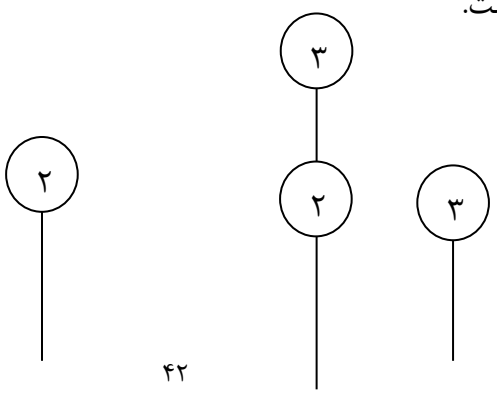
به دلیل اینکه کمترین خروج از مرکز عدد ۳ است پس مرکز گراف راس ۴ خواهد بود. در شکل ۲-۶ عدد داخل هر راس نشان‌دهنده فاصله هر راس تا مرکز گراف (u) است.



شکل ۲-۶ درخت T

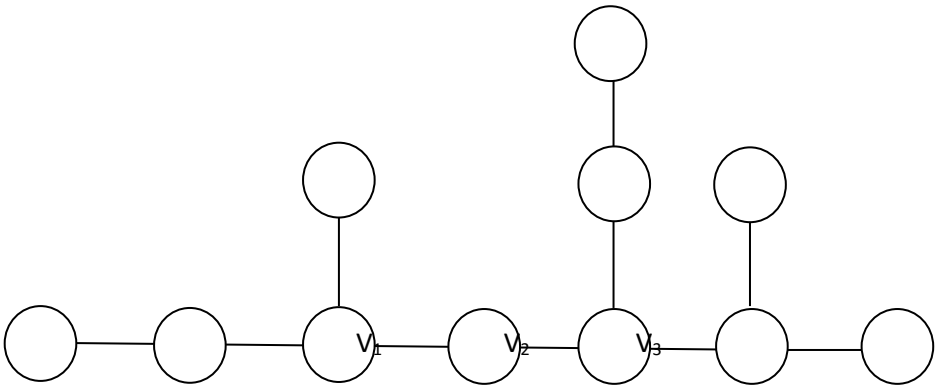
حال طبق الگوریتم چون مرکز گراف یک راس است پس وارد گام دوم می‌شویم.

گام دوم. در این گام باید T-u را بررسی کنیم. در شکل ۲-۷ عدد داخل هر راس نشان دهنده فاصله هر راس تا مرکز درخت T است.



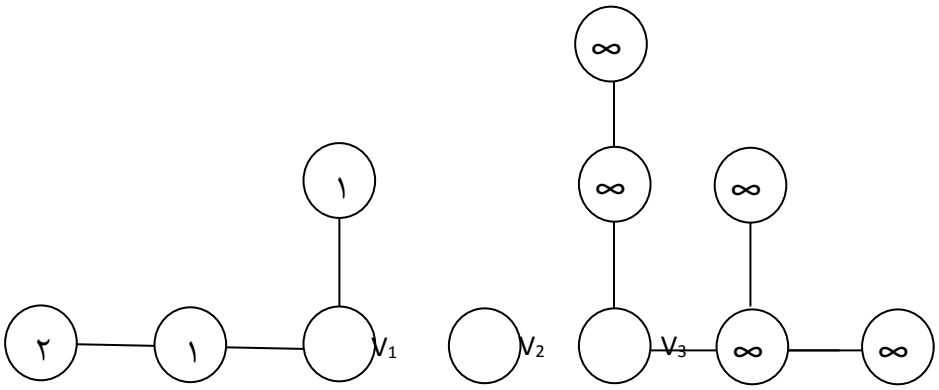
شکل ۲-۷ گراف T-u

چون $e(u,T)=3$ اســـت، پس باید در T-u به دنبال راس‌هایی در مولفه‌های مجزا باشـــیم به طوری که فاصـــله آن راس تا مرکز برابر ۳ باشـــد. چون تنها دو مولفه وجود دارد که دارای خاصـــیت فوق اســـت، پس می‌توان مسیر مرکزی را گسترش داد و قرار دهید $k=3$ و به گام سوم بروید.



شکل ۲-۸ مسیر مرکزی v_1,v_2,v_3

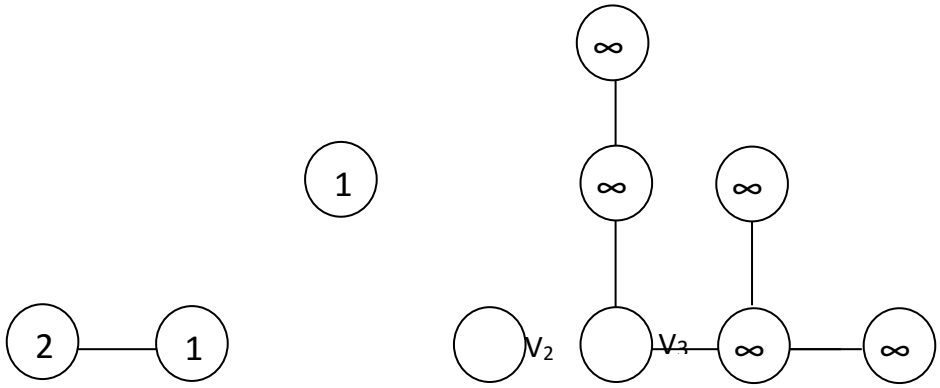
گام سوم. در گام سوم باید T_1-v_1 وT_3-v_3 را بررسی کنیم.



شکل۲-۹ گراف $T_1=T-(v_1,v_2)-(v_2,v_3)$

در شکل ۲-۹ و ۲-۱۰ عدد داخل هر راس نشان دهنده فاصله هر راس تا راس V_1 در گراف T_1 است. با توجه به شکل ۲-۹ داریم

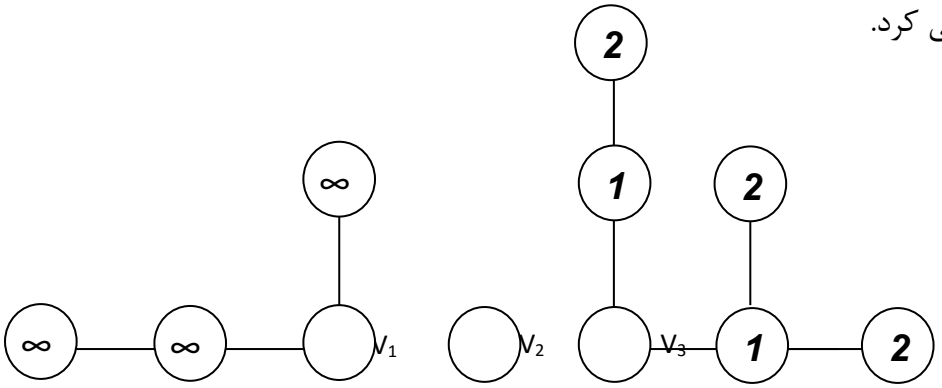
$$e_p(v_1)=e(v_1,T_1)=2.$$



شکل ۲-۱۰ گراف T_1 -v_1

با توجه به مقدار $e_p(v_1)$ در گراف T_1-v_1 به دنبال مولفه‌ای هستیم که شامل راسی باشد که فاصله‌اش تا راس v_1 در گراف T_1 برابر مقدار $e_p(v_1)$ باشد. با توجه به شکل ۲-۱۰ فقط یک مولفه وجود دارد که دارای خاصیت فوق است. پس از راس v_1 می توان مسیر مرکزی را گسترش داد.

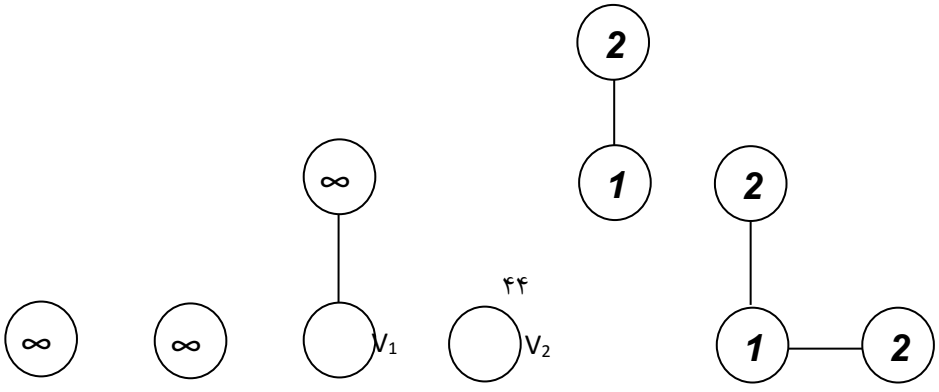
حال بررسی می‌کنیم که آیا می‌توان از راس v_3 نیز مسیر مرکزی را گسترش داد. برای اینکار باید گراف T_3-v_3 را بررسی کرد.



شکل ۲-۱۱ گراف $T_3 = T-(v_1,v_2)-(v_2,v_3)$

در شکل ۲-۱۱ عدد داخل هر راس نشان دهنده فاصله هر راس تا راس v_3 در گراف T_3 است. و با توجه به شکل ۲-۱۱ داریم

$e_p(v_3)=e(v_3,T_3)=2.$

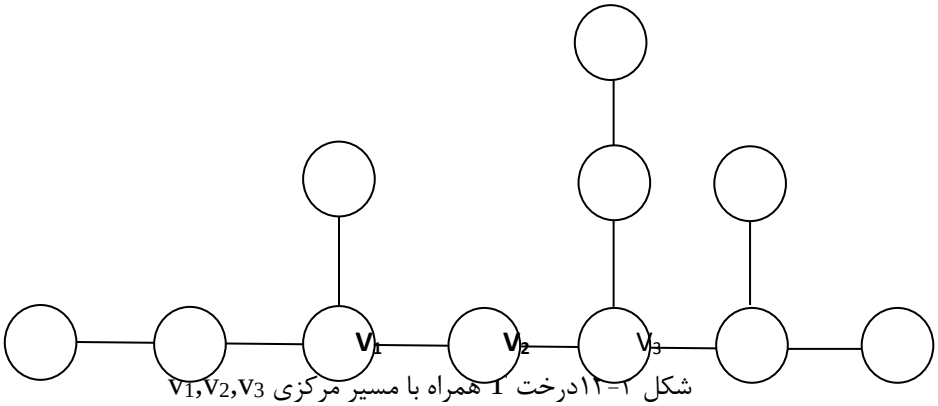


شکل ۲-۱۲گراف T_3-v_3

در گراف T_3-v_3 به دنبال مولفه‌ای هستیم که شامل راسی باشد که فاصله‌اش تا راس v_3 در گراف T_3 برابر مقدار $e_p(v_3)$ باشد. چون در گراف T_3-v_3 دو مولفه وجود دارد که دارای خاصیت فوق است پس طبق گام سوم نمی‌توان مسیر مرکزی را گسترش داد. پس به گام پنجم بروید.

گام پنجم. P مسیر مرکزی درخت T است.

$P = v_1,v_2,v_3$



شکل ۱-۱۱درخت T همراه با مسیر مرکزی v_1,v_2,v_3

فصل سوم

الگوریتم ژنتیک

برای مساله مسیر مرکزی روی درخت الگوریتم‌های خطی ارائه شده است. اما مساله مسیر مرکزی روی شبکه یک مساله NP- سخت[۹۳] است[۳۸]. به همین دلیل ما در این پایان‌نامه الگوریتم ژنتیک[۹۴] و الگوریتم ترکیبی برای حل این مساله و مساله مسیر مرکزی با طول محدود روی شبکه پیشنهاد و نتایج این دو الگوریتم را بررسی و با هم مقایسه خواهیم کرد. در این فصل الگوریتم ژنتیک را برای این دو مساله را بیان خواهیم کرد.

۳-۱ الگوریتم ژنتیک چیست؟

الگوریتم ژنتیک یک الگوریتم جستجوی تصادفی است که از اصول انتخاب طبیعی داروین برای یافتن فرمول بهینه جهت پیش‌بینی یا تطبیق الگو استفاده می کند. الگوریتم ژنتیک در سال‌های اخیر به طور

---

93 Np-hard
94 Genetic Algorithms(GA)

گسترده‌ای برای مسائل بهینه‌سازی مخصوصا در مورد مسائل مکان‌یابی به عنوان مثال مسائل میانه[۹۵] و مسائل مکان‌یابی واسطه[۹۶] به کار گرفته شده است [۴۶،۴۷،۴۸،۴۹].

۳-۲ ایده اصلی

در دهه هفتاد میلادی دانشمندی از دانشگاه میشیگان به نام جان هولند[۹۷] ایده استفاده از الگوریتم ژنتیک را در بهینه‌سازی رشته‌های مهندسی مطرح کرد. ایده اساسی این الگوریتم انتقال خصوصیات موروثی توسط ژن‌ها[۹۸] است. فرض کنید مجموعه خصوصیات انسان توسط کروموزوم‌های[۹۹] او به نسل بعدی منتقل می‌شوند. هر ژنی در این کروموزوم‌ها نماینده یک خصوصیت است. اگر همه این کروموزوم به نسل بعد انتقال یابند، تمامی خصوصیات نسل بعدی شبیه به خصوصیات نسل قبل خواهد بود. بدیهی است که در عمل چنین اتفاقی رخ نمی‌دهد. در واقع بصورت همزمان دو اتفاق برای کروموزوم‌ها می‌افتد. اتفاق اول ترکیب [۱۰۰] است. یعنی چسبیدن ابتدای یک کروموزوم به انتهای یک کروموزوم دیگر. این همان چیزی است که مثلا باعث می‌شود تا فرزند[۱۰۱] تعدادی از خصوصیات پدر و تعدادی از خصوصیات مادر را با هم به ارث ببرد و از شبیه شدن تام فرزند به تنها یکی از والدین جلوگیری می‌کند. اتفاق دوم جهش[۱۰۲] است. جهش به این صورت است که بعضی ژن‌ها بصورت کاملا تصادفی تغییر می‌کنند. البته تعداد این گونه ژن‌ها بسیار کم می‌باشند. اما در هر حال این تغییر تصادفی بسیار مهم است. عمل ترکیب به تعداد بسیار بیشتری نسبت به عمل جهش رخ می‌دهد. برای مطالعه بیشتر می‌توانید به کتاب‌های گلدبرگ ***[۵۰]*** و ریوز ***[۵۱]*** مراجعه کنید. مطالب این بخش برگرفته از مرجع [۵۲] است.

۳-۳ چارچوب کلی الگوریتم ژنتیک

---

95 Median problem
96 Hub location problem
97 John Holland
98 Gens
99 Chromosom
100 Crossover
101 Child or offspring
102 Mutation

موتور الگوریتم ژنتیک در حالت کلی با یک مجموعه از جواب‌های ممکن اولیه (که به صورت تصادفی و یا به صورت ابتکاری تولید می‌شوند) و به آن اصطلاحا جمعیت اولیه[۱۰۳] گوییم ، شروع می‌شود. عناصر هر جمعیت را اصطلاحا نفر[۱۰۴] گوییم. الگوریتم ژنتیک با انتخاب تعدادی از زوج‌های جمعیت جاری به عنوان والدین[۱۰۵] و انجام عمل ترکیب بر روی آنها چند جواب ممکن یا فرزند از این والدین‌ها برای جمعیت جدید تولید می‌کند. علاوه بر این تعدادی از جواب‌ها با تطابق[۱۰۶] بسیار کم و یا با مقدار تابع هدف زیاد از جمعیت جاری خارج می‌شوند. این فرایند ادامه دارد تا اینکه شرط پایانی برقرار شود. در حین اجرای الگوریتم جهش‌ها و مهاجرت‌ها[۱۰۷] با تغییر بعضی از افراد و یا جایگزین کردن آنها با افراد بهتر صورت می‌گیرد. گاهی متناوبا از روش‌های بهینه محلی نیز برای بهبود جمعیت استفاده می‌شود. که اصطلاحا این الگوریتم‌ها را الگوریتم‌های ادغامی[۱۰۸] گوییم. جستجو در فضا را می‌توان به چند قسمت تقسیم کرد که اصطلاحا به این کار رقابت[۱۰۹] گوییم. یک رقابت یعنی اجرای الگوریتم ژنتیک با جمعیت‌های اولیه متفاوت و توقف آنها قبل از همگرایی و سپس ساختن یک جمعیت بهتر با انتخاب جواب‌های نهایی هر یک از این اجراها [۵۳].

۳-۴ روش اجرای الگوریتم

۳-۴-۱ تعیین نحوه نمایش یا کدبندی[۱۱۰] مساله

باید به هر جواب، یک نماد ژنتیکی[۱۱۱] اختصاص داد. عموماً کدگذاری‌ها به صورت ۲ تایی صفر و یک نشان داده می‌شوند، ولی روش‌های نمایش دیگری هم وجود دارند. نحوه نمایش، ارتباط مستقیم با مساله مورد نظر دارد و معمولا در رسیدن به جواب مناسب اثر بسیاری داشته و باید با دقت انتخاب شود.

۳-۴-۱-۱ انواع روش‌های نمایش

---

103 Initial Population
104 Individual
105 Parent
106 Fitness
107 Immigration
108 Hybrid Algorithms
109 Tournament
110 Coding
111 Genotype

قبل از این که یک الگوریتم ژنتیک برای یک مساله اجرا شود، یک روش برای کدکردن ژن‌ها به زبان کامپیوتر باید به کار رود. یکی از روش‌های معمول کدکردن به صورت رشته‌های باینری است یعنی رشته‌های ۰و۱. یک راه حل مشابه دیگر کدکردن جواب‌ها در آرایه‌ای از اعداد صحیح یا اعشاری است. این راه حل در مقایسه با قبلی پیچیده‌تر و مشکل‌تر است. مثلاً این روش توسط استفان کرمر، برای حدس ساختار سه بعدی پروتئین موجود در آمینواسیدها استفاده شده است. همچنین الگوریتم‌های ژنتیکی که برای آموزش شبکه‌های عصبی استفاده می‌شوند، از این روش استفاده می‌کنند. سومین روش برای نمایش صفات در یک الگوریتم ژنتیک یک رشته از حروف است، که هر حرف نمایش دهنده یک خصوصیت از جواب‌ها است.

۳-۴-۲ تعریف میزان برازندگی

ارزیابی هر رشته از لحاظ میزان بهینگی بر اساس نحوه رفتار[۱۱۲] حاصل از آن و از طریق محاسبه تابع هدف و اختصاص عدد برازندگی به آن صورت می گیرد.

۳-۴-۳ تولید فرزند

یک گام مهم دیگر در الگوریتم، تولد است که در هر دوره اتفاق می‌افتد. برای هر فرد، یک جفت والد انتخاب می‌شود. انتخاب‌ها به گونه‌ای هستند که مناسب‌ترین عناصر انتخاب شوند تا حتی ضعیف‌ترین عناصر هم شانس انتخاب شدن داشته باشند. این کار باعث می‌شود که از نزدیک شدن به جواب محلی جلوگیری شود. چندین الگوی انتخاب وجود دارد: چرخ منگنه‌دار(رولت)[۱۱۳]، انتخاب مسابقه‌ای[۱۱۴] و غیره. بعد از عمل انتخاب والدین با توجه به عملگرهای ژنتیکی مثل ترکیب و جهش فرزند تولید می‌کنیم.

۳-۴-۳-۱ روش‌های انتخاب

روش‌های مختلفی برای انتخاب کروموزم‌ها در الگوریتم‌های ژنتیک وجود دارند. اما روش‌های لیست شده در زیر از معمول‌ترین روش‌ها هستند.

---

112 Phenotype
113 Roulette
114 Tournament

انتخاب نخبه‌گرایی[۱۱۵]. مناسب‌ترین عضو هر اجتماع انتخاب می‌شود.

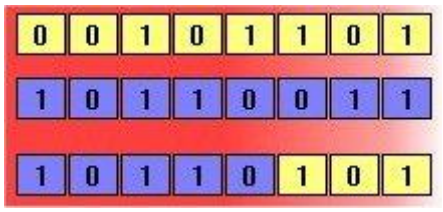
انتخاب رولت. یک روش انتخاب است که در آن عنصری که عدد برازش (تناسب) بیشتری داشته باشد، انتخاب می‌شود.

انتخاب مقیاسی[۱۱۶]. به موازات افزایش متوسط عدد برازش جامعه، سنگینی انتخاب بیشتر و جزئی‌تر می‌شود. این روش وقتی کاربرد دارد که مجموعه دارای عناصری باشد که عدد برازش بزرگی دارند و فقط تفاوت‌های کوچکی، آن‌ها را از هم تفکیک می‌کند.

انتخاب مسابقه‌ای. یک زیرمجموعه از صفات یک جامعه انتخاب می‌شوند و اعضای آن مجموعه با هم رقابت می‌کنند و سرانجام فقط یک صفت از هر زیر گروه برای تولید انتخاب می‌شوند.

۳-۴-۴ روش‌های تغییر

وقتی با روش‌های انتخاب، کروموزوم‌ها انتخاب شدند، باید به طور تصادفی برای افزایش تناسبشان اصلاح شوند. دو راه حل اساسی برای این کار وجود دارد. اولین روش ترکیب نام دارد. دو کروموزوم برای معاوضه بخشی[۱۱۷] از کدشان انتخاب می‌شوند. این فرآیند براساس فرآیند ترکیب کروموزوم‌ها، در طول تولیدمثل موجودات زنده شبیه‌سازی شده است. اغلب روش‌های معمول ترکیب، شامل برش یک‌نقطه‌ای[۱۱۸] هستند، که نقطه تعویض در جایی تصادفی بین ژن‌ها است. بخش اول قبل از نقطه، و بخش دوم بعد از نقطه ادامه پیدا می‌کند، که هر قسمت برگرفته از یک والد است، که ۵۰/۵۰ انتخاب شده‌اند.

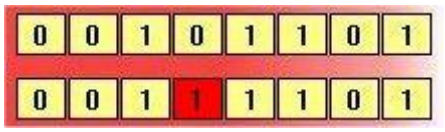


۳-۱ :تاثیر عملگر ترکیب بر روی کروموزوم‌ها

[115] Elitist
[116] Scaling
[117] Segment
[118] Single-point Crossover

کروموزوم را نشان می‌دهد که نقطه تعویض بین ۵ امین و ۶ امین مکان در کروموزوم قرار گرفته است. ایجاد یک کروموزوم جدید از پیوند این دو والد بدست می‌آید. معمولا عملگر ترکیب را به درصدی از جمعیت اعمال می‌کنند، که این درصد را با P_c نشان می‌دهند. و معمولا $P_c \in (0.5 , 1)$

دومین راه جهش نامیده می‌شود. به دلایل مختلف ممکن است تنوع و گوناگونی ژنی از بین برود و بازیابی آن با عملگر ترکیب غیرممکن شود. مثلا ممکن است در جمعیت تصادفی اولیه، همه رشته‌ها در یک ژن مخصوص مقدار یک داشته باشند ولی جواب بهینه در همان ژن مقدار صفر را داشته باشد. به این ترتیب هرگز به جواب بهینه نمی‌رسیم. بنابراین افزایش تنوع ژنی لازم و ضروری است. و این مهم با جهش صورت می‌گیرد. به عبارت دیگر اگر دست‌یابی به بعضی از نقاط فضای جستجو با عمل ترکیب ممکن نباشد، با جهش می‌توان به آن نقاط رسید. شکل زیر کروموزومی را نشان می‌دهد که ژن چهارم آن دچار جهش شده و صفر در آن مکان به یک تبدیل شده است.



۳-۲ :تاثیر عملگر جهش بر روی ژن چهارم

برای اعمال جهش:

۴- رشته‌ای از جمعیت را به تصادف انتخاب می‌کنیم؛

۵- یک ژن از آن رشته را به تصادف انتخاب می‌کنیم؛

۶- این ژن را تغییر می‌دهیم (مثلا در نمایش دودویی صفر به یک یا یک به صفر تغییر خواهند کرد).

نحوه اعمال جهش بر روی رشته بستگی به مساله و کدگذاری آن دارد. این عملگر روی درصدی از جمعیت و گاهی به صورت متناوب بر روی درصدی از جمعیت عمل می کند. این درصد را با P_m نمایش می‌دهیم. دو مقدار خوب برای این احتمال جهش یک بیت به صورت زیر پیشنهاد شده است:

$$P_m=\left(\sqrt{L}\times M\right)^{-1} \text{ یا } P_m=\frac{1}{L}$$

که در آن M، تعداد جمعیت و L طول کروموزم است. این فرآیندها باعث به وجود آمدن نسل جدیدی از کروموزوم‌هایی می شوند، که با نسل قبلی متفاوت هستند. این فرآیند تغییر تکرار می‌شود تا این‌که به آخرین مرحله برسیم.

۳-۴-۵ مشخص کردن شرط پایان

شرایط خاتمه الگوریتم‌های ژنتیک عبارتند از:

۶. به تعداد ثابتی از نسل‌ها برسیم.

۷. بودجه اختصاص داده شده تمام شود (زمان محاسبه/پول).

۸. یک فرد(فرزند تولید شده) پیدا شود که مینیمم (کمترین) ملاک را برآورده کند.

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

۱۰. ترکیب‌هایی از شرایط بالا.

۳-۵ چند نمونه از کاربردهای الگوریتم ژنتیک

نرم‌افزار شناسایی چهره با استفاده از تصویر ثبت شده به همت مبتکران ایرانی طراحی و ساخته شد. در این روش، شناسایی چهره براساس فاصله اجزای چهره و ویژگی‌های محلی و هندسی صورت می‌گیرد که تغییرات ناشی از تغییرات نور و افزایش سن کمترین تأثیر را خواهند داشت.

همچنین با استفاده از الگویتم‌های ژنتیک گراف‌هایی برای چهره‌های جدید ساخته شده است و با استفاده از یک تابع تشابه، قابل مقایسه با یکدیگر هستند که این امر تأثیر بسیار زیادی در افزایش سرعت شناسایی خواهد داشت.

از الگوریتم ژنتیک برای حل مسائل زیر نیز استفاده شده است:

توپولوژی‌های شبکه‌های کامپیوتری توزیع شده؛

بهینه‌سازی ساختار ملکولی شیمیایی؛

مهندسی برق برای ساخت آنتن مخصوص؛

مهندسی نرم‌افزار؛

بازی‌های کامپیوتری؛

مهندسی مواد؛

مهندسی سیستم؛

رباتیک[۱۱۹]؛

تشخیص الگو و استخراج داده؛[۱۲۰]

حل مساله فروشنده دوره‌گرد؛

آموزش شبکه‌های عصبی مصنوعی؛

یاددهی رفتار به ربات‌ها با استفاده از الگوریتم ژنتیک؛

یادگیری قوانین فازی با استفاده از الگویتم‌های ژنتیک؛

برای کسب اطلاعات بیشتر و کامل‌تر به آدرس www.talkorigins.org مراجعه شود.

۳-۶ کاربرد در جهان واقعی

مثال‌های بسیاری وجود دارد که برای حل آن‌ها از الگوریتم ژنتیک استفاده شده است. از جمله می‌توان به مساله فروشنده دوره‌گرد[۱۲۱] مساله زمان‌بندی[۱۲۲] و مساله بسته‌بندی[۱۲۳] اشاره کرد. مساله‌ی جدول زمانی پرسنل و کارکنان با استفاده از الگوریتم ژنتیک به صورت موفقیت‌آمیزی حل شده است. از الگوریتم ژنتیک برای لیست کاری پرستاران در یکی از بیمارستان‌های بزرگ انگلستان نیز استفاده شده است.

۳-۶-۱ کاربرد الگوریتم ژنتیک در پرتودرمانی

همچنین در ده‌های اخیر الگوریتم ژنتیک به عنوان ابزاری قوی برای بهینه‌سازی مسائل پرتودرمانی بکار رفته است. ایزل[۱۲۴] ، الگوریتم ژنتیک را برای یافتن ترکیب بهینه‌ای از پرتوهای خارجی بکار برد [۵۴]. پس

119 Robotics
120 Data mining
121 Traveling salesman problem
122 Scheduling problem
123 Bin packing problem
124 Ezzell

از آن لانگر[۱۲۵] از الگوریتم ژنتیک برای بهینه‌سازی تابع وزن پرتوها در درمان تومورهای بطنی استفاده کرد [۵۵]. ژیو و همکاران[۱۲۶] تکنیکی را برای بهینه‌سازی جهت و تابع وزن پرتوها با استفاده از الگوریتم ژنتیک ارائه کردند [۵۶]. از الگوریتم ژنتیک در بهینه‌سازی زاویه تابش پرتوهای پر انرژی نیز استفاده می‌شود. در حال حاضر در کلینیک‌های تخصصی پرتودرمانی برای تعیین زاویه تابش پرتوها از روش‌های تجربی استفاده می‌کنند. این در حال است که به طور قطع برای تعداد محدود و کم پرتوهای تابنده، زاویه تابش نقش مهمی را در عملکرد این روش تعیین می‌کند. بنابراین برای بهینه‌سازی تابع هدف، نیاز به یک الگوریتم انتخاب خودکار زاویه تابش ضروری می‌باشد. در این الگوریتم با توجه به پیچیدگی مساله و نامحدود بودن دامنه جستجو، بهینه‌سازی زاویه تابش و بهینه‌سازی نقشه شدت پرتوها به عنوان دو مساله جداگانه در نظر گرفته می‌شوند که برای بهینه‌سازی زاویه تابش و در بهینه‌سازی تابع وزن پرتوها به همراه ضریب اهمیت ارگان‌ها از الگوریتم ژنتیک استفاده می‌کنند [۵۷].

از الگوریتم ژنتیک برای برنامه‌ریزی فرآیند مبتنی بر کامپیوتر[۵۸] و برای انتخاب یک مجموعه دارایی از سهام بورس اوراق بهادار نیز استفاده می‌شود[۵۹].

۳-۷ الگوریتم پیشنهادی برای مساله مسیر مرکزی

۳ -۷-۱ کدگذاری

در کدگذاری که ما برای مساله مکان‌یابی مسیر مرکزی روی شبکه با استفاده از الگوریتم ژنتیک انجام می‌دهیم، کروموزم‌ها دارای تعداد ژن‌های مختلف هستند که ترتیب قرار گرفتن آنها مهم است. هر کروموزوم نشان دهنده یک مسیر روی شبکه است. و هر ژن متناظر است با اندیس راسی که در مسیر مورد نظر قرار

125 Langer
126 Zhu at el

می‌گیرد. به عنوان مثال کروموزم {۱،۲،۶،۹} با کروموزم {۲،۶،۹،۱} متفاوت است زیرا کروموزم اولی معرف مسیری روی شبکه از راس v_1 تا راس v_9 است که از راس‌های v_2 و v_6 نیز می‌گذرد. ولی کروموزم دومی معرف مسیری روی شبکه از راس v_2 تا v_1 است که به ترتیب از راس‌های v_6 و v_9 می‌گذرد.

۳-۷-۲ تعیین مطلوبیت

مطلوبیت یک کروموزم مقدار تابع هدف مساله به ازای جواب متناظر با این کروموزم می‌باشد. که در این مساله همان خروج از مرکز مسیر می‌باشد.

۳-۷-۳ جمعیت اولیه

اندازه جمعیت و جمعیت اولیه دو عامل مهم در همگرایی الگوریتم ژنتیک هستند. انتخاب جمعیت‌های بزرگ موجب کند شدن سرعت همگرایی شده و انتخاب جمعیت‌های کوچک باعث می‌شود که الگوریتم نتواند کل فضای جواب را برای یافتن جواب بهتر جستجو کند. همچنین انتخاب یک جمعیت اولیه مناسب موجب سرعت دادن به همگرایی الگوریتم می‌شود. ما اندازه جمعیت را n در نظر گرفتیم، که n تعداد رئوس شبکه می‌باشد.(با بررسی‌های انجام شده این اندازه جمعیت مناسب به نظر می‌آید) به عبارت دیگر برای تولید جمعیت اولیه باید n مسیر تولید کنیم. برای ساختن هر مسیر ما از یک راس شبکه شروع می‌کنیم و بقیه راس‌ها را به صورت تصادفی از بین رئوس مجاور انتخاب می‌کنیم. این روش را ادامه می‌دهیم تا جایی که راسی وجود نداشته باشد که بتوانیم به مسیر اضافه کنیم.

البته در ساختن مسیر به جای انتخاب راس‌ها به صورت تصادفی می‌توان از رئوسی استفاده کرد که دارای درجه بیشتر باشند که این روش مورد آزمایش قرار گرفت ولی به نتیجه بهتری منجر نشد زیرا این روش انتخاب باعث می‌شود که دائما یک سری از راس‌ها انتخاب شوند، در نتیجه تنوع ژنتیکی جمعیت کم می‌شود. و یا می‌توان در ساختن قسمتی از مسیر، ازکوتاهترین مسیر بین دو راس نیز استفاده کرد. که ما این روش را مورد آزمون قرار دادیم ولی به نتایج بهتری منجر نشد.(مدت اجرای الگوریتم نیز بالا می‌رود.)

۳-۷-۴ انتخاب والدین

برای تولید عضو جدید احتیاج به دو والد داریم که ما برای انتخاب والدین از بین جمعیت اولیه به گونه زیر عمل می‌کنیم: بهترین عضو جمعیت را به عنوان یکی از والدین انتخاب می‌کنیم. و والد دیگر را بطور تصادفی از بین جمعیت انتخاب می‌کنیم .البته روش‌های دیگری مثل انتخاب والدین به طور تصادفی و انتخاب اعضا با مطلوبیت بالاتر به عنوان والدین مورد آزمون قرار گرفت که به نتایج بهتری منجر نشد. همچنین انتخاب اعضای که مسیر بزرگتر دارند نیز به عنوان والدین مورد آزمون قرار گرفت که باز هم به نتایج بهتری منجر نشد.

۳-۷-۵ تولید عضو جدید

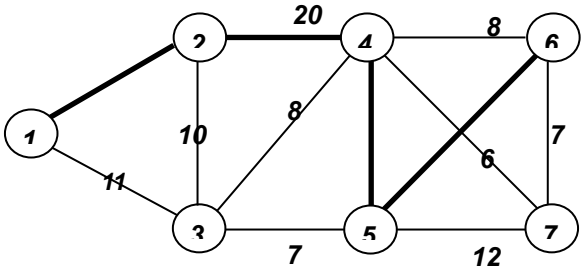
در الگوریتم ژنتیک معمولا کروموزم‌های والدین به دو قسمت شکسته شده و سپس با هم ترکیب می‌شوند تا دو عضو جدید تولید شود. ما در اینجا از یک روش دیگر استفاده می‌کنیم. ابتدا همه ژن‌های مشترک بین والدین را انتخاب می‌کنیم. ممکن است این ژن‌های مشترک با هم تشکیل مسیر ندهند یعنی جواب موجه نباشد. برای رفع این مشکل در این رشته بین هر دو زیرمسیر یک مسیر تولید می‌کنیم و به این رشته اضافه می‌کنیم. زیر مسیر جدید می‌تواند کوتاهترین مسیر بین آخرین راس زیرمسیر اول و راس ابتدایی زیرمسیر دوم باشد. این زیرمسیر در صورتی به رشته اصلی اضافه می‌شود که در رشته اصلی دور ایجاد نشود (به عبارت دیگر در صورت اضافه شدن زیرمسیر جدید هیچ راسی دو بار تکرار نشود). اگر در صورت اضافه شدن زیرمسیر به رشته اصلی دور ایجاد شد از کوتاهترین مسیر استفاده نمی‌کنیم.

در این صورت از زیرمسیری که بین این دو راس در یکی از والدین وجود دارد استفاده می کنیم این زیرمسیر جدید را به رشته اصلی اضافه می کنیم. برای انتخاب زیر مسیر، والدی را انتخاب می‌کنیم که تابع مطلوبیت بهتری داشته باشد. اگر در صورت اضافه کردن این زیرمسیر نیز دور ایجاد شد، در این صورت زیرمسیر دومی در رشته اصلی را حذف می‌کنیم. این روش را تا جای ادامه می‌دهیم که رشته اصلی به مسیر P_{new} تبدیل شود.

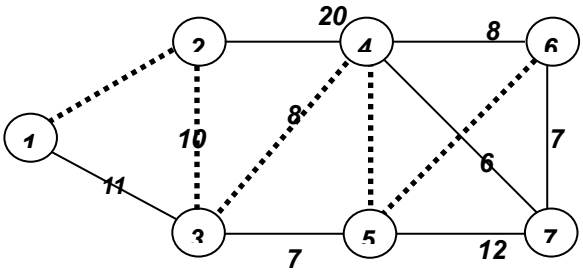
برای تولید زیرمسیر می‌توانیم به صورت تصادفی راس انتخاب کنیم تا زیر مسیر تولید شود اما تضمینی برای به دست آوردن زیر مسیر وجود ندارد. در نتیجه از این روش استفاده نمی‌کنیم.

۳-۷-۶ مثال

دو والد زیر را در نظر بگیرید. می‌خواهیم عضو جدیدی را تولید کنیم (در اشکال زیر بعضی یال‌ها بدون وزن هستند).



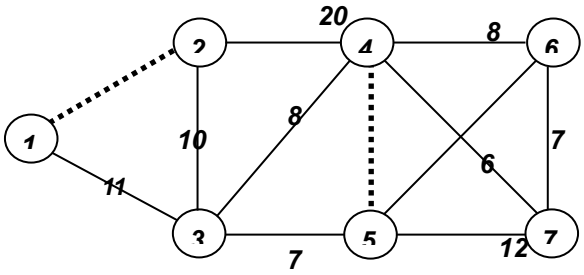
شکل۳-۳ والد اول(مسیرP_1 با خطوط مشکی مشخص شده است)



شکل ۳-۴ والد دوم (مسیرP_2 با خطوط نقطه چین مشخص شده است)

شکل ۳-۳ والد اول مسیرP_1= 1,2,4,5,6 را نشان می‌دهد که مقدار مطلوبیت آن هفت است یعنی $e(P_1)=7$. و شکل ۳-۴ والد دوم یعنی مسیر P_2= 1,2,3,4,5,6 را نشان می‌دهد که مقدار مطلوبیت این عضو برابر ۶ است، یعنی $e(P_2)=6$. حال برای تولید عضو جدید باید ژن‌های مشترک را مشخص کنیم.

{1, 2, 4, 5}=ژن‌های مشترک=رشته اصلی(S_{new})



شکل ۳-۵ ژن‌های مشترک دو والد(این ژن ها با نقطه چین مشخص شده است)

با توجه به شکل ۳-۵ بین راس دوم و راس چهارم مسیری وجود ندارد .در نتیجه باید بین این دو راس زیرمسیر ایجاد کنیم. زیرمسیرهای بین این دو راس شامل:

۴. مسیر P_1=2,4 که $L(P_1)$= 20

۵. مسیر P_2=2,3,4 که $L(P_2)$=18

۶. مسیر P_3=2,3,5,4 که $L(P_3)$=17

البته مسیرهای دیگری نیز بین این دو راس وجود دارد که به علت بالا بودن طول مسیر از ذکر آنها خودداری کردیم.

حال برای انتخاب زیرمسیر باید کوتاهترین مسیر را انتخاب کنیم. مسیر P_3=2,3,5,4 کوتاهترین مسیر است که باید این زیرمسیر را به رشته اصلی اضافه کنیم. در این صورت خواهیم داشت

S_{new}= {1, 2, 3, **5**, 4, **5**}

با توجه به S_{new} مشاهده می‌شود که راس ۵ دو بار تکرار شده است (یعنی دور ایجاد شده است) پس طبق روش گفته شده در بخش ۳-۷-۵ باید از زیرمسیری استفاده کرد که بین این دو راس در والد بهتر وجود دارد. والد بهتر والد دوم است (مقدار مطلوبیت آن کمتر است). در والد دوم بین این دو راس مسیر P_2 وجود دارد. که باید این مسیر را به S_{new} اضافه کرد. در این صورت خواهیم داشت

S_{new}= {1, 2, 3, 4, 5}

با توجه به S_{new} ،مشاهده می‌شود که هیچ راسی دو بار تکرار نشد پس رشته اصلی تبدیل به P_{new} شده است. داریم

P_{new}= {1, 2, 3, 4, 5}

۳-۷-۷ گسترش مسیر

حال از انتهای مسیر، راس اضافه می‌کنیم تا مسیرP_{new} گسترش یابد. این راس‌ها از بین رئوسی که در والدین هستند انتخاب می‌شوند. بنابراین برای همه راس‌های مجاور راس آخر مسیر P_{new} تابع g_1 را محاسبه می‌کنیم. اگر v یکی از رئوس مجاور با راس آخر باشد. آنگاه خواهیم داشت که

$$g_1(v)=\alpha\frac{deg(v)}{d\ max}+(1-\alpha)\frac{f(v)}{F(P)} \qquad (۱-۳)$$

ما با امتحان کردن α های مختلف مثل0/25, 0/75, 0/5, α=0 ,0/5 به این نتیجه رسیدیم که به ازای α=0/5 جواب بهتری به دست می‌آید. در رابطه فوق نمادها به صورت زیرهستند

deg (v) درجه راس v است.

d max بیشترین درجه راس‌های شبکه است.

F (P) تابع مطلوبیت مسیر P_{new} است.

f (v) مقدار بهبود تابع هدف به ازای اضافه شدن راس v به مسیر P_{new} است.

چرا تابع g_1 به این گونه تعریف شده است؟ در انتخاب راس‌ها برای ادامه مسیر دو نکته بسیار مهم است اول اینکه راسی که انتخاب می‌شود بهترین بهبود را در تابع هدف ایجاد کند. ما برای لحاظ این نکته کسر $\frac{f(v)}{F(P)}$ را در نظر گرفتیم و دوم اینکه راسی که انتخاب می‌شود دارای درجه راس بیشتری باشد تا باعث شود که مسیر ادامه پیدا کند و تنوع ژنتیکی جمعیت بیشتر شود (در این مساله محدودیت هزینه را در نظر نگرفتیم). برای لحاظ کردن این نکته ما کسر $\frac{deg(v)}{d\ max}$ را در نظر گرفتیم. برای در نظر گرفتن مقدار تاثیرگذاری این دو معیار از ضریب α استفاده کردیم. برای همه راس‌های مجاور با آخرین راس که در والدین وجود دارند، تابع g_1 را محاسبه می‌کنیم و با تابع احتمال زیر بهترین راس را انتخاب می‌کنیم.

$$p(v)=\frac{g_1(v)}{\sum_{v\in V'} g_1(v)} \qquad (۲-۳)$$

که 'V مجموعه راس‌های والدین هستند که می توان به مسیر P_{new} اضافه کرد. این روش اضافه کردن راس را ادامه می‌دهیم تا زمانی که راسی وجود نداشته باشد که بتوان به مسیر P_{new} اضافه کرد. و برای گسترش P_{new} همانند روش بالا از ابتدای مسیر نیز راس اضافه می‌کنیم.

چرا برای انتخاب راس‌ها از تابع احتمال p استفاده کردیم؟ما در انتخاب راس هم باید راس بهتر را انتخاب کنیم تا فرزندان بهتری تولید شوند و هم باید به نحوی راس‌ها را انتخاب کنیم که تنوع ژنتیکی جمعیت کم نشود. ما برای اینکه هر دو معیار رعایت شود از تابع احتمالp استفاده کردیم. البته روش‌های دیگری مانند

انتخاب راس به صورت تصادفی و انتخاب راس با تابع هدف بهتر نیز مورد آزمون قرار گرفت که به نتیجه بهتری منجر نشد.

۳-۷-۸ جهش

انجام عمل جهش در الگوریتم ژنتیک موجب می شود که الگوریتم بتواند از گیرافتادن در نقاط بهینه موضعی فرار کند. برای اجرای عمل جهش، بعد از تولید عضو جدید از ابتدا و انتهای مسیر همانند روش بالا راس اضافه می‌کنیم. با این تفاوت که برای انتخاب راس‌ها برای اضافه کردن به مسیر به جای انتخاب راس‌ها از والدین از همه رئوس شبکه استفاده می‌کنیم.

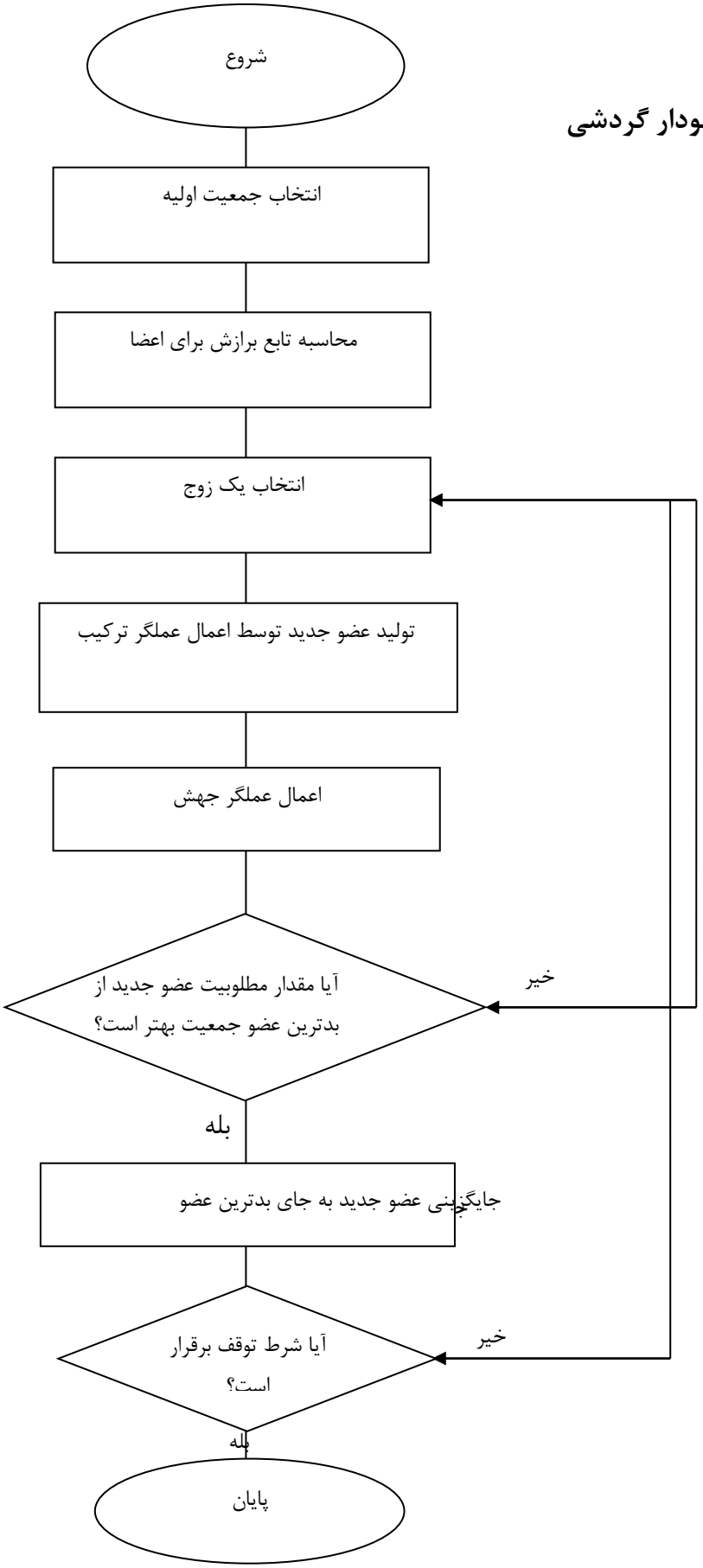
۳-۷-۹ جایگزینی

بعد از تولید یک عضو جدید و انجام عمل جهش باید این عضو را به جمعیت اضافه کنیم. اگر این عضو همانند هیچکدام از اعضای قبلی جمعیت نباشد و مقدار مطلوبیت آن از بدترین مقدار مطلوبیت جمعیت بهتر باشد آنگاه این عضو وارد جمعیت شده و عضوی از جمعیت که بدترین مقدار مطلوبیت را دارد از جمعیت خارج می‌شود.

۳-۷-۱۰ شرط توقف

شرط توقف می‌تواند رسیدن به حداکثر تعداد تکرار از ابتدای اجرای الگوریتم و یا بعد از آخرین بهبود باشد. همچنین یک حداکثر زمان اجرا را می‌توان به عنوان شرط توقف در نظر گرفت. ما رسیدن به n تکرار بعد از آخرین بهبود را به عنوان شرط پایانی در نظر گرفتیم.

۳-۷-۱۱ نمودار گردشی



مباحثی را که در بخش‌های قبل مطرح شد را می‌توان به صورت الگوریتم زیر بیان کرد.

۳-۸ الگوریتم ژنتیک برای مساله مسیر مرکزی(GAPCP[۱۲۷])

گام اول. جمعیت اولیه را با توجه به توضیحاتی که در بخش ۳-۷-۳ گفته شد، تولید کنید.

گام دوم. بهترین عضو و بدترین عضو جمعیت اولیه را پیدا کنید و مقدار تابع هدف هر کدام را (f_{best} , f_{worst}) نیز محاسبه کنید (بهترین عضو، عضوی است که دارای کمترین تابع هدف با کمترین طول همراه با بیشترین تعداد راس باشد). و بدترین عضو، عضوی است که دارای بیشترین تابع هدف با بیشترین طول همراه با کمترین تعداد راس باشد.

قرار دهید :

$r := 0 , r_{best} := 0$

گام سوم. تا زمانیکه $r\text{-}r_{best} \leq n$ مراحل زیر را انجام دهید:

۱) $r := r + 1$

۲) دو عضو p_1,p_2 را از جمعیت انتخاب کنید به طوری که p_1 بهترین عضو جمعیت و p_2 به صورت تصادفی انتخاب شود.

۳) عضو جدید تولید کنید:

۳-۱ همه یال‌های مشترک دو عضو جدید را انتخاب کنید و S_{new} بنامید.

۳-۲ برای هر دو راس u و v که در S_{new} قرار دارند و به هم وصل نیستند مراحل زیر را انجام دهید:

۳-۲-۱ کوتاهترین مسیر بین دو راس u وv را پیدا کنید (P_{uv}). اگر با اضافه کردن P_{uv} به S_{new} دوری[۱۲۸] ایجاد نشد به (۳-۲) بروید.

(الف) از زیرمسیری که بین u وv در والد با تابع مطلوبیت بهتر وجود دارد استفاده کنید. و اگر در صورت اضافه شدن این زیر مسیر به S_{new} باز هم دوری ایجاد نشد به ۳-۲ بروید.

---

127 Genetic Algorithm for Path Center Problem
128 Cycle

(ب) زیر مسیر دوم در S_{new} حذف کنید.

۳-۳ S_{new} را P_{new} بنامید.

۳-۴ x را راس پایانی P_{new} قرار دهید.

۳-۵ تا زمانیکه اضافه کردن راس دور ایجاد نکند مراحل زیر را انجام دهید:

۳-۵-۱ برای راس v ،که $v \in P_1 \cup P_2 \backslash P_{new}$ و به x وصل است $g_1(v)$ را محاسبه کنید و با استفاده از رابطه (۳-۲) بهترین راس را انتخاب کنید. و t بنامید.

۳-۵-۲ راس t را به P_{new} اضافه کنید.

۳-۵-۳ $x = t$

۴) انجام عمل جهش:

۴-۱ راس ابتدایی و انتهایی P_{new} را x_1, x_2 بنامید.

۴-۲ تا زمانیکه اضافه کردن راس دور ایجاد نکند مراحل زیر را انجام دهید:

۴-۲-۱ $x = x_1, x_2$.

۴-۲-۲ برای راس v، که $v \in G \backslash P_{new}$ و به x وصل باشد $g_1(v)$ را محاسبه کنید و با استفاده از تابع احتمال (۳-۲) راس t را انتخاب کنید.

۴-۲-۳ راس t را به P_{new} اضافه کنید.

۴-۴-۳ $x = t$

۵) تابع مطلوبیت را برای عضو جدید محاسبه کنید $F(P_{new})$. اگر عضو جدید در جمعیت وجود نداشت و $F(P_{new}) < f_{worst}$ مراحل زیر را انجام دهید:

۵-۱ عضو جدید را به جمعیت اضافه کنید و بدترین عضو را از جمعیت خارج کنید.

۵-۲ اگر $F(P_{new}) < f_{best}$ قرار دهید:

$f_{best} = F(P_{new})$, $r_{best} = r$.

گام چهارم. پایان

۳-۹ الگوریتم ژنتیک برای مساله مسیر مرکزی با طول محدودL (GAPCPBL[۱۲۹])

الگوریتم ژنتیک برای مساله مسیر مرکزی با طول محدودL همانند الگوریتم بخش ۳-۸ است با این تفاوت که در این الگوریتم از تابع g_2 به جای تابع g_1 استفاده می‌کنیم. تابع g_2 را به صورت زیر تعریف می‌کنیم:

اگر v یکی از رئوس مجاور با آخرین راس x باشد. آنگاه خواهیم داشت که

$$g_2(v)=\alpha\frac{f(v)}{F(P)}-(1-\alpha)\frac{L_{xv}}{L_2}. \qquad (۳-۳)$$

در تابع g_2 L_{xv} طول یال اضافه شده است و L_2 طول کل مسیر(P_{new}) است.

همانند تابع g_1 دو نکته در انتخاب راس‌ها بسیار مهم است اول اینکه راسی که انتخاب می‌شود بهترین بهبود را در تابع هدف ایجاد کند. ما برای لحاظ این نکته کسر $\frac{f(v)}{F(P)}$ را در نظر گرفتیم. در تابع g_1 محدودیت طولی نداشتیم و هدف این بود که مسیری پیدا کنیم که از همه راس‌های شبکه بگذرد. اما در تابع g_2 ما محدودیت طول داریم بنابراین کسر L_{xv}/L_2 را در نظر گرفتیم. و این کسر را منفی در نظر گرفتیم زیرا هر چه طول کمتر باشد برای ما بهتر است. ما از کسر $L_{xv}/\max L$ (max L=max w'_{ij}) به جای کسر L_{xv}/L_2 نیز استفاده کردیم که به نتایج تقریبا مشابه رسیدیم. به ازای تابع g_3 نیز الگوریتم را تست کردیم که به نتایج بهتری منجر نشد.

$$g_3=\frac{f(v)}{F(P)}+\frac{\deg(v)}{d\max}-\frac{L_{xv}}{L_2}. \qquad (۳-۴)$$

برای در نظر گرفتن مقدار تاثیرگذاری این دو معیار درتابع g_2 از ضریب α استفاده کردیم. تابع g_2 را برای ***α*** های مختلف مثل 0/4,0/6,0/5,0/25,0/75 تست کردیم که به ازای α =0/5 جواب بهتری به دست آمد.

با توجه به توضیحات بالا الگوریتم ژنتیک برای مساله مسیر مرکزی با طول محدود به صورت زیر خواهد بود:

گام اول. برای ساختن جمعیت اولیه همانند بخش ۳-۷-۳ عمل می کنیم با این تفاوت که برای ساختن هر مسیر و اضافه کردن راس تا جایی می‌توانیم راس اضافه کنیم که طول مسیر از L بیشتر نشود.

[129] Genetic Algorithm for Path Center Problem with Bounded Length

گام دوم. بهترین عضو و بدترین عضو جمعیت اولیه را پیدا کنید و مقدار تابع هدف بهترین عضو f_{best} و مقدار تابع هدف بدترین عضو f_{worst} را محاسبه کنید. و قرار دهید :

$r := 0 , r_{best} := 0$

گام سوم. تا زمانیکه $r - r_{best} \le n$ مراحل زیر را انجام دهید:

۱) $r := r + 1$

۲) دو عضو p_1, p_2 را از جمعیت انتخاب کنید به طوری که p_1 بهترین عضو جمعیت و p_2 به صورت تصادفی انتخاب شود.

۳) عضو جدید تولید کنید:

۳-۱ همه یال‌های مشترک دو عضو جدید را انتخاب کنید و S_{new} بنامید

۳-۲ برای هر دو راس u و v که در S_{new} قرار دارند و به هم وصل نیستند مراحل زیر را انجام دهید:

۳-۲-۱ کوتاهترین مسیر بین دو راس u وv را پیدا کنید (P_{uv}). اگر با اضافه کردن P_{uv} به S_{new} دوری ایجاد نشد به ۳-۲ بروید.

(الف) از زیرمسیری که بین u وv در والد با تابع مطلوبیت بهتر وجود دارد استفاده کنید. و اگر در صورت اضافه شدن این زیرمسیر به S_{new} باز هم دور ایجاد نشد به ۳-۲ بروید.

(ب) زیرمسیر دوم در S_{new} راحذف کنید.

۳-۳ S_{new} را P_{new} بنامید.

۳-۴ اگر طول P_{new} بیشتر از L بود کارهای زیر را انجام دهید:

۳-۴-۱ راس آخر مسیر P_{new} را حذف کنید

۳-۴-۲ بقیه مسیر را P_{new} بنامید.

۳-۵ x را راس پایانی P_{new} قرار دهید.

۳-۶ تا زمانیکه اضافه کردن راس دور ایجاد نکند و طول P_{new} بیشتر ازL نشود مراحل زیر را انجام دهید:

۳-۶-۱ برای راس v،که $v \in P_1 \cup P_2 \setminus P_{new}$ و به x وصل است $g_2(v)$ را محاسبه کنید

و با استفاده از تابع احتمال ۳-۵ راس t را انتخاب کنید.

$$p(v)=\frac{g_2(v)}{\sum_{v\in V'} g_2(v)} \qquad (۵-۳)$$

که 'V مجموعه راس‌های والدین هستند که می توان به مسیر P_{new} اضافه کرد.

۳-۶-۲ راس t را به P_{new} اضافه کنید.

۳-۶-۳ $x = t$

۴) انجام عمل جهش:

۴-۱ راس ابتدایی و انتهایی P_{new} را x_1,x_2 بنامید.

۴-۲ تا زمانیکه اضافه کردن راس دور ایجاد نکند و طول مسیر از L بیشتر نشود مراحل زیر را انجام دهید:

۴-۲-۱ $x = x_1,x_2$

۴-۲-۲ برای راس v، که $v\in G\backslash P_{new}$ و به x وصل است $g_2(v)$ را محاسبه کنید و با استفاده از تابع احتمال ۳-۵ راس t را انتخاب کنید.

۴-۲-۳ راس t را به P_{new} اضافه کنید.

۳-۴-۳ $x = t$

۵) تابع مطلوبیت را برای عضو جدید محاسبه کنید $F(P_{new})$. اگر عضو جدید در جمعیت وجود نداشت و $F(P_{new}) < f_{worst}$ مراحل زیر را انجام دهید:

۵-۱ عضو جدید را به جمعیت اضافه کنید و بدترین عضو را از جمعیت خارج کنید.

۵-۲ اگر $F(P_{new}) < f_{best}$ قرار دهید:

$f_{best} = F(P_{new})$, $r_{best} = r$.

گام چهارم. پایان

۳-۱۰ نتایج

ما الگوریتم ژنتیک برای مساله مسیر مرکزی و مساله مسیر مرکزی با طول محدود را توسط مسائل نمونه‌ای [60] تست کردیم. هر کدام از مسائل را پنج بار توسط الگوریتم اجرا کردیم. برای مساله مسیر مرکزی بهترین جواب را به عنوان جواب اصلی انتخاب کردیم. و برای مساله مسیر مرکزی با طول محدود میانه جواب‌ها

را به عنوان جواب اصلی انتخاب کردیم. در این مسائل وزن همه راس‌ها برابر یک است ولی وزن یال‌ها برابر یک نیست.

برای تست الگوریتم مسیر مرکزی با طول محدود از دو طول متفاوت استفاده کردیم. در شبکه مربوط به مسائل نمونه‌ای حداقل یک سیکل مشاهده می‌شود. ما از طول این سیکل برای تست کردن الگوریتم استفاده کردیم. دومین طولی که در نظر گرفتیم، طول ۲۵۰۰ است که برای همه مسائل مشترک در نظر گرفتیم.

در جدول ۳-۱ ستون.چهارم، پنجم و ستون ششم تابع هدف، تعداد راس و طول جواب اولیه است که الگوریتم با آن آغاز می‌شود. ستون هفتم، هشتم و ستون نهم تابع هدف ، تعداد راس و طول بهترین جوابی است که با استفاده از الگوریتم ژنتیک به دست آوردیم. و ستون دهم تابع هدف بدترین جواب در جمعیت نهایی مساله است. ستون یازدهم تعداد تکرار بعد از آخرین بهبود است. و ستون آخر نشان دهنده زمان اجرا به ازای هر تکرار می‌باشد.

در جدول ۳-۲ ستون چهارم طول سیکل نمونه ای برای هر مساله است. ستون پنجم، ششم و ستون هفتم تابع هدف، تعداد راس و طول جواب اولیه است که الگوریتم با آن آغاز می‌شود. ستون هشتم، نهم و ستون دهم تابع هدف، تعداد راس و طول جواب نهایی (بهترین جواب) الگوریتم است. و ستون یازدهم تابع هدف بدترین جواب در جمعیت نهایی مساله است.

با توجه به نتایج به دست آمده در جدول ۳-۲ تقریبا برای همه مسائل جواب نهایی نسبت به جواب اولیه بهبود داشته است. فقط در مساله سوم با طول سیکل نمونه و مساله دوم با طول ۲۵۰۰ بهبودی در تابع هدف جواب نهایی نسبت به جواب اولیه مشاهده نشده است. اما تعداد راس ها نسبت به تعداد راس جواب اولیه بیشتر شده است. یعنی تعداد بیشتری از راس ها تحت پوشش قرار گرفتند.

جدول ۳-۱ نتایج الگوریتم ژنتیک برای مساله مسیر مرکزی

Test #	n	Edge NO.	Objective function	*Initial* Vertex NO	Length P	Objective function	*Best* Vertex NO	LengthP	*Worst* Objective function	Last improv.iter	Cpu / Iter(sec)
1	100	200	82	57	3208	60	54	3515	98	138	0.104
2	100	200	86	65	3532	68	71	4742	98	179	0.123
3	100	200	72	75	3864	72	75	2982	72	176	0.125
4	100	200	91	65	3630	91	65	3630	116	101	0.109
5	100	200	77	60	3168	77	60	3168	92	101	0.1
6	200	800	24	170	8571	24	170	8571	68	201	0.388
7	200	800	38	134	6711	38	168	7350	55	209	0.259
8	200	800	46	140	7111	38	125	4948	60	473	0.281
9	200	800	46	149	7506	38	143	5548	71	278	0.396
10	200	800	34	146	6835	34	154	8546	59	317	0.376

جدول ۳-۲ نتایج الگوریتم ژنتیک برای مساله مسیر مرکزی با طول محدود

Test #	n	Edge NO.	Length	Initial Objective function	Initial Vertex NO	Initial LengthP	best Objective function	best Vertex NO	best LengthP	worst Objective function	Last improv.iter	Cpu / Iter(sec)
1	100	200	5264	82	57	3208	76	77	3393	76	181	0.106
2	100	200	5313	86	65	3532	57	78	4329	94	145	0.135
3	100	200	5235	69	80	3982	69	82	2065	69	181	0.118
4	100	200	5294	91	65	3630	81	70	3838	81	222	0.116
5	100	200	5573	83	55	2870	73	67	3698	87	148	0.154
6	200	800	9854	24	170	8571	23	171	5705	43	397	0.418
7	200	800	10208	38	134	6711	23	191	6078	23	964	0.441
8	200	800	10062	31	156	3129	24	177	4595	34	501	0.167
9	200	800	10403	46	149	7506	17	187	9039	38	500	1.17
10	200	800	9882	34	146	6835	22	180	8503	22	837	0.394

Test #	n	Edge NO.	Length	Initial Objective function	Initial Vertex NO	Initial LengthP	best Objective function	best Vertex NO	best LengthP	worst Objective function	Last improv.iter	Cpu / Iter(sec)
1	100	200	2500	88	47	2482	79	50	2466	90	237	0.084
2	100	200	2500	96	52	2468	96	54	2493	96	366	0.094
3	100	200	2500	92	42	2435	87	46	2489	87	239	0.07
4	100	200	2500	98	45	2499	96	46	2455	106	237	0.487
5	100	200	2500	86	45	2347	85	49	2463	85	437	0.066
6	200	800	2500	66	51	2499	61	51	2481	79	298	0.129
7	200	800	2500	63	53	2486	57	62	2498	63	917	0.137
8	200	800	2500	72	55	2491	71	60	2450	88	238	0.201
9	200	800	2500	69	59	2499	63	76	2491	63	761	0.327
10	200	800	2500	56	60	2497	51	62	2487	51	594	0.139

فصل چهارم

الگوریتم ترکیبی مورچه و ژنتیک

درفصل قبل برای دو مساله مسیر مرکزی روی شبکه و مساله مسیر مرکزی با طول محدود روی شبکه الگوریتم ژنتیک را پیشنهاد دادیم. با توجه به نتایج به دست آمده از الگوریتم ژنتیک و به منظور بهبود نتایج الگوریتم ترکیبی مورچه و ژنتیک را در این فصل پیشنهاد می‌دهیم. در انتهای این فصل نتایج این الگوریتم ارائه و با نتایج الگوریتم ژنتیک مقایسه می‌کنیم.

۴-۱ روش مورچه[۱۳۰]

روش مورچه، روشی نسبتا جدید برای حل مسائل بهینه‌سازی می باشد که اولین بار توسط دریگو[۱۳۱]، مانیزو[۱۳۲] و کلرنی***133*** برای بعضی مسائل مشکل بهینه‌سازی از قبیل مساله فروشنده دوره‌گرد و مساله تخصیص درجه دو[۱۳۴] مورد استفاده قرار گرفت [61]. همچنین الگوریتم مورچه برای دیگر مسائل بهینه‌سازی گسسته از قبیل مساله رنگ‌آمیزی گراف[۱۳۵]، مساله مرتب‌سازی تصادفی[۱۳۶]، حرکت در شبکه‌های ارتباطی[۱۳۷] مسائل زمان‌بندی[۱۳۸] و مسائل مکان‌یابی مورد استفاده قرار گرفته است. (برای اطلاعات بیشتر به [۶۲،۶۳،۶۴،۶۵] مراجعه کنید).

روش مورچه برگرفته از مشاهده رفتار مورچه‌های واقعی می‌باشد. مورچه‌ها تقریبا نابینا هستند. آنها حین حرکت بین لانه و غذا ماده‌ای به نام فرمون[۱۳۹] از خود به جای می‌گذارد. در بین تمام مسیرهای بین لانه و غذا

130 Ant colony
131 Dorigo
132 Maniezzo
133 Colorni
134 Quadratic assignment problem
135 Graph coloring
136 Sequential ordering
137 Routing in communication networks
138 Job-shop scheduling
139 Pheromone

یک مورچه مسیری را که دارای فرمون بیشتری است با احتمال بیشتری انتخاب می‌کند دیگر مورچه‌ها هم این روش را تکرار می‌کنند و بعد از مدتی کوتاهترین مسیر دارای بیشترین فرمون خواهد شد[۱].

الگوریتم ترکیبی ما براساس الگوریتم ژنتیک است. و فقط در هنگام اضافه کردن راس به مسیر در قسمت جهش و تولید عضو جدید از روش مورچه استفاده می‌کنیم.

ما در ابتدای الگوریتم به هر یال عضو جمعیت اولیه فرمون ***T*** را نسبت می‌دهیم. اگر $P_1,P_2,..,P_n$ اعضای جمعیت اولیه باشند آن گاه برای هر یال e_{uv} داریم

$$\tau(e_{uv})=\sum_{\{P_i\,|e_{uv}\in P_i\}}\frac{1}{F(P_i)} \qquad (۱-۴)$$

و در هر تکرار فرمون یال $e_{uv}\in P_{new}$ توسط فرمول زیر بهنگام می‌شود

$$\tau(e_{uv})=\rho\frac{1}{F(P_{new})}+(1-\rho)\tau(e_{uv}) \qquad \rho\in[0,1] \qquad (۲-۴)$$

در الگوریتم ترکیبی در گام‌های تولید عضو جدید و جهش(گام سوم و چهارم) و در هنگام اضافه کردن یال جدید به طرفین مسیر P_{new} به جای استفاده کردن از روش ژنتیک از تابع احتمال زیر استفاده می‌کنیم

$$P(e_{uv}):=\frac{\tau(e_{uv})}{\sum_{e_{xy}\in A}\tau(e_{xy})} \qquad (۳-۴)$$

که A مجموعه راس‌هایی است که می‌توان به مسیر P_{new} اضافه کرد.

۴-۲ الگوریتم ترکیبی مورچه و ژنتیک برای مساله مسیر مرکزی(GACPCP[۱۴۰])

ساختار الگوریتم پیشنهادی ما به صورت زیر است:

گام اول. جمعیت اولیه را با توجه به توضیحاتی که در بخش ۳-۷-۳ گفته شد، تولید کنید.

[140] Genetic and Ant colony for Path Center Problem

گام دوم. بهترین عضو و بدترین عضو جمعیت اولیه را پیدا کنید و مقدار تابع هدف هر کدام را (f_{best}, f_{worst}) محاسبه کنید. (تعریف بهترین عضو و بدترین عضو در بخش ۳-۸ آمده است.) و قرار دهید:

$r := 0 , r_{best} := 0$

فرمون ها را طبق معادله ۴-۱ به یال های اعضای جمعیت اولیه نسبت دهید.

گام سوم. تا زمانیکه $r\text{-}r_{best} \leq n$ مراحل زیر را انجام دهید:

۱) $r := r + 1$

۲) دو عضو p_1, p_2 را از جمعیت انتخاب کنید به طوری که p_1 بهترین عضو جمعیت و p_2 به صورت تصادفی انتخاب شود.

۳) عضو جدید تولید کنید:

۳-۱ همه یال های مشترک دو عضو جدید را انتخاب کنید و S_{new} بنامید

۳-۲ برای هر دو راس u و v که در S_{new} قرار دارند و به هم وصل نیستند مراحل زیر را انجام دهید:

۳-۲-۱ کوتاهترین مسیر بین دو راس u وv را پیدا کنید(P_{uv}). اگر با اضافه کردن P_{uv} به S_{new} دوری ایجاد نشد به ۳-۲ بروید.

(الف) از زیر مسیری که بین u وv در والد با تابع مطلوبیت بهتر وجود دارد استفاده کنید. و اگر در صورت اضافه شدن این زیر مسیر به S_{new} باز هم دور ایجاد نشد به ۳-۲ بروید.

(ب) زیر مسیر دوم در S_{new} راحذف کنید.

۳-۳ S_{new} را P_{new} بنامید.

۳-۴ x را راس پایانی P_{new} قرار دهید.

۳-۵ تا زمانی که اضافه کردن راس دور ایجاد نکند مراحل زیر را انجام دهید:

۳-۵-۱ با توجه به معادله ۴-۳ راس v را که $v \in P_1 \cup P_2 \setminus P_{new}$ و به x وصل است را انتخاب کنید و به P_{new} اضافه کنید.

۴) انجام عمل جهش:

۴-۱ راس ابتدایی و انتهایی P_{new} را x_1, x_2 بنامید.

۴-۲ تا زمانیکه اضافه کردن راس دور ایجاد نکند مراحل زیر را انجام دهید:

۴-۲-۱ $x = x_1, x_2$

۴-۲-۲ با توجه به معادله ۴-۳ راس v را که $v \in G \setminus P_{new}$ و به x وصل است را انتخاب کنید و به P_{new} اضافه کنید.

۵) تابع مطلوبیت را برای عضو جدید محاسبه کنید $F(P_{new})$.اگر عضو جدید در جمعیت وجود نداشت و $F(P_{new}) < f_{worst}$ مراحل زیر را انجام دهید:

۵-۱ عضو جدید را به جمعیت اضافه کنید و بدترین عضو را از جمعیت خارج کنید.

۵-۲ اگر $F(P_{new}) < f_{best}$ قرار دهید

$f_{best} = F(P_{new})$, $r_{best} = r$

۵-۳ برای هر یال $e_{uv} \in P_{new}$ ،طبق فرمول ۴-۲ فرمون ها را بهنگام کنید.

گام چهارم. پایان

4-3 الگوریتم ترکیبی مورچه و ژنتیک برای مساله مسیر مرکزی با طول محدود[۱۴۱]

[141] Genetic and Ant colony for Path Center Problem with Bounded Length (GACPCPBL)

الگوریتم ترکیبی را می‌توان برای مساله مسیر مرکزی با طول محدود به کار برد. ساختار الگوریتم پیشنهادی ما به صورت زیر است:

گام اول. جمعیت اولیه را با توجه به توضیحاتی که در گام اول الگوریتم ژنتیک بخش ۳-۹گفته شد، تولید کنید.

گام دوم. بهترین عضو و بدترین عضو جمعیت اولیه را پیدا کنید و مقدار تابع هدف هر کدام را (f_{best} , f_{worst}) نیز محاسبه کنید. و قرار دهید:

$r := 0$, $r_{best} := 0$

فرمون‌ها را طبق معادله ۴-۱ به یال‌های اعضای جمعیت اولیه نسبت دهید

گام سوم. تا زمانیکه $r - r_{best} \leq n$ مراحل زیر را انجام دهید:

۱) $r := r + 1$

۲) دو عضو P_1, P_2 را از جمعیت انتخاب کنید به طوری که P_1 بهترین عضو جمعیت و P_2 به صورت تصادفی انتخاب شود.

۳) عضو جدید تولید کنید:

۳-۱ همه یال‌های مشترک دو عضو جدید را انتخاب کنید و S_{new} بنامید.

۳-۲ برای هر دو راس u و v که در S_{new} قرار دارند و به هم وصل نیستند مراحل زیر را انجام دهید:

۳-۲-۱ کوتاهترین مسیر بین دو راس u وv را پیدا کنید (P_{uv}). اگر با اضافه کردن P_{uv} به S_{new} دوری ایجاد نشد به ۳-۲ بروید.

(الف) از زیرمسیری که بین u وv در والد با تابع مطلوبیت بهتر وجود دارد استفاده کنید. و اگر درصورت اضافه شدن این زیر مسیر به S_{new} باز هم دور ایجاد نشد به ۳-۲ بروید.

(ب) زیرمسیر دوم در S_{new} راحذف کنید.

۳-۳ S_{new} را P_{new} بنامید.

۳-۴ اگر طول P_{new} بیشتر از L بود کارهای زیر را انجام دهید:

۳-۴-۱ راس آخر مسیر P_{new} را حذف کنید

۳-۴-۲ بقیه مسیر را P_{new} بنامید.

۳-۵ x را راس پایانی P_{new} قرار دهید.

۳-۶ تا زمانی که اضافه کردن راس دور ایجاد نکند و طول P_{new} از L کمتر باشد مراحل زیر را انجام دهید:

۳-۶-۱ با توجه به معادله ۳-۴ راس v را که $v \in P_1 \cup P_2 \backslash P_{new}$ و به x وصل است را انتخاب کنید و به P_{new} اضافه کنید.

۴) انجام عمل جهش:

۴-۱ راس ابتدایی و انتهایی P_{new} را x_1, x_2 بنامید.

۴-۲ تا زمانیکه اضافه کردن راس دور ایجاد نکند و طول مسیر P_{new} از L بیشتر نشود مراحل زیر را انجام دهید:

۴-۲-۱ $x = x_1, x_2$

۴-۲-۲ با توجه به معادله ۳-۴ راس v را که $v \in G \backslash P_{new}$ و به x وصل است را انتخاب کنید و به P_{new} اضافه کنید.

۵) تابع مطلوبیت را برای عضو جدید محاسبه کنید$F(P_{new})$. اگر عضو جدید در جمعیت وجود نداشت و $F(P_{new}) < f_{worst}$ مراحل زیر را انجام دهید:

۵-۱ عضو جدید را به جمعیت اضافه کنید و بدترین عضو را از جمعیت خارج کنید.

۵-۲ اگر $F(P_{new}) < f_{best}$ قرار دهید

$f_{best} = F(P_{new})$, $r_{best} = r$

۵-۳ برای هر یال $e_{uv} \in P_{new}$ ،طبق فرمول ۴-۲ فرمون ها را بهنگام کنید.

گام چهارم. پایان.

۴-۴ نتایج دو الگوریتم

الگوریتم ترکیبی مساله مسیر مرکزی و مساله مسیر مرکزی با طول محدود را برای مسائل نمونه‌ای [۶۰] به کار برده شد و هر کدام از مسائل را ۵ بار تست و برای مساله مسیر مرکزی بهترین جواب و برای مساله مسیر مرکزی با طول محدود میانه جواب‌ها به عنوان بهترین جواب انتخاب شد. همه مسائل برای ρ=0/2,0/5,0/6,0/75,0/9 تست شده است، که با ρ=0/5 جواب بهتری را به دست آمد. در جدول ۴-۱ به دلیل اینکه در همه مسائل، جواب نهایی نسبت به جواب اولیه بهبود داشته است از ذکر جزئیات جواب اولیه خودداری کردیم.

جدول ۴-۱ نتایج الگوریتم ترکیبی برای مساله مسیر مرکزی

			Initial	*Best*			Worst		
Test #	**n**	**Edge NO.**	**Objective function**	**Objective function**	**Vertex NO**	**LengthP**	**Objective function**	**Last improv.iter**	**Cpu / Iter(sec)**
1	100	200	73	59	92	4200	63	251	0.136
2	100	200	81	33	90	5222	45	281	0.142
3	100	200	80	42	91	3290	53	487	0.14
4	100	200	83	55	75	4154	82	425	0.133
5	100	200	72	54	71	4130	70	330	0.123
6	200	800	24	21	183	6155	21	597	0.404
7	200	800	38	33	171	6351	52	390	0.364
8	200	800	44	27	182	4847	68	327	0.432
9	200	800	46	20	182	9948	68	246	0.427
10	200	800	34	23	181	8736	38	510	0.41

جدول ۴-۲مقایسه نتایج الگوریتم ترکیبی و ژنتیک برای مساله مسیر مرکزی

Test	n	Edge NO.	Objective function		Last improv.iter		Cpu / Iter(sec)	
			GACPCP	GAPCP	GACPCP	GAPCP	GACPCP	GAPCP
1	100	200	59	60	251	138	0.136	0.104
2	100	200	33	68	281	179	0.142	0.123
3	100	200	42	72	487	176	0.14	0.125
4	100	200	55	91	425	101	0.133	0.109
5	100	200	54	77	330	101	0.123	0.1
6	200	800	21	24	597	201	0.404	0.388
7	200	800	33	38	390	209	0.364	0.259
8	200	800	27	38	327	473	0.432	0.281
9	200	800	20	38	246	278	0.427	0.396
10	200	800	23	34	510	317	0.41	0.376

جدول ۴-۳نتایج الگوریتم ترکیبی برای مساله مسیر مرکزی با طول محدود

				Initial			best			worst		
Test #	**n**	**Edge NO.**	**Length**	**Objective function**	**Vertex NO**	**Length P**	**Objective function**	**Vertex NO**	**LengthP**	**Objective function**	**Last improv.iter**	**Cpu / Iter(sec)**
1	100	200	5264	73	57	2919	71	74	3499	71	228	0.134
2	100	200	5313	81	82	4001	64	77	4312	98	136	0.154
3	100	200	5235	69	80	3982	62	89	2602	69	263	0.152
4	100	200	5294	83	60	2998	72	70	3826	75	239	0.127
5	100	200	5573	83	55	2870	67	75	4128	70	205	0.127
6	200	800	9854	24	170	8571	14	184	6093	17	443	0.408
7	200	800	10208	38	134	6711	36	176	7318	36	480	0.340
8	200	800	10062	46	140	7114	34	177	6116	34	563	0.434
9	200	800	10403	46	149	7506	37	174	4649	37	493	0.241
10	200	800	9882	34	146	6835	34	176	8617	34	510	0.362

				Initial			best			worst		
Test #	**n**	**Edge NO.**	**Length**	**Objective function**	**Vertex NO**	**Length P**	**Objective function**	**Vertex NO**	**LengthP**	**Objective function**	**Last improv.iter**	**Cpu/Iter (sec)**
1	100	200	2500	88	47	2482	77	51	2496	89	533	0.099
2	100	200	2500	98	52	2454	94	48	2467	117	164	0.106
3	100	200	2500	74	49	2486	68	68	2493	98	288	0.188
4	100	200	2500	94	47	2494	91	48	2477	114	171	0.194
5	100	200	2500	85	51	2469	85	51	2469	105	101	0.084
6	200	800	2500	66	51	2499	56	69	2495	58	1067	0.185
7	200	800	2500	63	53	2486	55	80	2480	55	671	0.177
8	200	800	2500	72	55	2491	60	92	2461	63	1195	0.188
9	200	800	2500	69	59	2499	63	101	2498	63	557	0.182
10	200	800	2500	56	60	2497	53	60	2498	62	365	0.123

جدول ۴-۴ مقایسه نتایج دو الگوریتم برای مساله مسیر مرکزی با طول محدود

Test	n	Edge NO.	Length P	Objective function		Last improv.iter		Cpu / Iter(sec)	
				GACPCPBL	GAPCPBL	GACPCPBL	GAPCPBL	GACPCPBL	GAPCPBL
1	100	200	5264	71	76	228	181	0.134	0.106
2	100	200	5313	64	57	136	145	0.154	0.135
3	100	200	5235	62	69	263	181	0.152	0.118
4	100	200	5294	72	81	239	222	0.127	0.116
5	100	200	5573	67	73	205	148	0.127	0.154
6	200	800	9854	14	23	443	397	0.408	0.418
7	200	800	10208	36	23	480	964	0.340	0.441
8	200	800	10062	34	24	563	501	0.434	0.167
9	200	800	10403	37	17	493	500	0.241	1.17
10	200	800	9882	34	22	510	837	0.362	0.394

Test	n	Edge NO.	Length P	Objective function		Last improv.iter		Cpu / Iter(sec)	
				GACPCPBL	GAPCPBL	GACPCPBL	GAPCPBL	GACPCPBL	GAPCPBL
1	100	200	2500	77	79	533	237	0.099	0.084
2	100	200	2500	94	96	164	366	0.106	0.094
3	100	200	2500	68	87	288	239	0.188	0.07
4	100	200	2500	91	96	171	237	0.194	0.487
5	100	200	2500	85	85	101	437	0.084	0.066
6	200	800	2500	56	61	1067	298	0.185	0.129
7	200	800	2500	55	57	671	917	0.177	0.137
8	200	800	2500	60	71	1195	238	0.188	0.201
9	200	800	2500	63	63	557	761	0.182	0.327
10	200	800	2500	53	51	365	594	0.123	0.139

۴-۵ بررسی نتایج دو الگوریتم برای مساله مسیر مرکزی

با توجه به جدول ۴-۲ و جواب اولیه این نتیجه حاصل شد که در حل مسائل ۴، ۵ و ۶ جواب نهایی الگوریتم ژنتیک بهبودی نسبت به جواب اولیه الگوریتم نداشته است. جواب نهایی الگوریتم ترکیبی در حل همه مسائل نسبت به جواب اولیه بهبود خوبی داشته است.

۴-۶ مقایسه نتایج دو الگوریتم برای مساله مسیر مرکزی با طول محدود

با توجه به نتایج الگوریتم ترکیبی برای مساله مسیر مرکزی با طول محدود در همه مسائل بهبود در جواب نهایی دیده شده است. در حل مساله مسیر مرکزی جواب‌های نهایی الگوریتم ترکیبی بهتر از جواب‌های نهایی الگوریتم ژنتیک بود. ولی در حل مساله مسیر مرکزی با طول محدود فقط در بعضی از مسائل جواب‌های الگوریتم ترکیبی بهتر از الگوریتم ژنتیک بوده است.

پیشنهادات

در این پایان‌نامه ما مساله مسیر مرکزی روی شبکه را مورد بررسی قرار دادیم. در فصل دوم الگوریتم اسلیتر که برای مکان‌یابی مسیر مرکزی روی درخت است، را مورد بررسی قرار دادیم و در فصل سوم و چهارم بعضی روش‌های ابتکاری برای حل مساله مورد توجه قرار گرفت. حال پیشنهاداتی برای تحقیقات آینده ارائه می‌کنیم.

ما دو روش ابتکاری برای مساله مسیر مرکزی با وزن مثبت مورد بررسی قرار دادیم: الگوریتم ژنتیک و الگوریتم ترکیبی ژنتیک و مورچه. می‌توان این روش‌ها را برای مساله با وزن مثبت و منفی نیز به کار برد. به عنوان زمینه‌ای برای تحقیق دیگر می‌توان روش‌های ابتکاری مانند روش مورچه، سرد کردن تدریجی[۱۴۲]، جستجوی پراکنده[۱۴۳] و بهینه‌سازی دسته جمعی[۱۴۴] را برای مساله مسیر مرکزی با وزن مثبت و منفی به کار برد. و نتایج را با نتایجی که ما به دست آوردیم مقایسه کرد.

ضمیمه

برنامه الگوریتم ژنتیک و الگوریتم ترکیبی با زبان برنامه‌نویسی C++ نوشته شده است. به دلیل زیادی متن برنامه‌ها متن چهار برنامه را در یک برنامه خلاصه کردیم و در این قسمت فقط متن همان برنامه را می‌آوریم. توضیحات در متن برنامه آورده شده است. ما برنامه‌ها را با کامپیوتر پنتیوم چهار که واحد پردازش مرکزی آن ۳۶۰۰ مگا هرتز و حافظه آن ۵۱۲ مگا بایت است، تست کردیم.

Abstract

---

142 Simulated annealing
143 Scater search
144 Swarm optimization

when the facility to be located is too large to be modelled as a point, extensive facility location models arise .Extensive facility location models on graphs deal with the location of a special type of subgraphs such as paths, trees or cycles and can be considered as extensions of classical point location models. Examples of problems in which structures instead of points are required include the location of pipelines, evacuation routes, mass transit routes or routing a highway through a road network.

Let a graph G = (V, E) be given. The path center problem asks to find a path P in G such that the maximum weighted distances from P to the vertices in V is minimized. Hakimi et al showed that the path center problem on general network is NP-hard. In the case that the network is a tree Hedetniemi et al and Slater presented linear time algorithms to solve the problem. For this case when the path has a specified size Minieka proposed a linear time algorithm. We develop a genetic algorithm and a combination of genetic and ant colony algorithm to approach the optimal solution of the path center problem on a graph. Computational results for some test problems are reported.

Keywords: location theory, genetic algorithm, ant colony, path center.

فهرست منابع و مراجع

[۱]- فتحعلی ج، (۱۳۸۳)، رساله دکترا:" مساله P– میانه با وزنهای مثبت و منفی"، دانشکده علوم ریاضی و آمار ،دانشگاه فردوسی مشهد.

[2] ReVelle C. S. (**1997**) "A perspective on location science" **Location Science, 5,** pp **3-13**

[3] Kuhn H. W. (**1967**) "On a pair of dual nonlinear programs, pp**37-54**, In:"**Nonlinear Programming**", J. Abadie (eds), North-Holand Publishers Co Amsterdam.

[4] Courant R. and Robbins H. (**1941**)"**What is mathematics**?" Oxford: Oxford University Press.

[5] Weber A. (**1929**) "**Uber den Standort der Industrient. Tubingen**" 1909; English Trans.: Theory of Location of Industries, (C.J.Friedrich; Ed. And trans), Chicago University Press, Chicago, Illinois.

[6] Hakimi S.L. (**1964**) "Optimum location of switching centers and the absolute centers and medians of a graph'' **Operation Research**, **12,** pp **450-459.**

[7] Handler G. Y. and Mirchandani P. B. (**1979**) "**Location on networks: Theory and Algorithms"**. MIT Press, Cambridge.

[8] Hamacher H. W. and Nickel S. (**1998**) "Classification of location models" **Location Science**, **6**, pp **229-242**.

[9] Tansel B. C. Francis R. L. and Lowe T. J. (**1983**) "Location on networks A survey'' **Management Science**, **29,** pp **482-511**.

[10] Krarup J. and Pruzan P. M. (**1983**)"The simple plant location problem: survey and synthesis'' **European journal of Operational Research**, **12**, pp **36-81**.

[11] Hansen P., Labbe M., Peeters D. and J. F. (**1987**) "Single facility location on networks" **Annals of Discrete Math**., **31**, pp **113-146.**

[12] Owen S. H. and Daskin M. S. **(1998)** "Strategic facility location: A review "**J. of. European journal of Operational Research**, **111,** pp **423-477.**

[13] Cappanera P. (**1999**) "**A survey on obnoxious facility location problems**" Technical report: Tr-99-11, Dipartimento di Informatica, Universita di Pisa.

[14] Scaparra M. P. and Scutella M. G. (2001) "**Facilities, locations, customers: building blocks of location models: A survey**" Thechnical Report: tr-01-18, University of Piza, Italy.

[15] Current J., Daskin M. and Schilling D., (**1929**) "**Discrete network location models**" Forthcoming as chapter 3 in: Facility Location Theory and Methods. Z. Drezner and H. Hamacher (eds.),.Chicago university press, Chicago, Illinois.

[16] Francis R., McGinnis Jr. L. F. and White J. A., (**1992**) "**Facility Layote and Location: An Analytical Approach**" Prentice Hall.

[17] Handler G. Y. (**1973**) "Minimax Location of a facility in an undirected tree graph" **Transportation Science**, **7**, pp **287-293**

[18] Goldman A. J. (**1971**) "Optimal center location in simple networks" **Transportation Science**, **5**, pp 212-221.

[19] Hakimi L. S. (**1965**) "Optimum distribution of switching centers in a communication network and some related graph theoretical problems" **Operation Research**.**13,** pp **462-475**.

[20] Shier R. D. (**1977**) "A Min-Max theorem for p-center problem on a tree" **Transportation Science**, **11,** pp **243-252**.

[21] Halfin S. (**1974**) "On finding the absolute and vertex centers of a tree with distances" **Transportation Science**, **8,** pp **75-77**.

[22] Handler G. Y. (**1985**) "Medi-center of a tree" **Transportation Science**, **19**, pp **246-260.**

[23] Kariv O. and Hakimi S. L. (**1979**) "**An algorithmic approach to network location problem**" II: The p-median, SIAM Journal of Applied Mathematics **37,** pp **539-560.**

[24] Kim T. U., Lowe T. J., Tamir A., Ward J. E. (**1996**) "On the location of a tree-shaped facility" **Networks**, **28,** pp **167-175.**

[25] Slater P. J. (**1982**) "Locating central paths in a graph" **Transportation Science**, **16,** pp **1-18**.

[26] Morgan C. A. and Slater J. P. (**1980**) "A linear algorithm for a core of a tree with a specified length "**Journal of algorithms**, **1,** pp **247-258**.

[27] Becker R. I. (**1990**) "Inductive algorithms on finite trees" **Quaestions Mathematicae**, **13,** pp **165-181**.

[28] Peng S. and Lo W. (**1996**) "Efficient algorithms for finding a core of a tree with a specified length" **Journal of algorithms**, **20,** pp **445-458**.

[29] Becker R. I. ,Chiang Y. ,lari I. ,Scozzari A. ,Storchi G. (**2002**) "Finding the l-core of a tree"**. Discrete Applied mathematics**, **118,** pp **25-42**.

[30] Alstrup S. ,Lauridsen P. W. ,Sommerlund P. and Thorup M. (**1997**) "Finding cores of limited length " Technical Report the IT university of Copenhagen,. A preliminary version appeared in Proceeding of the 5th International workshop on algorithms and Data structures, Lecture Notes in **Computer Science, 1272** pp.**45-54.**

[31] Wang B. F. (**2000**) "Efficient parallel algorithms for optimally locating a path and a tree of a specified length in a weighted tree network" **Journal of Algorithms.**, **34,** pp **90-108**.

Chia A. M. (**2005**) "Conditional location [32] Tamir A., Puerto J., Mesa J. A. and Rodriguez – of path and tree shaped facilities on trees**" Journal of Algorithms**, **56,** pp **50-75**.

[33] Wang B. F., Shan-Chyun K. and Yong-Hsian H. (**2002**) "The conditional location of a median path" Lecture Notes in **Computer Science, 2387**, pp **494-503**.

[34] Wang B. F., Lin T. Ch., Lin Ch. H. and Ku Sh. (**2008**) "Finding the conditional location of a median path on a tree" **Information and Computation 26**, pp **828-839**

[35] Hedetniemi S. M., Cockayne E. J. and Hedetniemi S. T. (**1981**), "Linear algorithms for finding the Jordan center and path center of a tree."**Transportation Science**, **15**, pp **98-114.**

[36] Minieka E. (**1985**) "The optimal location of a path in a tree network", **networks**, **15,** pp **309-321**.

[37] Peng S. and Lo W. (**1993**) "The optimal location of a structured facility in a tree network" **Paralled Algorithms and Application**, **2**, pp **43-60**.

[38] Hakimi S. L., Schmeichel E. F. and Labbe M. (**1993**) "On locating path- or tree-shaped facilities on network" **Networks**, **23,** pp **543-555**.

[39] Mesa J. A. and Boffey B. T. (**1996**) "A review of extensive facility location in networks **European Journal of Operational Research**, **95,** pp **592-603**.

[40] LabbeM., Laporte G. and Rodriguez Martin I. (**1998**), "**Path, tree cycle Location**" In T.G. and Logistics Kluwer. Crainic and G. Laporte (eds.), Fleet Management

[41] Tamir A and Lowe T. (**1992**) "The generalized P-forest problem on a tree network "**Networks**, **22** pp **217-230**.

[42] Averbakh I. and Berman O. (**1999**) "algorithms for path medi – center of a tree" **Computers and Operations Research**,**26** pp **1395-1409**.

[43] Becker I., Lari I. and Scozzari A. (**2007**) "Algorithms for central- median paths with bounded length on trees**" European journal of Operational Research**, **179,** pp **1208-1220**.

[44] Ronald I. Becker, Chiang Y-I., Lari I., an Andrea Scozzari, (**2001**) "The Cent-dian Path Problem on Tree Networks". pp **743-755**.

[45] Jordan C. (**1869**) "Sur les assemblages' des lignes" **Reine Angew .Math**, **70,** pp **185-190**.

[46] Alp O., Erkut E. and Drezner Z. (**2003**), "An efficient genetic algorithm for the p-median Problem." **Annals of Operations Research, 122**, pp **21-42**.

[47] Chiou Y. C. and Lan L. W. (**2001**), "Genetic clustring algorithms." **European Journal of Operational Research**, **135,** pp **413-427.**

[48] Fathali J. (**2006**), "A genetic algorithm for the p-median problem with pos/neg weights," **Applied Mathematics and Computation**, **183**, pp **1071-1083**.

[49] Stanimirovic Z., Kratica J. and Dugosija D. (**2007**), "Genetic algorithms for solving the discrete ordered median problem," **European Journal of Operational Research**, **182**, pp **983-1001**.

[50] Goldberg D.E. (**1989**), "**Genetic Algorithms in Search, Optimization and Machine Learning"**. Menlo Park, CA: Addison-Weseley.

[51] Reeves C.R. (**1993**), "Genetic Algorithms," **In C.R Reeves (Ed.), Modern Heuristic Techniques for Combinatorial Problems**". pp **151-196.**

[۵۳] طاهری ح، (۱۳۸۱)، پایان نامه کارشناسی ارشد :" روش های ابتکاری برای مساله تخصیص درجه ۲"، دانشکده علوم ریاضی و آمار، دانشگاه فردوسی مشهد،

[54] Ezzell (**1996**) "Genetic and geometric optimization of three-dimensional radiation therapy treatment planning" **Med.Phys**. **23** pp **293–305.**

[55] Langer M. et al (**1996**) "A generic genetic algorithm for generating beam weights" **Med. Phys**. **23** pp **965–967.**

[56] Wu X. and Zhu Y. (**2000**) "A mixed-encoding genetic algorithm with beam constraint for conformal radiotherapy treatment planning" **Med. Phys**. **27** pp **2508–2516.**

[57] http://www.p30lords.com/forum/showthread.php?p=162294

[58] http://betsa.blogfa.com/post-28.asp

[۵۹] **مجله پژوهش های اقتصادی ایران** ، شماره ۲: ص ۱۷۵.

[60] Beasley J.E. (**1990**), "OR-Library: distributing test problems by electronic Mail." **Journal of the Operational Research Society**, **41, 11,** pp **1069-1072.**

[61] Dorigo M., Maniezzo V. and Colorni A. (**1991**), "**Positive feedback as asearch strategy**" Technical report 91-016, Milan, Italy: Politecnico di Milano.

[62] Bonabeau E., Dorigo M., and Theraulaz G. (**1999**), "from **natural toartifical swarm inteligence**". New York: Oxford University Press.

[63] Dorigo M., and Di Caro G. (**1999**), "**The ant colony optimization metaheuristic**"In D. Optimization. Maidenhead, UK: Corne, M. Dorigo and F. Glover (Eds.), new ideas in McGrawHill.

[64] Dorigo M., and Stutzle T. (**2004**), "**Ant Colony Optimization**" MIT Press, Cambridge (MA).

for the [65] Fathali J., Kakhki H. T. and Burkard R. E. (**2006**), "An ant colony algorithm pos/neg weighted p-median problem," **Central European Journal of Operations Research**, **14**, pp **229-246**.

.