گزارش نهایی طرح پژوهشی:

روشهای ابتکاری برای حل مساله مسیر میانه روی شبکه

با کد ۲۳۰۲۱

مجری: جعفر فتحعلی

عضو هیئت علمی دانشکده ریاضی

دانشگاه صنعتی شاهرود

همکار اصلی: مهدی زعفرانیه

عضو هیئت علمی دانشکده ریاضی

دانشگاه تربیت معلم سبزوار

این طرح با استفاده ااز اعتبارات پژوهشی دانشگاه صنعتی شاهرود انجام شده است و تاریخهای تصویب و خاتمه آن به ترتیب ۱۳۸۷/۶/۴ و ۱۳۸۸/۲/۲۰ است.

چکیده

مساله پیدا کردن مسیر میانه و هسته شبکه از مسائلی در نظریه مکانیابی هستند که کاربردهای فراوانی در دنیای واقعی دارند. هسته شبکه مسیری در شبکه است به گونه ای که مجموع وزنی فاصله رئوس شبکه تا این مسیر کمترین مقدار شود. هسته ای که کمترین طول را دارد مسیر میانه نامیده می شود. در این طرح دو الگوریتم ابتکاری شبیه سازی تبریدی و روش مورچه برای پیدا کردن هسته شبکه ارائه کرده و نتایج این دو روش را با هم مقایسه می کنیم.

فهرست مندرجات

فصل ۱

مقدمه

۱—۱ تاریخچه

مکان‌یابی یک مبحث مهم در علم مدیریت و تخصیص منابع است. اولین مطالعات در مکان‌یابی در قرن هفدهم بوسیله فرما[۱] انجام شد. در دست‌نوشته‌های مشهور او این‌طور آمده، فرض کنید سه نقطه در یک صفحه داده شده است چطور می‌توان مکان نقطه چهارمی را پیدا کرد که مجموع فاصله آن از این سه نقطه کمترین مقدار ممکن باشد. این مساله در سال ۱۶۴۰ در ایتالیا توسط توریچلی[۲] حل شد وی ثابت کرد که نقطه بهینه محل برخورد دوایری است که مثلثهای متساوی الاضلاعی را که بر روی هر ضلع و در بیرون مثلث فعلی قرار دارند در برگرفته‌اند. بدین دلیل این نقطه را نقطه توریچلی می‌نامند.

مطالعات در علم مکان‌یابی مدرن رسما در سال ۱۹۰۹ توسط آلفرد وبر[۳] آغاز

---

[۱] Fermat

[۲] Torricelli

[۳] Alfred Weber

شد. وی مساله پیدا کردن مکان یک سرویس‌دهنده را که مجموع فاصله آن تا چند مشتری کمترین مقدار ممکن است را معرفی کرده و مورد بررسی قرار داد. بعد از این مطالعه ابتدایی تعداد زیادی از محققین سعی در بسط و توسعه مفهوم عرضه شده توسط وبر کردند. بعنوان مثال هاتلینگ[۴] [۲۲] مساله فروشنده بستنی در ساحل را مطرح کرد که هدف در آن جذب بیشترین تعداد مشتری در یک بازارِ ساحلی مشترک بین دو یا چند فروشنده است.

مکان‌یابی در سال ۱۹۶۴ تولد مجددی را توسط حکیمی[۵] [۱۵] تجربه کرد. حکیمی مساله مکان‌یابی بر روی شبکه را برای پیدا کردن مکان گشتهای پلیس در بزرگراهها ومناطق شهری مورد استفاده قرار داد. برای نیل به این هدف وی مساله مکان‌یابی را در یک شکل کلی‌تر مطرح کرد او فرض کرد که تعداد سرویس‌دهنده‌ها (گشتهای پلیس) بیشتر از یکی باشد بنابراین مشتریها از بین سرویس‌دهنده‌ها کسی را انتخاب می‌کنند که کمترین فاصله را با او داشته باشند. بدین ترتیب از اواسط دهه ۶۰ مکان‌یابی باپیشرفت شگرفی مواجه شد.

لازم به ذکر است که برخی از محققین یک سری طبقه‌بندیهایی از مدلهای مسائل مکان‌یابی ارائه دادند. اولین طبقه‌بندی مدلهای مختلف مکانیابی توسط هندلر[۶] و میرچندانی[۷] [۲۰] ارائه شد. پس از آن طبقه‌بندی‌های دیگری از جمله توسط ایسلت[۸] و لاپورت[۹] [۱۲] و هاماخر[۱۰] و نیکل[۱۱] [۱۸] انجام شد. همچنین بررسیهایی دیگری در این زمینه توسط کراروپ[۱۲] و پروزن[۱۳] [۲۵]، هانسن[۱۴] و همکاران [۲۱]،

[۴] Hotelling
[۵] Hakimi
[۶] Handler
[۷] Mirchandani
[۸] Eiselt
[۹] Laporte
[۱۰] Hamacher
[۱۱] Nickel
[۱۲] Krarup
[۱۳] Pruzan
[۱۴] Hansen

میرچندانی و فرانسیس[۱۵] [۳۱]، فرانسیس و همکاران [۱۳]، اُون[۱۶] و دسکین[۱۷] [۳۳]، اسکاپارا[۱۸] و اسکاتلا[۱۹] [۳۷] و درزنر[۲۰] و هاماخر [۹] انجام شده است. هال[۲۱] [۱۷] لیستی شامل بیش از ۳۴۰۰ مقاله در مورد مسائل مکانیابی را در یک سایت اینترنتی جمع آوری کرده است.

۱–۲ حالتهای مختلف مکان‌یابی

مساله مکان‌یابی شامل چندین حالت است که در این جا به بیان چند حالت از آن می‌پردازیم

۱) مساله p–میانه[۲۲]

۲) مساله p–مرکز[۲۳]

۳) مساله مسیر مرکزی [۲۴]

۴) مساله هسته یا مسیر میانه [۲۵]

---

[۱۵] Francis
[۱۶] Owen
[۱۷] Daskin
[۱۸] Scaparra
[۱۹] Scutella
[۲۰] Drezner
[۲۱] Hale
[۲۲] p-median
[۲۳] p-center
[۲۴] central path
[۲۵] core

در همه حالتهای فوق چهار معیار را در نظر می‌گیریم که در ذیل به بررسی آنها می‌پردازیم.

۱— مجموعه‌ای از مشتریها که می توانند نقاطی در صفحه یا رئوس یک گراف باشند.

۲— مجموعه نقاطی که داوطلب قرار گرفتن سرویس‌دهنده هستند و می‌توانند متناهی یا نامتناهی باشند.

۳— سرویس‌دهنده‌ای که گاهی دارای ظرفیت محدود است و قادر به خدمات رسانی به طورکامل نمی‌باشد.

۴— تابع هدفی که به دنبال بهینه کردن (مینیمم یا ماکزیمم) آن هستیم.

۱—۲—۱ مساله p—میانه

همانطور که چرچ و رول[۲۶] [۷] متذکر شده‌اند یک روش مهم برای سنجش بهینگی مکان یک سرویس‌دهنده بررسی مسافتی است که مشتریها برای دسترسی به آن طی می‌کنند. وقتی که این مسافت افزایش می‌یابد دسترسی به سرویس‌دهنده سخت می‌شود و بهینگی از دست می‌رود. این توجیه در حوزه عملکرد بسیاری از اماکن عمومی معقول به نظر می‌رسد بعنوان مثال کتابخانه‌های عمومی، مدارس و بیمارستانها باید در مکانهایی احداث گردند که دسترسی به آنها تا حد امکان تسهیل گردد. این بحث را می‌توان برای تعیین مکان اماکن ناخوشایند[۲۷] مانند فرودگاهها ، مراکز هسته‌ای یا کارخانه‌های زباله‌سوزی نیز تعمیم داد.

در ابتدا با توجه به ظرفیتهای موجود برای مشتریها، یک وزن به آنها اختصاص می‌دهیم که می‌تواند بیانگر جمعیتِ موجود یا حساسیت و اهمیت آنها از لحاظ محیطی و زیست‌شناسی و یا بسیاری پارامترهای دیگر باشد. در مدل‌بندی مساله p—میانه که توسط حکیمی [۱۵] معرفی شده است، از این معیار برای تبیین مساله استفاده شده است. مساله p—میانه را می‌توان به شکل زیر تعریف کرد. مکان P

[۲۶] Church and ReVelle

[۲۷] undesirable

سرویس‌دهنده را به نحوی پیدا کنید که مجموع مسافت وزنی بین مشتریها و این سرویس‌دهنده‌ها کمترین مقدار ممکن باشد.

مساله p—میانه برروی یک شبکه یک مساله NP—کامل[۲۸] است. بهر حال محدود کردن انتخاب P مکان از بین N راس تعداد حالتهای ممکن را به

$$\left(\begin{array}{c} N \\ P \end{array} \right) = \frac{N!}{P!(N-P)!}$$

محدود می‌کند. اگر N و P خیلی بزرگ نباشند می‌توان به روش شمارشی جواب بهینه را پیدا کرد. برای یک P مشخص، این مساله یک مساله NP—سخت[۲۹] است. چنین مرتبه محاسباتی باعث بوجود آمدن الگوریتمهای کارایی برای حل مساله p—میانه شده است. با توجه به آنکه مدل مساله p—میانه یک مدل عدد صحیح است روشهای ابتکاری[۳۰] نیز برای بدست آوردن جواب مورد استفاده قرار گرفته‌اند.

در سال ۱۹۹۵ گویش[۳۱] و اسریدار[۳۲] مساله را برای حالت $p = ۲$ بررسی کرده و الگوریتمی با پیچیدگی زمانی $o(nlogn)$ ارائه دادند [۱۴].

۱—۲—۲ مساله p—مرکز

در مساله p—مرکز طبق تعریف حکیمی [۱۵] مکان P سرویس‌دهنده را به نحوی پیدا می کنیم که دورترین مسافت وزنی بین مشتریها و این سرویس‌دهنده‌ها کمترین مقدار ممکن باشد. کریو و حکیمی [۲۳] الگوریتمی با پیچیدگی زمانی $O(|E|nlogn)$ برای پیدا کردن —مرکز روی شبکه ارائه کرده اند. آنها همچنین در حالتی که وزن همه رئوس یکسان است برای مساله —مرکز روی شبکه الگوریتمی با پیچیدگی زمانی $O(|E|n)$ ارائه کرده اند.

---

[۲۸] NP-complete
[۲۹] NP-hard
[۳۰] heuristic
[۳۱] Gavish
[۳۲] Sridhar

مساله p—مرکز در حالت کلی یک مساله NP—سخت است و از روشهای ابتکاری برای حل آن استفاده می شود.

۱—۲—۳ مسیرمرکزی

گاهی در مسائل مکانیابی به مواردی برخورد می‌کنیم که مشتریهایی که در نقاط دوردست هستند و دسترسی آنها به سرویس‌دهنده مشکل‌تر است مورد توجه قرار گرفته‌اند و ما به دنبال تسهیل دسترسی اینگونه مشتریها به مسیر هستیم اگر در این مساله مجموعه مشتریها را رئوس یک گراف و مقدار تقاضای هر مشتری را وزن راس مربوط به آن در نظر بگیریم هدف پیدا کردن مسیری است که بیشترین فاصله راسهای موجود تا آن کمترین مقدار ممکن باشد مدل این مساله به صورت زیر است:

$$G(P) = \min_{P}(\max_{1 \le i \le n} w_i d(v_i, P)) \qquad (1)$$

کوکاینه[۳۳] و همکاران در سال ۱۹۸۱ این مساله را بررسی کردند و الگوریتم خطی روی درخت برای آن ارائه دادند. در سال ۱۹۸۲نیز این مساله بر روی شبکه توسط اسلیتر [۳۸] بررسی شد و الگوریتم خطی برای آن ارائه شد. در سال ۱۹۸۵ مینیکا به حل این مساله در حالت خاص که طول مسیر از اندازه مشخصی مانند l بیشتر نباشد پرداخت که در زیر به معرفی آن می‌پردازیم [۲۹].

یکی دیگر از مسائل مکان‌یابی مساله هسته یا مسیر میانه است که بحث اصلی ما در این طرح می‌باشد و در فصل بعد به تفصیل به آن خواهیم پرداخت.

[۳۳] Cocckayne

فصل ۲

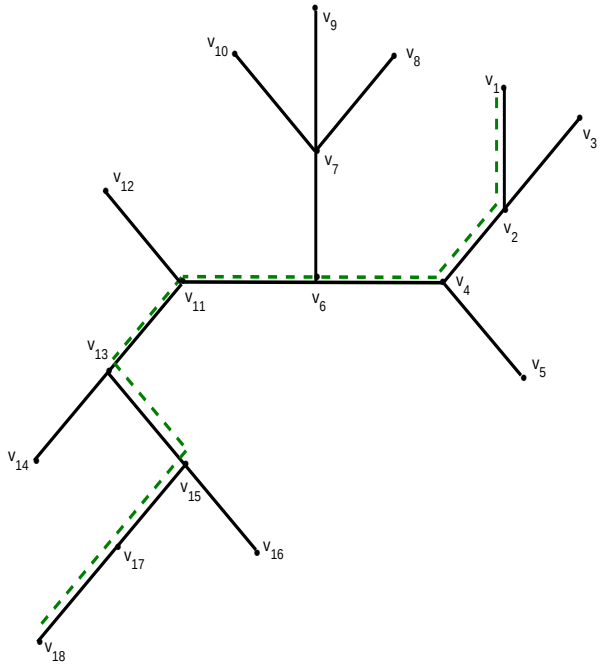
مسیر میانه یا هسته

مسأله مسیر میانه یکی ازمسائلی در نظریه مکانیابی است که کاربردهای فراوانی در دنیای واقعی دارد. فرض کنید شبکه $N = (V, E)$ با n راس داده شده باشد. هسته شبکه مسیری مانند P در شبکه است به گونه ای که مجموع وزنی فاصله رئوس شبکه تا این مسیر کمترین مقدار شود. هسته ای که کمترین طول را دارد مسیر میانه نامیده می شود. فاصله هر راس $v \in V$ از مجموعه P به صورت فاصله v تا نزدیکترین نقطه در P تعریف می شود. معمولا به هر راس v_i یک وزن w_i نیز تخصیص می‌یابد. که در این حالت هدف در مساله پیدا کردن هسته کمینه کردن مجموع وزنی فاصله ها خواهد بود. یعنی:

$$F(P) = \sum_{v_i \in N} w_i d(P, v_i)$$

که در آن $d(P, v) = \min_{x_i \in P} d(x_i, v)$ و $d(x, y)$ فاصله بین دو نقطهٔ x و y روی شبکه می‌باشد.

مورگان و اسلیتر [۳۲] الگوریتمی خطی برای پیدا کردن هسته درخت ارائه کرده اند. حکیمی و همکاران [۱۶] نشان دادند که این مساله روی شبکه یک مساله NP—سخت است. حالت کلی تر این مساله پیدا کردن l—هسته می باشد که در آن هدف پیدا کردن l مسیر مجزا در شبکه است به قسمی که مجموع وزنی فاصله رئوس



شکل ۲—۱: هسته

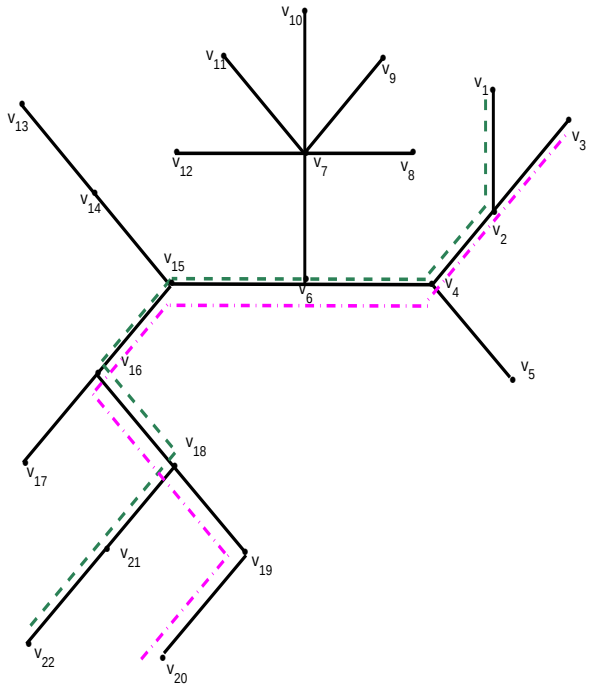
شبکه تانزدیکترین مسیر از این l مسیر کمترین مقدار شود. به ازای $l = ۲$ بکر و پرل [۳] برای مساله روی درخت الگوریتمی با پیچیدگی زمانی $O(nd(T))$ ارائه کرده‌اند که در آن $d(T)$ قطر درخت یعنی طول بلندترین مسیر در درخت می باشد. وانگ [۴۲] پیچیدگی زمانی الگوریتم را بهبود بخشید و یک الگوریتم خطی ارائه کرد.

۲—۱ خواص مساله هسته

اگر راس ابتدائی $v_۱$ و انتهائی v_n باشد هسته را با $P_{v_۱,v_n}$ نمایش می‌دهیم.

مثال ۱.۲ درشکل ۲—۱ هسته درخت $P_{v_۱,v_{۱۸}}$ است که مقدار تابع $F(P) = \sum_{i=۱}^{n} w_i d(v_i, P)$ به ازاء آن برابر با ۱۲ است. چون وزن رئوس و طول یالها برابر یک است فاصله دو راس را برابر با تعداد یالهای بین آن دو در نظر می‌گیریم. فاصله راسهای متعلق به مسیر برابر صفر دارند.

$$F(P) = ۱ + ۱ + ۱ + ۱ + ۱ + ۱ + ۲ + ۲ + ۲ = ۱۲$$



شکل ۲—۲: هسته‌های متفاوت

لم ۱.۲ [۳۲] رئوس ابتدایی و انتهایی هسته درخت برگ هستند.

هسته لزوما یکتا نیست یعنی امکان دارد دو مسیر متفاوت وجود داشته باشد در حالیکه مجموع فاصله رئوس از هر دو آنها به یک اندازه باشد.

مثال ۲.۲ در شکل ۲—۲ دو هسته متفاوت $P_{v_{۳},v_{۲۰}}, P_{v_{۱},v_{۲۲}}$ نشان داده شده است که مجموع فاصله رئوس از هر دو آنها برابر ۱۵ است.

۲—۲ هسته مقید

در بعضی موارد هدف پیدا کردن هسته‌ای است که طول آن مقدار مشخصی باشد چنین هسته‌ای را **هسته مقید**[۱] گویند. این مساله هنگامی که سرویس‌دهنده نمی‌تواند

[۱] conditional core

تمام شبکه را پوشش دهد یا امکانات کافی برای ایجاد یک سرویس‌دهنده بهینه با طول ماکزیمم وجود ندارد بوجود می‌آید. مدل این مساله به صورت زیر نوشته می‌شود.

$$F(P) = min \sum_{i=1}^{i=n} w_i d(v_i, P)$$
$$s.\ t. \qquad (1)$$
$$\|P\| \leq l$$

ادوارد مینیکا در سال ۱۹۸۵ این مساله را بررسی کرد[۲۹] و روشی را برای پیدا کردن هسته مقید پیشنهاد داد که دارای مرتبه زمانی $o(n^3)$ است. او همچنین در سال ۱۹۸۳ به همراه پاتل[۲] به این نتیجه رسید که لزومی ندارد که هسته مقید ۱—میانه را قطع کند [۳۰].

در سال ۱۹۹۶ پنگ و لو مساله هسته مقید را بررسی نمودند[۳۵] و الگوریتمی با پیچیدگی زمانی $o(nlogn)$ به روی درخت وزن‌داری که طول یالهای آن ۱ و تعداد راسهای آن n است پیاده سازی کردند.

بکر و همکاران در سال ۲۰۰۲ الگوریتمی برای پیدا کردن l-هسته ارائه دادند[۲] که برای درخت بدون وزن دارای مرتبه محاسباتی $o(nl)$ و برای حالت وزن‌دار $o(nlog^2 n)$ است.

آلستراپ و همکاران نیز مساله هسته مقید را برای درخت با طول یال دلخواه بررسی نموده و الگوریتمی با پیچیدگی زمانی $o(nl)$ ارائه دادند که در آن l طول هسته و n تعداد راسهای درخت است.

مینیکا و پاتل در سال ۱۹۸۳ همچنین رابطه بین ۱—میانه و هسته مقید را بررسی کردند[۳۰]. فرض کنید m، ۱—میانه درخت T باشد و P_l مجموعه مسیرهای با طول l باشد که شامل ۱—میانه نیستند و Q_l مجموعه مسیرهایی با طول طول l باشد که ۱—میانه را شامل می‌شوند. واضح است که هسته مقید با طول l متعلق به مجموعه $P_l \bigcup Q_l$ است.

---

[۲] Patel

فصل ۳

الگوریتمهای شبیه سازی تبریدی و مورچه برای پیدا کردن هسته

در این طرح دو الگوریتم ابتکاری شبیه سازی تبریدی (SA) و روش مورچه (AC) برای پیدا کردن هسته شبکه ارائه کرده و نتایج این دو روش را با هم مقایسه می کنیم.

روش مورچه روشی نسبتا جدید برای حل مسائل بهینه سازی می باشد که اولین بار توسط دریگو، مانیزو و کلرنی [۱۰] برای بعضی مسائل مشکل بهینه سازی از قبیل مساله فروشنده دوره گرد و مساله تخصیص درجه دو مورد استفاده قرار گرفت. الگوریتم مورچه همچنین برای حل بعضی دیگر از مسائل بهینه سازی گسسته از قبیل مساله رنگ آمیزی گراف، مساله مرتب سازی تصادفی و حرکت در شبکه های ارتباطی مورد استفاده قرار گرفته است (برای اطلاعات بیشتر به [۱۱] مراجعه کنید).

روش شبیه سازی تبریدی برای حل مسائل بهینه سازی توسط کیرکپاتریک و همکاران [۲۴] معرفی شد. ایده این روش از یک قاعده فیزیکی به نام قاعده متروپلیس الهام گرفته شده است که در آن اجسام رابه صورت تدریجی سرد می کنند تا به دمای پائین برسند. از این روش در حل مسائلی از قبیل مساله فروشنده دوره گرد، مسائل مکانیابی و مساله مسیر حرکت وسایل حمل و نقل استفاده شده است.

۳–۱ الگوریتم مورچه

روش مورچه برگرفته از مشاهده رفتار مورچه های واقعی می باشد. مورچه ها تقریبا نابینا هستند. آنها حین حرکت بین لانه و غذا ماده ای به نام فرمون از خود به جای میگذارند. در بین تمام مسیرهای بین لانه و غذا یک مورچه مسیری را که دارای فرمون بیشتری است با احتمال بیشتری انتخاب می کند. دیگر مورچه ها هم این روش را تکرار می کنند و بعد از مدتی، کوتاهترین مسیر دارای بیشترین فرمون خواهد شد.

در این بخش یک الگوریتم مورچه برای پیدا کردن هسته گراف ارائه می کنیم. ساختار این الگوریتم به صورت زیر است.

در هر الگوریتم مورچه، در هر مرحله تعدادی مورچه در نظر گرفته می شوند تا جوابهایی را تولید کنند که از آنها در فرمون ریزی مراحل بعد استفاده می شود. ما در ابتدا به تمام یالهای شبکه یک مقدار فرمون ثابت اختصاص می دهیم و از n مورچه استفاده می کنیم. برای تولید جوابهای مربوط به این مورچه ها از هر یک از n راس شبکه شروع کرده و به طور تصادفی به راس مجاور می رویم و این عمل را تا زمانی که دیگر نتوانیم راس جدیدی به مسیر اضافه کنیم ادامه می دهیم. از روال $Initial$ می توان برای تولید این جوابها استفاده کرد.

Procedure $[Initial]$

Set $S := \emptyset$.

For each vertex $v \in V$ **do the following**

1. $P(v) := v$, $EndP := v$.
2. **While** $Adj := \{u \in V | u \notin P(v)$, **u is adjacent to** $EndP\}$ **is not empty do the following**
   (a) **Select randomly a vertex** $u \in Adj$.

(b) **Add** u **to** $P(v)$.

(c) $EndP := u$.

End While.

3. $S := S \cup \{P(v)\}$

End For.

در تکرارهای بعد از تکرار اول به جای n مورچه از $m = \lceil \frac{n}{\alpha} \rceil$ مورچه ا ستفاده می کنیم که در آن $\alpha \geq ۱$ یک پارامتر است. روند تولید مسیرهای مربوط به این m مورچه مانند قبل است تنها با این تفاوت که‌ساخت هر یک از آنها از یالی شروع می شود که در بین m راس با بیشترین فرمون قرار دارند.

برای بروزرسانی فرمونها از قاعده نیمه—MMAS استفاده می کنیم. قاعده MMAS روشی است که اولین بار توسط بلن‌هایمر و همکاران [۵] برای حل مساله فروشنده دوره گرد استفاده شده است. در قاعده نیمه—MMAS ما جوابهای $P_۱, P_۲, ..., P_m$ را بر اساس مقدار تابع هدف آنها مرتب می کنیم. سپس به هر یال e_{uv} فرمون $\tau(e_{uv})$ را به صورت زیر اختصاص می دهیم.

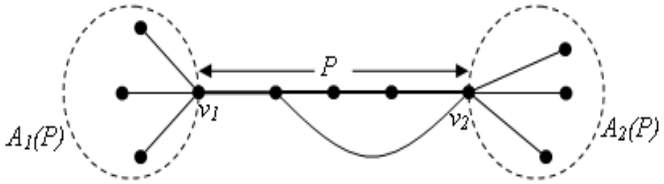
$$\tau(e_{uv}) = \rho\tau(e_{uv}) + \sum_{i=۱}^{\lceil \frac{m}{۲} \rceil - ۱} (\lceil \frac{m}{۲} \rceil - i) g_i(e_{uv}) \qquad (۱)$$

که در آن

$$g_i(e_{uv}) = \begin{cases} \frac{۱}{F(P_i)} & if\ e_{uv} \in P_i \\ ۰ & otherwise \end{cases} \quad i = ۱, ..., \lceil \frac{m}{۲} \rceil - ۱ \qquad (۲)$$

و $۰ < \rho < ۱$ پارامتری است که برای تبخیر فرمون استفاده می شود.

فرض کنید $v_۱$ و $v_۲$ دو راس انتهایی مسیر P باشد. دو مجموعه $A_۱(P)$ و $A_۲(P)$ به ترتیب مجموعه های راسهایی باشند که با $v_۱$ و $v_۲$ مجاورند و می توان به مسیر P اضافه کرد (شکل ۳—۱ را ببنید)، یعنی



شکل ۳—۱: مجموعه های $A_۲(P)$ و $A_۱(P)$

$$A_i(P) = \{e_{v_i u} \in E \mid u \notin P\} \quad i = ۱, ۲.$$

و به ازای $i = ۱, ۲$ احتمال اینکه یال $e_{v_i w}$ برای اضافه شدن به مسیر انتخاب شود به صورت زیر است.

$$p(e_{v_i w}) = \frac{\tau(e_{v_i w})}{\sum_{u \in A_i(P)} \tau(e_{v_i u})}.$$

با توجه به مطالبی که بیان شد الگوریتم پیشنهادی ما به صورت زیر می باشد.

Algorithm[ACPMP]:

Initial step:

1. $\tau(e) := C$, **for each** $e \in E$.
2. ρ **a number in** $(۰, ۱)$, $\alpha \in \{۱, ..., n\}$.
3. **Do** *Initial*.
4. $F_{bes} = \min_{P \in S} F(P)$.

Iteration step:

While stop condition is not met do the following

1. $\tau(e) := \rho\tau(e)$ **for each** $e \in E$.

2. **Let** $P_1, ..., P_m$ **be the ranked members of** S **with respect to their objective functions.**

3. **If** $F(P_1) = 0$ **then stop,** P_1 **is optimal and** $F_{bes} = 0$**.**

4. **For** $i = 1$ **to** $\lceil \frac{m}{2} \rceil - 1$ **do the following**

   (a) $\tau(e) := \tau(e) + (\lceil \frac{m}{2} \rceil - i)\frac{1}{F(P_i)}$ **for each edge** $e \in P_i$

   End for

5. $S := \emptyset$.

6. $m := \lceil \frac{n}{\alpha} \rceil$.

7. **Let** $SE := \{e_1, ..., e_m\}$ **be the set of edges with strongest pheromone amounts.**

8. **For each** $e \in SE$ **do the following**

   (a) $P := e$.

   (b) **Let** v_1, v_2 **be the two end vertices of** P **and** $A_i(P) := \{e_{v_i u} \in E | u \notin P\}$ **for** $i = 1, 2$**.**

   (c) **While** $A_1(P)$ **or** $A_2(P)$ **is not empty do the following**

      i. **For** $i = 1, 2$ **do the following**

         A. **if** $A_i(P) \neq \emptyset$ **add an edge** $e_{v_i w} \in A_i(P)$ **with probability**

$$p(e_{v_i w}) := \frac{\tau(e_{v_i w})}{\sum_{u \in A_i(P)} \tau(e_{v_i u})}$$

to the path P.

B. **Update** v_i.

C. **Update** $A_1(P)$ **and** $A_2(P)$.

End for.

End while.

(d) $S := S \cup \{P\}$

End for.

9. $F_{cur} = \min_{P \in S} F(P)$.

10. If $F_{cur} < F_{bes}$ **then** $F_{bes} = F_{cur}$.

End while.

شرط توقف می تواند براساس یکی از محدودیتهای حداکثر تعداد تکرار، حداکثر تعداد تکرار بعد از آخرین بهبود و یا حداکثر زمان مجاز CPU باشد.

۳–۲ الگوریتم شبیه سازی تدریجی

شبیه سازی تدریجی یک الگوریتم فرا ابتکاری برای حل مسائل بهینه سازی است. این الگوریتم از یک جواب اولیه و یک پارمتر اولیه T که **درجه حرارت** نامیده می شود شروع کرده و سعی در بهبود بخشیدن جواب می کند. در هر تکرار یک جواب جدید تولید شده و با جواب جاری مقایسه می شود، اگر از جواب جاری بهتر بود انتخاب می شود در غیر اینصورت انتخاب آن به طور احتمالی بستگی به تفاوت بین درجه حرارت فعلی و مقدار تابع هدف جواب به دست آمده دارد. پارامتر T در حین اجرای الگوریتم به تدریج کاهش می یابد.

الگوریتمی که ما برای پیدا کردن هسته گراف ارائه کرده ایم از یک جواب اولیه شروع می کند که این جواب اولیه بهترین جواب از بین n جوابی است که به وسیله روال $Initial$، که در بخش ۳—۱ بیان شد، تولید می شوند. فرض کنید P_c جواب فعلی و n_c تعداد راسهای P_c باشد. یک همسایگی از P_c به صورت زیر تعریف می کنیم

$$N(P_c) = \{P|\ P\ is\ a\ path,\ P \cap P_c \in S(P_c),\ |P \cap P_c| = \lceil \frac{n_c}{۲} \rceil\}$$

که در آن $S(P_c)$ مجموعه همه زیر مسیرهای P_c است. برای تولید یک همسایه جدید مانند $P_{new} \in N(P_c)$ برای $S(P_c)$، ابتدا یک زیرمسیر از $S(P_c)$ که دارای $\lceil \frac{n_c}{۳} \rceil$ راس است انتخاب می شود سپس این زیرمسیر با اضافه کردن یالهایی توسعه می یابد تا زمانیکه نتوان یال دیگری اضافه کرد. یعنی یالی موجود نباشد که با اضافه کردن آن باز هم مسیر داشته باشیم. پس از تولید این جواب جدید، P_{new} با احتمال زیر جایگرین P_c می شود

$$p(P_{new}) = \begin{cases} ۱ & if\ F(P_{new}) < F(P_c) \\ \exp(\frac{F(P_c)-F(P_{new})}{T}) & otherwise. \end{cases} \qquad (۳)$$

مقدار اولیه درجه حرارت T در ابتدا برابر تفاضل دو عدد که به طور تصادفی تولید می شود، قرار داده می شود و سپس به ازای هر β بار تغییر P_c، T با ضریب ρ کاهش می یابد، که $\beta \in N$ و $\rho \in (۰, ۱)$ پارامتر هستند.

با توجه به توضیحات داده شده الگوریتم پیشنهادی به صورت زیر خواهد بود.

Algorithm[SAPMP]:

Initial step:

1. ρ **a number in** $(۰, ۱)$.

2. $counter := ۱$, β **a natural number.**

3. **Do** *Initial*.

4. **Let** P_c**=arg**$\min_{P \in S} F(P)$.

5. $F_{bes} = F(P_c)$.

6. **Choose** $P_۱,\ P_۲$ **from** S**, in random.**

7. $T = |F(P_۱) - F(P_۲)|$.

Iteration step:

While stop condition is not met do the following:

1. $n_c := |V(P_c)|$.

2. **Choose a subpath** P **of** P_c **with** $|V(P)| = \lceil \frac{n_c}{۲} \rceil$.

3. **Let** $v_۱, v_۲$ **be the two end vertices of** P **and** $A_i(P) := \{e_{v_i u} \in E | u \notin P\}$ **for** $i = ۱, ۲$.

4. **While** $A_۱(P)$ **or** $A_۲(P)$ **is not empty do the following**

   (a) **For** $i = ۱, ۲$ **do the following**

      i. **if** $A_i(P) \neq \emptyset$ **add an edge from** $A_i(P)$ **to the path** P **in random.**

      ii. **Update** v_i.

      iii. **Update** $A_۱(P),\ A_۲(P)$.

   End For.

End While.

5. $P_{new} := P$.

6. **If** $F(P_{new}) = ۰$ **then stop,** P_{new} **is optimal and** $F_{bes} = ۰$.

7. **If** $F(P_{new}) < F_{bes}$ **then** $F_{bes} := F(P_{new})$.

8. **Replace** P_c **by** P_{new} **with probability**

$$p(P_{new}) = \begin{cases} ۱ & if\ F(P_{new}) < F(P_c) \\ \exp(\frac{F(P_c)-F(P_{new})}{T}) & otherwise. \end{cases}$$

9. **If** $P_c = P_{new}$ **then** $counter = counter + ۱$.

10. **If** $counter > \beta$ **then**

    (a) $T := \rho T$.

    (b) $counter := ۱$.

    End If.

End While.

شرط توقف می تواند مانند شرط توقف الگوریتم مورچه در نظر گرفته شود.

۳–۳ نتایج محاسباتی

برای مقایسه روشهای ابتکاری مطرح شده در بخشهای قبل ماز آنها برای پیدا کردن هسته شبکه هایی نمونه‌ای که در [۴] برای مساله p–میانه ارائه شده استفاده کردیم. الگوریتمها با استفاده از نرم افزار مطلب نوشته شده است و برای هر مساله ۱۰ مرتبه اجرا شده است و میانگین آنها گزارش شده است. همه اجراها روی کامپیوتر با پردازشگر پنتیوم ۴ همراه با یک گیگابایت حافظه و با سرعت دو مگاهرتز در ثانیه انجام شده است.

جدول ۳—۱ نتایج محاسباتی حاصل از الگوریتمهای ACPMP و SAPMP را نشان می دهد. جواب بهینه تمام مسائل برابر صفر است زیرا شبکه ها به گونه ای ساخته شده اند که هر یک حتما شامل یک دور همیلتونی[1] هستند. آخرین ستون جدول نشان دهنده زمان انجام هر الگوریتم در هر تکرار است.

شرط توقف برای ACPMP و SAPMP به ترتیب رسیدن تعداد تکرارها به n و $۳۰n$ می باشد. توجه کنید که اگرچه زمان اجرای هر تکرار برای SAPMP کمتر از ACPMP است اما در تمام موارد ازمان کل اجرای لگوریتم ACPMP کمتر از SAPMP است.

ما الگوریتم ACPMP را برای تمام مسائل به ازای $C = ۰/۰۱$ (فرمون اولیه)، $\rho = ۰.۰۲, ۰.۰۳, ..., ۰.۰۹, ۰.۱, ۰.۲, ۰.۵, ۰.۸$ (ضریب تبخیر) و $\alpha = ۱, ۱۰, \frac{n}{۱۰}$ که $m = \lceil \frac{n}{\alpha} \rceil$ تعداد مورچه ها در هر تکرار است، اجرا کردیم. بهترین جوابها به ازای $۱۰ \leq m \leq ۸$ و مقادیری از ρ که در جدول ۳—۲ آمده است، به دست آمد.

وقتی که m کمتر از ۸ است هر چه به۱ ۱ نزدیک شویم تعداد مورچه ها در هر تکرار کم می شود و زمان اجرا کاهش می یابد ولی مقدار تابع هدف به دست آمده افزایش می یابد. از طرف دیگر وقتی m از ۱۰ بیشتر است با افزایش آن و حرکت به سمت n زمان اجرا افزایش می یابد و نیز به دلیل تجمع فرمون روی بعضی مسیرهای خاص ممکن است الگوریتم در جوابهای بهینه موضعی گیر کند.

الگوریتم SAPMP برای تمام مسائل به ازای $\rho = ۰.۱, ۰.۲, \cdots, ۰.۹$ (ضریب کاهش درجه حرارت) و $\beta = \frac{n}{۵}, \frac{n}{۱۰}, \frac{n}{۲۰}, \frac{n}{۴۰}$ (تعداد تکرارهای داخلی) اجرا شد. بهترین جوابها به ازای $\beta = \frac{n}{۲۰}$ و $\rho = ۰.۵$ به دست آمد.

نمودار جوابهای مساله $pmed۲۰$ در شکلهای ۳—۲، ۳—۳ و۳—۴ آمده است. شکلهای ۳—۲ و ۳—۳ نشان دهنده مقادیر تابع هدف نسبت به تعداد تکرارها به ترتیب برای الگوریتمهای ACPMP و SAPMP به ازای ۴۰۰ تکرار می باشد.

[1] دور همیلتونی دوری است که شامل تمام رئوس گراف است.

			Objective function		CPU/Iter (sec)	
Test #	n	Edge number	ACPMP	SAPMP	ACPMP	SAPMP
pmed1	100	200	0	0	0.072	0.012
pmed2	100	200	0	25	0.075	0.013
pmed3	100	200	0	4	0.080	0.015
pmed4	100	200	0	41	0.081	0.013
pmed5	100	200	0	8	0.073	0.014
pmed6	200	800	0	16	0.315	0.058
pmed7	200	800	0	36	0.321	0.057
pmed8	200	800	0	41	0.320	0.055
pmed9	200	800	0	29	0.322	0.054
pmed10	200	800	0	26	0.325	0.056
pmed11	300	1800	0	39	0.863	0.138
pmed12	300	1800	0	40	0.867	0.144
pmed13	300	1800	0	36	0.857	0.139
pmed14	300	1800	0	47	0.853	0.137
pmed15	300	1800	0	48	0.861	0.136
pmed16	400	3200	0	45	1.804	0.254
pmed17	400	3200	0	51	1.803	0.265
pmed18	400	3200	0	54	1.780	0.261
pmed19	400	3200	0	56	1.795	0.260
pmed20	400	3200	0	44	1.805	0.258
pmed21	500	5000	24	83	3.262	0.415
pmed22	500	5000	15	59	3.268	0.393
pmed23	500	5000	28	52	3.261	0.397
pmed24	500	5000	18	90	3.281	0.395
pmed25	500	5000	17	84	3.278	0.389
pmed26	600	7200	12	77	5.413	0.613
pmed27	600	7200	9	70	5.345	0.603
pmed28	600	7200	25	74	5.362	0.620
pmed29	600	7200	35	90	5.376	0.621
pmed30	600	9800	34	83	5.432	0.634
pmed31	700	9800	21	63	8.344	0.853
pmed32	700	9800	25	65	8.269	0.864
pmed33	700	9800	18	63	8.300	0.823
pmed34	700	9800	28	95	8.260	0.871
pmed35	800	12800	29	68	12.246	1.127
pmed36	800	12800	36	73	12.169	1.120
pmed37	800	12800	32	53	12.208	1.125
pmed38	900	16200	24	52	17.012	1.432
pmed39	900	16200	30	48	16.826	1.417
pmed40	900	16200	31	59	17.026	1.446

جدول ۳—۱: نتایج الگوریتمهای $SAPMP$ و $ACPMP$

الگوریتم ACPMP بعد از ۲۸۵ تکرار به جواب بهینه رسیده است و الگوریتم قبل از رسیدن به شرط توقف، یعنی ۴۰۰ تکرار، به جواب بهینه رسیده است درحالیکه الگوریتم SAPMP بعد از ۴۰۰ تکرار نتوانسته است جواب خوبی پیدا کند. شکل ۳—۴ نشان دهنده تغییرات تابع هدف مساله $pmed$۲۰ در ۱۲۰۰۰ تکرار به ازای جوابهایی است که درالگوریتم SAPMP مورد قبول واقع شده اند، یعنی مقادیری از تابع هدف در تکرارهایی که جواب جدید تولید شده یعنی P_{new} با توجه به شرایط گفته شده در الگوریتم به جای جواب جاری یعنی P_c قرار می گیرد.

توجه کنید که در اکثر مسائل نمونه ای که برای مساله فروشنده دوره گرد[۲] (TSP) استفاده می شود، شبکه مورد استفاده یک شبکه کامل[۳] است که اینگونه شبکه‌ها دارای تعداد زیادی دور همیلتونی هستند که هر کدام از آنها هسته شبکه با مقدار تابع هدف صفر هستند. ما از اینگونه مسائل نمونه برای بررسی الگوریتمهای پیشنهادیمان استفاده نکردیم زیرا در هر دو الگوریتم روال $Initial$ در همان گام اولیه یک جواب بهینه برای مساله پیدا می کند و دیگر نمی توان به وسیله این مسائل کارایی دو الگوریتم را بررسی کرد.

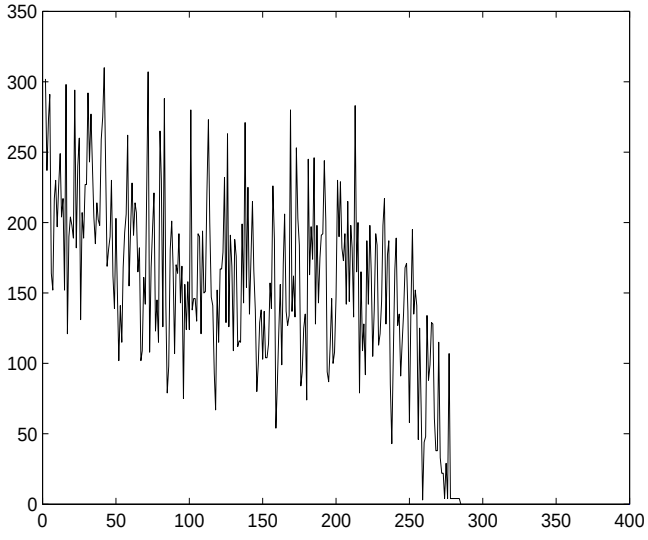
۳—۴ خلاصه و نتیجه گیری

در این طرح ما دو الگوریتم ابتکاری شبیه سازی تبریدی و روش مورچه برای پیدا کردن هسته شبکه ارائه کردیم و نتایج حاصل از این دو روش را برای ۴۰ مساله به دست آوردیم. مقایسه نتایج نشان می دهد که روش مورچه عملکرد بهتری نسبت به روش شبیه سازی تبریدی داشته است.

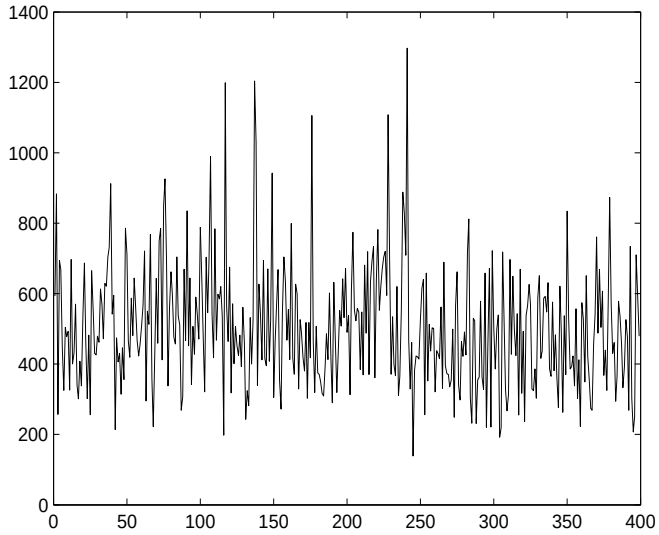
[۲] مساله فروشنده دوره گرد عبارت از پیدا کردن یک دور همیلتونی با کمترین طول است.
[۳] شبکه کامل شبکه ای است که در آن بین هر دو راس یک یال وجود دارد.

Test #	ρ
pmed1, ..., pmed5	0.5
pmed6, ..., pmed10	0.2
pmed11, ..., pmed15	0.2
pmed16, ..., pmed20	0.1
pmed21, ..., pmed25	0.1
pmed26, ..., pmed30	0.07
pmed31, ..., pmed34	0.07
pmed35, ..., pmed37	0.06
pmed38, ..., pmed40	0.06

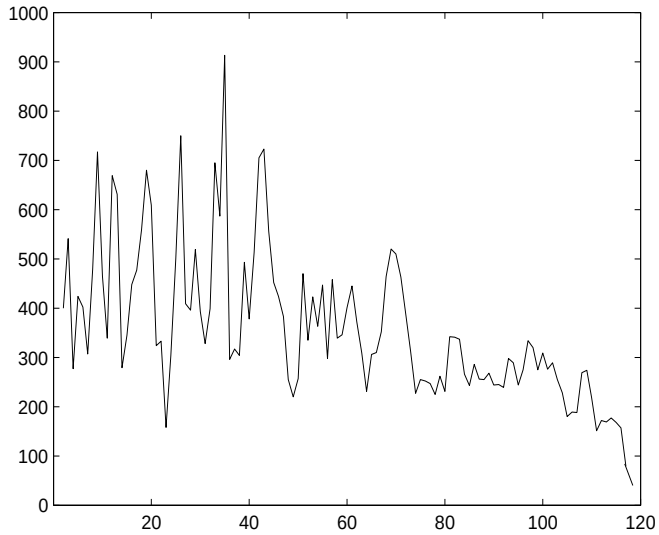
جدول ۳—۲: بهترین مقادیر پارامتر ρ برای الگوریتم $ACPMP$



شکل ۳—۲: نمودار مساله $ACPMP$ به ازای ۴۰۰ تکرار برای $pmed$۲۰



شکل ۳—۳: نمودار مساله $pmed$۲۰ به ازای ۴۰۰ تکرار برای $SAPMP$



شکل ۳—۴: نمودار تغییرات تابع هدف $pmed$۲۰ برای جوابهای مورد قبول درالگوریتم$SAPMP$

کتاب‌نامه

[1] Avella P., Boccia M., Sforza A. and Vasil'Ev I. (2005), "A branch-and-cut algorithm for the median-path problem", *Computational Optimization and Applications*, 32: 215-230.

[2] Becker R.I.,Chang Y., I. Lari, A. Scozzari, G. Storchi, (2002), "Finding the l-core of a tree", *Discrete Applied Mathematics* 118: 25-42.

[3] Becker R.I. and Perl Y., (1985) "Finding the two-core of a tree", *Discrete Appl. Math.*, 11: 103-113.

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

[5] Bullnheimer B., Hartl R. F. and Strauss C. (1997), "A new rank-based version of the ant system: a computational study", *Central European Journal for Operations Research and Economics.*

[6] Cerny V., (1985) "A thermodinamical approach to travelling salesman problem: an efficient simulated algorithm", *J. Optimiz. Theory Appl.*, 45: 41-55.

[7] Church. R.L, Revelle. C.S, *theoretcal and comptational links between p-median location set-covering and maximal covering location problem,* Geographical Analysis . 8 (1976) 406-415.

[8] Current J.R., Revelle C.S. and Cohon C.S. (1989), "The Median Shortest Path Problem: A multiobjective approach to analyze cost vs. accessibility in the design of transportation networks", *Transportation Science*, 21: 188-197.

[9] Drezner. H, Hamacher. H, *Facility location:Application and theory* Springer-Verlag,Berlin. 2002.

[10] Dorigo M., Maniezzo V. and Colorni A., (1996), "The Ant System: Optimization by a colony of cooperating agents", *IEEE Transactions on Systems, Man, and Cybernetics*ت*Part B*, 26: 1-13.

[11] Dorigo M., and Stützle T. (2004), *Ant Colony Optimization*, MIT Press, Cambridge (MA).

[12] Eiselt. H.A, Laporte. G, *Objectives in location problems.Chapter 8 in :Facility Location,* A Survey of Applications and Methods. Z. Drezner (eds),Springer-Verlag, New York, Inc: 1995.

[13] Francis. R.L, McGinnis. L.F, White. J.A, *Facility layout and location:An analytical approach, 2nd ed,* Prentice Hall. 1992.

[14] Gavish .B, Sridhar .S *Computing the 2-median on tree network in o(n logn) time ,* Networks. 26 (1995) 305-317.

[15] Hakimi. S.L, *Optimum locations of switching centers and the absolute centers and medians of a graph,* operations research . 12 (1964) 450-459.

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

[17] Hale. T., *Trevor Hale's location science references.* http:/wwwent.ohiou.edu/thale/thlocation.html. 2004.

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

[19] Handler. G.Y , *Minimax location of a facility in a undirected tree network,* Trans.Sci. 7 (1973) 287-293.

[20] Handler. G.Y, Mirchandani. P.B, *Location on network;Theory and Algorithms.* MIT Press ,Cambridge. 1979.

[21] Hansen. P, Labbe. M, Peeters. D, Thisse. J.F *Single facility Location on network.* Annals of Dscrete Math. 31 (1987) 113-146.

[22] Hotelling. H, *Stability in competetion.* Economic Journal . 39 (1929) 41-57.

[23] Kariv O, and Hakimi S. L. (1979), "An algorithmic approach to network location problems. Part I: p-centers." *SIAM J. Appl. Math.,* 37: 561-580.

[24] Kirkpatrick S., Gelatt Jr., Vecchi M. (1983), "Optimization by simulated annealing", *Science*, 220: 671-680.

[25] Krarup. J, Pruzan. P.M, *The simple plant location problem:Survey and synthesis.* European Journal of Opeartional Research. 12 (1983) 36-81.

[26] Labbe M., Laporte G., and Rodriguez Martin I. (1998), "Path, tree cycle location" in *Fleet Management and Logistics*, T.G. Crainic and G. Laporte (Eds.), Kluwer.

[27] Lari I., Ricca F. and Scozzari A. (2007), "Comparing different metaheuristic approaches for the median path problem with bounded length",To apear in *European Journal of Operational Research.*

[28] Levanova T. V. and Loresh A. (2004), "Algorithms of ant system and simulated annealing for the p-median problem", *Automation and Remote Control*, 65:80-88.

[29] Minieka E. (1985) "The optimal location of a path or tree in a tree network" *Networks*, 15: 309-321.

[30] Minieka E. and Patel N. H. (1983), "On finding the core of a tree with a specified length", *J. Algorithms*, 4: 345-352.

[31] Mirchandani .P.B, Francis.R, *Diccrete Location Theory.* Fuzzy sets and systems. 142 (2004) 393-405.

[32] Morgan C.A., Slater P.J. (1980), "A linear algorithm for a core of a tree", *Journal of Algorithms*, 1: 247-258.

[33] Owen. S.H, Daskin. M.S, *Strategic facility location: A review.* European Journal of Opeartional Research. 111 (1998) 423-447.

[34] Peng S., Lo W. (1994), "A simple optimal parallel algorithm for a core of a tree.", *Journal of Parallel and Distributed Computing*, 20: 388-392.

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

[36] Richey M.B. (1990), "Optimal location of a path or tree on a network with cycles", *Networks*, 20: 391-407.

[37] Scaparra. M.P, Scutella. M.G, *Strategic facility location customers: Building blocks of location models: A survey.* Thecnical Report: tr-01-18, University of Piza, Italy. 2001.

[38] Slater. P.J *Locating central path in a graph,* Transportation Science. 16 (1982) 1-18.

[39] Tavakkoli-Moghaddam R., Safaei N. and Gholipour Y. (2006), "A hybrid simulated annealing for capacitated vehicle routing prob-

lems with the independent route length", *Applied Mathematics and Computation*, 176: 445-454.

[40] Van-Laarhoven P. J. M., Aarts E. (1987), *Simulated annealing: theory and applications*, Kluwer, Dordrecht.

[41] Wang B.F. (2000), "Efficient parallel algorithms for optimally locating a path and a tree of a specified length in a weighted tree network"

[42] Wang B.F. (2002), "Finding a 2-core of a tree in linear time", *SIAM J. DISCRETE MATH.* 15: 193-210

پیوست A

واژه‌نامه‌ی انگلیسی به فارسی

الگوریتم مورچه ..... ant algorithm

همسایه ..... adjacent

لیست رئوس مجاور ..... adjacent point list

پسرو ..... backward

منصف ..... bisector

شاخه ..... branch

مرکز ..... center

مسیر مرکزی ..... central path

فرزند ..... child

هسته مقید ..... conditional-core

هسته ..... core

دور ..... cycle

درجه ..... degree

قطر ..... diameter

لیست نقطه بعدی ..... end point list

پیشرو ..... forward

بهینه سازی سراسری ..... global optimization

گراف ..... graph

ابتکاری ..... heuristic

روشهای ابتکاری ..... heuristic methods

شاخه داخلی ..... in-branch

همسایه داخلی ..... in-neighbor

l-هسته ..... l-core

الگوریتم اکثریت ..... majority algorithm

مسیر ماکسیمال ..... maximal path

Np-کامل ..... Np-complete

همسایگی ..... neighborhood

Np-سخت ..... Np-hard

مجموعه بیرونی ..... outset

پدر ..... parent

مسیر ..... path

فرمون ..... pheromone

p—میانه ..... p-median

ریشه ..... root

شبیه سازی تبریدی (تدریجی) ..... simulated anealing

سه— هسته ..... three-core

درخت ..... tree

دو— هسته ..... two-core

ناخوشایند ..... undesirable

گشت ..... walk