

مسائل مکانیابی متعادل معکوس با تغییر وزن یال روی درخت

شاهده امیدى نورابادی^۱، دکتر جعفر فتحعلی^۲

^۱دانشجو، دانشگاه صنعتی شاهرود

^۲استاد، دانشگاه صنعتی شاهرود

چکیده

در این مقاله، حالت جدیدی از تعادل را در یک مساله مکانیابی روی گراف ارائه داده ایم. هدف ما در این مساله این است که با اصلاح وزن یال‌ها با کمترین هزینه ممکن، تفاضل بین بیشترین و کمترین مجموع فاصله مشتری‌ها به هر سرویس‌دهنده کمینه شود. به عبارت دیگر با اصلاح وزن یال‌ها با کمترین هزینه ممکن، می‌خواهیم مجموع فاصله نقاط تقاضا از دو سرویس‌دهنده در تعادل باشد. در انتها الگوریتمی با پیچیدگی محاسباتی $O(n^2)$ برای حل این گونه مسائل ارائه شده است.

واژه‌های کلیدی: مکانیابی، مکانیابی متعادل، مکانیابی معکوس.

۱ مقدمه

مسائل مکانیابی متعادل با قرار دادن سرویس‌دهنده‌ها در مکانی، بطوریکه برابری در سرویس‌دهی به نقاط تقاضا حداکثر شود، سروکار دارند. این مسائل در چند سال اخیر، مورد توجه بسیاری از ریاضیدانان قرار گرفته است. ایرکوت در مقاله [۲] تعادل در فاصله درباره مشتری‌ها مورد مطالعه قرار داد. برمن در مقاله [۱] مکان سرویس‌دهنده طوری پیدا کرد که حداکثر بار تخصیص یافته به هر سرویس‌دهنده کمینه شود. در مسائل مکانیابی معکوس، اصلاح پارامترها با کمترین هزینه، بطوریکه مکان سرویس‌دهنده‌ها از پیش تعیین شده، با پارامترهای جدید بهینه شود. بورکارد [۲] مساله p میانه معکوس را بررسی کرد و الگوریتم $O(n \log n)$ برای ۱- میانه معکوس روی درخت ارائه داد. در اینجا قصد داریم حالت جدیدی از تعادل در فاصله در یک درخت با ۲- میانه مطرح کنیم. هدف ما این است که با تغییر وزن یال‌ها درخت با کمترین هزینه ممکن، تعادل بین مجموع فاصله مشتری‌ها از دو سرویس‌دهنده را ایجاد کنیم. در این مساله دو تابع هدف داریم، تعادل و کمترین هزینه (معکوس). همچنین الگوریتم ابتکاری برای حل این مساله ارائه شد.

۲ توصیف مساله

قرار دهید $T = (V, E)$ یک گراف بدون جهت باشد. بطوریکه V مجموعه رئوس گراف T و E مجموعه یال‌های گراف T و $|V| = n$ و $|E| = m$ باشد. هر راس v_i دارای یک وزن نامنفی w_i است. برای هر یال $e_i \in E$ ، فرض کنید $l_i \geq 0$

وزن یال e_i است. این وزن نامنفی را می‌توان فاصله یا زمان تعبیر نمود. برای هر جفت راس v_i و v_j در درخت T ، فرض کنید $d(v_i, v_j)$ کوتاهترین فاصله بین v_i و v_j در درخت T است. قرار دهید m_1 ، m_2 دو راس در درخت T که دو سرویس‌دهنده درخت T فرض شده است. فرض کنید

$$V_1 = \{v_i \in V | d(v_i, m_1) \leq d(v_i, m_2)\}$$

و $V_2 = V \setminus V_1$ ، مجموعه رئوسی هستند که به ترتیب به دو سرویس‌دهنده m_1 و m_2 تخصیص می‌یابند. ما در اینجا قصد داریم با تغییر وزن یال‌ها با کمترین هزینه، تفاضل بین مجموع فاصله مشتری‌ها به هر سرویس‌دهنده در V_1 و V_2 کمینه شود.

برای هر یال e_i ، فرض کنید هزینه افزایش هر واحد وزن l_i ، c_i^+ و هزینه کاهش هر واحد وزن l_i ، c_i^- است. همچنین فرض کنید q_i^+ و q_i^- به ترتیب مقادیر افزایش و کاهش وزن یال l_i است. قرار می‌دهیم

$$Q^+ = \{q_1^+, q_2^+, \dots, q_m^+\}, Q^- = \{q_1^-, q_2^-, \dots, q_m^-\}.$$

و همچنین برای هر $i = 1, \dots, m$ فرض کنید $\hat{l}_i = l_i + q_i^+ - q_i^-$

در این مساله، ما به دنبال کمینه کردن توابع زیر هستیم.

$$f_1(Q^+, Q^-) = \sum_{i=1}^m (c_i^+ q_i^+ + c_i^- q_i^-). \quad (1.2)$$

$$f_2(Q^+, Q^-) = \left| \sum_{i \in V_1} d(m_1, v_i) - \sum_{i \in V_2} d(m_2, v_i) \right|. \quad (2.2)$$

۳ الگوریتم حل مساله

در این بخش الگوریتمی برای حل این مسائل مطرح شده است. در این الگوریتم T_1, T_2 دو زیر درخت T است. اگر $\sum_{i \in V_1} d(m_1, v_i) > \sum_{i \in V_2} d(m_2, v_i)$ ، آنگاه مساله بهینه است. در غیر این صورت اگر $\sum_{i \in V_1} d(m_1, v_i) < \sum_{i \in V_2} d(m_2, v_i)$ ، افزایش وزن یال‌ها در زیر درخت T_1 و کاهش وزن یال‌ها در زیر درخت T_2 مقرون به صرفه نمی‌باشد. بدیهی است که در اینجا افزایش و کاهش وزن یال موثر (یالی است که با تغییر وزن یال، مساله به حالت بهینگی نزدیکتر می‌شود). مقرون به صرفه است. برای افزایش و کاهش وزن یال‌ها در مسیر اصلی بین دو میانه، بایستی تغییر وزن یال طوری باشد که تغییری در تخصیص سرویس‌دهنده‌ها به هر دو میانه، ایجاد نشود. بنابراین برای این یال‌ها تغییر به اندازه q_i واحد می‌باشد که q_i به صورت زیر تعریف می‌شود.

$$\sum_{i \in V_1} d(m_1, v_i) = a, \quad \sum_{i \in V_2} d(m_2, v_i) = b$$

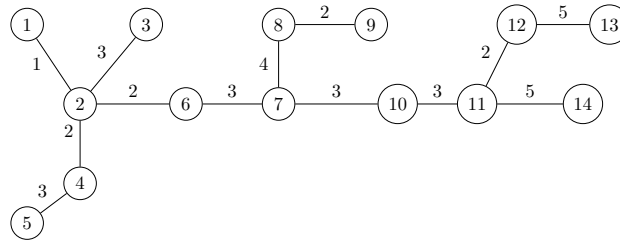
$$\min([(a-b)/k], u_i, |s_1 - s_2| \pm \epsilon) = q_i$$

s_1 و s_2 در عبارت بالا فاصله‌های اولین راس در درخت T_2 در مسیر بین دو میانه تا دو میانه در دو زیردرخت می‌باشد. k تعداد یال‌هایی هستند که با تغییر وزن یال i ام، فاصله آن‌ها تا سرویس‌دهنده دستخوش تغییرات می‌شوند. u_i کران تغییر وزن یال i ام است. برای یال‌های دیگر q_i به صورت زیر تعریف می‌شود.

$$\min([(a-b)/k], u_i) = q_i$$

در این الگوریتم با هر تغییر در وزن یال‌ها شرط بهینگی بررسی می‌شود و مقادیرهای a و b جدید جایگزین مقادیرهای a و b قدیم می‌شوند تا زمانی که مساله تا حد ممکن به تعادل نزدیک شود. مقدار ϵ در این الگوریتم مقدار ثابتی و نزدیک به صفر در نظر گرفته شده است.

در الگوریتم دو قرارداد وجود دارد.

شکل ۱: شکل درخت T با ۱۴ راس

• وزن یال حداقل یک در نظر گرفته شده است.

• چون تغییر وزن یال حذف یال تاثیری در حکم مساله ندارد پس هزینه های مرتبط آن را نادیده می گیریم.

برای توضیح الگوریتم مثال زیر را در نظر بگیرید.

مثال ۱.۳. درخت T را در شکل ۱ را در نظر بگیرید. وزن رئوس درخت T در جدول ۱ آمده است. همچنین $m_1 = v_2$ و

$m_2 = v_{11}$ ، دو میانه می باشند که به رئوس دیگر سرویس می دهند.

i	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴
w_i	۲	۲	۳	۲	۳	۲	۱	۳	۲	۳	۴	۴	۳	۲

جدول ۱: وزن رئوس درخت T

با مرتب کردن ضرایب هزینه به صورت صعودی داریم:

$$c_9^+ \leq c_1^+ \leq c_5^+ \leq c_{10}^+ \leq c_2^- \leq c_6^- \leq c_{13}^+ \leq c_4^+ \leq c_8^+ \leq c_9^- \leq c_3^+ \leq c_1^- \leq c_5^- \leq c_{10}^- \leq c_3^- \leq c_6^+ \leq c_8^- \leq c_{12}^- \leq c_7^+ \leq c_{11}^+ \leq c_2^+ \leq c_{14}^+ \leq c_{13}^- \leq c_{12}^+ \leq c_4^- \leq c_{11}^- \leq c_7^- \leq c_{13}^- \leq c_6^- \leq c_{11}^- \leq c_7^+ \leq c_{14}^+$$

شماره یال	یال	c^-	c^+	u^-	u^+
۱	۱-۲	۱/۸	۱/۲	$0 \leq q_1^- \leq 1$	$0 \leq q_1^+ \leq 3$
۲	۲-۳	۱/۶	۱/۲	$0 \leq q_2^- \leq 2$	$0 \leq q_2^+ \leq 3$
۳	۲-۴	۱	۱/۸	$0 \leq q_3^- \leq 4$	$0 \leq q_3^+ \leq 2$
۴	۴-۵	۱/۵	۱/۶	$0 \leq q_4^- \leq 2$	$0 \leq q_4^+ \leq 3$
۵	۲-۶	۱/۹	۱/۳	$0 \leq q_5^- \leq 1$	$0 \leq q_5^+ \leq 2$
۶	۶-۷	۱/۴	۱	$0 \leq q_6^- \leq 1$	$0 \leq q_6^+ \leq 2$
۷	۷-۸	۱/۳	۲	$0 \leq q_7^- \leq 3$	$0 \leq q_7^+ \leq 2$
۸	۸-۹	۱	۱/۷	$0 \leq q_8^- \leq 2$	$0 \leq q_8^+ \leq 4$
۹	۷-۱۰	۱/۸	۱	$0 \leq q_9^- \leq 4$	$0 \leq q_9^+ \leq 2$
۱۰	۱۰-۱۱	۱/۹	۱/۳	$0 \leq q_{10}^- \leq 1$	$0 \leq q_{10}^+ \leq 1$
۱۱	۱۱-۱۲	۱/۸	۱/۲	$0 \leq q_{11}^- \leq 2$	$0 \leq q_{11}^+ \leq 2$
۱۲	۱۲-۱۳	۱	۲	$0 \leq q_{12}^- \leq 1$	$0 \leq q_{12}^+ \leq 3$
۱۳	۱۱-۱۴	۱/۴	۱/۵	$0 \leq q_{13}^- \leq 4$	$0 \leq q_{13}^+ \leq 1$

جدول ۲: هزینه تغییر وزن یال ها در درخت T

برای به دست آوردن ۲ میانه درخت با استفاده از الگوریتم حذف یال، درخت به دو زیر درخت T_1 و T_2 تبدیل می شود. در

این مثال فرض کردیم m_1 و m_2 دو میانه زیر درخت T_1 و T_2 است.

با توجه به مشخص بودن میانه ها، در ابتدا

$$\sum_{i \in V_1} d(m_1, v_i) = 38 \quad \sum_{i \in V_2} d(m_2, v_i) = 17$$

وزن یال ۱۰ به اندازه $1 = \min([21/1], 1, |8-3| - \epsilon)$ واحد افزایش می یابد. بنابراین

$$\sum_{i \in V_1} d(m_1, v_i) = 38 \quad \sum_{i \in V_2} d(m_2, v_i) = 18$$

وزن یال ۲، $2 = \min([20/1], 2)$ واحد کاهش می یابد. بنابراین

$$\sum_{i \in V_1} d(m_1, v_i) = 36, \quad \sum_{i \in V_2} d(m_2, v_i) = 18$$

وزن یال ۱۳ به اندازه $1 = \min([18/1], 1)$ افزایش می یابد. بنابراین

$$\sum_{i \in V_1} d(m_1, v_i) = 36, \quad \sum_{i \in V_2} d(m_2, v_i) = 19$$

وزن یال ۲ به اندازه ۲ واحد کاهش می یابد و وزن یال ۱ بدون تغییر می ماند. (فرض شده وزن یال حداقل یک باشد). بنابراین

$$\sum_{i \in V_1} d(m_1, v_i) = 34, \quad \sum_{i \in V_2} d(m_2, v_i) = 19$$

وزن یال ۵ به اندازه $1 = \min([15/4], 1, |8-4| + \epsilon)$ واحد کاهش می یابد. بنابراین

$$\sum_{i \in V_1} d(m_1, v_i) = 30, \quad \sum_{i \in V_2} d(m_2, v_i) = 19$$

وزن یال ۳ به اندازه $1 = \min([11/2], 1)$ و وزن یال ۸ به اندازه $1 = \min([9/1], 1)$ کاهش می یابد. بنابراین

$$\sum_{i \in V_1} d(m_1, v_i) = 27, \quad \sum_{i \in V_2} d(m_2, v_i) = 19$$

وزن یال ۱۱ به اندازه $2 = \min([8/2], 2)$ افزایش می یابد و وزن یال ۷ به اندازه $2 = \min([4/2], 3)$ کاهش می یابد.

$$\sum_{i \in V_1} d(m_1, v_i) = 23, \quad \sum_{i \in V_2} d(m_2, v_i) = 23$$

الگوریتم به پایان می رسد.

بحث و نتیجه گیری

در این مقاله، مساله مکانیابی متعادل معکوس با تغییر وزن یال ها روی درخت را بررسی کردیم. تعادل روی مجموع فاصله مشتری ها از دو سرویس دهنده در نظر گرفته شد و الگوریتم $O(n^2)$ در حالتی که درخت دارای دو سرویس دهنده است، ارائه شد. برای ادامه کار نوع دیگری از تابع هدف متعادل از جمله کمینه تفاضل فاصله از یک سرویس گیرنده به نزدیکترین و دورترین سرویس دهنده پیشنهاد می شود. همچنین مدل های دیگر معکوس نیز می توان بررسی کرد.

مراجع

- [1] Berman O., Drezner Z., Tamir A., Wesolowsky G.O. (2009), Optimal location with equitable loads *Annals of Operations Research*, **167**, 307-325.
- [2] Burkard R. E., Pleschiutchnig C., Zhang J. Z. (2004), Inverse median problems, *Discrete Optimization*, **1**, 23-39.
- [3] Erkut E. (1993), inequality measures for location problems, *location science*, 199-217.