



ارائه الگوريتم ابتکاري برای مساله درخت فراگير کمبينه با محدوديت تاخير

آيدا كلانه آهنی^۱، مجید سالاري^۲، و سيد محمود حسیني^۲

گروه مهندسي صنایع، دانشکده مهندسي، دانشگاه فردوسي مشهد
ida.ahani@gmail.com
msalar@ferdowsi.um.ac.ir^۱
mhosseini@ferdowsi.um.ac.ir^۲

چکیده. در اين مقاله يك روش حل ابتکاري برای مساله درخت فراگير کمبينه با محدوديت تاخير ارائه مي گردد. در اين مساله هدف کمبينه کردن هزينه کل درخت بوده، بطوريكه ميزان تاخير در هر مسیر موجود از گره منبع به سائر گرهها از مقدار مشخصی بيشتر نشود. مقابله تابع اين روش با تابع بدست آمده از روش هاي حل دقیق موجود، کارابي اين الگوريتم را نشان مي دهد.

۱. پيش‌گفتار

مساله درخت فراگير کمبينه يكی از مسائل کلاسیک در بهینه‌سازی ترکیبیاتی می‌باشد که تا کنون با محدودیت‌های مختلفی مطرح شده است. از جمله‌ی این محدودیت‌ها، محدودیت تاخیر می‌باشد. به فرض گراف همیند و بدون جهت $G = (V, A)$ داده شده، که V مجموعه رئوس و A مجموعه پال‌های آن باشد. در این مساله به هر پال دو مقدار به عنوان هزینه و تاخیر نسبت داده می‌شود. هدف این مساله پیدا کردن درخت فراگيری است با حداقل هزینه و تاخیر بطوریکه مجموع تاخير پال‌های هر مسیر موجود از گره منبع به سائر گرهها از يك مقدار مشخص مانند H بيشتر نگردد. اگر مقدار تاخير برای تمامی پال‌ها برابر يك در نظر گرفته شود، اين مساله به مساله $HCMST$ ^۱ تبدیل خواهد شد [۱]. در [۲] اولین روش حل ابتکاري برای حل اين مساله مطرح گردید و در [۳] نيز روش‌های

2010 Mathematics Subject Classification. Primary 47A55; Secondary 39B52, 34K20, 39B82.

وازگان کليدي. درخت فراگير، الگوريتم ابتکاري، محدوديت تاخير.

¹Hob Constrained Minimum Spanning Tree

۲. کلاسه آهنی، م. سالاری، و س. م. حسینی

حل دقیق برای این مساله مورد بررسی قرار گرفت. در مقاله پیش روی روش حل ابتکاری $ALNS^3$ برای این مساله مطرح شده و برای داده های ارائه شده در [۲] که جواب های بهینه آنها در همین مقاله با استفاده از روش های حل دقیق بدست آمده، اجرا گردید.

۲. روش حل ابتکاری $ALNS$

این روش برای اولین بار در [۴] مطرح شد که برای انواع مختلفی از مسائل بهینه سازی کاربرد دارد. این روش از یک جواب اولیه شروع کرده و سپس در هر تکرار از الگوریتم های مخرب و سازنده متعددی برای جستجوی فضای شدنی بزرگتری از جواب استفاده می کند. بطور کلی در هر تکرار، بخشی از جواب فعلی با استفاده از یک الگوریتم مخرب، تخریب شده و سپس توسط یک الگوریتم سازنده مجدد بازسازی می شود. بعلاوه انتخاب این الگوریتم ها بسته به میزان تاثیرگذاری آنها در بهبود کیفیت جواب در تکرارهای قبلی می باشد. الگوریتم $ALNS$ ارائه شده در این مقاله شامل k_1 تکرار کلی بوده، که هر تکرار کلی نیز شامل k_2 زیر تکرار می باشد. ۵ الگوریتم مخرب و ۵ الگوریتم سازنده در ساختار این روش مورد استفاده قرار می گیرد. به هر کدام از این الگوریتم ها وزنی نسبت داده می شود که در ابتداء برای تعاملی آنها برابر یک در نظر گرفته شده است. بعلاوه به هر الگوریتم مخرب و سازنده یک نمره نسبت داده می شود که نشاندهندگی عملکرد آنها در هر بخش از الگوریتم $ALNS$ بوده و در ابتدای هر بخش نیز برابر صفر قرار داده می شوند. در هر تکرار یک الگوریتم سازنده و یک الگوریتم مخرب با استفاده از قاعده هی چرخ رولت انتخاب می شود. اگر جواب حاصل از اجرای الگوریتم مخرب α و سازنده β از بهترین جوابی که تا کنون بدست آمده بهتر باشد نمره این الگوریتم ها به اندازه α اضافه شده و اگر جواب حاصل بدتر از بهترین جواب موجود بوده اما از جواب قبل از اجرای الگوریتم سازنده و مخرب در تکرار فعلی بهتر باشد، نمره این الگوریتم ها به اندازه β اضافه می شود. چنانچه جواب بدست آمده بهبودی ایجاد نکند، اما با استفاده از قاعده هی شبیه سازی تبرید پذیرفته شود، نمره این الگوریتم ها به اندازه α اضافه خواهد شد. پس از ϕ_2 تکرار، وزن های این الگوریتم ها با توجه به نمراتی که در این تکرارها کسب کرده اند، بروز می شوند. در ادامه توضیحات مربوط به این الگوریتم ها آورده می شود.

الگوریتم مخرب ۱: یک گره به تصادف انتخاب شده و مسیر با کمترین میزان تاخیر اضافه می شود، سپس بالهایی که باعث ایجاد حلقه در جواب می شوند حذف می گردند.

الگوریتم مخرب ۲: مشابه الگوریتم مخرب ۱ بوده با این تفاوت که در هر تکرار ۲ گره به تصادف انتخاب می شود.

الگوریتم مخرب ۳: مشابه الگوریتم مخرب ۱ بوده با این تفاوت که در هر تکرار ۲ گره که هزینه بین آنها بیشتر از یک مقدار مشخص باشد انتخاب می شود.

الگوریتم مخرب ۴: یک بال از درخت به تصادف انتخاب و حذف می گردد. سپس یک

³ Adaptive Large Neighborhood Search

گره از جواب باقیمانده به تصادف انتخاب و زیر درخت جدا شده به این گره متصل می‌گردد. در صورت برقرار نبودن شرط محدودیت تا خیر گرهی دیگری انتخاب می‌شود. الگوریتم مخرب ۵: مشاهد الگوریتم مخرب ۴ بوده با این تفاوت که یکی از گرهای زیر درخت جدا شده، به تصادف انتخاب و سپس از سمت این گره مجدداً به درخت متصل می‌گردد.

الگوریتم سازنده ۱: یک پال به تصادف انتخاب شده و حذف می‌گردد. تمامی گرهای زیر درخت جدا شده با ترتیبی تصادفی مجدداً به یک نقطه‌ی اتصال شدنی با کمترین هزینه که می‌تواند بین دو گره متوالی در درخت (نوع ۱) و یا یک گره (نوع ۲) باشد، متصل می‌گرددند. این الگوریتم تا زمانیکه حذف تمامی پال‌ها بررسی شود ادامه می‌باید. الگوریتم سازنده ۲: تمامی گرهای برگ در جواب موجود جدا شده و سپس با ترتیبی تصادفی مجدداً به یک نقطه‌ی اتصال شدنی با کمترین هزینه (نوع ۱ و یا نوع ۲) متصل می‌گرددند.

الگوریتم سازنده ۳: یک پال به تصادف انتخاب و حذف می‌گردد. سپس از بین گرهای زیر درخت جدا شده به تصادف یک گره انتخاب می‌شود و به یک نقطه‌ی اتصال شدنی نوع ۲ که کمترین هزینه را ایجاد می‌کند، متصل می‌شود.

الگوریتم سازنده ۴: دو گره به تصادف انتخاب و با یکدیگر جایجا می‌شوند. در صورتیکه جواب حاصل هزینه کمتری داشته و شرط حداقل میزان تا خیر نیز برقرار باشد، پذیرفته می‌شود. جایجا یکی تمامی ترکیب‌های دوتابی با ترتیبی تصادفی بررسی می‌شود.

الگوریتم سازنده ۵: تمامی گرهای برگ جدا شده (مجموعه L) و تمامی رشته‌های یکی (S^l) و دوتابی (S^m) این گرهای برگ تولید می‌شود. سپس این رشته‌ها از طریق حل یک مدل ریاضی مجدداً بطور بهینه به جواب متصل می‌شوند. در ادامه توضیحات پارامترها، متغیرها و مدل مربوطه آورده شده است:

$$S_l = \text{مجموعه رشته‌های شامل گره } l$$

$$r_{ej} = \text{قابلیت اطمینان مسیر موجود تا گره } j$$

$$r_{ej} = \text{قابلیت اطمینان رشته } e \text{ (اگر رشته یکی باشد برابر یک در غیر اینصورت برابر}$$

$$r_{ej} = \text{قابلیت اطمینان بین دو گره تشکیل‌دهنده رشته } e \text{ خواهد بود}$$

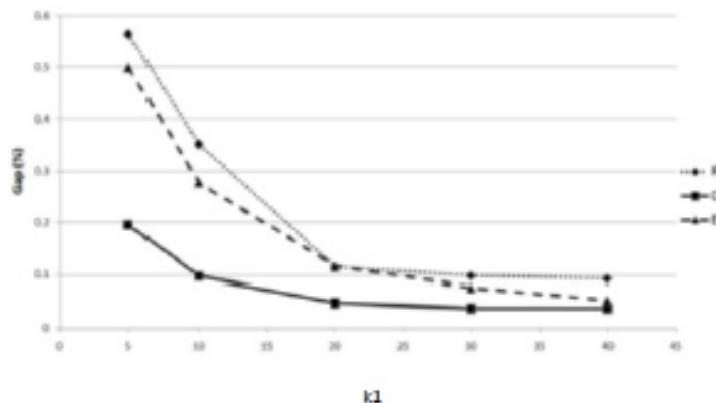
$$J_e = \text{مجموعه نقاط شدنی برای اتصال رشته } e$$

$$y_{ej} = \text{اگر رشته } e \text{ به نقطه } j \text{ متصل شود مقدار یک را بگیرد.}$$

$$\begin{aligned} \min & \sum_{s \in S^l \cup S^m} \sum_{j \in J_s} c_{sj} y_{sj} \\ \text{s.t.} & \sum_{s \in S_l} \sum_{j \in J_s} y_{sj} = 1 \quad \forall l \in L \\ & r_{ej}(1 - y_{ej}) + r_{ej} \times r_{es} \times r_{jb_s} \times y_{ej} \geq \alpha \quad \forall j \in JS^l, s \in S^l \cup S^m \\ & y_{ej} \in \{0, 1\} \end{aligned}$$

نتایج محاسباتی

الگوریتم *ALNS* مطرح شده در این مقاله بر روی داده‌های ارائه شده در [۲] اجرا گردید و با نتایجی که در همین مقاله با استفاده از روش‌های حل دقیق بدست آمده بود مقایسه شد. نحوه‌ی تولید این داده‌ها در [۲] شرح داده شده است. نموداری که در ادامه آورده شده، خلاصه این نتایج را نشان می‌دهد. این نمودار میانگین اختلاف بین نتایج حل دقیق و نتایج *ALNS* در کلیه‌ی داده‌ها را بر حسب تعداد تکرارهای کلی k_1 نشان می‌دهد. همان‌گونه که مشاهده می‌شود با افزایش تعداد تکرارها این اختلاف به سمت صفر میل می‌کند و حاکی از کارایی این الگوریتم می‌باشد.



شکل ۱: میانگین اختلاف نتایج حل دقیق و نتایج *ALNS*

مراجع

1. G. Dahl, L. Gouveia, C. Requejo *On formulations and methods for the hop-constrained minimum spanning tree problem*, In: Mauricio Resende, Panos Pardalos., editors, *Handbook of Optimization in Telecommunications*, Springer 2006, 493-515.
2. HF. Salama, DS. Reeves, Y. Vinotis *The delay-constrained minimum spanning tree problem*, Proc. ISCC, (1997), 699-703.
3. L. Gouveia, A. Palas, D. Sharma *Modeling and solving the rooted distance-constrained minimum spanning tree problem*, Comput. Oper. Res., (2008), 35:600-613.
4. S. Ropka, D. Pisinger *An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows*, Transport. Sci., (2006), 40:455-472.

چهل و چهارمین کنفرانس ریاضی ایران

شماره :

تاریخ :

پیوست :

۵ الی ۸ شهریور ۱۳۹۲

The 44th Annual Iranian Mathematics Conference

27-30 August 2013



گواهی می شود

سرکار خانم آیدا کلاتز آهنی

در چهل و چهارمین کنفرانس ریاضی کشور که از تاریخ ۵ الی ۸ شهریور ۱۳۹۲ در دانشگاه فردوسی مشهد برگزار گردیده شرکت نموده و

پوستری با عنوان

ارائه الگوریتم ابکاری برای مساله دخت فرگیر کینه با محمد و دیت تانیر

ارائه نموده اند.

با احترام

دکتر علی کنفرانس

دکتر محمد صالح مصلحیان



مشهد، میدان آزادی

دانشگاه فردوسی مشهد - دانشکده علوم ریاضی

صندوق پستی: ۹۱۷۷۵ - ۱۱۵۹

نمبر: ۴-۸۸۲۸۶۰۴ - ۵۱۱

imc44@um.ac.ir

منزلگاه: <http://imc44.um.ac.ir>