

الگوریتم جستجوی ممنوعه TS

الگوریتم‌های فرا ابتکاری

حسین کریمی

- مقدمه و تاریخچه
- همسایگی
- لیست ممنوعه
- معیار آزاد سازی از لیست (Aspiration Criteria)
- الگوریتم اولیه
- معیارهای توقف
- پراکندگی و شدت افزایی
- حافظه‌ها در TS
- مقایسه SA و TS

- عبارت Tabu(Taboo) از یک زبان پولنیزیایی ریشه می گیرد که توسط مردم بومی جزیره tonga برای مشخص کردن چیزهایی بکار می رود که مقدس و غیرقابل لمس و یا (بخاطر خطر داشتن) ممنوع شده هستند.
- ارتباط این کلمه با حافظه مردم آن منطقه از این جهت که تجربیات گذشته باعث شده است تا چنین تلقی امروزی در مورد یک مفهوم خاص بوجود آید، کلید اصلی ارتباط این کلمه با مفهوم ممنوعیت در Tabu Search است.
- عناصر ممنوع در Tabu Search با ارجاع به حافظه مشخص می شوند.

مقدمه و تاریخچه - ادامه

- چنانکه می‌دانید، الگوریتم‌های فرا ابتکاری بسیاری برای دستیابی به حداقل یک جواب خوب (نه لزوماً بهترین) برای یک مساله $NP-Hard$ بوجود آمده است.
- بسیاری از این روشها از یک مکانیزم جستجوی محلی (LS) بهره می‌گیرند.
- LS را می‌توان یک روال جستجوی تکرارشونده دانست که از یک جواب شدنی شروع می‌کند و با انجام اصلاحات جزئی (همان همسایگی‌ها)، آنرا تا رسیدن به یک بهینه محلی ادامه می‌دهد.
- با در نظر داشتن این نکته که در حالت معمول این بهینه محلی، چیزی بیش از یک جواب متوسط نیست.

مقدمه و تاریخچه – ادامه

- در LS معمولاً کیفیت جواب بدست آمده به حد زیادی بستگی به غنای حرکتهای تعریف شده دارد. و این یک مساله اساسی در رویکردهای مبتنی بر LS است.
- Tabu Search در سال ۱۹۸۶ توسط Fred Glover برای غلبه بر این مشکل ارائه شد.
- اصل اولیه در TS، مجاز دانستن حرکتهایی که بهبودی به همراه ندارند، برای ادامه دادن جستجو در LS است، وقتی که به یک بهینه موضعی بر میخوریم.

مقدمه و تاریخچه – ادامه

- البته در این روش برای اجتناب از دور زدن و رسیدن به جواب‌هایی که پیش از این بدست آمده، از حافظه‌ای بنام لیست ممنوعه Tabu List استفاده می‌کنیم.
- این حافظه جواب‌های اخیر و یا حرکتهای اخیر را در خود ضبط می‌کند.
- در واقع یک TS ساده را می‌توان ترکیبی از یک حافظه کوتاه مدت با LS دانست.



- از اولین مفاهیمی که در TS می‌باید بدان پرداخت، مفهوم همسایگی است.
- در هر تکرار، انتقالی (move) که بر روی جواب S اعمال می‌شود، مجموعه‌ای از جواب‌ها را در فضای جستجو تعریف می‌کند که جواب‌های همسایه گفته می‌شوند ($N(S)$)
- پس همسایگی، زیرمجموعه‌ای از فضای جواب است که به شکل زیر تعریف می‌شود:
- $N(S)$: مجموعه‌ی جواب‌هایی که با استفاده از یک انتقال، از جواب S بدست می‌آیند.





- چنانچه از تعریف بر می آید، ساختار همسایه، می تواند حتی شامل تمامی فضای جواب نیز باشد.
- برای یک مساله خاص، نوع انتقال یا move تعریف شده، نقشی اساسی در وسعت همسایگی بوجود آمده دارد.





- گفتیم که مفهوم اساسی در TS مجاز دانستن جواب‌هایی است که در تابع هدف بهبود ایجاد نمی‌کنند، اما ممکن است ما را به سمت جواب سراسری راهنمایی کنند، با این شرط که در لیست جواب‌های ممنوع قرار نداشته باشند.
- اما TS برای استفاده از چنین راه حلی، نیازمند آن است که از پدیده دور، که ناشی از بازگشت به جواب‌های پیشین است، جلوگیری کند. این، وظیفه Tabu ها است.



- مجموعه‌ای از انتقال‌های ممنوع که به حافظه سپرده می‌شوند تا چنین بازگشت‌هایی رخ ندهد.
- این انتقال‌های ممنوع، در حافظه‌ای کوتاه مدت (*Short Term Memory*) ذخیره می‌شوند تا (برای مدتی معین) انجام مجموعه‌ای معین از انتقال‌ها را ممنوع سازند. این مدت معین، یا اصطلاحاً *Tabu Tenure* بنا بر الگوریتم حل و نوع مساله و ماهیت انتقال‌ها متغیر است.
- حافظه مورد استفاده برای نگهداری Tabu ها معمولاً گردشی و دارای طول ثابت است.

معیار آزاد سازی از لیست (Aspiration Criteria)



- تحت شرایطی، بعضی از لیست‌های ممنوع بسیار قوی هستند و از حرکات‌های جذاب و خوب جلوگیری می‌کنند، این درحالی است که ممکن است هیچ خطری جهت افتادن در دور وجود نداشته باشد و یا ممکن است این حرکات باعث بهبود کلی جستجو شوند.
- از این رو لازم است که از الگوریتمی استفاده کنیم تا ممنوع بودن را برای آن مورد خاص لغو کنند و اجازه دهند آن متغیر وارد جستجو شود. این وسیله‌ها معیار رضایت نام دارند.
- ساده‌ترین و پراستفاده‌ترین معیار آزاد سازی که تقریباً در تمامی TSها استفاده می‌شود، به این صورت است که یک حرکت ممنوع، اگر به جوابی بهتر از جواب بهینه فعلی منجر گردد، باید آن حرکت انجام شود یعنی ممنوعیت آن لغو گردد (زیرا قبلاً چنین جوابی مشاهده نشده است).



معیار ۱

■ یک جواب باید از لیست ممنوع خارج شود، اگر از هر جوابی که قبلاً به دست آمده بر اساس مقدار تابع هدف بهتر باشد.

معیار ۲

■ هرگاه ساختار روش و لیست‌های ممنوع در مرحله‌ای از فرایند تکرار به گونه‌ای باشد که امکان هیچ گونه حرکت وجود نداشته باشد در این صورت حرکتی که در صف خارج شدن از لیست ممنوع از همه به خروج نزدیک‌تر است انتخاب و ممنوعیت آن برداشته می‌شود.

■ شروع :

- جواب اولیه امکان پذیر مانند S_0 را پیدا کنید و قرار دهید:
- Set $S := S_0$, $f^* := f(S_0)$, $S^* := S_0$, $T := \emptyset$.

■ جستجو

- تا زمانی که شرط پایانی برآورده نشده است، تکرار کن:
- یک S که عضو $\tilde{N}(S)$ است را انتخاب کن
- اگر $F(S) < F^*$ آنگاه قرار بده : $F^* := F(S)$ و $S^* := S$
- این حرکت را در لیست ممنوع قرار بده و قدیمی ترین ورودی لیست را اگر لازم است حذف کن.

شرایط توقف الگوریتم

توقف بعد از تعداد تکرار مشخص (مثلاً پس از کارکرد CPU به اندازه یک مدت زمان مشخص).

پس از تعدادی تکرار بدون اینکه جواب تابع هدف تغییر کند (این مورد، بیشترین استفاده را تا به حال داشته است).

وقتی که جواب به یک حد آستانه‌ای و مقدار از پیش تعیین شده برسد.

پراکندگی و شدت افزایی

- عبارت شدت افزایی یا Intensification در TS به مکانیسمی اطلاق می‌شود که بر اساس آن جواب‌ها و حرکتهایی که باعث رسیدن به جواب‌های خوب شده‌اند، تقویت می‌شوند.
- به عبارت بهتر، این استراتژی، به بازگشت به جواب‌های نخبه و جستجوی بیشتر در محدوده آن‌ها، اشاره دارد.
- از آنجا که جواب‌های نخبه برای مقایسه‌های بعدی مورد نیاز هستند، نحوه پیاده‌سازی این مفهوم شامل در نظر گرفتن یک حافظه مشخص برای این مجموعه از جواب‌ها است.

پراکندگی و شدت افزایی-ادامه

- استراتژی‌های شدت افزایی به ابزاری برای مشخص کردن مجموعه‌ای از جواب‌های نخبه نیاز دارند، تا پایه‌ای برای ترکیب خصوصیات خوب جهت ساخت جوابهای تازه، در دست داشته باشند.
- عضویت در این مجموعه از جواب‌ها اغلب با در نظر گرفتن یک آستانه برای مقدار تابع هدف انجام می‌پذیرد.

پراکندگی و شدت افزایی-ادامه

- یکی از مهمترین مشکلات روش‌های مبتنی بر جستجوی محلی، گرایش این روشها به محلی‌بودن است.
- آنها تمایل دارند تا بیشتر در یک منطقه محدود به جستجو بپردازند.
- از نکات منفی که می‌توان در این روشها به آن رسید این است که هر چند ممکن است جوابهای خوبی پیدا شوند، اما ممکن است تمامی ناحیه جواب بررسی نشود و ممکن است بهینه‌ای پیدا شود که از جواب بهینه فعلی پیدا شده بسیار بهتر باشد.
- پراکندگی یا Diversification در TS یک مکانیسم است که کوشش می‌کند این مشکل را با انتقال جستجو به ناحیه‌های کمتر مورد بررسی قرار گرفته و یا قرار نگرفته حل کند.

پراکندگی و شدت افزایی-ادامه

■ دو روش عمده برای پراکندگی وجود دارد:

■ شروع دوباره

■ جستجو را از نقاطی که کمتر مورد استفاده قرار گرفته‌اند و یا از جواب بهینه‌ای که پیدا کرده است، دوباره شروع می‌کند (نقاط دیگر را در داخل لیست ممنوع قرار می‌دهد).

■ هدایت جستجو

■ قرار دادن مکانیسمی در فرآیند جستجو که کار رعایت پراکندگی را از همان ابتدا در دستور کار قرار می‌دهد.

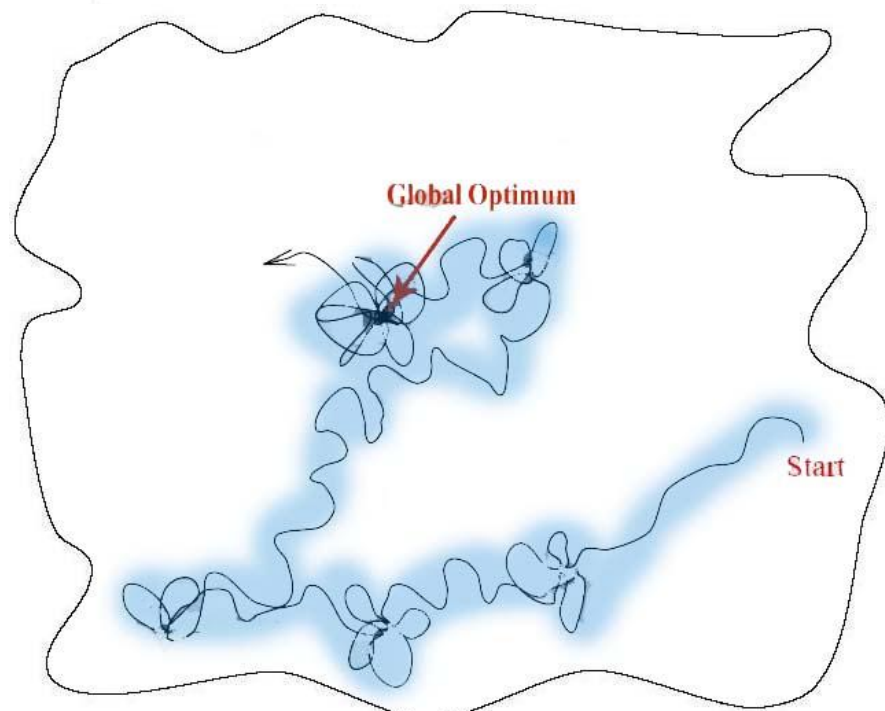
■ حافظه کوتاه مدت Short Term memory

- ساختار حافظه کوتاه مدت به طور معمول به بخشی از حافظه اطلاق می‌شود که یا برای پیاده سازی شدت افزایی به کار برده می‌شود و یا برای انجام LS مورد نیاز است. از آن جمله می‌توان حافظه اختصاص یافته به لیست ممنوع را، به عنوان حافظه‌ای که برای عدم تکرار بکار برده می‌شود، در این دسته قرار داد.

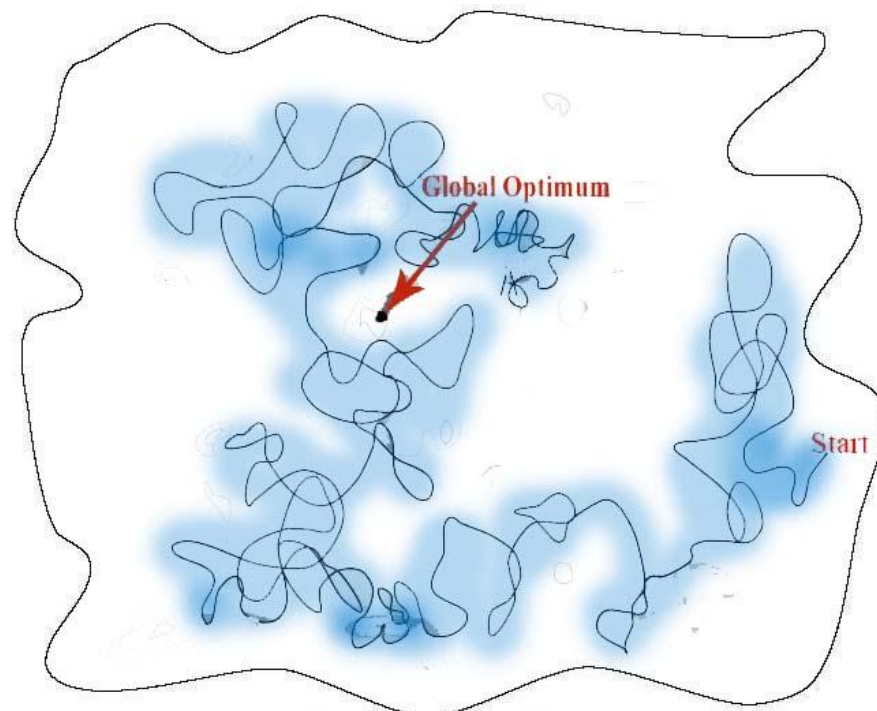
■ حافظه بلند مدت Long Term Memory

- حافظه بلند مدت با هدف جهت دادن به روند جستجو پیاده‌سازی می‌شود. به عبارت بهتر، معمولاً این حافظه پیاده‌سازی معیار پراکندگی است. تذکر این نکته لازم است که در TS برای افزایش پراکندگی جستجو و هدایت آن، معمولاً راه حل‌ها از یک مساله به مساله دیگر تفاوت می‌کند. یک روش، استفاده از روشی مشابه GLS (Guided Local Search) است.

تفاوت SA و TS



Tabu Search



Simulated Annealing

- معمولاً سریعتر به جواب بهینه همگرا می شود
- امکان قرار گرفتن در یک بهینه موضعی

- پراکندگی بیشتر

- سرعت همگرایی کمتر