بسمه تعالی

الگوریتم جستجوی ممنوعه

مسیریابی وسیله نقلیه

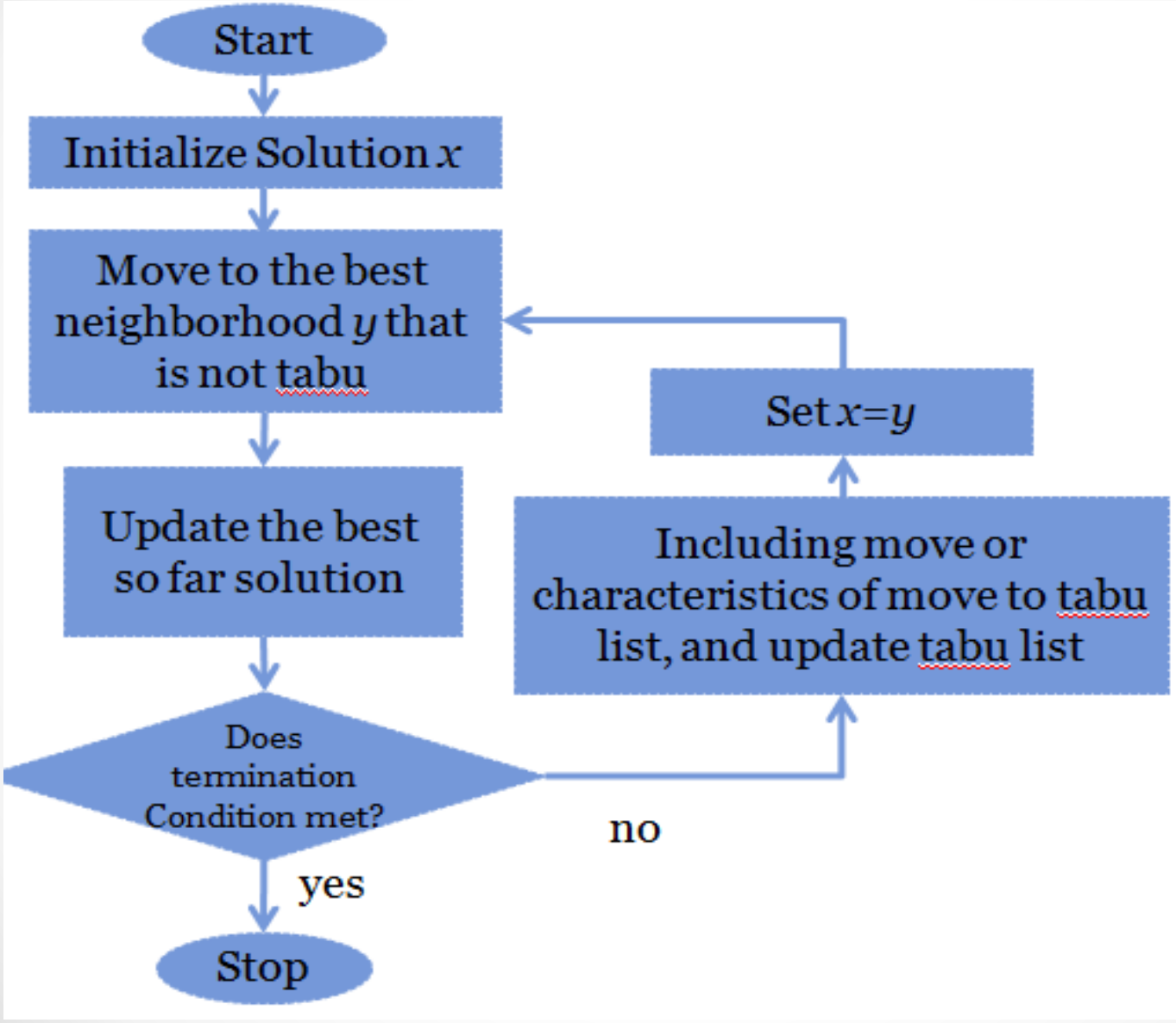
ساختار کلی جستجوی ممنوعه

- الگوریتم جستجوی ممنوع (TS) یک استراتژی جستجوی حافظه‌ای می‌باشد که برای اولین بار توسط گلوور در سال ۱۹۸۶ مطرح شده است.
- این الگوریتم تقریبا مانند الگوریتم های جستجوی محلی کار می‌کند، با این تفاوت که برای جلوگیری از دور و تسلسل در جواب‌ها و افتادن در دام جواب‌های بهینه محلی، از مفهومی به نام فهرست ممنوع استفاده می‌کند.

- جابه‌جایی از جواب جاری به جواب همسایه امکان‌پذیر زمانی انجام می‌شود که در فهرست تابو قرار نداشته باشد. در غیر اینصورت، جواب همسایه دیگری که در ارزیابی جواب‌های همسایه در رده بعدی قرار گرفته است، انتخاب شده و جابه‌جایی به آن صورت می‌گیرد.

- هرگاه ساختار همسایگی متقارن باشد، خطر افتادن در دور وجود دارد.

نمودار جریان الگوریتم جستجوی ممنوعه



حافظه الگوریتم می‌تواند از دو نوع recency و یا frequency باشد:

حافظه recency:

حافظه کوتاه مدت در روش جستجوی ممنوع نوعی از جستجوی فعال را جهت یافتن بهترین جواب ها (جواب‌هایی با بیشترین مطلوبیت) تشکیل می‌دهد و می‌توان این‌گونه بیان نمود که هسته اصلی جستجوی ممنوع، در فرایند کوتاه‌مدت مجسم می‌شود.

این حافظه لیستی با ابعاد N رکورد می باشد که N تا از آخرین حرکاتی را که الگوریتم با آن مواجه بوده است را به عنوان tabu نگهداری می‌کند.

حافظه frequency:

حافظه frequency که به عنوان حافظه بلندمدت شناخته می‌شود با اضافه نمودن اطلاعات تکمیلی دیگری از قبیل اینکه چند بار یک حرکت و یا جواب ممنوع جستجو شده است، مکمل حافظه recency می‌باشد.

در حالت کلی جستجوی ممنوع با در نظر گرفتن حافظه بلند مدت و استراتژی‌های مرتبط با آن قوی‌تر می‌شود.

لیست ممنوع یا Tabu

این لیست که دارای ابعادی ثابت یا متغیر می باشد، جابه‌جایی‌های منع شده را نگهداری می‌کند و کاربرد اصلی آن، پرهیز از همگرا شدن به جواب‌های بهینه محلی است. به عبارت دیگر، به کمک فهرست tabu جابه‌جایی به جواب‌هایی که اخیراً جستجو شده‌اند، ممنوع خواهد شد و فقط بخش‌هایی از مجموعه جواب که پیش از این مورد بررسی قرار نگرفته، مد نظر خواهند بود. نحوه ورود و خروج جواب‌ها به فهرست ممنوع به صورت FIFO است.

- مدت زمانی که یک جواب و یا یک حرکت در لیست می‌ماند بستگی به اندازه لیست ممنوع یا TLS دارد، TLS بزرگ به معنای این است که یک حرکت یا یک جواب به مدت بیشتری در لیست می‌ماند.
- TLS بزرگتر فرایند جستجو را محدود کرده و خیلی سریع آن را پایان می‌دهد.
- لیست Tabu دو نوع از اطلاعات recency و یک نوع از اطلاعات frequency را در خود ذخیره می‌کند.

- اطلاعات recency :

1. حرکاتی که انجام شده است.

ساختاری که این حرکات تحت آن نگهداری می‌شوندبه شکل زیر می‌باشد:

$$\{R1, node1, position1, R2, node2, position2\}$$

$R1$ و $R2$ دو مسیر تحت عملیات می‌باشند،$node\ 1$ گره‌ای است از $R1$ و $position\ 1$ موقعیت مکانی گره ۱ را نشان می‌دهد.

۲. جواب‌هایی که اخیرا ملاحضه شده‌اند.

این جواب ها به صورت یک رشته عدد صحیح کد می شوند.

مثال:

Route No 1: 0 – 3 – 2 – 4 – 5 – 0

Route No 2: 0 – 10 – 6 – 1 – 12 – 11 – 0

Route No 3: 0 – 9 – 8 – 7 – 0

رشته عدد صحیح:

(3 2 4 5 0 10 6 1 12 11 0 9 8 7 0)

- در نماد فوق عدد صفر به منظور جداسازی مسیرها از هم استفاده شده است که باعث می‌شود به راحتی یک رشته به یک جواب و بالعکس تبدیل شود.

- علاوه بر این کل هزینه هر جواب نیز به این رشته ضمیمه می‌شود.

- اطلاعات frequency

تعداد دفعاتی که یک حرکت و یا مسیر ممنوع، جستجو شده است.

این اطلاعات برای تعیین وضعیت فرایند جستجو مهم می‌باشد چرا که مقدار بزرگ برای آن نشان دهنده این است که فرایند جستجو در بهینه موضعی به دام افتاده است و جستجو باید خاتمه یابد.

لیست کاندید

- این لیست جواب‌های ممتازی را که در خلال فرایند جستجو کشف می‌شوند یا جواب هایی را که بهینه سراسری هستند در خود ذخیره می‌کند تا امکان جستجو در آینده برای آن ها فراهم باشد.
- این ذخیره یابی توسط رشته‌های صحیح به فرم قبل می‌باشد ولی رتبه بندی آنها در لیست بر اساس کل هزینه‌ای است که به رشته‌ها ضمیمه شده است.

- ممکن است برخی از این جواب‌های ممتاز توسط مسیرهای دیگر ملاقات شده و عنوان tabu را به خود بگیرند اما مدت زمان ماندن یک جواب در لیست کاندید بیشتر از لیست ممنوع می‌باشد.

فرآیند تقویت سازی

- فرآیند تقویت سازی با استفاده از روش جستجوی موضعی λ – $interchange$ انجام می شود.

- این فرایند زمانی پایان می‌یابد که کل همسایگی $N_2(s)$ مورد جستجو قرار گرفته و هیچ جواب بهتری یافت نشود.

روش جستجوی موضعی λ – interchange

- این روش برای اولین بار توسط عثمان و کریستوفایدز در سال ۱۹۸۹ مطرح شده است و بر مبنای تبادل مشتریان بین یک مجموعه از مسیرهای وسیله نقلیه می‌باشد.

$$S = \{R_1, \dots, R_p, \dots, R_q, \dots, R_k\}$$

فرض می‌کنیم S یک جواب موجه برای مساله باشد که در آن R_p مجموعه مشتریانی است که توسط مسیر وسیله نقلیه P سرویس می‌بینند.

$$S_1 \subset R_p \quad , \quad |S_1| \leq \lambda$$

$$S_2 \subset R_q \quad , \quad |S_2| \leq \lambda$$

زیرمجموعه های S_1 و S_2 بین دو مسیر جایجا می شوند.

$$\acute{R}_p = (R_p - S_1) \cup S_2$$

$$\acute{R}_q = (R_q - S_2) \cup S_1$$

$$\acute{S} = \{R_1, \dots, \acute{R}_p, \dots, \acute{R}_q, \dots, R_k\}$$

- بنابراین همسایگی $N_\lambda(S)$ برای جواب معین S برابر است باهمه همسایگی های $\{\acute{S}\}$ که به ازای یک λ مشخص حاصل شده اند.

- λ معمولا ۱ یا ۲ در نظر گرفته می شود.

- این روش قادر می‌سازد که، هم مشتریان بتوانند بین دو مسیر جابجا شوند و هم با یکدیگر بین دو مسیر مبادله شوند.

این روش دو استراتژی انتخاب دارد:

- ✓ اولین بهبود (FB)
- ✓ بهترین بهبود (GB)

به طور معمول استراتژی GB نتیجه بهتری را نسبت به FB تولید می‌کند.

فرآیند گوناگونی

بعد از یک چرخه از اجرای فرایند تقویت سازی، جواب فعلی S به عنوان جواب بهینه در همسایگی خودش در نظر گرفته می‌شود و سپس نیاز به یک گوناگونی در فرایند جستجو می‌باشد تا مناطق دیگری نیز جستجو شوند.

- با انجام یکسری از 2-interchange hop های رندوم یک جواب جدید S ایجاد می شود.

- جواب جدید در صورتی که tabu نباشد، در لیست کاندید ذخیره شده و بر اساس رتبه بندی هزینه برای تقویت‌سازی در انتظار می‌ماند .

- بعد از تعدادی تکرار فرایند گوناگونی خاتمه یافته و یک جواب با حداقل هزینه که tabu نمی‌باشد از لیست کاندید انتخاب می‌شود و جواب فعلی با این جواب به‌روز رسانی می‌شود.

- بر پایه این جواب جدید یک چرخه جدید از فرایندهای گوناگونی و تقویت سازی راه‌اندازی می‌شود.

- در پایان برای اطمینان بیشتر، روی جواب نهایی یک رویه 2-interchange (GB) صورت می‌گیرد.

شرط توقف:

- ✓ تعدادی تکرار معین از انجام الگوریتم گذشته باشد.
- ✓ گوناگونی ناموفق باشد، به این معنی که جهش موجهی یافت نشود.

روش اجرای الگوریتم

قدم اول: ایجاد یک جواب اولیه و به روز رسانی بهترین جواب سراسری با آن

قدم دوم: تشکیل لیست حرکات ممنوع، لیست جواب‌های ممنوع و لیست کاندید و وارد کردن جواب فعلی در لیست ممنوع

قدم سوم: انجام فرایند تقویت سازی با روش 2-interchange (GB)

قدم چهارم: انجام فرایند گوناگونی

قدم پنجم: اگر گوناگونی صورت نگرفته است و یا تعداد کل تکرارها کمتر از ماکزیمم تکرارها باشد به قدم سوم بروید.

قدم ششم: در غیر این صورت بهترین جواب سراسری بدست آمده تاکنون را توسط رویه 2-interchange (GB) بهبود بخشیده و جواب بهبودیافته را برگردانید.