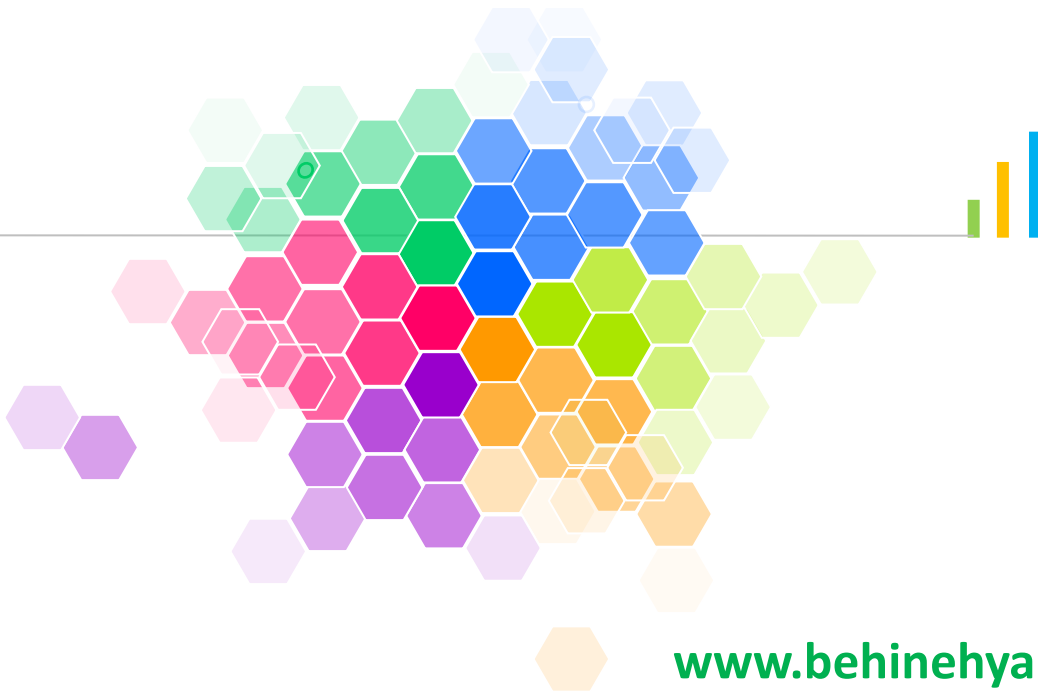


به نام خدا



درس ۱۵: الگوریتم فراابتکاری جستجوی ممنوع



TABOO

فهرست مطالب



مقدمه الگوریتم فراابتکاری جستجوی ممنوع

۱

اصول الگوریتم جستجوی ممنوع

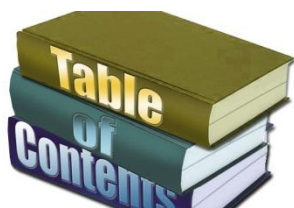
۲

گام های الگوریتم جستجوی ممنوع

۳

حل یک مثال

۴



مقدمه الگوریتم فراابتکاری جستجوی ممنوع

مقدمه

محاسبه راه حل بهینه *Optimal solution* برای اکثر مسایل بهینه سازی که در خیلی از زمینه های کاربردی و عملی مشاهده می گردند، کار دشوار و سختی است. در عمل، معمولاً به راه حل های خوب که از الگوریتم های هیوریستیک *Heuristic* یا متاهیوریستیک (همان فراابتکاری) *Metaheuristic* بدست می آید، اکتفا می گردد.

مقدمه الگوریتم فراابتکاری جستجوی ممنوع

روش جستجوی متاهیوریستیک را می‌توان به صورت متدولوژی‌های عمومی سطح بالایی که می‌توانند به عنوان یک استراتژی راهنما در طراحی هیوریستیک‌های اختصاصی برای حل مسایل بهینه‌سازی تخصصی به کار روند، تعریف کرد.

برخلاف روش‌های دقیق، متاهیوریستیک‌ها (فراابتکاری‌ها) برای مسایل با اندازه‌های بزرگ کاربرد دشوار دارد و راه‌حلهایی راضی‌کننده‌ای در زمان معقولی ارائه می‌نمایند. در این الگوریتم‌ها، هیچ گونه ضمانتی برای یافتن جواب بهینه سراسری یا حدودی از آن وجود ندارد.

مقدمه الگوریتم فراابتکاری جستجوی ممنوع

فراابتکاری‌ها در خیلی از زمینه‌ها از قبیل موارد زیر کاربرد دارند:

- ✓ طراحی مهندسی، بهینه سازی توپولوژی، بهینه سازی مسایل الکترونیک، آئرو دینامیک، دینامک سیالات، مخابرات، رباتیک
- ✓ یادگیری ماشین و کاوش داده‌ها
- ✓ مدل سازی سیستم‌ها، شبیه سازی و تحقیق در شیمی، فیزیک، بیولوژی، کنترل، سیگنال، و پردازش تصویر

مقدمه الگوریتم فراابتکاری جستجوی ممنوع



روش‌های مختلف الگوریتم‌های فراابتکاری یا متاهوریستیک تا به حال پیشنهاد شده است که

به صورت زیر است:

✓ بهینه سازی کلونی مورچگان *Ant colony optimization*

✓ بهینه سازی کلونی زنبوران *Bee colony*

✓ الگوریتم ژنتیک *Genetic algorithm*

✓ جستجوی محلی تکراری *Iterated local search*

✓ بهینه سازی توده ذرات *particle swarm intelligent*

✓ انجماد تدریجی *Simulated Annealing*

مقدمه الگوریتم فراابتکاری جستجوی ممنوع

در طراحی یک فراابتکاری، دو معیار متناقض شامل **کاوش** (*Exploration*) و در فضای جستجو (گوناگونی و تنوع) و **تبعیت** (*Exploitation*) از بهترین راه حل‌های پیدا شده، باید در نظر گرفته شوند.

در **کاوش** در ناحیه‌های جستجو نشده بررسی صورت می‌گیرد. در **تبعیت**، در ناحیه‌های امید بخش که تا به حال در آن ناحیه یک جواب خوب پیدا شده است بررسی بیشتر صورت می‌گیرد.

مقدمه الگوریتم فراابتکاری جستجوی ممنوع

معیارهای طبقه بندی زیادی ممکن است برای طبقه بندی فراابتکاری‌ها استفاده شود که در زیر به بعضی از آن‌ها اشاره می‌شود:

الهام گرفته از طبیعت در مقابل عدم الهام از طبیعت

نحوه استفاده از حافظه

قطعی در مقابل احتمالی

تکراری در مقابل حریصانه

اصول الگوریتم جستجو ممنوع

الگوریتم جستجو ممنوع

الگوریتم فراابتکاری جستجوی ممنوع را می‌توان به عنوان یک استراتژی جستجوی محلی در نظر گرفت. این جستجو شامل حرکت از یک محل (همان جواب یا نقطه) به راه حل دیگری در همسایگی او مطابق با بعضی از قواعد تعریف شده می‌باشد.

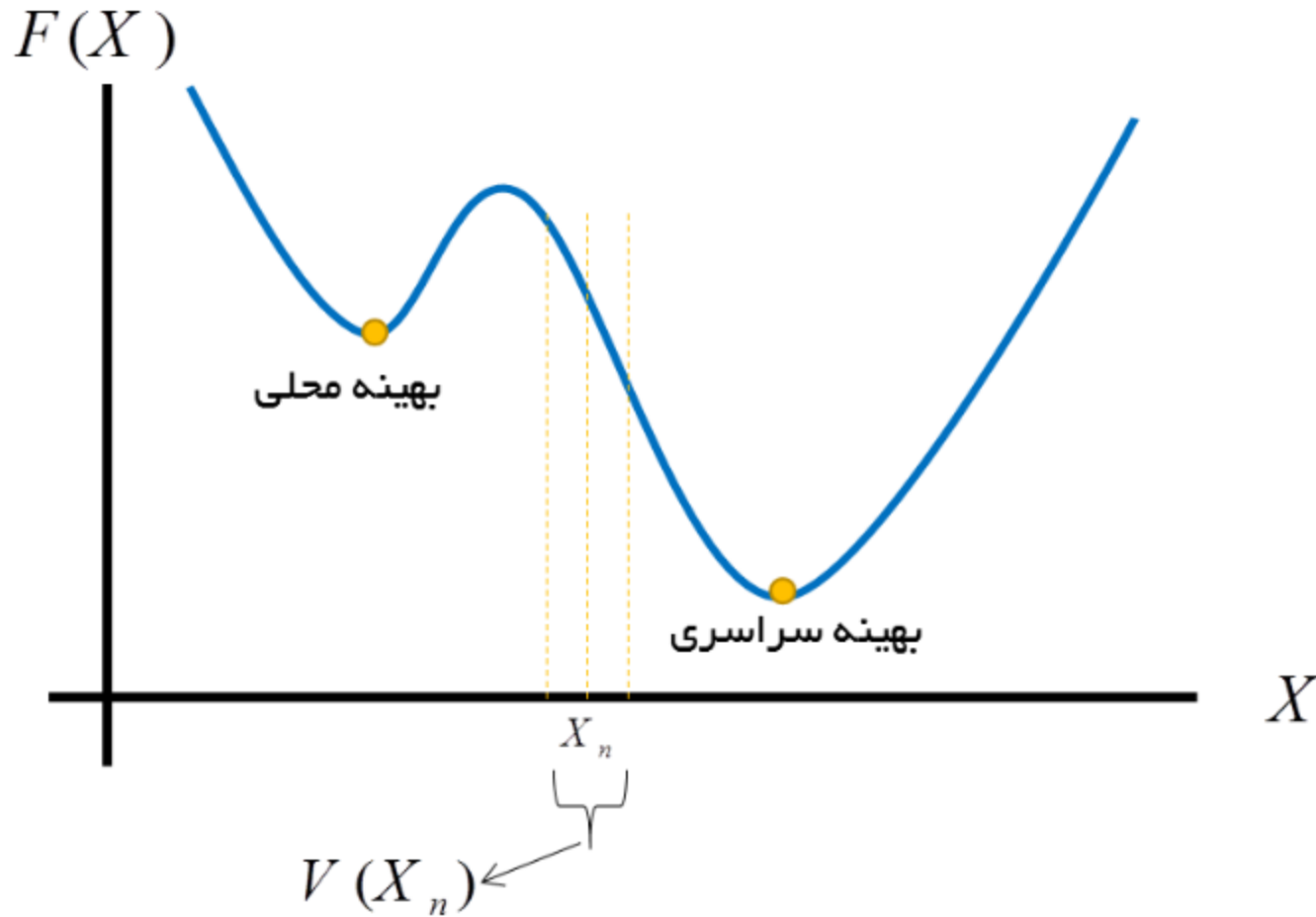
برای مثال، مسئله حداقل سازی یک تابع همچون $F(x)$ روی یک مجموع محدود از نقاط X را در نظر بگیرید. روش جستجوی ممنوع از یک راه حل اختیاری $x_1 \in X$ شروع شده و در هر مرحله، به جستجوی همسایگی می‌پردازد.

اصول الگوریتم جستجو ممنوع

در این جستجو همسایگی، به هر $x \in X$ یک زیر مجموعه $V(x) \subset X$ اختصاص داده می‌شود که همسایه x نامیده می‌شود. در مرحله n ، یک راه حل جدید x_{n+1} در همسایگی $V(x_n)$ از راه حل جاری x_n را انتخاب می‌نماید. معمولاً این راه حل بهترین راه حل در بین همسایه‌ها بوده و دارای شرط زیر می‌باشد.

$$F(x_{n+1}) \leq F(x) \quad \forall x \in V(x_n)$$

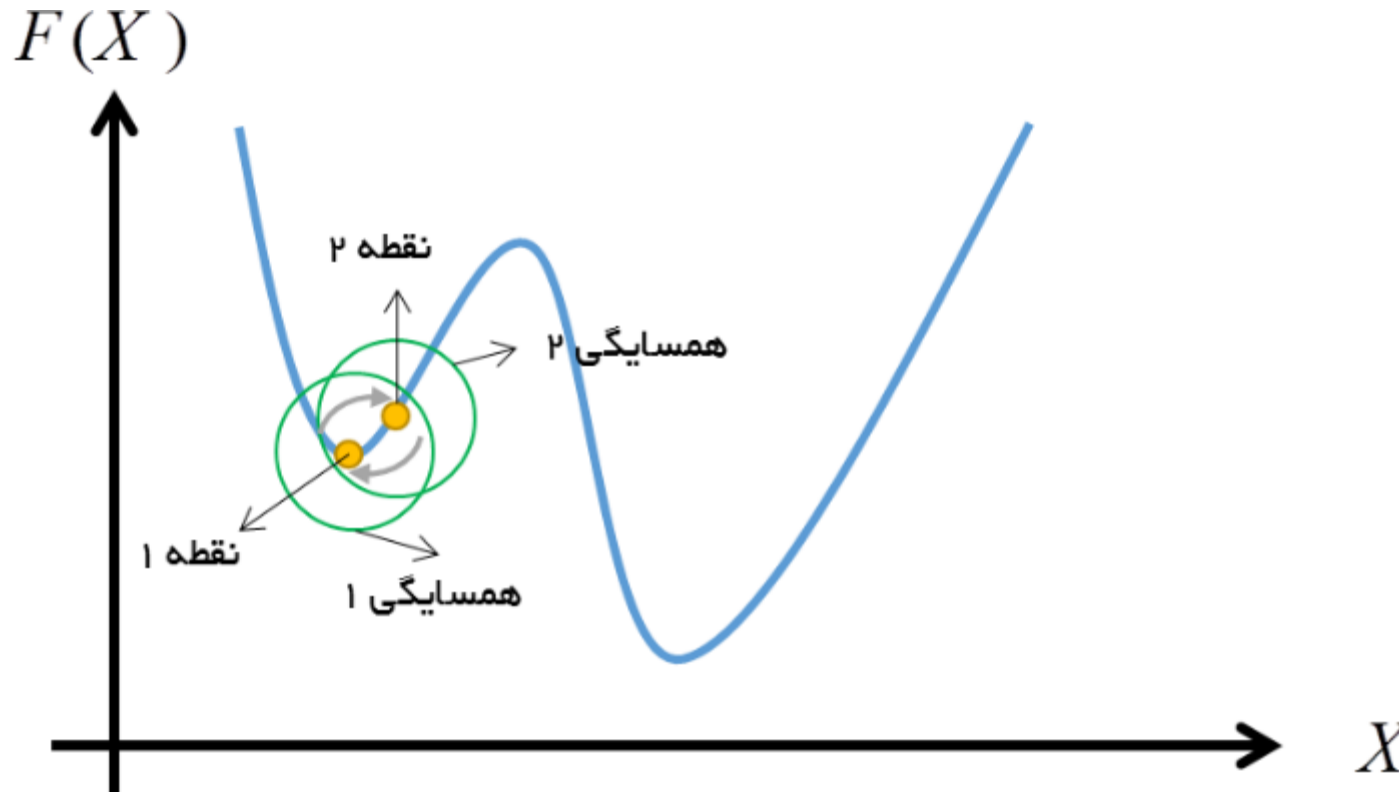
اصول الگوریتم جستجو ممنوع



اصول الگوریتم جستجو ممنوع

ممکن است به هنگام حرکت از یک نقطه به نقطه دیگر مشکلی پیش آید که به نام **افتادن در**

بهینه محلی معروف است. شکل زیر را در نظر بگیرید.



اصول الگوریتم جستجو ممنوع

الگوریتم جستجوی ممنوع برای جلوگیری از این مشکل از ابزار **لیست ممنوع** استفاده می‌کند. در هر مرحله، در هنگام حرکت از یک نقطه به نقطه دیگر، مشخصاتی از حرکت (برای مثال فاصله و جهت) را به حافظه سپرده و در نقطه جدید از انجام حرکتی که منجر به برگشت به عقب می‌شود خودداری می‌کند. به منظور کارایی بیشتر، به جای یک حرکت قبلی، مجموعه‌ای از حرکات‌های قبلی را به حافظه سپرده و آن‌ها در لیستی به نام **لیست ممنوع** ذخیره می‌نماید. این لیست پویا بوده و در طول الگوریتم به هنگام می‌گردد.

اصول الگوریتم جستجو ممنوع

قاعده توقف پایان الگوریتم جستجو را تعیین می کند. یک قاعده ساده می توان به صورت محدودیتی برای کل تعداد حرکات باشد. به عبارت دیگر، وقتی تعداد حرکات به عدد خاصی رسید الگوریتم متوقف گردد.



مراحل الگوریتم جستجوی ممنوع

قدم اول: تنظیمات الگوریتم

ساختار ساده الگوریتم جستجو ممنوع، در هنگام استفاده برای حل یک مسئله بهینه سازی، نیاز به تنظیمات خاصی دارد. در صورتی که این تنظیمات به صورت خوبی انجام شود، الگوریتم دارای کیفیت بالایی خواهد بود و در غیر این صورت کیفیت الگوریتم در یافتن جواب بهینه کاهش می یابد. یکی از این تنظیمات، نحوه رد کردن جواب یا راه حل‌ها می باشد. یکی از دیگر تنظیماتی که باید انجام گردد، تعریف ساختار همسایگی می باشد. ساختار همسایگی نحوه بدست آوردن همسایه‌ها و تعداد آن‌ها را برای هر راه حل مشخص می نماید.

مراحل الگوریتم جستجوی ممنوع

قدم دوم: تشکیل یک راه حل اولیه

الگوریتم جستجوی ممنوع با یک راه حل اولیه شروع می‌شود. در خیلی از مواقع، راه حل اولیه به صورت تصادفی انتخاب می‌شود. در صورتی که ساختار جواب ساده باشد این کار به سادگی انجام می‌شود اما در خیلی از مسئله‌ها به دلیل پیچیدگی فضای جواب و نحوه کدگذاری آن، ایجاد یک جواب اولیه امکان پذیر به صورت تصادفی، کار ساده‌ای نیست و در برخی اوقات نیاز به الگوریتم‌های کناری برای یافتن جواب اولیه مناسب و امکان پذیر است.

مراحل الگوریتم جستجوی ممنوع

قدم سوم: محاسبه مقدار تابع هدف

در این مرحله، مقدار تابع هدف یا تابع برازندگی برای راه حل اولیه در نظر گرفته شده، محاسبه می‌گردد. در شروع الگوریتم مقدار به دست آمده از جواب اولیه به عنوان مقدار بهینه نیز در نظر گرفته می‌شود. این مقدار در طول الگوریتم به هنگام می‌گردد. همچنین، نقطه اولیه به عنوان راه حل جاری نیز مدنظر قرار می‌گیرد.

مراحل الگوریتم جستجوی ممنوع

قدم چهارم: جستجوی همسایگی

در هر مرحله الگوریتم، جستجوی همسایگی به جستجوی همسایه‌های جواب جاری می‌پردازد. در این مرحله، با توجه به ساختار معرفی شده برای همسایگی، کلیه همسایه‌های راه حل جاری مورد بررسی قرار می‌گیرند.

قدم پنجم: یافتن بهترین همسایه

در جستجوی همسایگی، کلیه همسایه بررسی می‌گردد. در بین همسایه‌ها، بهترین همسایه انتخاب می‌گردد.

مراحل الگوریتم جستجوی ممنوع

قدم ششم: ارزیابی ممنوعیت حرکت

بعد از انتخاب بهترین همسایه، حرکت به آن راه حل یا لیست ممنوع مقایسه می‌گردد. در صورتی که این حرکت جز لیست ممنوع باشد، به **قدم هفتم** رفته و در صورتی که حرکت ممنوع نباشد، الگوریتم به **قدم هشتم** خواهد رفت

قدم هفتم: معیار آرمانی

در این قدم، حرکت ممنوع شده با معیار آرمانی مقایسه می‌گردد. در صورتی که معیار آرمانی برآورده شده باشد، حرکت انجام شده و در صورتی معیار آرمانی برآورده نشود، الگوریتم به **قدم پنجم** بر می‌گردد و بهترین راه حل بعدی را انتخاب می‌کند.

مراحل الگوریتم جستجوی ممنوع



قدم هشتم: انجام حرکت

در این قدم، راه حل جاری جدید به عنوان راه حل جاری در نظر گرفته می‌شود. به عبارت دیگر، از راه حل جاری به راه حل جدید حرکت می‌شود و راه حل جدید به عنوان راه حل جاری شناخته می‌شود.

مراحل الگوریتم جستجوی ممنوع

قدم نهم: بهنگام سازی بهترین جواب

در فرایند الگوریتم، همواره بهترین جواب ذخیره می‌گردد. در ابتدا نقطه اولیه، به عنوان بهترین جواب در نظر گرفته می‌شود و هر زمان که حرکتی انجام می‌گردد، جواب جدید با بهترین جوابی که تا آن لحظه به دست آمده است، مقایسه می‌گردد. در صورتی که راه حل جدید راه حل بهتری باشد، راه حل جدید به عنوان بهترین راه حل ذخیره می‌شود و در غیر این صورت، بهترین راه حل تغییر نمی‌کند.

مراحل الگوریتم جستجوی ممنوع

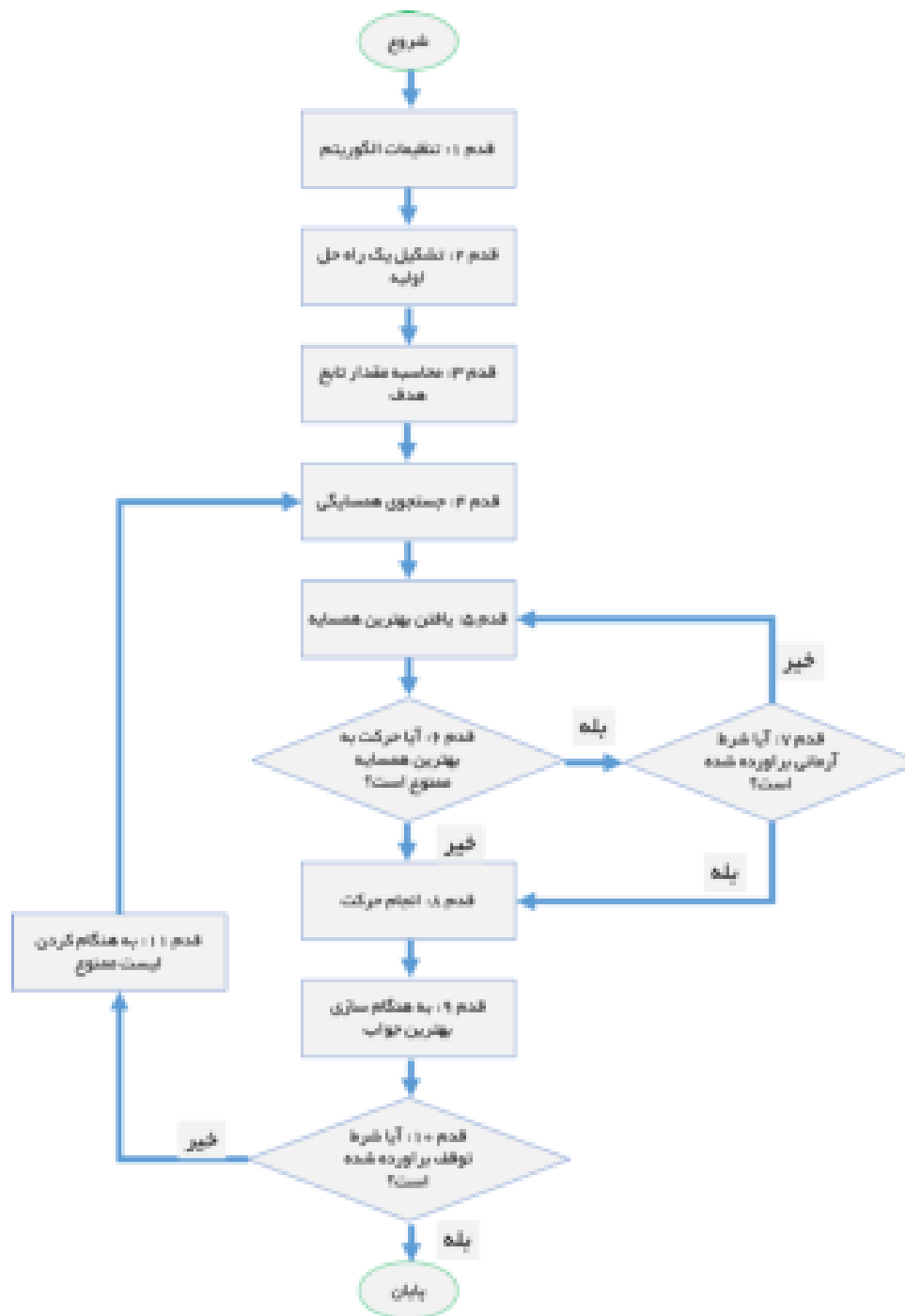
قدم دهم: شرط توقف

بعد از هر حرکت، شرط توقف مورد ارزیابی قرار می‌گیرد. در صورتی که شرط توقف برآورده شود، الگوریتم پایان یافته و نتایج آن گزارش می‌شود و در غیر این صورت، به **قدم یازده** منتقل خواهد شد.

مراحل الگوریتم جستجوی ممنوع

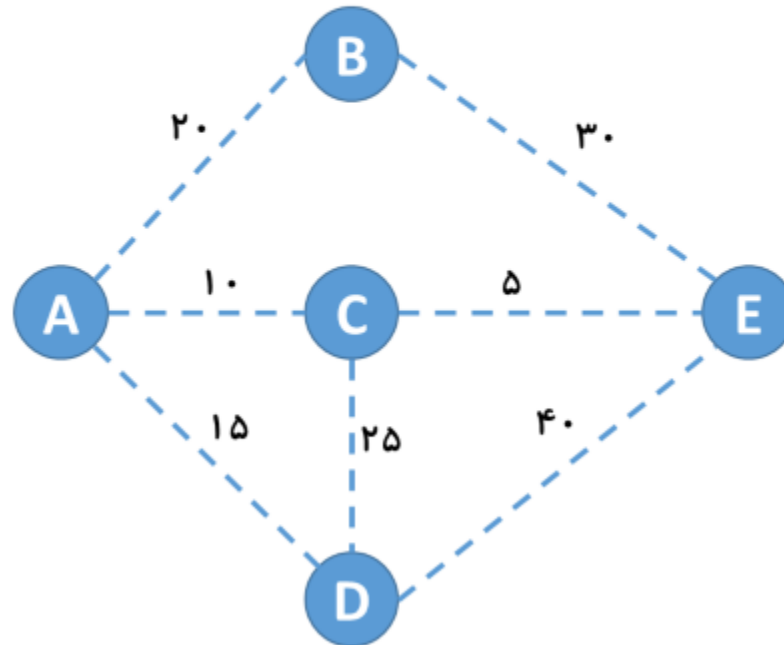
قدم یازدهم: بهنگام سازی لیست ممنوع

بعد از هر حرکت، مشخصه حرکت جدید وارد لیست ممنوع می‌گردد و سپس لیست ممنوع به هنگام می‌گردد. براساس این قاعده، در زمان انتقال یک حرکت به لیست ممنوع، ممکن است یک یا چند حرکت قدیمی از لیست ممنوع حذف گردد. بعد از بهنگام سازی لیست ممنوع، الگوریتم به **قدم چهارم** رفته و به جستجوی همسایگی در اطراف راه حل جدید می‌پردازد.



مثال

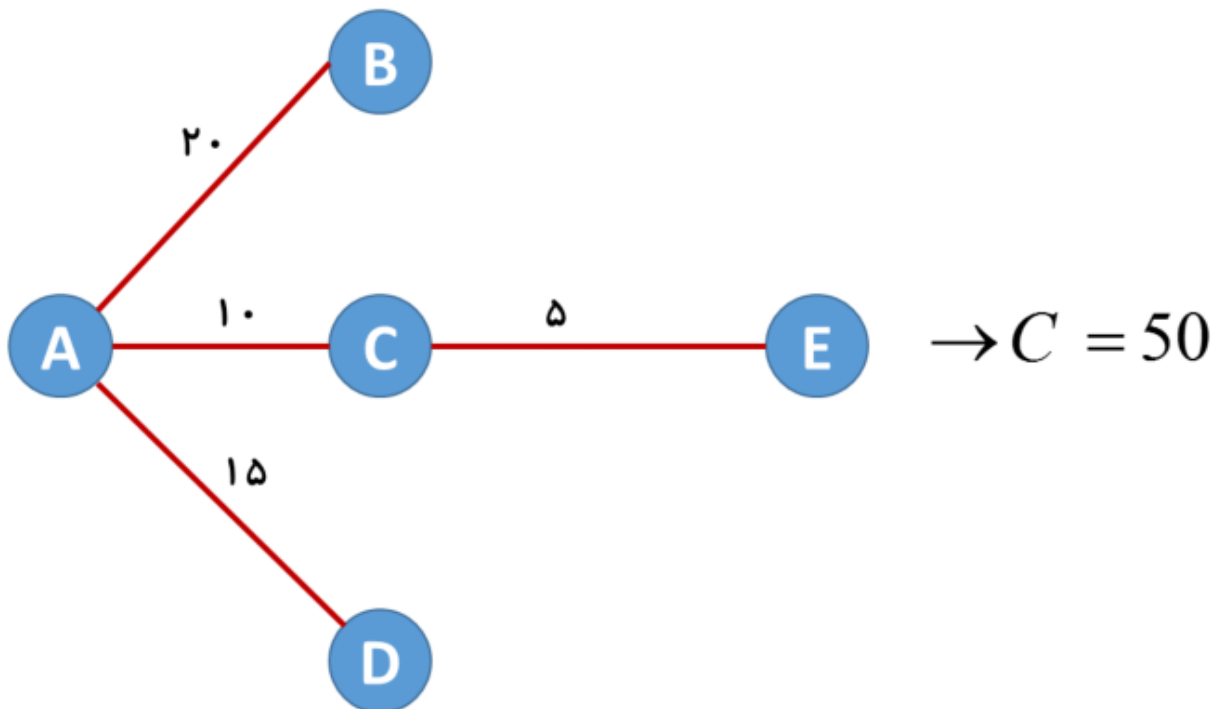
تمرین: درخت پوشای کمینه برای شبکه زیر را با استفاده از الگوریتم جستجوی ممنوع بیابید.



مثال

درخت پوشای کمینه در صورتی که محدودیتی در این مسئله وجود نداشته باشد به صورت زیر

می شود.



مثال

در این مسئله دو محدودیت زیر در نظر گرفته می شود.

محدودیت ۱: یال AD زمانی می تواند در جواب بهینه حضور داشته باشد که یال DE هم وجود

داشته باشد.

محدودیت ۲: یکی از سه یال AD ، CD و AB می تواند در شبکه نهایی حضور داشته باشد.



محدودیت ۱: ایجاد هزینه پنالتی ۱۰۰ اگر محدودیت ۱ نقض شود.

محدودیت ۲: ایجاد هزینه پنالتی ۱۰۰ در صورتی که از سه یال AD ، CD و AB در جواب وجود

داشته باشد و در صورتیکه هر سه یال در جواب حضور داشته باشد، هزینه جریمه برابر ۲۰۰ می

شود.

مثال

برای حل این مسئله با استفاده از روش جستجو ممنوع باید موارد زیر در نظر گرفته شود.

۱- فرایند جستجوی محلی: جواب ها از اضافه و کم کردن یال ها ایجاد می شود به شرطی که دور ایجاد نشود.

۲- حرکت به همسایگی: در این قسمت، به جواب هایی حرکت می کنیم که تا آنجا که امکان دارد محدودیت ها را نقض نکند و کمترین هزینه را داشته باشد.

۳- لیست ممنوعه: در این حالت کمان هایی که نباید حذف شوند قرار گیرد.

۴- به هنگام سازی لیست تابو: وقتی در یک مرحله یک یال به شبکه اضافه می شود این لینک وارد لیست تابو می شود.

۵- حداکثر اندازه لیست: اندازه این لیست برابر با ۲ (دو) در نظر گرفته می شود.

۶- قاعده توقف: توقف در الگوریتم زمانی رخ می دهد که پس از ۳ (سه) تکرار تغییری در

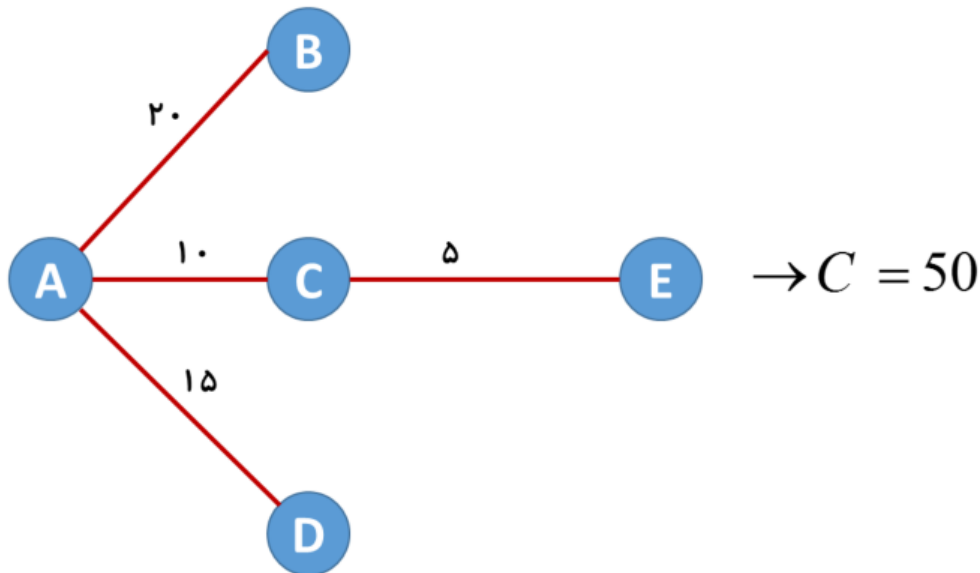
بهترین تابع هدف رخ ندهد.

مثال

تکرار ۱:

در ابتدای حل مسئله نیاز است که با یک جواب اولیه کار را آغاز کنیم. جواب اولیه زیر انتخاب

می شود که همان جواب روش ارایه شده از روش بهینه سازی درخت پوشای ارایه شده در درس ۷ است.



مثال

سه گزینه برای اضافه کردن لینک به شبکه بالا وجود دارد که عبارتند از $BE/CD/DE$.

اگر لینک BE اضافه شود دور $BE-CE-AC-AB$ ایجاد می شود و برای جلوگیری از این دور می توان یکی از سه لینک CE ، AC و AB را حذف کرد (تا حالا در لیست تابو لینکی اضافه نشده است).

اگر لینک CE حذف شود و لینک BE اضافه شود، هزینه کل ۲۵ واحد افزایش می یابد ($۲۵=۵-$).

و هزینه پناستی ۲۰۰ واحد باقی می ماند و لذا کل هزینه از ۲۵۰ به ۲۷۵ افزایش می یابد.

مثال

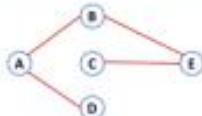
اگر لینک AB حذف شود هزینه لینک ها $30 - 20 = 10$ تغییر کرده و هزینه پناستی از ۲۰۰ به ۱۰۰ کاهش می یابد و لذا هزینه کل برابر با $160 = 100 + 10 + 50$ خواهد شد.

اگر لینک AC حذف شود، محدودیت های ۱ و ۲ نقض می شوند و کل هزینه برابر با ۲۷۰ می شود.

مثال

اگر لینک CD انتخاب شود که باید به شبکه اضافه شود، در این صورت سیکل $CD-AD-AC$ ایجاد می شود و لینک های AD و AC جز گزینه های حذف خواهد بود. در صورتیکه لینک AC حذف شود، محدودیت ۱ و محدودیت ۲ نقض خواهد. ولی در صورت حذف کردن لینک AD محدودیت ۱ برآورده می شود و فقط محدودیت ۲ نقض می شود.

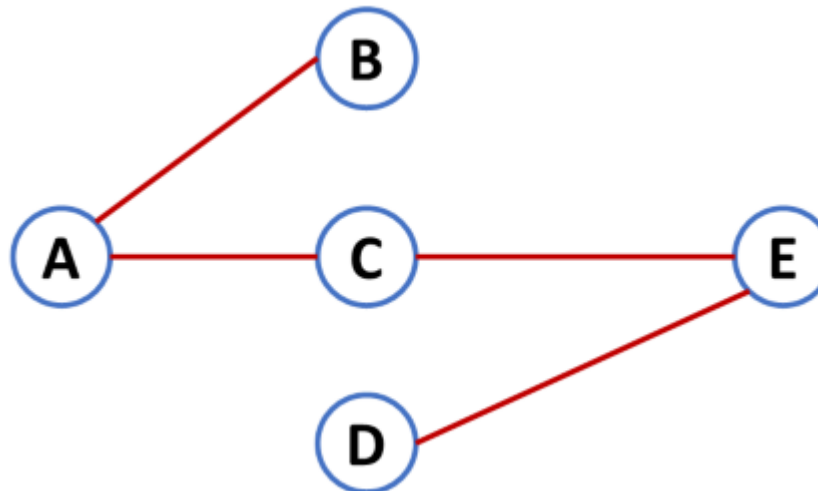
اگر لینک DE به شبکه اضافه شود آنگاه دور $DE-CE-AC-AD$ ایجاد می شود و می توان لینک های AD و AC, CE را حذف کرد. حذف هر یکی از این سه لینک باعث برقراری محدودیت ۱ خواهد شد ولی حذف لینک AD ، محدودیت ۲ را هم برقرار کند و که در این حالت هزینه کل شبکه برابر با $75 = (40 - 15) + 50$ می شود.

هزینه	شبکه	حذف کردن	اضافه کردن
محدودیت ۱ نقض می شود. محدودیت ۲ نقض می شود. $۲۰۰+۷۵=۲۷۵$		CE	BE
محدودیت ۱ نقض می شود. محدودیت ۲ نقض می شود. $۲۰۰+۷۰=۲۷۰$		AC	BE
محدودیت ۱ نقض می شود. محدودیت ۲ نقض نمی شود. $۱۰۰+۶۰=۱۶۰$		AB	BE
محدودیت ۱ نقض نمی شود. محدودیت ۲ نقض می شود. $۱۰۰+۶۰=۱۶۰$		AD	CD
محدودیت ۱ نقض می شود. محدودیت ۲ نقض می شود. $۳۰۰+۶۵=۳۶۵$		AC	CD
محدودیت ۱ نقض نمی شود. محدودیت ۲ نقض می شود. $۱۰۰+۸۵=۱۸۵$		CE	DE
محدودیت ۱ نقض نمی شود. محدودیت ۲ نقض می شود. $۱۰۰+۸۰=۱۸۰$		AC	DE
محدودیت ۱ نقض نمی شود. محدودیت ۲ نقض نمی شود. $۰+۷۵=۷۵$		AD	DE



مثال

با توجه به جدول فوق، جواب فعلی به جواب همسایگی زیر حرکت می‌کند و لینک DE به لیست تابو اضافه می‌شود و لذا لیست تابو به صورت $\{DE\}$ می‌شود.



هزینه	شبکه	حذف کردن	اضافه کردن
حرکت غیرمجاز به علت اینکه لینک DE جز لیست تابو است.		DE*	AD
محدودیت ۱ نقض نمی شود. محدودیت ۲ نقض می شود. $100 + 85 = 185$		CE	AD
محدودیت ۱ نقض نمی شود. محدودیت ۲ نقض می شود. $100 + 80 = 180$		AC	AD
محدودیت ۱ نقض نمی شود. محدودیت ۲ نقض نمی شود. $0 + 100 = 100$		CE	BE
محدودیت ۱ نقض نمی شود. محدودیت ۲ نقض نمی شود. $0 + 95 = 95$		AC	BE
محدودیت ۱ نقض نمی شود. محدودیت ۲ نقض نمی شود. $0 + 85 = 85$ کمینه **		AB	BE
حرکت غیرمجاز به علت اینکه لینک DE جز لیست تابو است.		DE*	CD
محدودیت ۱ نقض نمی شود. محدودیت ۲ نقض می شود. $100 + 95 = 195$		CE	CD

تکرار ۲

$$TL = \{DE\} \cup \{BE\}$$

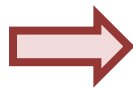


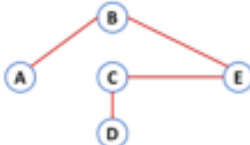
هزینه	شبکه	حذف کردن	اضافه کردن
حرکت ممنوعه	چون BE کمان در لیست تابو است این یک حرکت ممنوعه است.	BE	AB
محدودیت ۱ نقض نمی شود. محدودیت ۲ نقض نمی شود. $100+0=100$		CE	AB
محدودیت ۱ نقض نمی شود. محدودیت ۲ نقض نمی شود. $0+95=95$		AC	AB
محدودیت ۱ نقض می شود. محدودیت ۲ نقض نمی شود. $60+100=160$		DE	AD
محدودیت ۱ نقض نمی شود. محدودیت ۲ نقض نمی شود. $0+95=95$		CE	AD
محدودیت ۱ نقض نمی شود. محدودیت ۲ نقض نمی شود. $0+90=90$		AC	AD
محدودیت ۱ نقض نمی شود. محدودیت ۲ نقض نمی شود. $0+70=70$ کمیته		DE	CD
محدودیت ۱ نقض نمی شود. محدودیت ۲ نقض نمی شود. $105+0=105$		CE	CD

تکرار ۳

$$TL = \{DE, BE\} \cup \{CD\} - \{DE\}$$

$$TL = \{CD, BE\}$$



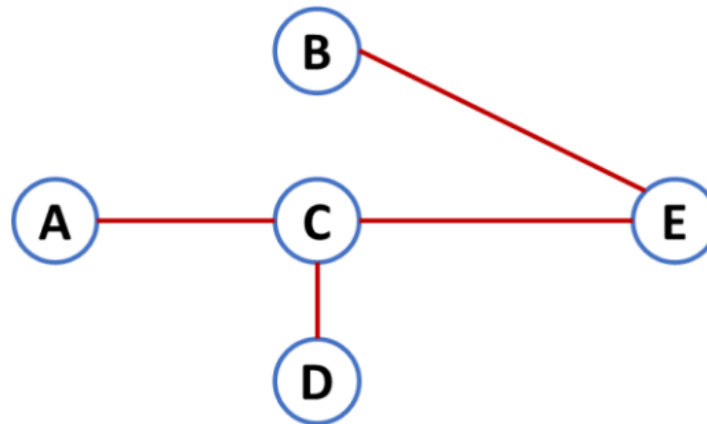
اضافه کردن	حذف کردن	شبکه	هزینه
AB	AC		محدودیت ۱ نقض نمی شود. محدودیت ۲ نقض می شود. $۸۰ + ۱۰۰ = ۱۸۰$
AB	CE		محدودیت ۱ نقض نمی شود. محدودیت ۲ نقض می شود. $۱۰۰ + ۸۵ = ۱۸۵$
AD	AC		محدودیت ۱ نقض می شود. محدودیت ۲ نقض می شود. $۱۰۰ + ۱۰۰ + ۷۵ = ۲۷۵$
AD	CD*		محدودیت ۱ نقض می شود. محدودیت ۲ نقض نمی شود. $۶۰ + ۱۰۰ = ۱۶۰$
DE	CD*		محدودیت ۱ نقض نمی شود. محدودیت ۲ نقض نمی شود. $۰ + ۸۵ = ۸۵$ کمینه **
DE	CE		محدودیت ۱ نقض نمی شود. محدودیت ۲ نقض نمی شود. $۰ + ۱۰۵ = ۱۰۵$

تکرار ۴

$$TL = \{CD, BE\} \cup \{DE\} - \{BE\}$$

$$TL = \{CD, DE\}$$





با تشکر

راه های ارتباطی با ما

www.behinehyab.com

behinehyab@gmail.com