

درس 18:

الگوریتم فراابتکاری الگوریتم اجتماع مورچگان

تهیه شده توسط گروه بهینه یاب



مقدمه

محاسبه راه حل بهینه *Optimal solution* برای اکثر مسایل بهینه سازی که در خیلی از زمینه‌های کاربردی و عملی مشاهده می‌گردند، کار دشوار و سختی است. در عمل، معمولاً به راه حل‌های خوب که از الگوریتم‌های هیوریستیک *Heuristic* یا متاهیوریستیک (همان فراابتکاری) *Metaheuristic* بدست می‌آید، اکتفا می‌گردد. متاهیوریستیک‌ها مجموعه‌ای از فنون بهینه سازی تقریبی *Approximate optimization techniques* را که عمدتاً در طول دو دهه گذشته شهرت پیدا کرده‌اند، در بر می‌گیرند. روش‌های فراابتکاری راه حل‌های قابل قبول در زمان معقول را برای مسایل پیچیده و سخت، در زمینه‌های مهندسی و علوم پایه ارایه می‌نمایند. برخلاف الگوریتم‌های بهینه سازی دقیق *Exact optimization algorithms*، فراابتکاری‌ها بهینه بودن جواب‌های بدست آمده را ضمانت نمی‌نمایند.

کلمه متاهیوریستیک از کلمه یونانی "*Heuriskein*" به معنای هنر کشف قواعد جدید برای حل مسایل گرفته شده است. پیشوند "متا" نیز از یک کلمه یونانی به معنای "متدولوژی" سطح بالا گرفته شده است. واژه "متاهیوریستیک" اولین بار توسط گلوور در سال ۱۹۸۶ ارایه گردید. 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 بهینه سازی تخصصی به کار روند، تعریف کرد.

برخلاف روش‌های دقیق، متاهیوریستیک‌ها (فراابتکاری‌ها) برای مسایل با اندازه‌های بزرگ کاربرد دشوار دارد و راه حل‌هایی راضی کننده‌ای در زمان معقولی ارایه می‌نمایند. در این الگوریتم‌ها، هیچ گونه ضمانتی برای یافتن جواب بهینه سراسری یا حدودی از آن وجود ندارد. متاهیوریستیک‌ها (فراابتکاری‌ها) در طول بیست سال گذشته شهرت زیادی پیدا کرده‌اند. کاربرد و استفاده از آن‌ها در خیلی از مسایل، کارایی و اثر بخشی آن‌ها را برای حل مسایل پیچیده و بزرگ نشان می‌دهد. فراابتکاری‌ها در خیلی از زمینه‌ها از قبیل موارد زیر کاربرد دارند:

✓ طراحی مهندسی، بهینه سازی توپولوژی، بهینه سازی مسایل الکترونیک، آئودینامیک،
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
- ✓ مسایل مسیردهی و برنامه ریزی، برنامه ریزی ربات، مسایل تولید و زمان بندی، حمل و نقل و لجستیک، مدیریت زنجیره تامین
- روش‌های مختلف الگوریتم‌های فراابتکاری یا متاهیوریستیک تا به حال پیشنهاد شده است که به صورت زیر است:

- ✓ بهینه سازی کلونی مورچگان *Ant colony optimization*
- ✓ بهینه سازی کلونی زنبوران *Bee colony*
- ✓ الگوریتم‌های ترتیبی *cultural algorithms*
- ✓ الگوریتم‌های با هم تکاملی *Co-evolutionary algorithms*
- ✓ الگوریتم ژنتیک *Genetic algorithm*
- ✓ جستجوی محلی تکراری *Iterated local search*
- ✓ بهینه سازی توده ذرات *particle swarm intelligent*
- ✓ انجماد تدریجی *Simulated Annealing*
- ✓ جستجوی ممنوع *Taboo search*
- ✓ جستجوی همسایگی متغیر *Variable neighbor search*

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. در صورتیکه به کاوش اهمیت بیشتری داده شود، الگوریتم رفتار تصادفی بیشتری خواهد داشت و به سمت رفتار تصادفی میل می‌کند.

کند و در صورتی که به رفتار تبعیت توجه بیشتری شود، الگوریتم از رفتار تصادفی فاصله می گیرد و جستجو تنها در محدوده راه حل های خوب به جستجو می پردازد.

معیارهای طبقه بندی زیادی ممکن است برای طبقه بندی فراابتکاری ها استفاده شود که در زیر به بعضی از آن ها اشاره می شود:

الهام گرفته از طبیعت در مقابل عدم الهام از طبیعت

خیلی از الگوریتم های فراابتکاری از فرایندهای طبیعی الهام گرفته شده اند: از قبیل الگوریتم های اجتماع مورچگان و زنبور عسل که از هوش توده ای از گونه های مختلف مورچگان و زنبوران استفاده می کنند.

نحوه استفاده از حافظه

بعضی از الگوریتم های فراابتکاری از قبیل انجماد تدریجی بدون حافظه هستند و حرکتهای قبلی را در جایی ذخیره نمی کنند. در مقابل الگوریتم جستجوی ممنوع از یک حافظه که بعضی از اطلاعات را در طول جستجو بدست می آورد، ذخیره می کند.

قطعی در مقابل احتمالی

یک الگوریتم قطعی، یک مسئله بهینه سازی را از طریق تصمیم گیری قطعی حل می نماید (برای مثال الگوریتم جستجوی محلی و جستجوی ممنوع). در الگوریتم های فراابتکاری احتمالی، بعضی از قواعد احتمالی در فرایند جستجو مورد استفاده قرار می گیرد که می توان به الگوریتم انجماد تدریجی و الگوریتم ژنتیک اشاره کرد. در الگوریتم های قطعی، با داشتن یک راه حل اولیه و اجراهای متفاوت، تنها یک جواب نهایی بدست می آید و در حالی که در الگوریتم های تصادفی، با داشتن یک راه حل اولیه و اجراهای متفاوت، ممکن است که جواب های نهایی متفاوتی بدست آید.

تکراری در مقابل حریصانه

در الگوریتم‌های تکراری، الگوریتم با یک راه حل کامل شروع شده و در هر تکرار راه حل یا راه حل‌ها تغییر پیدا می‌کنند. در الگوریتم‌های حریصانه یک راه حل کامل در اختیار نبوده بلکه با یک راه حل ساخته نشده شروع شده و در هر مرحله، یک متغیر تصمیم از مسئله یک قسمت از راه حل را می‌سازد. اغلب الگوریتم‌های فراابتکاری، الگوریتم‌های تکراری هستند.

در ادامه **الگوریتم اجتماع مورچگان** که الگوریتمی است که از طبیعت الهام گرفته است و در رده الگوریتم‌های احتمالی، و حریصانه با حافظه قرار می‌گیرد صحبت می‌شود.

معرفی الگوریتم اجتماع مورچگان

مورچه‌ها بر روی زمین از حدود ۱۰۰ میلیون سال پیش زندگی می‌کرده‌اند و در حال حاضر جمعیتی در حدود 10^{16} در دنیا دارند. تخمین زده شد که مجموع وزن مورچه‌ها در حدود مجموع وزن انسان‌های روی زمین می‌باشند. این حشرات موجوداتی اجتماعی می‌باشند و در کلونی‌هایی به اندازه‌ای از ۳۰ تا میلیون‌ها فرد زندگی می‌کنند. رفتارهای پیچیده کلونی مورچگان، انسان‌ها را تشویق کرده است که مطالعات متعددی در خصوص رفتارهای اجتماعی و فردی آن‌ها داشته باشند. رفتارهای گروهی از مورچگان در ارتباط با جستجو برای غذا، نظم کارگران، سازماندهی گورستان، مواظبت از بچه‌ها و ساختن لانه مورد مطالعه قرار گرفته است.

رفتارهای پیچیده مورچه‌ها، از رفتارهای جمعی مورچه‌های تکی خیلی ساده و بی‌حیله ناشی می‌شود. یک مورچه براساس اطلاعاتی که از محیط اطراف خود به دست می‌آورد، اقدام ساده‌ای را انجام می‌دهد که پایه‌ای تصادفی دارد.

یکی از اولین رفتارهایی که توسط محققین مورد مطالعه قرار گرفت، توانایی مورچه‌ها در یافتن کوتاهترین مسیر بین یک منبع غذا و لانه آن‌ها بود. بر اساس این مطالعات، اولین الگوریتم از رفتار مورچه

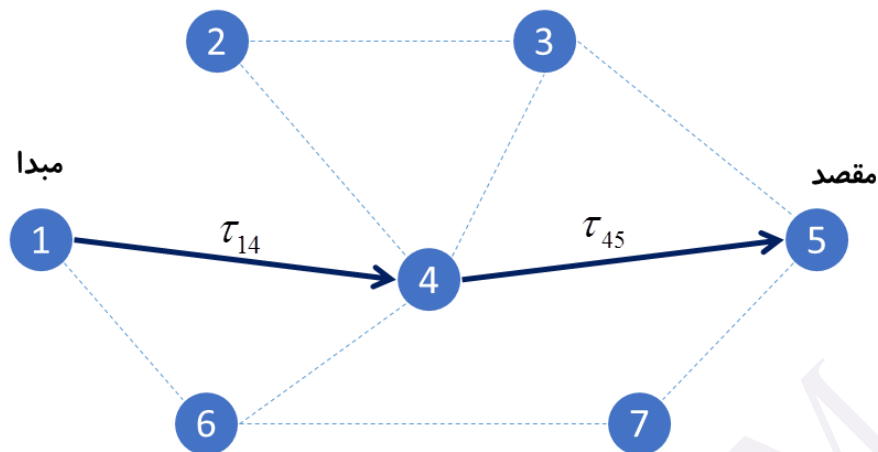
ها در جستجوی غذا توسط **مارکو دوریگو Marco Dorigo** ارائه گردید که خروجی رساله دکترای ایشان در دانشگاه میلان ایتالیا بود.

رفتار مورچه ها در جستجوی غذا

چگونه مورچه ها می توانند کوتاهترین مسیر بین محل غذا و لانه خود را بدون هیچ گونه مکانیسم هدایتی دیداری و مرکزی پیدا کنند؟ مطالعات رفتارهای بعضی از گونه های واقعی مورچه ها نشان داده است که یک الگوی تصادفی و شانسی در حرکت مورچه ها برای یافتن غذا وجود دارد. به محض این که یک منبع غذا در محلی در حوالی لانه مورچه ها قرار داده می شود، رفتار تصادفی به مرور سازماندهی بیشتری یافته و در نهایت منجر به رسیدن به یک مسیر مشابه توسط مورچه ها به این محل خواهد شد و به صورت جادویی دیده می شود که مورچه ها کوتاهترین مسیر را پیدا کرده اند. این رفتار عجیب نتیجه تاثیر مورچه هایی است که منبع غذا را پیدا کرده اند. مورچه هایی که غذا را یافته اند با سایر مورچه ها از طریق یک مکانیسم ارتباطی به نام **اثر فرومون Pheromone Trails** ارتباط برقرار می نمایند. زمانی که یک مورچه یک منبع غذا را پیدا می کند و غذا به خانه می برد، در مسیر حرکت خود ماده شیمیایی به نام **فرومون** منتشر می نماید. بقیه مورچه ها که به دنبال غذا می گردند، در هنگام انتخاب مسیر به غلظت فرومون مسیر توجه می نمایند. مسیرهایی با غلظت فرومون بیشتر، احتمال بیشتری برای انتخاب دارند. هر چقدر مورچه های بیشتری از یک مسیر عبور نمایند، غلظت فرومون آن مسیر افزایش یافته و در نتیجه، مورچه های بیشتری نیز نسبت به آن مسیر علاقه مند می شوند.

ساختار عمومی بهینه سازی اجتماع مورچگان

مساله یافتن کوتاهترین مسیر بین دو راس در یک گراف $G=(V,E)$ است که V مجموعه راس های و E ماتریسی است که ارتباطات بین راس ها را بیان می کند. این گراف $n_G=|V|$ راس دارد. طول، L^k ، که توسط مورچه k ام ساخته شده است، به صورت فاصله مبدا تا مقصد تعریف می گردد. یک گراف نمونه همراه با یک مسیر انتخابی، در شکل زیر نمایش داده شده است. طول مسیر انتخابی برابر با ۲ است. در این حالت، یک غلظت فرومون τ_{ij} به هر یال (i,j) از گراف، تخصیص داده می شود.

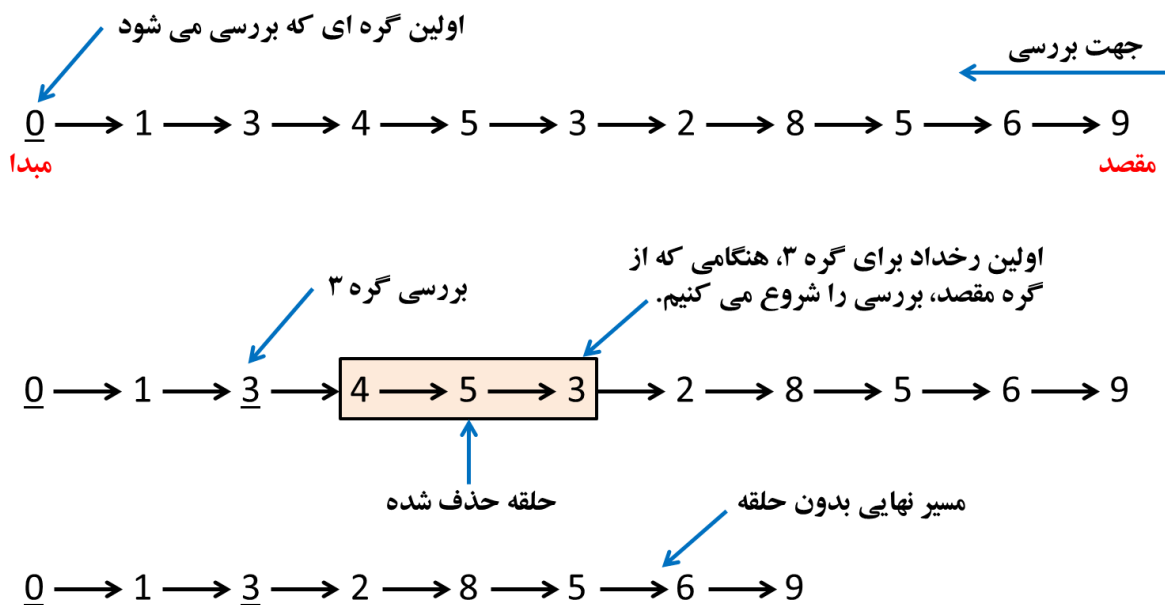


در الگوریتم استاندارد بهینه سازی اجتماع مورچگان یا همان **SACO** که مخفف **Simple Ant Colony Optimization** است، در ابتدا به هر یال، یک مقدار تصادفی به عنوان فرومون اولیه، τ_{ij}^0 ، تخصیص داده می شود. در ابتدای الگوریتم به علت نبود غلظت فرومون یا مقدار خیلی کم آن، مورچه ها به طور تصادفی مسیرها را انتخاب می نمایند. تعدادی از مورچه ها، $k=1, \dots, n$ ، در راس مبدا قرار داده می شود. در هر تکرار از الگوریتم SACO، هر مورچه به صورت مرحله به مرحله یک مسیر یا راه حل را بین مبدا و مقصد می سازد. در هر راس، هر مورچه فرایند تصمیم گیری را برای تعیین یال بعدی مسیر خود به کار می برد. اگر مورچه k در راس i قرار داشته باشد، راس بعدی $j \in N_i^k$ را بر اساس یک تابع **احتمال گذر** به شکل زیر تعیین می نماید.

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t)}{\sum_{j \in N_i^k} \tau_{ij}^\alpha(t)} & \text{if } j \in N_i^k \\ 0 & \text{if } j \notin N_i^k \end{cases}$$

که در آن N_i^k مجموعه راس های قابل انتخاب برای مورچه k ام در راس i است. در واقع N_i^k مجموعه یال های همسایه ای راس i است. در صورتیکه $N_i^k = \emptyset$ آنگاه راس پیشین راس i در نظر گرفته می شود و در واقع در صورتیکه مورچه به مسیر بسته برسد، برمی گردد. این مساله منجر به ایجاد یک حلقه می گردد، ولی مشکلی را ایجاد نمی کند زیرا پس از رسیدن به مقصد، در هنگام معرفی مسیر بین مبدا و مقصد، حلقه ها توسط مکانیزمی حذف می گردند. مکانیزم به این صورت که برای یک مسیر از گره مبدا شروع می کنیم و آن را در نظر می گیریم. از گره مقصد شروع می کنیم تا آن که اولین گره مشابه گره در نظر گرفته شده پیدا

شود. اگر موقعیت گره پیدا شده مشابه موقعیت گره در نظر گرفته شده نباشد، گره های بین این دو موقعیت را حذف می کنیم زیرا یک حلقه پیدا شده است. در شکل زیر این فرایند نشان داده شده است.



در معادله احتمال، $\square$ یک مقدار ثابت مثبت برای شدت بخشیدن به اثر غلظت فرومون است. مقدار بزرگ $\square$ ، اهمیت غلظت فرومون را افزایش داده و تاثیر رفتارهای تصادفی را کمتر می کند و منجر به همگرایی سریع به سمت یک بهینه محلی می گردد.

بعد از آن که تمامی مورچه ها، یک مسیر کامل بین مبدا تا مقصد را ساختند و حلقه های احتمالی حذف شدند، هر مورچه ردپای خود را به سمت راس مبدا دنبال می نماید و مقداری فرومون در هر یال (i,j) روی مسیر طی شده، مطابق با قاعده زیر منتشر می نماید.

$$\Delta \tau_{ij}^k(t) \propto \frac{1}{L^k(t)}$$

$L^k(t)$ معرف طول مسیر ساخته شده توسط مورچه k -ام در مرحله زمانی t می باشد. در نتیجه، میزان

غلظت فرومون هر مسیر مطابق با قاعده زیر به هنگام می گردد

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta \tau_{ij}^k(t)$$

که در رابطه فوق، n_k تعداد مورچه ها می باشد. با استفاده از رابطه $\tau_{ij}^k(t)$ مقدار فرومون روی هر یال تولید می گردد و این مقدار بستگی به کیفیت مسیریابی دارد که توسط مورچه ها با استفاده از آن یال ساخته شده است. به عبارت دیگر، هر مورچه مقداری فرومون در هر یال متناسب با کیفیت مسیرش به جا می گذارد. بعد از یک تکرار، میزان فرومون موجود در یک یال با توجه به مورچه های عبوری از آن یال تغییر می یابد. مقدار چگالی اضافه شده با مقدار قبلی چگالی فرومون آن مسیر طبق معادله $\tau_{ij}(t+1)$ به هنگام می شود.

در $SACO$ ، مقدار کیفیت هر راه حل با یک تابع ساده به صورت عکس طول مسیر طی شده بیان می گردد. البته معیارهای دیگری نیز می توان تعریف کرد. اگر $x^k(t)$ نشان دهنده یک راه حل در مرحله زمانی t باشد، آنگاه $f(x^k(t))$ کیفیت آن راه حل را بیان می کند. اگر مقدار فرومون انتشاری رابطه عکس با طول مسیر داشته باشد در آن صورت، مقدار بیشتر $f(x^k(t))$ موجب می گردد که مقدار کمتری فرومون در آن مسیر منتشر گردد. بنابراین، یک مسیر طولانی موجب می گردد که دارای جذابیت کمتری شوند.

در ادامه یکی از الگوریتم های معروف که نسبت به الگوریتم $SACO$ تغییراتی به آن داده شده است ارایه می شود.

سیستم مورچه Ant system

اولین الگوریتم اجتماع مورچگان که توسط آقای **مارکو دوریگو Marco Dorigo** معرفی شد، با نام الگوریتم سیستم مورچه **Ant System** شناخته می شود. p_{ij}^k به منظور در نظر گرفته اطلاعات هیوریستیکی یا پیش فرض پیشنهاد شده است. در الگوریتم سیستم مورچه، احتمال حرکت از راس i به راس j (p_{ij}^k) به صورت زیر بیان می گردد.

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{u \in N_i^k(t)} \tau_{iu}^\alpha(t) \eta_{iu}^\beta(t)} & \text{if } j \in N_i^k(t) \\ 0 & \text{if } j \notin N_i^k(t) \end{cases}$$

که در آن τ_{ij} اثر بخشی قیاسی یا **A Posteriori Effectiveness** یک حرکت از راس i به راس j که همان شدت فرومون مسیر (i,j) ، می باشد. η_{ij} اثربخشی با دلیل قبلی **A Priori Effectiveness** آن حرکت که توسط بعضی از هیوریستیک ها محاسبه می گردد. تابع احتمال گذر بالا در الگوریتم سیستم مورچه دارای دو جنبه است:

احتمال گذر مورد استفاده در الگوریتم سیستم مورچه توازنی را بین **غلظت فرومون**، τ_{ij} ، (به عبارت دیگر تاریخچه حرکت های موفق قبلی همان جنبه اول) و **اطلاعات هیوریستیکی**، η_{ij} ، (که جذابیت حرکت را بیان می کند همان جنبه دوم) برقرار می نماید. این امر منجر به توازن بین **کاوش** و **تبعیت** می گردد. بهترین توازن بین کاوش و تبعیت از طریق انتخاب مناسب پارامترهای α و β به دست می آید. اگر $\alpha = 0$ باشد، اطلاعات فرومونی مورد استفاده قرار نمی گیرد و سابقه گذشته جستجو فراموش می گردد و الگوریتم حالت جستجوی تصادفی را به خود می گیرد. اگر $\beta = 0$ باشد، جذابیت حرکت ها صرف نظر می گردد و الگوریتم ساختار **SACO** را به خود می گیرد. اطلاعات هیوریستیکی نحوه انتخاب مسیر ها را دچار تغییراتی می نماید. نحوه تعریف آن بسته به نوع مسئله دارد. برای مثال، برای یک مساله حداقل سازی مسیر، این تابع به صورت زیر قابل تعریف است:

$$\eta_{ij} = \frac{1}{d_{ij}}$$

که در آن d_{ij} فاصله یا هزینه بین راس i و j است.

مجموعه N_i^k ، مجموعه **راس های امکان پذیر** را برای مورچه k زمانی که در راس i قرار دارد، معرفی می نماید. مجموعه راس های امکان پذیر، تنها شامل همسایه های بعد از راس i می باشند. برای جلوگیری از ایجاد حلقه، ممکن است N_i^k به صورت مجموعه تمامی راس های همسایه ای که تاکنون توسط مورچه k

ملاقات نشده اند تعریف گردد. برای این منظور، یک لیست ممنوع **Taboo List** برای هر مورچه تعریف می‌گردد. هر زمانی که یک مورچه یک راس جدید را ملاقات می‌نماید، آن راس به لیست ممنوعه مورچه اضافه می‌گردد. برای این که یک راس تنها یک بار ملاقات شود، راس‌های قرار گرفته در لیست ممنوع، از مجموعه N_i^k خارج می‌گردند.

البته فرمول احتمال انتخاب همسایگی به فرمول اشاره شده منتهی نمی‌شود و تحقیقات گسترده‌ای در این خصوص انجام شده است. یکی از این تحقیقات توسط **منیزو و کلورنی Maniezzo and Colorni** انجام شده است که از فرمول زیر برای محاسبه احتمال گذر استفاده کردند.

$$p_{ij}^k(t) = \begin{cases} \frac{\alpha \tau_{ij}(t) + (1-\alpha) \eta_{ij}(t)}{\sum_{u \in N_i^k(t)} \alpha \tau_{iu}(t) + (1-\alpha) \eta_{iu}(t)} & \text{if } j \in N_i^k(t) \\ 0 & \text{if } j \notin N_i^k(t) \end{cases}$$

که در آن پارامتر $\square$ اهمیت نسبی غلظت فرومون، $\tau_{ij}(t)$ ، را نسبت به جذابیت، $\eta_{ij}(t)$ ، نشان می‌دهد. این فرومون به علت عدم نیاز به پارامتر $\square$ ، ساده‌تر از فرمول قبلی است.

روش‌های مختلفی برای نحوه و میزان انتشار فرومون توسط یک مورچه در یک مسیر وجود دارد که در زیر برخی از این موارد بیان می‌شود:

۱- مقدار فرومون منتشر شده متناسب با معکوس کیفیت مسیر ایجاد شده توسط یک مورچه است. در این فرمول، Q یک مقدار ثابت است.

$$\Delta \tau_{ij}^k(t) = \frac{Q}{f(x^k(t))}$$

۲- در این حالت برای تمامی یال‌های قرار گرفته در مسیر مورچه، مقدار ثابت و برابری از فرومون منتشر می‌گردد.

$$\Delta \tau_{ij}^k(t) = Q$$

۳- در این حالت مقدار فرومون منتشره وابسته به طول یال قرار گرفته در مسیر مورچه می باشد.

$$\Delta \tau_{ij}^k(t) = \frac{Q}{d_{ij}}$$

پس از انتخاب یکی از سه روش محاسبه میزان فرومون، در این مرحله، فرومون ها بر روی مسیرهای عبوری مورچه ها به صورت زیر به هنگام می شود.

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta \tau_{ij}^k(t)$$

برای اجبار مورچه ها به **کاوش بیشتر** و جلوگیری از **همگرایی زودرس**، اجازه داده شد که مقدار شدت فرومون در یک مسیر در هر تکرار از الگوریتم، قبل از ساخت مسیر دچار **تبخیر** شود. در این حالت مقداری از شدت فرومون هر مسیر **کاهش** یا همان تبخیر پیدا می کند. در این حالت نحوه به هنگام شدن مقدار شدت فرومون طبق معادله زیر انجام می گردد.

$$\tau_{ij}(t) \leftarrow (1-\rho)\tau_{ij}(t)$$

که در رابطه فوق، $\rho \in [0,1]$ یک مقدار ثابت است که **نرخ تبخیر فرومون** در هر مرحله را نمایش می دهد. این تبخیر منجر به فراموش شدن وقایع گذشته می گردد و به عبارت دیگر ρ یک پارامتر کنترلی تاریخچه جستجو است. برای مقادیر بالای ρ ، نرخ **بالا** تبخیر، توجه به گذشته **کم** شده و جستجو وضعیت **تصادفی و شانسی** به خود می گیرد. در صورتی که $\rho = 1$ باشد، جستجو کاملاً تصادفی و شانسی خواهد بود.

با توجه به موارد اشاره شده در بالا می توان **الگوریتم سیستم مورچه** را به صورت زیر بیان کرد:

الگوریتم سیستم مورچه به صورت خلاصه در شکل زیر آورده شده است:

گام (۱) تنظیم شرایط اولیه

$$T=0$$

(a) تنظیم تمامی پارامترها، $\square, \beta, \rho, Q, n_k, \tau_0$

(b) تمامی مورچه ها ($k = 1, \dots, n_k$) را در گره‌های مختلف شبکه قرار دهید.

(c) برای هر یال (i, j) انجام دهید.

$$\tau_{ij}(t) \approx U(0, \tau_0)$$

گام ۲) تکرار (تا شرط توقف برآورده شود)

(a) برای هر مورچه $k = 1, \dots, n_k$ انجام دهید.

$$x^k(t) = \emptyset$$

(b) کارهای زیر را تکرار کنید تا مسیر کامل ساخته شود

a. برای هر راس i ، راس بعدی j را براساس احتمال تعریف شده در معادله زیر تعیین کنید

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{u \in N_i^k(t)} \tau_{iu}^\alpha(t) \eta_{iu}^\beta(t)} & \text{if } j \in N_i^k(t) \\ 0 & \text{if } j \notin N_i^k(t) \end{cases}$$

b. یال (i, j) را به مسیر اضافه می‌کنیم و به صورت زیر مسیر اصلاح می‌شود.

$$x^k(t) = x^k(t) \cup \{(i, j)\}$$

(c) مقدار کیفیت مسیر را برای مورچه k توسط تابع $f(x^k(t))$ محاسبه شود.

گام ۳) به هنگام سازی مقدار فرمون ها

برای هر یال کارهای زیر را انجام دهید.

(a) معادله زیر را برای تبخیر استفاده می‌کنیم.

$$\tau_{ij}(t) \leftarrow (1 - \rho) \tau_{ij}(t)$$

(b) برای هر مورچه $k = 1, \dots, n_k$ ریختن فرومون به صورت زیر انجام شود.

a. برای هر یال که بر روی مسیر مورچه k ام است، همان $x^k(t)$ ، میزان فرومون زیر اضافه می‌شود.

$$\Delta \tau^k = \frac{Q}{f(x^k(t))}$$

b. مقادیر فرومون را با استفاده از معادله زیر به هنگام کنید.

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta \tau_{ij}^k(t)$$

(c) برای هر یال مقدار فرومون در زمان $t+1$ به زمان t تغییر می‌کند $(\tau_{ij}(t+1) = \tau_{ij}(t))$.

گام ۴) پایان تکرار.

مسیر $x^k(t)$ با کمترین $f(x^k(t))$ را به عنوان بهترین راه حل چاپ کنید.

سیستم مورچه رتبه ای

الگوریتم سیستم مورچه رتبه ای یا *Rank-based Ant System* برای بهبود الگوریتم سیستم مورچه توسط بولنهمر یا (Bullnheimer) در سال ۱۹۹۹ ارایه شد. در این روش، هر مورچه براساس رتبه ای که در میان مورچه های دیگر دارد، در مسیر فرومون ریزی می‌کند. این رتبه براساس کیفیت مسیر تولید شده تعیین می‌گردد. بدین معنا که مورچه با رتبه بالاتر، میزان فرومون بیشتری بر روی کمان ها از خود باقی می‌گذارد. تنها اختلاف این روش با سیستم مورچه، در نحوه به هنگام سازی فرومون کمان ها است. قبل از به هنگام سازی فرومون هر یک از کمان ها، مورچه ها براساس مطلوبیت مسیر تولید شده، به طور افزایشی رده بندی می‌شوند. در هر تکرار تنها $w-1$ مورچه نخست و "بهترین مورچه تا به حال" (این مورچه لزوما متعلق به اجرای کنونی نیست)، در جمع w مورچه اجازه دارند در مسیر فرومون ریزی کنند. بنابراین، به هنگام سازی فرومون کمان ها در سیستم مورچه رتبه به صورت زیر انجام می‌پذیرد:

$$\tau_{ij}(t) = \tau_{ij}(t) + \sum_{r=1}^{w-1} (w-r) \Delta \tau_{ij}^r + w \Delta \tau_{ij}^b$$

که در آن:

$(w-r) \Delta \tau_{ij}^r$ میزان فرومونی که مورچه مرتبه r -ام بر روی کمان (i,j) می‌ریزد و

$w \Delta \tau_{ij}^b$ میزان فرومونی که بهترین مورچه بر روی کمان (i,j) از خود باقی می‌گذارد، هستند.

همان طور که از رابطه بالا روشن است، مورچه با بهترین رتبه در مقایسه با دیگر مورچه ها، بیشترین میزان مشارکت در فرومون ریزی را دارد.

سیستم مورچه حداکثر – حداقل

الگوریتم سیستم مورچه حداکثر – حداقل یا *Max-Min Ant system* توسط شوتزل (Stützle) در سال ۱۹۹۶ معرفی شد. تفاوت این الگوریتم با سیستم مورچه سبب شده است که کارایی این روش در یافتن جواب بهینه در مقایسه با سیستم مورچه، بهبود چشم گیری یابد. در این روش، فرمون ریزی فقط توسط **بهترین مورچه در تکرار جاری** یا **بهترین مورچه تا به حال** انجام می شود. البته باید اولویت فرمون ریزی در این دو نوع مورچه مشخص شود، مثلاً ممکن است در مراحل **ابتدای** الگوریتم **بهترین مورچه در تکرار جاری** بر روی کمان های مسیر فرومون ریزی کند و با نزدیک شدن به **پایان** الگوریتم، **بهترین مورچه تا به حال** بر روی کمان ها فرومون ریزی کند. چنین روشی سبب می شود که اکثر مورچه ها به سمت یک جواب بهینه محلی که توسط دو مورچه مذکور ایجاد شده اند، همگرا شوند. تمایل مورچه ها به مسیر تولید شده توسط این دو مورچه، باعث افزایش مقدار فرومون کمان های متعلق به این دو مسیر می شوند. برای رفع این اشکال، یک محدوده مجاز برای مقدار فرومون هر کمان در نظر گرفته می شوند. زمانی که میزان فرومون کمانی از حد بالا تجاوز کرد، فرومون آن کمان برابر با حد بالا قرار می گیرد و اگر میزان فرومون یک کمان از حد پایین کمتر شد، فرومون آن کمان با حد پایین مقدار دهی می شود. تخمین از حد بالا و حد پایین برای فرومون هر کمان را برای مسائل کمینه سازی می توان به صورت زیر پیشنهاد کرد:

$$\tau_{\max} = \frac{1}{\rho f_{\text{best}}}$$

$$\tau_{\min} = \frac{\tau_{\max}}{a}$$

که در آن:

f_{best} تابع هدف برای بهترین مورچه تا به حال و

a پارامتر مسئله است.

در نهایت، به منظور افزایش احتمال یافتن بهترین جواب، بعد از تعدادی تکرار مشخص، یا در مواردی که بهبودی در بهترین جواب مسئله ایجاد نشد، مقدار فرمون تمامی کمان های دوباره با حد بالا مقدار دهی می شوند. با اعمال این سیاست، احتمال یافتن بهترین جواب در فضای جستجوی مسئله افزایش می یابد.

سیستم اجتماع مورچه

سیستم اجتماع مورچه یا سیستم کلونی مورچه (*Ant Colony System*) که با **ACS** نشان داده می شود توسط مارکو دوریگو و گامباردلا برای توسعه عملکرد سیستم مورچه ارایه شده است. الگوریتم ACS در سه زمینه با سیستم مورچه متفاوت است.

۱- **قاعده انتقال:** قاعده انتقال ACS برای ایجاد یک بالانس قوی بین قابلیت های کاوش و تبعیت الگوریتم توسعه داده شد. براساس آن، مورچه k که در راس i قرار دارد، راس بعدی j را برای حرکت از طریق قاعده زیر انتخاب می کند:

$$j = \begin{cases} \arg \max_{u \in N_i^k(t)} \{\tau_{iu}(t) \eta_{iu}^\beta(t)\} & \text{if } r \leq r_0 \\ j & \text{if } r > r_0 \end{cases}$$

که در آن $r \sim U(0,1)$ و $r_0 \in [0,1]$ پارامتری است که توسط کاربر معرفی می شود و $J \in N_i^k(t)$ راسی است که به طور تصادفی براساس احتمال زیر انتخاب می گردد:

$$p_{ij}^k(t) = \frac{\tau_{ij}(t) \eta_{ij}^\beta(t)}{\sum_{u \in N_i^k(t)} \tau_{iu}(t) \eta_{iu}^\beta(t)}$$

پارامتر r_0 برای توازن بین کاوش و تبعیت استفاده می گردد. اگر $r \leq r_0$ باشد، الگوریتم تمایل پیدا می کند بهترین راس را انتخاب کند (تبعیت) و اگر $r > r_0$ باشد، الگوریتم به کاوش می پردازد. بنابراین، با انتخاب مقادیر کوچک r_0 ، موجب می گردد که **تبعیت** کمتر اتفاق افتاده و بیشتر **کاوش** صورت گیرد.

۲- انتشار فرومون توسط بهترین مورچه: برخلاف سیستم مورچه، در الگوریتم سیستم اجتماع

مورچه تنها بهترین مورچه سراسری (در کل مسیر طی شده بهترین باشد و با $x^b(t)$ نشان داده

می شود) اجازه انتشار فرومون مجدد را در مسیر طی شده دارد. فرومون از طریق قاعده به هنگام

سازی سراسری زیر به هنگام می شود:

$$\tau_{ij}(t+1) = (1-\rho_l)\tau_{ij}(t) + \rho_l \Delta\tau_{ij}(t)$$

$$\Delta\tau_{ij}(t) = \begin{cases} \frac{1}{f(x^b(t))} & \text{if } (i,j) \in x^b(t) \\ 0 & \text{otherwise} \end{cases}$$

قاعده به هنگام سازی ACS به تشویق مورچه ها به سمت جستجو در اطراف بهترین راه حلی که تا

کنون پیدا شده، می پردازد. این استراتژی از خاصیت تبعیت حمایت بیشتری کرده و بعد از این که تمامی

مورچه ها مسیر را به انتها رساندند، به کار گرفته می شود. آقایان دوریگو و گامباردلا دو روش برای انتخاب

مسیر ارایه کردند:

✓ بهترین تکرار، که $x^b(t)$ بهترین مسیر یافته شده در طول تکرار جاری، t بوده است.

✓ بهترین سراسری که $x^b(t)$ بهترین مسیر یافته شده از اولین تکرار الگوریتم تاکنون بوده

است.

۳- تبخیر فرومون

تبخیر فرومون نیز متفاوت از سیستم مورچه AS است. علاوه بر قاعده به هنگام سازی اشاره شده در بالا

در واقع یک قاعده بهنگام سازی جهانی است، یک قاعده بهنگام سازی محلی هم در نظر گرفته می شود

که به صورت زیر تعریف می شود:

$$\tau_{ij}(t) = (1-\rho')\tau_{ij}(t) + \rho'\tau_0$$

که در آن ρ' یک مقدار ثابت بین صفر و یک و τ_0 یک مقدار ثابت مثبت کوچک است.

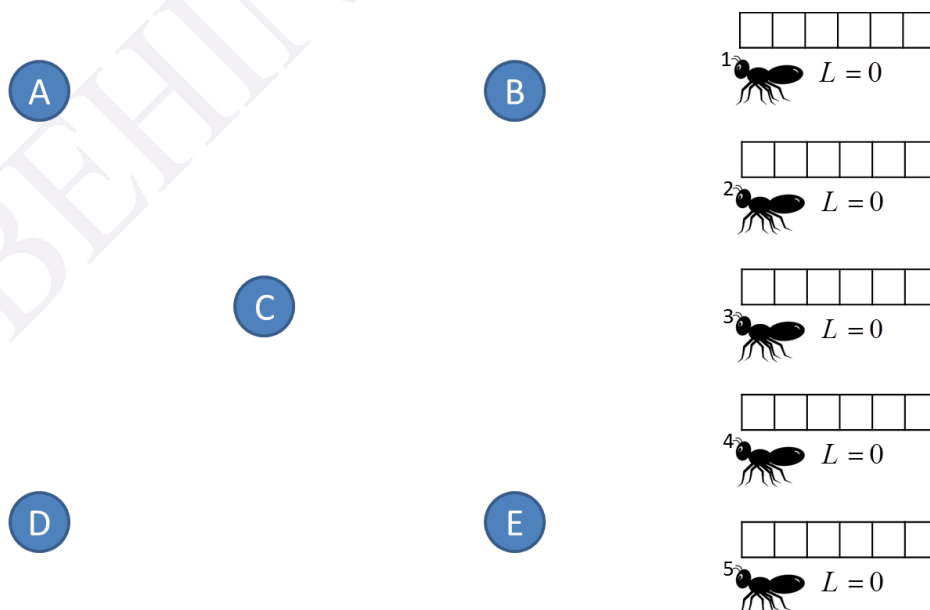
مثال:

در این مثال با استفاده از الگوریتم سیستم مورچه به دنبال حل مسئله فروشنده دوره گرد هستیم. مسئله فروشنده دوره گرد عبارت است از یافتن بهترین مسیر بین مجموعه ای از شهر ها به نحوی که از یک نقطه یا شهر شروع کرده و دوباره به همان نقطه یا شهر برگردد. از هر شهر باید یک بار عبور کرده و از تمامی شهر ها نیز عبور نماید. مسئله فروشنده دوره گرد با ۵ شهر را در نظر بگیرید.

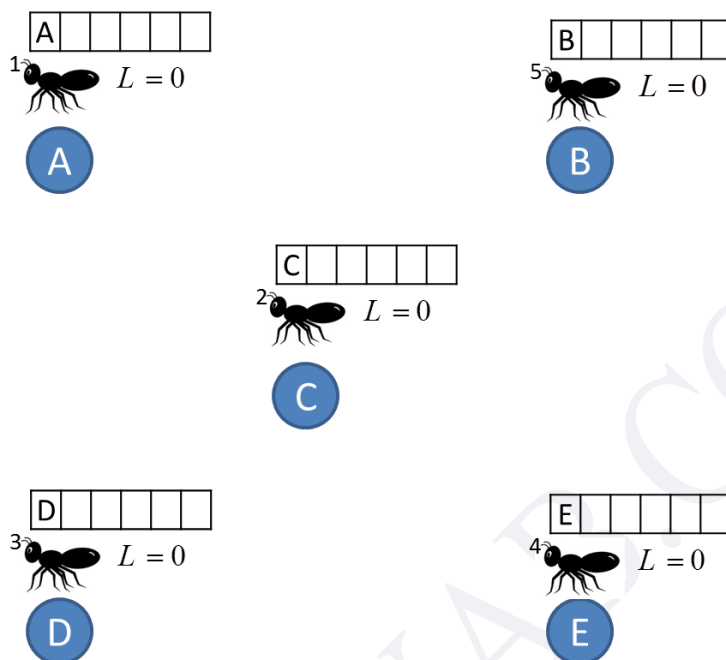
نقاط	A	B	C	D	E
A		48	60	29	151
B			79	86	40
C				105	80
D					129

حل:

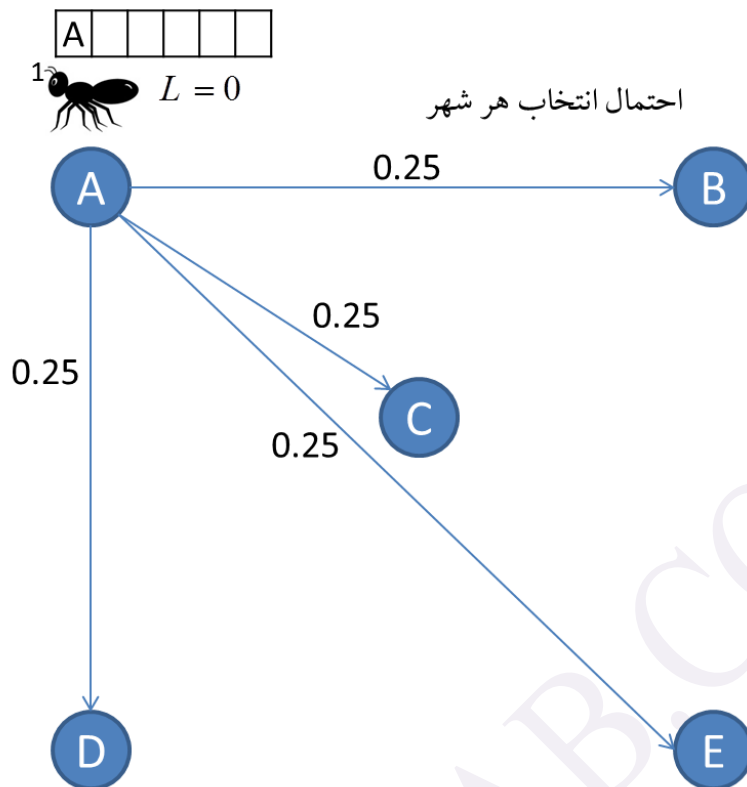
در ابتدا ۵ مورچه در نظر گرفته می شود. هر یک از این مورچه ها شماره گذاری شده و دارای یک لیست ممنوع به همراه خود می باشند. در شکل زیر این اطلاعات به صورت زیر است.



در مرحله اول، مورچه ها به تصادف روی شهرها قرار داده می شوند. در شکل زیر نحوه قرار گیری مورچه در گره های شبکه آورده شده است.



در صورتیکه یک مبدا و یک مقصد مشخص وجود داشت، باید تمامی مورچه ها را در مبدا قرار داد. ولی در این حالت هر شهر می تواند مبدا یا مقصد باشد توصیه شده است که مورچه ها در تمامی گره ها توزیع شود. وقتی که یک مورچه در شهری قرار می گیرد در این صورت آن شهر در لیست ممنوعه مورچه قرار می گیرد. بعد از این که مورچه ها در اولین شهر خود قرار داده شدند، باید شهر بعدی خود را انتخاب نمایند. انتخاب شهر بعدی، همان طور که قبلا تشریح شد، با استفاده از یک قاعده احتمالی براساس شده فرمون در کمان ها انتخاب می گردد. در اولین تکرار، با توجه به این که غلظت فرمون اولیه مساوی در کمان ها قرار دارد، انتخاب مسیر بعدی کاملا تصادفی خواهد بود و احتمال تمامی کمان ها با هم برابر خواهد بود. در تکرارهای بعدی، به دلیل تغییر غلظت فرمون در لبه ها، این انتخاب کاملا تصادفی نخواهد بود و مسیرهای دارای غلظت فرمون بیشتر، شانس بیشتری را برای انتخاب خواهند داشت. در شکل زیر انتخاب های ممکن مورچه نشان داده شده است.

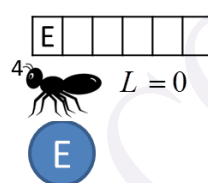
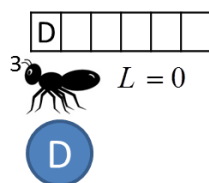
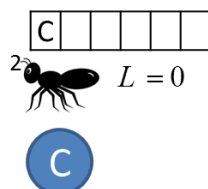
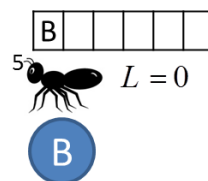
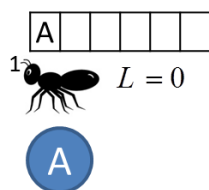


نکته: در کنار هر مورچه، متغیر L نشان دهنده کل مسیر طی شده توسط مورچه می باشد.

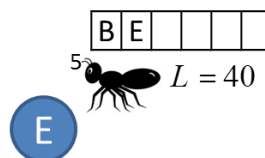
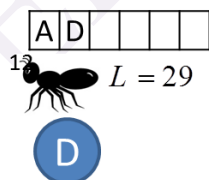
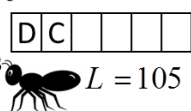
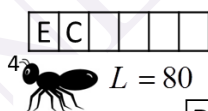
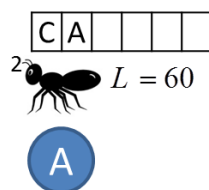
توجه: به این نکته توجه شود که این امکان وجود دارد که در برخی از شهرها مورچه ای قرار نداشته باشد.

فرایند انتخاب شهرها توسط مورچه تا زمانی که تمامی مورچه از تمامی شهرها عبور کنند ادامه پیدا می کند.

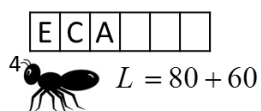
قرارگیری مورچه در شهر اول



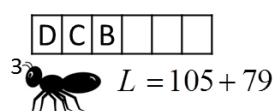
حرکت مورچه به شهر دوم



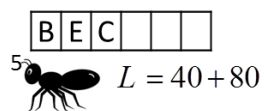
حرکت مورچه به شهر سوم



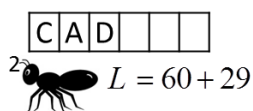
A



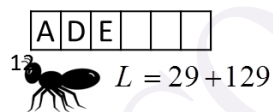
B



C

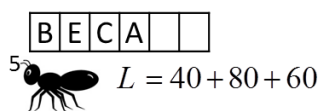


D

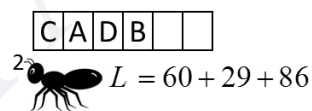


E

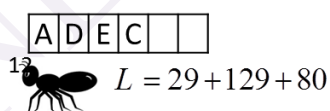
حرکت مورچه به شهر چهارم



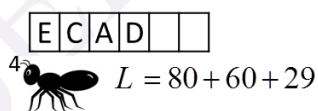
A



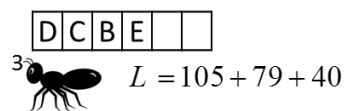
B



C



D



E

حرکت مورچه به شهر پنجم

$$\begin{array}{|c|c|c|c|c|} \hline D & C & B & E & A \\ \hline \end{array}$$

$$L = 105 + 79 + 40 + 151$$

A

$$\begin{array}{|c|c|c|c|c|} \hline E & C & A & D & B \\ \hline \end{array}$$

$$L = 80 + 60 + 29 + 86$$

B

$$\begin{array}{|c|c|c|c|c|} \hline A & D & E & C & B \\ \hline \end{array}$$

$$L = 29 + 129 + 80 + 79$$

C

$$\begin{array}{|c|c|c|c|c|} \hline B & E & C & A & D \\ \hline \end{array}$$

$$L = 40 + 80 + 60 + 29$$

D

$$\begin{array}{|c|c|c|c|c|} \hline C & A & D & B & E \\ \hline \end{array}$$

$$L = 60 + 29 + 86 + 40$$

E

بازگشت مورچه به شهر مبدا

$$\begin{array}{|c|c|c|c|c|c|} \hline A & D & E & C & B & A \\ \hline \end{array}$$

$$L = 29 + 129 + 80 + 79 + 48$$

A

$$\begin{array}{|c|c|c|c|c|c|} \hline B & E & C & A & D & B \\ \hline \end{array}$$

$$L = 40 + 80 + 60 + 29 + 86$$

B

$$\begin{array}{|c|c|c|c|c|c|} \hline C & A & D & B & E & C \\ \hline \end{array}$$

$$L = 60 + 29 + 86 + 40 + 80$$

C

$$\begin{array}{|c|c|c|c|c|c|} \hline D & C & B & E & A & D \\ \hline \end{array}$$

$$L = 105 + 79 + 40 + 151 + 29$$

D

$$\begin{array}{|c|c|c|c|c|c|} \hline E & C & A & D & B & E \\ \hline \end{array}$$

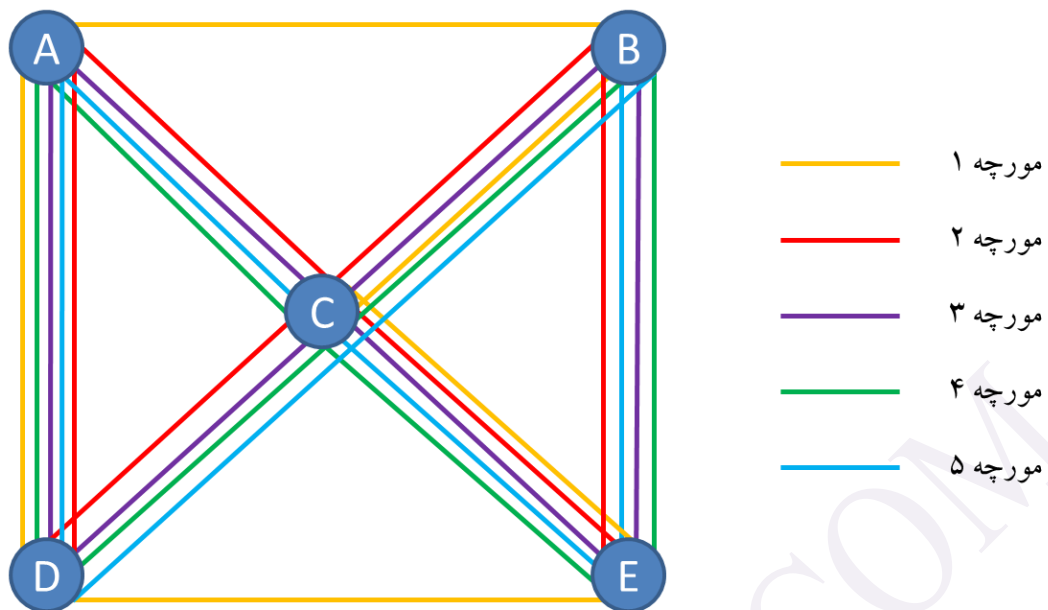
$$L = 80 + 60 + 29 + 86 + 40$$

E

شکل زیر مسیرهای طی شده توسط مورچه ها را نمایش می دهد. همان طور که در شکل مشاهده می

شود، از بعضی از مورچه بیشتری عبور کرده است. برای مثال از کمان CB پنج مورچه عبور می کند و از

کمان DE تنها یک مورچه عبور می کند.



هر چقدر تعداد تکرارها افزایش یابد، مورچه ها به سمت مسیر کوتاه تر تشویق شده و شانس انتخاب مسیرهای مشابه (همان مسیر کوتاهتر) افزایش می یابد. در صورتی که الگوریتم به خوبی تنظیم شده باشد، در تکرارهای انتهایی الگوریتم، اکثر مورچه ها مسیر کوتاه تر را انتخاب می نمایند.

بعد از ایجاد راه حل ها در هر تکرار، مرحله بعدی به هنگام سازی فرمون می باشد. پس از تکرار اول، ۵ راه حل برای ۵ مورچه ساخته می شود که براساس کیفیت این راه حل ها باید فرمون ریزی بر روی کمان ها انجام شود. در صورتیکه نرخ تبخیر برابر با ۰.۵ باشد و مقدار فرمون اولیه برابر با ۰.۰۵ باشد، مقدار فرمون هر کمان پس از تبخیر برابر با ۰.۰۲۵ می شود.

$$\tau_{ij}(1) = (1 - 0.5)\tau_{ij}(1) = 0.025$$

برای محاسبه مقدار فرمون ریزی از رابطه زیر استفاده می شود:

$$\Delta\tau_{ij} = \frac{Q}{f_k}$$

که در رابطه فوق، Q برابر با ۳۰ در نظر گرفته می شود. لذا مقدار فرمون برای هر یک از مورچه ها به صورت زیر بدست می آید.

مورچه	مسیر مورچه	مقدار تابع هدف	$\Delta\tau_{ij}$
۱	ADECBA	۳۶۵	۰.۰۸۲
۲	CADBEC	۲۹۵	۰.۱۰۲
۳	DCBEAD	۴۰۴	۰.۰۷۴
۴	ECADBE	۲۹۵	۰.۱۰۲
۵	BECADB	۲۹۵	۰.۱۰۲

مقدار فرومون محاسبه شده در جدول فوق در کلیه کمان های مسیر طی شده توسط هر مورچه توزیع می گردد. لذا برای محاسبه مقدار فرومون منتشر شده در هر یال، باید فرومون منتشر شده توسط کلیه مورچه ها را در نظر گرفت. در نتیجه مقدار فرومون منتشر شده در یال A به D و یال D به E از طریق رابط زیر محاسبه می گردد:

$$\Delta\tau_{AD}(1) = 0.082 + 0.102 + 0.074 + 0.102 + 0.102 = 0.462$$

$$\Delta\tau_{DE}(1) = 0.082$$

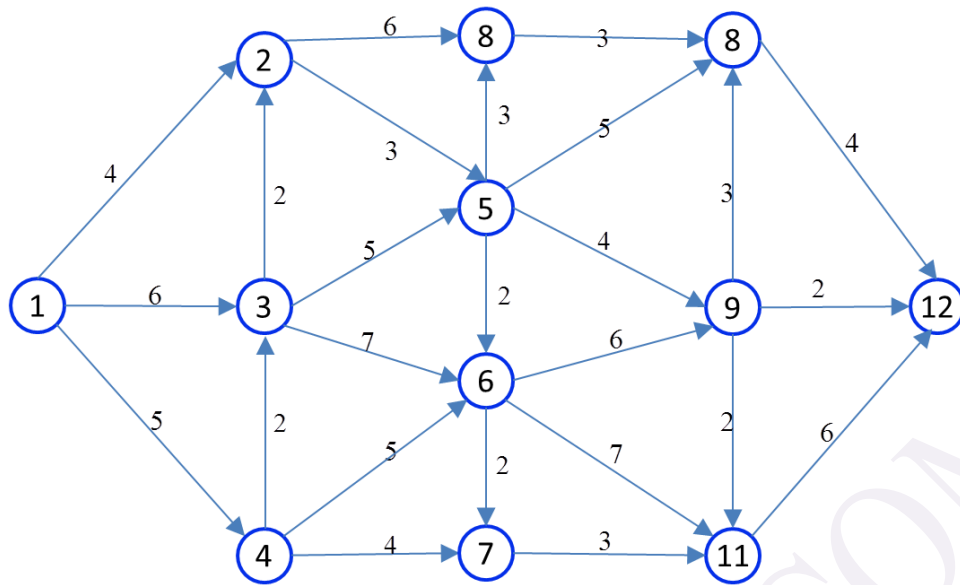
و در نتیجه مقدار فرومون آن بعد از به هنگام سازی به صورت زیر تغییر می یابد.

$$\tau_{AD}(2) = \tau_{AD}(1) + \Delta\tau_{AD}(1) = 0.025 + 0.462 = 0.487$$

$$\tau_{DE}(2) = \tau_{DE}(1) + \Delta\tau_{DE}(1) = 0.025 + 0.082 = 0.107$$

برای تمامی یال های دیگر نیز مقدار فرومون منتشر شده به شکل فوق قابل محاسبه می باشد. بعد از آنکه فرومون یال ها به هنگام شد، تکرار دوم توسط مورچه ها انجام می گردد.

مثال: با استفاده از روش الگوریتم سیستم مورچه، کوتاهترین مسیر در شبکه زیر از گره ۱ (مبدا) به گره ۱۲ (مقصد) را بدست آورید.



حل:

گام ۱: اطلاعات اولیه به صورت زیر در نظر گرفته می شود:

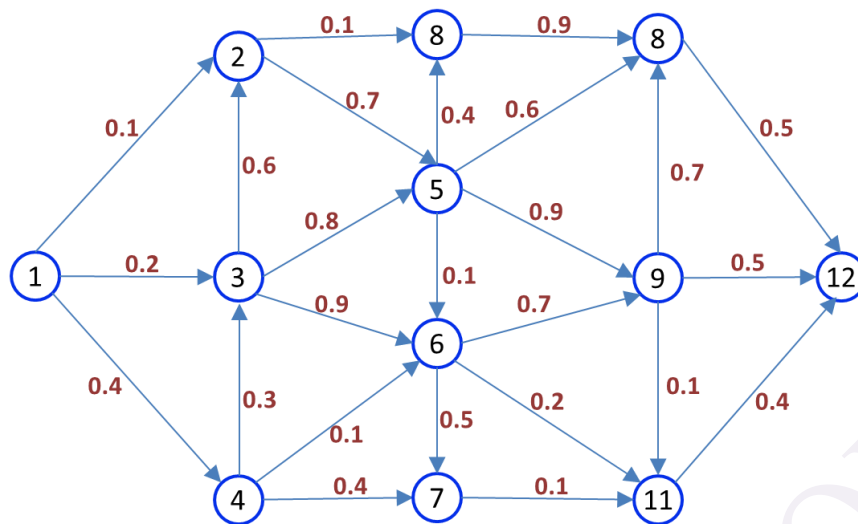
$$\alpha = 1, \beta = 1, \rho = 0.5, n_k = 2, \tau_0 = 1, t = 1$$

$$\eta_{ij} = \frac{1}{d_{ij}}, \Delta \tau_{ij}^k = \frac{Q}{\sum_{(i,j) \in x^k} d_{ij}}, Q = 10$$

دو مورچه را در گره ۱ قرار می دهیم.

برای هر کمان، فرومون اولیه به صورت تصادفی بین صفر و ۱ تولید می شود که در شکل زیر آورده

شده است.



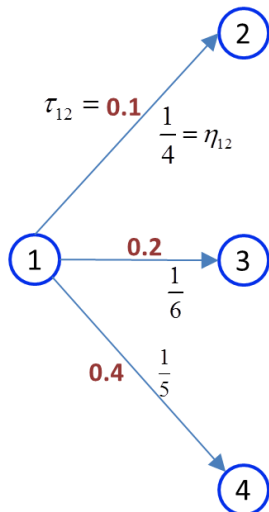
$$x^1(t) = \emptyset, x^2(t) = \emptyset$$

گام ۲: برای **مورچه ۱**، مسیر را کامل کنید. برای یافتن مسیر حرکت مورچه ۱ از مبدا تا مقصد، از تابع احتمال زیر استفاده می کنیم.

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{u \in N_i^k(t)} \tau_{iu}^\alpha(t) \eta_{iu}^\beta(t)} & \text{if } j \in N_i^k(t) \\ 0 & \text{if } j \notin N_i^k(t) \end{cases}$$

مسیریابی توسط مورچه ۱

از گره ۱ به گره های ۲، ۳ و ۴ کمان وجود دارد. احتمال هر یک از سه کمان به صورت زیر بدست می آید.



$$p_{12} = \frac{0.1 \times \frac{1}{4}}{0.1 \times \frac{1}{4} + 0.2 \times \frac{1}{6} + 0.4 \times \frac{1}{5}} = 0.18$$

$$p_{13} = \frac{0.2 \times \frac{1}{6}}{0.1 \times \frac{1}{4} + 0.2 \times \frac{1}{6} + 0.4 \times \frac{1}{5}} = 0.24$$

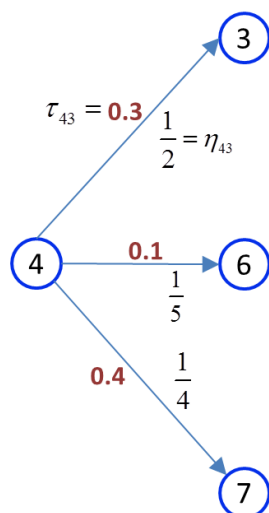
$$p_{14} = \frac{0.4 \times \frac{1}{5}}{0.1 \times \frac{1}{4} + 0.2 \times \frac{1}{6} + 0.4 \times \frac{1}{5}} = 0.58$$

تابع توزیع احتمال تجمعی انتخاب یکی از این سه کمان به صورت است:

$$P_{1j} = \begin{cases} 0.18 & j = 2 \\ 0.42 & j = 3 \\ 1 & j = 4 \end{cases}$$

یک عدد تصادفی بین صفر و یک تولید می کنید. این عدد برابر ۰.۶ می شود. در این صورت گره بعدی

برابر گره ۴ است. اکنون از گره ۴ به گره های همسایگی، تابع احتمال را محاسبه می کنیم.



$$p_{43} = \frac{0.3 \times \frac{1}{2}}{0.3 \times \frac{1}{2} + 0.1 \times \frac{1}{5} + 0.4 \times \frac{1}{4}} = 0.56$$

$$p_{46} = \frac{0.1 \times \frac{1}{5}}{0.3 \times \frac{1}{2} + 0.1 \times \frac{1}{5} + 0.4 \times \frac{1}{4}} = 0.07$$

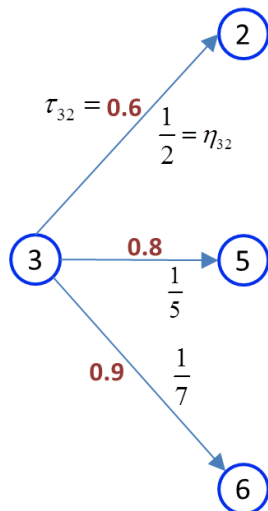
$$p_{47} = \frac{0.4 \times \frac{1}{4}}{0.3 \times \frac{1}{2} + 0.1 \times \frac{1}{5} + 0.4 \times \frac{1}{4}} = 0.37$$

یک عدد تصادفی بین صفر و یک، تولید می شود که برابر با ۰.۳۲ است. با توجه به تابع توزیع احتمال

که به صورت زیر است، گره ۳ انتخاب می شود.

$$P_{4j} = \begin{cases} 0.56 & j = 3 \\ 0.63 & j = 4 \\ 1 & j = 5 \end{cases}$$

اکنون از گره ۳ به گره های همسایگی، تابع احتمال را محاسبه می کنیم.



$$p_{32} = \frac{0.6 \times \frac{1}{2}}{0.6 \times \frac{1}{2} + 0.8 \times \frac{1}{5} + 0.9 \times \frac{1}{7}} = 0.51$$

$$p_{35} = \frac{0.8 \times \frac{1}{5}}{0.6 \times \frac{1}{2} + 0.8 \times \frac{1}{5} + 0.9 \times \frac{1}{7}} = 0.27$$

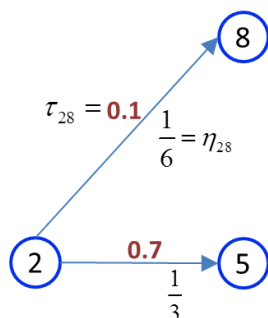
$$p_{36} = \frac{0.9 \times \frac{1}{7}}{0.6 \times \frac{1}{2} + 0.8 \times \frac{1}{5} + 0.9 \times \frac{1}{7}} = 0.22$$

یک عدد تصادفی بین صفر و یک، تولید می شود که برابر با ۰.۴۶ است. با توجه به تابع توزیع احتمال

که به صورت زیر است، گره ۲ انتخاب می شود.

$$P_{3j} = \begin{cases} 0.51 & j = 2 \\ 0.78 & j = 5 \\ 1 & j = 6 \end{cases}$$

اکنون از گره ۲ به گره های همسایگی، تابع احتمال را محاسبه می کنیم.



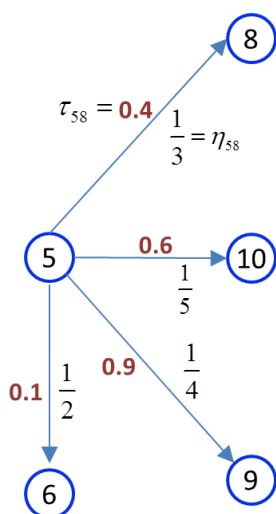
$$p_{28} = \frac{0.1 \times \frac{1}{6}}{0.1 \times \frac{1}{6} + 0.7 \times \frac{1}{3}} = 0.07$$

$$p_{25} = \frac{0.7 \times \frac{1}{3}}{0.1 \times \frac{1}{6} + 0.7 \times \frac{1}{3}} = 0.93$$

یک عدد تصادفی بین صفر و یک، تولید می شود که برابر با ۰.۸۸ است. با توجه به تابع توزیع احتمال که به صورت زیر است، گره ۵ انتخاب می شود.

$$P_{2j} = \begin{cases} 0.07 & j = 8 \\ 1.0 & j = 5 \end{cases}$$

اکنون از گره ۵ به گره های همسایگی، تابع احتمال را محاسبه می کنیم.



$$p_{58} = \frac{0.4 \times \frac{1}{3}}{0.4 \times \frac{1}{3} + 0.6 \times \frac{1}{5} + 0.9 \times \frac{1}{4} + 0.1 \times \frac{1}{2}} = 0.25$$

$$p_{5,10} = \frac{0.6 \times \frac{1}{5}}{0.4 \times \frac{1}{3} + 0.6 \times \frac{1}{5} + 0.9 \times \frac{1}{4} + 0.1 \times \frac{1}{2}} = 0.23$$

$$p_{59} = \frac{0.9 \times \frac{1}{4}}{0.4 \times \frac{1}{3} + 0.6 \times \frac{1}{5} + 0.9 \times \frac{1}{4} + 0.1 \times \frac{1}{2}} = 0.43$$

$$p_{56} = \frac{0.1 \times \frac{1}{2}}{0.4 \times \frac{1}{3} + 0.6 \times \frac{1}{5} + 0.9 \times \frac{1}{4} + 0.1 \times \frac{1}{2}} = 0.09$$

یک عدد تصادفی بین صفر و یک، تولید می شود که برابر با ۰.۲۸ است. با توجه به تابع توزیع احتمال که به صورت زیر است، گره ۱۰ انتخاب می شود.

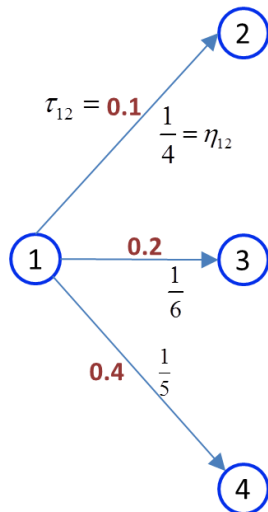
$$P_{5j} = \begin{cases} 0.25 & j = 8 \\ 0.48 & j = 10 \\ 0.91 & j = 9 \\ 1.0 & j = 6 \end{cases}$$

در این حالت، تنها یک گره است که همان گره مقصد است. لذا مورچه ۱ به مقصد رسیده است. حالا به

مورچه ۲ می پردازیم.

مسیریابی توسط مورچه ۲

از گره ۱ به گره های ۲، ۳ و ۴ کمان وجود دارد. احتمال هر یک از سه کمان به صورت زیر بدست می آید.



$$p_{12} = \frac{0.1 \times \frac{1}{4}}{0.1 \times \frac{1}{4} + 0.2 \times \frac{1}{6} + 0.4 \times \frac{1}{5}} = 0.18$$

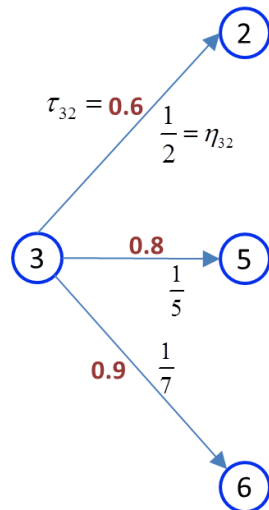
$$p_{13} = \frac{0.2 \times \frac{1}{6}}{0.1 \times \frac{1}{4} + 0.2 \times \frac{1}{6} + 0.4 \times \frac{1}{5}} = 0.24$$

$$p_{14} = \frac{0.4 \times \frac{1}{5}}{0.1 \times \frac{1}{4} + 0.2 \times \frac{1}{6} + 0.4 \times \frac{1}{5}} = 0.58$$

یک عدد تصادفی بین صفر و یک، تولید می شود که برابر با ۰.۲۹ است. با توجه به تابع توزیع احتمال که به صورت زیر است، گره ۳ انتخاب می شود.

$$P_{1j} = \begin{cases} 0.18 & j = 2 \\ 0.42 & j = 3 \\ 1 & j = 4 \end{cases}$$

از گره ۳ به گره های ۲، ۵ و ۶ کمان وجود دارد. احتمال هر یک از سه کمان به صورت زیر بدست می آید.



$$p_{32} = \frac{0.6 \times \frac{1}{2}}{0.6 \times \frac{1}{2} + 0.8 \times \frac{1}{5} + 0.9 \times \frac{1}{7}} = 0.51$$

$$p_{35} = \frac{0.8 \times \frac{1}{5}}{0.6 \times \frac{1}{2} + 0.8 \times \frac{1}{5} + 0.9 \times \frac{1}{7}} = 0.27$$

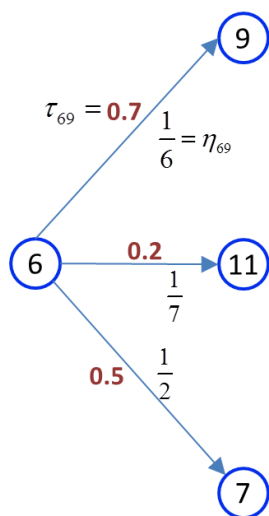
$$p_{36} = \frac{0.9 \times \frac{1}{7}}{0.6 \times \frac{1}{2} + 0.8 \times \frac{1}{5} + 0.9 \times \frac{1}{7}} = 0.22$$

یک عدد تصادفی بین صفر و یک، تولید می شود که برابر با ۰.۸ است. با توجه به تابع توزیع احتمال که به صورت زیر است، گره ۶ انتخاب می شود.

$$P_{3j} = \begin{cases} 0.51 & j = 2 \\ 0.78 & j = 5 \\ 1 & j = 6 \end{cases}$$

از گره ۶ به گره های ۹، ۱۱ و ۷ کمان وجود دارد. احتمال هر یک از سه کمان به صورت زیر بدست می

آید.



$$p_{69} = \frac{0.7 \times \frac{1}{6}}{0.7 \times \frac{1}{6} + 0.2 \times \frac{1}{7} + 0.5 \times \frac{1}{2}} = 0.30$$

$$p_{6,11} = \frac{0.2 \times \frac{1}{7}}{0.7 \times \frac{1}{6} + 0.2 \times \frac{1}{7} + 0.5 \times \frac{1}{2}} = 0.07$$

$$p_{67} = \frac{0.5 \times \frac{1}{2}}{0.7 \times \frac{1}{6} + 0.2 \times \frac{1}{7} + 0.5 \times \frac{1}{2}} = 0.63$$

یک عدد تصادفی بین صفر و یک، تولید می شود که برابر با ۰.۶۶ است. با توجه به تابع توزیع احتمال که به صورت زیر است، گره ۷ انتخاب می شود.

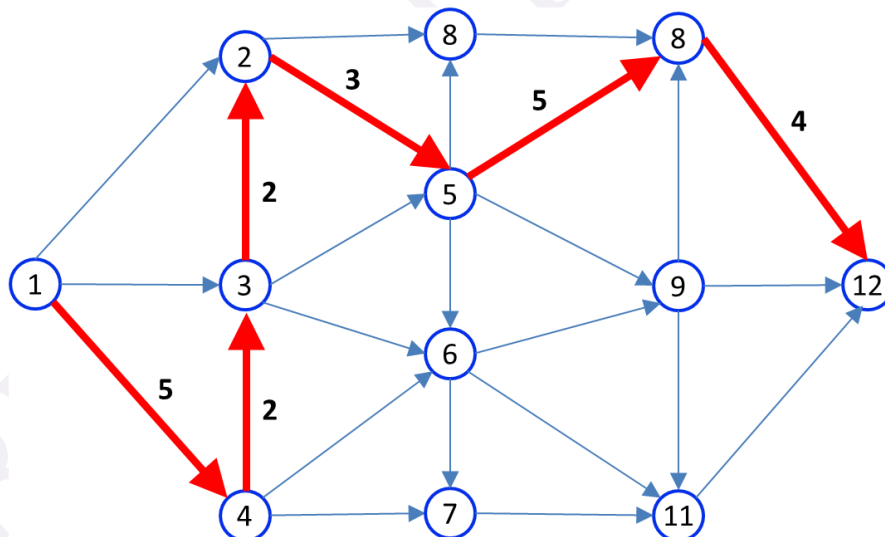
$$P_{6j} = \begin{cases} 0.3 & j = 9 \\ 0.37 & j = 11 \\ 1 & j = 7 \end{cases}$$

چون یک گره فقط در همسایگی است، لذا به گره ۱۱ حرکت می کنیم.

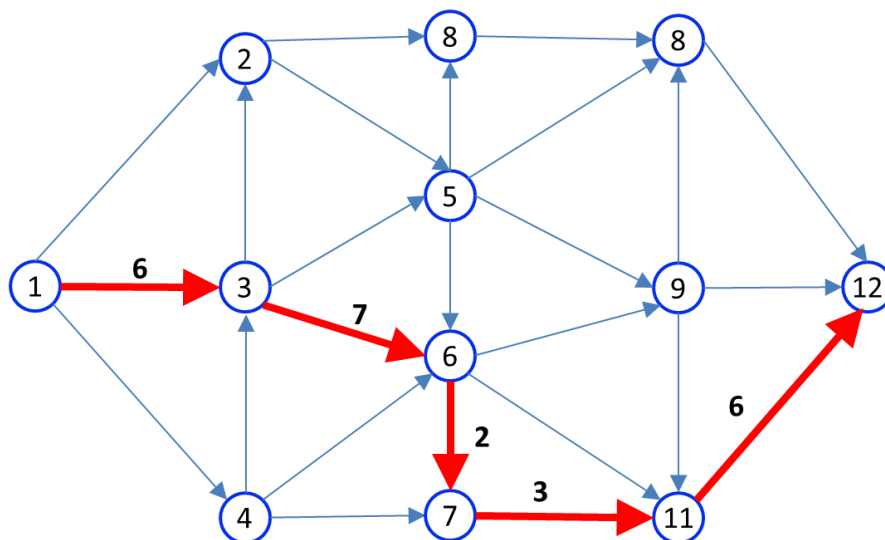


چون فقط یک گره، ۱۲، در همسایگی داریم، لذا به گره ۱۲ می رویم که گره مقصد است و لذا به مقصد می رسیم و مورچه ۲ هم به مقصد می رسد.

مسیر مورچه ۱ به صورت زیر است:



مسیر مورچه ۲ به صورت زیر است.



طول مسیر مورچه ۱ برابر ۲۱ و طول مسیر مورچه ۲ برابر ۲۴ است لذا کیفیت مسیر مورچه ۱ برابر $\frac{10}{21}$ و کیفیت مسیر مورچه ۲ برابر $\frac{10}{24}$ است. بر این اساس، میزان فرومون ریزی برای مورچه های ۱ و ۲ به صورت زیر است.

$$\Delta \tau^1 = \frac{10}{21} = 0.47, \Delta \tau^2 = \frac{10}{24} = 0.42$$

گام ۳: در قسمت اول این گام، تبخیر فرومون براساس میزان تبخیر ۰.۵ صورت می گیرد. در قسمت

دوم این گام، فرومون ریزی براساس میزان فرومون ریزی بر روی کمان های عبوری در مورچه های ۱ و ۲ انجام می شود. در جدول زیر، نتایج محاسبات آورده شده است و ستون آخر مقدار فرومون در انتهای این تکرار آورده شده است. براساس مقدار فرومون جدید، به گام ۲ بر می گردید و مسیریابی برای مورچه ۱ و ۲ تکرار می شود. گام ها ان قدر تکرار می شود تا به گام ۴ که پایان الگوریتم است برسیم.

فرمون جدید	فرمون ریزی مورچه ۲	فرمون ریزی مورچه ۱	فرمون پس از تبخیر	فرمون قبل از تبخیر	گره مقصد	گره مبدا
0.05	0	0	0.05	0.1	2	1
0.52	0.42	0	0.1	0.2	3	1
0.67	0	0.47	0.2	0.4	4	1
0.05	0	0	0.05	0.1	8	2
0.82	0	0.47	0.35	0.7	5	2
0.77	0	0.47	0.3	0.6	2	3
0.4	0	0	0.4	0.8	5	3
0.87	0.42	0	0.45	0.9	6	3
0.62	0	0.47	0.15	0.3	3	4
0.05	0	0	0.05	0.1	6	4
0.2	0	0	0.2	0.4	7	4
0.77	0	0.47	0.3	0.6	10	5
0.45	0	0	0.45	0.9	9	5
0.05	0	0	0.05	0.1	6	5
0.34	0	0	0.34	0.7	9	6
0.1	0	0	0.1	0.2	11	6
0.67	0.42	0	0.25	0.5	7	6
0.47	0.42	0	0.05	0.1	11	7
0.45	0	0	0.45	0.9	10	8
0.35	0	0	0.35	0.7	10	9
0.25	0	0	0.25	0.5	12	9
0.05	0	0	0.05	0.1	11	9
0.72	0	0.47	0.25	0.5	12	10
0.62	0.42	0	0.2	0.4	12	11

برای دریافت بسته‌های آموزشی گروه **بهینه‌یاب** به وب سایت ما به نشانی

www.behinehyab.com مراجعه کنید.

در صورت هر گونه سوال از طریق ایمیل به نشانی behinehyab@gmail.com و یا

بخش تماس با ما وب سایت گروه **بهینه‌یاب** با ما در تماس باشید.

با تشکر از توجه شما

گروه آموزشی **بهینه‌یاب**