



اولین همایش ملی اقتصاد
دانشگاه آزاد اسلامی واحد خمینی شهر، اسفند 1388

انطباق الگوریتم لانه‌ی مورچگان بر مسائل اقتصادی

مهناز دیانی دانشجو کارشناسی اقتصاد نظری دانشگاه آزاد اسلامی واحد خمینی شهر Dayani.mahnaz@gmail.com	مهرداد ترابی دانشجو کارشناسی مکانیک جامدات دانشگاه آزاد اسلامی واحد خمینی شهر Mehrdadorabi64@gmail.com	امیررضا پهلوان پور دانشجو کارشناسی مکانیک جامدات دانشگاه آزاد اسلامی واحد خمینی شهر AR.pahlevanpour@gmail.com
---	---	--

چکیده

امروزه بهینه‌سازی یکی از مسائل مهم پیش روی محققان علوم مختلف از جمله اقتصاد است. در این مقاله به معرفی یکی از جدیدترین تئوری‌های بهینه‌سازی به نام الگوریتم مورچگان پرداخته می‌شود که الهام گرفته از طبیعت است. چون این الگوریتم توانایی حل مسائل در محیط‌های پیوسته را ندارد در ادامه شرایطی معرفی می‌گردد که امکان استفاده از آن در این نوع محیط نیز فراهم شود. سپس برای نشان دادن توانایی الگوریتم، مسئله نگهداری پیش‌گیرانه مطرح گردیده و به وسیله این الگوریتم حل شده است. انتظار می‌رود این الگوریتم در بسیاری از مسائل اقتصادی با پیدا کردن کوتاه‌ترین و در عین حال بهینه‌ترین مسیر باعث کاهش هزینه‌ها گردد.

واژه‌های کلیدی: بهینه‌سازی، الگوریتم لانه‌ی مورچگان، کوتاه‌ترین مسیر، نگهداری پیش‌گیرانه، کاهش هزینه‌ها

مقدمه

بشر تقریباً در همه موارد زندگی روزمره خود به تصمیم‌گیری می‌پردازد و سعی دارد تا بهترین تصمیم را اتخاذ کند. در طول سالیان درازی که جهان با ریاضیات و مسائل آشنا شده است، راه‌های بسیاری برای بهینه‌سازی ارائه شده است، که مطمئناً هر کدام برای نوعی خاص از مسائل کاربرد دارند. انواع ابزارهایی که در بهینه‌سازی مطرح می‌شوند، از ساده‌ترین روش‌های یافتن ریشه توابع ریاضی تا روش‌های پیشرفته ریاضی را شامل می‌شوند. تمام این روش‌ها تعدادی دستور را در بر دارند که باید به ترتیب و پشت سر هم انجام شوند که یک الگوریتم را پدید می‌آورند.

الگوریتم‌های فراابتکاری¹ مجموعه الگوریتم‌هایی برای حل مسائل بهینه‌یابی هستند که به صورت تصادفی اما هدفمند و ساده در فضای مسئله به دنبال جواب بهینه مطلق حرکت می‌نمایند. این روش حرکت معمولاً از طبیعت الهام گرفته می‌شود؛ زیرا برخی از پدیده‌های طبیعی با وجود تصادفی بودن به طرز جالبی دارای حرکتی به سمت حالت‌های نزدیک به حالت بهینه هستند. در سال 1991، دریگو و کلرنی² الگوریتم فراابتکاری مورچگان³ را برای حل مسائل بهینه‌یابی ترکیبی⁴ ارائه نمودند [1]. این الگوریتم برای حل مسائل پیچیده نظیر مسائل فروشنده دوره گرد⁵ و تخصیص درجه دوم⁶ بسیار موفق عمل نمود و مورد توجه محققان قرار گرفت. الگوریتم کلونی مورچه الهام گرفته از نوعی رفتار اجتماعی مورچه‌ها در آنچه که به آن هدف مندی رفتار هر جزء برای ماندگاری جامعه تعریف می‌گردد؛ گرفته شده است. این رفتار را گرس نوعی هماهنگی اجزاء از طریق تغییرات ایجاد شده در محیط⁷ بیان می‌نماید که به گونه‌ای هوشمند هر یک از اجزاء سیستم را در رسیدن به هدف مشترک نشان می‌دهد.

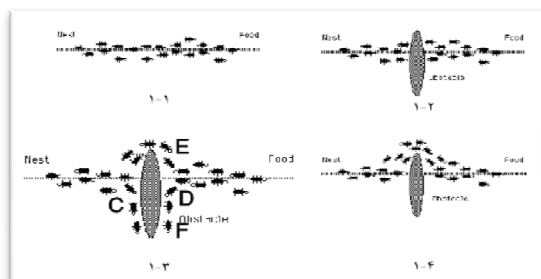
پیدا کردن کوتاه‌ترین مسیر توسط مورچگان

مورچه‌ها هنگام راه رفتن از خود ردی از ماده شیمیایی فرومون به جای می‌گذارند که البته این ماده به زودی تبخیر می‌شود. یک رفتار پایه‌ای ساده در مورچه‌ها وجود دارد:

آنها هنگام انتخاب بین دو مسیر به صورت احتمالی مسیری را انتخاب می‌کنند که فرومون بیشتری داشته باشد یا به عبارت دیگر مورچه‌های بیشتری قبلاً از آن عبور کرده باشند. حال دقت کنید که همین تمهید ساده چگونه منجر به پیدا کردن کوتاه‌ترین مسیر خواهد شد:

همان‌طور که در شکل (1-1) می‌بینیم مورچه‌ها روی مسیر AB در حرکت اند (در دوجبهت مخالف) اگر در مسیر مورچه‌ها مانعی قرار دهیم (شکل 2-1) مورچه‌ها دو راه برای انتخاب کردن دارند. اولین مورچه از A می‌آید و به C می‌رسد، در مسیر هیچ فرومونی نمی‌بیند بنابراین برای مسیر چپ و راست احتمال یکسان می‌دهد و به طور تصادفی و احتمالاتی مسیر CED را انتخاب می‌کند. اولین مورچه‌ای که مورچه اول را دنبال می‌کند زودتر از مورچه اولی که از مسیر CFD رفته به مقصد می‌رسد. مورچه‌ها در حال برگشت و به مرور زمان یک اثر بیشتر فرومون را روی CED حس می‌کنند و آن را به طور احتمالی و تصادفی (نه حتماً قطعاً) انتخاب می‌کنند. در نهایت مسیر CED به عنوان مسیر کوتاه‌تر برگزیده می‌شود. در حقیقت چون طول مسیر CED کوتاه‌تر است زمان رفت و برگشت از آن هم کم‌تر می‌شود و در نتیجه مورچه‌های بیشتری نسبت به مسیر دیگر آن را طی خواهند کرد چون فرومون بیشتری در آن وجود دارد. طبق این تعریف در حل یک مسئله اقتصادی راه حلی استفاده می‌گردد که زودتر به هدف برسد و هزینه کمتری را به سیستم تحمیل کند.

نکته بسیار با اهمیت این است که هر چند احتمال انتخاب مسیر پر فرومون تر توسط مورچه‌ها بیشتر است ولی این کماکان احتمال است و قطعیت نیست. بلکه تنها می‌توان گفت که مثلاً 90% مورچه‌ها از مسیر کوتاه‌تر عبور خواهد کرد. اگر تصادفاً اولین مورچه مسیر CFD (مسیر دورتر) را انتخاب می‌کرد و ردی از فرومون بر جای می‌گذاشت آن‌گاه همه مورچه‌ها به دنبال او حرکت می‌کردند و هیچ وقت کوتاه‌ترین مسیر یافت نمی‌شد. بنابراین تصادف و احتمال نقش عمده‌ای را در الگوریتم بر عهده دارند.



نکته دیگر مسئله تبخیر شدن فرومون بر جای گذاشته شده است. بر فرض اگر مسیر AB بر داشته شود و فرومون تبخیر نشود مورچه‌ها همان مسیر قبلی را طی خواهند کرد، ولی در حقیقت این طور نیست. تبخیر شدن فرومون و احتمال به مورچه‌ها امکان پیدا کردن مسیر کوتاه‌تر جدید را می‌دهند.

تبخیر شدن فرومون و احتمال - تصادف باعث انعطاف در حل هر گونه مسئله بهینه‌سازی می‌شوند. مثلاً در گراف شهرهای مسئله فروشنده دوره گرد، اگر یکی از یال‌ها (یا گره‌ها) حذف شود، الگوریتم این توانایی را دارد تا به سرعت مسیر بهینه را با توجه به شرایط جدید پیدا کند. به این ترتیب که اگر یال (یا گره‌ای) حذف

شود دیگر لازم نیست که الگوریتم از ابتدا مسئله را حل کند بلکه از جایی که مسئله حل شده تا محل حذف یال (یا گره) هنوز بهترین مسیر را داریم، از این به بعد مورچه ها می توانند پس از مدتی کوتاهی مسیر بهینه (کوتاه ترین) را بیابند به عبارت دیگر اگر در یک مسئله اقتصادی شرطی مانند تورم عوض شود دیگر لازم نیست از ابتدا به حل مسئله پردازیم بلکه تا جایی که شرط عوض شده هنوز بهترین راه حل را داریم.

بهینه سازی مسئله فروشنده دوره گرد به روش کلونی مورچه

در مسئله فروشنده دوره گرد یک فروشنده سفر خود را از یک شهر آغاز کرده و پس از یک سفر کامل دوباره به شهر خودش باز می گردد و از هر شهر فقط یک بار عبور می کند و در ضمن باید از همه شهرها عبور کند. هدف یافتن کوتاه ترین مسیر برای این سفر می باشد.

d تعداد شهرها می باشد d_{ij} فاصله بین شهر i و j می باشد و فاصله اقلیدسی بین دو شهر i و j برابر است

$$d_{ij} = [(x_i - x_j)^2 + (y_i - y_j)^2]^{0.5}$$

N یک نمونه از TSP به وسیله یک گراف (N, E) که N مجموعه شهرها و E مجموعه یال ها (فاصله ها) بین شهرها می باشد. الگوریتم مورچه برای حل مسئله فروشنده دوره گرد به کاررفته است؛ $b_i(t), i=1, 2, \dots, d$ را تعداد جمعیت (مورچه ها) در شهر i در زمان t و $m = \sum_{i=1}^d b_i(t)$ کل جمعیت مورچه ها می باشد. هر مورچه یک نماینده ساده با ویژگی های زیر است:

- 1- آن یک شهر را برای رفتن به آن انتخاب می کند که تابعی از فاصله شهر و مقدار بو (اثر) موجود در آن مسیر می باشد. به عبارت دیگر در یک مسئله اقتصادی که با این الگوریتم حل می شود تمام شرایط و محدودیت ها در نظر گرفته می شود و هیچ کدام حذف نمی گردد و شرایطی که مهم تر است، در اولویت قرار می گیرد.
- 2- برای وادار کردن مورچه ها جهت انجام سفرهای منطقی، سفر به شهرهایی که یک بار از آن عبور کرده اند ممنوع می باشد. یعنی در روند حل برای رسیدن به راه حل بهینه و کاهش هزینه ها انتخاب های قبلی که بهینه نبودند دوباره در لیست انتخاب جهت سیکل های بعدی قرار نمی گیرند.
- 3- هنگامی که یک مورچه یک سفر کامل انجام می دهد مقداری فرومون بر روی هر مسیر j, i باقی می گذارد.

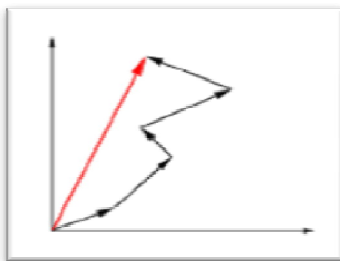
τ_{ij} شدت بو (اثر) در مسیرهای (i, j) در زمان t می باشد. هر مورچه در زمان t شهر بعدی را انتخاب می کند که در زمان $t+1$ در آن خواهد بود. بنابراین اگر یک تکرار از الگوریتم سیستم مورچه m حرکت انجام شده به وسیله m مورچه در فاصله زمانی $(t, t+1)$ در نظر گرفته شود، بعد از d تکرار الگوریتم (یک سیکل) هر مورچه یک سفر کامل انجام داده است. در این نقطه شدت بو (اثر) بر اساس رابطه $\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \Delta \tau_{ij}^k$ محاسبه می شود. ρ یک ضریب است به طوری $(1-\rho)$ که تبخیر ماده بودار را در فاصله زمانی $(t, t+1)$ نشان می دهد و $\Delta \tau_{ij}^k = \sum_{k=1}^m \Delta \tau_{ij}^k$ که $\Delta \tau_{ij}^k$ مقدار ماده بودار در هر واحد طول می باشد که بر روی مسیر (i, j) به وسیله k امین مورچه در فاصله امین مورچه در فاصله $(t, t+1)$ بر جای گذاشته می شود.

طراحی الگوریتم خانه مورچگان برای جستجو در فضا های پیوسته

1-1 مدل سازی محیط لانه:

گونه های موجود و گسسته ی الگوریتم توانایی حل مسائل مدل سازی با قیود زیاد را دارند اما باید در نظر داشت روش خانه مورچگان نمی تواند به طور مستقیم بر فضاهای پیوسته اعمال گردد مگر آن که نوعی خاصی از مرتب سازی خلق گردد. [2]

این مرتب سازی به وسیله ی ارائه یک تعداد محدود از جهات به عنوان بردارهای که از یک نقطه ی مرکزی آغاز می شوند صورت می پذیرد. این بردارها می توانند توسط یک مقدار که نمایش دهنده ی میزان تمرکز فرومون است دارای وزن گردند. در اقتصاد اگر فرومون با اثر بیان کنیم می توان گفت که هر چه اثر گذاری یک متغیر بر سیستم بیشتر باشد احتمال انتخاب آن بیشتر شود که باید در مدل سازی مورد توجه قرار بگیرد. مورچه ها در به روز رسانی این وزن بردارها (فرومون) توانا هستند در نتیجه این مقادیر به احتمال یافتن یک راه حل مناسب در ناحیه ای که توسط آن بردار تعریف می گردد، منجر می شود. نقاطی که توسط بردارها تعیین می گردد می تواند در طول زمان بر اساس میزان موفقیت مورچه ها، توسعه یابد (شکل شماره 2). این کار به طور موثر اجازه می دهد که رد فرومون بر نواحی خاصی منطبق گردد و مورچه ها به طور افزایشده ای در پیدا کردن راه حل های بهتر توانا گردند. (ناحیه اقتصادی در نمودار)



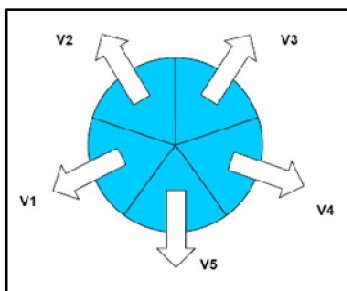
شکل 2

استفاده از بردارهای جهتی و میزان تمرکز فرومون به منظور ذخیره کردن اطلاعات در باره ی مناسب بودن راه حل هایی که مورچه ها پیدا می کنند به دلایل مختلفی مفید است. فایده ی اصلی آن این است که نواحی دارای راه حل های ضعیف به طور موثری اولویت پایین تری خواهند داشت؛ باعث خواهد شد که منابع عددی تنها در نواحی ای که پیش بینی می شود دارای راه حل های مناسب است مورد بررسی قرار گیرد. به زبان دیگر اگر چه ممکن است به طور افزایشده ای شانس کمتری وجود داشته باشد که مورچه ها مسیرهای دارای فرومون کمتر را برای جستجو انتخاب کنند اما جستجوی نواحی دارای اولویت پایین، بطور مطلق ممنوع نخواهد بود و فقط پیشرفت الگوریتم شانس این نواحی کاهش یابد. علت این پدیده نیز آن است که انتخاب جهت بردارها بر اساس تابع احتمال صورت پذیرفته و تضمینی وجود ندارد که بهترین مسیر ها در هر بار تکرار الگوریتم انتخاب شوند. در مسئله چگونگی مصرف بودجه یک کشور فرض هزینه کردن این

بودجه برای زیر سیستم هایی که در اولویت پایین تری قرار دارند از همان ابتدا حذف نمی گردد بلکه با پیشرفت الگوریتم شانس آن ها برای تخصیص بودجه کم می شود.

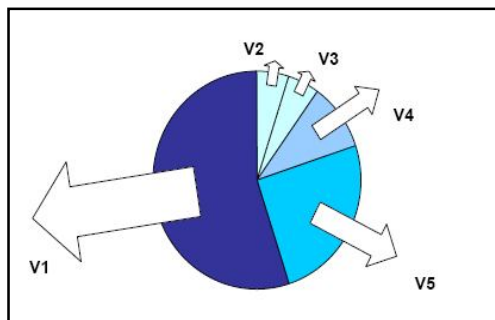
2-1 انتخاب جهت بردار:

هنگامی که یک مورچه از لانه خارج می شود باید در مورد مسیری که در آن سفر خواهد کرد تصمیم بگیرد. این تصمیم تحت تاثیر رد فرمون ای است که خارج از لانه موجود است که این رد فرمون در الگوریتم ما به وسیله بردارهای جهتی دارای وزن نمایش داده می شوند. به منظور تعریف یک مکانیزم مناسب برای انتخاب جهت بردارها که منطبق بر احتمالات باشد یک ساختار ویژه در نظر می گیریم که تا حد زیادی به ماشین گردنده شبیه است. در ابتدا مجموعه ای از n قسمت مساوی که نمایشی از n جهت بردار در سیستم خواهد بود. هر کدام از این n قسمت به یک بازه θ مشخص از اعداد مربوط است سپس یک عدد تصادفی تولید شده که به یک بخش از n بخش مربوط خواهد بود. فرآیند انتخاب تصادفی اعداد و ارتباط آن با هر بخش به فرایند چرخش یک چرخ گردنده شبیه است. (شکل 3). در همان مسئله بودجه یک کشور در ابتدا تمام زیر سیستم ها دارای شانس یکسان برای تخصیص بودجه هستند.



شکل 3

هنگامی که مورچه ها شروع به اکتشاف ویژگی ها و مشخصات محیط مورد جستجو می پردازند، آن ها میزان تمرکز فرمون که با بردار جهت جستجویشان در ارتباط است را نیز به روز می نمایند. وجود سطحی مشخصاً بالا از فرمون در مقایسه با سایر رد فرمون ها نشان دهنده ای آن است که مورچه ها در آن مسیر خاص (بردار خاص) به درجاتی از موفقیت دست یافته اند و در واقع ناحیه ای را یافته اند که احتمالاً شانس یافتن راه حل مناسب در آن زیاد است. حال سیستم به آن بردار خاص بازه ی بزرگ تری از اعداد را اختصاص خواهد داد به معنی آن که اکنون احتمال بزرگ تری در انتخاب آن مسیر خاص هنگامی که عدد تصادفی انتخاب می شود، وجود دارد. در اقتصاد چنین می توان گفت در یک پروژه اقتصادی با تعدادی متغیر، ابتدا متغیرها مورد بررسی و در صورت لزوم به روز می شوند، سپس راه حل هایی که در آن احتمال بیشتری برای رسیدن به هدف مورد نظر با توجه به محدودیت ها پیش بینی می شود دارای احتمال (بازه) بزرگ تری برای استفاده می شوند. (شکل 4)



شکل 4

3-1 نحوه حرکت مورچه ها:

در سیستم خانه مورچگان ، مورچه ها نماینده ای ساده با دیدی محدود هستند که برای جستجوی یک ناحیه از لانه بیرون می روند در این سیستم یک مورچه تقریباً نابینا است و نمی تواند ببیند که سایر مورچه ها چه می کنند. اما مورچه ها می توانند به صورت آگاهانه درباره ی مسیرهایی که برای سفر انتخاب می کنند بر اساس بررسی اطلاعات میزان تمرکز فرمون تصمیم بگیرند.

هنگامی که یک مورچه یک جهت مشخص را برای سفرش بر می گزیند این تصمیم بر اساس آن است که مورچه های موفق قبلی همان جهت را انتخاب کرده اند یا نه، با توجه به این نکته مورچه یک حرکت تصادفی به سوی ناحیه ای جدید را اتخاذ می کند که این حرکت در حداکثر شعاع جستجویی که برای آن مرحله ای جستجو تعریف شده است صورت خواهد پذیرفت. این شعاع جستجو به طور مداوم با پیشرفت فرآیند جستجو کاهش می یابد که باعث می گردد مورچه ها بتوانند به یک راه حل با درجه ی بالایی از دقت دست یابند. روش اولیه ای که برای مدل سازی نحوه ی حرکت مورچه در نظر گرفته می شود آن است که نواحی نزدیک به محل کنونی مورچه از احتمال بالاتری برای جستجو برخوردار باشد. این روش از یک توزیع گوسی⁸ استفاده می کند که بر اساس آن در هنگام انتخاب مسیر حرکت مورچه جا به جایی کوچک بر جابه جایی های بزرگ اولویت دارد. مورچه ها همچنین توسط یک شعاع جستجو محدود می شوند، اگر چه مورچه ها معمولاً علاقه

ای به حرکت تا حداکثر فاصله ی حرکت موجود برای آن ها را ندارند اما این فرایند به طور غیر ضروری مورچه ها را محدود نموده و باعث می گردد که شعاع جستجوی موثر به طور قابل ملاحظه ای از شعاع پیشینه کم تر باشد. برای مقابله با این مشکل نواحی احتمالاتی را به طور کامل از الگوریتم حذف نموده و با یک سیستم ساده تر "موقعیت دهی کاملاً اتفاقی" جایگزین می کنیم.

در این سیستم تمامی نواحی داخلی شعاع جستجو از شانس یک سانی برای انتخاب برخوردارند البته به شرط آن که از شرایط مرزی ناحیه ی اصلی جستجو تجاوز نکنند. هنگامی که یک مورچه به یک مکان جدید حرکت می کند، پس از حرکت میزان موفقیت خود را ارزیابی می نماید. میزان موفقیت یک مورچه معیاری از کار آمدی راه حلی است که موقعیت مکانی مورچه ارائه می دهد. اگر مورچه میزان موفقیتی بزرگ تر از بهترین موفقیت کنونی ثبت شده برای جهت بردار کنونی جستجو را گزارش دهد آن گاه مکان بردار به روز شده و میزان تمرکز فرمون بردار به طور مناسبی افزایش خواهد یافت اما اگر موقعیت مکانی جدید مورچه یک پیشرفت به حساب نیاید به آن توجه ای نخواهد شد و مورچه به بهترین موقعیت قبلی خود در سیکل قبلی باز می گردد.

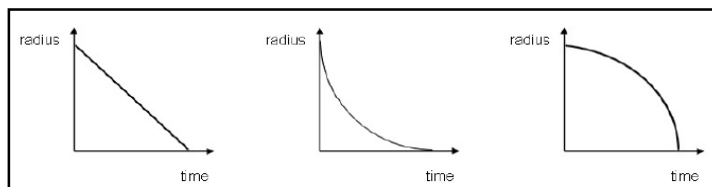
هم چنان که الگوریتم پیش می رود و میزان تمرکز فرمون نواحی پرامید را در محدوده ی جستجو آشکار می سازد، شعاع جستجو برای حرکت همورچه به خصوص شروع به کاهش می کند. این کاهش تدریجی در بازه ای در دسترس برای مورچه ها به الگوریتم اجازه می دهد تا بر یک پاسخ بسیار دقیق متمرکز گردند. کاهش شعاع جستجو به طور موثری به مورچه ها فضای کمتری می دهد که این امر باعث می گردد میزان جستجوی نواحی جدید کاهش یابد و مورچه ها را مجبور می کند که به بهینه های محلی در هنگام جستجو توجه بیشتری نمایند. در مسئله تصمیم گیری برای چگونگی هزینه کردن بودجه یک کشور ابتدا تمام زیر سیستم ها دارای شانس یکسانی برای تخصیص قسمتی از بودجه به آن ها هستند اما با پیشرفت راه حل زیر سیستم هایی که در اولویت پایین تری قرار دارند از احتمال کمتری برای تخصیص بودجه برخوردار می شوند و طبق تعریف راه حل ها برای رسیدن به هدف بهینه کمتر می شود یعنی به زیر سیستم ها با اولویت بالاتر توجه بیشتری می شود.

روش های متفاوتی برای کاهش شعاع در نظر گرفته می شود که تمامی بر سه روش اصلی مبتنی هستند

1- روش اول با یک شیب ثابت با گذشت زمان از شعاع جستجو می کاهد که این خود باعث می گردد در هر سیکل الگوریتم؛ مورچه ها به طور نسبی از فضای جستجو کم تری برخوردار گردند (شکل 5).

2- روش دوم به صورت نمایی با گذشت زمان از شعاع جستجو می کاهد، این روش خاص از کاهش شعاع جستجو به مورچه ها اجازه می دهد که در مراحل ابتدایی الگوریتم با سرعت بیش تری فضای جستجو را کاهش دهند و در مراحل بعدی زمان بیشتری به جزئیات در جستجوی محلی راه حل؛ اختصاص یابد (شکل 6).

3- روش سوم به صورت تابعی مرتبه ی دوم از شعاع جستجو با گذشت زمان می کاهد که این مکانیزم باعث می گردد که مورچه ها زمان بیشتری در جستجو و اکتشاف کل فضای جستجو داشته باشند و تمرکز سریع تری روی راه حل ها در مراحل پایانی الگوریتم بنمایند (شکل 7).



شکل 5

شکل 6

شکل 7

برای تمامی این مدل ها مقدار (یا نرخ) کاهش شعاع بسیار کوچک در نظر گرفته می شود علت نیز آن است که مطمئن شویم الگوریتم قبل از آنکه مورچه ها زمان کافی برای جستجوی کامل فضای جستجو و رد فرمون داشته باشد؛ پایان نیابد. استفاده از نرخ کاهش بسیار اندک هم چنین ما را مطمئن می سازد که مورچه زودتر از موعد بر راه حل هایی که ممکن است بهینه های فراگیر نباشد تمرکز نکند اگر شعاع به سرعت کاهش یابد موقعیتی به وجود می آید که در آن یک مورچه دیگر بازه ی مورد نیاز برای فرار از بهینه های محلی موجود را نخواهد داشت بدیهی است که این رفتار مطلوب نیست و در نتیجه همان گونه که اشاره شد نرخ کاهش کوچکی برای جلوگیری از بروز این رفتار نیاز است.

4-1 مدل سازی چگونگی مصرف مواد غذایی:

مورچه هایی که در فرایند جستجو به موفقیت دست نمی یابند در فرایند رد گذاری شرکت نمی کنند و در فرایند عکس آن (تبخیر رد) کمک می کنند تا توجه از ناحیه مورد بررسی دور گردد. همان گونه که در قبل اشاره شد، فرایند تبخیر رد باعث نمی گردد که یک بردار بطور کامل ناپدید گردد بلکه باعث می گردد که احتمال انتخاب یک بردار با امید پایین هیچ گاه به طور کامل به صفر سقوط نکند و مورچه ها را تشویق می کند که بردارهایی با احتمال موفقیت بالاتر را مورد بررسی بیش تری قرار دهند.

هنگامی که یک مورچه در ناحیه ای گیر می کند که به نتیجه ای مطلوب دست نمی یابد، تمرکز فرمون برداری که مورد بررسی قرار می گیرد شروع به کاهش می کند. بطور واضح اگر یک مورچه نتواند به یک راه حل بهتر دست یابد، فرمون شروع به تبخیر خواهد نمود تا اینکه یک احتمال بسیار کوچک وجود داشته باشد که مورچه بعدی وقت خود را برای جستجو در آن مسیر تلف نماید. اگر این شرایط پیش بیاید (که شبیه مصرف منابع غذایی در یک محیط کلونی واقعی مورچه است)، مورچه باید راه دیگری را برای جستجو انتخاب نماید یا ریسک گرسنگی را به جان بخرد. برای مدل سازی این فرایند واقعی در الگوریتم از مکانیزمی استفاده می کنیم که به موجب آن مورچه هایی که مسیریابی با تمرکز فرمون کمینه راجستجو می کنند "مورچه های گرسنه" شناخته می شوند و در نتیجه به آن ها توانایی های ویژه ای داده می شود. این مورچه ها می توانند از حداکثر شعاع جستجوی تعریف شده در هر لحظه ی مشخص فرایند جستجو تجاوز نمایند. این به آن ها اجازه می دهد که به طور امیدوارانه ای از ناحیه ی ناکارایی که در آن قرار دارند، خارج شده و با منابع غذایی غنی تری روبه رو شوند این مورچه های مایوس در نتیجه می توانند از بهینه های محلی بگریزند و راه حل های بهتری را در جای دیگر جستجو نمایند.



نمودارهای بالا ایده ی نحوه ی مدل سازی مصرف منابع غذایی و مورچه های گرسنه را به خوبی نمایش می دهد. در شکل 3-18 پنج بردار نمایش داده شده است که هر کدام دارای میزان تمرکز یکسانی از فرمون هستند. شرایطی که در مرحله ی اول انجام الگوریتم می تواند موجود باشد. شکل 3-19 نشان دهنده همان سیستم اندکی بعد است که بردارهای V1 و V2 به طور قابل ملاحظه ای پیشرفت نموده اند ولی سه بردار باقی مانده خیر. شکل 3-20 شرایطی را نشان می دهد که بردارهای V3 و V4 و V5 آن قدر فرمون شان تبخیر شده است که به حداقل میزان مجاز خود رسیده است از این زمان به بعد به هر مورچه ای که یکی از این بردارها را برای جستجو انتخاب کنند توانایی های ویژه ای اعطاء خواهد شد و اجازه خواهد داشت، تا فواصل بسیار دور برای جستجو حرکت نماید. در شکل 3-20 می بینیم که بردار V5 به طور قابل ملاحظه ای پیشرفت کرده است که این خود نشان دهنده ی آن است که یکی از مورچه های گرسنه به طور موفقیت آمیزی به یک موقعیت مکانی بهتری دست یافته است هنگامی که یک بردار به طور مکفی باز یابی گردد. مورچه های گرسنه باز به شرایطی قبلی خود باز می گردند یعنی مورچه ی عادی به شمار آمده و شعاع عادی مرزی جستجو باز اعمال می گردد.

1-5 مدل سازی شرایط ابتدایی:

لانه نقطه ای است که از آن تمامی مورچه ها در ابتدا شروع به حرکت می نمایند و معمولاً در مرکز فضای جستجو قرار می گیرد. این مکان به وسیله ارزیابی مرزهای ناحیه ی جستجو محاسبه گشته به مثابه مبدأ مختصات برای بردارها خواهد بود. البته توجه به این نکته ضروری است برای مسائلی که در آن لانه به طور اتفاقی دقیقاً در مکان بهینه ی قطعی سیستم قرار گرفته است مکان لانه را می توان به طور دستی تغییر داد. تعیین مکان جایگزین برای لانه می تواند به روش های مختلفی انجام گیرد به عنوان مثال: تعیین تصادفی مکان یا قرار دهی لانه در مکانی به اندازه ی کافی دور.

1-6 شرایط پایانی الگوریتم:

یکی از نکات مهمی که برای طراحی یک الگوریتم باید در نظر گرفت آن است که تصمیم بگیریم چه شرایطی باید برای الگوریتم اتفاق بیافتد تا آن را پایان یافته تلقی کنیم. شرایطی که الگوریتم تحت آن ها استفاده می شود زیاد و متنوع است، اما بهترین راه جهت استفاده در مسائل اقتصادی آن است که شعاع جستجو به صفر برسد. این حاکی از آن است که مورچه ها دیگر نمی توانند پیش روی کنند زیرا شعاع صفر به معنی فضای جستجوی صفر برای هر مورچه است. شرط شعاع مساوی صفر باشد در واقع الگوریتم را به تعداد ثابتی از ارزیابی (تا زمانی که کاهش شعاع فقط به زمان بستگی داشته باشد) محدود می کند. بنابراین اگر ما اجازه دهیم که کاهش شعاع تدریجی باشد می توانیم به میزان بیشتری مطمئن باشیم که مورچه ها به پاسخی دست یافته اند که تا حد زیادی دقیق است. در نهایت، اگر الگوریتم را می خواهیم تست کنیم و میزان سرعت دست یابی به پاسخی را دقیقاً را بیابیم می توانیم شرط اختتام الگوریتم را آن طور تعیین کنیم که پاسخ به دست آمده توسط الگوریتم در یک بازه ی داده شده و قابل قبول به آن پاسخ دقیقی که از قبل می دانیم نزدیک باشد این به ما اجازه می دهد که تعیین نماییم دقیقاً به چند ارزیابی قبل از آن که الگوریتم به طور موفقیت آمیزی به پاسخی به حد کافی نزدیک به پاسخ قطعی دست یابد، نیاز است.

خلاصه ای از روند حل مسئله به وسیله الگوریتم مورچگان

در این قسمت به طور مختصر و کیفی به تشریح مسئله تصمیم گیری برای چگونگی هزینه کردن بودجه عمرانی یک استان در یک سال کاری می پردازیم. قبل حل مسئله ابتدا باید متغیرها مشخص شده و میزان تاثیر گذاری آن ها با توجه به محدودیت ها در نظر گرفته شود. در موضوع مورد بحث هر پروژه عمرانی دارای تعدادی متغیر است برای مثال در ساخت یک پل باید هزینه و مدت زمان ساخت و... را به عنوان متغیر در نظر گرفت سپس میزان تاثیر گذاری هر پروژه را بر زندگی روزمره مردم مورد بحث قرار داد. (هر پروژه روزانه چند صرفه جویی به همراه دارد اعم از صرفه جویی مالی، وقت و...) پس از مشخص شدن متغیرها و اولویت ها به حل مسئله می پردازیم. یاد آوری این نکته ضروری است که متغیرها در صورت تغییر قابل به روز شدن هستند یعنی اگر در یک پروژه عمرانی برای مثال مصالح گران شود می توان این متغیر را تغییر داد و لازم نیست از ابتدا به حل مسئله پردازیم. به علت استفاده از تابع احتمال در الگوریتم تمام پروژه ها دارای شانس یکسان برای اجرایی شدن هستند. حل مسئله که پیش می رود تک تک پروژه ها با تمام متغیرهایشان مورد بررسی قرار می گیرد پروژه هایی که دارای اولویت

بالا تری با توجه به محدودیت ها هستند دارای شانس بیشتری می شوند که در الگوریتم با معرفی وزن بردارها مشخص شده است و از طرفی الگوریتم بر پروژه هایی متمرکز می شود که مهم تر است. با پیشرفت مسئله راه حل های مسئله کوچک و کوچک تر می شود. حال ممکن است اولویت ها نیز در میان روند حل نیز عوض شوند که در این صورت باز هم الگوریتم در به روز رسانی آن ها توانمند است. حال پس از طی چند سیکل که راه حل بهینه نهایی مشخص شد حل مسئله پایان یافته است و این گونه مهم ترین پروژه مشخص می شود.

کاربرد الگوریتم مورچگان در نگهداری پیش گیرانه

توانایی واحد صنعتی در تولید محصولات متنوع با کیفیت مطلوب و با کمترین هزینه تمام شده به عنوان مهمترین شاخص ارزیابی مطرح می باشد. عوامل متعددی بر کاهش هزینه تمام شده محصول نهایی تاثیر دارند که مجموعه آن ها را می توان در قالب مدیریت بهره وری تولید تعریف نمود. مدیریت بهره وری بر دو اصل اساسی استوار است:

الف) استفاده به موقع از منابع موجود

ب) استفاده مؤثر از امکانات و تجهیزات موجود در واحد صنعتی جهت بهره مندی هرچه بیشتر از منابع اولی

یکی از مهمترین عوامل مؤثر در بهبود بهره وری تولید و کاهش قیمت تمام شده محصول تولیدی، اجرای راهکارهای صرفه جویی انرژی می باشد. مهمترین دلایل اتلاف انرژی در صنایع عبارت است از:

الف) پایین بودن راندمان عملیاتی سیستم

ب) عدم استفاده مناسب از ظرفیت های موجود در سیستم (موجود و مازاد)

ج) عدم به کارگیری برنامه تعمیرات و نگهداری پیش گیرانه مناسب

نگهداری پیش گیرانه⁹ شامل یک سری از فعالیت های تکنیکی، اجرایی و مدیریتی برای بهینه سازی طول عمر اجزاء به منظور بهبود در دسترس بودن و قابلیت اطمینان یک سیستم (کاهش احتمال شکست یا سطح تنزل اجزاء یک سیستم) می باشد. پس از انجام نگهداری پیش گیرانه اجزاء مانند ابتدای کارشان، نو می شوند. سیاست نگهداری پیش گیرانه موضوع بسیاری از مطالعات در سال های اخیر بوده است این مطالعات معیارهای مختلفی از قبیل هزینه، عمر اقتصادی، ریسک یا ترکیبی از این معیارها را مد نظر قرار داده اند. سیاست نگهداری پیش گیرانه به طور گسترده در سیستم های بزرگ از قبیل سیستم های حمل و نقل، سیستم های تولید و غیره مورد استفاده قرار می گیرد. [3]

از اجرای سیستم تعمیرات و نگهداری پیش گیرانه نتایج زیر قابل انتظار می باشد:

1. توسعه اتوماسیون در خط تولید

2. کاهش هزینه نیروی انسانی

3. افزایش بهره وری تولید

4. بهبود کیفیت تولید

5. استفاده بهینه از طول عمر مفید دستگاه

6. کاهش مصرف انرژی

7. توسعه بستر مناسب جهت بهبود مکانیزم زمان سنجی در فرآیند تولید

در سیستم نگهداری پیش گیرانه مجموعه به تعدادی زیر سیستم موازی و سری تقسیم بندی می شود که هر کدام از آن ها نیز دارای اجزائی می باشد. ابتدا تعدادی از متغیرهای مورد نیاز را تعریف می کنیم و تعدادی از آن ها را نیز در حین حل مسئله معرفی می کنیم:

*TM

* $To = (To(1); To(2); \dots; To(N))$

بردار زمان اولین بازرسی

$Tp = (Tp(1); Tp(2); \dots; Tp(N))$

* $P(i, j)$

بردار حل پریودهای بازرسی اجزاء سیستم

* Tpk

بردار حل مربوط به K امین مورچه

* Ek

تعداد اجزاء k امین زیر سیستم موازی

در دسترس بودن سیستم در $A(t)^*$

t

زمان

* Ao

محدودیت در دسترس بودن - حد پایین

* AS1; AS2; ACS1

سه الگوریتم مورچه

* $C(e(i; k))$

هزینه یک باز بازرسی i امین جزء در K امین زیر سیستم موازی

*GA

الگوریتم ژنتیک

* Tp^k

لیست زمان هایی که k امین مورچه می تواند برای i امین جزء انتخاب کند

*N

تعداد کلی اجزاء

تکنیک بهینه سازی هزینه :

مساله نگهداری پیش گیرانه مورد مطالعه :

سیاست نگهداری پیش گیرانه ، تعیین شرایط تعویض اجزاء (تاریخ ها و اجزاء مورد بازرسی) در سیستم های سری و موازی با کمینه کردن هزینه مورد ملاحظه قرار می دهند . راه حل های یافته شده محدودیت های در دسترس بودن و قابلیت اطمینان را مد نظر قرار می دهند. [4]

هزینه بازرسی نگهداری کل به این صورت تعریف می شود :

$$C_{PM} = \sum_{K=1}^K \sum_{i=1}^{E_K} n_e(i, k) \times C_{(e(i,k))}$$

* در صورتی که هزینه بازرسی ها برای همه اجزاء ثابت باشد :

$$n_{e(i,k)} = 1 + \left[\frac{T_{M(e(i,k))} - T_{P(e(i,k))}}{T_{P(e(i,k))}} \right]$$

[] علامت جزء صحیح در عبارت فوق می باشد .

TM و Tp و To زمان های ماموریت (هدف) ، پریود بازرسی ، زمان اولین بازرسی (به ترتیب) برای I امین جزء در K امین زیر سیستم موازی هستند. و تعداد کل اجزاء برابر است با :

$$N = \sum_{K=1}^K \sum_{I=1}^{E_K} e(i, k)$$

هزینه کل نیز معادل $C_{PM} = \sum_{k=1}^K \sum_{i=1}^{E_K} \sum_{j=1}^{n_{e(i,k)}} C_{J(e(i,k))}$ می باشد که $n_e(i; k)$ تعداد کل بازرسی های i امین جزء در k امین زیر سیستم موازی در طول

زمان ماموریت ارائه می دهد. $C_{J(e(i,k))}$ هزینه J امین بازرسی i امین جزء در k امین زیر سیستم موازی باشد و k تعداد زیر سیستم های موازی است.

مسیر حل مسئله

هر جزء لیستی از زمان های بازرسی ممکن دارد که آن زمان ها به طور تصادفی در فاصله زمانی $[L_B, U_B]$ بر اساس یک توزیع یکسان ایجاد شده اند. (LB حد پایین و UB حد بالا است) هر زمان یک راه حل بالقوه است. تعداد زمان های ایجاد شده مستقل از تعداد اجزاء است و با محدودیت های مساله مرتبط است. به عنوان مثال برای $A_0=0.5$ ، $A_0=0.8$ این تعداد به طور قابل ملاحظه ای افزایش می یابد. هر بازرسی در حالی که از یک زیر سیستم به زیرسیستم دیگر حرکت می کند راه حلی برای حل این مساله ارائه می دهد. هر زمان که الگوریتم (مورچه) از یک لیست به لیست دیگر حرکت می کند، لیست بعدی را برای جستجوی بهترین زمان ، مطابق با جزء مورد ملاحظه ، کاوش می کند. زمان انتخاب شده در لیست ممنوع قرار داده می شود به طوری که مورچه های دیگر در همان سیکل آن را انتخاب نخواهند کرد و بنابراین جستجو را در فضای حل ممکن متنوع می سازد.

در طول یک سیکل هر مورچه $K = 1, \dots, m$ یک سفر کامل انجام می دهد . این سفر یک بردار از زمان های ممکن برای نگهداری پیش گیرانه اجزاء و سیستم است . تعداد عناصر این بردار برابر تعداد اجزاء است که در مسئله حل شده در این قسمت 1 می باشد.

هر سفر $T_p^k(t)$ یک بردار کاندیدا برای بهترین بردار T_p یافته شده در این روش می باشد . در طول این سفر انتخاب زمان بازرسی بعدی یعنی زمان J از زمان جاری i به معیار های متعددی بستگی دارد:

1- زمان J تا به حال رویت شده باشد یا نشده باشد . مورچه ها یک حافظه شخصی (یا لیست ممنوع) از زمان ها دارند که در طول سفر به دست آورده اند . زمان های رویت شده در لیست ممنوع قرار داده می شوند . لیست ممنوع در شروع هر تکرار به صفر برگردانده می شود .

2- زمان های بازرسی ، ترجیحا تا حد ممکن احتمالی هستند. در نتیجه برای هر دو لیست متوالی یک ماتریس $P(i; j)$ (ماتریس قابلیت مشاهده) که شامل ضرائب تصادفی بین یک و صفر هستند و بر اساس یک نظم صعودی چیده شده اند به طوری که ضرایب بزرگ تر مربوط به فواصل زمانی دورتر (زمان های بازرسی دورتر) هستند ، تعریف می کنیم .

مقدار فرمون با مسیری که زمان I را به زمان J وصل می کنند مرتبط است . این اطلاعات پویا مطلوبیت حاصل را نشان داده و بر انتخاب J به عنوان مقصد I تاثیر می گذارد . در واقع فرمون در پایان هر سیکل برای انعکاس آزمایش مورچه ها (کیفیت راه حل بدست آمده) در طول سفرشان تعدیل می شود. به عنوان مثال آن به طور معکوس متناسب با ارزش هزینه T_p^k تعدیل می شود به طوری که سفرهای مورچه ای که هزینه های بالایی دارند در سیکل های بعدی مطلوبیت کم تری خواهند داشت.

جهت آشنایی با انواع الگوریتم مورچگان و توضیحاتی پیرامون آن ها به پیوست 1 مراجعه شود.

مطالعه موضوع

باید دقت کنیم که سیاست نگهداری مذکور محدودیت های دسترس بودن از قبیل $A_0 A(t) \geq A_0$ دسترس بودن (حد پایین) ارائه می دهد و همه ی زمان ها $0 < T < TM$ و یک زمان هدف T را در نظر می گیرد . محاسبات برای هدف محاسبه شده است : 25 و 50 سال . برای زمان هدف 50 سال دو ارزش A_0

در نظر گرفته شده است: 0.8; 0.9 در حالت 25 سال محاسبات برای AO=0.8 انجام شده است. یک مقایسه بین الگوریتم ژنتیک و الگوریتم های مورچه از نظر هزینه و زمان صرف شده برای یافتن این راه حل ها به عمل آمده است دو پارامتر برای مقایسه الگوریتم های مورچه با الگوریتم ژنتیک مورد استفاده قرار گرفته است: GAPt که برای مقایسه زمان ها و GAPc که برای مقایسه هزینه ها به کار می رود و به شکل زیر محاسبه می شوند.

$$GAPt(GA, AS1) = \frac{t_{GA} - t_{AS1}}{t_{GA}} \quad (2)$$

$$GAPt(GA, AS1) = \frac{t_{GA} - t_{AS1}}{t_{GA}} \quad (3)$$

$$GAPt(GA, AS1) = \frac{t_{GA} - t_{AS1}}{t_{GA}} \quad (4)$$

$$GAPc(GA, AS2) = \frac{C_{GA} - C_{AS2}}{C_{GA}} \quad (5)$$

$$GAPc(GA, AS2) = \frac{C_{GA} - C_{AS2}}{C_{GA}} \quad (6)$$

دست آمده توسط ACS1, AS2, AS1, GA می باشد. [5]

نتیجه گیری

محاسبات برای زمان هدف TM= 50 سال و محدودیت در دسترس بودن AO=0.8 راه حل های به دست آمده در جدول شماره 1 ارائه شده است. همه الگوریتم ها راه حل هایی با هزینه های نگهداری مساوی برای زمان هدف داده شده تولید کرده اند. به هر حال زمان پردازش برای الگوریتم ها متفاوت است. ACS1; AS1 سریع تر از الگوریتم ژنتیک (GA) بودند: [6]

$GAPt(GA; ACS1) = 0.70$, $GAPt(GA; AS1) = 0.49$ در این جا باید دقت نمود که AS2 کند ترین الگوریتم بوده است:

$GAPt(GA; AS2) = -7.12$ تعداد زیاد تکرار می تواند دلیل این کندی باشد.

جدول 1: بهترین هزینه ها و بردارهای حل بدست آمده (Tp) با پربردهای بازرسی اجزاء سیستم

اجزاء	1	2	3	4	5	6	7	8	9	10	11
To	18	15	12	13	20	10	16	12	11	10	10
Tp(GA)	10.6	13.7	13.4	14.3	13.7	14.5	11.5	10.5	14.25	15.21	14.93
CGA	153.9	153.9	153.9	153.9	153.9	153.9	153.9	153.9	153.9	153.9	153.9
Tp(ACS1)	14.72	15.42	11.24	14.66	13.75	15.06	12.55	15.19	13.2	15.09	13.24
CACS1	153.9	153.6	153.9	153.9	153.9	153.9	153.9	153.9	153.9	153.9	153.9
Tp(AS1)	11.29	15.3	14.77	15.43	15.48	15.24	15.47	15.02	15.29	15.27	14.86
CAS1	153.9	153.9	153.9	153.9	153.9	153.9	153.9	153.9	153.9	153.9	153.9
Tp(AS2)	9.57	14.04	15.05	14.08	13.24	15.08	15.45	14.75	12.64	15.15	13.21
CAS2	153.9	153.9	153.9	153.9	153.9	153.9	153.9	153.9	153.9	153.9	153.9

به طور کلی مشاهده شد که استفاده از الگوریتم لانه مورچگان نه فقط میزان سرعت و دقت دسترسی به پاسخی دقیق را افزایش داد؛ بلکه با تعیین صحیح پارامترهایی از پیش تعریف شده برای الگوریتم، منجر به یافتن راه حل های منطقی شد. هم چنین نتایج به دست آمده در یک سری از فعالیت های تکنیکی، اجرایی و مدیریتی (با هدف بهینه سازی طول عمر اجزا و کاهش هزینه اقتصادی) نشان داد که الگوریتم مورچگان در مقایسه با الگوریتم های دیگر (الگوریتم ژنتیک و...) بسیار کارآمدتر عمل کرده و نتایج ارائه شده با مشاهدات تجربی کاملاً منطبق است.

به طور کلی الگوریتم مورچگان دارای ویژگی های زیر است:

1. چند منظوره می باشد، به عبارت دیگر می تواند برای انواع مشابه یک مسئله به کار رود.
2. قوی می باشد، یعنی با کمترین تغییرات برای دیگر مسائل بهینه سازی ترکیبی به کار برده می شود.

3. یک روش مبتنی بر جمعیت می‌باشد.

در نهایت پیشنهاد می‌شود از این الگوریتم در مسائل عرضه و تقاضا استفاده شود چون همان گونه که اشاره شد الگوریتم توانایی به روز رسانی کلیه متغیرها، محدودیت‌ها و اولویت‌ها را حتی در حین حل مسئله دارد. امید بر آن است که نتایج حاصل از این مقاله مورد توجه و کاربرد استادان و پژوهشگران عزیز قرار گیرد و بتواند به عنوان مقدمه‌ای برای سایر پژوهش‌ها جهت استفاده از این الگوریتم مهندسی در مسائل اقتصادی گردد.

تشکر و قدردانی

در پایان از مساعی استاد گرانقدر جناب آقای دکتر مصطفی رجیبی که ما را صمیمانه در تهیه ی این مقاله یاری نمودند و متحمل زحمات فراوانی شدند، تقدیر و تشکر می‌نماییم.

همچنین از زحمات مهندس امیرحسین پهلوان پور نیز قدردانی می‌نماییم.

مراجع

- [1] Marco Dorigo and Thomas stutzle. Ant colony optimization
- [2] Santhosh Swaminathan, B.E. Rule induction Ant colony optimization for mixed variable attributes
- [3] R. Meziane, Y. Massim, A. Zeblah, A. Ghoraf and R. Rahli. Reliability optimization using ant colony algorithm under performance and cost constraints. Electric power systems research .2005; 76:1-16
- [4] Amr Badr and Ahmed fahmy. A Proof of convergence of ants algorithms. Information science 2004 ; 160 : 267-279
- [5] Coit D, Smith A. Reliability optimization of series-parallel systems using genetic algorithm . IEEE Trans Reliab 1996; 45(2): 254-266
- [6] Bris R, Chatelete E, Yalaoui F. New methods to minimize the preventive maintenance cost of series –parallel systems .Reliab Eng syst saf 2003;82:

پیوست 1

1-1 انواع الگوریتم خانه مورچگان

سه نوع الگوریتم مورچه وجود دارد:

Ant Cycle-1

Ant Quantity-2

Ant Density-3

در دو الگوریتم اول مقدار فرمون در هر تکرار تعدیل می شود (به روز در آورده می شود) اما در الگوریتم سوم بعد از یک سیکل عمل تعدیل صورت می گیرد.

1-2 بررسی انواع الگوریتم ها

در الگوریتم اول داریم:

$$\Delta \tau_{ij}^k(t, t+1) = \begin{cases} Q_1 & \text{اگر } k \text{ امین مورچه در فاصله زمانی } (t, t+1) \text{ از } i \text{ به } j \text{ می رود} \\ 0 & \text{در غیر این صورت} \end{cases}$$

که Q یک مقدار ثابت است که در هر واحد طول در (i, j) مسیر بر جای گذاشته می شود.

در الگوریتم دوم:

اگر k امین مورچه در فاصله زمانی $(t, t+1)$ از i به j برود.

$$\Delta \tau_{ij}^k(t, t+1) = \begin{cases} Q_2/d_{ij} & \\ 0 & \text{در غیر این صورت} \end{cases}$$

که Q_2 یک مقدار ثابت و d_{ij} فاصله بین شهر i, j میباشد.

در الگوریتم سوم:

$$\Delta \tau_{ij}^k(t, t+d) = \begin{cases} Q_3/L^k & \text{اگر } k \text{ امین مورچه از مسیرهای } (i, j) \text{ در سفرش استفاده کند} \\ 0 & \text{در غیر این صورت} \end{cases}$$

که Q_3 یک مقدار ثابت و L^k طول سفر k امین مورچه می باشد.

ضریب ρ باید دارای ارزش کوچک تراز یک باشد تا از انباشتگی نامحدود ماده بودار جلوگیری کند. در این مقاله شدت بو (اثر) در زمان صفر یعنی $\tau_{ij}(0)$ برابر یک مقدار ثابت C در نظر گرفته شده است. به منظور ارضای این محدودیت که یک مورچه باید از همه d شهر مختلف دیدن کند (عبور کند) به هر مورچه یک ساختار داده به نام لیست ممنوع نسبت داده شده است که شهرهایی را که مورچه تاکنون دیدن نموده است ذخیره می کند و مانع از آن می شود که تا قبل از پایان تکرار (یک سیکل)

مورچه دوباره از آنها عبور کند. هنگامی که یک سفر کامل انجام شد لیست ممنوع¹⁰ برای محاسبه راه حل جاری مورچه مورد استفاده قرار می گیرد (طول سفری که بوسیله مورچه انجام شده است) لیست ممنوع سپس تخلیه شده و مورچه ها دوباره برای انتخاب کردن آزاد گذاشته می شوند $Taboo_k$ یک بردار رشد یابنده پویا است که شامل لیست ممنوع k امین مورچه است.

برای قابلیت رویت (visibility) η_{ij} و مقدار $1/d_{ij}$ تعریف میشود. بنابراین احتمال گذر (انتقال) از شهر i به شهر j برای k امین مورچه به صورت زیر خواهد بود:

$$p_{ij}^k(t) = \begin{cases} \frac{(\tau_{ij}(t))^\alpha \times (\eta_{ij})^\beta}{\sum_0 (\tau_{ik}(t))^\alpha \times (\eta_{ik})^\beta} & \text{If } j, k \in \text{allowed}_k \\ \text{در غیر اینصورت} \end{cases}$$

در رابطه فوق $allowed_k = \{N - Taboo_k\}$ و α و β پارامترهایی هستند که اهمیت نسبی بو (اثر) را در مقابل قابلیت رویت (visibility) کنترل می کنند. بنابراین احتمال گذر (انتقال) یک مبادله بین قابلیت رویت (که می گوید شهرهای نزدیک باید با احتمال بالایی انتخاب شوند) و شدت بو (اثر) در زمان t (که می گوید اگر در مسیر i به j ترافیک زیادی وجود داشته باشد، آن مسیر بسیار مطلوب خواهد بود یک فرآیند خود تقویتی را به کار می گیرد) می باشد.

3-1 معرفی الگوریتم ها

با توجه به تعاریف قسمت قبل الگوریتم ها معرفی می شوند. در زمان صفر یعنی مرحله شروع، مورچه ها در شهر های مختلف مستقر می شوند و ارزش اولیه شدت بو (اثر) بر روی مسیرها $\tau_{ij}(0)$ تعیین می شود. شهر آغازین به عنوان اولین عنصر وارد لیست می شود. سپس هر مورچه از شهر i به سمت j شهر با تابع احتمال π_{ij} حرکت می کند که خود تابعی از دو معیار مطلوبیت می باشد. یکی $\tau_{ij}(t)$ که اطلاعاتی راجع به اینکه در گذشته چه تعداد مورچه از مسیر (i, j) رفته اند ارائه می دهد و دیگری قابلیت رویت (visibility) که می گوید شهرهای نزدیک برای مورچه ها مطلوب تر هستند.

بعد از d تکرار همه مورچه ها یک سفر کامل انجام داده اند و لیست ممنوع آنها پر شده است. در این نقطه برای هر مورچه ارزش L^k محاسبه شده و $\Delta\tau_{ij}$ تعدیل می شود. همچنین کوتاه ترین مسیر یافته شده به وسیله مورچه ها ذخیره شده و لیست های ممنوع خالی می شوند.

این فرایند تا زمانی که تعداد سیکل ها به حد اکثر خود برسد یا همه مورچه ها یک سفر یکسان انجام دهند ادامه می یابد. این حالت را رفتار رکودی می نامند زیرا الگوریتم جستجوی راه حل ها یکدیگر را متوقف می کند.

در دو الگوریتم Ant Density و Ant Quantity مقدار فرومون در پایان هر تکرار تعدیل می شود اما در الگوریتم Ant Cycle بعد از پایان یک سیکل این عمل انجام می گیرد. پس از انجام شدن چند سیکل فقط به مورچه ای که بهترین مسیر را می پیماید اجازه ترشح کردن فرومون داده می شود، این عمل به منظور جهت دار کردن جستجو انجام می گیرد یعنی با این کار مورچه ها در همسایگی بهترین مسیری که تاکنون یافته شده به جستجو می پردازند.

دو الگوریتم اول دقیقا مثل هم می باشند و فقط از لحاظ تعدیل کردن فرومون (update) با هم متفاوت هستند. این الگوریتم ها برای نمونه در صفحه ی بعد آورده شده اند.

```

1: Initialize
Set  $t=0$ 
For every edge  $(i,j)$  set an initial value  $\tau_{ij}(t)$  for trail intensity and  $\Delta\tau_{ij}(t,t+1)=0$ 
Place  $b_i(t)$  ants on every node  $i$ 
Set  $s=1$ 
For  $i=1$  to  $d$  do
For  $k=1$  to  $b_i(t)$  do
 $taboo_k(s) = i$ 
2. Repeat until taboo list is full (step repeated  $(d-1)$  times)
.
2. 0 set  $s=s+1$ 
2. 1For  $i=1$  to  $d$  do
For  $k=1$  to  $b_i(t)$  do
Choose the town  $j$  to move to with probability  $P_{ij}(t)$ 
Move  $k$ th ant to  $j$  { this instruction creates new values  $b_j(t+1)$  }

Insert node  $j$  in  $taboo_k(s)$ 

Set  $\tau_{ij}(t,t+1) = \Delta\tau_{ij}(t,t+1) + Q1$  (Ant density model)
 $\tau_{ij}(t,t+1) = \Delta\tau_{ij}(t,t+1) + Q2 / dij$  (ant quantity model)

2.2For every edge  $(i,j)$  compute  $\tau_{ij}(t,t+1)$  according to equation
3. Memorize shortest tour found up until now
If  $(t < T)$  or (not all ants chose the same tour)
then
Empty all  $taboo$  lists
Set  $s=1$ 
For  $i=1$  to  $d$  do
For  $k=1$  to  $b_i(t)$  do
 $taboo_k(s) = i$ 
Set  $t = t+1$ 

For every edge  $(i,j)$  set  $\Delta\tau_{ij}(t,t+1)=0$ 
Go to step 2
else

Print shortest tour and stop

```

1-4 ساختار کلی الگوریتم

ساختار اصلی الگوریتم ارائه شده توسط دوریگو بدین گونه است:

```

Initialize colony ()

Evaluate (t)

While (not termination_condition) {

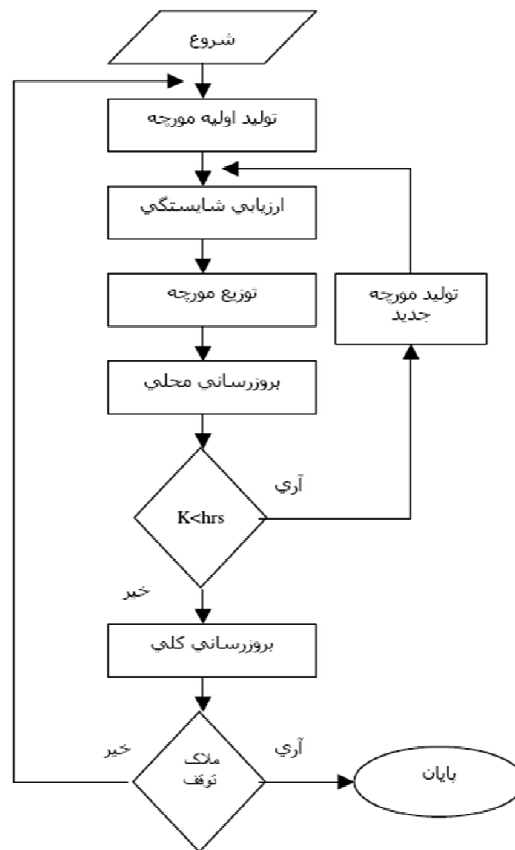
time ← time + 1
  add_trail (t)
  send_ants (t)
  evaluate (t)
  evaporate (t)

}
end

```

1-1-4 فلوجارت کلی الگوریتم

- 1 - مقدار دهی اولیه تولید مورچه: در این مرحله کلونی مورچه تولید می شود. مورچه ها در حالت اولیه قرار می گیرند و فرومون اولیه مقدار دهی می شود.
- 2 - ارزیابی شایستگی : در این مرحله سازگاری کلیه مورچه ها بر پایه تابع هدف ارزیابی می شود. با ارزیابی نظیر به نظیر مورچه ها، فرومون به مسیر خاص شامل مورچه ها اضافه می شود.
- 3- توزیع مورچه : در این مرحله مورچه ها بر اساس سطح فرومون و میزان مسافت توزیع می شوند.
- 4 - معیار تمام تکرار : فرآیند تا رسیدن به حداکثر تعداد مورچه یا عدم بهبود جواب ادامه می یابد. کلیه مسیرهای عبوری توسط ها مورچه باید در هر بار تکرار ارزیابی شوند و در صورت پیداشدن یک مسیر بهتر باید آن مسیر ذخیره گردد. بهترین مسیر انتخاب شده بین همه تکرار ها به عنوان بهترین مسیر انتخاب می شود. در شکل شماره ی 1-1 این فلوجارت ارائه گشته است:



شکل شماره 1-1