دانشگاه شهید باهنر کرمان

دانشکده فنی و مهندسی

گروه مهندسی برق

گزارش فنی

الگوریتم جستجوی جمعیت مورچگان

عصمت راشدی

دکترحسین نظام آبادی پور

بهار ۱۳۸۲

فهرست

۱- مقدمه

۲- هوش جمعی

۳- رفتار مورچه ها در طبیعت

۴- ACO

۵- شکل کلی مسائل قابل حل با ACO

۶- دستورالعمل ساده ACO

۷- قوانین حرکت بین گرهها در ACO

۸- قوانین به روز کردن ردپای مسیرها

۹- پارامترهای ACO

۱۰- انواع مسائل قابل حل با ACO

۱۱- مراجع

۱- مقدمه

امروزه با بزرگ شدن مسائل و اهمیت یافتن سرعت رسیدن به پاسخ، دیگر روشهای کلاسیک جوابگو نیست و بیشتر از الگوریتمهای جستجوی تصادفی به جای جستجوی همه جانبه [۱] فضای مسئله، استفاده می‌شود. خصوصا از الگوریتمهای جستجوی ابتکاری[۲] (شهودی) که در سالهای اخیر رشد چشمگیری داشته‌اند.

الگوریتمهای جستجوی ابتکاری، روشهایی هستند که با الهام از فرایندهای فیزیکی، بیولوژیکی و ... به وجود آمده اند. نمونه های این الگوریتم‌ها به قرار زیر است:

- الگوریتمهای تکاملی[۳] با الهام از علم وراثت و تکامل
- پخت شبیه سازی شده[۴] (SA) با الهام از مشاهدات ترمودینامیک
- TB [۵] بر پایه پاسخ حافظه[۶]
- شبکه های عصبی[۷] (NN) بر پایه عملکرد مغز
- سیستمهای فازی بر پایه قواعد زبانی انسان[۸]

اخیرا محققان سعی کرده اند با مطالعه رفتار حشرات، از آنها در طراحی الگوریتمهای ابتکاری بهره بگیرند. این مطالعات منجر به طراحی جالب‌ترین الگوریتم ابتکاری، الگوریتم بهینه سازی به کمک کولونی مورچه ها[۹](ACO) با نام دیگر الگوریتم جستجوی کولونی مورچه ها[۱۰] (ACSA) شد که با الهام از رفتار مورچه ها طراحی شده است و موضوع بحث این گزارش می‌باشد.

---

[1] - Exhustive search
[2]- Heuristic.
[3] - Evolutionary algorithms
[4] - Simulated anealing
[5] - Tabu search
[6] - Memory response
[7] - Neural network
[8] - Linguistic rules
[9] - Ant Colony Optimization
[10] - Ant Colony Search Algorithm

۲- هوش جمعی

واژه هوش جمعی[1] در ابتدا در مورد سیستمهای رباتیکی سلولی[2]، برای توضیح خود سازمانده بودن این سیستمها به کار رفت. در واقع هوشمندی توده‌ای یکی از انواع جالب هوشمندی است و در مواردی مطرح می‌شود که جمعیتی از اعضاء، اعمال ساده ای انجام می‌دهند ولی در نهایت تمام گروه یک مسئله پیچیده را به طور دقیق حل می‌کند. نمونه بارز این نوع هوشمندی در رفتار حشراتی دیده می‌شود که به صورت کولونی زندگی می‌کنند، مثل مورچه ها، زنبورها، موریانه ها و انواعی از سوسکها.

مشاهده همین موجودات بود که دانشمندان را به سمت ساختن ماشینها، یا طراحی الگوریتم‌هایی از این نوع هدایت کرد. آنها با بررسی رفتار این حشرات، مشاهده کردند که با وجود ساده و کم هوش بودن هر حشره، مجموعه آنها یک کار نسبتا پیچیده را انجام می‌دهد. همچنین دریافتند عامل اصلی ایجاد هوش جمعی در کولونی‌ها، خود سازمانده[3] بودن آنهاست. خود سازمانده بودن، باعث می‌شود یک جمعیت، با وجود داشتن اجزاء بسیار ساده، قادر به انجام یک وظیفه پیچیده باشد. مثلا خانه منظم، دقیق و سلولی شکل زنبورها را در نظر بگیرید. هر کدام از زنبورها، فقط وظیفه ساختن سلولهای کوچک خانه را بر عهده دارد، خانه منظم و هندسی شکل خودش به وجود می‌آید. چرا که، شکل هر سلول فعلی، ساختمان سلولهای جدید کناری را تعیین می‌کند. یعنی خودسازماندهی، هوش جمعی را باعث می‌شود.

مثال دیگر می‌تواند توده موریانه ها، با هدف ساختن خانه باشد. در جریان خانه سازی، در ابتدا صدها موریانه به صورت تصادفی در محیط حرکت می‌کنند. هر موریانه به محض رسیدن به فضایی که کمی بالاتر از سطح زمین قرار دارد، خاک را با بزاغ خود آغشته کرده، گلوله های کوچک خاکی درست می‌کند (رفتار اول). بعد از مدتی محیط از تپه های کوچک پر می‌شود و هر موریانه، به محض رسیدن به یک تپه کوچک، با انرژی بسیار بالایی، تپه ها را به کمک خاک و بزاق خود به ستون تبدیل می‌کند (رفتار دوم) و وقتی ارتفاع هر ستون به حد خاصی رسید، سراغ ستون بعدی می‌رود تا تعداد ستونهای نزدیک به هم، قابل توجه شود. در این وقت، رفتار سومی از خود بروز می‌دهد که وصل کردن ستونها به یکدیگر است. این رفتارها، تا ساختن خانه کامل ادامه دارد. یعنی هر موریانه، مثل یک سیستم ساده، تنها سه رفتار از خود بروز می‌دهد که بروز هر کدام، بستگی به اطلاعات محیطی دارد. اما در انتها یک

[1] - Swarm intelligence
[2] - Cellular robotic systems
[3] - Self organization

خانه دقیق ساخته می‌شود که بدون داشتن نقشه قبلی و یا وجود یک مغز مرکزی، تنها با هماهنگی بین رفتارهای اعضاء، به وجود آمده است.

در مورد بالا دیده شد که خودسازماندهی یک نتیجه عمومی دارد در حالی که از اطلاعات محلی[1] بهره می‌گیرد. خود سازماندهی در رسیدن به این هدف، مدیون عوامل زیر است[۱].

- جستجوی تصادفی در فضا و ترقی و تنزل[2]
- بازخورد مثبت[3]
- بازخورد منفی[4]
- وجود عملکردهای متنوع موثر بر هم[5]

برای توضیح فرض کنید جمعیتی از اجزاء ساده در فضای مسئله دنبال جواب بهینه می گردند. این اجزاء فضا را به صورت تصادفی جستجو می‌کنند که معمولا با جابجایی بین جوابهای با درجات متفاوت مناسب بودن (ترقی و تنزل) همراه است. در این میان مجموعه عواملی وجود دارد که باعث تقویت جوابهای مناسبتر می‌شود و جمعیت را به سمت جوابهای هر چه بهتر می‌کشاند (بازخورد مثبت) و مجموعه محدودیتهایی نیز هستند که باعث می‌شوند جوابهای دیگر هم تا حدی بررسی شود (بازخورد منفی).

مورد جالب‌تری که در کولونیها دیده می‌شود، تأثیرپذیری اعضاء جمعیت از یکدیگر است. یعنی رفتار یک عضو تا حدی تأثیر گرفته از رفتارهای قبلی اعضاء است و در عین حال بر رفتارهای بعدی اعضاء اثر می‌گذارد. در واقع این عملکردهای موثر بر هم، که بسیار متنوع نیز هستند، در جهت خود سازماندهی جمعیت بسیار کمک کننده‌اند. این تأثیرات می‌تواند به دو صورت مستقیم یا غیر مستقیم باشد. در تأثیر پذیری مستقیم، هماهنگی بین اعضاء از طریق ارتباطات مستقیم دیداری، فیزیکی یا شیمیایی است و در تأثیر پذیری غیر مستقیم، از طریق رفتارهای stigmegy صورت می‌گیرد. یعنی رفتارهایی که طی آن، اعضاء از طریق اثر گذاری بر محیط با هم ارتباط داشته و با یکدیگر هماهنگ می شوند. مثل نحوه ارتباط موریانه‌ها در جریان ساخت خانه.

مجموعه عوامل بالا، در رفتار مورچه ها هنگام پیدا کردن غذا مشخصا دیده می‌شود که در بخش سوم آورده شده است.

---

[1] - Local information
[2] - Fluctuation
[3] - Posetive feedback
[4] - Negative feedback
[5] - Multiple interactions

۳- رفتار مورچه ها در طبیعت

یک نمونه جالب از هوشمندی توده ای، در کولونی مورچه ها دیده می شود. مورچه ها موجوداتی کور، بی‌حافظه و بسیار کم هوشند. اما در جریان جستجوی غذا[1]، دیده می‌شود که به سرعت کوتاهترین مسیر رفت و برگشت بین خانه تا غذا را پیدا می‌کنند. یعنی حل یک مسئله نسبتا پیچیده بدون داشتن یک مغز مرکزی که توده را هدایت کند.

هر مورچه هنگام جستجوی غذا تنها دو رفتار ساده به صورت زیر از خود نشان می‌دهد:

• هر مورچه هنگام گذشتن از هر مسیر مقداری فرمون از خود به جا می‌گذارد که حکم ردپای مورچه را دارد. این رد پا به ردپای فرمونی[2] مشهور است.

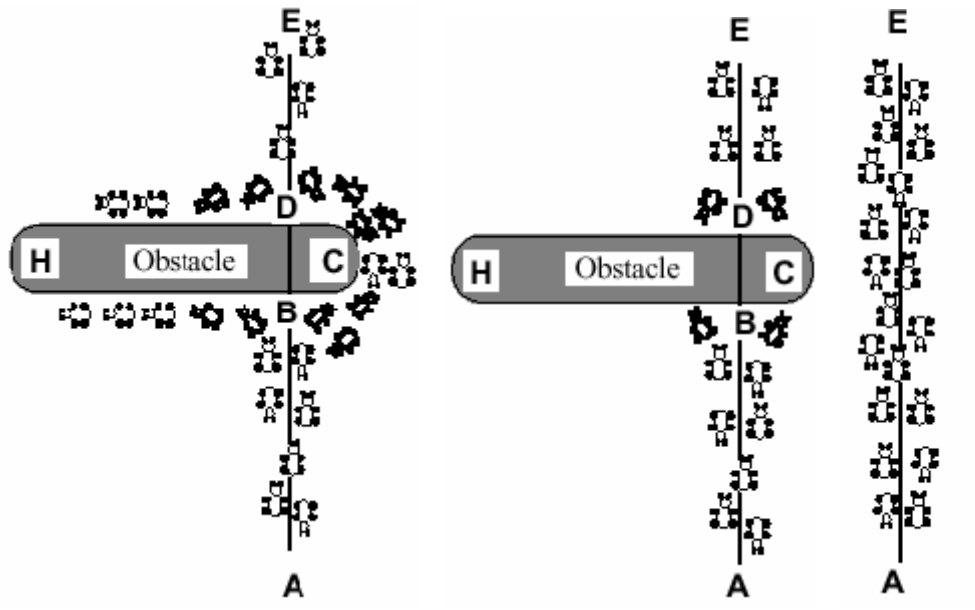
• هر مورچه مسیر جستجویش را به طور تصادفی انتخاب می‌کند که در این میان احتمال انتخاب مسیرهای دارای فرمون بیشتر، زیادتر است. یعنی هر مورچه ترجیح می دهد از مسیری عبور کند که قبلا مورچه های بیشتری از آن گذشته باشند.

همانطور که از مشخصات هوشمندی توده‌ای است، مورچه ها با همین دو رفتار ساده طوری سازمان می‌یابند که پس از مدتی به کوتاهترین مسیر رفت و برگشت بین خانه و غذا همگرا می‌شوند. در مثال زیر نشان داده می شود که چگونه این دو رفتار ساده، مورچه ها را به مسیر کوتاهتر همگرا می‌کند.

در شکل ۱ مورچه ها مسیر A تا E را بین خانه تا غذا طی می‌کرده اند. ناگهان مانعی مثل CH سر راه آنها قرار می‌گیرد. هر مورچه ای که در مسیر رفت و برگشت به مانع می‌رسد دو راه برای انتخاب دارد که چون هر دو خالی از فرمون هستند احتمال انتخاب مساوی دارند. اما مورچه‌هایی که مسیر BCD را انتخاب کنند زودتر به طرف مقابل می‌رسند. بنابراین مورچه هایی که در جهت عکس آنها حرکت می‌کنند، این مسیر را نسبت به راه دوم دارای فرمون بیشتری می‌بینند و احتمال انتخاب آن بیشتر شده، مورچه های بیشتری در آن مسیر قرار می‌گیرند. تا اینکه پس از مدتی تقریبا همه مورچه ها این مسیر را انتخاب می‌کنند. یعنی به مسیر کوتاهتر همگرا می‌شوند.

---

[1] - Forage for food
[2] - Pheromone trail



ج ب الف

شکل۱- رفتار مورچه های واقعی[۲]. الف) مورچه‌ها مسیر رفت و برگشت AE را طی می‌کنند. ب) مانعی مثل CH سر راه آنها قرار می‌گیرد و هر مورچه دو راه برای انتخاب دارد. ج) میزان فرمون موجود در مسیر کوتاهتر بالا می‌رود.

در بخش دوم، عوامل منجر به خود سازماندهی و در نهایت هوش جمعی در کولونی ها توضیح داده شد. حالا چگونگی این عوامل در رفتار مورچه ها بررسی می‌شود.

الف- جستجوی تصادفی در فضا: مورچه ها در انتخاب مسیر حرکات کاملا تصادفی دارند یعنی اگرچه احتمال انتخاب مسیرهای غنی از فرمون بیشتر است اما این احتمال وجود دارد که یک مورچه، مسیر دیگری را برای رفتن انتخاب کند یا به اصطلاح گم شود. یعنی همیشه امکان پیدا کردن جوابهای بهتر (مسیرهای کوتاهتر از مسیر فعلی) وجود دارد.

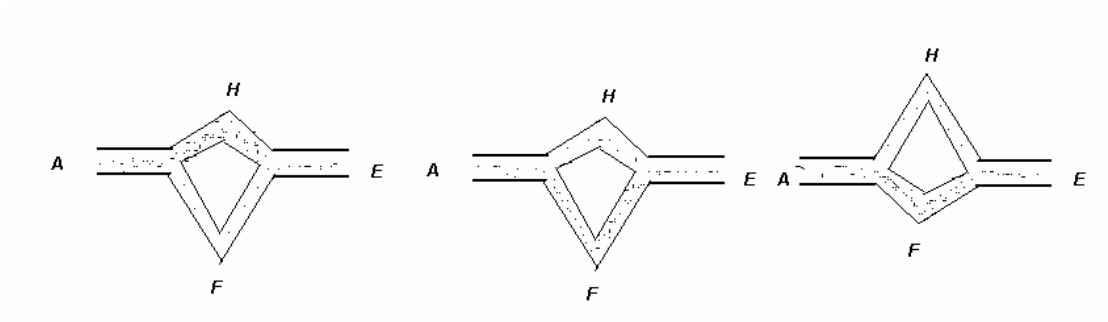
ب- بازخورد مثبت: مورچه ها مسیری را که قبلا مورچه های بیشتری از آن گذشته اند، بیشتر انتخاب می‌کنند و مسیرهایی که بیشتر انتخاب شوند در مراحل بعد جذابیت بیشتری برای طی شدن دارند. یعنی وجود یک بازخورد مثبت که همانطور که نشان داده شد، جوابهای مناسبتر را تقویت می‌کرد.

ج- بازخورد منفی: در مورد طی مسیر مورچه ها، یک محدودیت محیطی به نام تبخیر[1] ردپا ها وجود دارد به این صورت که هر رد پایی که از مورچه ها به جا می ماند، بعد از مدتی در محیط شروع به تبخیر شدن می‌کند و چون

[1] - Evaporation

مورچه ها، فضا را به صورت تصادفی جستجو می‌کنند، اگر جواب بهتری پیدا شد، تبخیر کمک می‌کند که جواب جدید هم بتواند خودش را نشان بدهد. یعنی تبخیر مثل یک بازخورد منفی مسیر فعلی را تضعیف می‌کند تا جواب جدید هم بتواند به اندازه کافی تقویت شود. این پدیده به مورچه ها در جهت اینکه خود را با شرایط محیطی وفق دهند، بسیار کمک می‌کند. مثلا شکل ۲ را در نظر بگیرید. در ابتدا مورچه ها مسیر AHE را که مسیر کوتاهتر بوده، تا غذا طی می‌کرده‌اند. اما در اثر یک تغییر محیطی، مسیر بهینه به AFE تغییر می‌کند. در این زمان هنوز تراکم مورچه ها در AHE است اما احتمال این وجود دارد که تعدادی مورچه به طور تصادفی در AFE قرار بگیرند و چون این مسیر کوتاهتر است به مرور، بازخورد مثبت، مسیر AFE را تقویت کرده و بازخورد منفی مسیر AHE را تضعیف می‌کند.

به فرایند همگرا شدن مورچه ها به یک مسیر حالت رکود[1] گفته می‌شود. تبخیر از ایجاد رکود در مراحل اولیه جلوگیری می‌کند. یعنی اجازه نمی‌دهد همواره اولین جوابها به عنوان بهترین جواب انتخاب شوند.



ج ب الف

شکل۲- اثر تبخیر. الف) مورچه ها مسیر رفت و برگشت AHE را طی می‌کنند. ب) مسیر بهینه عوض می‌شود. ج) مورچه ها مسیر بهینه جدید را پیدا می‌کنند.

د- وجود عملکردهای گوناگون موثر بر هم: مورچه ها موجوداتی کور هستند و با یکدیگر ارتباط غیر مستقیم دارند. ارتباط مورچه‌ها از طریق اثر گذاری بر محیط (رفتار stigmergy) است و از طریق ردپای فرمونی صورت می‌گیرد. مورچه ها به کمک این رد پا گذاری اطلاعات مسیر را به یکدیگر منتقل می‌کنند و به این ترتیب با هم هماهنگ شده، در جهت پیدا کردن مسیر بهینه سازماندهی می‌شوند.

[1] - Stagnation

نکته جالب دیگر در رفتار مورچه ها این است که انتخاب یک مسیر در یک زمان باعث می‌شود احتمال انتخاب آن در زمانهای بعدی بیشتر شود. کولونی مورچه ها با بهره گیری از بازخورد مثبت، خود را در جهت جوابهای هر چه بهینه‌تر تقویت می‌کند. گفته می‌شود فرایند انتخاب مسیر بهینه در مورچه ها، یک فرایند خود دگرگون[1] [2] است. به فرایندهایی که تصمیم اتخاذ شده در زمان t_0 احتمال اتخاذ همان تصمیم در زمان $t>t_0$ را افزایش دهد، فرایندهای خود دگرگون می‌گویند. در این نوع فرایندها از بازخورد مثبت بهره گرفته می‌شود.

۴- الگوریتم ACO

الگوریتم بهینه سازی به روش کولونی مورچه ها (ACO)، از دسته الگوریتمهای جستجوی فرا ابتکاری[2] می‌باشد که با الهام از رفتار مورچه ها در طبیعت ایجاد شده است. چرا که آنها قادرند با وجود کور بودن، کوتاهترین مسیر رفت و برگشت از خانه تا غذا را پیدا کنند. دانشمندان دریافتند این قابلیت نتیجه ردپای فرمونی است که مورچه ها از آن برای برقراری ارتباط با یکدیگر و جابجایی اطلاعات مسیر استفاده می‌کنند.

در الگوریتم ACO رفتار مورچه های طبیعت، در مورچه های مصنوعی[3] شبیه سازی شده است. ضمن اینکه بعضی قابلیتهای اضافی مثل حافظه نیز به آنها داده شده است. مهندسان از این مورچه های مصنوعی برای حل مسائل مختلف بهینه سازی که عموما گسسته نیز هستند، استفاده می‌کنند.

در ابتدا، آقای مارکو دوریگو[4] بود که بحث الگوریتم مورچه[5] را در دنیای مهندسی باز کرد. از آن زمان، از واژه الگوریتم مورچه برای توضیح هر الگوریتمی که خط مشی ACO را دنبال می‌کند، اما لزوما تمام جوانب فرا ابتکاری آن را دنبال نمی‌کند، استفاده می‌شود.

نسخه اولیه ACO در سال ۱۹۹۱ توسط دوریگو، تحت عنوان سیستم مورچه[6] معرفی شد[۲]. بعد از آن نسخه-های تصحیح شده بیشتری توسط افراد مختلف ارائه شد (Dorigo, Maniezzo& colorni, 1996; Stutzle& hoos,1997; Bullnheimer, Harl& Strauss, 1999; Dorigo, Bonabeau& Theraulaz, 2000) [۷].

---

[1] - Autocatalytic
[2] - Meta heuristic.
[3] - Artifitial ants
[4] - Marco Dorigo
[5] - Ant algorithm
[6] - Ant system

از محسناتی که توجه محققان را به سوی این الگوریتم جلب کرد، داشتن بازخورد مثبت، محاسبات توزیع شده[1] و هیوریستیک آزمند سازنده[2] می‌باشد[۴].

بازخورد مثبت، منجر به کشف سریع جوابهای خوب می شود. محاسبات توزیع شده از همگرایی زود و بی‌موقع جلوگیری می کند و هیوریستیک آزمند سازنده به کشف جوابهای قابل قبول در مراحل اولیه جستجو کمک می‌کند.

ویژگیهای خاص ACO، به آن کمک می‌کند که اگوریتمی ماهر[3]، نیرومند[4] و قابل کنترل باشد. ماهر است، به این دلیل که انواع مسائل بهینه سازی متقارن (مثل TSP) و نا متقارن (مثل ATSP) را حل می‌کند و نیرومند است چون با تغییر کوچکی می‌تواند مسائل متفاوت‌تری مثل QAP یا JSCP را نیز حل کند. همچنین محاسبه جمعی و توزیع شده، آن را به یک الگوریتم جستجوی موازی قابل کنترل تبدیل کرده است که می‌تواند به سمت همگرا شدن به جوابهای بهتر کنترل شود.

از معایب ACO می‌توان به داشتن پارامترهای زیاد و حساسیت الگوریتم به آنها اشاره کرد. ضمن اینکه هیچ تضمینی در مورد همگرایی الگوریتم به جواب بهینه وجود ندارد.

در ادامه، در بخش ۵، شکل کلی مسائل قابل حل با ACO و در بخش ۶، روند طراحی الگوریتم ACO آورده شده است.

۵- شکل کلی مسائل قابل حل با ACO [۳] و [۱۸].

برخلاف مورچه های طبیعت که مسئله خود را در محیطی پیوسته حل می کنند، از الگوریتم جستجوی ACO ، غالبا برای حل مسائل بهینه سازی گسسته[5] استفاده می‌شود. گسسته به این معنی، که هر جواب مسئله از اجزاء خاصی تشکیل شده است.

برای توضیح کلی اینگونه مسائل می توان مسئله کمینه سازی S,F,Ω را در نظر گرفت که در آن S مجموعه جوابهای مسئله، $F(s,t)$ تابع هزینه که باید کمینه شود ($s \subset S$) و Ω قیود مسئله می‌باشند.

الگوریتم جستجو باید در فضای مسئله (مجموعه جوابهای S) دنبال جواب مناسبی بگردد که به ازاء آن تابع هزینه کمینه شود. ضمن اینکه ممکن است محدودیتهایی مثل Ω نیز به مسئله اعمال شده باشد که با وجود آنها،

[1] - Distributed computation
[2] - Constructive Greedy Heuristic
[3] - Versatile
[4] - Robust
[5] - Discrite Optimization

بعضی جوابهای مسئله غیر عملی می شود. مجموعه جوابهای عملی که جوابهای مسئله را برآورده کند با S^* نشان داده شده است که طبعا $S \subset S^*$. مسئله گسسته شامل اجزاء خاصی است، این اجزاء در مجموعه C بیان می‌شوند.

$$C=\{c_1, c_2, \ldots, c_{CN}\} \quad (۱)$$

وضعیت مسئله در هر لحظه با X نشان داده می‌شود که مجموعه ای از اجزاء مسئله است.

$$X=\langle c_i, c_j, \ldots, c_k \rangle \quad (۲)$$

مجموعه تمام وضعیتهای ممکن با X نشان داده می‌شود، ضمن اینکه مجموعه محدودیتهای Ω، مجموعه وضعیتهای $\tilde{X}$ را تعریف می کند و $X \subset \tilde{X}$. همچنین هر جواب عملی مسئله، یک وضعیت ممکن از مسئله است یعنی $\tilde{X} \subset S^*$.

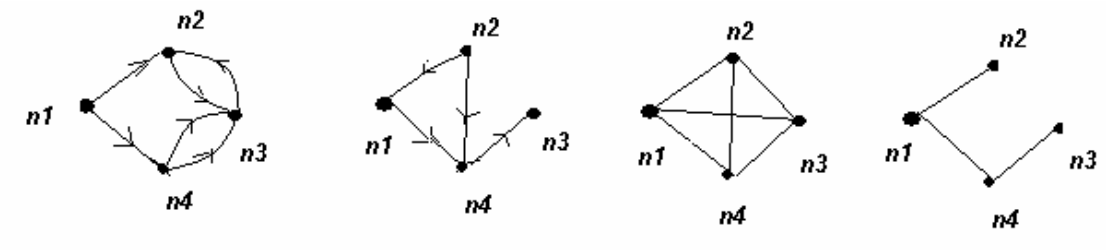
۶- دستورالعمل ساده ACO

برای حل مسائل مختلف بهینه سازی (مطابق قسمت قبل) توسط الگوریتمهای مختلف مورچه، می‌توان دستورالعمل ساده زیر را پیشنهاد کرد:

۱. تبدیل مسئله به پارامترهای عددی: در الگوریتم مورچه، معمولاً مسائل به صورت گراف $G<N,L>$ در نظر گرفته می‌شوند که هر گره گراف یک جزء مسئله است. یعنی مجموعه اجزاء مسئله (C) توسط مجموعه گره های گراف نمایش داده می‌شوند.

$$N=\{n_1, n_2, \ldots, n_{CN}\} \quad (۳)$$

این گره‌ها طبق روابط بین اجزاء مسئله، توسط یالهایی به هم متصل می‌شوند. پس به خاطر روابط متنوع گرافهای مختلفی می‌تواند وجود داشته باشد. مثل انواع گرافهای غیر جهتی (روابط دو طرفه)، گرافهای جهتی (روابط یکطرفه)، گرافهای تمام اتصال[1] یا حتی گرافهای جهتی که یال رفت و برگشت متفاوت دارند (مطابق شکل۳).



الف ب ج د

[1] - Fully connected

شکل۳- انواع گرافها. الف: گراف غیر جهتی . ب:گراف غیر جهتی تمام اتصال. ج:گراف جهتی. د:گراف جهتی نامتقارن

در این گرافها اگر اتصال بین دو گره n_i وn_j با یال L_{ij} نشان داده شود، هر یال L_{ij} دارای دو مشخصه τ_{ij} و η_{ij} می باشد که τ_{ij} میزان ردپای موجود در یال و η_{ij} مقدار بینایی آن است. این مشخصات در گرافهای جهتی نا متقارن، در یال رفت و برگشت، از هم متفاوت می‌باشند.

معمولا در همه یالهای گراف، یک مقدار کوچک ردپا به عنوان پیش فرض در نظر گرفته می‌شود که مقدار آن در طول حل مسئله تغییر می‌کند. الگوریتم به گونه‌ای هدایت می‌شود که در یالهایی که منجر به جوابهای بهتر می-شوند، ردپای بیشتری قرار بگیرد. η_{ij} در هر مسئله به طور شهودی تعیین می‌شود که در غالب موارد، تابع F (تابع هزینه) مقدار آن را پیشنهاد می کند. به η_{ij} بینایی[1] مسئله گفته می‌شود.

۲. در نظر گرفتن چند مورچه مصنوعی: در ACO به منظور شبیه سازی رفتار مورچه های واقعی، یک کولونی از مورچه های مصنوعی شامل m مورچه در نظر می گیرند که مثل مورچه های واقعی دو رفتار از خود نشان می‌دهند:

الف) می‌توانند به یالهایی که از آن گذشته اند، ردپا اضافه کنند.

ب) در انتخاب مسیر، مسیرهای دارای ردپای بیشتر را ترجیح می‌دهند.

این مورچه ها در سه مورد با مورچه های طبیعت متفاوتند:

الف) کاملاً کور نیستند، یعنی قادر به تشخیص η (بینایی مسئله) می‌باشند.

ب) حافظه کوچکی دارند که آنها را قادر می‌سازد اطلاعات مربوط به مسیر خود (مثل گره‌هایی که به ترتیب از آنها گذشته‌اند) را تا زمان کوتاهی نگه دارند. از این اطلاعات برای ارزیابی سفر مورچه به منظور ردپاگذاری مسیرها، استفاده می‌شود. حافظه مورچه K با M_k نشان داده می‌شود.

ج) در محیطی با زمان گسسته زندگی می‌کنند. به این معنی که آنها را در طول مسیر نمی‌بینیم، بلکه در هر لحظه از زمان در یک گره هستند و به عبارتی سرعتی برابر یک گره در واحد زمان دارند.

۳. جایابی اولیه برای مورچه ها: برای شروع حل مسئله، مورچه ها وارد گراف می‌شوند و هر مورچه را روی یکی از گرهها می‌نشیند. برای توزیع m روی n گره معمولا از توزیع تصادفی استفاده می‌شود. بعد از این، در هر لحظه از

[1] - Visibility

زمان[1]، هر مورچه روی یک گره قرار دارد، به طوری که اگر تعداد مورچه های روی گره i در زمان t با $b_i(t)$ نشان داده شود در آن صورت رابطه ۴ برقرار است.

$$m=\sum b_i(t) \qquad (۴)$$

یعنی تا آخرین مرحله حل مسئله، مورچه ای از دست نرفته و مورچه ای نیز اضافه نمی‌شود.

۴. حرکت بین گرهها و جستجوی گراف توسط مورچه ها: مورچه ها باید گراف را که نماینده مسئله است به منظور پیدا کردن جواب بهینه جستجو کنند. هر گره در گراف، جزئی از مسئله و هر جواب مسئله، مجموعه ای از این اجزاء است. پس مورچه ها باید برای ساختن جواب، بین گره‌ها حرکت کنند.

ACO بر اصل تکرار[2] استوار است. به این صورت که با تکرار حرکات کوچک، سفر مورچه و با انجام سفر هر m مورچه یک سیکل و با تکرار چند سیکل، الگوریتم ACO انجام می‌شود. توضیح جامعتر این است که اگر به جابجایی مورچه از گره i به j یک قدم[3] یا یک حرکت همیلتونی گفته شود، با تکرار چند حرکت پیاپی طبق قوانین حرکت، سفر[4] مورچه کامل می‌شود. هر سفر مورچه یک جواب از مسئله است. با اتمام سفر هر m مورچه، یک سیکل[5] پایان می‌یابد. در این زمان سفر مورچه‌ها به کمک حافظه‌شان ارزیابی شده، یک مرحله به روز کردن ردپا، طبق قوانین به روز رسانی ردپا، انجام می‌شود. سپس سیکل بعد با ردپای جدید انجام می‌شود. این سیکلها را تا پیش آمدن شرایط توقف و رسیدن به جواب شبه بهینه تکرار می‌شوند.

دو نکته اساسی در اگوریتم ACO که مسئول هدایت آن به سمت جواب بهینه است: الف) قوانین احتمالی حرکت[6] و ب) قوانین به روز کردن ردپا در مسیرها[7] می‌باشند. قوانین حرکت شامل توابع احتمالاتی است که طبق آنها مورچه ها روی یالهای گراف، حرکات تصادفی می‌کنند. یعنی اگر در یک لحظه از زمان روی یک گره خاص هستند به کمک این توابع احتمالاتی گره بعد را برای رفتن انتخاب می‌کنند. قوانین به روز کردن ردپا، مجموعه توابعی هستند که مطابق آنها در یک زمان خاص، مثل بعد از هر یک حرکت همیلتونی یا بعد از سفر هر مورچه یا بعد از هر سیکل[8]، ردپای مسیرها برای حرکات بعدی عوض می‌شود.

---

[1] - زمان به صورت گسسته در نظر گرفته می شود.

[2] - Iteration.
[3] - Step.
[4] - Tour.
[5] - Cycle.
[6] - Probability state transition rules.
[7] - Pheromone update rules.

[8] - بسته به نوع الگوریتم

نحوه انجام این دو گزینه، از این جهت که روی جواب نهایی از نظر قابل اطمینان بودن اثر می‌گذارند، بسیار مهم است. در واقع نسخه های اولیه ACO با تکمیل همین قوانین رو به پیشرفت گذاشت. که توضیح آنها در بخشهای جداگانه آورده شده است.

۵. ملاک توقف[1]: تکرار سیکلها باید تا پیش آمدن شرایط توقف ادامه داشته باشد. بعد از آن الگوریتم متوقف می-شود. ضمن اینکه مایلیم الگوریتم زمانی متوقف شود که ما را به جواب شبه بهینه رسانده باشد. به این منظور می توان دو شرط برای توقف الگوریتم در نظر گرفت. اول اینکه الگوریتم بعد از تکرارهای مشخص که احتمال می‌رود جواب مناسبی پیشنهاد کند، متوقف شود. حالت دیگر زمانی است که وضعیت رکود [2] پیش آمده باشد. یعنی زمانی که همه مورچه ها به یک مسیر همگرا شده باشند، چون بعد از این مرحله الگوریتم پیشرفتی نخواهد داشت. در واقع در نسخه های مختلف ACO نیز سعی شده از راکد شدن الگوریتم در مراحل اولیه (سیکلهای ابتدایی) جلوگیری شود. چرا که در این مراحل هنوز دامنه جستجو وسعت چندانی پیدا نکرده، جواب قابل اطمینانی به دست نیامده است.

۷- قوانین حرکت بین گرهها در ACO

این قوانین توابعی هستند که طبق آنها هر مورچه گره بعد را برای رفتن انتخاب می‌کند. همه توابعی که در این مورد در الگوریتمهای متفاوت ACO تعریف شده‌اند از نوع توابع احتمالاتی می‌باشند. یعنی اگرچه در آنها بعضی عوامل در انتخاب مورچه اثر می‌گذارند ولی به طور کلی حرکات به صورت تصادفی انجام می‌شود.

در سیستم مورچه یا AS[3] [۲] این تابع به صورت رابطه ۵ تعریف شده است.

$$p_{ij}^{k}(t)=\begin{cases}\dfrac{\tau_{ij}^{\alpha}\times\eta_{ij}^{\beta}}{\sum\limits_{k\subset admissible\,nodes}\tau_{ik}^{\alpha}\times\eta_{ik}^{\beta}} & if \quad j\subset admissible\,nodes\\ 0 & otherwise\end{cases} \qquad (۵)$$

[1] - Stopping criterion.
[2] - Stagnation.

[3] - Ant System ، نسخه اولیه ACO ،Dorigo، ۱۹۹۲.

تابع P^k_{ij} احتمال بودن مورچه k در گره j در زمان t+1 می باشد، اگر در زمان t در گره i باشد. τ_{ij} نیز ردپای موجود در یال L_{ij} در زمان t و η_{ij} مقدار بینایی[1] آن (هیوریستیک مسئله) می باشد. مقادیر α و β نیز پارامترهای مسئله اند که میزان اهمیت ردپا در مقابل بینایی مسئله را کنترل می کنند.

در تابع P ، برای هر یالی که مورچه مجز به طی کردن آن است، یک مقدار $\eta_{ij}^{\beta} \times \tau_{ij}^{\alpha}$ در نظر گرفته شده، با تقسیم به حاصل‌جمع همه آنها نرمالیزه می‌شود و یک احتمال برای طی کردن هر یال به مورچه می‌دهد. سپس مورچه با یک حرکت تصادفی یکی از این یالها را طی می‌کند. در این میان احتمال انتخاب یالهای دارای ردپای بیشتر و مقدار هیوریستیک بهتر، بیشتر است. چرا که تابع فوق مقدار بیشتری به آنها نسبت می‌دهد.

در تابع P احتمال انتخاب گره‌های غیر مجاز صفر است. گره‌های غیر مجاز در زمان t گره‌هایی هستند که به خاطر بعضی محدودیتهای مسئله، مورچه حق ندارد در زمان t+1 در آن گره‌ها قرار بگیرد.

تابع دیگری که بعد از P مطرح شد، تابع S است که در الگوریتم ACS[2] [۴] به صورت رابطه ۶ آورده شده است:

$$S_{ij}^{K} = \begin{cases} \underset{j \subset admissible \quad nodes}{\arg\max} \quad [\tau_{ij}^{\alpha} \times \eta_{ij}^{\beta}] & if\ rand < q_0 \\ p_{ij}^{k}(t) & \text{otherwise} \end{cases} \qquad (۶)$$

در این تابع، طبق یک احتمال، یالی که بهترین شرایط را داشته باشد، انتخاب می‌شود. یعنی یالی که دارای بیشترین مقدار فرمون و بهترین مقدار بینایی است. در غیر اینصورت مطابق تابع احتمال P عمل می‌شود.

در تابع S ، با یک احتمال، یک حرکت اجباری انجام می‌شود به همین دلیل به آن تابع احتمال تصادفی کاذب[3] گفته می شود. در این تابع q_0 به پارامترهای مسئله اضافه می‌شود که $0 \le q_0 \le 1$.

[1] - Visibility.
[2]- Ant Colony System.
[3] -Pseudo random proportion rules.

۸- قوانین به روز کردن ردپای مسیرها

به روز کردن ردپا شامل دو مرحله تبخیر[1] و اضافه کردن ردپای مورچه ها است. یعنی در یک زمان مشخص، مقدار کوچکی از ردپای قدیمی موجود در مسیرها تبخیر، سپس ردپای جدید اضافه می‌شود. با اضافه کردن ردپا بازخورد مثبت الگوریتم در جهت گرایش دادن مورچه ها به سمت جوابی شبیه بهترین جواب کشف شده تا آن لحظه، تقویت می‌شود که در نهایت منجر به کشف سریع جوابهای نسبتا خوب می‌شود. البته نباید بازخورد مثبت به قدری قوی شود که همواره اولین جوابها به عنوان جواب نهایی در نظر گرفته شود. یعنی در همان سیکلهای اولیه رکود پیش بیاید. مقدار تبخیر نیز باید در حدی تنظیم شود که همواره درصدی از مورچه ها، جوابهایی غیر از جواب بهینه انتخاب کنند تا در صورت تغییر جواب بهینه، الگوریتم توانایی دنبال کردن[2] جواب جدید را داشته باشد. ضمن اینکه تبخیر با ایجاد بازخورد منفی از حالت رکود جلوگیری می‌کند. اما اگر مقدار آن زیاد شود، اثر بازخورد مثبت را از بین برده، از حرکت الگوریتم به سمت جواب بهینه جلوگیری می‌کند.

برای به روز کردن ردپا، روشهای متفاوتی در الگوریتمهای مختلف مورچه آمده است که عامل اصلی تفاوت آنها با یکدیگر می‌باشد. در زیر نسخه‌های مختلف الگوریتم مورچه برای توضیح نحوه به روز کردن ردپا در هر کدام آورده شده است.

- **سیستم مورچه**

سیستم مورچه یا AS^3 [۲] که نسخه اولیه ACO می‌باشد، شامل سه الگوریتم Ant-cycle، Ant-dencity و Ant-quantity می‌باشد.

در الگوریتم Ant-cycle در انتهای هر سیکل، ابتدا ردپای قبلی هر یال را با ضریب ρ که ضریب تبخیر[۴و۵] نام دارد، تبخیر شده سپس ردپای مورچه‌ها به آن اضافه می‌شود. به روز کردن ردپا در انتهای هر سیکل به صورت رابطه ۷ انجام می‌شود.

$$\tau_{ij}(new) = (1-\rho) \times \tau_{ij}(old) + \Delta\tau_{ij} \qquad (۷)$$

---

[1] - Evaporation.
[2] - Tracking.
[3] - Ant System.
[4] - Evaporation coefficient.
[5] - رابطه گاهی به صورت $\tau_{ij}(new) = \rho \times \tau_{ij}(old) + \Delta\tau_{ij}$ نوشته می‌شود که در آن صورت به ρ ضریب ماندگاری (Coefficient of persistence) گفته می‌شود.

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \quad , \quad \Delta\tau_{ij}^{k} = \begin{cases} Q/F^{k} & if\ the\ Kth\ ant\ traverse\ L_{ij} \\ 0 & otherwise \end{cases} \qquad (۸)$$

$\Delta\tau_{ij}$ اضافه ردپا برای یال L_{ij} می‌باشد و مجموع ردپای تمام مورچه‌هایی است که در سیکل فوق، از این یال عبور کرده‌اند. در این الگوریتم، هر مورچه می‌تواند متناسب با عکس هزینه سفرش، به یالهایی که از آنها گذشته ردپا اضافه کند. F^{k} هزینه سفر مورچه k است. پس مورچه‌هایی که سفر بهتری داشته‌اند، ردپای بیشتری از خود به جا می‌گذارند. Q جزء پارامترهای مسئله است.

در الگوریتم Ant-cycle ، گاهی یک عمل تشویق اضافی برای بهترین مسیر نیز انجام می‌شود که به آن نخبه گرایی[1] می‌گویند و طی آن مورچه های نخبه، یعنی مورچه هایی که سفر بهتری داشته‌اند، یک ردپاگذاری اضافه نیز انجام می‌دهند. مطابق روش فوق رابطه ۷ به شکل رابطه ۹ کامل می‌شود.

$$\tau_{ij}(new) = (1-\rho)\times\tau_{ij}(old) + \Delta\tau_{ij} + \Delta\tau_{ij}^{e}$$

$$\Delta\tau_{ij}^{e} = \begin{cases} e\times Q/F^{e} & if\ L_{ij} \in best\ tour \\ 0 & otherwise \end{cases} \qquad (۹)$$

F^{e} هزینه سفر بهترین مورچه (کمترین هزینه سفر) و e یک پارامتر است. به چنین اعمال اضافی اعمال فوق العاده[2] گفته می‌شود. که در هر الگوریتم شکلی از آن به صورت اختیاری انجام می‌شود. در دو الگوریتم دیگر AS ، مورچه ها بلافاصله بعد از گذشت از هر یال، ردپای خود را در آن یال به جا می‌گذارند. ردپای هر مورچه در الگوریتم Ant-density مطابق رابطه ۱۰و در الگوریتم Ant-quantity به ترتیب مطابق رابطه ۱۱ می‌باشد[۲].

$$\Delta\tau_{ij}^{k} = \begin{cases} Q & if\ the\ K's\ ant\ travers\ L_{ij} \\ 0 & otherwise \end{cases} \qquad (۱۰)$$

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{f_{ij}} & if\ theKs\ ant\ travers\ L_{ij} \\ 0 & oterwise \end{cases} \qquad (۱۱)$$

F^{e} هزینه یال L_{ij}است. از میان این سه الگوریتم، Ant-cycle جوابهای بهتری ارائه می‌کند. البته هر سه آنها نسبت به نسخه‌های بعدی ضعیفترند. چرا که Ant-cycle فقط روی جستجوی عمومی و Ant-quantity وAnt-density فقط روی تکنیکهای جستجوی محلی استوارند. در صورتی که ترکیب آنها، جوابهای مناسبتری

[1] - Elitist strategy
[2] - Daemon

ارائه می‌دهد. در نسخه‌های بعدی سعی شده با افزودن تکنیکهای جستجوی محلی، قابلیتهای Ant-cycle بیشتر شود.

- سیستم کولونی مورچه

در سیستم کولونی مورچه یا ACS[1] [۴] هر دو مرحله به روز رسانی فرامحلی و محلی انجام می‌شود. الگوریتم ACS در شکل ۴ آورده شده است.

در این الگوریتم دو مرحله به روز کردن ردپا انجام می‌شود. در به روز کردن فرامحلی[۲] در پایان هر سیکل، مسیر نخبه به صورت روابط ۱۲ و ۱۳ تشویق می‌شود.

$$\tau_{ij} = (1-\alpha)\times\Delta\tau_{ij} + \alpha\times\Delta\tau_{ij}^{b} \qquad 0<\alpha<1 \tag{۱۲}$$

$$\Delta\tau_{ij}^{b} = \begin{cases} \dfrac{1}{F^{e}} & L_{ij}\in elitist\ \ tour \\ 0 & otherwise \end{cases} \tag{۱۳}$$

F^{e} هزینه سفری است که بین سفرهای انجام شده از همه کم هزینه‌تر باشد. بهترین سفر[۳] می‌تواند از میان سفرهای سیکل فوق[۴] یا از میان تمام سفرها از ابتدای الگوریتم[۵] انتخاب شود. تجربه‌ها نشان داده نتایج فوق، فرق چندانی با هم ندارند.

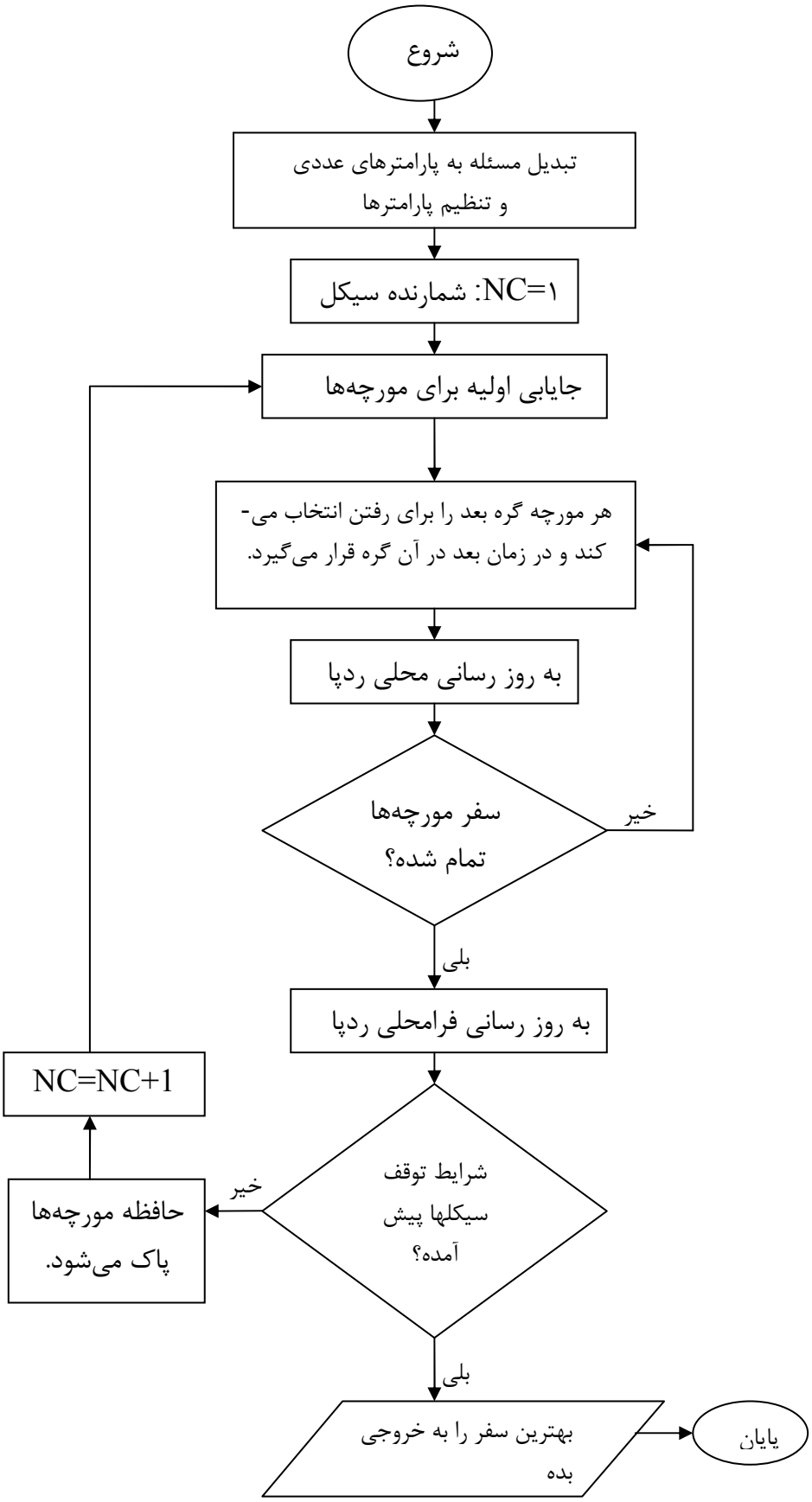
در ACS به روز کردن محلی[۶] بعد از انجام هر قدم، وقتی هر مورچه یک یال را طی کرد، برای یال‌هایی که مورچه‌ها دیده‌اند به صورت روابط ۱۴ و ۱۵ انجام می‌شود.

$$\tau_{ij} = (1-\rho)\times\tau_{ij} + \rho\times\Delta\tau_{ij} \qquad 0<\rho<1 \tag{۱۴}$$

$$\Delta\tau_{ij} = \begin{cases} 0 \\ \tau_{0} \\ \gamma\times\max\tau_{jl} \quad 0<\gamma<1 \end{cases} \tag{۱۵}$$

---

[1] - Ant Colony System.
[2] - Global update.
[3] - Best tour.
[4]- Iteration best.
[5]- Global best.
[6] - Local update.


شروع
تبدیل مسئله به پارامترهای عددی و تنظیم پارامترها
NC=۱: شمارنده سیکل
جایابی اولیه برای مورچه‌ها
هر مورچه گره بعد را برای رفتن انتخاب می‌کند و در زمان بعد در آن گره قرار می‌گیرد.
به روز رسانی محلی ردپا
سفر مورچه‌ها تمام شده؟
خیر
بلی
به روز رسانی فرامحلی ردپا
شرایط توقف سیکلها پیش آمده؟
خیر
حافظه مورچه‌ها پاک می‌شود.
NC=NC+1
بلی
بهترین سفر را به خروجی بده
پایان

در رابطه ۱۵، اگر $\Delta\tau_{ij} = 0$ باشد، یعنی به طور کلی از به روز کردن محلی صرفنظر شده است. اما استفاده از به روز کردن محلی نتایج بهتری به دست می‌دهد که دو گزینه بعد را شامل می‌شود.

به روش به روز کردن محلی مطابق رابطه $\Delta\tau_{ij} = \gamma \times \max \tau_{jl}$ ، روش Ant-Q [۵و۲۶] گفته می‌شود. این روش از الگوریتم یادگیری تقویت شده[1] Q-learning الهام گرفته است و اگر چه روش مناسبی است اما محاسبات الگوریتم را پیچیده می‌کند.

به طور کلی روش به روز کردن محلی منجر به استفاده بهتر از اطلاعات فرمونی می‌گردد و باعث می‌شود انتقال اطلاعات مسیر نه تنها بین سیکلها، بلکه بین مورچه های یک سیکل هم صورت بگیرد. یعنی هر مورچه در یک سیکل به کمک تجربه مورچه هایی که قبل از او مسیری را طی کرده‌اند، مسیرهای مناسبتری انتخاب کرده، فرمون کمتری در مسیرهای نامناسب به هدر می‌دهد. ضمن اینکه سرعت کشف جواب بهینه بالا می‌رود.

- **سیستم مورچه بیشینه-کمینه**

در سیستم مورچه بیشینه-کمینه[2] یا MAX-MIN AS [۶] به روز کردن ردپا مطابق ACS است. با این تفاوت که ردپای مسیرها در یک محدوده خاص کنترل می‌شود. یعنی همیشه رابطه ۱۶ برقرار است.

$$\tau_{\min} \le \tau_{ij} \le \tau_{\max} \qquad (۱۶)$$

مقادیر $\tau_{\max}$ و $\tau_{\min}$ با توجه شرایط مسئله تعیین می‌شود. معمولا $\tau_{\max}$ را با توجه به میانگین طول یالها محاسبه می‌کنند. روش MAX-MIN AS از بزرگ شدن سریع بازخورد مثبت و همگرایی زود هنگام جلوگیری می‌کند.

- **سیستم مورچه بهترین-بدترین**

در سیستم مورچه بهترین-بدترین یا Best-worst AS [۲۰] از نخبه گرایی استفاده می‌شود. ضمن اینکه فرمونها را از یال‌هایی از مسیر بدتر، که در بهترین مسیر وجود ندارد، حرکت می‌دهند.

- **سیستم مورچه مبتنی بر رتبه**

در سیستم مورچه مبتنی بر رتبه یا Rank-based AS [۲۰و ۲۵] رابطه ۱۷ برای به روز رسانی ردپا استفاده می‌شود.

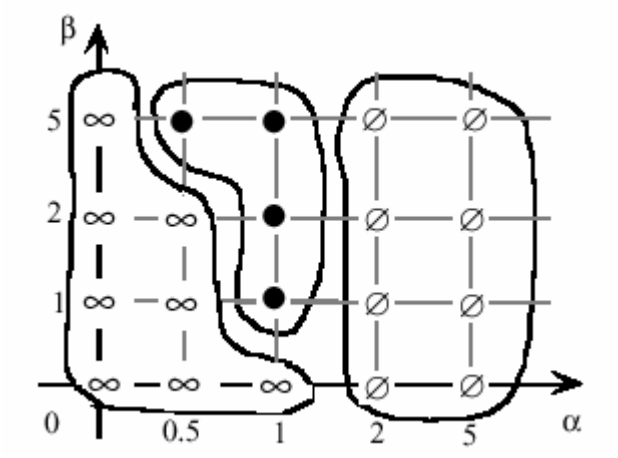
$$\tau_{ij}(t) = (1-\rho)\times\tau_{ij}(t-1) + \sum_{r=1}^{w-1}(w-r)\times\Delta\tau_{ij}(t-1) + w\times\Delta\tau_{ij}^{b}(t-1) \qquad (۱۷)$$

[1] - Reinforced learning.
[2] - MAX-MIN AS.

یعنی در هر تکرار، فقط بهترین مورچه‌های رتبه‌بندی شده[1] می‌توانند فرمون‌گذاری کنند ضمن اینکه بهترین مورچه کلی نیز تشویق می‌شود.

۹- پارامترهای ACO

یکی از معایب ACO داشتن پارامترهای زیاد و حساسیت الگوریتم به آنها اشاره کرد. مثل پارامترهای $Q, \alpha, \beta,$ ρ, η و ... . از میان این پارامترها، حساسیت الگوریتم به بعضی از آنها کمتر و به بعضی بیشتر است. مثلا Q از مقادیری است که انتخابهای متفاوت آن نقشی در عملکرد الگوریتم ندارد [۲]. اما در عوض مقادیر مختلف α و β رفتار الگوریتم را به شدت تحت تاثیر قرار می‌دهند. به عنوان مثال دوریگو در شکل ۵ رفتار الگوریتم ant cycle را در مقابل مقادیر مختلف α و β نشان داده است.



شکل ۵- رفتار AS در مقابل مقادیر مختلف α و β [۲]. الگوریتم جواب مناسب را بدون پیش آمدن وضعیت رکود پیشنهاد می‌کند. ∞ الگوریتم به هیچ جوابی همگرا نمی‌شود. Φ الگوریتم به جواب مناسب همگرا می‌شود.

در شکل ۵ مشاهده می‌شود که مقادیر بزرگ α الگوریتم را به جواب نامناسب همگرا می‌کند و مقادیر کوچک آن جلوی همگرایی الگوریتم را می‌گیرد. زیرا مقدار α تاثیر ردپا را در حرکات مورچه‌ها تنظیم می‌کند. پس اگر کوچک باشد اثر بازخورد مثبت را از بین می‌برد و اگر هم بزرگ باشد منجر به این می‌شود که اولین جوابها به عنوان جواب نهایی در نظر گرفته شود.

[1] - Best ranked ants.

مقدار پارامتر ρ، ضریب تبخیر باید طوری تنظیم شود که بازخورد منفی نقش خود را به خوبی ایفا کرده، بدون خنثی کردن اثر بازخورد مثبت، از رکود جلوگیری کند. پارامتر η، بینایی مسئله بوده، انتخاب آن مستقل از الگوریتم، کاملا به مسئله بستگی دارد.

۱۰- انواع مسائل قابل حل با ACO

به کمک الگوریتم ACO مسائل مختلفی حل می‌شود که بعضی عناوین آن به صورت زیر است:

1-Traveling Salseman Problem (TSP).[4,8]

2-Assymetrical Traveling Salseman Problem (ATSP).[8]

3-Quadratic Assignment Problem (QAP).[2,9]

4-Job-shop Scheduling Problem (JSP).[2,21]

5-Graph coloring.[10]

6-Seqential ordering.[22]

7-Network routing

8-Vehicle routing problem.[23]

9-Economic dispatch.[11,16]

10- Single machine total tardiness problem.

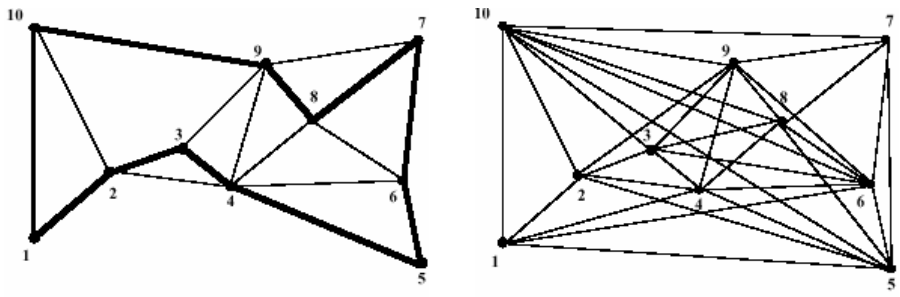
11-Short term generation scheduling problem.[12]

همچنین از این الگوریتم در کاربردهای مختلفی مثل data mining‘ [13] Optimal polygonal approximation [14] و classification rule discovery [15] نیزاستفاده می شود. در ادامه بعضی از معروفترین مسائل قابل حل ACO آورده شده است.

۱۰-۱- مسئله فروشنده دوره گرد

مسئله فروشنده دوره گرد **TSP**[1] از جمله مسائل بهینه‌یابی است که تاکنون توسط مسائل مختلف بهینه‌یابی حل شده است. اما تواناترین الگوریتم در حل آن ACSA می‌باشد. در این مسئله مطابق شکل ۶ تعدادی شهر وجود دارد که قرار است کوتاهترین مسیر بین آنها پیدا شود. این مسیر باید از یک نقطه شروع شده، از تمام شهرهای دیگر هر کدام یکبار عبور کند و به نقطه شروع برگردد و از هر شهر فقط یکبار بگذرد.

[1] - Traveling Salesman Problem.



ب الف

شکل ۶- الف) یک نمونه مسئله TSP و ب) کوتاهترین مسیر [۲].

مسئله فوق را می‌توان به صورت یک گراف تمام اتصال در نظر گرفت که هر گره آن نماینده یک شهر است. در این مسئله، تابع هزینه، همان تابع مسافت است یعنی برای هر یال $f_{ij} = d_{ij}$ و هزینه رفتن از گره i به j برابر d_{ij}، مسافت بین دو شهر i و j است. در این مسئله، بینایی هر یال توسط رابطه ۱۸ یعنی برابر عکس طول یال محاسبه می‌شود.

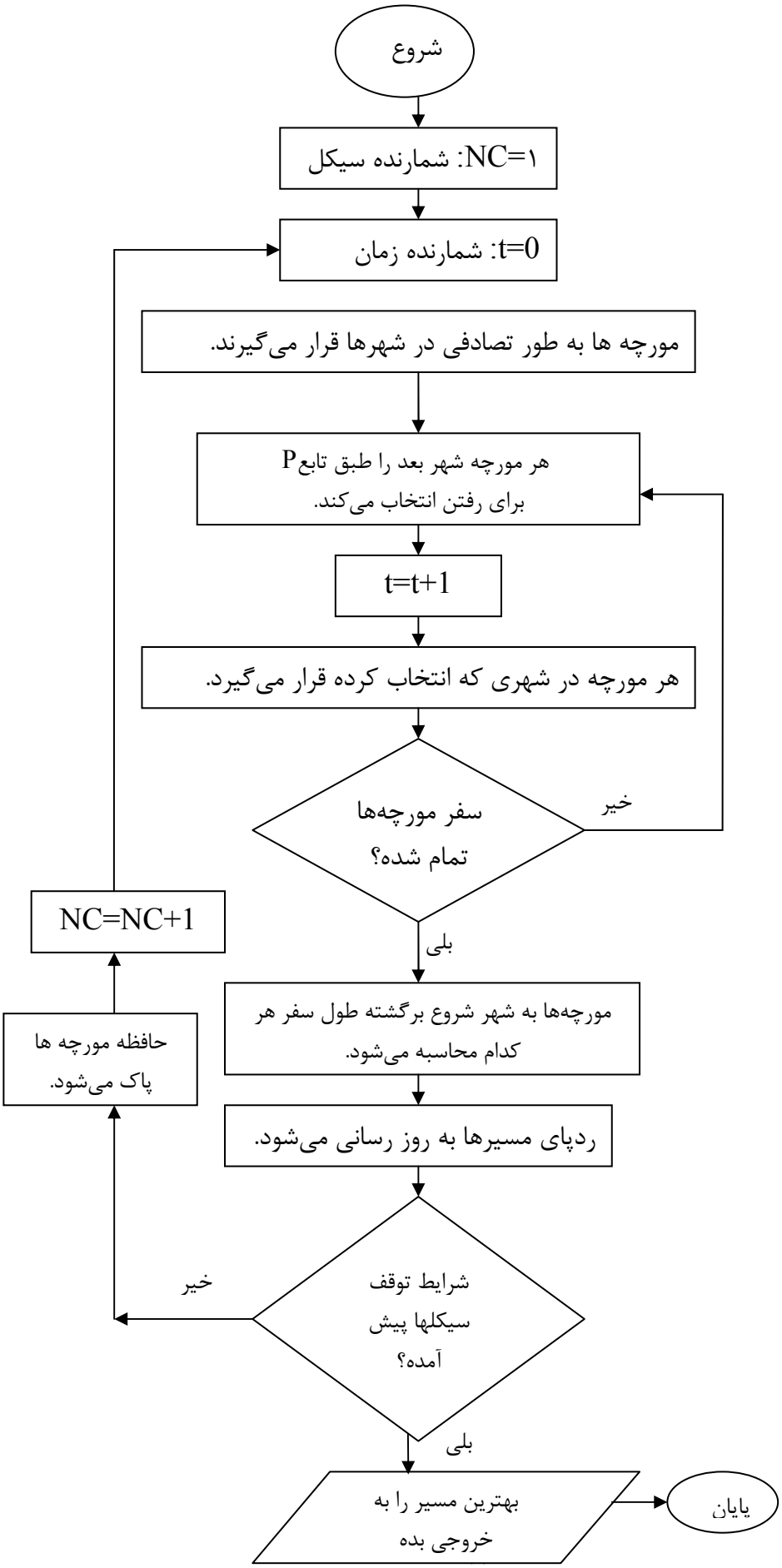
$$\eta_{ij} = \frac{1}{d_{ij}} \tag{۱۸}$$

در دیاگرام بلوکی شکل ۷، الگوریتم Ant-cycle برای حل مسئله فروشنده دوره گرد آورده شده است. در این الگوریتم، تابع P به صورت رابطه ۱۹ در می‌آید. و به روز کردن ردپا نیز مطابق روش AS طبق روابط ۲۰ و ۲۱ انجام می‌شود. در این روابط D^k طول سفر مورچه k است.

$$P_{ij}^k = \begin{cases} \dfrac{(d_{ij})^{-\alpha} \times (\tau_{ij})^{\beta}}{\sum\limits_{l \in admissible\ nodes} (d_{il})^{-\alpha} \times (\tau_{il})^{\beta}} & if\ \ the\ \ Kth\ \ ant\ dont\ \ traverse\ j \\ 0 & if\ \ the\ \ Kth\ \ ant\ \ traverse\ \ j \end{cases} \tag{۱۹}$$

$$\tau_{ij}^k = (1-\rho) \times \tau_{ij} + \Delta\tau_{ij} \tag{۲۰}$$

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^k \quad , \quad \Delta\tau_{ij}^k = \begin{cases} Q/D^k & if\ the\ \ Kth\ \ ant\ \ traverse\ L_{ij} \\ 0 & otherwise \end{cases} \tag{۲۱}$$


شروع
NC=۱: شمارنده سیکل
t=0: شمارنده زمان
مورچه ها به طور تصادفی در شهرها قرار می‌گیرند.
هر مورچه شهر بعد را طبق تابع P
برای رفتن انتخاب می‌کند.
t=t+1
هر مورچه در شهری که انتخاب کرده قرار می‌گیرد.
سفر مورچه‌ها
تمام شده؟
خیر
بلی
NC=NC+1
مورچه‌ها به شهر شروع برگشته طول سفر هر
کدام محاسبه می‌شود.
حافظه مورچه ها
پاک می‌شود.
ردپای مسیرها به روز رسانی می‌شود.
شرایط توقف
سیکلها پیش
آمده؟
خیر
بلی
بهترین مسیر را به
خروجی بده
پایان

نحوه استفاده ازحافظه مورچه در TSP به این صورت است که در حافظه هر مورچه لیستی به نام لیست ممنوع[1] وجود دارد که مورچه، شهرهایی را که به ترتیب در سفر اخیرش از آنها عبور کرده، در آن قرار می‌دهد. از اطلاعات این لیست برای محاسبه طول سفر مورچه در انتهای سفر و تعیین گره‌های غیر مجاز در هر زمان t استفاده می-شود.

۱۰-۲- مسئله فروشنده دوره گرد نامتقارن

مسئله فروشنده دوره گرد نامتقارن یا ATSP[2]، نمونه ای نامتقارن از مسئله TSP می‌باشد. این مسئله مشابه TSP است با این تفاوت که ماتریس هزینه آن نامتقارن است یعنی هزینه یال رفت آن متفاوت از هزینه یال برگشت می‌باشد. $(f_{ij} \sim= f_{ji})$. پس یک گراف جهتی وجود دارد که بین هر دو گره آن مسیر رفت و برگشت متفاوت است. به طوری که مقداربینایی و ردپای موجود در آنها لزوما مشابه نیستند. مورچه برای حرکت روی این گراف به جهت گراف توجه می‌کنند.

۱۰-۳- مسئله تخصیص دو بعدی

در مسئله تخصیص دو بعدی یا **QAP**[3]، n خدمت به n موقعیت واگذار می‌شود. فاصله بین موقعیتها را در ماتریس D و هزینه انتقال اطلاعات، تولیدات یا ... بین این خدمات در ماتریس F موجود است.

اگر Π نشاندهنده تخصیص خدمت $h=\Pi(i)$ به موقعیت i باشد، هدف تشخیص Π برای شماره ردیف و ستونهای ماتریس می‌باشد به نحوی که هزینه کل مطابق رابطه ۲۲ کمینه گردد.

$$\min Z = \sum_{i,j=1}^{n} d_{ij} \times f_{\Pi(i)\Pi(j)} \qquad (۲۲)$$

d_{ij}: فاصله دو موقعیت i و j و f_{hk} هزینه انتقال اطلاعات بین دو خدمت h و k می‌باشند.

برای حل این مسئله توسط ACO ، ابتدا بردارهای پتانسیلی D^p وF^p ساخته می‌شوند که حاصلجمع سطرهای ماتریسهای D و F می‌باشد. به کمک این دو ماتریس، ماتریس سوم S به وجود می‌آید که هر درایه آن به صورت رابطه ۲۳ محاسبه می‌شود.

$$S_{ih} = d_i \times f_h \qquad (۲۳)$$

حالا یک ماتریس هزینه نامتقارن وجود دارد که نماینده مسئله QAP می‌باشد و می‌تواند با تبدیل به گراف، مشابه یک مسئله ATSP حل شود.

---

[1]- Tabu list.
[2]- Asymmetrical Traveling Salesman Problem.
[3] - Quadratic Assignment Problem.

۱۱- مراجع

[1] P.Tarasewich, and P.R.McMullen, "Swarm intelligence: power in numbers", Appears in communication of ACM, Agust 2002, 45(8), 62-67.

[2] M.Dorigo, V.Maniezzo, and A.Colorni, "The Ant System: optimization by a colony of cooperating agents", IEEE Transaction on systems, Man, and Cybernetics-part B, vol. 26, no.1, 1996, pp. 1-13.

[4] M.Dorigo, and L.M.Gambardella, "Ant Colony System: A cooperating learning approach to the traveling salesman problem", IEEE Transactions on evolutionary computation, vol. 1, no. 1, 1997.

[5] M.Dorigo, and L.M.Gambardella, "A study of some properties of Ant-Q",Proceeding of PPSN fourth intrnational conference on Parallel Problem Solving from Nature, H.-M. Viot, W.Ebeling, I.Rechenberg and H.-S.Schwefel(Eds), springer-Verlag, Berlin, pp. 656-665, 1996. www.schatten.info/info/citations/22.html.

[6] T.Stutzle, and H.Hoos," Max-Min Ant-System and local search for the traveling salesman problem", Technical university of Darmstadt, , Proc. 1997 IEEE International Conference on Evolutionary Computation, 1997.

www.cs.ubc.ca/spider/hoos/publ/fgcsoo.pdf.

[7] S.J.Shyu, P.Y.Yin, and B.M.T.Lin, "An ant colony optimization algorithm for the minimum weight vertex cover problem", www.winforms.phil.tu-bs.de/lehre/vorlesungen/puet/unterlagen/shy-etul-papper.pdf.

[8] L.M.Gambardella, and M.Dorigo, " Solving symmetric and asymmetric TSPs by ant colonies",In IEEE Conference On Evolutionary Computation(ICEC96)T, IEEE press, pp. 622-624, 1996.

[9] T.Stutzle and M.Dorigo, "ACO algorithms for the Quadratic Assignment Problem", In D. Corne, M. Dorigo, and F. Glover, editors, New Ideas in Optimization, pages 33-50, McGraw-Hill, 1999.

www.users.cs.york.ac.uk/~susan/bib/nf/d/marcodrg.htm.

[10] Krzysztof.Wolkowiak, "Graph coloring using ant algorithms", www.zssk.pwr.wroc.pl/~wolkow.user/pappers/kw-graph-kosyrol.pdf.

[11] Y.H. Song, C.S.Chou, and T.J.Stonham," Combind heat and power economic dispatch by improved ant colony search algorithm", Electric Power System Research, Vol. 52, pp. 115-121, 1999.

[12] I.K.Yu, Y.H.Song, "A novel short-term generation scheduling of thermal units using ant colony search algorithms", Electrical power and energy systems, Vol.23, pp. 471-479, 2001.

[13] Rafael S.Parpinelli, Heitor S.Lopes, and Alex A.Freitas, "Data mining with an ant colony optimization algorithm", IEEE Transactions on evolutionary computation, vol. 6, no. 4, August 2002, www.ppgia.pucpr.br/alex/pub-papers.dir/Ant-IEEE-TEC.pdf.

[14] Peng-Yeng Yin, "Ant colony search algorithms for optimal polygonal approximation of plane curves", journal of the pattern recognition society,2003.

[15] Rafael S.Parpinelli, Heitor S.Lopes, and A.A.Freites, "An ant colony algorithm for classification rule discovery", www.ppgia.pucpr.br/~alex/pub-papers.dir/heuristic-DM-bk.pdf.

[16] Yun-He Hou, Yao-Wu Wu, Li-Juan Lu, and Xin-Xin Xiong, "Generalized ant colony optimization for economic dispetch of power systems", www.psasp.com.cn/powercon/tslist.htm.

[17] M.Dorigo, G.D.Caro, and L.M.Gambardella, "Ant algorithms for discrete optimization", Artifitial Lhfe, Vol. 5, No. 3, pp. 137-172, 1999.

[18] M.Dorigo, and G.D.Caro, "Ant colony optimization: A new meta-heuristic",Proceedings of the evolutionary computation, Washington DC, pp. 1470-1477,july 1999.

[19] R.Keinprasit, and P.Chongstiratana, "High-level synthesis by dynamic ant", www.cp.eng.chula.ac.th/faculty/pjw/pappers/dynamic-ant.pdf.

[20] T.Stutzle, "Ant colony optimization", www.intellektik.informatik.tu-damstadat.de/~tom/ssol/folien/v8b.pdf.

[21] V. Ramos, F. Muge, and P. Pina, "Self organized data and image retrieval as a consequence of inter-dynamic synergistic relationships in artifitial ant colonies", 2nd Int. Conf. on Hybrid Intelligent Systems, IOS Press, Vol. 87, ISBN 1

[22] I. Gokcen, I. H. Pineda, X.Yuan, C. Koutsougeras, and B. P. Buckles, "Image segmentation using ant colony system", Late Breaking Papers at the 2000 Genetic and Evolutionary Computation Conference (GECCOlb), 2000.