

بسمه تعالی



دانشکده برق و کامپیوتر
دانشگاه صنعتی اصفهان

هوش جمعی

گزارش سمینار درس یادگیری ماشین

نیمسال اول ۸۶-۸۵

مدرس: جناب آقای دکتر پالهنگ

ارائه دهنده: نرجس خاتون حبیبی

زمستان ۸۵

فهرست مطالب

- ۱- هوش جمعی (SI) ۳
- ۱-۱- الگوریتم های SI ۳
- ۲-۱- کاربردهای SI ۴
- ۲- روش مورچه ها برای یافتن غذا ۴
- ۳- الگوریتم AS ۷
- ۴- ACO ۷
- ۱-۴- تعریف مساله بهینه سازی گسسته ۷
- ۲-۴- الگوریتم ACO ۸
- ۳-۴- رفتار کلونی های مورچه ها ۹
- ۴-۴- ACO برای حل چه مسائلی مناسب است؟ ۱۲
- ۵- مساله فروشنده دوره گرد (TSP) ۱۲
- ۶- الگوریتم مسیریابی AntNet ۱۳
- ۷- نتیجه گیری ۱۵
- ۸- مراجع ۱۶

۱- هوش جمعی (SI)

هوش جمعی یک شیوه هوش مصنوعی بر اساس مطالعه رفتار جمعی در سیستم های غیرمتمرکز خود سامانده^۲ است. سیستم های SI معمولا از جمعیتی از عوامل ساده تشکیل شده که به طور محلی با یکدیگر و با محیطشان تعامل دارند. هر چند که در این سیستم ها یک ساختار کنترلی متمرکز وجود ندارد که رفتار عامل ها را به آن ها تحمیل کند، تعاملات محلی باعث به وجود آمدن یک رفتار سراسری^۳ هوشمندانه می شود.

مثال هایی از سیستم های SI در طبیعت یافت می شود، کلونی های مورچه ها، دسته های پرندگان، گله های حیوانات، تجمعات باکتری ها و گله های ماهی ها.

۱-۱- الگوریتم های SI

الگوریتم های مختلفی از رفتارهای جمعی طبیعی الهام گرفته شده است که در ادامه به سه مورد اشاره می شود.

- الگوریتم بهینه سازی کلونی مورچه ها (ACO)^۴ : برای یافتن راه حل های تقریبی در مسائل بهینه سازی ترکیبی استفاده می شود. در ACO مورچه های مصنوعی راه حل را با حرکت بر روی گراف مساله می سازند و با تقلید از رفتار مورچه های واقعی، بر روی گراف فرومون باقی می گذارند، به نحوی که مورچه های آینده بتوانند جواب های بهتری بیابند.
- الگوریتم بهینه سازی جمعی ذره ها (PSO)^۵ : روش PSO یک روش سراسری کمینه سازی است که با استفاده از آن می توان با مسائلی که جواب آنها یک نقطه یا سطح در فضای n بعدی می باشد، برخورد نمود. در این چنین فضایی، فرضیاتی مطرح می شود و یک سرعت ابتدایی به آنها اختصاص داده می شود، همچنین کانال های ارتباطی بین ذرات در نظر گرفته می شود. سپس این ذرات در فضای پاسخ حرکت می کنند، و نتایج حاصله بر مبنای یک "ملاک شایستگی"^۶ پس از هر بازه زمانی محاسبه می شود. با گذشت زمان، ذرات به سمت ذراتی که دارای ملاک شایستگی بالاتری هستند و در گروه ارتباطی یکسانی قرار دارند، شتاب می گیرند. مزیت اصلی این روش بر استراتژی های کمینه سازی دیگر این است که تعداد فراوان ذرات ازدحام کننده، باعث پایداری روش در برابر مشکل پاسخ کمینه محلی می گردد.

¹ Swarm Intelligence

² Self-organized

³ global

⁴ Ant Colony Optimization

⁵ Particle Swarm Optimization

⁶ Fitness criterion

• الگوریتم جستجوی انتشار تصادفی (SDS)^۷: یک روش جستجوی سراسری احتمالی برای مسائلی که تابع هدف می تواند به چند تابع جزئی مستقل^۸ تجزیه شود. هر عامل فرضیه ای دارد که مرتباً توسط تابع هدف جزئی تصادفی ارزیابی می شود. این ارزیابی باعث فعال شدن یا غیرفعال شدن عامل می گردد. اطلاعات فرضیه ها، بوسیله ارتباطات بین عامل ها، در جمعیت پخش می شود. SDS یک روش جستجو و بهینه سازی کارا، ارزیابی شده است.

۲-۱- کاربردهای SI

روش های مبتنی بر SI در موارد بسیاری، به طور موفق، به کار گرفته شده اند، از جمله در کنترل وسایل بدون سرنشین، نقشه برداری نجومی، کنترل حرکت روبات های بسیار کوچک^۹ در بدن به منظور از بین بردن غدد سرطانی و ساخت محیط های محاوره ای پیچیده. (به عنوان مثال در فیلم های شیر شاه و ارباب حلقه ها)

۲- روش مورچه ها برای یافتن غذا

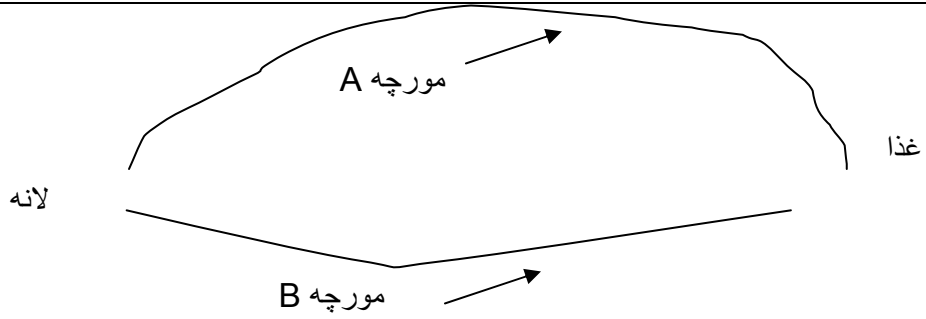
روش غذایی مورچه ها به این صورت است که وقتی به یک منبع غذایی می رسند، در بازگشت به لانه در مسیر ماده ای به نام فرومون ترشح می کنند. فرومون محرکی قوی است، به طوری که مورچه های بعدی با حس آن، با احتمال بیشتری مسیر فرومون دار را انتخاب می کنند. مقدار فرومون یک مسیر نشان دهنده تعداد مورچه هایبست که آن مسیر را پیموده اند. هر چند فرومون شانس دنبال کردن مسیری خاص را افزایش می دهند، اما تضمین نمی کند که تمام مورچه ها مسیر مذکور را طی کنند. این خصوصیت برای تطبیق به محیط های در حال تغییر ضروری است. ممکن است مسیر نزدیک تری به غذا وجود داشته باشد که توسط مورچه ای که از مسیر فرومون دار نمی رود، کشف می شود. در یک بازه زمانی، تعداد مورچه هایی که بین غذا و لانه از مسیر کوتاه تر حرکت می کنند، از تعداد مورچه هایی که مسیر قدیمی را می پیمایند بیشتر است، در نتیجه فرومون آن با سرعت بیشتری افزایش می یابد. به علاوه فرومون به تدریج تبخیر می شود، پس میزان فرومون مسیر قدیمی کاهش یافته و مورچه ها به مسیر کوتاه تر همگرا می شوند. شکل ۱ مثالی را نشان می دهد.

شکل ۱

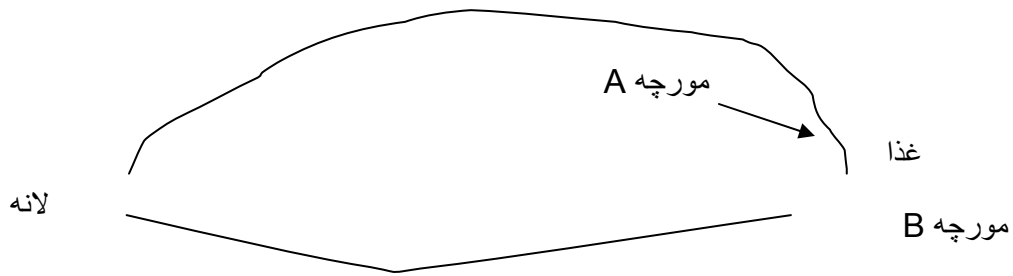
⁷ Stochastic Diffusion Search

⁸ Multiple independent partial-function

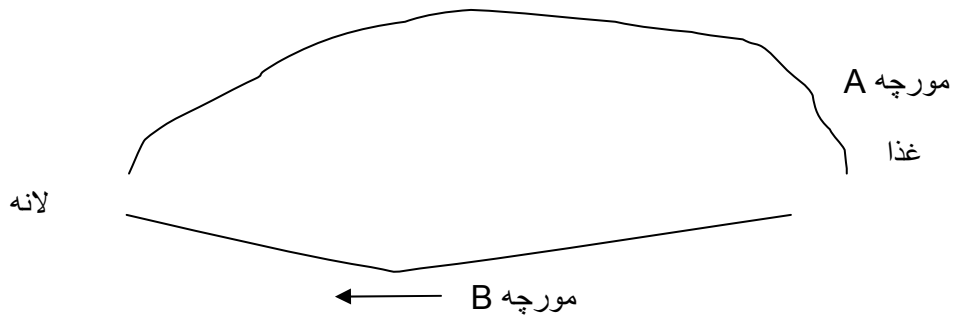
⁹ nanobots



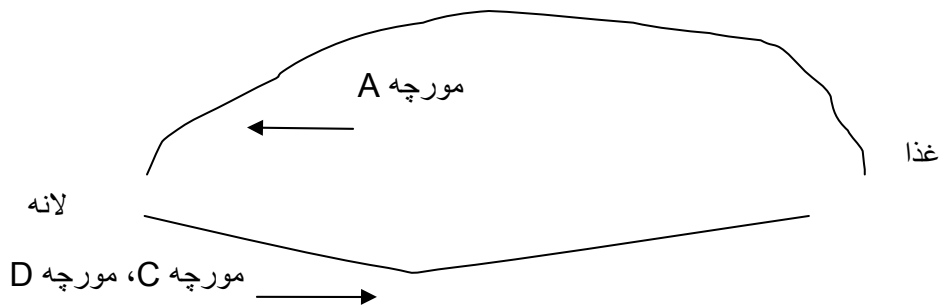
شکل ۱-۱ آغاز جستجو برای یافتن کوتاه ترین مسیر به غذا



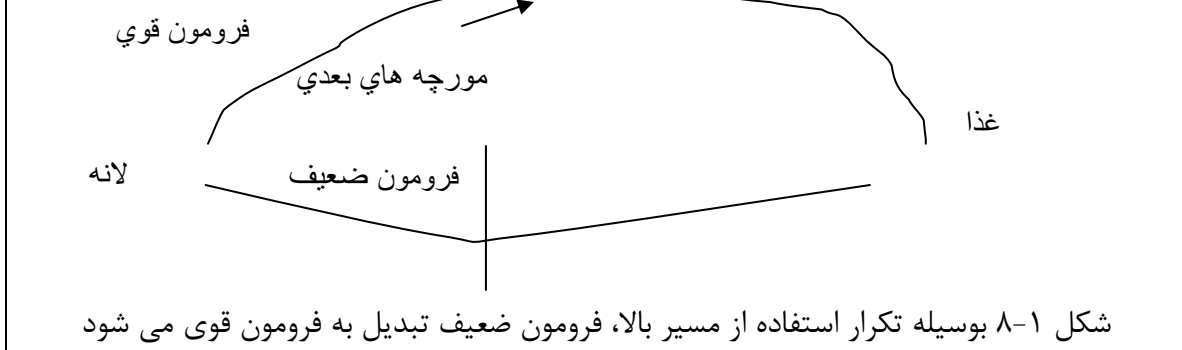
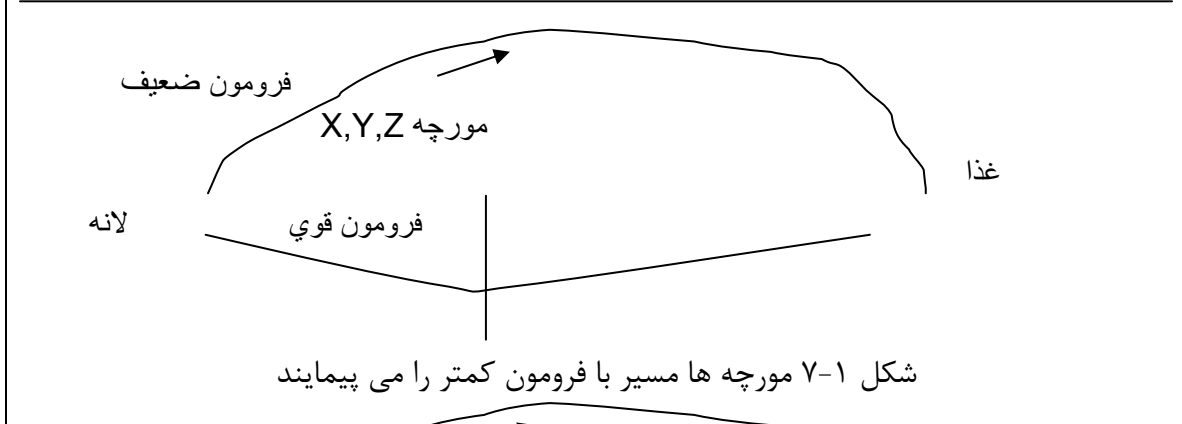
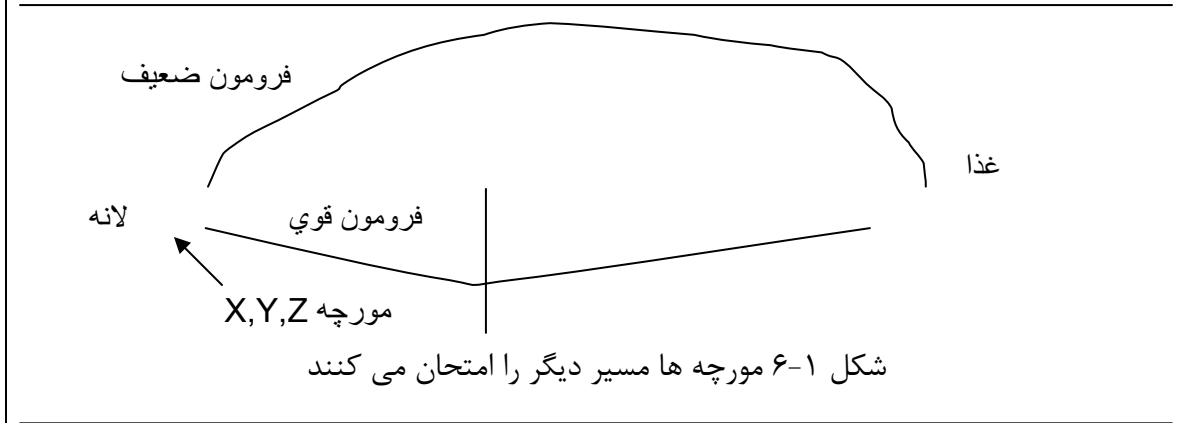
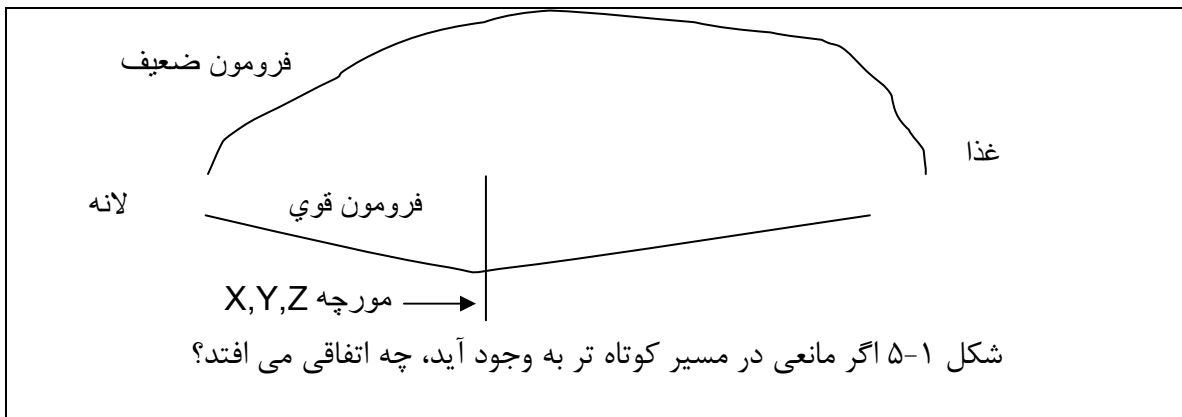
شکل ۲-۱ مورچه B غذا را پیدا می کند



شکل ۳-۱ مورچه A غذا را پیدا می کند. مورچه B در مسیر برگشت فرمون ترشح می کند



شکل ۴-۱ مورچه C و D مسیر فرمون دار که B طی کرده را انتخاب می کنند



دو خصوصیت مهم روش غذایی مورچه ها ۱- انتخاب تصادفی مسیر و ۲- تبخیر شدن فرومون در طول زمان است.

۳- الگوریتم AS^{10}

الگوریتم AS در سال ۱۳۷۰ شمسی^{۱۱} به عنوان روشی ابتکاری^{۱۲} برای حل مسائل بهینه سازی ترکیبی^{۱۳} پیشنهاد شده است. این الگوریتم از روش جستجوی غذای مورچه ها الهام گرفته و در ابتدا برای حل مساله کلاسیک فروشنده دوره گرد به کار رفت.

نسخه های توسعه یافته AS عبارتند از: ACS ، $Max-Min$ و AS_{rank}

از مسائل بهینه سازی که AS به آنها اعمال شده است، می توان تخصیص درجه دوم^{۱۴}، مسیریابی وسیله نقلیه، مسیریابی شبکه، رنگ آمیزی گراف و کوله پشتی را نام برد. الگوریتم AS در زمره بهترین روش های ابتکاری موجود قرار دارد.

۴- ACO^{15}

ACO چارچوبی است که برای تمام نسخه های AS تعریف شده است. در ادامه نحوه مدل کردن یک مساله برای ACO ، الگوریتم آن و موارد مناسب برای حل با ACO شرح داده می شود.

۴-۱- تعریف مساله بهینه سازی گسسته

یک مساله بهینه سازی گسسته به صورت زیر تعریف می شود:

۱. مجموعه متناهی از اجزا^{۱۶}: $C = \{c_1, c_2, \dots, c_N\}$

۲. مجموعه متناهی از اتصالات/انتقالات میان اجزای زیر مجموعه ای از $C \times C$:

$$L = \{l_{c_i c_j} \mid c_i, c_j \in C \times C\}, |L| \leq N^2$$

۳. تابع هزینه ارتباط که به هر $l_{c_i c_j} \in L$ نسبت داده می شود و ممکن است به زمان t وابسته

$$J_{c_i c_j} = J(l_{c_i c_j}, t) \text{ باشد}$$

۴. مجموعه متناهی از محدودیت ها که به اجزای C و L نسبت داده می شود:

$$\Omega = \Omega(C, L, t)$$

¹⁰Ant System

¹¹ 1991 AC

¹² heuristic

¹³ combinatorial

¹⁴ quadratic assignment

¹⁵ Ant Colony Optimization

¹⁶ components

۵. ترتیبی بر روی اجزای C که یک وضعیت هم نامیده می شود:

$$S = \langle C_i, C_j, \dots, C_k, \dots \rangle$$

اگر S مجموعه تمام ترتیب های ممکن باشد، \tilde{S} زیرمجموعه ای از S است که دنباله های آن با توجه به محدودیت های Ω محتمل اند.^{۱۷}

۶. ساختار همسایگی: دو وضعیت s_1 و s_2 همسایه نامیده می شوند اگر s_1 و s_2 در S باشند و بتوان در یک گام منطقی از s_1 به s_2 رسید. (اگر آخرین جزء ترتیبی s_1 باشد، $c_2 \in C$ موجود باشد که $l_{c_1 c_2} \in L$ و $\langle s_1, c_2 \rangle \equiv s_2$). همسایگی وضعیت S با N_s نمایش داده می شود.

۷. ψ یک راه حل است اگر عضوی از \tilde{S} باشد و تمام شرایط مساله را ارضا کند.

۸. $J_\psi(L, t)$: هزینه هر راه حل ψ که تابعی از تمام هزینه های $J_{c_i c_j}$ متعلق به ψ است.

اگر گراف $G = (C, L)$ را به مساله ACO مرتبط کنیم، راه حل مساله بهینه سازی می تواند به صورت مسیرهای محتمل روی G در نظر گرفته شود. الگوریتم ACO برای یافتن مسیر محتمل (با توجه به محدودیت های Ω) با حداقل هزینه به کار می رود. به عنوان مثال مساله فروشنده دوره گرد را به صورت زیر می توان مدل کرد:

- C = شهرها
- L = مسیرهای میان شهرها
- ψ = مسیر همیلتونی

۴-۲- الگوریتم ACO

این الگوریتم از جمعیتی از مورچه ها برای حل دسته جمعی مساله استفاده می کند. اطلاعاتی که در طول پروسه جستجو توسط مورچه ها جمع آوری می شود، به صورت دنباله های فرومون که به هر یال l_{ij} نسبت داده میشود، ذخیره می گردد. در واقع دنباله های فرومون یک حافظه بلند مدت اند. یال ها می توانند یک مقدار η_{ij} ابتکاری داشته باشند که نشان دهنده ۱- یک اطلاع پیشین در مورد مساله است و یا ۲- اطلاعات زمان اجرا که توسط منبعی غیر از مورچه ها فراهم می شود.

الگوریتم ACO سه رویه اصلی دارد:

- رویه تبخیر فرومون^{۱۸}: پروسه کاهش دنباله فرومون بر روی یال ها در طول زمان. تبخیر فرومون به منظور جلوگیری از همگرایی سریع الگوریتم به یک ناحیه زیر بهینه^{۱۹} انجام می شود. هدف اکتشاف نواحی جدید فضای جستجو است.

¹⁷ Feasible states

¹⁸ pheromone evaporation

- رویه برنامه کمکی کنش^{۲۰}: جزء اختیاری ACO است که می تواند برای پیاده سازی عملیات مرکزی که توسط هر مورچه نمی تواند انجام شود، به کار گرفته شود. برای مثال یک برنامه کمکی کنش می تواند مسیر یافت شده توسط تمام مورچه ها را مشاهده کند و فقط فرمون های کوتاه ترین مسیر را افزایش دهد. به این روش "به روز رسانی برون خطی فرمون"^{۲۱} گفته می شود.
- رویه فعالیت مورچه ها: هر کولنی از مورچه ها به طور موازی و غیر همزمان در طول وضعیت های مساله حرکت می کنند. وضعیت بعدی هر مورچه با یک سیاست تصمیم گیری محلی تصادفی از روی اطلاعات جدول مسیریابی محلی گره فعلی مشخص می شود. به این ترتیب مورچه ها به تدریج جواب مساله را می سازند. بعد از ساخت جواب و یا در حین ساخت، هر مورچه جواب (جزئی) را ارزیابی و طبق خوب بودن آن مقادیر فرمون های مسیری را که طی کرده به روز رسانی می کند. فرمون ها، جستجوی مورچه های آینده را هدایت می کنند. در ادامه این رویه به تفصیل بررسی می شود.

۳-۴ - رفتار کولنی های مورچه ها

۱. هر مورچه برای راه حل محتمل بهینه جستجو می کند:

$$\hat{J}_\Psi = \min_{\Psi} J_\Psi(L, t)$$
۲. مورچه K حافظه M^k دارد که برای ذخیره اطلاعات در مسیر خود از آن استفاده می کند. حافظه می تواند برای ذخیره راه حل، ارزیابی راه حل و برگشت مسیر پیموده شده، به کار رود.
۳. مورچه K در وضعیت $\langle S_{r-1}, i \rangle$ می تواند به هر گره j واقع در همسایگی گره i یعنی N_i^k حرکت کند.
۴. به هر مورچه یک وضعیت شروع S_s^k و یک یا چند شرط پایان نسبت داده می شود.
۵. حرکت مورچه ها از وضعیت شروع آغاز شده و به سمت وضعیت های همسایه ادامه می یابد و راه حل به صورت تدریجی ایجاد می گردد.
۶. مورچه K در گره i می تواند به گره j در N_i^k حرکت کند. حرکت بر اساس یک قانون تصمیم گیری احتمالی انتخاب می شود.
۷. قانون تصمیم گیری احتمالی مورچه ها تابعی از موارد زیر است:

¹⁹ Sub-optimal

²⁰ Daemon action

²¹ offline delayed pheromone update

- محدودیت های مساله
 - حافظه اختصاصی مورچه که تاریخچه گذشته را ذخیره می کند.
 - اطلاعاتی که در ساختمان داده محلی گره به نام "جدول مسیریابی مورچه" ذخیره شده است ($A_i = [a_{ij}]$). این اطلاعات با ترکیب دنباله های فرومون و مقادیر ابتکاری به دست می آید.
۸. مورچه ها هنگام حرکت از گره i به j می توانند دنباله فرومون Γ_{ij} یال (i, j) را به روزرسانی کنند که به آن به "روز رسانی مرحله به مرحله برخط فرومون"^{۲۲} گفته می شود.
۹. مورچه می تواند بعد از ساخت یک راه حل، مسیر آمده را برگردد و دنباله های فرومون را بر روی یال های طی شده به روز رسانی کند. به این روش "به روز رسانی با تاخیر برخط فرومون"^{۲۳} گفته می شود.
۱۰. بعد از ساخت راه حل مورچه می میرد و تمام منابع اختصاص یافته را آزاد می کند. در این روش راه حل از تعامل جمعی مورچه ها ایجاد می شود. هر مورچه تنها از ۱- اطلاعات خصوصی خود و ۲- اطلاعات محلی گره ای که ملاقات می کند، استفاده می نماید. ارتباط میان مورچه ها به طور غیر مستقیم و از راه تغییری که در مقادیر فرومون ها می دهند، برقرار می شود. در شکل ۲ شبه کد الگوریتم ACO نمایش داده شده است.

ACO-meta-heuristic()	۱
تا زمانی که شرط خاتمه ارضا نشده است، انجام بده	۲
فعالیت ها را زمان بندی کن	۳
ants-generation-and-activities()	۴
رویه تبخیر فرومون را فراخوانی کن	۵
رویه برنامه کمکی کنش را فراخوانی کن {اختیاری}	۶
پایان زمان بندی فعالیت ها	۷
پایان حلقه	۸
پایان رویه	۹
ants-generation-and-activities()	۱

²² online step-by-step pheromone update

²³ online delayed pheromone update

تا زمانی که منابع ²⁴ آزاد وجود دارد، انجام بده	۲
ایجاد مورچه بعدی را زمان بندی کن	۳
new-active-ant()	۴
پایان حلقه	۵
پایان رویه	۶
رویه new-active-ant() {دوران حیات یک مورچه}	۱
مورچه را مقداردهی اولیه کن	۲
$M \rightarrow$ حافظه مورچه را به روز رسانی کن	۳
تا زمانی که وضعیت جاری \neq وضعیت هدف، انجام بده	۴
$A \rightarrow$ جدول مسیریابی محلی گره فعلی را بخوان	۵
$\rho \rightarrow$ احتمال انتخاب هر گره همسایه را محاسبه کن (A, M, Ω)	۶
وضعیت بعدی \rightarrow سیاست تصمیم گیری مورچه را اعمال کن (Ω, ρ)	۸
به وضعیت بعدی برو	۹
اگر روش به روزرسانی فرمون "مرحله به مرحله برخط" است	
بر یال ملاقات شده فرمون باقی بگذار	۱۰
جدول مسیریابی مورچه را به روزرسانی کن	۱۱
$M \rightarrow$ حالت داخلی مورچه را به روزرسانی کن	۱۲
پایان حلقه	۱۳
اگر روش به روزرسانی فرمون "با تاخیر برخط" است	۱۵
برای هر یال ملاقات شده عضو جواب انجام بده	۱۶
بر تمام یال های ملاقات شده فرمون باقی بگذار	۱۷
جدول مسیریابی مورچه را به روزرسانی کن	۱۸
به حیات خود پایان بده	۱۹
پایان رویه	۲۰

شکل ۲- شبه کد الگوریتم ACO

جزئیات پیاده سازی سه رویه در ACO، به طراح واگذار شده است.

²⁴ resources

۴-۴-ACO برای حل چه مسائلی مناسب است؟

الگوریتم ACO، برای مسائلی مناسب است که الگوریتم های کلاسیک مثل برنامه نویسی پویا کارایی خوبی ندارند. از جمله در موارد زیر:

- مسائل غیر چندجمله ای سخت^{۲۵} که ابعاد گراف کامل فضا-حالت به طور توانی با ابعاد مساله افزایش می یابد. در این حالت مورچه ها از یک گراف بسیار کوچکتر G که از اجزای مساله ایجاد می شود، استفاده می کنند.
- مسائل یافتن کوتاهترین مسیر که خصوصیات گراف در طی زمان تغییر می کند. به عنوان مثال در مسیریابی شبکه های کامپیوتری هزینه اتصالات می تواند ثابت نباشد.
- مسائلی که ساختار محاسباتی توزیع شده دارند. ACO به علت ماهیت چندعاملی و توزیع شده اش، برای این دسته از مسائل مناسب است.

۵- مساله فروشنده دوره گرد (TSP)

فرض می کنیم C مجموعه ای از شهرها و L مجموعه ارتباطات میان شهرها و $l_{c_i c_j}$ فاصله میان دو شهر c_i و c_j باشد. مساله TSP پیدا کردن مدار همیلتنی با طول حداقل روی گراف $G = (C, L)$ است. فاصله میان دو شهر ممکن است نامتقارن بوده و گراف می تواند کاملاً متصل نباشد. TSP اولین مساله ای بود که ACO بر روی آن اعمال شد. در این قسمت راه حل TSP به روش AS توضیح داده می شود: m مورچه، همزمان، به صورت تصادفی روی m شهر قرار می گیرند. در حافظه M^k مورچه K، شهر شروع قرار داده می شود. در هر مرحله مورچه ای که بر گره i است، جدول مسیریابی این گره را خوانده و مقدار نسبت داده شده به هر همسایه را که طبق فرمول زیر محاسبه شده، می یابد:

$$a_{ij} = \frac{[\Gamma_{ij}(t)]^\alpha [n_{ij}]^\beta}{\sum_{l \in N_i} [\Gamma_{il}(t)]^\alpha [n_{il}]^\beta}$$

α و β دو مقداری هستند که میزان تاثیر فرمون و مقدار ابتکاری را کنترل می کنند. در این فرمول اگر $\alpha = 0$ باشد، نزدیکترین شهر بعدی انتخاب می شود و در واقع الگوریتم به صورت حریصانه عمل می کند. اگر $\beta = 0$ باشد، ایستایی^{۲۶} به وجود می آید. بدین مفهوم که تمام مورچه ها یک مسیر را خواهند پیمود که به احتمال خیلی زیاد بهینه نخواهد بود. سپس بوسیله قانون تصمیم گیری زیر شهر بعدی را انتخاب می کند:

²⁵ NP-hard

²⁶ stagnation

$$p_{ij}^k(t) = \frac{a_{ij}(t)}{\sum_{l \in N_i^k} a_{il}(t)}$$

مورچه شهر انتخاب شده را به حافظه خود اضافه می کند و به شهر بعدی می رود. در AS، "به روزرسانی مرحله به مرحله برخط فرومون" وجود ندارد. بعد از اتمام سفر مورچه ها، هر مورچه با استفاده از حافظه خود مسیر آمده را برمی گردد و مقدار فرومون Γ_{ij} را طبق رابطه زیر برای تمام یال های ملاقات شده افزایش می دهد:

$$\Delta \Gamma^k(t) = 1/J_{\psi}^k(t) \quad \forall l_{ij} \in \psi^k(t), k = 1, \dots, m$$

$$\Gamma_{ij}(t) \leftarrow \Gamma_{ij}(t) + \Delta \Gamma^k(t)$$

در انتها می میرد و تمام منابع تخصیص یافته به خود را آزاد می کند. بعد از به روزرسانی فرومون ها توسط تمام مورچه ها، رویه تبخیر فرومون فعال می شود و به صورت زیر مقدار فرومون تمام یال ها را کاهش می دهد:

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

قرار دادن مقادیر $\alpha = 1$ ، $\beta = 0.5$ و $\rho = 0.5$ به طور تجربی توسط M. Dorigo خوب ارزیابی شده است.

۶- الگوریتم مسیریابی AntNet

AntNet قراردادی^{۲۷} برای مسیریابی^{۲۸} بسته ها^{۲۹} در شبکه های packet-switched است که توسط M. Dorigo and G. Di Caro ابداع شده است. این الگوریتم جایگزینی برای قرارداد OSPF، بر اساس ACO است.

دو نوع عامل (مورچه) در AntNet وجود دارد:

- عامل جلورونده^{۳۰}: در مورد وضعیت شبکه اطلاعات جمع آوری می کند.
- عامل عقب رونده^{۳۱}: از اطلاعات جمع آوری شده برای تطبیق جدول های مسیریابی با وضعیت شبکه استفاده می نماید.

²⁷ protocol
²⁸ routing
²⁹ packets
³⁰ forward
³¹ backward

هر مسیریاب جدول خاصی دارد که برای تمام مقصدهای شبکه، احتمالی را به تمام درگاه های خروجی ، نسبت می دهد. به علاوه مسیریاب شامل یک مدل آماری برای ذخیره میانگین و انحراف معیار زمان سفر به تمام مقصدها در جدول مسیریابی است. در شکل ۳ مثالی از یک جدول مسیریابی AntNet نمایش داده شده است.

پرت خروجی مقصد	میانگین زمان مسافرت				واریانس زمان مسافرت	
	n1	n2	n3	n4	زمان مسافرت	واریانس زمان مسافرت
۱	۰.۱۵	۰.۱۵	۰.۴	۰.۳	۱۲	۳
۲	۰.۱	۰.۶	۰.۱	۰.۲	۱۴	۴
۳	۰.۴	۰.۴	۰.۱	۰.۱	۱۳	۲

شکل ۳- یک جدول مسیریابی در AntNet

به طور متناوب هر مسیریاب یک مورچه جلورونده به مقصدی تصادفی می فرستد. مورچه از هر مسیریابی که می گذرد، زمان طی شده از مبدا را به همراه شماره مشخصه مسیریاب در حافظه خود-که از نوع پشته است-ذخیره می کند. سپس بر اساس احتمالات معین شده در جدول، مسیریاب بعدی را انتخاب می کند.

لزوم انتخاب احتمالی مسیر بعدی، اکتشاف شبکه و عکس العمل سریع به تغییرات مانند ازدحام^{۳۲} و از بین رفتن برخی خطوط ارتباطی است.

هنگامی که مورچه جلورونده به مقصد می رسد، تبدیل به مورچه عقب رونده شده، مسیر آمده را برمی گردد و در هر مسیریاب بر حسب اطلاعات موجود در حافظه اش و مدل آماری، جدول را به روزرسانی می نماید. اگر μ میانگین زمان سفر به گره d در گره i تا این لحظه و C عامل مقیاس^{۳۳} باشد، r' این گونه تعریف می شود:

$$\text{If } \frac{T}{c\mu} < 1 \rightarrow r' = \frac{T}{c\mu} .$$

$$\text{Else } \rightarrow r' = 1$$

ورودی جدول برای پرت f که مورچه جلورونده توسط آن از i به d رفته بود، افزایش می یابد:

$$p_{df} \leftarrow p_{df} + (1-r').(1-p_{df})$$

و برای سایر پرت ها کاهش پیدا می کند:

$$\forall j \in N_k, \quad j \neq f$$

$$p_{dj} \leftarrow p_{dj} - (1-r')p_{dj} ;$$

³² congestion

³³ Scale factor

مورچه های عقب رونده اولویت بالاتری نسبت به بسته های داده دارند، در نتیجه به سرعت توسط مسیریاب پردازش می شوند و مسیریابی را سریع تر با وضعیت شبکه تطبیق می دهند. انحراف معیار نشانگر پایداری شبکه است. مقدار بزرگ نشان دهنده وضعیت ناپایدار در شبکه است که در این حالت احتمالات کمتر تغییر می کنند.

۷- نتیجه گیری

سیستم های بسیاری در طبیعت وجود دارند که از تعدادی عامل غیر هوشمند و یا با هوش اندک تشکیل شده اند. علی رغم عدم وجود یک ساختار کنترلی متمرکز، تعاملات ساده میان عوامل باعث به وجود آمدن نوعی رفتار جمعی هوشمندانه می گردد. با الهام از این سیستم ها روش هایی برای حل مسائل بهینه سازی ترکیبی ابداع و در بسیاری موارد با موفقیت به کار گرفته شده اند. از جمله می توان مساله فروشنده دوره گرد و مسیریابی شبکه های کامپیوتری را نام برد. تحقیق بر روی این زمینه نسبتا جدید ادامه دارد و در آینده شاهد کاربردهای بیشتر این روش ها خواهیم بود.

- Marco Dorigo, Gianni Di Caro. "Ant Colony Optimization: A New Meta-Heuristic".
- Vincent Verstraete, Matthias Strobbe, Erik Van Breusegem, Jan Coppens, Mario Pickavet, Piet Demeester. "AntNet: ACO routing algorithm in practice".
- Benjamín Barán, Rubén Sosa. "AntNet Routing Algorithm for Data Networks based on Mobile Agents".