

Hangya kolónia algoritmus (Ant Colony)

Hangya kolónia algoritmus (Ant Colony)

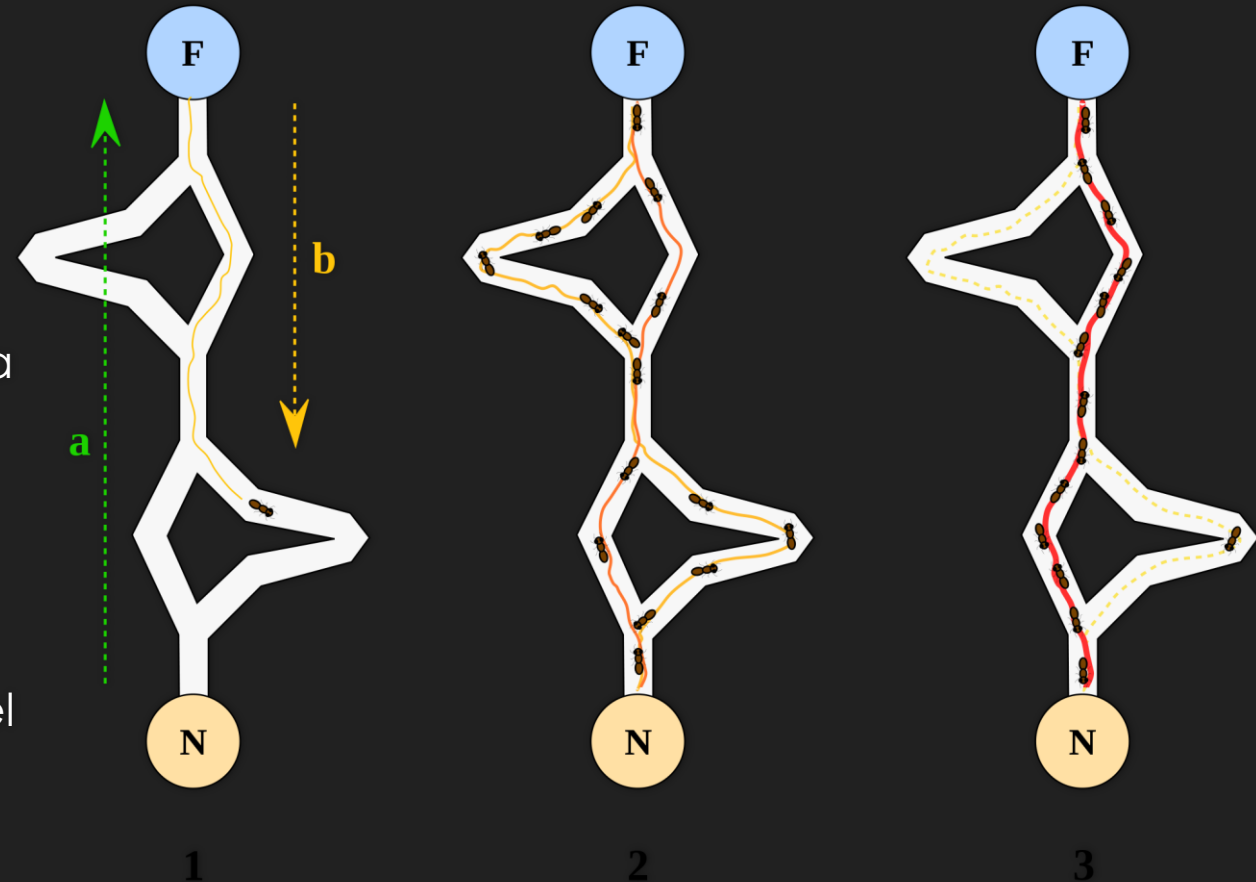
- A mesterséges intelligencia egyik ága, a raj intelligencia legismertebb módszere
- Kutatása, fejlesztése 1989-ben indult el
- Kutatók az argentin hangyák viselkedését vizsgálták (1990)
- *A legrövidebb út keresésére 1991-ben Marco Dorigo és társai alkalmazták.*
- 1996-kiadott dolgozatukban az utazóügynök problémájára alkalmazták

Hangya kolónia algoritmus (Ant Colony)

- Megfigyelhető, hogy az alacsony intelligenciájú élőlények, (méhek, hangyák) egyedileg nem sok figyelmet fordítanak egymásra.
- Az egymás közötti egyszerű kommunikáció segítségével rajként, csoportokat alkotva, igen hatékonyan dolgoznak
- Az ilyen rendszerek tervezése alapvetően eltérő a korábbi szemlélettől.
- Az egész rendszer vizsgálata helyett a hangsúlyt a kifinomult egyedekre fekteti, akik a kívánt viselkedés érdekében együttműködnek.

Hangya kolónia algoritmus (Ant Colony)

- A hangyák a táplálékhoz vezető utat egy speciális hormon, a **feromon** kibocsátásával jelölik meg
- Ezeket a feromon utakat a többi hangya érzékeli és újra „konfigurálja”
- A legtöbb hangya által bejárt útvonalakon a feromon mennyisége nagy lesz, a kevésbé használt utakon a feromon a párolgás miatt egyre kevesebb lesz.
- nagyobb feromonú utak kerülnek kiválasztásra
- Rövidebb utakon több feromon helyezhető el

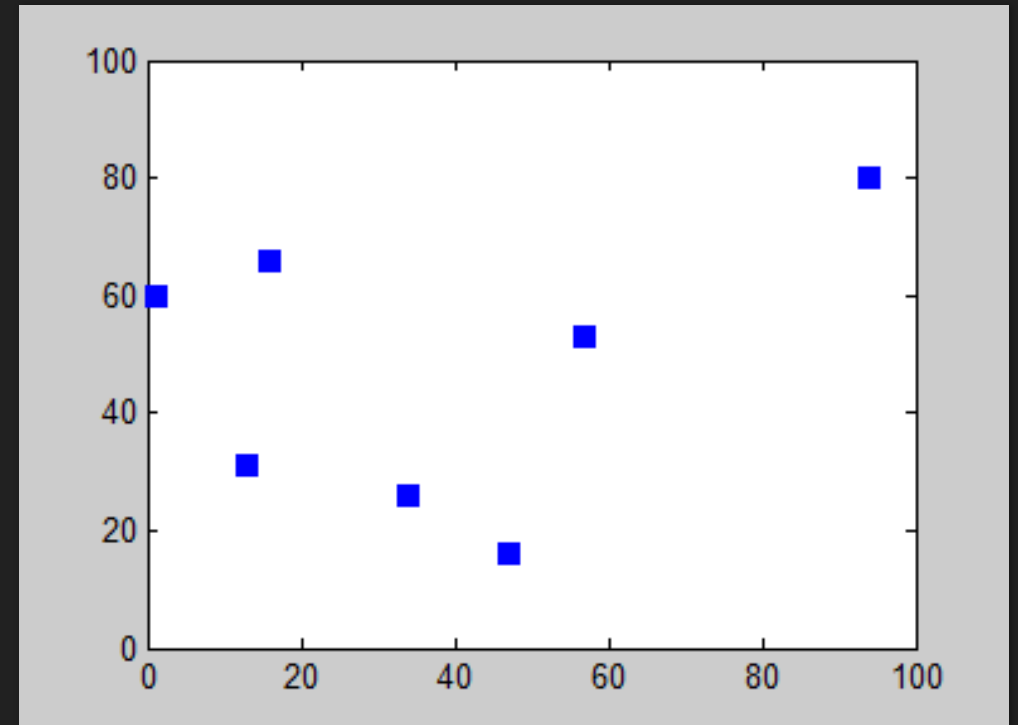


Hangya kolónia algoritmus (Ant Colony)

- A hangyakolónia algoritmussal megoldható feladatok köre az elmúlt években jelentősen kibővült. Néhány ilyen feladat a következő:
- Kommunikációs hálózatok adatforgalmának irányítása – optimális hálózati áteresztés;
- Gyártósor ütemezés;
- Járműforgalom optimalizálása;
- Gráf- vagy térképszínezés.

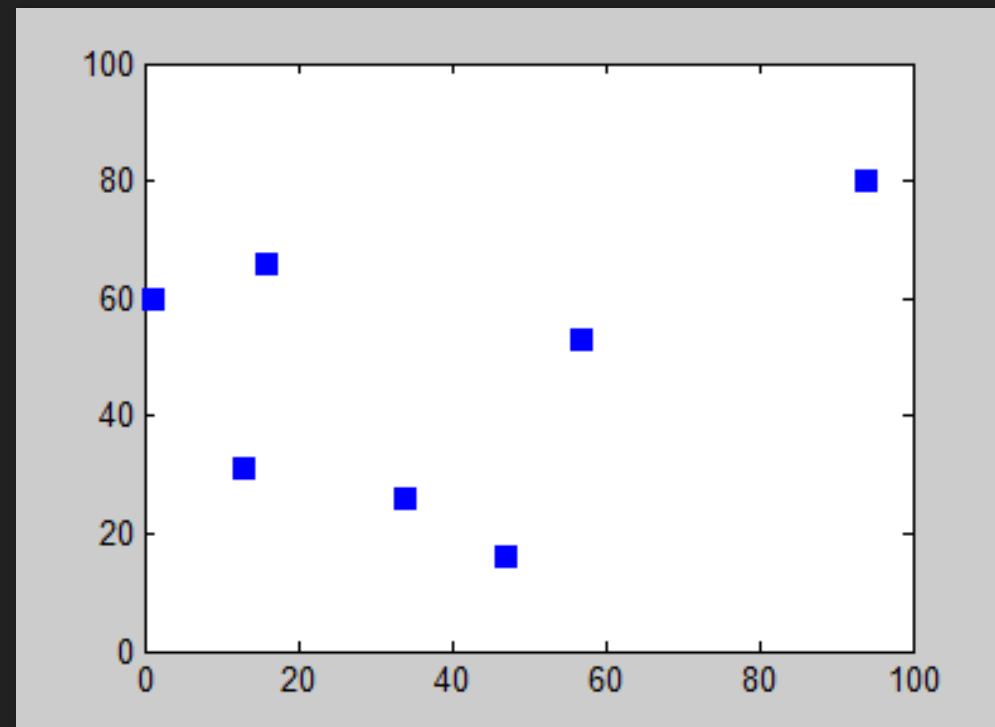
Hangya kolónia algoritmus (Ant Colony)

- A hangyakolónia algoritmusban a hangyák egy-egy ágenst reprezentálnak
- Az útvonalak száma addig növekszik, amíg az összes meglátogatandó hely bele nem kerül az útvonalba.
- minden hangya a kiindulási pontból indul.
- Az ágensek kapacitásának alapján történik a következő pont kiválasztása



Hangya kolónia algoritmus (Ant Colony)

- A hangya akkor fog visszatérni a kiinduló pontjához, ha az összes lehetséges meglátogatandó helyszínt bejárta, vagy ha a maximális kapacitását elérte.
- L- a virtuális hangya által megtett út
- A második hangya az első hangya visszaérkezése után indul,
- Az előre megadott hangyaszámig ismétlődik a program



Hangya kolónia algoritmus (Ant Colony)

- Minden hangyának minden pontot érintenie kell. A kiválasztáshoz a következő formula alkalmazható:

$$j = \arg \max \{ (\tau_{ij}) (\eta_{ij})^\beta \} \quad \text{amíg } j \notin M_k,$$

ha $q \leq q_0$, különben S ,

- ahol τ_{ij} az i jelenlegi pozíció és a j lehetséges pozíció közötti feromon mennyiség
- η_{ij} pedig i és a j közötti távolság inverze
- β a távolság fontosságát a feromon mennyiségével szemben kifejező tényező
- M_k pedig a hangya által már bejárt utat tartalmazza
- q egy véletlen változó a $[0,1]$ zárt intervallumon, q_0 pedig egy paraméter.

Hangya kolónia algoritmus (Ant Colony)

$j = \arg \max \{ (\tau_{ij}) (\eta_{ij})^\beta \} \text{ amíg } j \notin M_k,$
ha $q \leq q_0$, különben S ,

- Amennyiben minden választás megtörtént, a hangya a legnagyobb értékű élt választja a fenti egyenletből, ha csak q nem nagyobb, mint q_0 .
- Ekkor a hangya egy véletlenszerű S -t választ, mely a p_{ij} eloszlás valószínűségén alapszik, ami a magas feromon tartalmú útvonalaknak kedvez

Hangya kolónia algoritmus (Ant Colony)

Az út megtervezése

- Kezdetben minden k hangya egy véletlenszerűen választott városban van.
- Egy k hangya minden konstrukciós lépésben alkalmaz egy valószínűségen alapuló döntési szabályt.
- A valószínűség, amivel egy k hangya, ami jelenleg az i városban van, a j városban megy át az algoritmus t -edik iterációjában:

$$p_{ij}^k(t) = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_j^k} [\tau_{il}(t)]^\alpha * [\eta_{il}]^\beta}$$

$$\text{Ha } j \in N_j^k$$

Hangya kolónia algoritmus (Ant Colony)

$$p_{ij}^k(t) = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_j^k} [\tau_{il}(t)]^\alpha * [\eta_{il}]^\beta}$$

$Ha j \in N_j^k$

Az α és β paraméterek meghatározzák, hogy a feromonösvénynek és a heurisztikus információnak mekkora legyen a hatása.

- N_j^k pedig azon városok halmaza, melyeket a k -adik hangya még nem látogatott meg.
- Ha $\alpha=0$, akkor a közelebb lévő városokat választják a hangyák
- Ha $\beta=0$, akkor csak a feromon alapján folyik a keresés.
- Ez gyors stagnáláshoz vezet,
- Stagnálás: ugyanazt a feromon ösvényt követi és ugyanazt a megoldást hozza létre.

Hangya kolónia algoritmus (Ant Colony)

A feromonok frissítése

- először egy konstans értékkel csökkentjük a feromon mennyiséget minden élen, majd a hangyák az általuk látogatott élekre feromont helyeznek:

$$\tau_{ij}(t+1) = (1-p) * \tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t)$$

- p megakadályozza a feromon felhalmozódását így az algoritmusnak lehetősége van elfelejteni a régebbi rossz döntéseket.
- A nem választott él feromon erőssége exponenciálisan csökkenni fog.
- A k -adik hangya által elhelyezett feromon mennyiség:

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{1}{L^k(t)} & \text{ha az } (i,j)\text{ élen átment a } k - \text{adik hangya} \\ 0 & \text{különben} \end{cases}$$

Hangya kolónia algoritmus (Ant Colony)

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{1}{L^k(t)} & \text{ha az } (i,j)\text{élen átment a } k - \text{adik hangya} \\ 0 & \text{különben} \end{cases}$$

- $L^k(t)$ a k -adik hangya útjának hossza.
- Az egyenlet szerint annál jobb a hangya útja, minél több feromon van az azt alkotó éleken.
- Általánosságban, azok az élek, melyeken sok hangya megy keresztül és, amelyek rövidebb utakhoz tartoznak, azokat az algoritmus későbbi iterációiban a hangyák nagyobb valószínűséggel fogják választani.

Hangya kolónia algoritmus (Ant Colony)

