

Speciális útkereső algoritmusok

Kelemen Zsolt

Röviden

Útkeresési algoritmusok:

- „gráfok” keresés
- „gráf nélküli” keresés
- „ant search” (hangya alapú keresés)

Általában

- a mesterséges intelligencia egyik legrégebbi területe az útkeresés
- gráfokkal modellezik a világot
- a gráfot minden egyes problémára külön meg kell adni
- hátrányok: - időigényes
 - mi van ha ismeretlen a környezet?

Feladat

- 2 vagy több dimenzió - még 2,5 D vagy 3D lehet, ennél több nem ☺
- két, koordinátával megadott pont között utat találni
- pl. robot a Marson

Algoritmusok célja

nem egy konkrét út megtalálása, hanem olyan két vagy akár több dimenziós sáv kijelölése, amelyben a célpont felé navigáló objektumok már – csak a lokális környezetet vizsgáló algoritmussal is - komolyabb nehézségek nélkül haladhatnak előre, még akkor is ha közben ki kell kerülniük néhány akadályt (például társakat)

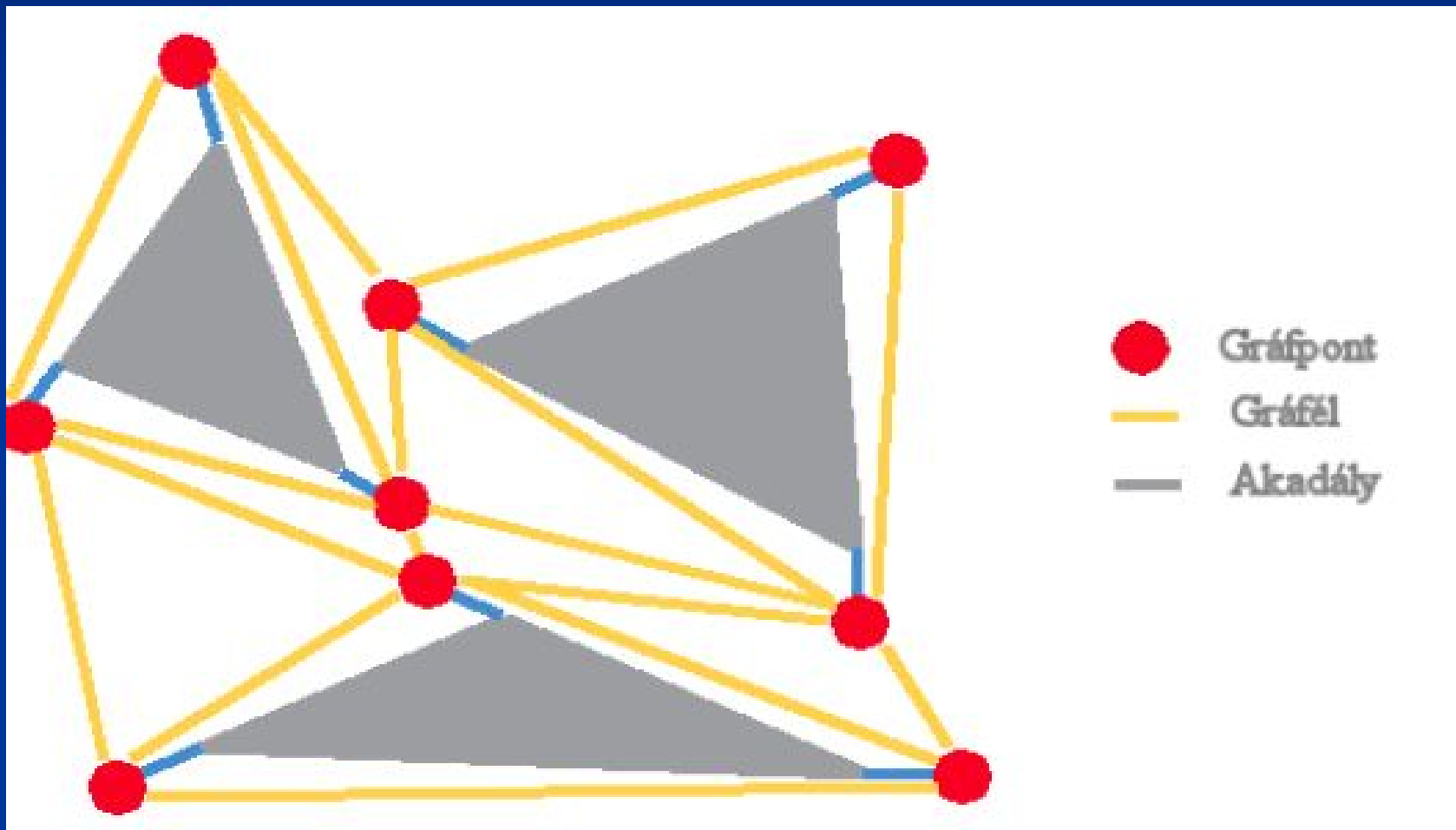
Térfelosztó algoritmusok

- látható csomópontok (PoV)
- konvex cellák (C-Cells)
- maximális területekre bontás
- kvadratikus fa
- hullámfront-terjesztés (wavefront expansion)

Látható csomópontok (PoV)

- az akadályok „sarkai” a lényegesek
- a gráf pontjai az akadályok sarkaitól adott távolságra vannak, épp annyira, hogy a csomópontokon áthaladó ágens ne ütközzön bele az akadályba
- élek azon pontok között, melyek „látják” egymást
- előny: ha a cél közel van a falhoz, rövid utat talál
- hátrány: sok gráfpontra van szükség

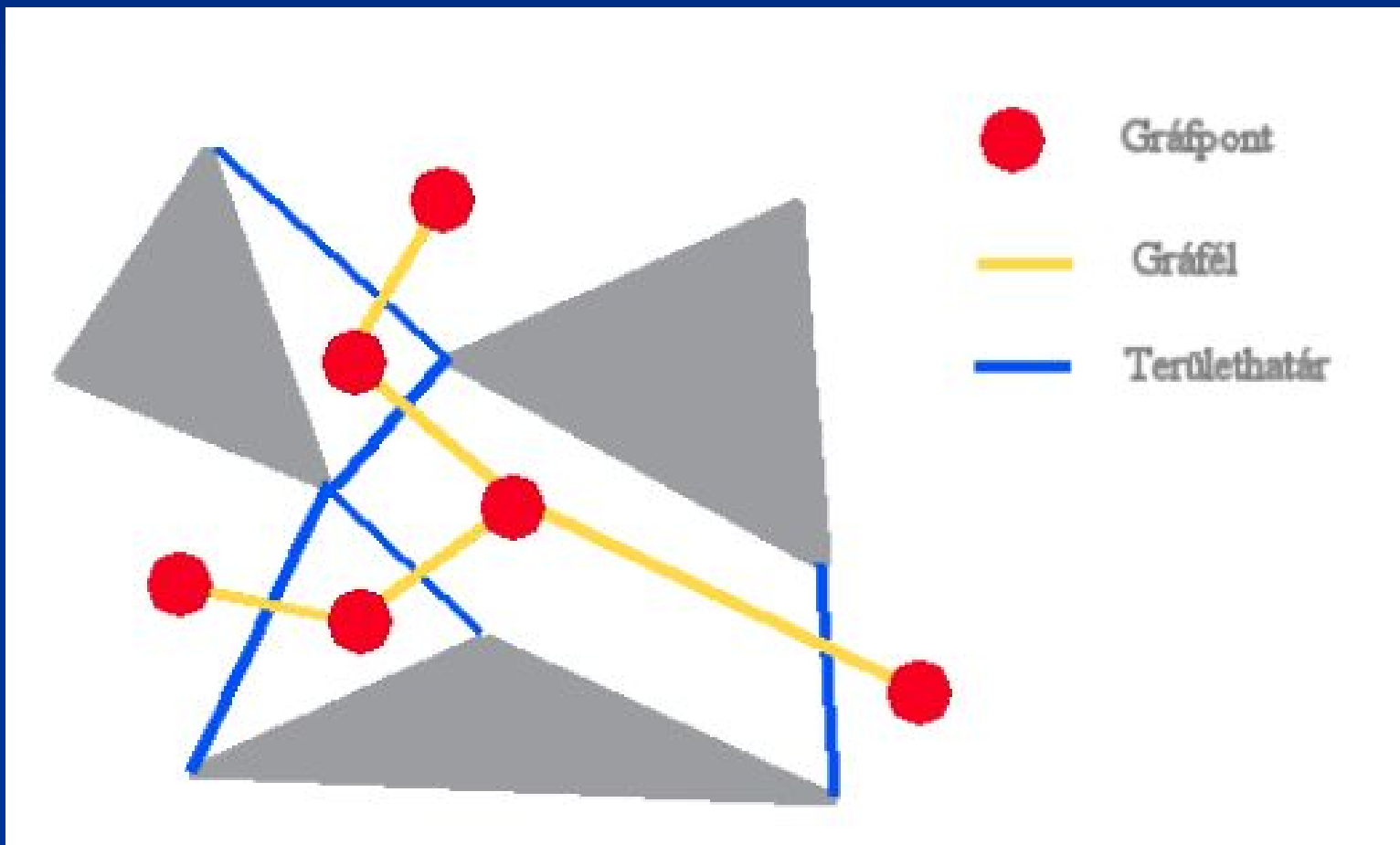
Látható csomópontok (PoV)



Konvex Cellák (C-Cells)

- az akadályok sarkait csak az onnan látható legközelebbi akadálysarkokkal köti össze, az így kialakult szakaszok nem élek lesznek, hanem területhatárok
- minden területhez egy gráfpontot rendelünk, míg az éleket a szomszédsági viszonyok alapján húzzuk be
- kevés gráfpont, de az utvonalak sem olyan rövidek

Konvex Cellák (C-Cells)



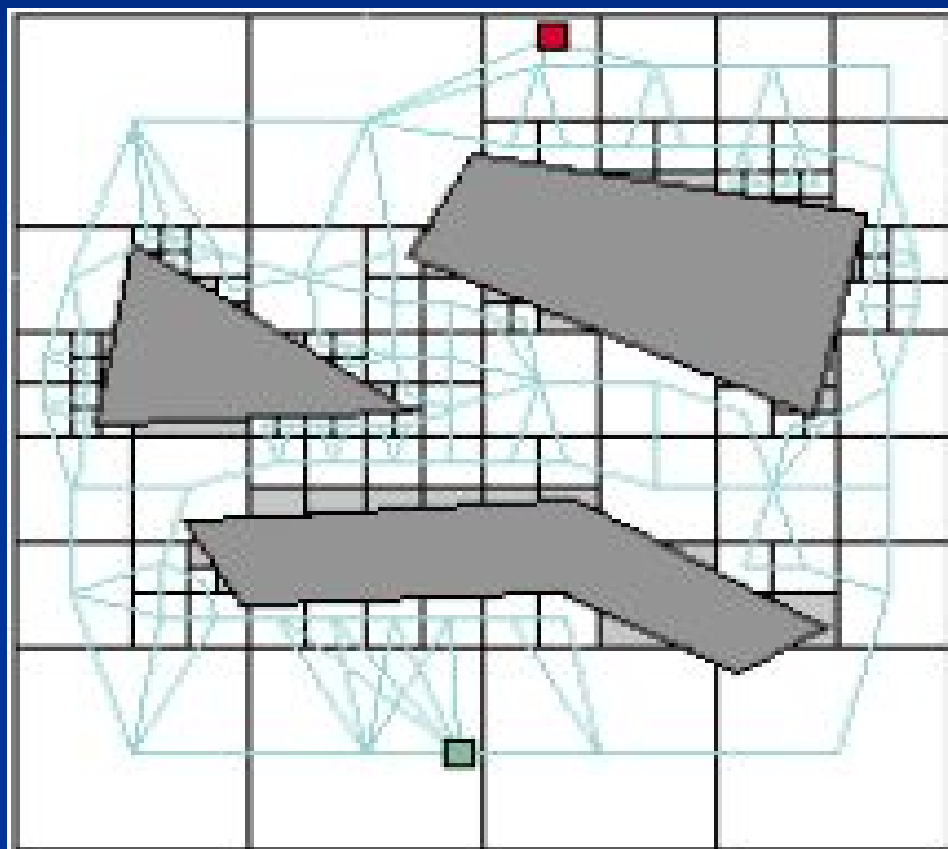
Maximális területekre bontás

- azonos méretű darabokra vágja a síkot
- az akadályok konvex sarkait vetíti a legközelebbi akadályra
- az így kapott vágási vonalat összeveti a C-Cells eredményével és a 2 szakasz közül a rövidebbet tartja meg
- az így kapott területekhez gráfpontokat rendelünk, amelyet a szomszédsági viszonyok alapján kötünk össze éllel

Kvadratikus Fa

- négyzetekre bontjuk a síkot
- ha a felbontás nem elég pontos, akkor egy-egy négyzetet negyedakkora területű négyzetekre vágunk
- ezt a lépést addig ismételjük, amíg a hiba kellően kicsi nem lesz
- hátrány: ferde vonalat nem lehet jól követni
- előny: alkalmazható 3D-ben is → Oktális Fa

Kvadratikus Fa



Hullámfront terjesztés

- tároljuk a síkot egy tömbben (a térképet kvantáljuk kis négyzetekre) és mindegyikhez hozzárendelünk egy térerőt
- Elindulunk egy cellától, 0 értéket rendelünk hozzá, majd az összes szomszédos cellájának 1-et adunk, az ezekkel szomszédosoknak 2-t, ...
- hibája: általában nem a legjobb utat adja, mert az átlósan elhelyezkedő mezők 1,4-szer akkora távolságlépést jelentenek mint a többi

Hullámfront terjesztés, javítások

I.

- a tömböt a távolságmérést figyelembevéve töltjük ki

II.

- hullámfront-terjesztést az akadályokból kiindulva végezzük → mindegyiknek saját erőtere
- ha egy cellába több különböző tér értéke kerülne, akkor azt megjelöljük és nem is lépünk belőle tovább
- olyan térképet kapunk, amin az akadályoktól azonos távolságra lévő útvonalak vannak megjelölve

Hullámfront terjesztés

	2	2	2	2	2	
	2	1	1	1	2	
	2	1	1	1	2	
	2	1	1	1	2	
	2	2	2	2	2	

	2,8	2,4	2	2,4	2,8	
	2,4	1,4	1	1,4	2,4	
	2	1	1	1	2	
	2,4	1,4	1	1,4	2,4	
	2,8	2,4	2	2,4	2,8	

	■	1	2	3	4,3	
	1	1	2	3,2	2	
	2	2	2,2	3,1	1	
	3	3,3	3,2	1	■	
	4,4	3	2	1	1	

Gráfépítés keresés közben

- az előbbi módszerek közös hátránya: szükség van a teljes világ ismeretére
- olyan eljárás kell, amely csak a keresés által bejárt területeket alakítja diszkrétte, sőt a keresés közben végzi ezt az átalakítást
- szükség van egy gráfra, amit keresés közben építünk fel, ennek a pontjait véges kiterjedésű testekkel helyettesítünk

(folytatás)

- vigyázni a méret megválasztásánál
- legegyszerűbb a pontokhoz gömböket rendelni
- gömbökkel nem lehet a teret hézagmentesen lefedni, de a hiba csökkenthető, ha megengedjük, hogy a gráfponatok kismértékben átlapolódjanak
- ha nagyon precízek akarunk (vagy kell!) lenni, akkor más testeket használunk (pl. téglatest)
→ az algoritmus bonyolultabb lesz

Akadályok és falak tárolása

- az akadályokat véletlenszerű mintavételezéssel írjuk le, mivel az algoritmus működéséhez ugyanis általában elég csak akadályonként néhány pont ismerete
- ezekhez a pontokhoz is méretet rendelünk → minél nagyobb a pontméret, annál nagyobb lesz a minimális átjáró, amit megtalálunk

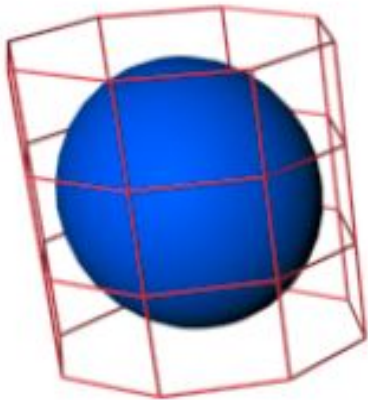
(folytatás)

- ha nem tudunk semmit a környezetről, akkor célszerű az egység orientáció-független méretének (legkisebb befogadó gömb átmérője) legfeljebb felét használni → a legalább kétszer ekkora lyukakat biztosan tudja detektálni
- csapdák mérete

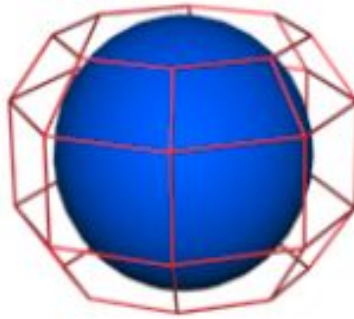
Keresés menete

- a kezdőpont lesz az első gráfpont
- minden szomszédot azonos méretűnek veszünk
- kiszámítjuk, hogy a gömb sugarával megegyező sugarú kör körül milyen távolságra kell elhelyezni a pontokat
- további alakítások segítségével behúzzuk a szomszédokat a gömb felületére

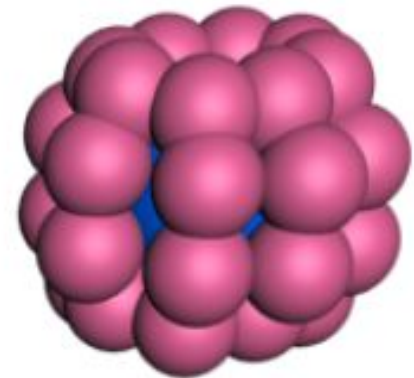
Keresés menete



A méretből kiszámoljuk
a gráfponatok síkbeli helyét,
ebből kapjuk a henger-
formát



A hengert gömbszerűvé
alakítjuk a vektorok
normalizálásával

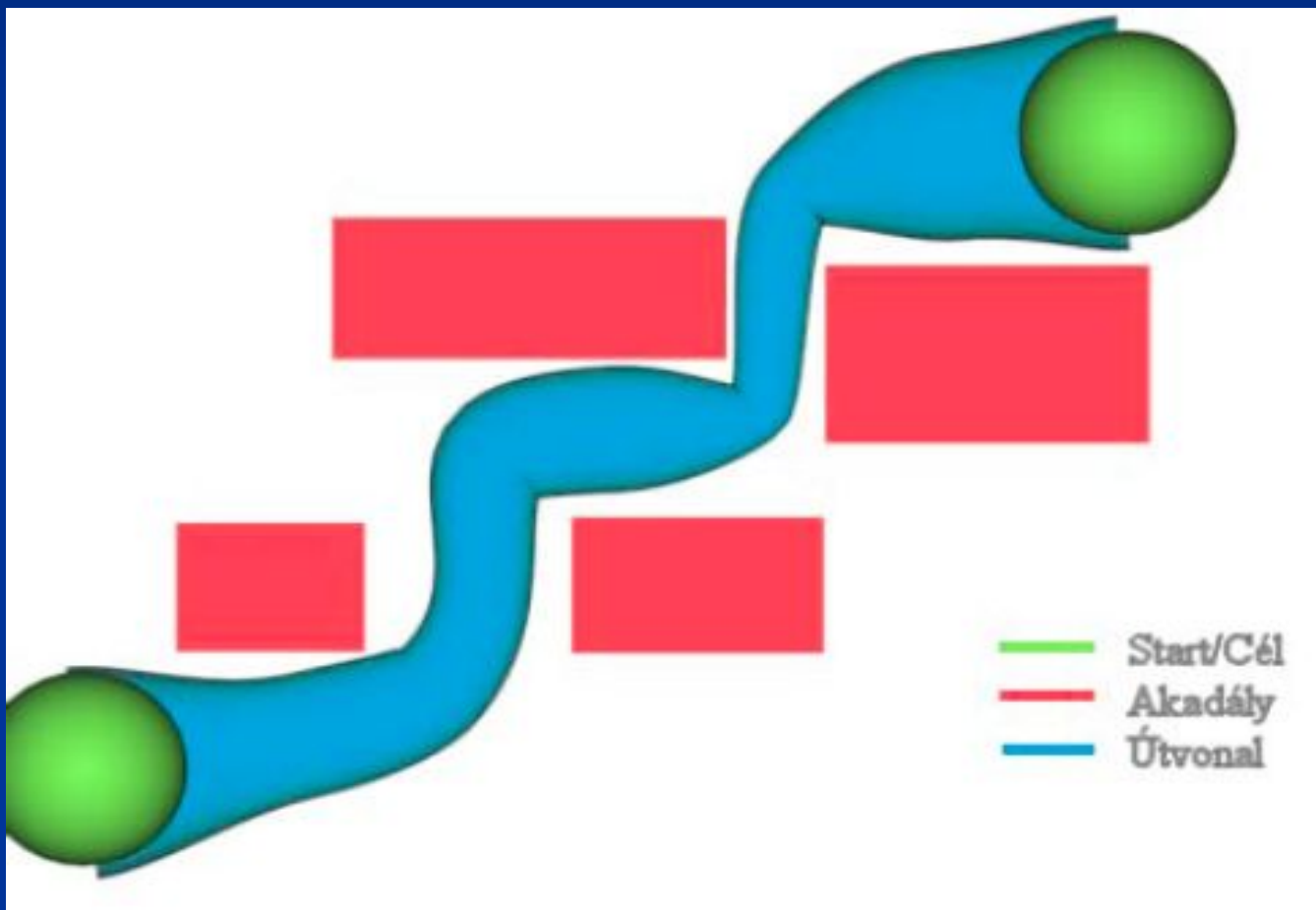


A gráfponatok
elhelyezése

Előnyök - Hátrányok

- az útvonal nem darabos, mert a keresés eredménye nem egy út, hanem egy bizonyos szélességű sáv
- az eljárás 3D-ben is működik
- alacsony sebesség: a klasszikus módszereknél ember tervezi a gráfot → kevesebb pontot fog tartalmazni

Az eredmény



Ant Search

- egyes célpontok között nagyon sok gráfpontra vezethet át az út
- négyzetrácsos térkép
- algoritmus: a térkép távolabbi pontjai között útvonalkeresést hivatott elvégezni
- a keresési térben elég nagy az élek száma → magát a végeredményt is széles utak képezik

A természetben

- az egyedeknek nem kell ismerniük az egész térképet, csak a közvetlen környezetüket, ennek ellenére idővel globálisan jó megoldást találnak
- feromon nevű hormon
- ha egy hangya feromon-nyomot (vagy egy a sajátjánál erősebb feromon-nyomot) talál, akkor nagy valószínűséggel elkezdi követni azt
- sok hangya: a kezdeti véletlenszerű bolyongás után kialakulnak azok az útvonalak, amelyeket a jelentős részük használ

A természetes módszer szimulációja

Sok, egyszerű „működésű” hangyára van szükség:

- minden hangya egyszerre egy négyzetrácsnyit tud lépni a 4 szomszédos mező egyikére
- minden h. minden lépésben feromon-nyomot húz
- a továbbhaladási irányukat a környezetükben észlelt nyomvonalak erősségének függvényében, a véletlen bevonásával választják
- előnyben részesítik az egyenes vonalú mozgást

(folytatás)

- a visszafele irányt kisebb valószínűséggel válasszák
- nem mehetnek át az akadályokat jelentő falakon

A nem használt nyomvonalaknak idővel „el kell illaniuk”, hogy nem zavarják a keresést

Bővítési lehetőségek

- két új hormon bevezetése, az egyik a céltól, a másik a bolytól távolodva gyengül

Ez még kevés, ha sok hangya van egy kisebb területen, akkor jó eséllyel beszorulnak, mivel folyton egymásnak fognak ütközni → bővíteni kell a hangyákat vezérlő szabályokat:

- csak akkor húzzanak nyomot, ha haladnak is
- mászhassanak át egymáson

(folytatás)

- az előző két módszer nem hatékony, abban az esetben a beszorulnak a hangyák valahová
- a „Stacked Canyon” (torlódási hormon) bevezetése: ha a hangyák ütköznek egymással kibocsájtják ezt a hormont, amit utána minden hangya igyekszik kerülni

Ennek a hormonnak hamar el kell utána illania.

További bővítések

A hangyák által egyénileg készített mérések:

- elmozdulás: minden h. kiszámolja, hogy n lépés alatt milyen messzire jutott el
- hormonszint változása: az aktuális céltól távolodva gyengülő hormon mennyiségének a változása n lépés alatt
- lépésszám: hány lépést tettek meg a legutolsó célpont elérése óta

Magyarázat

Ha egy hangya egy bizonyos lépésszám alatt nem éri el a célt, akkor:

- nagyobb az út hossza mint a lépésszámkorlát
- rossz irányba ment
- „beszorult” valahová

„Beszorult” valahová

A helyzet megoldása:

- nem teszünk semmit
- ha egy hangya túllépte a megengedett lépésszámot, akkot már nem húz feromonnyomot maga után

Felhasználás

- az algoritmus eredménye két hormon mennyiségi térképe: útvonalak és torlódási pontok
- a modern RTS (RealTime Strategy) játékok használják fel
- Bővíthető: még egy hormon bevezetésével megoldható, hogy bizonyos területeket kerüljön el (pl. az ellenség bázisa)

Elemzés

- jól skálázható
- jól hangolható a szabad erőforrások mennyiségéhez (hangyszám)
- széles utak esetén az eredményben szereplő nyomvonal is szélesebb
- a torlódási hormon szolgáltat fontos információkat

Hátrányok

- lassan stabilizálódik a megoldás
- sok erőforrást igényel, mivel hosszasan kell futtatni a háttérben a szimulációt (az RTS játékok egyenlőtlen erőforrásigénye miatt nem jelent gondot)
- kell egy magasabb szintű réteg, amely a paramétereket hangolja

Köszönöm a figyelmet!