FIATAL MŰSZAKIAK TUDOMÁNYOS ÜLÉSSZAKA

Kolozsvár, 2004. március 26-27.

SPECIÁLIS ÚTKERESŐ ALGORITMUSOK ÉS ALKALMAZÁSAIK MODERN JÁTÉKPROGRAMOKBAN

Csorba Kristóf, Tódor Balázs

In this article we present two special path finding algorithms to be used in continuous environments, which are very different from the classic methods and can be useful in the modern game artificial intelligence programming. The first method described extends the A* search to work without a (previously defined) search graph, while the latter uses the ant search to find paths. The goal of both is to find a wide path that the agents can track with only local navigation algorithms even if formerly unknown (e.g.: dynamic) obstacles appear in the way.

Ebben a cikkben két olyan speciális útkereső eljárást fogunk bemutatni, melyek jelentősen eltérnek a klasszikus megoldásoktól, és igen hasznosak lehetnek többek között a modern játékprogramokban felmerülő különböző problémák megoldására. A „gráf nélküli” keresés és a hangya alapú keresés futása közben térképezi fel a keresési teret, majd a terepviszonyoknak megfelelően, a lehetőségekhez mérten széles utakat jelöl ki, melyben a célpont felé navigáló objektumok már egy - csak a lokális környezetet vizsgáló algoritmussal is - komolyabb nehézségek nélkül haladhatnak előre még akkor is, ha közben ki kell kerülniük néhány kisebb akadályt (például társakat).

1. A „gráf nélküli” keresés

A mesterséges intelligencia egyik legrégebbi területe az útkereséshez és útvonaltervezéshez használható algoritmusokkal [5] foglalkozik. A klasszikus eljárások a világot gráfokkal modellezzik, amiket általában emberek készítenek - bár a feladat egyszerű, gyakran mégis időigényes. Ha elvonatkoztatunk a játékprogramozástól, akkor könnyen találhatunk olyan problémakört is, amikor egyszerűen nem alkalmazhatunk embereket a megoldáshoz, például, ha a feladat egy ismeretlen környezetben navigáló robot tervezése.

Természetesen már korábban is születtek keresési gráfot készítő eljárások. Ilyenek például a látható csomópontok (Points of Visibility, [1, 6, 7]), a konvex cellák módszere (C-Cells, [6, 3]), vagy maximális területekre bontás (Maximum-Area Decomposition, [2, 6]), és az általános hengerek algoritmus (Generalized Cylinders, [6]). A fentiektől különbözik a kvadratikussal ill. az októális fával (quadtree, octtree, [6]), mivel ezek lehetővé teszik a hierarchikus útkereső módszerek alkalmazását [4]. Az útkereséshez nem feltétlenül szükséges a folytonos keresési teret diszkrét alakítani. Elég, ha

az akadályokhoz taszító erőt rendelünk, a célponthoz pedig vonzót. Erre épül az erőter módszere (potential fields, [8, 3]). A lokális minimumok problémájának megoldására több megoldás is született már, például a hullámfront-terjesztés (Wavefront expansion, [8]), ami azonban csak diszkrét keresési tereken (egyenközű rácson) működik. A fenti módszerekhez szükség van a teljes világ ismeretére. Ennek kiküszöbölésére a problémát máshogy közelítjük meg és csak a bejárt területeket alakítjuk keresés közben diszkrétte. A gráfot keresés közben építünk fel: gráfpontokhoz véges kiterjedésű testeket (hengereket, 2D-s esetben köröket) rendelünk (ld. **1. ábra**). Az ágens sugárkövetéses távolságméréssel érzékeli a környezetét. A távolságinformációból és a saját (ismert) helyéből kiszámítja az akadályok helyeit, majd ezeket egy listában tárolja. Az **1. ábrán** a világos pontok a már tárolt, a sötétek pedig az éppen látott akadálypontok.

1. ábra - A szimuláció

2. ábra - Az átfedés

$$O_{N_1N_2} = \frac{d(P_1, P_2)}{r_1 + r_2} \tag{1}$$

A keresés elején az ágens helyére rakjuk le az első gráfpontot. Az ezután következő lépésekben az A* szabályai szerint kiválasztjuk a legjobb heurisztikus értékű pontot és annak a szomszédait berakjuk a gráfba. Az útvonal garantált járhatósága érdekében a köröknek átfedésben kell lenniük, ld. **2. ábra** ill.(1).

3. ábra - A lefedetlen terület (csíkozott)

4. ábra - A lefedetlen terület "szélessége"

$$d_A \leq 2O_{\max}r \tag{2}$$

Mivel a gráfpontokat reprezentáló körök nem metszhetik a falakat, ezért mindig marad valamennyi lefedetlen terület (**3. ábra**) a keresési gráf és a falak között, melynek maximális szélessége d_A (ld. **4. ábra**, ill. (2)). Az eljárás akkor és csak akkor teljes, ha a környezetben található legszűkebb átjáró is

legalább $2 \cdot 2O_{\max} r + 2r$ széles, mivel így „belefér” egy kör és mellé a két oldalra a maximális szélességű lefedetlen terület is.

2. A hangyás keresés

A természetben a hangyák által alkalmazott keresési módszer egy feromon hormon folyamatos kibocsátásán alapszik, melyet a hangyák később követnek. A sokszor használt útvonalak megjelölése a kezdeti bolyongás után globálisan is elfogadható megoldáshoz vezet, mivel a rövidebb utat többször lehet megtenni ugyanannyi idő alatt, így azon a feromonkoncentráció is magasabb.

A mi módszerünk alapvetően eltér a szakirodalomban [9]-[14] olvasható ant search alkalmazásoktól, mivel sokkal inkább térképen keresünk és nem gráfban, mint például az utazóügynök probléma esetében. Egy ilyen térkép gráfként rengeteg élt tartalmaz. A mi alkalmazásunkban számos ponton ki kellett egészítenünk a módszert, hogy hatékony legyen. Ilyen kiegészítés többek között a torlódási hormon, a hangyákra lokális és globális statisztikák.

Ennek a módszernek az alkalmazásához sok, egyenként egyszerű hangyára van szükségünk.

A keresés sebességét nagymértékben növeli, ha a hangyák tudják, hogy légvonalban merre kell keresni a célpontot. Ez azonban növeli annak az esélyét is, hogy zsákutcákba kerülnek, ahonnan a sok bentrekedő hangya miatt folyamatosan növekedő feromonkoncentráció miatt nehezen jutnak ki.

Erre egy lehetséges megoldás egy ún. „torlódási hormon” bevezetése, melyet a hangyák egymással való ütközéskor (tehát elsősorban torlódás esetén) bocsátanak ki és ezzel párhuzamosan folyamatosan igyekeznek elkerülni az így megjelölt területeket.

További javítást érthetünk el, ha a hangyák folyamatosan számontartják, hogy hány lépést tettek meg a legutolsó célpont elérése óta. Ha ez az érték egy bizonyos (átlagérték alapján meghatározott) küszöb fölé emelkedik, akkor a hangya jó eséllyel rossz irányba ment, ezért felfüggeszti a feromonkibocsátást és elindul visszafelé a legutóbb érintett célponthoz. (Feltételezzük, hogy egy célpont elérése után a hangyák mindig új célpontot kapnak.)

Azt, hogy a keresés mikor ér véget, a legkönnyebben az egységnyi idő alatt célponthoz érő hangyák számával tudjuk meghatározni. Ha ez egy a feladat méretéhez viszonyítottan magas értéken stabilizálódik, akkor a hangyák már a megtalált utat követik. Ekkor a megoldást a feromonkoncentráció térbeli eloszlása adja.

Az eredményül kapott feromontérkép formájában olyan utakat kapunk, melyek szélesebbek egy sima élsorozatnál, így az út elhagyása nélkül végrehajthatók a dinamikus térben szükséges kitérés manőverek a navigáló ágensek között.

A torlódási hormon jelenléte olyan átjárókat jelöl, melyek áteresztő képessége kritikus, amennyiben sok ágens kíván egyszerre átjutni rajta. Ilyenkor előre lehet tervezni és az ágensek egy csoportját eleve más útvonalra lehet terelni. Így elkerülhetők a torlódások, melyek feloldása ágensnavigációs szempontból igen komoly feladat.

A módszer előnyei:

- A hangyák számának módosításával a keresés pillanatnyi erőforrásigénye (de ezzel együtt természetesen hatásfoka is) jól skálázható a rendelkezésre álló erőforrások mennyiségének és a feladat komplexitásának függvényében.
- Új hormon bevezetésével képessé tehető a keresés bizonyos területek elkerülésére vagy előnyben részesítésére.
- „Széles utakat” ad eredményül, mely egyszerűsíti az ágensek navigációját.
- A torlódási hormon lehetővé teszi a torlódások elkerülését.

Hátrányok

- A hagyományos útvonalkereső algoritmusokhoz képest (pl. A*) lassú. (Cserébe azonban sokkal több információt nyújt.)
- Sok erőforrást igényel, mivel hosszan kell futtatni a háttérben a szimulációt.

A módszer többek között modern stratégiai játékprogramokban lehet hasznos, ahol a fenti hátrányokat a játékok jellege miatt (kezdetben sok szabad erőforrás, sok idő a háttérben futtatáshoz) áthidalhatók.

- [1] Game Programming Gems II. (Charles River Media), 3.10. (324.-329.)
- [2] René Greulich: Diplomarbeit („Charakterverhalten in Computerspielen: Pathfinding”), <http://www.gm.fh-koeln.de/~faeskom/diplom/diplomgreulich.pdf>
- [3] Yong K. Kwang, Narendra Ahuja: „Gross motionplanning—a survey” (ACM Computing Surveys, ISSN: 0360-0300, 1992.)
- [4] Marco Pinter: „Towards More Realistic Pathfinding”, www.gamasutra.com, 2001. 03. 14., (http://www.gamasutra.com/features/20010314/pinter01_.htm)
- [5] Stuart J. Russell - Péter Norvig: *Mesterséges intelligencia modern megközelítésben*, Panem-Prentice Hall, 1995.
- [6] W. Bryan Stout: „Smart Moves: Intelligent Pathfinding”, www.gamasutra.com, 1999. 02. 12., (http://www.gamasutra.com/features/19990212/sm_09.htm)
- [7] *Optimising A** <http://www.ur.co.nz/urcorp/default.asp?data=242>
- [8] Stefan Baert: „Motion Planning Using Potential Fields”, www.gamedev.net, <http://www.gamedev.net/reference/articles/article1125.asp>
- [9] Hozefa M. Botee and Eric Bonabeau: „EvolvingAnt Colony Optimization” (Santa Fe Institute, 1999.) <http://www.santafe.edu/sfi/publications/wpabstract/199901009>
- [10] Simon Kaegi and Tony White: „UsingLocalInformation To Guide Ant BasedSearch” (School of Computer Science, Carleton University, 2003.) <http://www.scs.carleton.ca/~arpwhite/documents/lbt-iea-aie-2003-final.pdf>
- [11] Marc Reimann: „Ant Based Optimization in Goods Transportation” <http://www.ads.tuwien.ac.at/teaching/ws02/EvolAlg/ACO-Teil2.pdf>
- [12] David Gordon: „Collective Intelligence in SocialInsects” (AI depót) <http://ai-depot.com/Essay/SocialInsects-Swarm.html> (Partiele swarm optimization)
- [13] „Ant Colony Optimization”, <http://iridia.ulb.ac.be/~mdorigo/ACO/publications.html>
- [14] 3. ANTS 2002 (Brussels, Belgium) <http://www.informatik.uni-trier.de/~ley/db/conf/antal/antal2002.html>

Csorba Kristóf informatikushallgató, kristof@impulzus.sch.bme.hu

Tódor Balázs villamosmérnök-hallgató, balash@freemail.hu

Budapesti Műszaki és Gazdaságtudományi Egyetem, Villamosmérnöki Kar, Méréstechnika és Információs Rendszerek Tanszék, 1117 Budapest, XI. Magyar tudósok körútja 2. Informatika ép. E szárny, IV. em. E444