آینده‌پژوهی دفاعی، سال پنجم، شماره ۱۶، بهار ۱۳۹۹

تاریخ دریافت: ۱۳۹۹/۰۲/۰۷ تاریخ پذیرش: ۱۳۹۹/۰۵/۱۳

از صفحه ۷ تا صفحه ۲۸

کاربرد الگوریتم هندسه زبان‌شناختی در حل مسأله تعقیب و گریز بر روی گراف با اضافه نمودن شرایط واقعی محیط بازی جنگ

احمد الهیاری*[۱]

الیپس مسیحی[۲]

چکیده

امروزه تنوع و گستردگی مسائل در شاخه‌های گوناگون علوم تا حد بسیار زیادی افزایش یافته است و یافتن پاسخ برای چنین مسائلی در مدت زمان کم، چالشی بسیار اساسی محسوب می‌شود. استفاده از هوش مصنوعی می‌تواند فرآیند حل مسائل پیچیده را تا حد قابل قبولی سرعت بخشیده و زمان پاسخ‌دهی را به نحو محسوسی کاهش دهد. مسأله تعقیب و گریز از جمله مسائلی است که می‌تواند دارای سطح بالایی از پیچیدگی باشد. از جمله عوامل ایجاد پیچیدگی می‌توانیم به تعداد عوامل درگیر، برد میدان دید اعضاء و موانع موجود در زمین بازی اشاره کنیم. تاکنون الگوریتم‌های گوناگونی برای حل مسأله تعقیب و گریز ارائه شده که هر یک دارای نقاط ضعف و قوت خود هستند. در این مقاله با استفاده از الگوریتم هندسه زبان‌شناختی، به بررسی بازی تعقیب و گریز بطور خاص در یک مسأله با ابعاد زمین بازی ۹×۹ و برای بررسی تعمیم کارایی الگوریتم در مسائلی با ابعاد مختلف پرداخته شده است. نشان داده شده که این رهیافت می‌تواند بیش از ۹۰٪ بهبود در سرعت پاسخ‌یابی ایجاد کند. در این مقاله عوامل مؤثر در واقع‌گرایی بازی تعقیب و گریز با دقت بیشتری در هندسه زبان‌شناختی بررسی می‌شوند و در نهایت با ساده‌سازی فضای مسأله به تعدادی زیر فضا که در آن گشت‌های حرکت هر عامل بازی مشخص است، اقدام به حل مسأله می‌کنیم. نشان می‌دهیم علی‌رغم پیچیده‌تر شدن فضای مسأله، الگوریتم هندسه زبان‌شناختی بهبودی در حدود ۹۱٪ نسبت به سایر الگوریتم‌ها ایجاد می‌کند.

واژه‌های کلیدی:

مسأله‌ی تعقیب و گریز، هندسه‌ی زبان‌شناختی، مسأله‌ی دزد و پلیس، الگوریتم جستجو، هرس آلفا-بتا.

---

[۱]. دانشجوی دکتری مهندسی صنایع، دانشکده مهندسی صنایع، دانشگاه آزاد اسلامی واحد تهران جنوب

[۲]. استادیار دانشگاه پلی تکنیک کالیفرنیا در پومونا، ایالات متحده آمریکا

* نویسنده مسئول: Email:st_a_allahyari@azad.ac.ir

مقدمه

بازی‌های جنگ[1]، دسته‌ای از بازی‌های تحلیلی هستند که جنبه‌هایی از میدان نبرد، مانند تاکتیک، عملیات و یا سطح استراتژی را شبیه‌سازی می‌کند. از آنها برای آزمودن مفاهیم نبرد، تمرین و آموزش فرماندهان و تحلیلگران، کشف سناریوها و چگونگی ارزیابی نقشه‌ها و گزینه‌های مؤثر بر روی عملیات جنگی استفاده می‌شود. بدین صورت با استفاده از بازی‌های جنگ، فرماندهان نظامی می‌توانند بدون متحمل شدن تلفات و خسارات مادی و معنوی، سناریوهای مختلف نظامی را مورد ارزیابی قرار دهند تا بهترین آنها را برای استفاده در شرایط واقعی انتخاب کنند. در سال‌های اخیر واکنش مناسب و بموقع اهمیت بسیار زیادی پیدا کرده است. در همین راستا، توجه به بازی جنگ در میان کشورهایی مانند ایالات متحده آمریکا نسبت به گذشته بسیار بیشتر شده است (rand.com, 2020).

یک از شکل‌های بازی جنگ حالت پایش یک منطقه تحت کنترل و تعقیب و گریز[2] احتمالی است. مسأله تعقیب و گریز به دسته‌ای از مسائل در حوزه‌ی ریاضیات و علوم کامپیوتر اطلاق می‌شود که در یک فضای خاص، یک گروه تعقیب‌کننده (جویندگان و یا پلیس‌ها) در پِی دستگیرکردن گریزنده (دزد) هستند (Isaacs, 1965:51). جویندگان می‌توانند برای مثال نمایانگر سربازان، پلیس، ربات‌ها، نرم‌افزارهای امنیتی یا وسایل نقلیه دشمن و یا حتی گازهای سمی باشند. انواع گوناگونی از مسائل تعقیب و گریز وجود دارد که هر یک از آنها عناصر خاص خود، مانند تعیین قوانین بازی، مشخصات محیط، نوع تابع هدف، قوانین دستگیری و غیره را دارا هستند (Winkler, 1983) و (Aigner, 1984).

مسأله تعقیب و گریز برای چند فرضیه خاص حل شده است. از جمله این فرضیه‌ها می‌توان به نوع حرکت، نحوه‌ی دستگیری و اتمام بازی (Vieira, 2009) ، سرعت حرکت اعضاء (Gerkey, 2006) و میزان آگاهی از یکدیگر (Suzuki, 1992)، اشاره کرد.

مسأله دزد و پلیس حالت کلاسیک و مورد خاصی از مسأله تعقیب و گریز است که روی گراف مطرح می‌شود (Allis, 1994). در این روش، اعضای هر گروه روی رئوس گراف قرار دارند و فقط مجاز هستند به رئوس مجاور حرکت کنند. در این ساختار گراف، دو رأس مجاور با یک یال به یکدیگر متصل هستند (Bonato, 2009). همچنین هر رأس به وسیله یال طوقه به

---

1. Wargames
2. Pursuit & Evasion Problems

خودش نیز متصل است. هدف از لحاظ کردن یال طوقه این است که هر یک از اعضای بازی بتوانند در نوبت حرکت خود ثابت باقی بمانند (Allis, 1994).

دو پرسش بنیادی در مسائل تعقیب و گریز عبارتند از (Bonato, 2009):

۱- کمینه تعداد تعقیب کننده برای اطمینان از این‌که الگوریتم ارائه شده همگرا شود چه تعداد است؟

۲- مناسب‌ترین استراتژی حرکت و الگوریتم برای دستیابی به نتیجه در کمترین زمان چیست؟

نشان داده شده است که اگر در بازی دزد و پلیس تعداد دزدها ۱ و تعداد پلیس‌ها ۲ باشد، الگوریتم حتماً همگرا است (Alspach, 2004). در این مقاله، به بررسی پرسش دوم پرداخته شده است به نحوی که، ابتدا به حل مسأله تعقیب و گریز در حالت تعمیم نیافته با استفاده از روش‌های کمینه بیشینه[۱]، هرس آلفا-بتا[۲] و هندسه‌ی زبان‌شناختی[۳] می‌پردازیم و نتایج آنها را با یکدیگر مقایسه می‌کنیم. از طرف دیگر برای توسعه و واقعیت‌گرایی بیشتر مسأله، برخی جنبه‌های جدید را مورد بررسی قرار می‌دهیم. از جمله این موارد می‌توانیم به افزایش اندازه مسأله تا ۱۸×۱۸، ایجاد هفت درجه آزادی بجای پنج درجه آزادی در گره‌ها به دلیل تقسیم‌بندی ساختار زمین به شش ضلعی منتظم بجای چهار ضلعی، شروع تصادفی بازی در زمین با اعمال محدودیت‌های دید و در نهایت وجود موانع در زمین بازی اشاره کنیم. این مسأله را با استفاده از روش هندسه زبان‌شناختی حل می‌کنیم و مزیت آن را بر سایر روش‌ها نشان می‌دهیم. قابل ذکر است که با در نظر گرفتن سلول‌های شش‌ضلعی به جای چهارضلعی، فاکتور شاخه‌زنی[۴] (b) در درخت جستجو از ۴ به ۶ افزایش می‌یابد و در نتیجه پیچیدگی زمانی جستجوی کامل از $O(b^4)$ به $O(b^6)$ افزایش می‌یابد که بسیار قابل ملاحظه می‌باشد.

مبانی نظری و پیشینه پژوهش

هندسه‌ی زبان‌شناختی

هندسه زبان‌شناختی (LG) رویکردی جدید در حوزه‌ی نظریه بازی است که کاربرد آن حل مسائل بزرگ در زمان اندک است. طبق نظریه ارائه شده توسط بوریس استیلمن[۵] (۲۰۰۵)، هندسه‌ی زبان‌شناختی "*شامل ابزارهای نحوی برای ارائه دانش و استدلال در مورد سیستم‌های*

---

1. Minimax
2. Alfa Beta Pruning
3. Linguistic Geometry (LG)
4. Branching factor
4. Boris Stilman

پیچیده *چندگانه*" است. در هندسه زبان‌شناختی یک سیستم پیچیده (به عنوان مثال، یک بازی در گراف) به صورت رشته زیر نمایش داده می‌شود:

$$? \; X, P, R_P, SPACE, val, S_0, S_T, TR \; \mathrm{A}$$

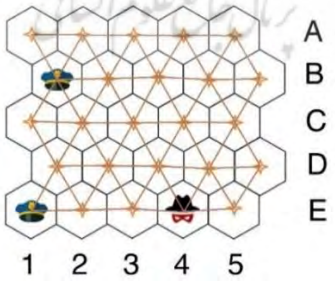
برای روشن شدن تعاریف، مثال ساده‌ای از مسأله دزد و پلیس در شبکه‌ی ۵×۵ در شکل (۱) نشان داده شده است. هشت عنصر هندسه زبان‌شناختی به شرح زیر است:

۱- $X \cong \{x_i\}$، نشان دهنده نقاط محیط یا به طور دقیقتر رأس‌های گراف است. در شکل (۱)، x نشان دهنده‌ی مجموعه‌ی ۲۵ نقطه (سلول) محیط است.

۲- $P \cong \{p_i\}$، عامل[1]های بازی را نمایش می‌دهد که حاصل اجتماع دو مجموعه‌ی جدا از طرف‌های مخالف است. به عنوان مثال در شکل (۱) {پلیس۱، پلیس۲} یک مجموعه، و {دزد} مجموعه مقابل را تشکیل می‌دهند و P اجتماع این دو مجموعه است.

۳- R_p، مجموعه‌ای از روابط دودویی از دستیابی عامل P در x است به‌طوری‌که $R_p +x, y, \cong ۱$ ، یعنی عامل p می‌تواند از نقطه فعلی خود یعنی x به نقطه y برسد. عامل‌ها ممکن است پرواز کنند، بخزند و یا بپرند و در بازی دزد و پلیس ممکن است به همسایگی‌های مجاز خود حرکت کنند. اگر عامل‌های p و q در تقابل باشند مقدار $OPPOSE +p, q, \cong 1$ خواهد بود.

مقدار $MAP_{x,p} +y, = K$ یعنی P می‌تواند در K مرحله از x به y برسد و بنابراین $y \angle M^{K}_{x,p}$ که مجموعه‌ی نقاط قابل دسترسی در K مرحله است. لازم به ذکر است که اغلب شبکه‌های گرافی مورد مطالعه در مقالات قبلی بصورت بخش‌بندی مربعی بوده‌اند (Bhattacharya, 2010).



شکل (۱) شبکه در اندازه ۵ با دو پلیس و یک دزد و گراف حاصل از حرکت‌های عامل‌ها

[1]. Agent

۴- SPACE، صحنه‌ی بازی یا وضعیت‌های ممکن را شرح می‌دهد یعنی اینکه چطور تمام عوامل در x واقع شده‌اند و نوبت کدام یک از آن‌هاست. مقدار $ON\#P, = x$، یعنی عامل P مکان x را در وضعیت S اشغال می‌کند. با استفاده از رئوس گراف شکل (۱) می‌توان ترکیبات مختلفی از وضعیت قرارگیری عامل‌های درگیر در بازی ایجاد کرد که مجموعه کلیه ترکیبات ممکن SPACE را تشکیل می‌دهد. به‌عنوان نمونه یکی از حالت‌های SPACE به صورت زیر است.

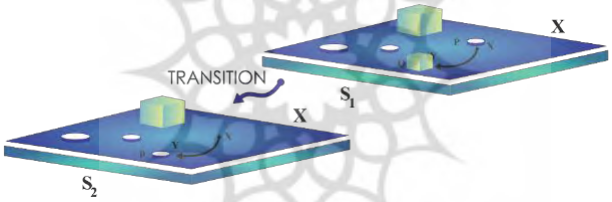
$ON(1\text{پلیس})=B_1$ $ON(2\text{پلیس})=E_1$ $ON(\text{دزد})=E_4$

۵- Val، نشان دهنده ارزش عامل P است که در ضمن بیانگر ارزش حذف این عامل از بازی نیز هست.

۶- S_0، نشان دهنده وضعیت اولیه بازی است.

۷- S_t، نشان دهنده وضعیت نهایی بازی است که در آن حداقل یک پلیس، دزد را دستگیر می‌کند.

۸- TR، نشان دهنده مجموعه‌ای از انتقالات[1] (گذار) سیستم از یک وضعیت به وضعیت دیگر است.



شکل (۲) نمایش گرافیکی یک انتقال (گذار)

سلسله مراتب زبان‌های رسمی

در هندسه زبان‌شناختی، سلسله مراتب زیرسیستم‌ها به عنوان سلسله مراتب زبان‌های رسمی نمایش داده می‌شود. قبل از توصیف زبان رسمی نیاز به بعضی از تعاریف زیر داریم:

الف) نماد: عنصری انتزاعی مانند a، t، $a(xy)$ یا $t(p_2, t_2, \pi_2)$ است.

ب) رشته: دنباله‌ای متناهی از نمادهای پیوندی است. به عنوان مثال، اگر $a(x_1), a(x_2), ..., a(x_n)$ نماد باشند، آنگاه $a(x_1)a(x_2)...a(x_n)$ نشان دهنده یک رشته است.

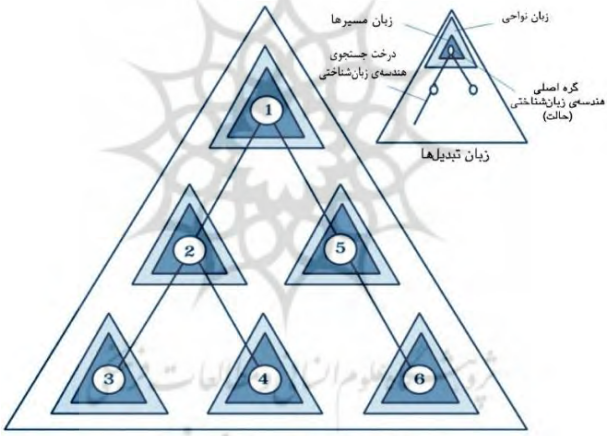
پ) الفبا: مجموعه‌ای متناهی از نمادها است.

ت) زبان رسمی: یک زبان رسمی مجموعه‌ای از نمادهای یک الفبا و رشته‌ها است.

[1] Transition

برای حل یک مسأله همانطور که در شکل (۳) نشان داده شده است، هندسه زبان‌شناختی سلسله مراتبی سه سطحی از زبان‌های رسمی تولید می‌کند (Stilman , 2010). این سلسله مراتب به‌ترتیب عبارتند از زبان مسیرها[۱] و زبان نواحی[۲] و زبان تبدیل‌ها[۳]. فرایند یافتن پاسخ برای یک مسأله با تولید رشته‌ای از زبان تبدیل‌ها از طریق ایجاد یک زبان نواحی (سطح دوم) و زبان مسیرها (پایین‌ترین سطح، تیره‌ترین مثلث‌ها) برای تعدادی از حالت‌های سیستم انجام می‌شود. به این مجموعه «درخت تبدیل‌ها[۴]» گفته می‌شود که در شکل (۳) بزرگترین مثلث در بالاترین سطح است.

در حالت خاص، برای هر نماد باید یک زوج از این زبان‌ها تولید شود، که شامل درخت تبدیل‌ها، همراه با وضعیت سیستم هندسه زبان‌شناختی که توسط گره‌ها در شکل (۴) نشان داده شده است. در سلسله مراتب زبان‌های رسمی، هر رشته از زبان سطح پایین مربوط به یک نماد سطح بالاتر است (Maddahi & Masehian, 2017).



شکل (۳) سلسله مراتب زبان‌های رسمی (Maddahi & Masehian, 2017)

رشته‌ی زبان تبدیل‌ها نشان‌دهنده‌ی درخت جستجو[۵] است (Stilman, 2002). زیرشاخه‌ای (رشته) که زیردرخت مطلوب را نشان می‌دهد، یک راه‌حل برای مسأله داده شده است. هندسه

---

1. Language of Trajectories
2. Language of Zones
3. Language of Translations
4. Tree of Translations
5. LG Search Tree

زبان‌شناختی شامل چهار عنصر ضروری است که عبارتند از مسیرها؛ نواحی؛ تبدیل‌ها و جستجو.

از آنجایی که برای هر کدام از سطح‌های اول، دوم و چهارم دو عنصر دستور زبان و زبان وجود دارد، می‌توانیم بگوییم یک دستور زبان برای مسیرها[۱] (G_t) و زبان مسیرها[۲] (L_t) وجود دارد. به طور مشابه دستور زبان برای نواحی[۳] (G_Z) و زبان نواحی[۴] (L_Z) و دستور زبان برای جستجوها[۵] (G_S) و زبان جستجوها[۶] (L_S) نیز وجود دارند. در شکل (۴)، یک روش معمول برای حل مسأله توسط هندسه زبان‌شناختی که از پایین سمت چپ شروع می‌شود و در بالا سمت راست به پایان می‌رسد نشان داده شده است. روش به این صورت است که ابتدا از روی دستور زبان مسیرها، زبان مسیرها ساخته می‌شود. در مرحله بعد، دستور زبان ناحیه (ناحیه‌ها) ساخته می‌شود. از روی دستور زبان ناحیه (ناحیه‌ها)، زبان ناحیه (ناحیه‌ها) ساخته می‌شوند. در گام بعد، تبدیل‌ها[۷] وارد عمل می‌شوند. «تبدیل‌ها» می‌توانند زبان مسیرها و یا نواحی و همچنین دستور زبان مسیرها و یا نواحی را تغییر دهند. اصولاً تبدیل‌ها به صورتی کارآمد، مسیرها و نواحی (سلسله مراتب ساختارها) را (در حالی که از یک حالت به حالت دیگر بازخوانی G_t عبور می‌کنند) تنظیم می‌کنند.

در گام آخر (سطح چهارم)، دستور زبان جستجو، با استفاده از ورودی سطح پیشین خود، زبان جستجو(L_S) را تولید می‌کند (Maddahi & Masehian, 2017). در بخش بعدی توضیحات مفصل‌تری پیرامون زبان مسیرها، زبان نواحی و زبان تبدیل‌ها ارائه می‌کنیم.

زبان مسیرها

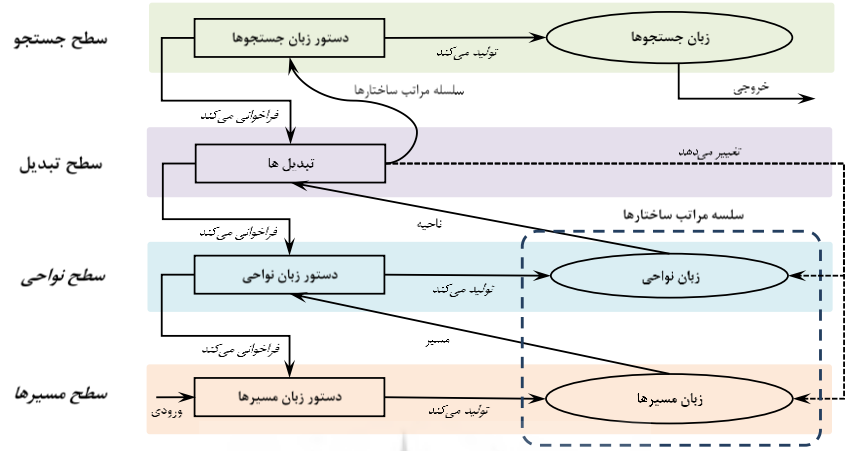
یک مسیر (راه) توسط یک رشته از الفبا (حروف) نمایش داده می‌شود و یک مسیر برنامه‌ریزی شده است که باید توسط عامل خاص برای رسیدن به هدفش مشخص شود. یک مسیر نوعی می‌تواند با $t_0 \cong a(x)a(x_1)..a(x_n)$ نمایش داده شود که در آن a یک نماد و x_i یک نقطه در سیستم است.

نتیجه‌ی اجرای L_t مجموعه‌ای از مسیرهای قابل پذیرش از درجه k، از وضعیت فِعلی عامل‌ها به هدف است (Maddahi, 2012) و (Maddahi & Masehian, 2017). همه مسیرها

---

1. Grammar of Trajectories
2. Language of Trajectories
3. Grammar of Zones
4. Language of Zones
5. Grammar of Searches
6. Language of Searches
7. Translations

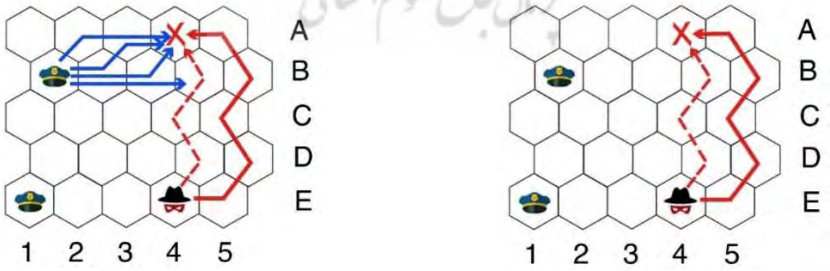
برای عامل p که از x شروع شده و با طول l به y ختم می‌شود مجموعه‌ای را تشکیل می‌دهند که توسط $t_p(x,y,l)$ نمایش داده می‌شود.



شکل (۴) عناصر تشکیل‌دهنده‌ی LG (Maddahi & Masehian, 2017)

یک مسیر قابل پذیرش از درجه ۱، کوتاه‌ترین مسیر بین دو نقطه است. در شکل (۵)الف فرض بر این است که دزد می‌خواهد به هدف خودش در نقطه A4 برسد، کوتاهترین مسیرش (که با خط چین قرمز نشان داده شده) $t_S \cong a+E4, a+D4, a+C4, a+B4, a+A4,$ است.

یک مسیر مجاز از درجه‌ی k یک مسیر t از $t_p(x,y,l,$ است که اگر نقطه‌ی $z \angle x$ وجود داشته باشد، پیوندی از مسیر مجاز درجه‌ی k-۱ از $t_p(z,y,l_1,$ و کوتاه‌ترین مسیر $t_p(z,y,l_2,$ که $l_1 \cdot l_2 \cong 1$ است. بر این اساس مسیر مجاز درجه ۲ شامل دو کوتاه‌ترین مسیر است. در شکل (۵)الف یک مسیر مجاز از درجه ۲ (که با خط قرمز نمایش داده شده) $t_0' \cong a+E4, a(E5)a+D5, a+C5, a+B5, a+A5, a(A4)$ است.



شکل (۵) نمایش برخی مسیرهای ممکن در زمین شش‌ضلعی از اندازه ۵

زبان مسیرها L_t^H یک توصیف رسمی از مجموعه‌ی زیر سیستم‌های پایین‌ترین سطح است، یعنی مجموعه‌ی همه‌ی مسیرها از طول کم‌تر یا مساوی H بین نقاط یک سیستم پیچیده است.

این مجموعه همچین شامل رشته‌ی تهی (مسیرها) با طول صفر است. زبان مسیرها توسط دستور زبان کوتاه‌ترین مسیر $G_t(1)$ (که یک دستور زبان کنترل شده است) تولید شده است.

$$\begin{aligned} &1.S \to aA \\ &2.A \to aA \\ &3.A \to a \end{aligned} \qquad (۵)$$

که در آن $V_N = \{S, A\}$ شامل نمادهای غیر پایان‌بخش[1] (S نماد شروع است) و $V_T = \{a\}$ شامل نمادهای پایان‌بخش[2] است که V_N و V_T به ترتیب الفبای نمادهای غیر پایان‌بخش و پایان‌بخش هستند.

معادله‌ی (۵) در چند نسخه گنجانده شده است و کل مجموعه‌ی نسخه‌های برحسب i، "دو" نامیده می‌شود و با تابع $next_i$ تولید می‌شود که بخش ضروری دستور زبان است.

(۶) تابع $next_i$ به شرح زیر تعریف می شود:

$D = next_i, \cong X \times Z \times X^2 \times Z \times P$
$SUM \cong \{v \mid v \text{ from } X,\ MAP_{X_0,P}(v) \cdot MAP_{y_0,P}(v) \cong l_0\},$
$ST_k(X) \cong \{v \mid v \text{ from } X, MAP_{X,P}(v) \cong k\},$
$MOVE_l(X)$ is an intersection of the following sets:
$ST_1(X),\ ST_{l_0 01.1}(X_0)$ and SUM.
if
$MOVE_l(X) \cong \{m_1, m_2, \ldots, m_r\} \approx \cup$
then
$next_i(X, l) \cong m_i$ for $i \infty r$ and
$next_i(X, l) \cong m_r$ for $r < i \infty n$,
else
$next_i(X, l) \cong X.$
endif

از آنجایی که استفاده مجدد از معادله (۵) یک رشته از a را به وجود می‌آورد تا $a^1, a^2, \ldots, a^n$ تابع $next_i$ پارامترهای این رشته را ارزیابی کند (Stilman,1997).

---

[1]. Nonterminal
[2]. Terminal

در مورد دستورالعمل (دستور زبان) تولید کوتاه‌ترین مسیر به تفصیل توسط استیلمن[1] (۱۹۹۴) مطالبی ارائه شده است که در صورت بیان ناقص گام‌های گسترده الگوریتم برای خوانندگان غیرقابل درک خواهد. هدف این مقاله پرداختن به ماهیت الگوریتم LG نبوده بلکه به دنبال نشان دادن کارایی آن در مواجهه با مسائل با ابعاد زیاد است.

زبان نواحی

زبان نواحی شبکه‌ای از مسیرهای متصل به هم است. یکی از این مسیرها «مسیر اصلی» است. مسیر اصلی، سمت حرکتی است که عامل اصلی، به طولی که توسط کران H و مقداری وابسته برای دامنه‌ی مسأله‌ای خاص تعیین‌شده است، به سمت هدف طی می‌کند (Stilman, 1995).

نماد $t(P_0, t_0, \omega_1)$ مربوط به مسیر اصلی t_0 از عامل اصلی P_0 و ω_1، نشان‌دهنده‌ی زمان مربوط به طی مسیر t_i است. «عامل اصلی» عنصری است که باید به هدفش برسد، در حالی که عامل‌های مقابل سعی دارند با قطع‌کردن مسیر، مانع از رسیدن عامل اصلی به هدف شوند.

در شکل (۵)ب فرض کنید که پلیس‌ها بازی را شروع می‌کنند و از مکان دزد مطلع هستند. لذا زبان نواحی به صورت غیررسمی اینگونه توصیف می‌شود که عامل اصلی (r / دزد) مسیر اصلی $t_0 \cong a+E4, a+D4, a+C4, a+B4, a+A4,$ را در $\omega_1 \cong 4$ گام طی می‌کند، در حالی که عامل رقیب (C_1 / پلیس بالایی) می‌تواند هر کدام از مسیرهای زیر را طی کند تا مسیر حرکت r را قطع کند.

$$t_1 \cong a+B_1, a+B_2, a+B_3, a+B_4, \quad (۷)$$

$$t_2 \cong a+B_1, a+B_2, a+B_3, a+A_4, \quad (۸)$$

$$t_3 \cong a+B_1, a+B_2, a+A_3, a+A_4, \quad (۹)$$

$$t_4 \cong a+B_1, a+A_2, a(A_3)a+A_4, \quad (۱۰)$$

بنابراین زبان نواحی وابسته به این شبکه از مسیرها به صورت زیر ساخته می‌شود:

$$L_Z \cong t+r, t_0, 4, t+C_1, t_1, 4, t+C_1, t_2, 4, t+C_1, t_3, 4, t(C_1, t_4, 4) \quad (۱۱)$$

در پایان یادآوری می‌کنیم که یک رشته از زبان مسیرها مانند $a+1, a+2, a+3,$ معادل یک نماد در زبان نواحی است.

[1]. Stilman

زبان جستجو و زبان تبدیل

مفهوم جستجو در هندسه زبان‌شناختی یک "ساختار" است، به این مفهوم که بر خلاف بسیاری از روش‌های موجود، کل یا فضای کوچکتر شده‌ای را جستجو نمی‌کند (Stilman & Yakhnis, 2002). "جستجو" با دو ساختار دیگر زبان نواحی و زبان تبدیل در ارتباط است. زبان نواحی نمایش‌دهنده‌ی مسیرهای x است که ما را به مکان اهداف موضعی راهنمایی می‌کند، در حالی که زبان تبدیل نمایش‌دهنده‌ی مسیرهای فضای مربوط به هدف‌های کلی مسأله است. در واقع جستجو به معنای گذارهای چندگانه بین وضعیت‌ها[1] است، که طی آن زبان نواحی در هر مرحله ساخته و به‌روز می‌شود. به این معنا که تنها یک رشته از زبان تبدیل ساخته می‌شود و بنابراین عمل ساخته‌شدن، کنترل‌کننده‌ی جستجو است (Stilman & Yakhnis, 2010).

خانواده‌ای از زبان‌های جستجو شامل زیرخانواده‌ای از جستجوهای کاهش داده‌شده G_{rs} از بازی‌های دو نفره است، مانند آنها که توسط الگوریتم‌های جستجوی معمول مانند الگوریتم جستجوی کامل[2]، تپه نوردی[3]، جستجوی اول عمق[4]، کمینه بیشینه[5] و هرس آلفا-بتا[6] و... تولید می‌شوند و می‌تواند توسط زبان‌های جستجوهای کاهش‌داده شده نمایش داده شود. این‌چنین کلاس تولیدشده توسط ابزارهای هندسه زبان‌شناختی به صورت «زبان تبدیل‌ها[7]» نام‌گذاری شده است.

در ادامه توضیحی اجمالی از چند الگوریتم جستجو ارائه می‌شود. این قابلیت در هندسه زبان‌شناختی وجود دارد که از هر کدام از این الگوریتم‌ها به عنوان موتور جستجو در سطح چهارم (سطح جستجو) هندسه زبان‌شناختی استفاده نماید.

انواع روش‌های جستجو

الف) جستجوی کامل

در این روش تمام فضای پاسخ در هر حرکت بدون هدف، به طور کامل مورد پیمایش قرار می‌گیرد. سپس پاسخ‌های مسأله را استخراج کرده و نتیجه الگوریتم را باز می‌گرداند. چنین روشی برای مسائل ساده و دارای پیچیدگی اندک به راحتی به پاسخ می‌رسد. ولی باید به این

---

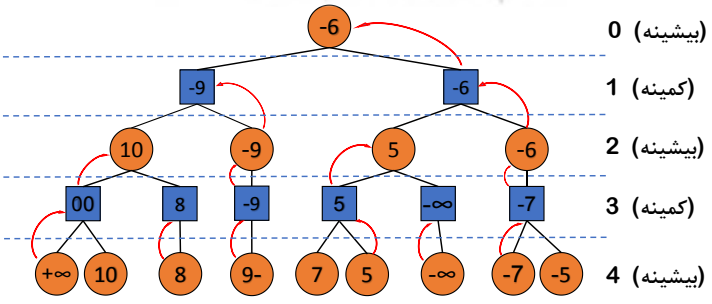
1. States
2. Brute Force
3. Hill climbing
4. Depth first
5. Minimax
6. Alpha-Beta Pruning
7. Language of Translations

نکته مهم دقت کنیم که با افزایش پیچیدگی مسأله، زمان پاسخ‌دهی به صورت نمایی افزایش پیدا می‌کند و این مورد یکی از مهمترین نقاط ضعف این الگوریتم جستجو است (Bal, 2015). امروزه از این روش تنها برای ارزیابی پاسخ‌هایی که با کمک سایر روش‌ها استخراج می‌شود، استفاده می‌کنند (Chung, 2011).

ب) الگوریتم کمینه بیشینه

این الگوریتم، الگوریتمی بازگشتی برای انتخاب حرکت بعدی توسط یکی از بازیکنان است و در مواردی کاربرد دارد که بازی میان دو فرد یا گروه انجام می‌شود. برای روشن شدن الگوریتم کمینه بیشینه می‌توان به بازی "دوز" اشاره کرد. این بازی مثالی ساده از یک بازی دو نفره است. در این بازی بازیکن X به طور ابتکاری در پی یافتن بهترین مسیر برای پیروزی در بازی است، در حالی که بازیکن O قصد دارد راه این بازیکن را سد کند. اگر هیچ یک از دو بازیکن مرتکب اشتباهی نشوند، بازی میان دو بازیکن کارآزموده همواره به طور مساوی به پایان می‌رسد. بازیکن X به مسیری نیاز دارد که حتی با وجود تلاش بازیکن O برای سد مسیر نیز به پیروزی برسد. الگوریتم کمینه بیشینه روشی برای یافتن چنین مسیری (در صورت وجود) است. به طور سنتی به دو بازیکن در بازی‌های دو نفره بیشینه (MAX) و کمینه (MIN) گفته می‌شود.

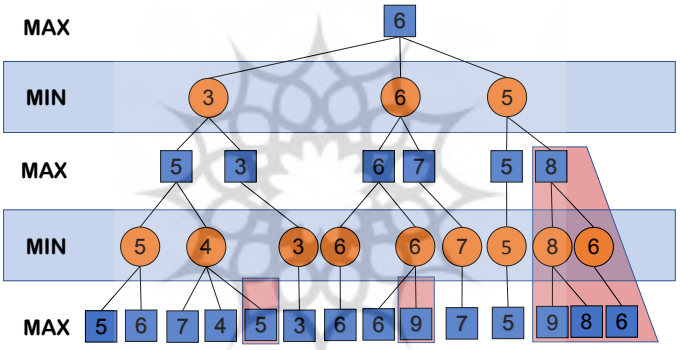
در هر گام از بازی، بیشینه حرکت خود را در جهتی انجام می‌دهد که احتمال بُرد خود را افزایش دهد و هم زمان احتمال بُرد حریف را کمینه کند. بنابر این هر حرکت به موقعیت یا مکان بازی وابسته است. با توجه به اینکه حرکات بازیکن کمینه در بازی غیر قابل پیش بینی است، برای پیروزی در بازی، بیشینه باید در پی یک استراتژی باشد که بدون توجه به عملکرد کمینه، به یک حالت پایانی برنده دست یابد. در این استراتژی، بیشینه باید حرکات درست را در حالت از حرکت کمینه اجرا کند (Stuart, 2009:167).



شکل (۶) روش عملکرد الگوریتم کمینه بیشینه (Ali, 2020)

پ) الگوریتم هرس آلفا-بتا

روش هرس آلفا-بتا یک الگوریتم جستجو است که به دنبال کاهش فضای گره‌هایی است که توسط الگوریتم کمینه بیشینه در درخت جستجو ارزیابی می‌شوند. کارکرد این الگوریتم به این صورت است که زمانی که حداقل یک امکان وجود داشته باشد که ثابت کند این حرکت از حرکتی که پیشتر آزمایش‌شده بدتر است، ارزش‌گذاری به این حرکت را به طور کامل متوقف می‌کند. یک درخت کمینه بیشینه را در نظر بگیرید، هرس آلفا-بتا همان حرکت روش کمینه بیشینه را انجام می‌دهد، با این تفاوت که شاخه‌هایی که تاثیری در جواب نهایی ندارند را دور می‌ریزد (Bonato, 2009). بدین ترتیب می‌توان زمان جستجو را به یک زیردرخت محدود کرد. لذا در همان زمان می‌تواند جستجوی عمیق‌تری انجام دهد. این بهینه‌سازی کارایی را تا دو برابر نسبت به کمینه بیشینه ساده بهبود می‌دهد (Maddahi, 2012). در ادامه و در شکل (۷)، مقایسه‌ای از نوع کارکرد روش کمینه بیشینه و هرس آلفا-بتا ارائه شده است.



شکل (۷) نوع کارکرد روش هرس آلفا-بتا در کمینه بیشینه (Kopec, 2016)

توسعه و حل مسأله

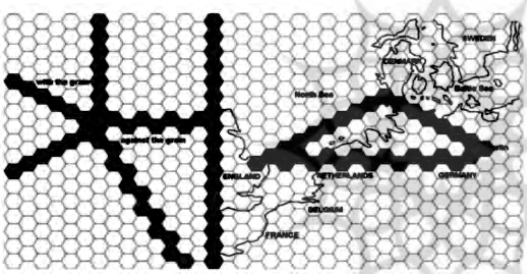
مطالعات پیشین روی هندسه زبان‌شناختی، نشان داده است که این روش در حل برخی از مسائل مرتبط با بازی‌های تخته‌ای انتزاعی[1]، نسبت به سایر روش‌ها عملکرد بهتری دارد. از سویی دیگر نمونه‌ای از مسائل که با روش هندسه زبان‌شناختی قابل حل است، دسته‌ی مسائل بازی جنگ و نبردهای نظامی است (Umansky & Stilman & Yakhnis, 2010). در این مقاله سعی شده عملکرد پایش منطقه و پاکسازی یک دسته زرهی که به صورت استاندارد در اکثر سازماندهی ارتش‌های دنیا از ۴ ارابه تانک تشکیل می‌گردند در مقابل تانک‌های مهاجم مورد

[1]. Abstract Board Games

بررسی قرار گیرد تا درصورتی که در آینده سیر پارادایم اتوماسیون سازی و نبردهای رباتیک بر صحنه نبردهای آتی مستولی گردید راهکاری برای حرکت بهینه ادوات جنگی بدست آید.

موارد گسترش مسأله بازی که از اهداف این مقاله می‌باشد به این شرح است:

۱- در مطالعات مربوط به بازی جنگ با رویکرد گسسته، لازم است میدان نبرد بخش‌بندی[1] شود. تاکنون دو نوع بخش‌بندی مربعی و یا شش‌ضلعی به این منظور پیشنهاد شده است. در بخش‌بندی مربعی، بازیکنان در چهار جهت امکان حرکت خواهند داشت و به‌طور کامل می‌توان میدان نبرد مورد نظر را با مربع‌های به هم پیوسته بخش‌بندی نمود. بخش‌بندی مربعی میدان نبرد باعث وجود محدودیت در حرکت بازیکنان می‌شود. به‌همین دلیل عموماً استفاده از تقسیم‌بندی شش‌ضلعی میدان نبرد در مطالعات امروزه کاربرد بیشتری دارد. با بخش‌بندی شش‌ضلعی، قابلیت تحرک بازیکنان بیشتر شده و امکان حرکت در شش جهت را بدست خواهند آورد (GAMMETER, 2013).



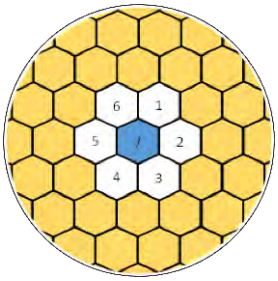
شکل (۸) بخش‌بندی شش‌ضلعی در بازی جنگ (Sabin, 2014)

شکل (۹) شطرنج شش‌ظلعی (Washburn, 2009)

شکل شماره (۹) صفحه شطرنجی را نشان می‌دهد که خانه‌های آن بصورت شش ضلعی است که امکان حرکت مهره‌ها به سمت‌های بیشتری را فراهم می‌نماید که در سال ۱۹۳۰ محبوبیت داشت.

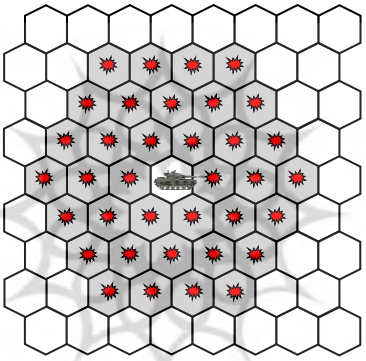
برای افزایش آزادی عمل حرکت عوامل در هر نوبت، دامنه‌ی حرکتی اعضای بازی را از مربع به شش ضلعی منتظم گسترش دادیم. در عمل، در هر گام، برای هر عامل در بازی امکان ۷ حرکت (۶ حرکت به طرفین و ۱ حالت سکون) همانند شکل(۱۰) وجود دارد.

[1]. Segmentation



شکل (۱۰) مکان‌های قابل دسترسی برای هر عامل در هر گذار

۲- با توجه به اینکه در واقعیت یک تانک رزمی تا جایی از زمین بازی دید دارد و بطور طبیعی دارای محدودیت در برد میدان دید است، در این مقاله این محدودیت به مسأله اضافه گردیده و میدان دید هر عضو بازی در هر جهت به اندازه‌ی سه خانه در نظر گرفته شده است، شکل (۱۱).

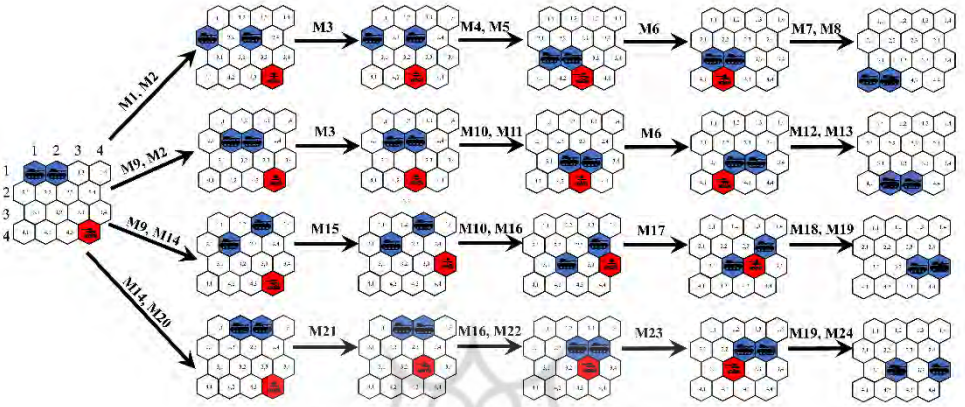


شکل (۱۱) نمایش برد میدان دید عامل دنبال کننده

۳- آغاز بازی کاملاً به طور تصادفی است، ولی به محض قرار گرفتن یکی از عوامل متجاوز در محدوده دید یکی از دنبال کننده‌ها و مشاهده شدن آن، بازی از حالت تصادفی به آگاهانه تبدیل شده و در ادامه تمامی حرکات در تعقیب و گریز با استفاده از الگوریتم هندسه زبان‌شناختی و با این هدف انجام می‌گردد که دنبال کننده‌ها در کم‌ترین تعداد حرکت، گریزندگان متجاوز را به‌صورت نوبتی دستگیر می‌نمایند.

با توجه به اینکه پیچیدگی مسأله با گسترش فضای بازی به شدت افزایش می‌یابد برای نشان‌دادن گذارها در هر نوبت از بازی و تبدیل موقعیت قبلی به موقعیت جدید، تعداد بسیار زیادی از گذارها باید نمایش داده شوند که این موضوع امکان‌پذیر نیست. به همین دلیل یک مسأله تعمیم‌یافته بسیار کوچک با اندازه ۴ در نظر گرفته شده و نحوه‌ی ساخته‌شدن گذارها توسط الگوریتم هندسه زبان‌شناختی در این مسأله به تصویر کشیده شده است. در این مثال،

آدرس‌دهی ستون‌ها و سطرها با شماره‌های ۴،۱،۲،۳ انجام شده است که حالتی اختیاری است. وضعیت اولیه شروع بازی برای دنبال کننده‌ها از خانه (۱،۱) و (۱،۲) و برای گریزنده از خانه (۴،۴) می‌باشد. بر این اساس در ۵ گام گریزنده موردنظر توسط ۲ دنبال کننده دستگیر می‌شود.



شکل (۱۲) بخشی از زبان مسیرها، تولید شده توسط الگوریتم هندسه زبان‌شناختی

جدول (۱) بخشی از زبان مسیرها، تولید شده توسط الگوریتم هندسه زبان‌شناختی

M1 = Transition(P1, (1,1), (1,1))	M13 = Transition(P2, (3,3), (4,3))
M2 = Transition(P2, (1,2), (2,3))	M14 = Transition(P2, (1,2), (1,3))
M3 = Transition(E, (4,4), (4,3))	M15 = Transition(E, (4,4), (3,4))
M4 = Transition(P1, (2,1), (3,1))	M16 = Transition(P2, (1,3), (2,4))
M5 = Transition(P2, (2,3), (3,2))	M17 = Transition(E, (3,4), (3,3))
M6 = Transition(E, (4,3), (4,2))	M18 = Transition(P1, (3,2), (3,3))
M7 = Transition(P1, (3,1), (4,1))	M19 = Transition(P2, (2,4), (3,4))
M8 = Transition(P2, (3,2), (4,2))	M20 = Transition(P1, (1,1), (1,2))
M9 = Transition(P1, (1,1), (2,2))	M21 = Transition(E, (4,4), (3,3))
M10 = Transition(P1, (2,2), (3,2))	M22 = Transition(P1, (1,2), (2,3))
M11 = Transition(P2, (2,3), (3,3))	M23 = Transition(E, (3,2), (3,3))
M12 = Transition(P1, (3,2), (4,2))	M24 = Transition(P1, (2,3), (3,2))

گذارهای بالا نحوه عملکرد روش هندسه زبان‌شناختی در دستگیری عامل گریزنده در بازی را نمایش می‌دهد. همانطور که در گام‌های ارائه‌شده در شکل (۱۲) به ترتیب اجرا مشخص است، استراتژی دنبال کننده‌ها، تنگ‌کردن عرصه بازی برای گریزنده است. این مطلب دقیقاً مشابه مراحل انتهایی[1] بازی شطرنج است که تلاش می‌شود مهره شاه حریف در گوشه‌ای گیر انداخته و مات شود.

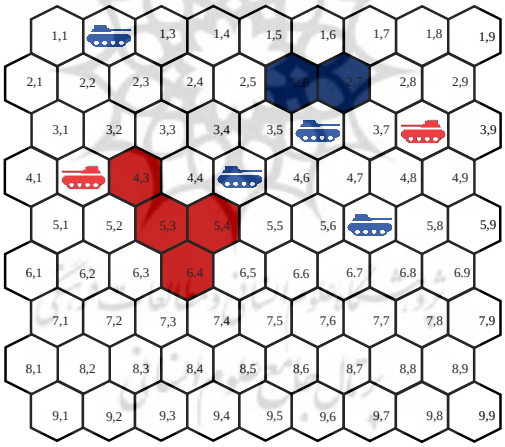
[1]. End Game

برای اطمینان بیشتر از کارایی الگوریتم، مسأله‌هایی در اندازه‌های ۶، ۹ و ۱۸ حل شده و به جهت دستیابی به دقت مورد نظر، از هر مسأله ۱۰ بار خروجی گرفته شده و سپس میانگین نتایج استخراج گردیده و در نهایت این نتایج با الگوریتم‌های دیگر برای حل مسأله تعقیب و گریز مقایسه شده است.

به عنوان مثال برای یک مسأله در اندازه ۹×۹ نقطه ابتدایی قرارگیری هر عضو در یک بازی به شکل تصادفی به صورت زیر در نظر گرفته شده است.

* دنبال کننده شماره ۱ در خانه شماره (۱،۲)؛
* دنبال کننده شماره ۲ در خانه شماره (۳،۶)؛
* دنبال کننده شماره ۳ در خانه شماره (۴،۵)؛
* دنبال کننده شماره ۴ در خانه شماره (۵،۷)؛
* گریزنده شماره ۱ در خانه شماره (۳،۸)؛
* گریزنده شماره ۲ در خانه شماره (۴،۲)؛

با توجه به شکل(۱۳) خانه‌های شماره(۶،۲)، (۷،۲)، (۳،۴)، (۳،۵)، (۴،۵) و (۴،۶) به عنوان مانع درنظر گرفته شده که حرکت به داخل این خانه‌ها برای هر دو گروه ممنوع است. این موانع نشان دهنده این واقعیت است که ممکن است نقشه زمین بازی دارای عوارضی مانند رودخانه و یا کوه دارای شیب زیاد باشد که برای ارابه تانک غیرقابل عبور است.



شکل(۱۳) تصویر موقعیت ابتدایی عوامل در بازی با ابعاد ۹×۹

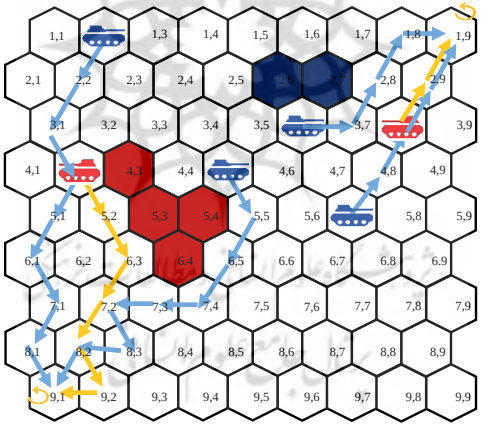
رفتار هر عامل در هر گام به‌این صورت است که در هر حرکت می‌تواند به یکی از ۶ خانه‌ی کناری خود حرکت کند و یا در جایگاه خود بماند. بر این اساس بازی به تناوب به گروه گریزنده و دنبال کننده داده می‌شود.

با قرارگیری یک عامل در میدان دید گروه مقابل، وجود آن برای تمامی اعضای تیم مقابل آشکار شده و اعضای تیم مقابل درصدد نابودی آن بر می‌آیند. گروه دنبال کننده با دیدن عامل‌های گروه گریزنده گروه‌های ۲تایی تشکیل داده و به دنبال عامل گریزنده می‌روند (بر

اساس الگوریتم ابتدایی هندسه زبان‌شناختی و این نکته که حداقل تعداد لازم دنبال کننده عدد ۲ است) تا آن را دستگیر (نابود) کنند.

این مسأله تحت شرایط آغازین بالا به‌وسیله‌ی نرم‌افزار متلب روی کامپیوتری با ۱۶ گیگابایت رم و پردازنده ۶ هسته‌ای Core i5 با قدرت ۲/۴ گیگاهرتز، شبیه‌سازی شده که خروجی حاصل به شرح زیر می‌باشد:

- مدت زمان اجرا: ۱۴ ثانیه
- تعداد حرکت هر گروه: ۱۴
- تعداد گره پردازش شده: ۵۵۹۶
- همانگونه که در شکل(۱۴) مشخص است، پس از همگرایی در گذار پایانی دنبال کننده شماره۱ و شماره۳ یک تیم تشکیل داده و در نقاط (۸،۱) و (۲،۸) گریزنده شماره۲ را در نقطه (۱،۹) دستگیر می‌نمایند و دنبال کننده شماره۲ و شماره۴ نیز تیم دوم را تشکیل داده و در نقاط (۸،۱) و (۹،۲) گریزنده شماره۱ را در نقطه (۹،۱) دستگیر می‌نمایند.



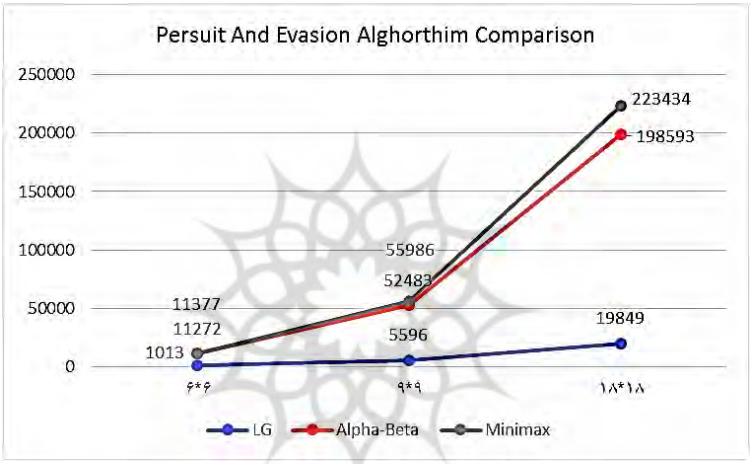
شکل (۱۴) گام‌های اجرای الگوریتم بازی

بعد از ارائه روش و بررسی حل در مقیاس‌های مختلف، جهت بررسی توانایی بهبود الگوریتم LG نسبت به سایر الگوریتم‌های مرسوم قابل استفاده، میانگین زمان صرف شده برای اتمام بازی و تعداد گره‌های مورد پردازش توسط روش‌های مختلف مورد بررسی قرار گرفته است.

جدول (۲) مقایسه تعداد گره‌های بررسی‌شده در هر روش

	۶∂۶	۹∂۹	۱۸∂۱۸
LG	۱۰۱۳	۵۵۹۶	۱۹۸۴۹
Alpha-Beta	۱۱۲۷۲	۵۲۴۸۳	۱۹۸۵۹۳
Minimax	۱۱۳۷۷	۵۵۹۸۶	۲۲۳۴۳۴

جدول (۲)، تعداد دقیق گره‌های مورد بررسی مسأله در اندازه‌های ۶، ۹ و ۱۸ در هر روش را درج نموده است. بر این اساس روش هندسه زبان‌شناختی، تقریباً تعداد ۹۱٪ گره کم‌تر را مورد بررسی قرار می‌دهد. اطلاعات جدول (۲) در شکل (۱۵) ترسیم شده است.

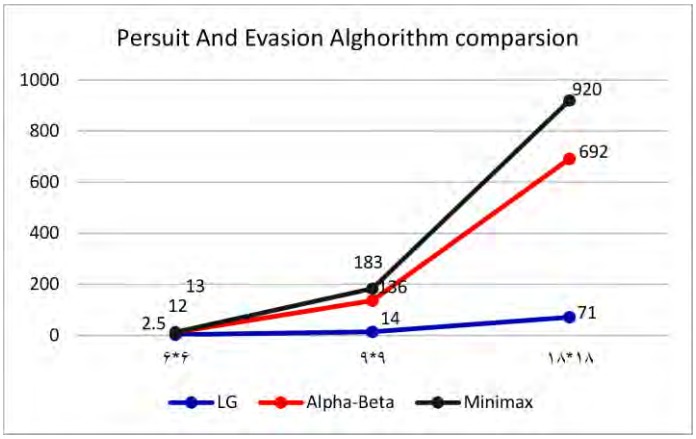


شکل (۱۵) نمودار تعداد گره پردازش‌شده برای روش‌های مختلف در اندازه‌های مختلف

پس از بررسی تعداد گره‌ها، مرتبه زمانی نیز مورد بررسی قرار گرفته که میانگین خروجی مرتبه‌های زمانی در جدول شماره ۳ و نمودار شکل (۱۶) ارائه گردیده است.

جدول (۳) مقایسه مرتبه زمانی روش‌های مختلف برای حل مسأله تعقیب و گریز

	۶∂۶	۹∂۹	۱۸∂۱۸
LG	۲/۵	۱۶	۷۱
Alpha-Beta	۱۲	۱۳۶	۶۹۲
Minimax	۱۳	۱۸۳	۹۲۰



شکل (۱۶) نمودار مرتبه‌ی زمانی برای روش‌های مختلف از اندازه‌های مختلف

نتیجه‌گیری

در مسائل تعقیب و گریز میزان پیچیدگی مسأله با افزایش ابعاد فضای مورد بررسی به شدت افزایش می‌یابد و الگوریتم‌های قدیمی مرسوم این حوزه مانند کمینه بیشینه و هرس آلفا-بتا و غیره که برای حل این مسائل مورد استفاده قرار می‌گیرند فقط در مورد مسائل با ابعاد کوچک و محدودیت‌های اندک بر روی گراف قابل استفاده بوده و برای مسائل پیشرفته این حوزه که در طبقه مسائل بازی جنگ طبقه‌بندی می‌گردند کارایی لازم در زمان مناسب را ندارند. در سال‌های اخیر الگوریتم هندسه‌ی زبان‌شناختی برای بازی‌های تخته‌ای ارائه گردیده که در مسائل ساده این حوزه بهبود چشمگیری نسبت به الگوریتم‌های جستجوی قبلی ایجاد کرده است.

در این مقاله سعی گردید تحت شرایط مختلف برای مسأله تعقیب و گریز، با نزدیک کردن شرایط اجرای مسأله به شرایط دنیای واقعی توان بهبود عملی الگوریتم هندسه زبان‌شناختی مورد بررسی قرار گیرد. جهت توسعه و نزدیک‌کردن مسأله به واقعیت در گام نخست به‌واسطه تقسیم‌بندی ساختار زمین بازی به شش ضلعی منتظم به‌جای چهارضلعی، درجه آزادی عوامل شرکت کننده در بازی از ۵ به ۷ در هر گره افزایش یافته، همچنین به مانند یک مأموریت رزمی پاکسازی و تامین، شروع بازی به‌صورت ناآگاهانه و تصادفی در نظر گرفته شده و محدویت طبیعی برد میدان دید در نظر گرفته شده و در نهایت موانعی در محیط قرار داده شده که عوامل بایستی در حرکت خود آنها را نیز مدنظر قرار دهند. در این تحقیق با اعمال شرایط و فرضیات جدید و با محل‌های ابتدایی گوناگون برای عوامل، مسأله چندین بار و در

ابعاد مختلف تا اندازه فضای جستجوی ۱۸×۱۸ مورد بررسی قرار گرفته که نتایج حاصله بهبود ۹۱ درصدی نسبت به بهترین الگوریتم حل مسأله‌ی تعقیب و گریز را نشان می‌دهد که نشان‌دهنده کارایی فوق‌العاده الگوریتم LG در حل مسائل بازی جنگ است.

منابع

≠ Aigner, M. &. (1984). A game of cops and robbers. *Discrete Mathematics, 8*(1), 1-12.

≠ Allis, L. V. (1994). Proof Number search. *Artificial Intelligence, 66*(1), 91-124.

≠ Alspach, B. (2004). Searching and sweeping graphs: a brief survey. *Le Matematiche, 59*(1-2), 5-37.

≠ Bal, D, Bonato. A, Kinnersley. W. B, Pralat. P. (2015). *Lazy Cops and Robbers on hypercubes.* Cambridge University Press.

≠ Bhattacharya, G. P. (2010). A cops and robber game in multidimensional grid. *Discrete Mathematics*.

≠ Bonato, P. G. (2009). The Capture time of a graph. *Discrete Mathematics*.

≠ Bonato, P. G. (2009). The capture of time graph. *Discrete mathematics*, 5588-5595.

≠ Chung, T. (2011). Search and pursuit-evasion in mobile robotics. *Robots*, 299-316.

≠ Deo, C. R., & Ali, M. (2020). Modeling wheat yield with data-intelligent algorithms: artificial neural network versus genetic programming and minimax probability machine regression. In *Hnadbook of Probbabillistic Models* (pp. 37-87). Butterworth-Heinmann.

≠ Fomin, F. V. (2008). An annotated bibliography on guaranteed graph searching. *Theoritical Computer Scienece, 399*(3), 236-245.

≠ GAMMETER, S. (2013). *LINGUISTIC GEOMETRY (LG) AND ITS APPLICATION TO HISTORICAL CONFLICTS.* mathesis, University of Colorado at Denver, University of Colorado at Denver.

≠ Gerkey, B. T. (2006). Visibility based pursuit with limited field of view. *International Journal of Robotics Research, 25*(4), 20-27.

≠ Hahn, G. (2007). Cops, robbers and graphs. *Tatra Mt. Math. Paul*, 163-176.

≠ Isaacs, R. (1965). *Differential Games, A mathematical theory with applications warfare and pursuit.* New York: Wiley.

≠ Lucci, S., & Kopec, D. (2016). *ArtificiAl Intelligence in the 21st Century A Living Introduction* (2 ed.). Dulles: Mercury learning And information.

≠ Maddahi, A. (2012). *Solving pursuit and evasion problem in multi-agent systems through a linguistics geometry approach.* mathesis.

≠ Maddahi, A., & Masehian, E. (2017). Linguistic Geometry approach for solving cops and robber problem in grid environments. *Information Sciences*.

≠ Moldenhauer, C. (2009). *Game tree search algorithms for the game of Cops and Robber.* University of Alberta: School of Computing Science.

≠ Stuart. R, Norvig. P. (2009) Artificial Intelligence: A Modern Approach. Prentice Hall.
≠ Sabin, P. (2014). *Simulating War: Studying Cnflict through Simulation Games.* Bloomsbury.
≠ Stilman, B. (1994). Translations of network languages. *Int. J. Computers and Mathematics with Applications, 27*(2), 65-98.
≠ Stilman, B. (1995). Linguistc geometry: methodology and techniques. *Cybernetics and Systems, an International Journal, 26*(5), 535-597.
≠ Stilman, B. (1997). Managing search complexity in linguistic geometry. *IEEE Transactions and Systems, 27*(6), 978-998.
≠ Stilman, B. (2000). *Linguistic Geometry: From Search to Construction.* Springer.
≠ Stilman, B. (2010). *Linguistics Geometry Tools with Demo DVD, LG-PACKAGE.*
≠ Stilman, B. (2010). Linguistic geometry: The age of maturity. *Journal of Advanced Computational Intelligence Informatics, 14*(6), 684-699.
≠ Stilman, B., Yakhnis, V., & Umansky, O. (2010). *Discovering Role of Linguistic Geometry*. Berlin, Heidelberg: Springer.
≠ Suzuki, I. &. (1992). Searching for a mobile intruder in a polygonal region. *SIAM Journal on Computing, 21*(5), 863-888.
≠ Vieira, M. A. (2009). Scalable and practical pursuit and evasion robots. *Intelligent Service Robotics, 2*(4), 247-263.
≠ Washburn, A., & Kress, M. (2009). *Combat Modeling-International Series in Operations Research & Management Science*. Springer.
≠ Winkler, R. N. (1983). Vertext to vertex pursuit in a graph. *Discrete Mathematics.*
≠ Yakhnis, V., & Stilman, B. (2002). Knowledge acquisition and strategy generation with LG wargaming tools. *International journal of computational intelligence and applications, 2*(4), 385-410.
≠ (2020, Jun 11). Retrieved from https://www.RAND.com/