



بهبود رفتار ربات در یک بازی بلادرنگ استراتژیک با استفاده از روش گرگ خاکستری

1- پرهام زیلوچیان مقدم

دانشجوی مهندسی کامپیوتر دانشگاه کاشان

Email: p.zilouchian@gmail.com

2- فرید لطفعلی

دانشجوی مهندسی کامپیوتر دانشگاه کاشان

Email: farid9lotfali@gmail.com

3- جواد سلیمی سرتختی*

Email: salimi@kashanu.ac.ir

چکیده

در این مقاله قصد داریم تا قدرت و نتایج استفاده از روش گرگ خاکستری را در طراحی و توسعه بازی‌های کامپیوتری بسنجیم. به طور خاص در این مقاله به بررسی بازی جنگ سیاره‌ها که برای رقابت هوش مصنوعی گوگل در سال 2010 انتخاب شد پرداخته می‌شود. این بازی به یک بازیکن (عامل) نیاز دارد (در این بازی‌ها به آن‌ها ربات نیز گفته می‌شود) که قادر است با چندین هدف تعامل داشته باشد، وقتی که قرار است به یک درجه‌ای خاص از سازگاری برسد برای این که حریفان متفاوتی را در سناریوها متفاوتی شکست دهند. موتور تصمیم این ربات براساس مجموعه‌ای از قوانین که براساس یادگیری تجربی به دست می‌آید تعریف می‌شود. الگوریتم‌های تکاملی¹ برای تنظیم مجموعه‌ای از مقادیر ثابت، وزن‌ها و احتمالات که قوانین را تعریف می‌کنند استفاده می‌شود و بنابراین عملکرد عمومی ربات را تعریف می‌کنند. الگوریتم گرگ خاکستری یک روش متا هیوریستیک است که برگرفته از سلسله مراتب زندگی گرگ‌های خاکستری و نحوه شکار کردن آن‌ها در طبیعت است. به طور کلی گرگ‌های خاکستری شامل سه مرحله شکار، جستجو برای شکار، محاصره طعمه و حمله به طعمه را شامل می‌شود. در این پژوهش با استفاده از الگوریتم گرگ‌های خاکستری سعی شده است بازی جنگ سیاره‌ها به سطحی از هوشمندی در مواجهه با کاربر برسد.

کلمات کلیدی: جنگ سیاره‌ها، الگوریتم بهینه‌سازی گرگ خاکستری، الگوریتم‌های تکاملی، Google Artificial Intelligence، بازی

1. مقدمه

1-1 معرفی کلی بازی‌های RTS

بازی‌های استراتژی بلادرنگ² [1] یک زیرشاخه از بازی‌های مبتنی بر استراتژی هستند که در آن بازیکنان برای کنترل مجموعه‌ای از منابع، واحدها و سازه‌هایی که در عرصه بازی توزیع می‌شوند، تلاش می‌کنند. یک کنترل و استراتژی صحیح برای رسیدگی به این واحدها برای پیروزی در بازی ضروری است که این پیروزی، به طور معمول از بین بردن تمام واحدهای

¹ Evolutionary Algorithms

² RTS (Real Time Strategy)



دشمن است، یا در بعضی اوقات نیز زمانی که اهداف مشخصی از بازی به دست می‌آیند. ویژگی اصلی آن‌ها طبیعت بلادرنگ¹ آنهاست، یعنی بازیکن مجبور نیست منتظر نتایج بازی‌های دیگر هنگامی که نوبت آنان است، باشد Command RTS . Warcraft TM، Starcraft TM، Conquer TM و Age of Empires TM نمونه‌هایی از بازی‌های RTS هستند.

دو سطح AI معمولاً در بازی RTS در نظر گرفته می‌شود: اولین مورد، که توسط شخصیت غیرتماشایی² [3] تفسیر می‌شود یک ربات است، که تصمیم‌گیری در مورد تمام مجموعه‌های واحد (کارگران، سربازان، ماشین‌آلات، وسایل نقلیه و یا حتی ساختمان‌ها) را بر عهده دارد؛ سطح دوم سطح اختصاصی است که مربوط به پیاده‌سازی رفتار هر یک از این واحدهای کوچک است. این دو سطح اقدامات، که می‌تواند به لحاظ استراتژیک و تاکتیکی در نظر گرفته شود، به طور ذاتی دشوار است که توسط یک انسان طراحی شود؛ اما این مشکل توسط ماهیت واقعی خود معمولاً با محدود کردن زمانی که هر ربات می‌تواند برای تصمیم‌گیری مورد استفاده قرار دهد، افزایش می‌یابد. به همین علت در این مقاله یک روش برنامه‌ریزی ژنتیکی [4] (GP) [5] به عنوان یک روش خودکار برای ایجاد موتور هوش مصنوعی (AI) در یک RTS پیشنهاد شده است. هدف GP ایجاد توابع یا برنامه‌هایی برای حل مشکلات تعیین شده است، که در آن طرز نمایش هر عنصر معمولاً در قالب یک درخت تشکیل شده توسط اپراتورها و متغیرها (پایانه‌ها) است.

هدف از استفاده از GP در این زمینه، ایجاد موتورهای مبتنی بر قانون رفتاری است که به دنبال فرایند اکتشافی، الگوریتمیک و اتوماتیک است. بنابراین، به جای اجرای آن‌ها از ابتدا توسط انسان (متخصص یا غیر متخصص)، این روش مجموعه‌ای از قوانینی را که می‌تواند پیچیده تر (و یا ساده‌تر) از آنچه که توسط انسان تعریف شده، تعریف کند. علاوه بر این، این الگوریتم قادر به ارزیابی هر مجموعه ممکن از قوانین است، و به این مجموعه با توجه به عملکرد ربات مربوطه (در طول جنگ) مقدارهایی را اختصاص می‌دهد. به این ترتیب، این مجموعه‌ها در طول الگوریتم اجرا می‌شود تا بتواند عملکرد ربات را بهبود دهند [6].

2-1- معرفی بازی جنگ سیاره‌ها

مسابقات AI گوگل (GAIC) [2] یک رقابت AI است که در آن افراد شرکت کننده برنامه‌هایی (ربات‌ها) برای رقابت با یکدیگر ایجاد می‌کنند. بازی انتخاب شده برای مسابقه، جنگ سیاره‌ها³، در این مقاله مورد مطالعه قرار گرفته است به این سبب که موتور رفتاری یک ربات با کارایی بالا طراحی شود. جنگ سیاره‌ها یک نسخه ساده از Galcon است، از آنجایی که هدف از آن انجام مبارزه‌های ربات است. نسخه مسابقه‌ای این بازی برای دو بازیکن است.

مسابقه جنگ سیاره‌ها بر روی یک نقشه انجام می‌شود که شامل چندین سیاره است، روی هر یک از آنها یک عدد قرار دارد که نشان دهنده تعداد فضاپیماهایی است که میزبان آن است (شکل 1 را ببینید). در یک مرحله زمانی معین از بازی، هر سیاره تعداد خاصی از فضاپیما دارد و ممکن است متعلق به بازیکن، دشمن و یا خنثی باشد (یعنی به هیچ کس تعلق ندارد). مالکیت با یک رنگ نشان داده می‌شود، آبی برای بازیکن، قرمز برای دشمن، و خاکستری برای خنثی (شخصیت غیر بازی). علاوه بر این، هر سیاره دارای سرعت رشدی است که نشان می‌دهد که چه تعداد فضاپیما در طول هر مرحله از عملیات تولید می‌شود و به ناوگان فضاپیماهای بازیکن که مالک آن سیاره است اضافه می‌شود.

¹ (real time)

² NPC



هدف بازی این است که تمام سیارات حریف تسخیر شود. اگر چه جنگ سیاره‌ها یک بازی RTS (استراتژی) است، پیاده‌سازی آن را به یک بازی نوبتی تبدیل کرده است، و هر بازیکن دارای حداکثر تعداد چرخش برای رسیدن به هدف است. بازیکن با ستاره‌های بیشتر در پایان مسابقه برنده می‌شود [7, 8].

هر سیاره دارای خواص است: مختصات X و Y ، شناسه مالک سیاره، تعداد فضایی‌ها و نرخ رشد. بازیکنان ناوگان فضایی‌ها را برای تسخیر سیارات دیگر (یا تقویت خود) ارسال می‌کنند، و هر ناوگان همچنین دارای مجموعه‌ای از خواص: شناسه مالک، تعداد فضایی‌ها، Source، شناسه مقصد می‌باشد. زمان نوبت شبیه‌سازی شده یک ثانیه است و ربات فقط حداکثر در این زمان یک لیست اقدامات بعدی را انجام می‌دهد.

علاوه بر این، یک ویژگی از این مسئله این است که ربات قادر به ذخیره هر گونه دانش در مورد اقدامات خود در نوبت‌های قبلی، اقدامات حریف خود و یا نقشه بازی نیست. به طور خلاصه، در هر بار از شبیه‌سازی (منظور در هر نوبت)، ربات دوباره با یک نقشه ناشناخته مانند یک بازی جدید روبرو می‌شود. این ناتوانی در ذخیره‌سازی دانش در مورد گیم پلی ایجاد این ربات را به یک چالش جالب تبدیل می‌کند [9].

در حقیقت، هر ربات به عنوان یک تابع اجرا می‌شود که به عنوان ورودی لیست سیارات و ناوگان (وضعیت فعلی بازی)، که هر کدام از ورودی‌ها مقادیر خاص خود را دارند و ربات بر اساس آن‌ها یک سری اقدامات انجام می‌دهد [10]. در هر شبیه‌سازی، یک بازیکن باید جایی را برای ارسال ناوگان از فضایی‌ها، خروج از یکی از سیارات بازیکن و رفتن به سیاره دیگر بر روی نقشه انتخاب کند. این تنها نوع اقداماتی است که ربات مجاز به انجام آن است. ناوگان‌ها می‌توانند گام‌هایی برای رسیدن به مقصد خود بردارند. هنگامی که ناوگان به یک سیاره می‌رسد، با نیروهای موجود دشمن مبارزه می‌کند و در صورتی که تعداد واحدهای دشمن از تعداد واحد‌های فضایی‌های بازیکن کمتر باشد، بازیکن مالک آن سیاره می‌شود. اگر سیاره در حال حاضر به بازیکن تعلق دارد، ناوگان ورودی به عنوان تقویت به آن افزوده می‌شود. در هر زمان گام، نیروها در هر سیاره متعلق به بازیکن یا دشمن (نه سیاره‌های "خنثی") با توجه به سرعت رشد این سیاره افزایش می‌یابد. بنابراین، هدف طراحی یک تابع است که وضعیت نقشه را در هر شبیه‌سازی در نظر بگیرد و اقدامات لازم را برای به دست آوردن مزیت بیشتر نسبت به دشمن، و در نهایت، برنده شدن در بازی را تعیین کند [11].

2. الگوریتم گرگ‌های خاکستری

یک الگوریتم متاهیوریستیک که خود زیر دسته الگوریتم‌های تکاملی هستند، است. الگوریتم‌های تکاملی زیر مجموعه‌ای از محاسبات فرگشتی است و در شاخه هوش مصنوعی قرار می‌گیرد و شامل الگوریتم‌هایی جهت جستجو است که در آن‌ها عمل جستجو از چندین نقطه در فضای جواب آغاز می‌شود.

علت استفاده از الگوریتم گرگ‌های خاکستری در این پژوهش شباهت بسیار زیاد نحوه عملکرد بازی جنگ سیاره‌ها با نحوه عمل این الگوریتم است به گونه‌ای که به راحتی می‌توان این الگوریتم را با جنگ سیاره‌ها مطابقت داد. در این بخش در ابتدا الهام بخشی روش گرگ خاکستری مطرح می‌شود. و سپس مدل ریاضی آن بیان می‌شود. گرگ خاکستری به خانواده سگ سانان تعلق دارد. گرگ‌های خاکستری از بهترین شکارچیان به حساب می‌آیند به این معنی که آن‌ها در راس هرم غذایی هستند. گرگ‌های خاکستری بیشتر ترجیح می‌دهند تا به صورت گروهی زندگی کنند. سائز گروه آن‌ها به طور متوسط بین 5 تا 12 است. به عنوان یک مثال گرگ‌های خاکستری یک سلسله مراتب بسیار سخت‌گیرانه مبتنی بر برتری دارند که در شکل 1 نشان داده شده است. رهبران گروه نرها و یا ماده‌هایی هستند که آلفا نامیده می‌شوند. آلفاها عموماً مسئول هستند تا در مورد شکار، مکان خواب، زمان بیدار شدن و ... تصمیم‌گیری کنند و تصمیمات آلفاها به اعضای گروه دیکته می‌شود.



هرچند که بعضی رفتارهای دموکراتیک نیز مشاهده میشود که در آنها آلفاها سایر گرگ‌های دسته را دنبال میکنند. در گردهمایی‌های آن‌ها سایر گرگ‌ها، آلفاها را با بالابردن دست خود تصدیق میکنند. همچنین گرگ آلفا گرگ مافوق نیز نامیده میشود چون دستوراتش باید به وسیله دسته دنبال شود. گرگ‌های آلفا همچنین فقط اجازه دارند تا در دسته خودشان جفت‌گیری کنند. نکته‌ای که جالب است این است که گرگ آلفا لزوماً قوی‌ترین عضو گروه نیست اما بهترین عضو از لحاظ مدیریت گروه است. این موضوع نشان می‌دهد که تشکیلات، نظم و انضباط گروه از قدرت و زور بسیار مهم‌تر است. سطح بعدی سلسله مراتب گرگ‌های خاکستری، بتا هست. بتا گرگ‌های زیردست هستند آلفا را در تصمیم‌گیری و سایر فعالیت‌های گروهی یاری میدهند. گرگ‌های بتا میتوانند نر یا ماده باشند، و در حالتی که یکی از گرگ‌های آلفا بمیرد یا پیر شود بتا میتواند بهترین کاندید برای آلفا شدن باشد. گرگ بتا باید به آلفا احترام بگذارد، اما به سایر گرگ‌های زیر دست درون دسته دستور میدهد. بتا نقش یک نصیحت‌کننده را برای آلفا و نقش نظم‌دهنده را برای دسته بازی میکند. بتا همچنین دستورات آلفا را در گروه پخش میکند و به اطلاع همه می‌رساند و فیدبکی که از گروه دریافت میکند را به اطلاع آلفا می‌رساند.

پایین‌ترین رتبه در دسته گرگ‌های خاکستری امگا نام دارد. امگا نقش قربانی را بازی می‌کند. امگا باید همواره گرگ‌های مافوق خود را تصدیق کند. آن‌ها آخرین گرگ‌هایی هستند که اجازه دارند غذا بخورند. شاید این‌طور به نظر برسد که امگا به تنهایی اهمیت چندانی در گروه نداشته باشد، اما مشاهده شده است که کل گروه با درگیری‌های داخلی و مشکلات زیادی موقع از دست دادن امگا مواجه میشود. این موضوع به خاطر تخلیه خشونت، ناامیدی و ناکامی تمامی گرگ‌ها به وسیله امگا(ها) است. این موضوع به رضایت‌مندی تمام دسته و پایداری ساختار تسلط در گروه کمک بسیاری میکند. در بعضی از مواقع نیز گرگ‌های امگا نقش نگهداری از بچه‌های گروه را بر عهده دارند. اگر یک گرگ آلفا، بتا یا امگا نباید در نتیجه اوان گرگ، گرگ تابع نامیده میشود (و در بعضی از منابع به آن‌ها دلتا نیز گفته می‌شود). گرگ‌های دلتا باید آلفاها و بتاها را تایید کنند اما آنها بر امگاها غالب هستند و بر آنها حکم فرما هستند. دیدبانان، ارشدها، مراقبان، کشیک‌ها، محافظان و شکارچیان به این دسته تعلق دارند. دیدبانان وظیفه دارند تا مرزهای قلمرو را دیدبانی بدهند و به گروه را در صورت خطر هشدار بدهند. نگهبانان از گروه محافظت می‌کنند و سلامت گروه را تضمین میکنند. ارشدها گرگ‌های باتجربه‌ای هستند قرار است آلفا یا بتا شوند. شکارچیان موقع شکار طعمه گرگ‌های آلفا و بتا را کمک می‌کنند و برای گروه غذا فراهم می‌کنند. و مراقبان نیز وظیفه دارند تا به ضعیف‌ها، زخمی‌ها و بیماران گروه کمک کنند و یا آن‌ها را حمل کنند.

علاوه بر سلسله مراتب اجتماعی گرگ‌ها، شکار گروهی یک رفتار اجتماعی جالب گرگ‌های خاکستری است. فازها و قسمت‌های اصلی شکار گرگ‌های خاکستری عبارت‌اند از :

- پیگیری، تعقیب و نزدیک شدن به طعمه
 - دنبال کردن، محاصره، خسته کردن و آزار و اذیت طعمه تا از حرکت باز بایستند.
 - حمله کردن به سمت طعمه
- این مراحل در شکل 2 نشان داده شده‌اند.

سلسله مراتب اجتماعی گرگ‌های خاکستری

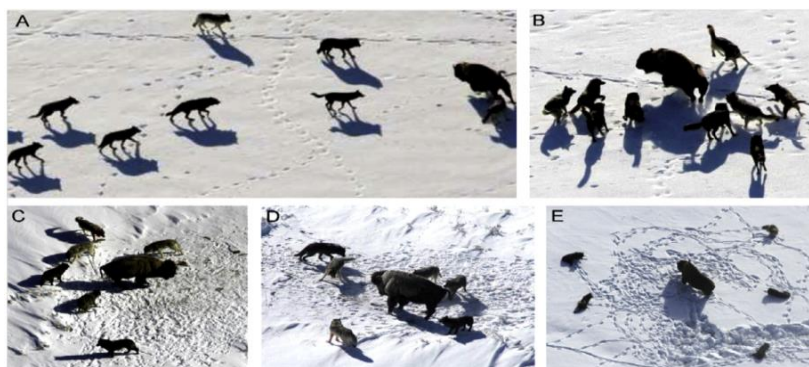
برای این که به صورت ریاضی سلسله مراتب اجتماعی گرگ‌ها را موقعی که الگوریتم گرگ خاکستری را طراحی می‌کنیم، مدل کنیم، ما مناسب‌ترین راه‌حل را به عنوان آلفا (α). در نتیجه دومین و سومین بهترین راه‌حل به دست آمده بتا (β) و دلتا



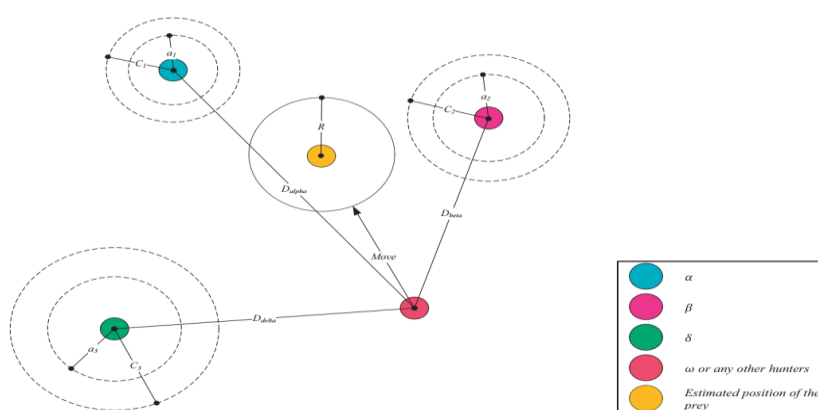
(δ) معرفی می‌شوند. و مابقی راه‌حل‌های کاندیدا به عنوان امگا (ω) معرفی می‌شوند. در الگوریتم گرگ خاکستری شکار (بهینه‌سازی) به وسیله α ، β و δ انجام می‌شود. و گرگ‌های ω این سه دسته گرگ را دنبال می‌کنند.

حمله به طعمه

همان‌طور که در بالا اشاره شد، گرگ‌های خاکستری شکار را با حمله کردن به طعمه هنگامی که از حرکت باز می‌ایستد تمام می‌کنند. برای این که به صورت ریاضی نزدیک شدن به طعمه را مدل کنیم، ما مقدار \vec{a} را کاهش دادیم. توجه کنید که بازه نوسان \vec{A} نیز به وسیله \vec{a} کاهش می‌ابد. به بیان دیگر \vec{A} یک مقدار تصادفی در بازه $[-2a, 2a]$ است که a مقدارش از 2 تا 0 در هر تکرار کاهش می‌ابد. وقتی که مقادیر تصادفی در بازه $[-1, 1]$ هستند، موقعیت بعدی یک عامل جستجو می‌تواند در هر موقعیتی بین موقعیت فعلی و موقعیت قرارگیری طعمه باشد [12].



شکل 1: رفتار شکاری گرگ‌های خاکستری: (A) دنبال کردن، نزدیک شدن و ردیابی طعمه (B-D) تعقیب کردن، آزار و اذیت و محاصره کردن (E) متوقف کردن طعمه و حمله کردن [15]



شکل 2: به‌روزرسانی موقعیت در GWO [15]

با متغیرهایی که تاکنون مطرح شده‌اند، الگوریتم GWO به عامل‌های جستجویش اجازه می‌دهد تا موقعیتشان را براساس محل قرارگیری آلفا، بتا و دلتا به‌روزرسانی کنند؛ و به سمت طعمه حمله کنند. هرچند، الگوریتم GWO به ایستایی در



راه‌حل‌های محلی با این متغیرها تمایل دارد. این موضوع درست است که روش محاصره کردن بیان شده اکتشاف را تا حدی نشان می‌دهد، اما GWO به متغیرهای بیشتری نیاز دارد تا به اکتشاف اهمیت بدهد [13].

جستجو برای طعمه (اکتشاف)

گرگ‌های خاکستری بیشتر بر طبق موقعیت آلفا، بتا و دلتا جستجو می‌کنند. آن‌ها از یک دیگر دور می‌شوند تا به جستجوی طعمه بپردازند و به یکدیگر نزدیک می‌شوند تا به طعمه حمله کنند. برای این که دور شدن را به صورت ریاضی مدل کنیم، ما \vec{A} را با مقادیر تصادفی بیشتر از 1 یا کمتر از -1 به کار گرفتیم تا عامل جستجو را وادار کنیم که از طعمه دور شود. این موضوع به اکتشاف تاکید می‌کند و به الگوریتم GWO اجازه می‌دهد تا به صورت عمومی‌تر جستجو کند. شکل 5(b) همچنین نشان می‌دهد که $|A| > 1$ گرگ‌های خاکستری را مجبور می‌کند تا از طعمه دور شوند به این امید که یک طعمه مناسب‌تری را پیدا کنند.

شبه کد الگوریتم GWO در شکل 6 معرفی شده است.

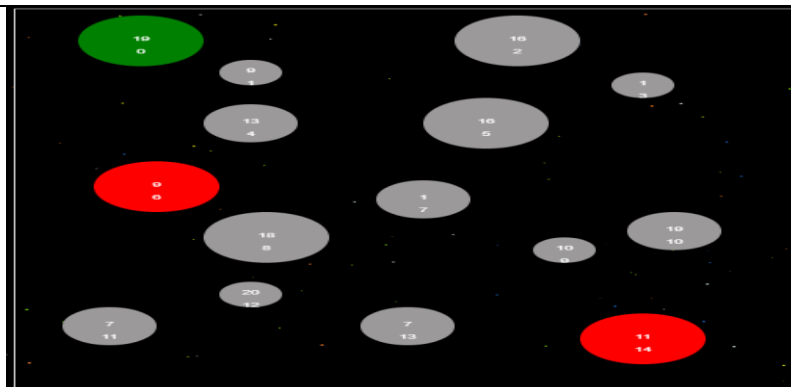
```

Initialize the grey wolf population  $X_i$  ( $i = 1, 2, \dots, n$ )
Initialize  $a$ ,  $A$ , and  $C$ 
Calculate the fitness of each search agent
 $X_\alpha$  = the best search agent
 $X_\beta$  = the second best search agent
 $X_\delta$  = the third best search agent
while ( $t < \text{Max number of iterations}$ )
    for each search agent
        Update the position of the current search agent by equation (3.7)
    end for
    Update  $a$ ,  $A$ , and  $C$ 
    Calculate the fitness of all search agents
    Update  $X_\alpha$ ,  $X_\beta$ , and  $X_\delta$ 
     $t = t + 1$ 
end while
return  $X_\alpha$ 
    
```

شکل 3: شبه کد الگوریتم GWO

3. عملیات ربات گرگ خاکستری

همان‌طور که در الگوریتم گرگ‌خاکستری ما توضیح داده شد در اینجا نیز از شیوه شکار گرگ‌های خاکستری برای طعمه، برای گرفتن سیاره‌ها استفاده می‌شود. همان‌طور که در شکل 6 مشاهده می‌شود در نسخه طراحی شده سه نوع سیاره وجود دارد که در اینجا سیاره قرمز که دشمن محسوب می‌شود، در آن از الگوریتم گرگ خاکستری استفاده شده است. سیاره‌های خاکستری رنگ، سیاره‌هایی هستند که هنوز تسخیر نشده‌اند و همچنین سیاره‌های سبز رنگ، سیاره‌هایی هستند که بازیکن آن‌ها را در اختیار دارد.



شکل 4: در این شکل سیاره‌های قرمز رنگ که در آن از گرگ خاکستری استفاده شده است و همچنین خاکستری که هنوز مالکیت ندارند و سیاره سبز که در اختیار بازیکن است را می‌توانید مشاهده کنید.

در ادامه به توضیح ساز و کار و عملکرد گرگ خاکستری را در این بازی توضیح می‌پردازیم.

4. نحوه عملکرد

همان‌طور که پیش‌تر نیز به آن اشاره شد در بازی جنگ سیاره‌ها تیمی (از بین تیم‌های قرمز و سبز) برنده می‌شود که بتواند تمامی سیاره‌های دشمن را بگیرد و سیاره‌ای را در اختیار دشمن باقی نگذارد. ما در این جا پس از چند بار انجام بازی به این نتیجه رسیدیم که برای پیاده سازی الگوریتم گرگ خاکستری در این بازی راهی که وجود دارد این است که بیاییم و الگوریتم را به شکلی پیاده سازی کنیم که بسته به شرایط هوش مصنوعی (سیاره قرمز رنگ) بیاید و در صورتی که ما هنوز سیاره‌ای را نگرفته‌ایم سیاره‌های خاکستری (سیاره‌هایی که هنوز مالکیتی ندارند) با بزرگترین اندازه و کمترین تعداد نیروی پیش فرض درون آن‌ها را تصرف کند زیرا این سیاره‌ها تولید نیروی بیشتری برای ما در مقایسه با سیاره‌های کوچکتر دارند. و در صورتی که تیم مقابل هوش مصنوعی (سیاره‌های سبز رنگ) اقدام به تصرف سیاره‌ای کرده بودند بیاید و با بررسی آن سیاره و نیز سیاره‌های بدون مالکیت (خاکستری رنگ) باقی مانده ببیند کدام یک بهره‌وری بیشتری دارد و بیاید و آن یکی را تصرف کند. و کلاً هدف ما با استفاده از این الگوریتم این است که در کمترین زمان ممکن بتوانیم سیاره‌ها را در اختیار بگیریم و بر بازیکن انسانی پیروز شویم [14].

در این استراتژی به این صورت عمل می‌کنیم که از آن سیاره‌ای که در ابتدای بازی در آن قرار داریم نیروها را به سمت سیاره دشمن هدایت نماییم و سعی در تصرف این سیاره اولیه می‌کنیم. به دلیل آنکه دشمن سیاره اولیه خود را برای تصرف دیگر سیاره‌ها خالی کرده است تصرف آن راحت‌تر به نظر می‌رسد. سپس به کمک سیاره‌های جدید و نیروهای آن شروع به محاصره مابقی سیاره‌هایی می‌کنیم که دشمن در آن قرار دارد. برای این منظور از سیاره‌های خود به سمت سیاره‌ای از دشمن که نزدیک‌تر است و اندازه بزرگ‌تری دارد (به این علت که تولید نیرو در سیاره‌هایی که اندازه بزرگ‌تری دارد، بیشتر است) نیرو ارسال می‌کنیم. پس از تصرف یکی از سیارات به این شکل ادامه می‌دهیم تا دشمن در یک محدوده محاصره شود و چنانچه دشمن سعی در تصرف سیاره‌ای که خارج از محدوده ما است نمود نیروها را به آن سیاره ارسال کرده و آن



سیاره را از او پس می‌گیریم و به همین صورت حلقه محاصره را بر او تنگ می‌کنیم و از تمام سیارات نیروها را به سیارات داخل محاصره خود ارسال می‌کنیم.

این استراتژی منطبق بر استراتژی گرگ خاکستری است. گرگ‌های خاکستری نیز در شکار طعمه خود به این صورت عمل می‌کنند که گرگ‌های آلفا حمله را رهبری می‌کنند و سایر گرگ‌ها از او تبعیت می‌کنند و حلقه محاصره را تنگ‌تر کرده و اجازه تحرک طعمه را از او می‌گیرند تا طعمه در مسیری که آن‌ها می‌خواهند حرکت کند و پس از خسته کردن طعمه (به دلیل عدم توانایی مانور طعمه) حمله نهایی را آغاز می‌کنند.

5. توضیح الگوریتم به کار برده شده

در ادامه ما انواع توابعی را که در این مسئله به کار برده‌ایم را توضیح می‌دهیم:

در ابتدای آغاز شروع به کار بازی، شروع می‌کند به ساختن یک جمعیت اولیه به صورت کاملاً تصادفی در فضای مشخص شده از سیاره‌ها و این سیاره‌ها در سه اندازه شامل: کوچک، بزرگ و متوسط هستند. پس از این کار شروع به محاسبه شایستگی هر یک از عضوهای جمعیت براساس اساس معیاری که برای آن تعریف کرده‌ایم، می‌کند. تمامی مقادیری که در ادامه اشاره می‌کنیم تا زمانی که یکی از طرفین قرمز یا سبز در بازی پیروز نشود، ادامه پیدا می‌کند. اول از همه که در یک فاصله زمانی مشخصی، مقادیر نیروهای موجود در داخل هر سیاره را بسته به اندازه‌ای که دارد محاسبه می‌کند و سپس یک تابعی اجرا می‌شود که برنده شدن یکی از طرفین را بررسی می‌کند و در صورتی که هر یک از طرفین پیروز شده بود بازی خاتمه پیدا می‌کند. سپس به ازای هر تغییری که در چینش نیروها صورت می‌گیرد تابع fitness برای محاسبه شایستگی تمام اعضا موجود فراخوانی می‌شود تا سیاره بعدی‌ای را برای حمله به آن مشخص کند و تابع شایستگی براساس تعداد نیروهایی که در آن است و اندازه سیاره و میانگین فاصله‌ای که از نیروهای ما دارد محاسبه می‌شود. پس از محاسبه تابع شایستگی حمله انجام می‌شود و تمامی سیارات نیروهای خود را به سیاره‌ای که بیشترین شایستگی را دارد ارسال می‌کنند. سپس مجدداً موقعیت‌ها را به روزرسانی کرده و شایستگی‌های جدید را محاسبه می‌کند و به مرحله بعدی اجرای الگوریتم می‌رود.

Initialize the planets populations $X_i (i = 1, 2, 3, \dots, n)$

Calculate the fitness of each planet for attack at the start of the game $F_i (i = 1, 2, 3, \dots, n)$

Initiate the assault, based on the fitness of the planets that we have calculated earlier

While (none of the players have won)

Update the plenty of the forces in each planet

Calculate and update the new fitness based on the new situation

Choose the best planet for attack

Initiate the attack

Check for winner

End while

Return the winner

شکل 5: شبه کد پیاده سازی انجام شده از بازی Planet Wars

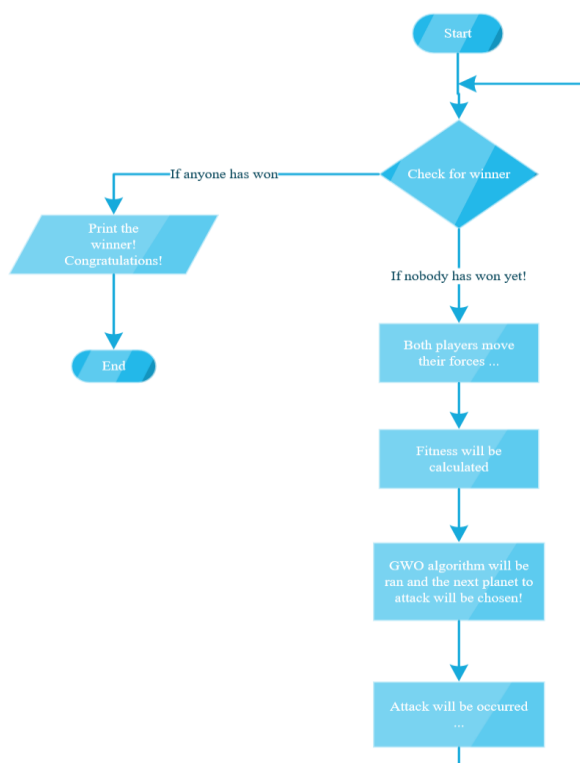


6. پیاده‌سازی بازی جنگ‌های سیاره‌ای (Planet Wars)

ما این بازی را برای این که بتوانیم به آن ظاهر گرافیکی خوبی بدهیم و پیاده‌سازی آن نیز زمان زیادی نگیرد، تصمیم گرفتیم که با استفاده از HTML, CSS, JavaScript پیاده‌سازی کنیم. خوبی این زبان‌ها این است که پیاده‌سازی سریعی دارند و به ما ظاهر خوبی را در کمترین زمان می‌دهند و قابلیت انعطاف بالایی دارند و نیز می‌توان آن را به سادگی در مرورگر وب هر سیستمی بدون نیاز به نصب نرم‌افزار خاصی اجرا کرد.

ما نیز کدهای نوشته شده را در گیت‌هاب که یک سیستم مدیریت محتوا متن باز است بارگذاری کردیم به آدرس گفته شده در پایین صفحه¹ که به راحتی می‌توانید به آن دسترسی داشته باشید.

پس از شروع بازی مشاهده میکنیم که بازیکن قرمز رنگی که با الگوریتم گرگ خاکستری کار می‌کند به سرعت شروع به اجرا شدن و عمل می‌کند و سیاره‌های بدون مالکیت با تعداد نیروهای کمتر را تصرف می‌کند و به محض این که ما نیز اقدامی را انجام بدهیم و سیاره‌ای را درصدد تصرفش برپاییم، سریع وارد عمل می‌شود و آن سیاره را از ما می‌گیرد و به ما اجازه نمی‌دهد که قلمرومان را گسترش دهیم. هم‌چنین دائماً حلقه محاصره را بر ما تنگ‌تر می‌کند و در لحظه آخر نیز حمله نهایی را به ما آغاز می‌کند و سیاره اصلی طرف مقابل (سبز رنگ‌ها) را می‌گیرد و بازی را می‌برد.



شکل 6: فلوچارت نحوه عملکرد بازی جنگ سیاره‌ها با استفاده از الگوریتم GWO



ما در حدود 30 باری که بازی کردیم فقط یک بار تونستیم از هوش مصنوعی تقویت شده با گرگ خاکستری برنده شویم و تمام سیاره‌های قرمز رنگ را بگیریم و این نکته بهبود بسیار خوبی را در مقایسه با روش‌های پیشین استفاده شده مانند ربات Google AI و نیز AresBot و نیز GeneBot نشان میدهد [15]. که برای این که بتوانید بازی را در نسخه موبایل که با این روش‌ها پیاده سازی شده است ببینید پیشنهاد می‌کنم که از طریق لینک موجود در پانویس¹ آن را دانلود کنید (البته این لینک برای سیستم عامل اندروید است)

7. نتیجه گیری

در این مقاله ما توانستیم در بازی جنگ سیاره‌ها از الگوریتم گرگ خاکستری استفاده کنیم و نتایج خوبی را نیز در این رابطه به دست آوریم. چون نحوه عملکرد الگوریتم گرگ خاکستری و شرایط آن در این بازی به خوبی مشاهده می‌شود در نتیجه این الگوریتم هم به خوبی از پس انجام این بازی برمی‌آید. در پیاده‌سازی که انجام دادیم ما توانستیم یک بازی‌ای را طراحی کنیم که حریف مقابل با استفاده از الگوریتم بهینه سازی گرگ خاکستری می‌تواند با سرعت مطلوبی از ما برنده شود. ما در این پیاده سازی خوبی به کارگیری روش‌های متا-هیورستیک در پیاده سازی بازی‌ها را نشان دادیم. هر چند نیاز است تا برای استفاده بهتر و بهینه‌تر کارهای بسیاری انجام شود. در ادامه کار خود ما می‌خواهیم به پیاده‌سازی این بازی با استفاده از سایر روش‌ها بپردازیم و سپس یک مقایسه‌ای بین آن روش‌ها و روش گرگ خاکستری انجام دهیم تا بتوانیم به صورت بسیار دقیق‌تری کارایی این روش را بسنجیم.

8. مراجع

1. A. Fern´andez-Ares, P.G.1-S.a., A.M. Mora, P.A. Castillo, and J.J. Merele, *Designing Competitive Bots for a Real Time Strategy Game using Genetic Programming*.
2. A. Fern´andez-Ares, A.M.M., J.J. Merele, P. Garc´ia-S´anchez and C. Fernandes, *Optimizing Player Behavior In A Real-Time Strategy Game Using Evolutionary Algorithms*
3. Coello Coello CA, M.M.E., *Constraint-handling in genetic algorithms through the use of dominance-based tournament selection*. 2002.
4. Mora, A., *Effect of Noisy Fitness in Real-Time Strategy Games Player Behaviour Optimisation Using Evolutionary Algorithms*. 2012.
5. Liang J, S.P., Deb K, *Novel composition test functions for numerical global optimization*. In: *Swarm intelligence symposium*. 2005.
6. He Q, W.L., *An effective co-evolutionary particle swarm optimization for constrained engineering design problems*. 2007.
7. Dorigo M, B.M., Stutzle T. Ant colony optimization, *Comput Intell Magaz,IEEE* 2006.

¹ <https://play.google.com/store/apps/details?id=com.chris.pwars>



چهارمین کنفرانس ملی و دومین کنفرانس بین‌المللی
«بازی‌های رایانه‌ای؛ فرصت‌ها و چالش‌ها»



اسفند 1397

8. Beni G, W.J., *Swarm intelligence in cellular robotic systems. In: Robots and biological systems: towards a new bionics?* 1993.
9. Wolpert DH, M.W., *No free lunch theorems for optimization. Evolut Comput, IEEE Trans* 1997.
10. Bonabeau E, D.M., Theraulaz G. *Swarm intelligence, from natural to artificial systems.* 1999.
11. Kirkpatrick S, J.D., Vecchi MP, *Optimization by simulated annealing.* 1983.
12. Basturk B, K.D., *An artificial bee colony (ABC) algorithm for numeric function optimization. In: IEEE swarm intelligence symposium.* 2006.
13. Olorunda O, E.A., *Measuring exploration/exploitation in particle swarms using swarm diversity. In: Evolutionary computation.* 2008.
14. Mirjalili S, L.A., *S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization.* 2013.
15. van den Bergh F, E.A., *A study of particle swarm optimization particle trajectories.* 2006.