

## روشی نوین برای محاصره هدف توسط یک سیستم چندعاملی

### مبثنی بر رویداد در بازی های رایانه ای

بهروز مینایی بیدگلی<sup>۱</sup>، علیرضا صحاف نائینی<sup>۲</sup>، عماد آقاجانی<sup>۳</sup>، مهرداد آشتیانی<sup>۴</sup>

<sup>۱</sup>استاد دانشگاه علم و صنعت ایران، مدیر عامل بنیاد ملی بازیهای رایانه ای، B\_minaei@iust.ac.ir

<sup>۲</sup>دانشجوی کارشناسی دانشگاه علم و صنعت ایران، آزمایشگاه تحقیقاتی بازی سازی دانشگاه علم و صنعت ایران، alisahaf70@comp.iust.ac.ir

<sup>۳</sup>دانشجوی کارشناسی دانشگاه علم و صنعت ایران، آزمایشگاه تحقیقاتی بازی سازی دانشگاه علم و صنعت ایران، Emad\_Aghajani@comp.iust.ac.ir

<sup>۴</sup>دانشجوی دکتری دانشگاه علم و صنعت ایران، آزمایشگاه تحقیقاتی بازی سازی دانشگاه علم و صنعت ایران، m\_ashtiani@comp.iust.ac.ir

### چکیده

صنعت بازی های رایانه ای امروزه به یکی از پرطرفدارترین و سرگرم کننده ترین صنایع موجود در جهان تبدیل شده است. یکی از اصلی ترین عوامل ایجاد هیجان و جذابیت در بازی های رایانه ای عنصر هوش مصنوعی بازی است. سعی سازندگان بازی های رایانه ای استفاده هرچه بیشتر از این عنصر به منظور بهبود روند بازی است. هوش مصنوعی بازی ها خود شامل چندین بخش مانند شبیه سازی جمعیت، کنترل وسایل نقلیه و غیره است، که در این مقاله، یک سیستم هوشمند برای تعقیب و محاصره و در نهایت دستگیری یک هدف (عامل خارجی) طراحی و پیاده سازی گردیده است. یکی از مهمترین چالش ها در طراحی این دسته از سیستم ها، که اغلب چندعاملی هستند، تعیین محل مناسب هر عامل برای هرچه بهتر رسیدن به هدف و محاصره آن است. مدل پیشنهادی ارائه شده از یک سیستم رخداده محور و چند سیستم پیش بینی کلی تشکیل شده است و سعی دارد به کمک اطلاعات رسیده از هر عامل، بهترین چیدمان برای قرار دادن عامل ها را پیش بینی کند.

کلمات کلیدی: هوش مصنوعی، سیستم های چندعاملی، هوش مصنوعی مرکزی، هوش مصنوعی توزیع شده.

### ۱. مقدمه

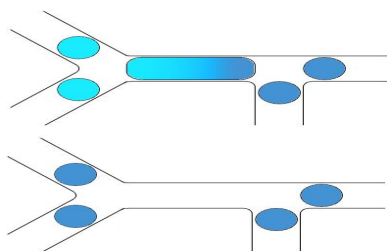
شود. فرض می شود عامل هادر ابتداب ر اساس مسیری که برایشان از قبل تصویر شده است در حال گشت زنی هستند و با دیدن هدف، به تعقیب او می پردازند. مسیریابی [۸] عامل ها با استفاده از سیستم تورجسم ناوبری [۹] و الگوریتم [۹A\*] انجام می گیرد. هر عاملی که به تعقیب بازیکن بپردازد ابتدا به فرمانده (هوش مصنوعی مرکزی)، موقعیت و وضعیت خود را گزارش می دهد سپس فرمانده بر مبنای اطلاعات رسیده، عامل های مفید را از بقیه جدا می کند. (منظور از عامل مفید، عاملی است که توانایی کمک به تعقیب بازیکن را داشته باشد. برای مثال خیلی دور نباشد یا در ماموریت دیگری قرار نگرفته باشد). سپس با محاسبه فاصله واقعی هر کدام از عامل های مفید تا بازیکن که با استفاده از الگوریتم A\* بر روی خانه های ساخته شده با سیستم تورجسم ناوبریه دست می آید، تصمیماتی را در خصوص این که هر کدام از عامل ها در کجا قرار بگیرند را اتخاذ می کند و به هر کدام از عامل ها تصمیم متناظر

دانش هوش مصنوعی نقش کلیدی در جذابیت و صنعتی شدن بازی های رایانه ای دارد. در هوش مصنوعی به کارگیری عامل های چندگانه [۱۱] [۲] در بازی هایی که در آنها تعقیب و گریز مطرح است بسیار رایج است و بطور کلی مسایلی که به تعقیب و گریز در دنیای هوش مصنوعی می پردازند در قالب سیستم های چند عامله و با عنوان هایی گوناگون از قبیل "Cows" [۳-۷] "Herders" مطرح می شوند. در این مدل از هوش مصنوعی، یک عامل مرکزی وجود دارد که نقش فرماندهی و ارتباط یک عامل با عامل های دیگر را ایفا می کند و یک هوش توزیع شده [۲] که هر کدام از عامل ها را در انتخاب تصمیم ها یاری می کند. همچنین تعدادی عامل در اختیار داریم که هر کدام از آن ها در مکان های مختلفی از نقشه قرار گرفته اند. وظیفه کلی این سیستم، دستگیری بازیکن هدفی است که در مکانی نامشخص (از قبل تعیین نشده) از نقشه قرار دارد. بازیکن آزادانه می تواند در هر مسیری که بخواهد جابه جا

<sup>1</sup>Navigation Mesh

پیشنهادی، رؤسگراف به منزله یک نقطه اطلاع رسانی هستند، در این روش از نگرش پیشامد محور<sup>۳</sup> در انجام محاسبات فوق استفاده می کنیم. به این معنا که در هر بار عبور فرد در حال تعقیب از یک گذرگاه، محاسبات فوق یکبار انجام می شود.

از آنجاکه مسیر پیشروی فرد در دست تعقیب رابهکمک<sup>۲</sup> راساخری که وی از آن هاعبور کرده است، محاسبه می کنیم، الزام مهمی پدید می آید که این دوراس باید متوالی باشند. از طرف دیگر، تنها در صورتی راداعبور از یک رأس را پردازش می کنیم که یکی از عامل های در حال تعقیب، شخص در حال فرار را در آن لحظه ببیند، در غیر این صورت مانند دنیای واقعی هیچ رخدادی صادر نمی شود. همچنین در ادامه سعی می کنیم رابهنه سازی گراف، تغییراتیدرانتخابرئوسداشته باشیم. بدین منظور سعی می کنیم راساس فاصله دوررئوس متوالی واهمیتناحیه بین دوراس (که نماینده یک یال در گراف، و یک مسیر در نقشه است) رؤس را با یکدیگر ادغام کنیم. در شکل ۲، دو نمونه از نحوه بهینه انتخاب رؤس شکل ۱ آورده شده است.



شکل ۲ نحوه بهینه انتخاب رؤس

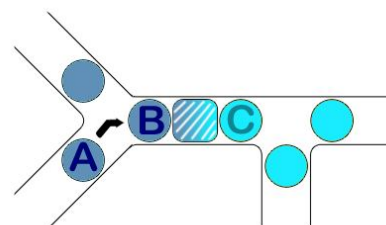
### ۳. درخت پیش بینی مسیر

همان طور که در بخش ۲ به درخت پیش بینی اشاره شد، در هر بار عبور از رؤس، یک پردازش برای بررسی محل بعدی فرد در حال گریز انجام می دهیم. همچنین همزمان با به روزرسانی اطلاعات فوق، یک درخت پیش بینی مسیر نیز تشکیل می دهیم. علت همزمانی این دو عملیات، عدم امکان تغییر مسیر در بین دو رأس است. یعنی پس از اینکه فرد در حال تعقیب از یک گذرگاه عبور کرد، تا رسیدن فرد به رأس بعدی، تغییر مسیری نمی تواند صورت گیرد که این فرصت مناسبی را برای تشکیل درخت مهیا می کند. (شایان ذکر است که گذر از رؤس که منجر به ورود به یک ناحیه چند راهیمی شود به شکل دیگری بررسی می شود که از بیان این حالت خاص صرف نظر می کنیم). هدف از تشکیل درخت فوق که به نوعی از گراف اصلی تولید شده ولی به شکل خاص استفاده می کند، به دست آوردن احتمال حرکت فرد در حال گریز به محل های

خودش را ارسال می کند. پس از دریافت دستور، عامل وظیفه اطاعت از آن را بر عهده خواهد داشت. هدف اصلی فرمانده محاصره کردن بازیکن و بستن راه های فرار وی است و این کار را با فرستادن عامل ها به انتهای مسیرهایی که احتمال حضور بازیکن در آن ها بیشتر است انجام می دهد.

### ۲. سیستم ساختن نقشه

سیستم پیاده سازی شده به منظور دریافت و مدل سازی محیط اطراف خود، زمین و راه های قابل راهروی در آنرا، تحت قالب یک گراف راهروی مدل سازی می کند. در این مدل، راس ها به محل های ورود و خروج اطلاق می شود.



شکل ۱ یافتن مسیر عامل

به عنوان مثال در شکل ۱، نقاط مشخص شده نشان دهنده یک رأس در گراف ایجاد شده هستند (به این دلیل که محلول و دیالیز خروجی کمسیر جدید هستند) و خطوط درون شکل نشان دهنده یال های باشند. در این گراف، یال AB نشان می دهد که یک مسیر از رأس A به B و بالعکس وجود دارد که نیاز به عبور از هیچ رأس دیگری ندارد. استفاده از این شیوه مدل سازی به کمک کند که مستقلاً مختصات مکانی هدف و تنها به کمک آگاهی از آخرین رؤسی که از آن ها عبور کرده، محل نسبی فرد را تشخیص دهیم. همچنین به کمک روشی که در زیر معرفی می شود، قادر به تشخیص تغییر مسیرها خواهیم بود. به این منظور گراف را به کمک ماتریس مجاورت<sup>۴</sup> [۱۰] پیاده سازی کرده ایم. در نتیجه، اگر فرض کنیم که می دانیم فرد در حال تعقیب، بهترین باز رأس های A و B عبور کرده باشد، مشخص می شود که این فرد در حال حرکت به سمت رأس C است و در ناحیه هاشور خورده مشخص شده در شکل ۱ است. اینکار (تشخیص مکان بعدی و در نتیجه آن، محاسبه فرد) به کمک تفريق مجموعه رؤس مجاور B از مجموعه رؤس مجاور رأس A به دست می آید، این کار سبب خواهد شد که بار پردازشی بسیار کمی بر روی پردازنده قرار گیرد. از آنجا که در مدل

<sup>3</sup> Reference

<sup>4</sup> Event Based

<sup>۲</sup> Adjacent Matrix

عنوان مثال، یک مکان که دارای آیتم هایی برای پنهان شدن بازیکن باشد، نسبت به یک مسیر که به بن بست ختم می شود، مناسب تر در نظر گرفته می شود و از ارزش بالاتری برخوردار خواهد بود.

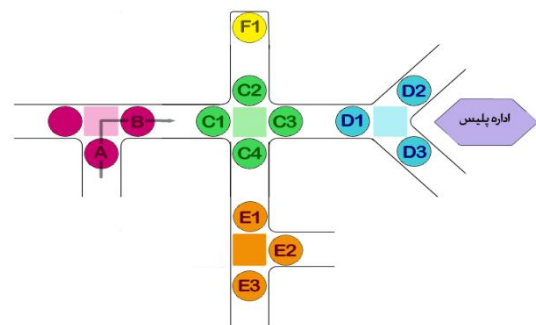
۲. **تاریخچه مسیرهای انتخابی:** در این قسمت، یک سیستم ارزش دهی جداگانه برای گذرگاه ها در نظر گرفته می شود که بر اساس مشاهدات عامل های تعقیب کننده در حین تعقیب و بر اساس انتخاب های فرد در دست تعقیب، ارزش دهی می شود. به عنوان مثال، اگر بازیکن چند بار در یک دوراهی از مسیر سمت راست برود و این کار توسط عامل ها دیده شود، باعث می شود رأس انتهایی مسیر سمت راست از ارزش بیشتری نسبت به رأس موجود در مسیر سمت چپ، برخوردار شود.

در شکل ۵ نمای نقشه ای که نگارندگان در موتور بازی سازی [۱۱ UDK] برای انجام آزمایش های خود پیاده سازی کرده اند نشان داده شده است.

#### ۴. آزمایش هوش مصنوعی فردی

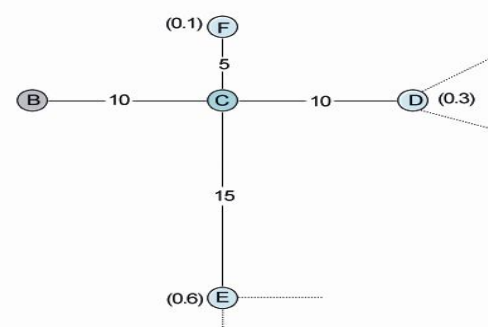
سیستم هوش مصنوعی فردی با استفاده از ساختار ماشین حالت متناهی<sup>۵</sup> [۱۲] ساخته شده است. در این سیستم، هر عامل دارای چندین حالت است که با توجه به شرایط و فرمان هایی که از سیستم هوش مرکزی می گیرد، وارد یکی از این حالت ها می شود. در شکل ۶ حالت های مختلف در نظر گرفته شده

پیش روی او است. نتیجه ای که از این درخت برای ما اهمیت دارد، به دست آوردن مکان های مناسب برای قرار دادن عامل ها و بستن راه ها است. برای نمونه و بر اساس شکل ۳، رئوس C1 تا C4 از نظر مفهومی نمایانگر یک منطقه قابل پوشش توسط نیروهای تعقیب کننده است، پس در درخت خود، برای چنین رئوسی، یک رأس رابه عنوان نماینده برای قرارگیری عامل استفاده می کنیم.



شکل ۳ منطقه های قابل پوشش توسط عامل تعقیب کننده

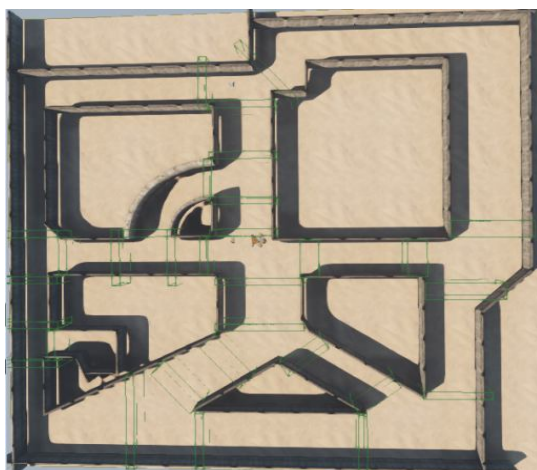
نحوه تشکیل این درخت را با یک مثال نشان می دهیم. اگر فرض کنیم فرد در دست تعقیب به ترتیب از گذرگاه های A و B گذشته باشد، برای عمق ۲، درختی مانند شکل ۴ خواهیم داشت:



شکل ۴ مثال درخت جستجو در عمق ۲

در این درخت وزن یال ها نمایانگر طول مسیر و وزن برگ ها نشان دهند احتمال آن مکان است (احتمال آنکه فرد در حال فرار به آنجا بگریزد). در محاسبه این احتمالات، ما ۲ معیار اصلی را مورد توجه قرار دادیم:

۱. **میزان خطر گذرگاه ها:** این معیار به صورت دلخواه ارزش دهی می شود و به شرایط فیزیکی یک رأس و عواملی از این دست بستگی دارد. به



شکل ۵ نقشه مسیر ها و گذرگاه های آزمایش

و در ادامه توضیح قسمت های مهم آن را مشاهده می کنید.

<sup>5</sup>Finite State Machine

حالت تصمیم گیری<sup>۷</sup>، که اولین و اصلی ترین حالت عامل و به نوعی واحد تصمیم گیری منطقی عامل است، بر اساس اولویت های تعیین شده در داخل آن تصمیم به رفتن به حالت دیگری می گیرد. اولویت های قرار گرفته در مدل ایجاد شده برای محاصره بازیکن به صورت زیر است:

۱. در صورت داشتن دستوری از فرمانده رفتن به حالت دستورفرمانده.
۲. در صورت دیدن بازیکن و تحت تعقیب بودن آن رفتن به حالت تعقیب.
۳. در صورت داشتن مسیر برای گشت زدن رفتن به حالت گشت زنی.

همچنین اکثر حالت ها پس از انجام کارهای تعیین شده به حالت تصمیم گیری وارد می شوند و به این صورت حلقه اصلی تفکر عامل تشکیل می شود.

#### ۵. سیستم هوش مصنوعی مرکزی

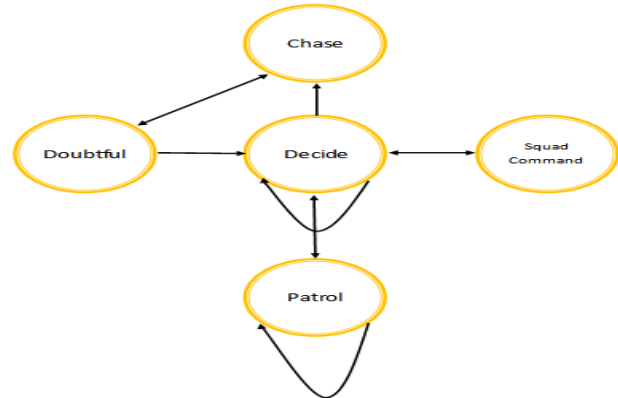
به منظور کنترل و هماهنگی بین نیروهای موجود از هوش جمعی استفاده می کنیم. این واحد مرکزی وظیفه دارد که اطلاعات و مشاهدات تمام نیروها را گرفته و پس انجام پردازش ها، دستور لازم را به عامل ها ارسال کند [۱۳]. از آنجایی که این فرآیند به صورت متمرکز انجام می شود، به این واحد، واحد مرکزی اطلاق می کنیم. این واحد در دو مقطع زمانیه تصمیم گیری می پردازد:

۱. هنگامی که بازیکن در حال تعقیب از یک گذرگاه عبور کند و در عین حال توسط یک عامل قابل مشاهده باشد.
۲. دریافت اطلاعات از عامل ها.

بسته به مقطع زمانی تصمیم گیری دو حالت می تواند اتفاق بیفتد. این دو حالت در ادامه مقاله توضیح داده خواهند شد.

#### الف) حالت اول

حالت اول، همانطور که در بخش های قبل معرفی شد، با ایجاد یک درخت پیش بینی مسیر آغاز می گردد. عمق این درخت بر اساس معیارهای گوناگونی از قبیل تعداد عامل ها و موقعیت عامل ها، تعیین می گردد. در مرحله اول درخت با عمق یک ایجاد می کنیم. نتیجه این عمل، درختی با تنها یک برگ است. بر اساس اطلاعات قرار گرفته در برگ این درخت، امکان بسته شدن مسیر توسط یکی از نیروها برسی می گردد. اگر بتوان عاملی بدین منظور پیدا



شکل ۶ ماشین گذر متناهی از ماشین

#### ۴-۱ حالت تعقیب

حالت تعقیب<sup>۸</sup>، هنگامی رخ می دهد که عامل، کاربر را مشاهده کرده و از طرف فرمانده دستوری نداشته باشد. در این حالت بسته به موقعیت بازیکن و عامل دو حالت می تواند اتفاق بیفتد:

۱. اگر عامل از بازیکن عقب تر باشد و در دو درجه حرکت نکنند (عامل در حالت تعقیب از پشت سر بازیکن) عامل به مکانیک های یک نمرامی بیند می رود و در هر لحظه مکانیک های یک نمرامی و زورسا نیکر دهبه فرمانده ها اطلاع می دهد.
۲. در صورتیکه عامل از پشت سر بازیکن در تعقیب او نباشد (به طور مثال از روبرو یا از طرفین) مکانا احتمالاً به خورد بازیکن توسط فرمول زیر به دست آمد و عامل به موقعیت Position فرستاده شده تا در زمان صحیح به بازیکن رسید و او را دستگیر کند.

$$\Delta S = S_{agent} - S_{Goal}$$

$$\Delta V = v_{agent} - v_{Goal}$$

$$T = \frac{\Delta S}{\Delta V}$$

$$Position = S_{agent} + v_{agent} \times T$$

در صورتیکه در حین مراحل تعقیب، بازیکن از دید عامل ناپدید شد، عامل وارد حالت مشکوک شده و در صورت مشاهده بازیکن به حالت تعقیب باز می گردد.

#### ۴-۲ حالت تصمیم گیری

<sup>7</sup> Deciding State

<sup>8</sup> Chasing State

مرحله اول سعی کنیم که در لایه یاولیه موفقیت برسیم. به این منظور باید به دنبال عاملی باشیم که بتواند قبل از رسیدن بازیکن در حال گریز، خود را به مکان  $K_1$  برساند. ولی از آنجا که برای این لایه تمام اختلاف زمان ها منفی است، مشخص است که امکان چنین چیزی وجود ندارد. پس باید به بررسی لایه بعدی بپردازیم.

موفقیت در هر لایه منوط به بسته شدن تمام برگ های آن لایه است که در مورد لایه دو، بسته شدن مکان های  $K_1-2$  و  $K_2-2$  مدنظر است. همان طور که در جدول مشخص است، اینکار با قرار دادن نیروی  $P_1$  در مکان  $K_2-2$  و یکی از دو نیروی  $P_2$  یا  $P_3$  در مکان  $K_1-2$  قابل انجام است. اگر در شرایطی در این لایه نیز قادر به بستن تمام راه ها نمی شدیم، باید برای مسیرهای بسته نشده، گراف را تا یک لایه بعد گسترش دهیم. روند افزایش لایه ها زمانی خاتمه می یابد که یا در لایه فعلی به موفقیت برسیم و بتوانیم تمامی مکان های آن لایه را (برگ های آن لایه را) پوشش دهیم یا مجموع اختلاف زمان های لایه فعلی از لایه قبلی کمتر شود یا تعداد برگ ها بیشتر از عامل های موجود گردد.

هم چنین به منظور کاهش محاسبات و دریافت نتیجه بهتر، گسترش درخت به صورت تناحیه ای<sup>۸</sup> صورت می پذیرد. به این معنا که اگر در یک لایه قسمتی از مکان ها قابل پوشش و قسمتی غیر قابل پوشش باشند تنها بررسی لایه بعد را از سمت برگ های غیر قابل پوشش ادامه می دهیم و گسترش به صورت تناحیه ای صورت می پذیرد که این کار باعث کاهش چشمگیری در هزینه پردازش های گرددد. در رابطه با نحوه چیدمان عامل ها در مکان ها، چند حالت ممکن است اتفاق بیفتد، که هر یک را جداگانه شرح می دهیم:

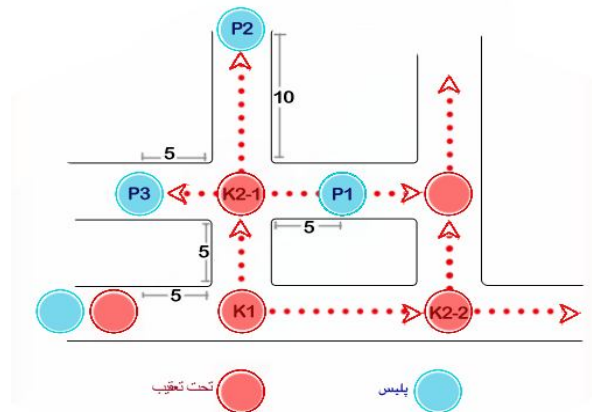
۱. **تعداد عامل ها بیشتر یا برابر تعداد برگ ها باشد:** این حالت به معنی وجود عامل های اضافی است. در این حالت، از عامل ها با چند رویکرد متفاوت و بسته به مکان و فاصله آن ها از فرد در حال گریز و فاصله با سایر برگ ها استفاده می کنیم.

الف) بستن مسیر از پشت (اینکار به منظور پشتیبانی عامل های اصلی بکار می رود)

ب) افزایش عامل ها در مکان های محتمل تر (با توجه به وزن برگ ها)

۲. **تعداد عامل ها کمتر از تعداد برگ ها باشد:** در این حالت، که معمول تر از حالت قبلی است، سعی بر آن است که یک عامل به دو یا چند ناحیه اختصاص داده شود. در این حالت، یک عامل بر اساس میزان احتمال برگ هایی که به وی اختصاص داده شده و فاصله وی از هر یک

کرد مشکل حل شده است، در غیر این صورت تعداد حرکت ها افزایش می یابد. بر اساس عمق درخت و وزن یال ها و مکان فعلی عامل ها جدول شماره ۱ را از شکل ۷ به وجود می آوریم.



شکل ۷ نقشه مسیر ها و گذرگاه های آمایش

فرض کنید تمامی افراد از جمله فرد در حال فرار، بتوانند با سرعت ثابت ۱ واحد بر ثانیه بدونند. در این صورت مدت زمانی که برای پیمودن  $M$  واحد لازم است، برابر  $M$  ثانیه خواهد بود.

جدول ۱ جدول ساختن جدول برای اساس شکل ۷

	K1	K2-1	K2-2
P1	-5	+5	+5
P2	-10	0	-10
P3	-5	+5	-5

با این فرض، خانه های جدول بالا، اختلاف زمان رسیدن فرد در حال فرار به یکی از مکان های حاصل از برگ درخت پیش بینی، نسبت به هر یک از عامل های  $P_1$  تا  $P_3$  را نشان می دهد. یعنی:

$$(1) \Delta t = (Kx - T_{yp}) - (Kx - T_{yA})$$

$$(2) T_{yp} = \text{زمان رسیدن بازیکن به خانه}$$

$$(3) T_{yA} = \text{زمان رسیدن عامل P به خانه}$$

اعداد منفی جدول، نشان دهنده میزان تاخیر عامل ها برای رسیدن به هدف در نقطه مشخص شده است. همانطور که از شکل ۷ و جدول ۱ مشخص است، در

<sup>8</sup> Regional

SHABaN: Multi-agent Team To Herd Cows. In Programming Multi-Agent Systems, Koen V. Hindriks, Alexander Pokahr, and Sebastian Sardina (Eds.). Lecture Notes In Artificial Intelligence, Vol. 5442. Springer-Verlag, Berlin, Heidelberg 248-252

[4] Steen Vester, NiklasSkamriis Boss, Andreas Schmidt Jensen, and JørgenVilladsen. 2011. Improving multi-agent systems using Jason. Annals of Mathematics and Artificial Intelligence 61, 4 (April 2011), 297-307

[5] VahidRafe, Amin Nikanjam, and Mohammad Rezaei. 2011. Galoan: a multi-agent approach to herd cows. Annals of Mathematics and Artificial Intelligence 61, 4 (April 2011), 333-348

[6] GregorBalthasar, Jan Sudeikat, and Wolfgang Renz. 2009. On Herding Artificial Cows: Using Jadex to Coordinate Cowboy Agents. In Programming Multi-Agent Systems, Koen V. Hindriks, Alexander Pokahr, and Sebastian Sardina (Eds.). Lecture Notes In Artificial Intelligence, Vol. 5442. Springer-Verlag, Berlin, Heidelberg 233-237

[7] GregorBalthasar, Jan Sudeikat, and Wolfgang Renz. 2010. On the decentralized coordination of herding activities: a Jadex-based solution. Annals of Mathematics and Artificial Intelligence 59, 3-4 (August 2010), 411-426

[8] In-Cheol Kim. 2006. Real-Time search algorithms for exploration and mapping. In Proceedings of the 10th international conference on Knowledge-Based Intelligent Information and Engineering Systems - Volume Part I (KES'06), BogdanGabrys, Robert J. Howlett, and Lakhmi C. Jain (Eds.)

[9]Xiao Cui; Hao Shi, "A\*-based Path finding in Modern Computer Games" IJCSNS International Journal of Computer Science and Network Security, VOL.11 No.1, January 2011

[10] Godsil, Chris; Royle, Gordon Algebraic Graph Theory, Springer (2001), ISBN 0-387-95241-1, p.164

[11] UDK Unreal Developer's Kit, <http://www.unrealengine.com/udk>, last visit: January 2013.

[12]Anna I. Esparcia-Alcazár, Anais Martínez-García, Antonio M. Mora, J. J. Merelo, and Pablo García-Sánchez. 2010. Genetic evolution of fuzzy finite state machines to control bots in a first-person shooter game. In Proceedings of the 12th annual conference on Genetic and evolutionary computation

[13] Karl Tuyls and Simon Parsons. 2007. What evolutionary game theory tells us about multiagentlearning. Artif. Intell. 171, 7 (May 2007), 406-416

از آن ها، در حد فاصلبرگ ها قرار می گیرد. مثالی از انجام این کار بدین شکل است که فرض کنید یک عامل داریمو دو نقطه باید بسته شود، مدت زمان رسیدن بازیکن به هر کدام از این دو نقطه را محاسبه و سپس بر مبنای کمترین زمان، عامل را در مسیر بین دو نقطه و در فاصله ای از نقطه نزدیک تر به بازیکن قرار می دهیم که اگر از طرف فرمانده متوجه شد که بازیکن وارد کدام مسیر شده است، زمان لازم برای رسیدن به آن نقطه را داشته باشدو خود را به آن مکان برساند.

## ب) حالت دوم

برای کنترل بیشتر و طبیعی شدن رفتار عامل ها، از یک ارتباط دوطرفه بین واحد مرکزی و عامل ها استفاده می کنیم. بدین منظور از مشاهدات عامل ها در موقعیت های مختلف استفاده کرده و پس از بررسی شرایط عامل ها، دستور واحد مرکزی را اجرا خواهیم کرد.

## ۶. نتیجه گیری

در این مقاله، هدف اصلی، ارایه مدلی ساده و در عین حال کارا برای تعقیب و محاصره ی یک هدف خارجی توسط مجموعه ای از عامل ها است. نگارندگان در این مقاله، به منظور هرچه بیشتر طبیعی جلوه دادن مدل هوش مصنوعی و کاهش نیاز به تبادل اطلاعات مابین عامل ها، تاکید بر استفاده از روش رویداد محور، و نه مکان دقیق عامل ها دارند. مدل پیشنهادی فوق، علاوه بر سرعت پردازش بالا، امکان استفاده همزمان از دو روش هوش مصنوعی توزیع شده و مرکزی را به ارمغان می آورد. روش ارائه شده از ترفندهای رایج هوش مصنوعی در بازی های رایانه ای استفاده نمی کند. این حرف به این معنا است که در روش ارائه شده، مکان و مشخصات بازیکن در اختیار هیچ یک از عوامل موجود در دنیای بازی قرار ندارد. این موضوع می تواند منتهی به طبیعی شدن فراوان هوش مصنوعی به کار رفته در بازی شده و محیط پویای رقابتی با سربار پردازشی بسیار کمی را به مخاطبان خود ارائه کند.

## ۷. مراجع

[1]Simon Parsons and Michael Wooldridge. 2002. Game Theory and Decision Theory in Multi-Agent Systems. Autonomous Agents and Multi-Agent Systems 5, 3 (September 2002)

[2] Winikoff, M., Padgham, L.: Developing intelligent agent systems: a practical guide. In: WileySeries in Agent Technology. Wiley, New York, NY (2004)

[3] Adel T. Rahmani, AlirezaSaber, Mehdi Mohammadi, Amin Nikanjam, EhsanAdeliMosabbab, and MonirehAbdoos. 2009.