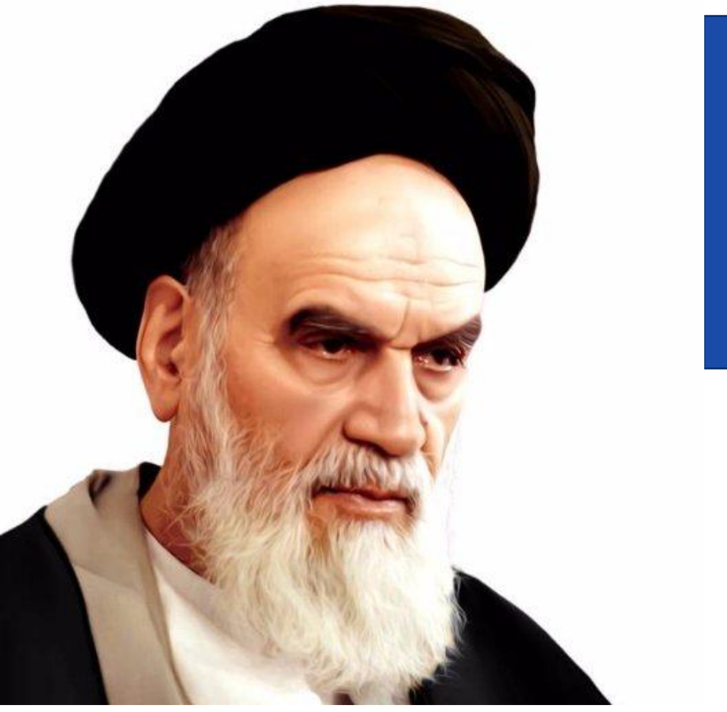


کاربرد الگوریتم ژنتیک در شبیه‌سازی بازی جنگ

نویسنده: میلاد بیاری(کارشناس ارشد هوش مصنوعی، Milad.biyari70@gmail.com)

چکیده

امروزه علم هوش-مصنوعی جایگاه خاصی را در محیط-های نظامی پیدا کرده است و هدف اصلی آن شبیه-سازی رفتار انسان است. این مقاله در ابتدا قصد دارد که یک معرفی جامع از مبانی نظری مبحث الگوریتم-های ژنتیک داشته باشد، سپس یک نگاه کلی به برخی از کاربردهای الگوریتم-های هوش-مصنوعی در بازی-جنگ شده و در مرحله بعد مروری به کاربرد الگوریتم--های ژنتیک در شبیه-سازی بازی-جنگ شده است، در پایان یک مثال از بازی جنگ که با استفاده از الگوریتم ژنتیک شبیه-سازی شده است مورد بررسی خواهد گرفت. در انتها نتایج مورد تحلیل و بررسی قرار گرفته و به نظر می-رسد که می‌توان به‌عنوان یک راه-حل کاربردی از آن استفاده کرد.

مقدمه

تحول-های شدید و سریع محیط امنیتی جهانی باعث شده است تا دنیا با محیطی پر ابهام مواجه باشد. در چنین محیطی پیروزی نیازمند هوشمندی بسیار بالایی است. سؤال اینجاست که بهترین راه برای پیروزی چیست؟ شاید این سخن امیرالمؤنین علی علیه‌السلام رهنمونی است برای پیروزی در این صحنه‌ها :پیروزی در دوراندیشی است، دوراندیشی در به‌کارگیری درست اندیشه ، و بکار گیری درست اندیشه در رازداری است. واقعیت این است که برنامه‌ریزی-های میان‌مدت و بلندمدت در دنیای پیچیده و غیرقابل‌پیش‌بینی امروز، بدون آگاهی ازآنچه که در پیرامون ما می‌گذرد امکان‌پذیر نیست. امروزه دیده‌بانی و پایش محیط پیرامونی یکی از مهم‌ترین ابزار برنامه-ریزی‌های بلندمدت و تصمیم‌گیری برای آینده است. دیده-بانی مستمر ازآن‌جهت اهمیت دارد که تحولات جهانی سریع و پیش‌بینی‌ناپذیر شده‌اند و در صورت نداشتن هوشیاری، موجب غافلگیری می‌شوند. از کاربردهای مهم فرآیند طرح‌ریزی می‌توان به موارد زیر اشاره کرد: تدابیر عملیات مشترک، برنامه‌ریزی برای جلوگیری از درگیری و اقدام-های پس از درگیری، برنامه‌ریزی کلان جنگ‌های شبکه. در این مقاله از الگوریتم‌های هوش مصنوعی برای حل مسئله تصمیم‌گیری در بازی جنگ استفاده شده است. امروزه هوش مصنوعی به‌عنوان ابزاری کارآمد می‌تواند در حل مسائل پیچیده که گاه انسان یا امکان حل آن را ندارد یا اگر قادر باشد به زمانی بسیار طولانی برای حل آن نیاز دارد، مورداستفاده قرار گیرد. با توجه به این‌که به‌طور معمول در مقاله-های علمی موارد تئوری به‌صورت خیلی اجمالی موردبررسی قرار می-گیرد-، این مقاله در ابتدا لازم می-داند مبانی- عملی را مورد بررسی قرار دهد سپس با استفاده از الگوریتم-های هوش مصنوعی که در این مقاله به‌صورت موردی از الگوریتم-های ژنتیک استفاده شده است، یک فضای(مجموعه-)راه-حل ایجاد کرده است. سپس با برنامه‌نویسی ژنتیک پیاده-سازی شده و با استفاده از شبیه‌سازی تبرید (انلینگ) در مسئله بازی جنگ به حالت اجرا درمی‌آید. در انتها نتایج مورد تحلیل و بررسی قرار گرفته و به نظر می-رسد که می-توان به‌عنوان یک راه-حل کاربردی از آن استفاده کرد.

برنامه نویسی ژنتیک

یکی از مسائلی که در علوم کامپیوتر مطرح است، اینست که چگونه به کامپیوتر این قابلیت را بدهیم که تنها به وی بگوئیم که یک کار خاص را انجام دهد، و کامپیوتر آن کار را بدون اینکه به وی اعلام کنیم که چگونه آنرا انجام بده، انجام دهد. برنامه‌نویسی ژنتیک این مشکل را حل کرده است، به این صورت که یک برنامه کامپیوتری برای یک مساله سطح بالا تولید می-کند. برنامه‌نویسی ژنتیک در واقع به هدف ما، یعنی برنامه‌نویسی اتوماتیک با استفاده از پرورش یک جمعیت از برنامه-های کامپیوتری که با استفاده از مفاهیم تئوری داروین (الگوریتم‌های ژنتیک)، پرورش داده می-شوند، می-رسد. پرورش این جمعیت با استفاده از نسخه الهام گرفته از عمل‌های انتخاب و تولید مثل بیولوژیکی صورت می-گیرد. همانطور که در الگوریتم‌های ژنتیک بیان شده این اعمال بازترکیبی و جهش می-باشد، که با ترکیب با اعمال انتخاب والدین و انتخاب بازماندگان به این مهم دست پیدا می-کنند.

مبانی نظری

الگوریتم ژنتیک نوع خاصی از الگوریتم‌های تکامل است که از تکنیک‌های زیست‌شناسی تکاملی مانند وراثت، جهش زیست‌شناسی و اصول انتخابی داروین برای یافتن فرمول بهینه جهت پیش‌بینی یا تطبیق الگو استفاده می‌شود. الگوریتم‌های ژنتیک اغلب گزینه خوبی برای تکنیک‌های پیش‌بینی بر مبنای رگرسیون هستند. در مدل‌سازی الگوریتم ژنتیک یک تکنیک برنامه‌نویسی است که از تکامل ژنتیکی به‌عنوان یک الگوی حل مسئله استفاده می‌کند. مسئله‌ای که باید حل شود دارای ورودی‌هایی می‌باشد که طی یک فرایند الگوبرداری شده از تکامل ژنتیکی به راه‌حل‌ها تبدیل می‌شود، سپس راه‌حل‌ها به‌عنوان کاندیداها توسط تابع ارزیاب مورد ارزیابی قرار می‌گیرند و چنانچه شرط خروج مسئله فراهم‌شده باشد الگوریتم خاتمه می‌یابد. به‌طورکلی یک الگوریتم مبتنی بر تکرار است که اغلب بخش‌های آن به‌صورت فرایندهای تصادفی انتخاب می‌شوند که این الگوریتم‌ها از بخش‌های تابع برازش، نمایش، انتخاب و تغییر تشکیل می‌شوند. الگوریتم‌های ژنتیک یکی از الگوریتم‌های جستجوی تصادفی است که ایده آن برگرفته از طبیعت می‌باشد. در الگوریتم‌های ژنتیک ابتدا به‌طور تصادفی یا الگوریتمیک، چندین جواب برای مسئله تولید می‌کنیم. این مجموعه جواب را جمعیت اولیه می‌نامیم. هر جواب را یک کروموزوم می‌نامیم. سپس با استفاده از عملگرهای الگوریتم ژنتیک پس از انتخاب کروموزوم‌های بهتر، کروموزوم‌ها را باهم ترکیب کرده و جهشی در آن‌ها ایجاد می‌کنیم. درنهایت نیز جمعیت فعلی را با جمعیت جدیدی که از ترکیب و جهش در کروموزوم‌ها حاصل می‌شود، ترکیب می‌کنیم. مثلاً فرض کنید گونه خاصی از افراد، هوش بیشتری از بقیه افرادِ یک جامعه یا کولونی دارند. در شرایط کاملاً طبیعی، این افراد پیشرفت بهتری خواهند کرد و رفاه نسبتاً بالاتری خواهند داشت و این رفاه، خود باعث طول عمر بیشتر و باروری بهتر خواهد بود. حال اگر این خصوصیت (هوش) ارثی باشد بالطبع در نسل بعدی همان جامعه تعداد افراد باهوش به دلیل زاد و ولد بیشترِ این‌گونه افراد، بیشتر خواهد بود. اگر همین روند را ادامه دهید خواهید دید که در طی نسل‌های متوالی دائماً جامعه نمونه ما باهوش و باهوش‌تر می‌شود. بدین ترتیب یک مکانیزم ساده طبیعی توانسته است در طی چند نسل عملاً افراد کم‌هوش را از جامعه حذف کند علاوه بر اینکه میزان هوش متوسط جامعه نیز دائماً در حال افزایش است. بدین ترتیب می‌توان دید که طبیعت با بهره‌گیری از یک روش بسیار ساده (حذف تدریجی گونه‌های نامناسب و درعین‌حال تکثیر بالاتر گونه‌های بهینه)، توانسته است دائماً هر نسل را از لحاظ خصوصیات مختلف ارتقاء بخشد. گونه‌های متکامل‌تری وجود دارند که نمی‌توان گفت صرفاً حاصل تکامل تدریجی گونه قبلی هستند. در این میان آنچه شاید بتواند تا حدودی ما را در فهم این مسئله یاری کند مفهومی است به نام تصادف یا جهش. در هر نسل جدید بعضی از خصوصیات به صورتی کاملاً تصادفی تغییر می‌یابند سپس براثر تکامل تدریجی درصورتی‌که این خصوصیت تصادفی شرایط طبیعت را ارضا کند حفظ می‌شود در غیر این صورت به شکل خودکار از چرخه طبیعت حذف می‌گردد. درواقع می‌توان تکامل طبیعی را به این صورت خلاصه کرد: جستجوی کورکورانه (تصادف یا جست-و-جوی-کوکورانه) + بقای قوی‌تر. حال ببینیم که رابطه تکامل طبیعی با روش‌های هوش مصنوعی چیست. هدف اصلی روش‌های هوشمندِ به کار گرفته شده در هوش مصنوعی، یافتن پاسخ بهینه مسائل مهندسی است. به‌عنوان مثال اینکه چگونه یک موتور را طراحی کنیم تا بهترین بازدهی را داشته باشد یا چگونه بازوهای یک ربات را متحرک کنیم تا کوتاه‌ترین مسیر را تا مقصد طی کند (دقت کنید که در صورت وجود مانع یافتن کوتاه‌ترین مسیر دیگر به سادگی کشیدن یک خط راست بین مبدأ و مقصد نیست) همگی مسائل بهینه‌سازی هستند. روش‌های کلاسیک ریاضیات دارای دو اشکال اساسی هستند. اغلب این روش‌ها نقطه بهینه محلی را به‌عنوان نقطه بهینه کلی در نظر می‌گیرند و نیز هر یک از این روش‌ها تنها برای مسئله خاصی کاربرد دارند. این دو نکته را با مثال‌های ساده‌ای روشن می‌کنیم.

کاربردهای الگوریتم ژنتیک

- روندیابی هیدرولوژیکی رواناب جاری در شبکه رودخانه خشک
- کمک در حل مسایل تصمیم گیری چند معیاره
- بهینه سازی چند هدفه در مدیریت منابع آبی
- بهینه سازی و بارآرایی شبکه های توزیع نیروی برق

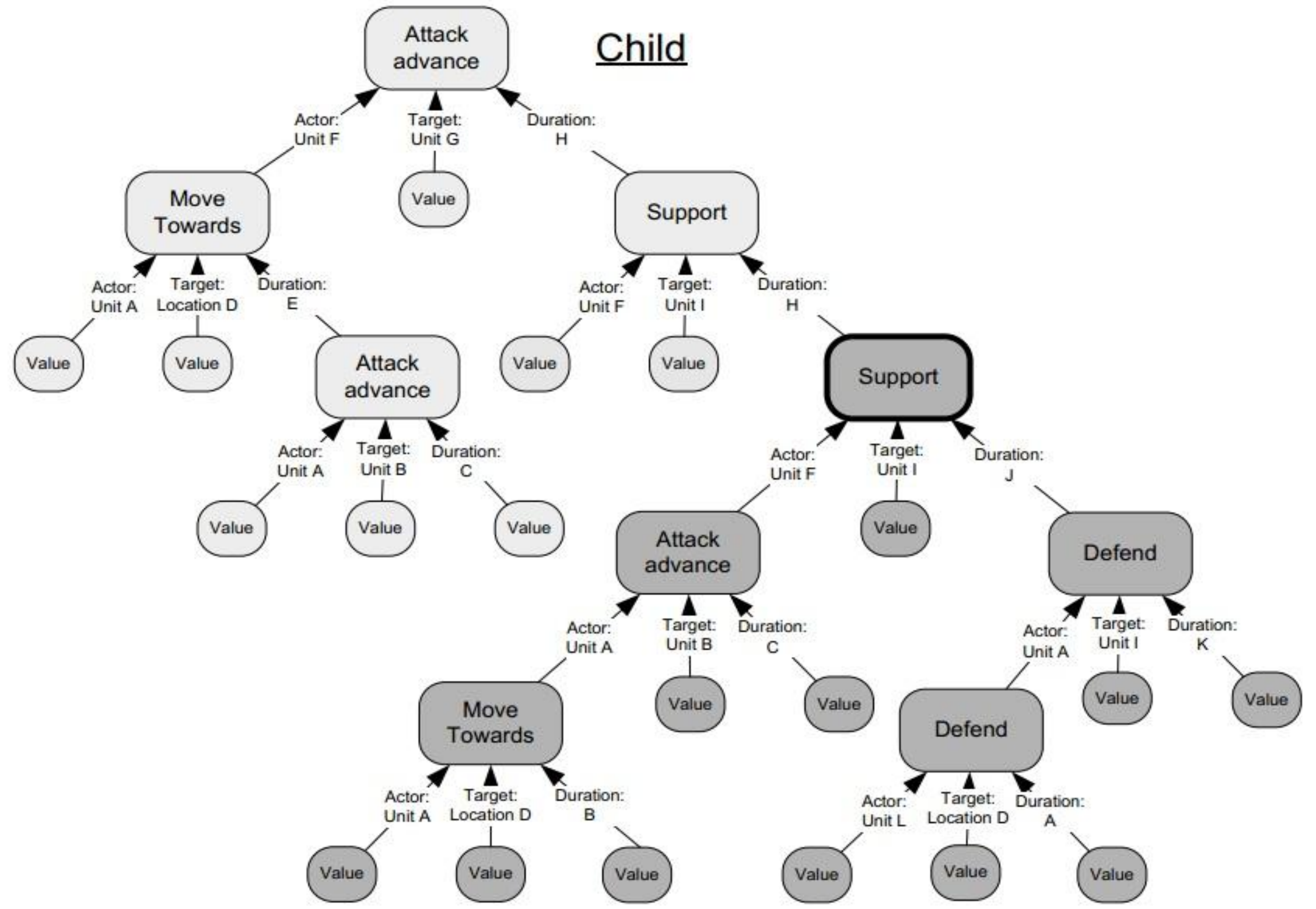
شرایط خاتمه الگوریتم‌های ژنتیک عبارتند از:

- به تعداد ثابتی از نسل‌ها برسیم.
- بودجه اختصاص داده‌شده تمام شود(زمان محاسبه/پول).
- یک فرد(فرزند تولید شده) پیدا شود که مینیمم (کمترین) ملاک را برآورده کند.
- بیشترین درجه برازش فرزندان حاصل شود یا دیگر نتایج بهتری حاصل نشود.
- بازرسی دستی.

روش کار

روش الگوریتم ژنتیک دارای یک توصیف چکیده از راه‌حل انتخاب شده برای یک مسئله به‌عنوان یک موجودیت می-باشد. به‌صورت نرمال یک جریان کوتاه، یک رشته باینری از ۰ و ۱ هاست، سپس همه این-ها به یک راه‌حل مسئله خاص توسط یک فرایند به نام کدگشایی ترجمه شده است. برای یک مسئله بازی جنگ، ۰ و ۱ را می-توانیم به ترتیب (حرکت۰۰۰)،(حمله۰۰۱)،(دفاع۰۱۰)،(پشتیبانی۰۱۱)،(عقب‌نشینی۱۰۰)مشخص کنیم. بیت-های ارائه شده می-توانند به‌عنوان ارزش-های ورودی یک فرمان برای مثال بازیگر یا واحدهای هدف، زمان-بندی-ها، و مناطق و... در نظر گرفته شوند.یک جمله دستوری سرتاسر واحدها در بازی جنگ می-توانند بر پایه روند بیت ساخته شوند. این مجموعه دستور در یک مدل بازی جنگ ارزیابی شده تا یک اندازه مناسب، یک اندازه کمی از چگونگی خوب بودن راه-حل بدست آوریم. یک جمعیتی از نهادها در نظر گرفته شده و به‌صورت تصادفی مقداردهی اولیه شده‌اند. این جمعیت سپس در نسل-ها توسط راه-حل-های پدر و مادر که توسط یک روش کاملاً برازنده شده، نشان‌دهنده فشار تکاملی، رشد کرده است. فرزندان سپس توسط جهش و یا متقاطع جریان-های بیت والدین-ها تولید می-شود. بهترین راه-حل سپس بعد از یک تعداد از نسل-ها به دست می‌آید .انتخاب والدین مبتنی بر امتیاز آن از تابع برازش، که اجازه می-دهد جمعیت به راه-حل-های مناسب-تر تکامل یابد. روش انتخاب می-بایست از گوناگونی ژنتیک در جهت رسیدن به یک راه-حل جامع خوب، نگهداری کند. زیرا یک راه-حل ممکن است امتیاز برازندگی ضعیفی داشته باشد، اما ازآن‌جهت شامل ویژگی-های خوبی باشد که فقط احتیاج به یک تغییر کوچک برای رسیدن به یک نتیجه خوب باشد. انتخاب ترنومتنت یک روش دستیابی به این مهم است. زمانی که یک والدین را انتخاب می-کنیم، یک تعداد کاندید به‌صورت تصادفی انتخاب می-شوند (بدون در نظر گرفتن امتیاز برازششان)، سپس بهترین آن‌ها انتخاب می-شود. این باعث افزایش اطمینان ما می-شود که یک تعداد کوچکی که دارای امتیاز برازندگی بالاتری هستند، اما موجودیت-های مربوطه که همیشه انتخاب نمی-شوند، جمعیت را از بین می‌برند. اندازه برازش هسته تمامی الگوریتم-های بهینه-سازی تصادفی می-باشد. یک اندازه مناسب از برازش به الگوریتم اجازه می-دهد که فشار انتخاب را بکار برده و اطمینان دهد که مناسب-ترین ویژگی-های جمعیت رشد پیدا کند.برای مسئله بازی جنگ اندازه برازش ابتدا توسط کدگشایی کردن یک موجودیت به یک مجموعه دستور گرفته می-شود .این دستورات سپس ارزیابی شبیه-سازی بازی جنگ و نتایج آن را با توجه به شکست-ها و موفقیت-ها (موقعیت-هایی که گرفته شده یا به دشمن داده شده. یا دشمن آن‌ها را از دست داده و یا خنثی‌سازی کرده)، خطرات(مجاورت دشمن که آیا دشمن آن‌ها را پشتیبانی می-کند یا نه) و کارایی (حداقل میزان مصرف منابع) انجام می-دهد .مراقبت ویژه باید در نظر گرفته شود تا اندازه برازش بهتر فهمیده و تحمیل شود، که اغلب مشکل است که به‌عنوان راه‌حل‌های بد در نظر گرفته شود. یک اندازه کمی موفقیت یا برازش درباره مناسب بودن انتخاب بین دو راه-حل احتمالی و اجازه دادن به الگوریتم مسیریابی کند راه-حل-های "بد" را و فضای کل راه-حل را کشف کند، مهم است. بسیار آسان است که با استفاده از نرمالیزه کردن واضح و مشابه همه اندازه-ها و پارامترهای برازش بدست بیاید. بسیار مهم است که توجه کنیم تکنیک-های بهینه-سازی تصادفی برای پیدا کردن روزنه-ها در منطق مدل و استاندارد ارزیابی استفاده شده برای اندازه-گیری برازش، به‌صورت منفی مشهور است. علت این موضوع این است که الگوریتم فهمی از مسئله حل شده نداشته و به‌صورت ساده میان ۰ و ۱ ها تغییر می-کند و محتوایی درباره خوب یا بد بودن راه‌حل نتیجه گرفته شده ندارد. اگرچه این دستیابی روزنه-ها گیج-کننده است. اگر هدف این باشد که بهترین راه‌حل را پیدا کنیم می-تواند مفید نیز باشد. بخصوص با نمایش روزنه-ها یک روش کمکی در تائید و تصدیق معادلات پیچیده را ارائه کند. نکته مهم این است که درحالیکه مثال یک بازی جنگ بصورت مصور نشان داده در یک الگوریتم ژنتیک یه نهاد به سادگی یک گره درخت است. هر گره دارای تعدادی مقادیر ورودی و گره-های فرزند می-باشد. در این الگوریتم یک گره و یا ورودی معنی ندارد بلکه در زمان رمز گشایی هر گره ترجمه می-شود و تعیین می-شود که منتقل شود. در اصل تفاوتی نیست بین نمایش ریاضی یک بیت الگوریتم ژنتیک و یک گره درخت در الگوریتم برنامه-نویسی ژنتیک. با این حال بسیار ساده می-توان رمز گشایی کرد و فرم عمومی آن را به یک راه-حل برای مسئله خاص ترجمه کرد. کارایی الگوریتم برای هر روش بهینه-سازی تصادفی حیاتی است. الگوریتم-های ناکارآمد فضای راه حل را کشف نمی-کنند زیرا همگرایی به یک راه حل خوب نیاز به یک احتمال خوب برای ایجاد راه-حل بهتر از هر نسل دارد در غیر این صورت پیشرفت-های بالقوه از بین می-روند. الگوریتم ناکارآمد فقط به سادگی به یک راه حل "خوب" نزدیک می-شود؛ ممکن است هرگز همگام نشود. یک الگوریتم کارآمد برای تولید فرزندان دارای قابلیت اندازه-گیری سازگار باید اجازه دهد برای مقایسه بین موجودیت-ها در انتخاب مناسب عملی تکامل یافته. راه-حل "بد" از لحاظ الگوریتمی قادر به جستجو در کل فضای راه-حل هستند که مهم است. با این حال، باید اندازه-گیری دقیق از چگونگی "بد" بودن این راه-حل-ها باشد، به طوری که الگوریتم می-تواند به راه-حل-های بهتر حرکت کند. بنابراین همیشه راه-حل-های حیاتی وجود دارد برای رمزگشایی (برای مثال مجموعه مرتب) که می-تواند قابلیت اندازه-گیری داشته باشد و هر گونه تغییر در موجودیت(در شکل عمومی آن) به یک تغییر معنادار در راه-حل ترجمه میکند و تغییر میدهد. این رمزگشایی کارآمد برای یک درخت گره/ورودی یک الگوریتم برنامه-نویسی ژنتیکی بسیار ساده تر از آن است که برای یک جریان کمی استفاده می-شود. با این حال، لازم به ذکر است که تفاوت بین تکنیک برنامه-نویسی ژنتیکی و الگوریتم ژنتیک کمی با یک رمزگشای خوب نوشته نشده است. اگر همیشه یک نهاد ترجمه شود به یک مجموعه مرتب قابل قبول ، مهم نیست که چگونه این نهاد تغییر می کند. بنابراین نمی-توان انتظار داشت که الگوریتم برنامه نویسی ژنتیکی با سایر الگوریتم-های ژنتیکی متفاوت باشد.

یک مثال



شکل ۲ : درخت ادغام تصادفی والدین برای مسئله بازی جنگ

یک مثال برای شرح کاربرد الگوریتم هوش-مصنوعی درشبیه-سازی بازی-جنگ- شرح سناریو:

بازیکنان: آبی (دوست)؛ قرمز (دشمن)

هدف بازی: شکستن محاصره توسط بازیکن آبی

تعداد و نوع کشتی‌های بازیکن آبی: تابعی از زمان

راهبرد نیروی قرمز: ثابت

گزارش هزینه‌ها: هزینه‌های مربوط به خرید و از دست دادن کشتی‌ها در طول جنگ

مدل‌سازی: بر اساس شبیه‌سازی مرحله زمان

تعیین کارایی سیستم‌های سلاح و سنسورها: به‌صورت احتمالی

تابع برازندگی

سود برازندگی= (وزن زمان *(ماکزیمم سود ممکن / سود)* هزینه زمان بازی)- (وزن نیرو * (ماکزیمم سود ممکن / سود * هزینه نیروی آبی))

سود : درصد پیشروی انجام شده به سمت هدف ؛ ماکزیمم سود ممکن ۱۰۰ فرض می‌شود.

روش‌های محاسبه سود : روش معیار ؛ روش شمارش

هزینه نیروی آبی : خرید و از دست دادن کشتی‌ها

هزینه زمان : تعداد مراحل زمانی یک آزمایش

وزن-های نیرو و زمان : اهمیت نسبی در عبارت هزینه در عبارت هزینه که توسط کاربر مشخص می-شود.

قالب الگوریتم ژنتیک :

- اندازه جمعیت ۳۰
- عملگر باز ترکیب
- جهش
- متوسط اجرا برای یک فرد در یک نسل ۱۰۰
- آزمایش‌ها تا ۵۰۰۰ نسل
- احتمال جهش ۰.۰۰۱
- آزمایش ممکن است با و بدون نخبه‌گرایی انجام شود.
- بهترین قالب استفاده از نخبه‌گرایی با روش شمارش است.

نتایج مثال مذکور:
نتایج برای سه آزمایش نشان داده شده که در آن یا وزن هزینه یا معیار پیروزی تغییر می‌کند. نتایج ،راهبرد مختلفی را برای انجام بازی نشان می‌دهد که شامل برخی شباهت‌ها و اختلافات مهم تولید شده توسط تغییر شرایط آزمایش هستند.

نتیجه گیری

این نشان داده شده است که الگوریتم-های هوش-مصنوعی ، مانند شبیه‌سازی تبرید و برنامه-نویسی ژنتیک می-توانند به‌طور مؤثر برنامه-هایی برای مشکلات تاکتیکی تولید کنند، در این صورت فعالیت-های سطح زمین را در بر می-گیرند و برنامه-هایی تولید می-کنند که شبیه تصمیم-گیری-های انسانی است. دو عنصر مهم برای این موفقیت بوده است. برای استفاده از الگوریتم-ها با استفاده از دستورالعمل نظامی، اول، استفاده از راه-حل-ها است. ثانیاً، استفاده از الگوریتم هوش-مصنوعی به‌طورکلی، مجموعه ابزار توانسته است یک معماری "plug-and-play"را استفاده کند. این امکان استفاده از یک مدل متا را فراهم می‌کند که به هوش مصنوعی اجازه می-دهد تا برنامه-هایی را در برابر یک مجموعه است مشکل را کاهش داده ، و فقط عناصر ضروری مشکل کامل را تولید می-کند. سپس راه‌حل تولید شده می-تواند بر اساس مشکل کامل (به‌عنوان‌مثال SimBrig)ارزیابی شود. این روش اجازه می-دهد که مدل متا ساده، سریع و قوی باشد و از محدودیت-های سنتی تکنیک-های هوش-مصنوعی استفاده شود که نیازمند ارزیابی بسیاری از برنامه-ها است و همچنین تمایل به استفاده از خطاهای موجود در منطق مدل-های ارزیابی دارد. مدل متا همچنین یک رویکرد موفق به شبیه-سازی بازی جنگ نشان داد. با تمرکز بر مدل-سازی فقط آنچه برای مسئله موردنظر موردنیاز است، به یک سطح مناسب از جزئیات، ایجاد یک شبیه-سازی جامع و قابل اعتماد از تعامل زمین در سطح تیپ را از ابتدا، به دست آمده که در بودجه و زمان محدود است. برنامه-ریز عملیاتی اجازه می‌دهد تا اکتشاف یک منطقه بزرگ‌تر از فضای راه‌حل بالقوه را از طریق سندهای رفتار انسانی بررسی کند. به‌طور خاص، به طیف گسترده-ای از واکنش-های ممکن قرمز برای دوره-های خاصی عمل می-کند و درک ارزش اطلاعات را بهبود می-بخشد. گام بعدی طرح-ریزی نقشه را در یک مدل قرار می-دهد تا توانایی آن در برنامه-ریزی در یک وضعیت پویا را بررسی کند.

منابع

- Y. Azaria and M. Sipper (2005). GP-gammon: Genetically programming backgammon players. Genetic Programming and Evolvable Machines, 6(3):283–300, September 2005a. ISSN 1389-2576.
- Y. Azaria and M. Sipper (2005). Using GP-gammon: Using genetic programming to evolve backgammon players. In M. Keijzer, et al., editors, Proceedings of the 8th European Conference on Genetic Programming, volume 3447 of Lecture Notes in Computer Science, pages 132–142, Lausanne, Switzerland, 30 March - 1 April 2005b. Springer. ISBN 3-540-25436-6.
- S.G. Lucek, NSC & S.J. Collander-Brown, Dstl : Artificial intelligence algorithms and new approaches to wargame simulationInternational Symposium on Military Operational Research (ISMOR 31), July 2014.
- D. Jackson (2005). Evolving defence strategies by genetic programming, In M. Keijzer, et al., editors, Proceedings of the 8th European Conference on Genetic Programming, volume 3447 of Lecture Notes in Computer Science, pages 281-290, Lausanne, Switzerland, 30 March - 1 April 2005. Springer. ISBN 3-540-25436-6.
- J. Lohn, G. Hornby, and D. Linden (2004). Evolutionary antenna design for a NASA spacecraft. In U.-M. O’Reilly, et al., editors, Genetic Programming Theory and Practice II, chapter 18, pages 301–315. Springer, Ann Arbor, 13-15 May 2004. ISBN 0-387-23253-2.
- J. R. Koza (1992). Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge, MA, USA, 1992. ISBN 0-262-11170-5.
- J. R. Koza, F. H. Bennett, III, and O. Stiffelman (1999). Genetic programming as a Darwinian invention machine. In R. Poli, et al., editors, Genetic Programming, Proceedings of EuroGP’99, volume 1598 of LNCS, pages 93–108, Goteborg, Sweden, 26-27 May 1999. Springer-Verlag. ISBN 3-540-65899-8.
- S. Luke (1998). Evolving soccerbots: A retrospective. In Proceedings of the 12th Annual Conference of the Japanese Society for Artificial Intelligence, 1998. (The simulation of teams of softbot programs in simulated soccer matches)
- N. Metropolis, et al. (1953). "Equation of State Calculations by Fast Computing Machines". The Journal of Chemical Physics 21 (6): 1087

نخستین همایش ملی واکاوی تهدیدهای نوپدید دفاعی - نظامی

دافوس آجا - آبان ماه ۱۳۹۷

دانشگاه فرماندهی و ستاد آجا