

## به نام خدا

موضوع پروژه: Squid Game Optimizer (SGO): یک الگوریتم فراابتکاری جدید

نام اعضای گروه: سارا مظاهری – فائزه رحمتی

نام استاد: جناب آقای دکتر علی جمالیان

### مقدمه:

در این مقاله، Squid Game Optimizer (SGO) به عنوان یک الگوریتم فراابتکاری جدید با الهام از قوانین اولیه یک بازی سنتی کره ای پیشنهاد شده است.

بازی Squid یک بازی چند نفره با دو هدف اصلی است: هدف مهاجمان تکمیل هدف خود در حالی که تیم ها سعی می کنند یکدیگر را حذف کنند و معمولاً در زمین های باز و بزرگ بدون دستورالعمل تعیین شده برای اندازه و ابعاد بازی می شود.

زمین بازی برای این بازی اغلب شبیه یک ماهی مرکب است و با توجه به زمینه تاریخی، به نظر می رسد تقریباً نصف یک زمین بسکتبال استاندارد باشد.

برای ارزیابی اثربخشی الگوریتم SGO پیشنهادی، از 25 تابع آزمون ریاضی بدون محدودیت با 100 بعد، در کنار شش فراابتکاری رایج دیگر برای مقایسه استفاده می شود.

100 اجرای بهینه سازی مستقل برای SGO و سایر الگوریتم ها با شرایط توقف از پیش تعیین شده برای اطمینان از اهمیت آماری نتایج انجام می شود.

معیارهای آماری مانند میانگین، انحراف معیار، و میانگین ارزیابی تابع هدف مورد نیاز محاسبه می شوند.

مسائل بهینه سازی دنیای واقعی تقریباً در هر زمینه وظایف کاملاً چالش برانگیز و مسائل پیچیده در نظر گرفته می شوند و آنها را در دسته های متفرقه، از جمله محدود یا غیرمحدود، تک یا چند هدفه، پیوسته یا گسسته، و ایستا یا پویا طبقه بندی می کنند.

توسعه برخی الگوریتم های مبتنی بر انسان از شبیه سازی رفتارهای انسانی مختلف، از جمله بهینه سازی مبتنی بر آموزش 51(TLBO)، الگوریتم فرهنگی 52(CA)، و جستجوی هارمونی 53(HS) الهام گرفته شده است.

TLBO بر اساس فعالیت های آموزشی و یادگیری بین معلمان و دانش آموزان توسعه یافته است.

در این تحقیق، یک الگوریتم بهینه سازی فراابتکاری مبتنی بر جمعیت جدید به نام Squid Game Optimizer (SGO) پیشنهاد شده است.

با استفاده از این رقابت‌ها، هدف این مطالعه بهبود ارزیابی الگوریتم‌های فراابتکاری و افزایش اثربخشی آنها در حل مسائل بهینه‌سازی دنیای واقعی است.

کاربرد این الگوریتم در برخورد با انواع مسائل بهینه‌سازی چالش اصلی این الگوریتم مبتنی بر جمعیت است که در آن کاندیداهای راه‌حل به دو گروه بازیکنان تهاجمی و تدافعی تقسیم می‌شوند در حالی که بازیکن تهاجمی در میان بازیکنان تدافعی می‌رود تا بازی را شروع کند. مبارزه ای که از طریق یک حرکت تصادفی به سمت بازیکنان تدافعی مدل سازی می‌شود.

### الگوریتم بهینه بازی مرکب

بازی Squid که با نام ojingo نیز شناخته می‌شود، بر اساس یک بازی زمین بازی کودکان کره ای است و اساساً ترکیبی از تگ و هاپسکاچ است.

در همین حال، هیچ توصیه یا دستورالعملی در مورد اندازه و ابعاد زمین بازی وجود ندارد.

بر اساس تاریخچه بازی ماهی مرکب، به نظر می‌رسد که پلی فلد به شکل ماهی مرکب تقریباً نصف یک زمین بسکتبال معمولی است.

بخش B دقیقاً در مرکز خط افقی در امتداد پایین ماهی مرکب قرار دارد و باید به اندازه کافی جادار باشد تا بسیاری از شرکت کنندگان در آنجا بایستند، که در اصلی یا دروازه ورودی به زمین بازی ماهی مرکب است.

این بازی با ایستادن در بخش A شروع می‌شود. با این حال، بازیکنان تهاجمی نمی‌توانند وارد بخش C شوند و بلافاصله بازی را برنده شوند.

با این حال، زمانی که هر بازیکنی از بخش‌های D، A، B یا E خارج می‌شود، این مزیت تدافعی به پایان می‌رسد. مشابه بازیکنان تهاجمی، بازیکنان تدافعی باید با یک پا بیرون از زمین بپرند و تیم آنها زمانی برنده می‌شود که با موفقیت تمام بازیکنان‌های تهاجمی را حذف کنند.

### نحوه پیاده سازی الگوریتم

روش SGO به تفصیل در زیر توضیح داده می‌شود، در حالی که فلوچارت و کد مجازی در تصاویر 2 و 3 نشان داده شده است.

مرحله 1: موقعیت اولیه حل‌های پتانسیلی، که به عنوان (Xi) یا "بازیکنان" در فضای جستجو نمایش داده می‌شوند، از طریق یک فرآیند انتخاب تصادفی تعیین می‌شوند.

- مرحله 2: بازیکنان به دو گروه با جمعیت‌های مساوی تقسیم می‌شوند، به عنوان نیروهای حمله (Of) و تدافعی (Def).

- مرحله 3: هر بازیکن حمله‌ای به سمت اجتماع بازیکن دفاعی حرکت می‌کند و یک بازیکن دفاعی خاص را انتخاب می‌کند تا به یک جنگ شروع کند.

- مرحله 4: تابع هدف برای هر بازیکن حمله‌ای و دفاعی بررسی می‌شود و به عنوان وضعیت برنده (WS) نشان داده می‌شود.

- مرحله 5: بازیکنان حمله‌ای می‌توانند به گروه حمله موفق (SOG) بپیوندند اگر وضعیت برنده آن‌ها بالاتر از بازیکن دفاعی باشد و به سمت بهترین بازیکنان حمله در میدان بازی حرکت می‌کنند.

- مرحله 6: بازیکنان دفاعی می‌توانند به گروه دفاع موفق (SDG) بپیوندند اگر وضعیت برنده آن‌ها بالاتر از بازیکن حمله باشد و به سمت اجتماع بازیکنان حمله و یک بازیکن حمله خاص حرکت می‌کنند تا برای یک جنگ دیگر آماده شوند.

- مرحله 7: برای هر بازیکن حمله‌ای موفق در SOG، روش به‌روزرس

انی موقعیت آن توسط حرکت به سمت جمعیت بازیکنان دفاعی موفق برای عبور از پل انجام می‌شود.

- مرحله 8: شرط پایانی بررسی می‌شود.

### توضیح الگوریتم برنامه

برای نامزدهای راه حل اولیه، ارزیابی کنید ارزش تناسب را به عنوان وضعیت برنده بازیکنان ( )

در حالی که تکرار (تعداد ارزیابی تابع) کمتر از حداکثر تعداد تکرارها (ارزیابی توابع) است،

n بازیکن را به عنوان حمله و دفاع بازیکنان با m بازیکن در هر گروه تقسیم کنید

بازیکن با بهترین وضعیت برنده ( ) در زمین بازی تعیین کنید.

```
for i=1:
    The ith offensive player moves toward the crowd of defensive players
    (Eq. 6)
    Determine the winning state of ith offensive ( ) and defensive players ( )
    if ≤
        The ith offensive player moves toward the crowd of best offensive
        players (Eq. 9)
        The ith offensive player joins the
    else >
        The ith defensive player moves toward the crowd of offensive players
        (Eq. 12)
        The ith defensive player joins the
    end if
end for
for i=1:
    The ith successful offensive player moves toward the crowd of
    successful defensives (Eq. 13)
end if
end while
Return the player with the best winning state ( ).
end Procedure
```

## مدل ریاضی

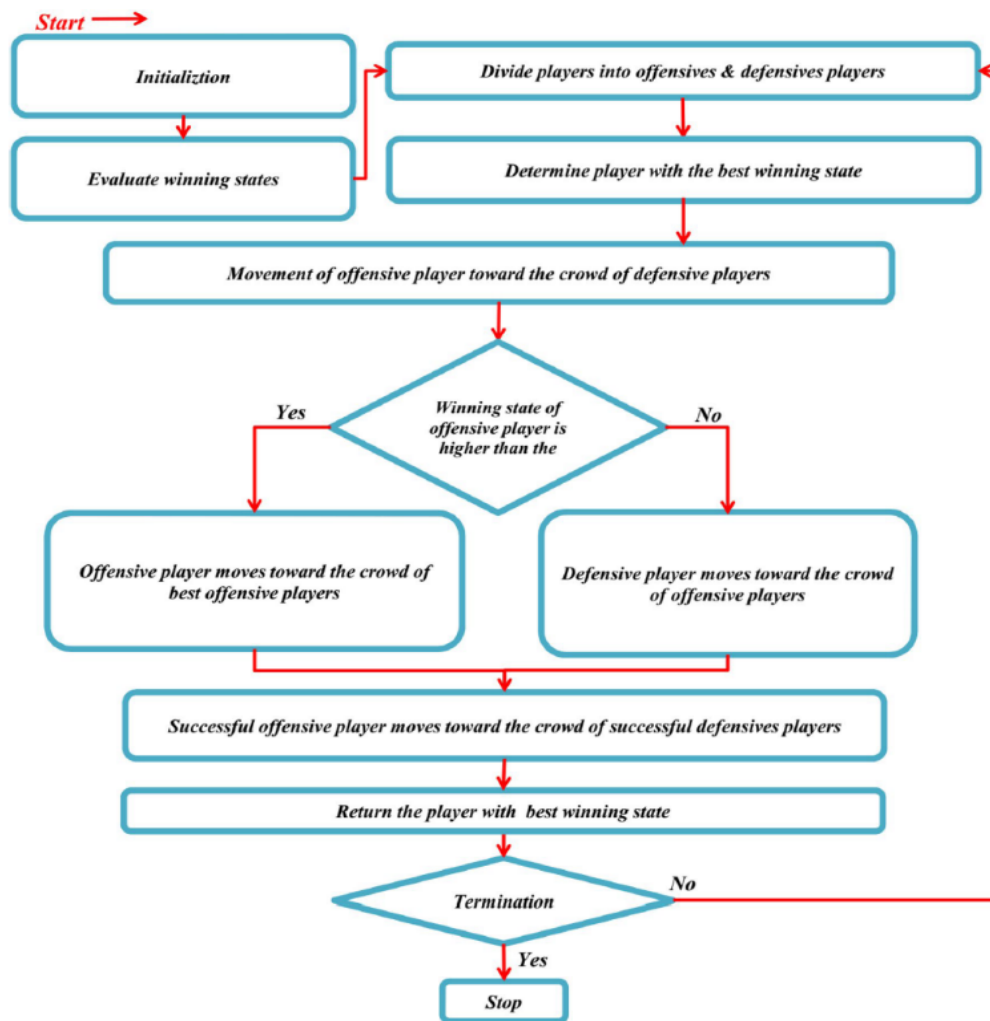
ارائه ریاضی SGO به عنوان یک الگوریتم بهینه سازی در این قسمت با استفاده از استراتژی بازی Squid که در قسمت قبل به تفصیل مورد بحث قرار گرفته است، توضیح داده شده است.

در مرحله اول، فرآیند مقداردهی اولیه به صورت زیر انجام می شود؛ در حالی که فضای جستجو به صورت بخش خاصی از زمین بازی در نظر گرفته می شود و نامزدهای راه حل ( $X_i$ ) بازیکنان فرض می شوند :

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_i \\ \vdots \\ X_n \end{bmatrix} = \begin{bmatrix} x_1^1 x_1^2 \cdots x_1^j \cdots x_1^d \\ x_2^1 x_2^2 \cdots x_2^j \cdots x_2^d \\ \vdots \cdot \vdots \\ x_i^1 x_i^2 \cdots x_i^j \cdots x_i^d \\ \vdots \cdot \vdots \\ x_n^1 x_n^2 \cdots x_n^j \cdots x_n^d \end{bmatrix}, \begin{cases} i = 1, 2, \dots, n. \\ j = 1, 2, \dots, d. \end{cases} \quad (1)$$

$$x_i^j = x_{i,min}^j + rand. (x_{i,max}^j - x_{i,min}^j), \begin{cases} i = 1, 2, \dots, n. \\ j = 1, 2, \dots, d. \end{cases} \quad (2)$$

که در آن  $n$  تعداد کل بازیکنان (کاندیداهای راه حل) را در زمین بازی (فضای جستجو) نشان می دهد؛  $d$  بعد مسئله در نظر گرفته شده است.  $x_j^i$  به متغیر تصمیم زام که برای تعیین موقعیت نامزد اولیه  $i$  ام



استفاده می شود اشاره دارد.  $x_{i,min}^j$  و  $x_{i,max}^j$  کران های بالایی و پایینی متغیر زام در کاندید  $i$  ام هستند. rand به یک عدد تصادفی که به طور یکنواخت در محدوده  $[0, 1]$  توزیع شده است، اشاره دارد.

در مرحله دوم الگوریتم، بازیکنان به دو گروه با اندازه مساوی تقسیم می شوند: مهاجمان (Off) و مدافعان (Def). نمایش ریاضی این مولفه ها در زیر ارائه شده است:

$$X^{Off} = \begin{bmatrix} X_1^{Off} \\ X_2^{Off} \\ \vdots \\ X_i^{Off} \\ \vdots \\ X_m^{Off} \end{bmatrix} = \begin{bmatrix} x_1^1 x_1^2 \dots x_1^j \dots x_1^d \\ x_2^1 x_2^2 \dots x_2^j \dots x_2^d \\ \vdots \cdot \vdots \\ x_i^1 x_i^2 \dots x_i^j \dots x_i^d \\ \vdots \cdot \vdots \\ x_m^1 x_m^2 \dots x_m^j \dots x_m^d \end{bmatrix}, \begin{cases} i = 1, 2, \dots, m. \\ j = 1, 2, \dots, d. \end{cases} \quad (3)$$

$$X^{Def} = \begin{bmatrix} X_1^{Def} \\ X_2^{Def} \\ \vdots \\ X_i^{Def} \\ \vdots \\ X_m^{Def} \end{bmatrix} = \begin{bmatrix} x_1^1 x_1^2 \dots x_1^j \dots x_1^d \\ x_2^1 x_2^2 \dots x_2^j \dots x_2^d \\ \vdots \cdot \vdots \\ x_i^1 x_i^2 \dots x_i^j \dots x_i^d \\ \vdots \cdot \vdots \\ x_m^1 x_m^2 \dots x_m^j \dots x_m^d \end{bmatrix}, \begin{cases} i = 1, 2, \dots, m. \\ j = 1, 2, \dots, d. \end{cases} \quad (4)$$

که در آن  $m$  تعداد کل بازیکنان در هر گروه در بازی است.  $X_i^{Off}$ ،  $i$  امین بازیکن تهاجمی است.  $x_i^{Def}$ ،  $i$  امین بازیکن دفاعی است.

در ادامه، پس از شروع بازی، یکی از بازیکنان تهاجمی به میان بازیکنان دفاعی می رود تا بازی را شروع کند.

شایان ذکر است که هر بازیکن تهاجمی باید با یک پا حرکت کند و بجنگد، درحالیکه دفاع با هر دو پا آزاد است. نمایش ریاضی این اجزا ارائه شده است

$$DG = \frac{\sum_{i=1}^m X_i^{Def}}{m}, i = 1, 2, \dots, m \quad (5)$$

$$X_i^{OffNew1} = \frac{X_i^{Off} + r_1 \times DG - r_2 \times X_{r_3}^{Def}}{2}, i = 1, 2, \dots, m \quad (6)$$

$DG$  گروه دفاعی است که از جمعیت بازیکنان دفاعی تقلید می کند.  $X_i^{OffNew}$  بردار  $i$  ام موقعیت آتی بازیکن

مهاجم ( $X_i^{Off}$ ) در زمین بازی است؛  $r_1$  و  $r_2$  دو عدد تصادفی در محدوده  $[0, 1]$  هستند که نشان

دهنده توانایی بازیکنان مهاجم در رسیدن به هر موقعیتی بین  $DG$  و یک بازیکن دفاعی که به طور

تصادفی انتخاب شده است ( $X_{r_3}^{Def}$ )؛  $r_3$  یک عدد صحیح تصادفی از 1 تا  $m$  می باشد.

در مرحله بعد، پس از مبارزه بین  $i$  امین بازیکن مهاجم ( $X_i^{Off}$ ) و یک بازیکن دفاعی مشخص ( $X_{r_3}^{Def}$ )، ارزیابی تابع هدف برای هر بازیکن انجام می شود و به عنوان برنده ی بازیکنان ( $WS$ ) شناخته می شود. اگر حالت برد بازیکن دفاعی کمتر از حالت برنده بازیکن مهاجم باشد ( $WS_i^{Def} \leq WS_i^{Off}$ )، بازیکن مهاجم به عنوان برنده بازی در نظر گرفته می شود و به گروه تهاجمی موفق ملحق می شود. (نقطه C در شکل 2) با توجه به قوانین اولیه بازی ماهی مرکب، در این زمان بازیکن مهاجم می تواند از هر دو پا استفاده کند. نمایش ریاضی این وجوه را می توان به صورت زیر بیان کرد:

$$X^{SccOff} = \begin{bmatrix} X_1^{SccOff} \\ X_2^{SccOff} \\ \vdots \\ X_i^{SccOff} \\ \vdots \\ X_o^{SccOff} \end{bmatrix} = \begin{bmatrix} x_1^1 x_1^2 \cdots x_1^j \cdots x_1^d \\ x_2^1 x_2^2 \cdots x_2^j \cdots x_2^d \\ \vdots \\ x_i^1 x_i^2 \cdots x_i^j \cdots x_i^d \\ \vdots \\ x_o^1 x_o^2 \cdots x_o^j \cdots x_o^d \end{bmatrix}, \begin{cases} i = 1, 2, \dots, o. \\ j = 1, 2, \dots, d. \end{cases} \quad (7)$$

$$SOG = \frac{\sum_{i=1}^o X_i^{SccOff}}{o}, i = 1, 2, \dots, o \quad (8)$$

$$X_i^{OffNew2} = X_i^{OffNew1} + r_1 \times SOG - r_2 \times BSi = 1, 2, \dots, m \quad (9)$$

که در آن  $o$  تعداد بازیکنان تهاجمی موفق در  $SOG$  است که از جمعیت بازیکنان تهاجمی موفق تقلید می کند؛ ( $X_i^{OffNew2}$ ) بردار موقعیت آتی  $i$  امین بازیکن تهاجمی را نشان می دهد ( $X_i^{OffNew1}$ )؛  $BS$  بهترین نامزد راه حل یا موفق ترین بازیکن تهاجمی را در  $SOG$  نشان می دهد؛  $r_1$  و  $r_2$  دو عدد تصادفی در محدوده  $[0,1]$  هستند.

اگر حالت برد بازیکن دفاعی بالاتر از حالت برنده بازیکن مهاجم باشد ( $WS_i^{Def} > WS_i^{Off}$ )، بازیکن دفاعی به عنوان برنده بازی در نظر گرفته می شود و به گروه دفاعی موفق ( $SDG$ ) می پیوندد. بازیکنان دفاعی این گروه باید از نقطه حساس زمین بازی که پل نامیده می شود محافظت کنند (نقطه F در شکل 2).

در این بین بازیکنان موفق دفاعی به میان جمعیت بازیکنان هجومی می روند تا برای شروع یک مبارزه جدید آماده شوند. ارائه ریاضی این جنبه ها به شرح زیر است:

$$SDG = \begin{bmatrix} X_1^{SccDef} \\ X_2^{SccDef} \\ \vdots \\ X_i^{SccDef} \\ \vdots \\ X_p^{SccDef} \end{bmatrix} = \begin{bmatrix} x_1^1 x_1^2 \cdots x_1^j \cdots x_1^d \\ x_2^1 x_2^2 \cdots x_2^j \cdots x_2^d \\ \vdots \quad \vdots \quad \vdots \\ x_i^1 x_i^2 \cdots x_i^j \cdots x_i^d \\ \vdots \quad \vdots \quad \vdots \\ x_p^1 x_p^2 \cdots x_p^j \cdots x_p^d \end{bmatrix}, \begin{cases} i = 1, 2, \dots, p. \\ j = 1, 2, \dots, d. \end{cases} \quad (10)$$

$$OG = \frac{\sum_{i=1}^m X_i^{Off}}{m}, i = 1, 2, \dots, m) \quad (11)$$

$$X_i^{DefNew1} = X_i^{Def} + r_1 \times OG - r_2 \times X_{r_3}^{Off} \quad i = 1, 2, \dots, m \quad (12)$$

OG گروه تهاجمی است که از جمعیت بازیکنان تهاجمی تقلید می کند.  $X_i^{OffNew1}$  بردار موقعیت بازیکن دفاعی i ام ( $X_i^{Def}$ ) در زمین بازی (فضای جستجو)؛  $r_1$  و  $r_2$  دو عدد تصادفی در محدوده  $[0, 1]$  هستند که نشان دهنده توانایی بازیکنان دفاعی در رسیدن به هر موقعیتی بین OG و یک بازیکن تهاجمی به طور تصادفی انتخاب شده ( $X_{r_3}^{Off}$ ) برای شرکت در یک مبارزه جدید است؛  $r_3$  یک عدد صحیح تصادفی است از 1 تا m .

برای تنظیم هوشمند مرحله اکتشاف و بهره برداری از الگوریتم پیشنهادی، حلقه جستجوی دیگری در الگوریتمی پیاده سازی می شود که در آن بازیکنان مهاجم در SOG سعی می کنند از پلی که توسط بازیکنان دفاعی در SDG محافظت می شود عبور کنند. برای این منظور، یک روش به روز رسانی موقعیت برای همه بازیکنان تهاجمی در SOG انجام می شود، با حرکت به سمت یک بازیکن دفاعی خاص در SDG (که شبیه فرآیند پاس کردن پل است) و بهترین راه حلی که تاکنون پیدا شده است (که پاداش بازیکن مهاجم برای عبور از پل را تقلید می کند). ارائه ریاضی این جنبه ها به شرح زیر است:

$$X_i^{OffNew3} = X_i^{SccOff} + r_1 \times BS - r_2 \times X_k^{SccDef} \begin{cases} i = 1, 2, \dots, o. \\ k = 1, 2, \dots, p. \end{cases} \quad (13)$$

که در آن o و p تعداد بازیکنان موفق تهاجمی و دفاعی به ترتیب در SOG و SDG است؛  $X_i^{OffNew3}$  بردار موقعیت آینده i امین بازیکن تهاجمی موفق را نشان می دهد ( $X_i^{OffNew1}$ ) که از پل می گذرد؛ BS



بهترین گزینه یا موفق ترین بازیکن تهاجمی را در SOG نشان می دهد؛  $r_1$  و  $r_2$  دو تصادفی در محدوده  $[0, 1]$  هستند.