

## شبیه‌سازی نبرد با استفاده از شبکه‌های عصبی زمان پیوسته

محمد مقدس<sup>۱\*</sup>

حمید بیگدلی<sup>۲</sup>

### چکیده

این مقاله روی مدل‌سازی رفتار فرماندهان در شبیه‌سازی نبرد تمرکز می‌کند. یک مأموریت نظامی اغلب با اهداف متضاد چندگانه از جمله موفقیت در وظایف، زمان اتمام، نابودی دشمنان و بقای نیروهای خودی همراه است. در این مقاله با در نظر گرفتن سناریوهای غیردفاعی و دفاعی و با استفاده از بهینه‌سازی چند هدفی، مدلی به منظور کمینه‌سازی تلفات نیروهای خودی و بیشینه‌سازی نابودی دشمنان معرفی می‌شود. همچنین با استفاده از روش وزن‌دهی و شرایط بهینگی کاروش-کان-تاکر، یک مدل شبکه عصبی بازگشتی زمان پیوسته برای حل مسأله‌ی بهینه‌سازی چند هدفی معرفی شده، طراحی می‌شود. ایده‌ی اصلی رهیافت شبکه عصبی برای مسأله‌ی بهینه‌سازی چند هدفی معرفی شده، ایجاد یک دستگاه دینامیکی به صورت معادلات دیفرانسیل معمولی مرتبه اول است. شبکه عصبی معرفی شده نیاز به هیچ پارامتر تنظیم‌کننده‌ای ندارد و ساختار آن یک پیاده‌سازی سخت افزاری ساده را امکان‌پذیر می‌سازد. روش معرفی شده می‌تواند به‌عنوان یک مشاور برای فرماندهی که برای نیروهای تحت امر خود تصمیم می‌گیرد، عمل کند. در پایان، با استفاده از یک مثال اعتبار و کارایی مدل پیشنهادی نشان داده می‌شود.

### واژه‌های کلیدی:

شبیه‌سازی نبرد، شبکه عصبی، بهینه‌سازی چندهدفی.

۱. دانشجوی دکتری گروه ریاضی، دانشکده علوم پایه، دانشگاه آزاد اسلامی واحد تهران مرکزی

۲. پژوهشگر پژوهشکده عالی جنگ دانشگاه فرماندهی و ستاد آجا

## مقدمه

شبیه‌سازی یکی از فنون آینده‌پژوهی است و فرماندهان نظامی می‌توانند در تحلیل و بررسی طرح‌ها و نتایج آن‌ها از آن استفاده کنند. زمینه‌های کاربردی فراوانی مانند حوزه‌های نظامی، صنایع خدماتی، سازمان‌های تولیدی و شرکت‌های حمل‌ونقل در سراسر جهان وجود دارند که از شبیه‌سازی‌های کامپیوتری به‌طور گسترده استفاده می‌کنند (لو و کلتون، ۱۹۹۱). برای مثال در حوزه‌های نظامی، شبیه‌سازی کامپیوتری برای آموزش سربازان در تانک‌ها و هواپیماها، ارزیابی و آزمایش برنامه‌های عملیاتی و ساختارهای نیروی نظامی و نیز ارزیابی و مقایسه سامانه‌های دفاعی و سلاح‌های جدید مورد استفاده قرار می‌گیرد (أسوات، ۱۹۹۳). شبیه‌سازی کامپیوتری دارای مزایای متعددی نسبت به سایر روش‌های مرسوم ارزیابی سامانه‌ها مانند روش‌های پایلوت است. دلیل اصلی استفاده از شبیه‌سازی این است که آن‌ها نتایج واقعی را با هزینه‌ی پایین‌تری نسبت به روش‌های دیگر ارائه می‌دهند. گاهی اوقات به دلیل اندازه و پیچیدگی سامانه‌ها در دنیای واقعی، تنها ابزار کارا برای ارزیابی آن‌ها شبیه‌سازی است. همچنین سامانه‌هایی وجود دارند که استفاده مستقیم از آن‌ها خطرناک و پرهزینه است، بنابراین استفاده از شبیه‌سازی ضروری به نظر می‌رسد. سایر مزایای استفاده از شبیه‌سازی توانایی کنترل کامل بر شرایط آزمایشی و بررسی عملکرد سامانه در یک بازه زمانی طولانی مدت (لو و کلتون، ۱۹۹۱) هستند.

در سال‌های اخیر با استفاده از روش‌های گوناگون به شبیه‌سازی نبرد پرداخته شده است. کیلمر (۱۹۹۶) نشان داد که می‌توان شبکه‌های عصبی مصنوعی را به‌عنوان ابزاری مؤثر در توسعه‌ی متامدل‌ها برای شبیه‌سازی نبرد به کار گرفت. ژو و همکاران (۲۰۱۰) یک الگوریتم LMBP<sup>۱</sup> تعمیم‌یافته برای شبیه‌سازی نبرد معرفی کردند. تنگ و همکاران (۲۰۱۲) یک شبکه عصبی خود سازمانده به نام TD-FALCON را برای شبیه‌سازی مانورهای هوایی به کار گرفتند. بوتسلیس و رینگرز (۲۰۱۳) شبکه‌های عصبی و مدل‌های جمعی تعمیم‌یافته<sup>۲</sup> GAMLSS را برای مدل‌سازی شبیه‌سازی نبرد با یکدیگر مقایسه کردند.

از آنجا که زمان مورد نیاز برای حل مسائل بهینه‌سازی و به‌خصوص مسائل بهینه‌سازی غیرخطی بسیار وابسته به بعد و ساختار مسأله است بنابراین الگوریتم‌های عددی کارایی کمی از خود نشان می‌دهند. یک رهیافت امیدوارکننده برای حل مسائل بهینه‌سازی با ابعاد بالا استفاده از شبکه‌های عصبی مصنوعی می‌باشد. قابلیت محاسبات توزیع‌شده موازی سنگین و همگرایی سریع از جمله

<sup>۱</sup>. Levenberg-Marquardt algorithm

<sup>۲</sup>. Generalised Additive for Location, Scale and Shape (GAMLSS)

ویژگی‌های شبکه‌های عصبی مصنوعی هستند که آن‌ها را برای حل مسائل بهینه‌سازی زمان حقیقی با ابعاد بالا کارا می‌سازد. از دیدگاه ریاضیات در طراحی شبکه‌های عصبی برای حل مسائل بهینه‌سازی ساختن یک تابع انرژی (تابع لیاپانوف) و یک دستگاه دینامیکی که نمایشگر مدل آن شبکه است، اهمیت ویژه‌ای دارد. این دستگاه دینامیکی به‌طور معمول متشکل از معادلات دیفرانسیل معمولی مرتبه اول است.

تانک و هاپفیلد (۱۹۸۶) اولین مدل شبکه عصبی را برای حل مسائل بهینه‌سازی معرفی کردند و نشان دادند که وضعیت شبکه در هر تکرار به‌گونه‌ای تغییر می‌کند که تابع انرژی متناظر با آن به‌طور یکنواخت کاهش می‌یابد تا جایی که به نقطه مینیمم خود می‌رسد و این نقطه مینیمم متناظر با نقطه تعادل شبکه عصبی می‌باشد. کندی و چا (۱۹۸۸) با افزودن یک پارامتر جریمه متناهی به شبکه هاپفیلد آن را توسعه دادند و از آن برای حل مسأله برنامه‌ریزی غیرخطی محدب استفاده نمودند. برای جلوگیری از به‌کار بردن پارامتر جریمه یک شبکه عصبی سوئیچ-خازن توسط رودریگز-وازکز (۱۹۹۰) و همکاران معرفی شد. همچنین محققان زیادی براساس روش لاگرانژ، روش تابع تصویر و غیره، برای حل مسائل برنامه‌ریزی خطی و درجه دوم (بوزردوم و پاتیسن، ۱۹۹۳؛ ژو و بیان، ۲۰۰۷؛ گائو و لیائو، ۲۰۱۰؛ عفتی و همکاران، ۲۰۱۱) و مسائل برنامه‌ریزی غیرخطی (ما و شانبلات، ۱۹۹۲؛ ژانگ و کنستانتینیدس، ۱۹۹۲؛ ژیا و همکاران، ۲۰۰۲؛ ژیا و وانگ، ۲۰۰۴، ۲۰۰۵؛ ناظمی و عفتی، ۲۰۱۳؛ عفتی و مقدس، ۲۰۱۶) چندین مدل شبکه عصبی را معرفی کرده‌اند.

در این مقاله نوعی نبرد انتزاعی بین نیروهای خودی و دشمن، و همچنین دو سناریوی غیردفاعی و دفاعی در نظر گرفته می‌شود. سپس با استفاده از بهینه‌سازی چند هدفی، مدلی به‌منظور کمینه‌سازی تلفات نیروهای خودی و بیشینه‌سازی نابودی دشمنان معرفی می‌شود. همچنین با استفاده از روش اسکالر سازی وزن‌دهی<sup>۱</sup>، شرایط بهینگی کاروش-کان-تاکر (KKT<sup>۲</sup>) و نظریه سیستم‌های دینامیکی یک مدل شبکه عصبی بازگشتی زمان پیوسته برای حل مسأله بهینه‌سازی چند هدفی معرفی شده، طراحی می‌شود. شبکه عصبی معرفی شده پایدار به مفهوم لیاپانوف و همگرایی سراسری به جواب بهینه مسأله بهینه‌سازی چند هدفی معرفی شده، است. در پایان نتایج شبیه‌سازی، کارایی و عملکرد شبکه عصبی معرفی شده را نشان می‌دهند.

<sup>۱</sup>. Weighting method

<sup>۲</sup>. Karush-Kuhn-Tucker (KKT)

## بررسی مبانی نظری و پیشینه‌ها

### بهینه‌سازی چندهدفی

مسئله بهینه‌سازی چندهدفی را به صورت زیر در نظر می‌گیریم.

یافتن بردار  $x \in (x_1, x_2, \dots, x_n) \in \Omega$  به طوری که

$$\begin{aligned} \max \quad & F(x) = (f_1(x), f_2(x), \dots, f_k(x))^T \\ \text{subject to} \quad & x \in \Omega (\subseteq \mathbb{R}^n), \end{aligned} \quad (1)$$

که در آن  $F(x)$  بردار توابع هدف و  $f_i: \mathbb{R}^n \rightarrow \mathbb{R}$ ،  $i=1, \dots, k$  و  $\Omega = \{x \in \mathbb{R}^n \mid g(x) \leq 0, h(x) = 0\}$  فضای تصمیم است به طوری که  $g(x) = (g_1(x), g_2(x), \dots, g_m(x))^T$  یک تابع برداری پیوسته  $m$ -بعدی است. همچنین  $h(x) = Ax - b$ ،  $A \in \mathbb{R}^{l \times n}$ ،  $h(x) = Ax - b$  و  $0 \leq l < n$ ،  $\text{rank}(A) = l$ ،  $b \in \mathbb{R}^l$ . در طول این مقاله فرض می‌شود توابع  $f_i$  و  $g_j$  محدب و دوبار مشتق پذیر هستند.

تعریف (ساکاوا، ۱۹۹۳: ۱۶).  $x^* \in \Omega$  جواب بهینه پارتوی مسئله بهینه‌سازی چند هدفی (۱) نامیده می‌شود هرگاه نقطه‌ای مانند  $x \in \Omega$  موجود نباشد به طوری که برای هر  $i=1, \dots, k$ ،  $f_i(x) > f_i(x^*)$  و برای حداقل یک  $i$ ،  $f_i(x) \geq f_i(x^*)$ .

از جمله روش‌های اسکالر سازی متداول برای مسائل بهینه‌سازی چندهدفی، روش وزن دهی است. در روش وزن دهی برای یافتن جواب بهینه پارتو باید مسئله معادل با (۱) را که تابع هدف آن مجموع وزن دار شده توابع هدف (۱) است، حل کنیم. بنابراین مسئله وزن دهی شده به صورت زیر تعریف می‌شود:

$$\begin{aligned} \max \quad & F_w(x) = \sum_{i=1}^k w_i f_i(x) \\ \text{subject to} \quad & x \in \Omega, \end{aligned} \quad (2)$$

به طوری که  $w = (w_1, w_2, \dots, w_k)$  بردار وزن تخصیص داده شده به توابع هدف است و فرض می‌شود که  $w \geq 0$ . اثبات می‌شود اگر  $x^* \in \Omega$  به ازای یک بردار  $w > 0$  جواب بهینه مسئله (۲) باشد آنگاه  $x^*$  جواب بهینه پارتو مسئله چند هدفی (۱) است (ساکاوا، ۱۹۹۳: ۱۶).

دیگر روش‌های اسکالرسازی متداول برای مسائل بهینه‌سازی چندهدفی روش معیار سراسری<sup>۱</sup>، برنامه‌ریزی آرمانی<sup>۲</sup>، برنامه‌ریزی فازی<sup>۳</sup> و روش‌های تعاملی<sup>۴</sup> هستند. تمام این روش‌ها نیازمند نقاط مرجع ابتدایی برای به دست آوردن جواب یا جواب‌های سازگار هستند. به‌طور هندسی جواب سازگار، جوابی است که کمترین فاصله از نقطه ایده‌آل را دارد (یو، ۱۹۷۳). اکنون ما علاقه‌مند به یافتن جواب سازگار مسأله (۱) هستیم.

فرض کنید  $F^+ = (f_1^+, f_2^+, \dots, f_k^+)$  نقطه ایده‌آل یا ایده‌آل مثبت برای بردار توابع هدف باشد به طوری که  $f_i^+ = \max_{x \in \Omega} f_i(x)$  و  $i=1, \dots, k$  و  $F^- = (f_1^-, f_2^-, \dots, f_k^-)$  نقطه ضدایده‌آل یا ایده‌آل منفی برای بردار توابع هدف باشد به طوری که  $f_i^- = \min_{x \in \Omega} f_i(x)$ ،  $i=1, \dots, k$ . حال از متر  $L_p$  یا نرم  $p$  برای به دست آوردن فاصله استفاده می‌کنیم. متر  $L_p$  فاصله بین  $F^+$  و  $F(x)$  را به صورت زیر تعریف می‌کند:

$$d_p = \left( \sum_{i=1}^k w_i (f_i^+ - f_i(x))^p \right)^{\frac{1}{p}}, \quad p=1, 2, \dots, \infty \quad (3)$$

که  $w_i$  اهمیت نسبی  $i$  امین تابع هدف را نشان می‌دهد و فرض می‌شود که  $w_i > 0$ . با مفروض بودن بردار وزن  $w = (w_1, w_2, \dots, w_k)$ ، جواب  $x_w^p$  حاصل از کمینه کردن (۳) را جواب سازگار مسأله (۱) می‌نامیم. باید توجه شود که مجموعه جواب‌های  $x_w^p$  به ازای  $1 \leq p \leq \infty$  در صورتی که  $w > 0$  جواب‌های بهینه پارتو را تشکیل می‌دهند (ساکاوا، ۱۹۹۳: ۱۶).

اگر توابع هدف  $f_i(x)$ ،  $i=1, \dots, k$  هم مقیاس نباشند از یک تابع مقیاس‌گذاری برای توابع هدف استفاده می‌شود. در این حالت متر زیر مورد استفاده قرار می‌گیرد:

$$d_p = \left( \sum_{i=1}^k w_i \left( \frac{f_i^+ - f_i(x)}{f_i^+ - f_i^-} \right)^p \right)^{\frac{1}{p}}, \quad p=1, 2, \dots, \infty \quad (4)$$

## روش‌شناسی پژوهش

### مدل شبکه عصبی

در ابتدای این بخش به مرور بعضی از مفاهیم و تعاریف اولیه می‌پردازیم.

**تعریف.** سیستم دینامیکی زیر را در نظر بگیرید:

1. Global criterion method
2. Goal programming
3. Fuzzy programming
4. Interactive approaches

$$\frac{dx}{dt} = H(x(t)), x(t_0) = x_0 \in \mathbb{R}^n, \quad (5)$$

که در آن  $H$  یک تابع از  $\mathbb{R}^n$  به  $\mathbb{R}^n$  می‌باشد.  $x^*$  یک نقطه تعادل (۵) نامیده می‌شود اگر  $H(x^*) = 0$ . اگر همسایگی  $\Omega^* \subseteq \mathbb{R}^n$  از  $x^*$  وجود داشته باشد که  $H(x^*) = 0$  و به ازای هر  $x \in \Omega^* \setminus \{x^*\}$ ،  $H(x) \neq 0$ ، آنگاه  $x^*$  یک نقطه تعادل تنها نامیده می‌شود.

**تعریف.** فرض کنید که  $x(t)$  یک جواب برای (۵) باشد، نقطه تعادل تنهای  $x^*$ ، پایدار به مفهوم لیاپانوف است اگر برای هر  $x_0 = x(t_0)$  و هر  $\varepsilon > 0$ ، یک  $\delta > 0$  وجود داشته باشد به طوری که اگر  $\|x(t_0) - x^*\| < \delta$  آنگاه  $\|x(t) - x^*\| < \varepsilon$  برای هر  $t \geq t_0$ .

**تعریف.** فرض کنید  $\Omega \subseteq \mathbb{R}^n$  یک همسایگی باز از  $\bar{x}$  باشد. تابع به طور پیوسته مشتق پذیر

$V: \mathbb{R}^n \rightarrow \mathbb{R}$  یک تابع لیاپانوف در  $\bar{x}$  روی مجموعه  $\Omega$  برای (۵) است اگر

$$(1) \quad V(\bar{x}) = 0 \text{ و به ازای هر } x \in \Omega \setminus \{\bar{x}\}, V(x) > 0$$

$$(2) \quad \forall x \in \Omega \text{ به ازای هر } \dot{V}(x) = [\nabla_x V(x)]^T f(x(t)) \leq 0$$

**لم (میلر و میشل، ۱۹۸۲: ۱۱).** نقطه تعادل تنهای  $x^*$  از سیستم (۵) پایدار لیاپانوف است اگر یک تابع لیاپانوف روی همسایگی  $\Omega^*$  از  $x^*$  وجود داشته باشد.

**تعریف.** فرض کنید  $X^*$  مجموعه نقاط تعادل (۵) باشد. سیستم دینامیکی (۵) همگرای سراسری به  $X^*$  گفته می‌شود اگر برای هر نقطه آغازین دلخواه، مسیر  $x(t)$  در رابطه

$$\text{dist}(x, \Omega^*) = \inf_{y \in \Omega^*} \|x - y\| \text{ صدق کند که در آن } \lim_{t \rightarrow \infty} \text{dist}(x(t), \Omega^*) = 0$$

ماهیت شبکه‌های عصبی زمان پیوسته که برای حل مسائل بهینه‌سازی به کار می‌روند یک دستگاه دینامیکی است. این دستگاه دینامیکی به طور مرسوم از معادلات دیفرانسیل معمولی مرتبه اول تشکیل می‌شود. انتظار می‌رود به ازای یک نقطه آغازین منحنی‌های این دستگاه دینامیکی به سمت نقطه تعادل آن که متناظر با جواب بهینه‌ی مسأله موردنظر است میل کنند. در این بخش به معرفی یک شبکه عصبی برای حل مسأله (۲) و در نتیجه حل مسأله (۱) می‌پردازیم. برای این منظور ابتدا تابع لاگرانژ متناظر با (۲) را به صورت زیر در نظر می‌گیریم:

$$L(x, u, v) = -F_w(x) + \frac{1}{2} \sum_{j=1}^m u_j^2 g_j(x) + \sum_{s=1}^l v_s h_s(x)$$

که  $u \in \mathbb{R}^m$  و  $v \in \mathbb{R}^l$  ضرایب لاگرانژ هستند. از کتاب برنامه‌ریزی غیرخطی (بازار، ۲۰۰۵: ۱) می‌دانیم که  $x^* \in \mathbb{R}^n$  یک جواب بهینه پارتو برای (۱) است اگر و تنها اگر  $u^* \in \mathbb{R}^m$  و  $v^* \in \mathbb{R}^l$  وجود داشته باشند به طوری که  $(x^{*T}, u^{*T}, v^{*T})^T$  در شرایط KKT زیر صدق کند:

$$\begin{aligned} u^* \geq 0, \quad g(x^*) \leq 0, \quad u^{*T} g(x^*) &= 0, \\ \nabla F_w(x^*) + \nabla g(x^*)^T u^* + \nabla h(x^*)^T v^* &= 0, \\ h(x^*) &= 0 \end{aligned} \quad (۶)$$

بردار ضرایب لاگرانژ متناظر با  $x^*$  نامیده می‌شود. اکنون یک شبکه عصبی برای حل (۲) و در نتیجه حل (۱) با سیستم دینامیکی زیر معرفی می‌کنیم:

$$\begin{aligned} \frac{dx}{dt} &= -\nabla_x L(x, u, v), \\ \frac{du}{dt} &= \nabla_u L(x, u, v) = \text{diag}(u_1, \dots, u_m) g(x), \\ \frac{dv}{dt} &= \nabla_v L(x, u, v) = h_s(x), \quad s = 1, \dots, l, \end{aligned} \quad (۷)$$

با نقطه آغازین  $(x_0^T, u_0^T, v_0^T)^T$  و  $u_j(t_0) \neq 0$  ( $j=1, \dots, m$ ). توجه کنید در شبکه عصبی (۷) ضرایب متناظر با محدودیت‌های نامساوی به صورت  $u_j^2$  ( $j=1, \dots, m$ ) تعریف شده‌اند. مزیت اصلاح ضرایب متناظر با محدودیت‌های نامساوی این است که محدودیت نامنفی ( $u \geq 0$ ) می‌تواند حذف شود و شرایط KKT (۶) برقرار شوند. حال با در نظر گرفتن  $z = (x^T, u^T, v^T)^T \in \mathbb{R}^{n+m+l}$

$$\Phi(x, u, v) = \begin{bmatrix} -\left( \nabla F_w(x) + \frac{1}{2} \nabla g(x)^T u^2 + \nabla h(x)^T v \right) \\ \text{diag}(u_1, \dots, u_m) g(x) \\ h(x) \end{bmatrix},$$

شبکه عصبی (۷) را می‌توان به صورت زیر نوشت:

$$\begin{cases} \frac{dz}{dt} = \Phi(z), \\ z(t_0) = z_0, \quad u(t_0) \neq 0. \end{cases} \quad (۸)$$

پایداری و همگرایی شبکه عصبی (۸) را با استفاده از قضایای زیر می‌توان مورد بررسی قرار داد.

قضیه (ناظمی، ۲۰۱۲: ۱۲). فرض کنید  $z^* = (x^{*T}, u^{*T}, v^{*T})^T$  نقطه تعادل شبکه عصبی (۸) باشد. آنگاه  $x^*$  در شرایط KKT (۶) صدق می‌کند.

قضیه (ناظمی، ۲۰۱۲: ۱۲). اگر  $\nabla^2 F_w(x)$  معین مثبت و  $\nabla^2 g_j(x)$  ( $j=1, \dots, m$ ) نیمه معین مثبت باشد یا اگر  $\nabla^2 F_w(x)$  نیمه معین مثبت و  $\nabla^2 g_j(x)$  ( $j=1, \dots, m$ ) معین مثبت باشد، آنگاه شبکه عصبی (۸) پایدار سراسری به مفهوم لیاپانوف و همگرای سراسری به نقطه KKT،  $z^* = (x^{*T}, u^{*T}, v^{*T})^T$  می‌باشد به طوری که  $x^*$  جواب بهینه مسأله (۲) است.

## یافته‌های پژوهش

### طراحی آزمایش

در این بخش کاربردی از شبکه‌ی عصبی معرفی شده در شبیه‌سازی نبرد را می‌بینیم. SIMBAT یک شبیه‌سازی تصادفی از جنگ‌های متعارف است که مفاهیمی مانند تحرک، خرابی، خستگی، استفاده از سوخت و مهمات، آتش مستقیم و غیرمستقیم و تصمیمات فرماندهی را مدل می‌کند. کاربر می‌تواند یک سناریو (رزمندگان، زمین عملیات، اهداف و ...) بر اساس مواردی مثل وسایل نقلیه زرهی که در یگان‌ها طبقه‌بندی می‌شوند و ممکن است ترکیبی از اجزای مختلف (تانک‌ها، ضدتانک، توپ، پیاده‌نظام و غیره) باشند، تعریف کند. سناریوی استفاده‌شده در این مقاله درگیری بین نیروهای آبی متشکل از دودسته تانک و دودسته پیاده‌نظام زرهی مجهز به وسایل نقلیه جنگی پیاده‌نظام (IFV<sup>۱</sup>) و نیروهای قرمز متشکل از دودسته تانک و یک دسته پیاده‌نظام زرهی مجهز به IFV، می‌باشد.

در این مقاله ما نیروهای آبی را به‌عنوان نیروهای خودی و نیروهای قرمز را به‌عنوان نیروهای دشمن در نظر می‌گیریم. نوعی نبرد انتزاعی بین نیروهای خودی و دشمن را در نظر می‌گیریم. در هر دور واحدها به‌طور هم‌زمان مبارزه می‌کنند. نبرد بین دو گروه از واحدها آغاز می‌شود و بعد از یک دنباله از مبارزات یک گروه حذف می‌شود. در نبردها فرض می‌کنیم واحدها دارای ویژگی‌های زیر هستند:

h: قدرت دفاعی واحد<sup>۲</sup>

a: ارزش حمله<sup>۳</sup> (مقدار خسارتی که یک واحد می‌تواند وارد کند که ثابت فرض می‌شود).

1. Infantry Fighting Vehicles

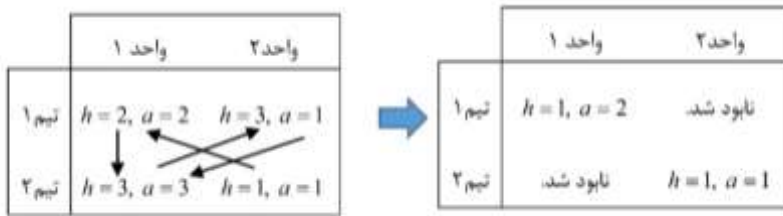
2. Hitpoints

3. Attack value



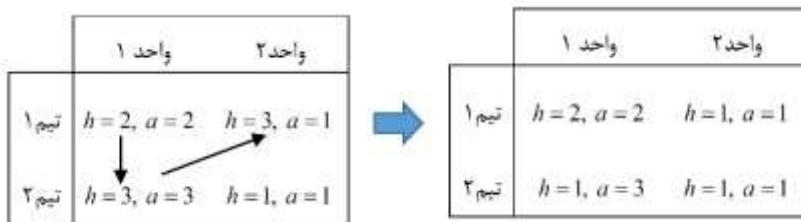
c: زمان انتظار تا حمله بعدی<sup>۱</sup> (که مقداری ثابت فرض می‌شود).

هر واحد دارای یک سلاح با ارزش حمله  $a$  و زمان انتظار تا حمله بعدی  $c$  مشخص است. اگر سلاح خنک شده باشد، یعنی از آخرین شلیک سلاح  $c+1$  دور گذشته باشد می‌تواند دوباره آتش کند. قبل از حمله، واحدها باید هدف خود را انتخاب کنند و در گام بعد آن‌ها می‌توانند تنها به آن هدف شلیک کنند. اگر یک واحد می‌خواهد هدف دیگری را مورد حمله قرار دهد، باید آن را نشانه بگیرد و این کار یک گام زمانی هزینه در بردارد. دو سناریوی غیر دفاعی و دفاعی را در نظر می‌گیریم. در سناریوی غیردفاعی هنگامی که به یک واحد حمله می‌شود قدرت دفاعی او به اندازه مقدار و ارزش حمله مهاجم کاهش می‌یابد. اگر قدرت دفاعی واحدی زیر ۱ بیاید، آن واحد نابود و از نبرد حذف می‌شود. نمونه‌ای از تغییر قدرت دفاعی واحدها را در یک سناریوی غیردفاعی در شکل (۱) می‌بینید.



شکل (۱) سناریوی غیردفاعی و تغییر قدرت دفاعی واحدها در نبرد (فلش‌ها حملات را نشان می‌دهند) همچنین می‌توان سناریوی دفاعی را پیش گرفت. اقدام دفاعی، واحد را قادر می‌سازد تا از ارزش حمله  $a$  خود برای دفاع استفاده کند. به‌طور مشخص، اگر یک واحد به‌جای حمله تصمیم به دفاع از خود بگیرد، یک نسبت معین (احتمالاً بزرگ‌تر از یک) از ارزش حمله‌اش را به‌طور موقت به قدرت دفاعی ( $h$ ) خود اضافه می‌کند. هنگامی که به این واحد حمله می‌شود، تنها زمانی مهاجم باعث آسیب به واحد خواهد شد که ارزش حمله‌اش بزرگ‌تر از نسبت ارزش حمله اضافه‌شده برای دفاع واحد باشد. اگر به یک واحد حمله نشود، قدرت دفاعی بدون تغییر باقی می‌ماند و افزایش نمی‌یابد. علاوه بر این، یک واحد دفاعی نیز می‌تواند باعث آسیب به مهاجم شود. موضع دفاعی یک واحد در کنار اینکه مزایای مشخصی دارد ریسک نیز محسوب می‌شود زیرا واحدی که اقدام به دفاع می‌کند ممکن است مورد حمله قرار نگیرد و در نتیجه یک دور را هدر می‌دهد. سناریوی نبرد دفاعی در شکل (۲) نشان داده شده است.

<sup>۱</sup>. Cool-down period



شکل (۲) سناریوی دفاعی و تغییر قدرت دفاعی واحدها در نبرد

در سناریوهای شرح داده شده می‌توان قدرت دفاعی ( $h_i$ ) یک واحد را به‌عنوان برآوردگر امید به زندگی آن واحد استفاده کرد. درحالی‌که ارزش حمله تقسیم‌بر زمان انتظار تا حمله بعدی به‌اضافه یک ( $a_i/(c_i+1)$ ) را می‌توان به‌عنوان میانگین خسارتی که یک واحد وارد می‌کند در نظر گرفت. بنابراین می‌توان مجموع طول عمر و آسیب‌زندگی گروهی متشکل از  $n$  واحد را

به‌صورت  $\sum_{i=1}^n \frac{h_i \times a_i}{c_i + 1}$  در نظر گرفت. برای هر فرمانده بهتر است که نیروهایی با توزیع قدرت دفاعی یکنواخت داشته باشد تا این که بخشی از نیروهایش قدرت دفاعی پایین و بخشی دیگر قدرت دفاعی بالا داشته باشند. علت این موضوع این است که واحدهای با قدرت دفاعی پایین به نابودی نزدیک‌تر هستند. بنابراین برای آن که توزیع یکنواخت‌تر قدرت دفاعی را برقرار کنیم ریشه دوم قدرت دفاعی هر واحد یعنی  $\sqrt{h_i}$  را در نظر می‌گیریم. می‌توان دو تابع هدف برای این مسأله در نظر گرفت. بیشینه‌سازی قدرت دفاعی نیروهای خودی و نابودی نیروهای دشمن و کمینه‌سازی قدرت دفاعی نیروهای دشمن و تلفات نیروهای خودی. بنابراین می‌توان توابع هدف را به‌صورت زیر در نظر گرفت:

$$\begin{cases} \max f_1 = \sum_{i=1}^{n_1} \frac{\sqrt{h_i^f} \times a_i^f}{c_i^f + 1} \\ \min f_2 = \sum_{i=1}^{n_2} \frac{\sqrt{h_i^e} \times a_i^e}{c_i^e + 1} \end{cases} \quad (9)$$

که  $n_1$  و  $n_2$  به ترتیب تعداد نیروهای خودی و نیروهای دشمن هستند. در نتیجه مطابق با مسأله وزن دهی شده (۲) می‌توان تابع هدف معادل با توابع هدف (۹) را به‌صورت زیر در نظر گرفت:

$$\max w_1 \left( \sum_{i=1}^{n_1} \frac{\sqrt{h_i^f} \times a_i^f}{c_i^f + 1} \right) - w_2 \left( \sum_{i=1}^{n_2} \frac{\sqrt{h_i^e} \times a_i^e}{c_i^e + 1} \right) \quad (10)$$

بطوریکه  $w_1, w_2 > 0$ . حال شبکه عصبی معرفی شده (۸) را برای حل این مسأله به کار می‌بریم. شبیه‌سازی در نرم‌افزار مطلب 2014a و با حل‌کننده ode45 انجام می‌شود. جدول ۱ خروجی شبکه عصبی (۸) را برای مسأله (۱۰) و با سناریوی تشریح شده با وزن‌های مختلف نشان می‌دهد:

جدول (۱) نتایج شبیه‌سازی با استفاده از مدل (۱۰) و شبکه عصبی (۸) و وزن‌های مختلف

نوع سناریو	$(w_1, w_2)$	درصد پیروز شدن نیروهای خودی	مقادیر توابع هدف $(f_1, f_2)$
دفاعی	(5,10)	90%	(50.36, 5.02)
دفاعی	(5,5)	73%	(40.50, 11.00)
غیردفاعی	(10,5)	82%	(43.00, 7.75)

نتایج عددی در جدول (۱) نشان می‌دهند که استفاده از مدل (۱۰) و شبکه عصبی معرفی شده (۸) در افزایش درصد پیروزی نیروهای خودی و نابودی نیروهای دشمن مؤثر بوده‌اند.

### نتیجه‌گیری و پیشنهادها

در این مقاله یک شبکه عصبی زمان پیوسته برای حل مسائل بهینه‌سازی چند هدفی و شبیه‌سازی نبرد معرفی شد. شبکه عصبی معرفی شده با استفاده از شرایط بهینگی KKT و نظریه دستگاه‌های دینامیکی به دست آمد. شبکه عصبی ارائه شده پایدار به مفهوم لیاپانوف و همگرای سراسری به جواب بهینه مسأله بهینه‌سازی چند هدفی معرفی شده است. از مزایای مدل شبکه عصبی معرفی شده می‌توان به ساختار ساده، توانایی حل مسائل با ابعاد بالا و پیاده‌سازی سخت‌افزاری قدرتمند آن اشاره کرد. همچنین در این مقاله به طراحی یک نبرد انتزاعی بین دو گروه پرداخته شد و منجر به یک مسأله چند هدفی گردید و نشان داده شد که شبکه عصبی معرفی شده در حل انکار است. نتایج عددی نشان دادند که مدل معرفی شده در بالا بردن موفقیت نیروهای خودی و شکست نیروهای دشمن موفق است. بنابراین مدل معرفی شده می‌تواند به‌عنوان یک مشاور برای فرماندهی که برای نیروهای تحت امر خود تصمیم می‌گیرد عمل کند. استفاده از روش شبکه‌های عصبی برای حل مدل‌های نبرد و سناریوهای بازی جنگ

پیشنهاد می‌شود. به‌عنوان پژوهش‌های آتی هدف آن است که با بهبود ساختار شبکه عصبی، زمان شبیه‌سازی را کاهش دهیم.

## منابع

- Bazaraa, M. S., Sherali, H. D. & Shetty, C. M. (2005). *Nonlinear Programming*. John Wiley & Sons, Inc.
- Boutselis, P. & Ringrose, T.J. (2013). GAMLSS and neural networks in combat simulation metamodeling: A case study. *Expert Systems with Applications*, 40(15): 6087-6093.
- Bouzerdoum, A. & Pattison, T. R. (1993). Neural network for quadratic optimization with bound constraints. *IEEE Transactions on Neural Networks*, 4(2): 293-304.
- Effati, S. & Moghaddas, M. (2016). A novel neural network based on NCP function for solving constrained nonconvex optimization problems. *Complexity*, 21(6):130-141.
- Effati, S., Ghomashi, A., & Abbasi, M. (2011). A novel recurrent neural network for solving MLCPs and its application to linear and quadratic programming. *Asia Pacific Journal of Operational Research*, 28 (4): 523-541.
- Gao, X. & Liao, L. Z. (2010). A new one-layer neural network for linear and quadratic programming. *IEEE Transactions on Neural Networks*, 21(6): 918-929.
- Kennedy, M. & Chua, L. (1988). Neural networks for nonlinear programming. *IEEE Transactions on Circuits and Systems*, 35(5): 554-562.
- Kilmer, R. A. (1996). Applications of artificial neural networks to combat simulations. *Mathematical and Computer Modelling*, 23(1-2): 91-99.
- Law, A. & Kelton, W. (1991). *Simulation Modeling and Analysis*, McGraw-Hill, New York.
- Maa, C. Y. & Shanblatt, M. A. (1992). A two-phase optimization neural network. *IEEE Transactions on Neural Networks*, 3(6):1003-1009.
- Miller, R. & Michel, A. (1982). *Ordinary Differential Equations*. Academic Press, Inc.
- Nazemi, A. & Effati, S. (2013). An application of a merit function for solving convex programming problems. *Computers & Industrial Engineering*, 66(2): 212-221.
- Nazemi, A. (2012). A dynamic system model for solving convex nonlinear optimization problems. *Communications in Nonlinear Science and Numerical Simulation*, 17(4):1696-1705.
- Oswalt, I. (1993). Current applications, trends, and organizations in the U.S. military simulation and gaming. *Simulation and Gaming*, 24(2):153-189.

- Rodriguez-Vazquez, A., Dominguez-Castro, R., Rueda, A., Huertas, J., & Sanchez-Sinencio, E. (1990). Nonlinear switched capacitor 'neural' networks for optimization problems. *IEEE Transactions on Circuits and Systems*, 37(3): 384–398.
- Sakawa, M., (1993) *Fuzzy Sets and Interactive Multiobjective Optimization*. Plenum Press, New York and London.
- Tank, D. & Hopfield, J.J. (1986). Simple 'neural' optimization networks: An a/d converter, signal decision circuit, and a linear programming circuit. *IEEE Transactions on Circuits and Systems*, 33(5): 533–541.
- Teng, T., Tan, A., Tan, Y., and Yeo, A. (2012). Self-organizing neural networks for learning air combat maneuvers, *The 2012 International Joint Conference on Neural Networks (IJCNN)*, Brisbane, QLD, pp. 1-8.
- Xia, Y. & Wang, J. (2004). A recurrent neural network for nonlinear convex optimization subject to nonlinear inequality constraints. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 51(7): 1385–1394.
- Xia, Y. & Wang, J. (2005). A recurrent neural network for solving nonlinear convex programs subject to linear constraints. *IEEE Transactions on Neural Networks*, 16(2): 379–386.
- Xia, Y., Leung, H., and Wang, J. (2002). A projection neural network and its application to constrained optimization problems. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 49(4), pp. 447–458.
- Xue, Q. et al. (2010). Improved LMBP algorithm in the analysis and application of simulation data, *International Conference on Computer Application and System Modeling (ICASM 2010)*, Taiyuan, pp. 545–547.
- Xue, X. & Bian, W. (2007). A project neural network for solving degenerate convex quadratic program. *Neurocomputing*, 70(13-15): 2449–2459.
- Yu, P.L. (1973). A class of solutions for group decision problems. *Management Science*, 19: 936-946.
- Zhang, S. & Constantinides, A. (1992). Lagrange programming neural networks. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 39(7): 441–452.