دانشگاه صنعتی اصفهان
دانشکده علوم ریاضی

بهینه‌سازی دینامیکی در محیط‌های چندعاملی

بررسی نظری و کاربردی بازی‌های دیفرانسیلی

پروژه کارشناسی

نام دانشجو: علی عبدالوند

استاد راهنما: دکتر رسول عاشقی

خرداد ۱۴۰۵

تقدیم به کسانی که سرآغاز تولد من هستند؛
از یکی زاده می‌شوم و از دیگری جاودانه؛
پدری که سپیدی را بر تخته سیاه زندگیم کشاند؛
و مادری که تار مویی از آن به پایم سیاه نماند.

تشکر و قدردانی

«... وَقُل رَّبِّ زِدْنِی عِلْمًا...» (سوره طه، آیه ۱۱۴)

با استعانت از درگاه قادر متعال، که منبع لایزال علم و هدایت است، فصل پایانی دوره کارشناسی را در رشته ریاضی، که با تلاش و ممارست به سرانجام رسیده است، آغاز می‌کنم.

در این مرحله، وظیفه خود می‌دانم مراتب قدردانی و سپاس خود را تقدیم نمایم به:

استاد راهنمای ارجمند، جناب آقای دکتر رسول عاشقی:

از دانش عمیق، راهنمایی‌های راهگشا و حمایت‌های بی‌دریغ ایشان در تمامی مراحل تدوین و اجرای این پژوهش، کمال امتنان را دارم. سعه صدر و دقت نظر جنابعالی، چراغ راهی در مسیر تحقق اهداف علمی این پروژه بوده است.

اساتید گرامی دانشکده علوم ریاضی دانشگاه صنعتی اصفهان:

از آموزش‌های ارزشمند و تجربیات گرانبهایی که در طول دوران تحصیلم در کلاس درس اساتید گرانقدر این دانشکده کسب نمودم، صمیمانه سپاسگزارم. آموخته‌هایم از محضر شما، همواره چراغ راه من خواهد بود.

خانواده گرامی:

از پدر و مادر عزیزم که همواره با تکیه‌گاه معنوی و حمایت‌های بی‌شائبه خود، پشتوانه اصلی اینجانب در تمامی مراحل تحصیلی و پژوهشی بوده‌اند، سپاسگزارم. ایثار و همراهی همیشگی آنان، گرانبهاترین سرمایه من است.

همکلاسی‌ها و دوستان ارجمند:

از همراهی، همدلی و تبادل دانش با دوستان و همکلاسی‌های گرامی در دانشکده علوم ریاضی دانشگاه صنعتی اصفهان که در فضایی علمی و دوستانه، همکاری ارزشمندی را با اینجانب در طول دوران تحصیل و نگارش این پایان‌نامه داشته‌اند، تشکر و قدردانی می‌نمایم.

توفیق روزافزون همگان را از درگاه ایزد منان مسألت دارم.

علی عبدالوند

aliabdolvandmath@gmail.com

چکیده

این پروژه به بررسی عمیق و جامع مبحث بهینه‌سازی سیستم‌های دینامیکی با تمرکز ویژه بر بازی‌های دیفرانسیلی می‌پردازد. در دنیای پیچیده امروز، بسیاری از سیستم‌ها شامل چندین عامل تصمیم‌گیرنده هستند که هر یک اهداف مشخص و گاهاً متضادی را دنبال می‌کنند. مدلسازی و کنترل چنین سیستم‌هایی با استفاده از رویکردهای سنتی کنترل بهینه، که برای یک تصمیم‌گیرنده واحد طراحی شده‌اند، ناکافی است. اینجاست که بازی‌های دیفرانسیلی به‌عنوان ابزاری قدرتمند برای تحلیل تعاملات استراتژیک بین عوامل در طول زمان وارد صحنه می‌شوند و نقش ایفا می‌کنند [۱].

جهان پیرامون ما مملو از سیستم‌های پویا و تعاملات پیچیده بین عامل‌های تصمیم‌گیر است. از تعامل بین تولیدکنندگان و مصرف‌کنندگان در بازارهای مالی گرفته تا رقابت بین شرکت‌های تجاری، نبردهای نظامی بین نیروهای متخاصم، و حتی ترافیک شهری بین رانندگان، همگی نمونه‌هایی از محیط‌های چندعاملی محسوب می‌شوند. در تمام این موارد، هر عامل (اعم از انسان، شرکت، ربات یا نرم‌افزار) به دنبال بهینه‌سازی عملکرد خود است، اما این بهینه‌سازی مستقل نیست؛ زیرا عملکرد هر عامل به‌شدت تحت تأثیر تصمیمات دیگر عامل‌ها قرار دارد.

نظریه بازی‌ها به‌عنوان شاخه‌ای از ریاضیات کاربردی، ابزار قدرتمندی برای تحلیل این نوع تعاملات استراتژیک فراهم می‌کند. در حالت‌های ایستا، بازی‌ها با ماتریس‌های بازده مدل‌سازی می‌شوند، اما بسیاری از مسائل دنیای واقعی ذاتاً پویا هستند و وضعیت سیستم در طول زمان تغییر می‌کند. برای مدل‌سازی این تغییرات زمانی و رفتار پویای سیستم، **بازی‌های دیفرانسیلی** که تلفیقی از نظریه بازی‌ها و کنترل بهینه هستند، به کمک ما می‌آیند [۲]. بازی‌های دیفرانسیلی اولین بار توسط روفوس ایزاکز در دهه ۱۹۵۰ و در زمینه مسائل نظامی (مانند تعقیب و گریز) مطرح شدند [۱]. از آن زمان تاکنون، این نظریه توسعه چشمگیری یافته و کاربردهای متنوعی در اقتصاد، مدیریت، علوم سیاسی، زیست‌شناسی، مهندسی و هوش مصنوعی پیدا کرده است [۸].

کلمات کلیدی: سیستم‌های دینامیکی، کنترل بهینه، بازی‌های دیفرانسیلی، تعادل نش، برنامه‌ریزی دینامیکی، اصل ماکسیمم پونتریاگین، یادگیری تقویتی چندعاملی

فهرست مطالب

فصل ۱

مبانی نظری و مرور ادبیات

۱.۱ مقدمه

در این فصل، مبانی مورد نیاز برای درک و تحلیل مسائل بهینه‌سازی دینامیکی در محیط‌های‌چندعاملی ارائه می‌شود. ابتدا مروری بر مفاهیم سیستم‌های دینامیکی، حساب تغییرات، نظریه بازی‌ها و مفاهیم بنیادی آن خواهیم داشت، سپس به معرفی بازی‌های دیفرانسیلی به‌عنوان تعمیم پویای بازی‌های ایستا می‌پردازیم [۳، ۲].

۲.۱ تعریف و اهمیت سیستم‌های دینامیکی

سیستم‌های دینامیکی به مجموعه‌ای از عناصر و روابطی اطلاق می‌شود که وضعیت آن‌ها در طول زمان تغییر می‌کند و این تغییرات توسط مجموعه‌ای از معادلات (اغلب معادلات دیفرانسیل) قابل توصیف هستند [۳]. از پرتاب یک موشک در فضا تا نوسانات بازار سهام، رشد جمعیت یک گونه زیستی، یا حتی فرآیندهای شیمیایی در یک کارخانه، همگی نمونه‌هایی از سیستم‌های دینامیکی هستند. در واقع، زندگی روزمره ما مملو از پدیده‌هایی است که با دینامیک پیچیده گره خورده‌اند. اهمیت مطالعه این سیستم‌ها در توانایی آن‌ها برای **مدل‌سازی دقیق رفتار پدیده‌ها**، **پیش‌بینی حالت‌های آینده**، و در نهایت **اعمال کنترل برای دستیابی به اهداف مطلوب** نهفته است. این توانایی‌ها در حوزه‌های مختلفی مانند مهندسی (کنترل ربات‌ها، طراحی مدارهای الکتریکی)، فیزیک (مدل‌سازی حرکت سیارات، مکانیک سیالات)، زیست‌شناسی (دینامیکی جمعیت‌ها، انتشار بیماری‌ها) و اقتصاد (مدل‌های رشد، نوسانات اقتصادی) بسیار حیاتی هستند [۹].

یک سیستم دینامیکی، وضعیت خود را در طول زمان تغییر می‌دهد و این تغییرات معمولاً با معادلات دیفرانسیل توصیف می‌شوند. رایج‌ترین فرم مدل‌سازی یک سیستم دینامیکی، نمایش فضای حالت است:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t),$$

- که در آن، $\mathbf{x}(t) \in \mathbb{R}^n$ بردار حالت سیستم در زمان t است. این بردار شامل متغیرهایی است که وضعیت فعلی سیستم را به‌طور کامل توصیف می‌کنند (مثلاً موقعیت، سرعت، دما، غلظت ماده).
- $\mathbf{u}(t) \in \mathbb{R}^m$ بردار ورودی کنترل است. این‌ها متغیرهایی هستند که می‌توانیم آن‌ها را دستکاری کنیم تا بر رفتار سیستم تأثیر بگذاریم (مانند نیروی خارجی، ولتاژ، نرخ تولید).

- $\mathbf{f}$ یک تابع برداری (معمولاً غیرخطی) است که دینامیک سیستم را توصیف می‌کند.
- t زمان است.

سیستم‌های دینامیکی می‌توانند **خطی** یا **غیرخطی** باشند. در سیستم‌های خطی، میدان برداری $\mathbf{f}$ یک تابع خطی از $\mathbf{x}$ و $\mathbf{u}$ است که حل و تحلیل آن‌ها ساده‌تر است. اما اکثر سیستم‌های واقعی **غیرخطی** هستند که پیچیدگی‌های محاسباتی را به همراه دارند [۳].

۳.۱ نظریه بازی‌ها: مفاهیم پایه

نظریه بازی‌ها چارچوبی ریاضی برای تحلیل تعاملات استراتژیک بین عامل‌های عاقل است [۲]. هر بازی شامل سه عنصر اصلی است:

۱. **بازیکنان :** مجموعه عامل‌های تصمیم‌گیرنده $N = \{1, 2, ..., n\}$.

۲. **استراتژی‌ها:** مجموعه اقداماتی که هر بازیکن می‌تواند انجام دهد. استراتژی بازیکن i با $u_i \in U_i$ نشان داده می‌شود.

۳. **توابع پرداخت:** در نظریه بازی‌ها، تابع پرداخت ابزاری ریاضی برای اندازه‌گیری میزان رضایت یا سود یک بازیکن از نتیجه‌ی نهایی بازی است [۲]. اگر N بازیکن داشته باشیم و S_i مجموعهٔ استراتژی‌های بازیکن i باشد، آنگاه یک نیمرخ استراتژی به صورت

$$s = (s_1, s_2, \ldots, s_n);$$

نوشته می‌شود. تابع پرداخت بازیکن i نیز به صورت زیر تعریف می‌شود:

$$u_i : S \to \mathbb{R},$$

که در آن u_i نتیجهٔ انتخاب استراتژی‌های همهٔ بازیکنان را به یک عدد حقیقی نسبت می‌دهد.

تابع پرداخت در تحلیل رفتار عقلایی بازیکنان اهمیت زیادی دارد، زیرا هر بازیکن معمولاً می‌کوشد مقدار پرداخت خود را بیشینه کند. از این رو، مفاهیمی مانند تعادل نش بر پایهٔ همین توابع تعریف می‌شوند [۲].

توابع پرداخت را می‌توان در دو قالب اصلی نمایش داد:

- **فرم نرمال** : نمایش جدولی برای بازی‌های ساده و ماتریسی.
- **فرم گسترده**: نمایش درختی برای بازی‌های مرحله‌ای.

در بازی‌های تصادفی یا دارای عدم قطعیت، از **امید ریاضی** برای محاسبهٔ پرداخت مورد انتظار استفاده می‌شود:

$$E[u_i] = \sum p(\omega)\, u_i(\omega).$$

۱.۳.۱ دسته‌بندی بازی‌ها

- **بر اساس همکاری:** بازی‌های همکارانه (بازیکنان می‌توانند با یکدیگر ائتلاف کنند) و غیرهمکارانه (هر بازیکن به دنبال منافع شخصی خود است) [۲].
- **بر اساس اطلاعات:** بازی با اطلاعات کامل (ساختار بازی برای همه معلوم است) و اطلاعات ناقص [۲].
- **بر اساس مجموع بازده‌ها:**بازی‌های حاصل‌جمع‌ـصفر (منافع یک بازیکن مساوی با ضرر دیگری است) و حاصل‌جمع‌ـغیرصفر [۷].

۲.۳.۱ مفهوم تعادل

مهم‌ترین مفهوم در نظریه بازی‌ها، تعادل است که پیش‌بینی می‌کند نتیجه یک بازی عاقلانه چه خواهد بود [۲].

تعریف ۱.۱ (تعادل نش). یک ترکیب استراتژی $(u_1^*, u_2^*, ..., u_n^*)$ تعادل نش نامیده می‌شود اگر هیچ بازیکنی با تغییر یک‌جانبه استراتژی خود نتواند بازدهی خود را بهبود بخشد. به بیان ریاضی:

$$J_i(u_i^*, u_{-i}^*) \leq J_i(u_i, u_{-i}^*) \quad \forall u_i \in U_i, \quad \forall i \in N,$$

که در آن u_{-i}^* نشان‌دهنده استراتژی همه بازیکنان به جز i در حالت تعادل است.

تعریف ۲.۱ (تعادل استکلبرگ). در بازی‌های سلسله‌مراتبی که یک بازیکن به‌عنوان رهبر و دیگران به‌عنوان پیرو عمل می‌کنند، رهبر ابتدا استراتژی خود را اعلام کرده و سپس پیروان با آگاهی از تصمیم رهبر، استراتژی بهینه خود را انتخاب می‌کنند [۲].

۴.۱ بهینگی پارتو

یک وضعیت (توزیع منابع یا استراتژی) را «بهینه پارتو» می‌نامیم، اگر هیچ راهی برای بهبود وضعیتِ حداقل یک نفر وجود نداشته باشد، مگر آنکه وضعیت حداقل یک نفر دیگر بدتر شود [۲].

به‌عبارت دیگر، در یک وضعیت غیربهینه، ما می‌توانیم با تغییر شرایط، کسی را بدون آسیب‌رساندن به دیگری، خوشحال‌تر کنیم. اما وقتی به «بهینگی پارتو» رسیدیم، «برد‌ـبرد» دیگری وجود ندارد و هرگونه پیشرفت برای یک نفر، مستلزم هزینه برای نفر دیگر است.

به زبان ریاضی، استراتژی s^* بهینه پارتو است اگر هیچ استراتژی دیگری مانند s وجود نداشته باشد،به‌گونه‌ای که برای همه بازیکنان،مطلوبیت حاصل از s کمتر از مطلوبیت حاصل از s^* نباشد و درعین‌حال، برای دست‌کم یک بازیکن، این مطلوبیت به طور اکید بیشتر باشد.

این مفهوم نشان‌دهنده یک مرز کارایی است؛ فراتر از این نقطه، هرگونه بهینه‌سازی فردی، هزینه‌های اجتماعی (تحمیل ضرر به دیگری) در بر خواهد داشت [۸].

۵.۱ مثال دوراهی زندانی: تعادل نش در مقابل بهینگی پارتو

سناریو

دو مظنون جدا از هم بازجویی می‌شوند. هر کدام می‌توانند سکوت یا اعتراف را انتخاب کنند.

تابع پرداخت: مجازات بر حسب سال زندان

جدول زیر، مجازات‌ها را برای هر دو بازیکن نشان می‌دهد. عدد اول مربوط به بازیکن ردیف و عدد دوم مربوط به بازیکن ستون است:

بازیکن ۱ / بازیکن ۲	سکوت	اعتراف
سکوت	$(1,1)$	$(10,0)$
اعتراف	$(0,10)$	$(5,5)$

تحلیل

تعادل نش:

- اگر بازیکن ۲ سکوت کند، بازیکن ۱ با اعتراف کردن، یعنی دریافت مجازات صفر سال، بهتر از سکوت کردن، یعنی دریافت مجازات یک سال، عمل می‌کند.
- اگر بازیکن ۲ اعتراف کند، بازیکن ۱ با اعتراف کردن، یعنی دریافت مجازات پنج سال، بهتر از سکوت کردن، یعنی دریافت مجازات ده سال، عمل می‌کند.
- بنابراین، اعتراف، استراتژی غالب بازیکن ۱ است. به‌دلیل تقارن، اعتراف، استراتژی غالب بازیکن ۲ نیز هست.
- **تعادل نش:** وضعیت «اعتراف، اعتراف» با مجازات $(5,5)$ است. در این نقطه، هیچ بازیکنی با تغییر یک‌جانبه استراتژی خود، وضعیتش را بهتر نمی‌کند.

بهینگی پارتو

- **وضعیت «سکوت، سکوت» با مجازات $(1,1)$:** این وضعیت **بهینه پارتو** است؛ زیرا نمی‌توان مجازات یکی از بازیکنان را کاهش داد، مگر اینکه مجازات بازیکن دیگر افزایش یابد.
- **وضعیت «اعتراف، اعتراف» با مجازات $(5,5)$:** این وضعیت **بهینه پارتو نیست**؛ زیرا وضعیت $(1,1)$ برای هر دو بازیکن بهتر است.
- وضعیت‌های $(10,0)$ و $(0,10)$ نیز **بهینه پارتو نیستند**؛ زیرا وضعیت $(1,1)$ برای هر دو بازیکن بهتر است.

نتیجه‌گیری

در دوراهی زندانی، تعادل نش یعنی وضعیت $(5,5)$ بهینه پارتو نیست. رسیدن به وضعیت بهینه پارتو، یعنی وضعیت $(1,1)$، نیازمند همکاری و اعتماد است؛ اما در چارچوب منطق خودخواهانه بازی، که در آن هر بازیکن تنها به دنبال کاهش مجازات خود است، چنین همکاری به‌صورت طبیعی شکل نمی‌گیرد.

۶.۱ مقدمه‌ای بر کنترل بهینه

کنترل بهینه شاخه‌ای از ریاضیات کاربردی است که به دنبال یافتن یک قانون کنترل برای یک سیستم دینامیکی است، به‌گونه‌ای که یک معیار عملکرد یا یک تابع هزینه، حداقل (یا حداکثر) شود [۳]. به بیان ساده، هدف کنترل بهینه این است که سیستم را در طول زمان به بهترین شکل ممکن هدایت کنیم، در حالی که محدودیت‌های فیزیکی یا عملیاتی نیز رعایت شوند. این حوزه که در دهه ۱۹۵۰ با کارهای پیشگامانه‌ای نظیر لو پونتریاگین و ریچارد بل‌من شکوفا شد، به ابزارهای قدرتمندی مانند اصل ماکسیمم پونتریاگین (PMP) و برنامه‌ریزی دینامیکی مجهز است [۹، ۱۰]. اصل ماکسیمم پونتریاگین شرایط لازم برای بهینگی را به‌صورت معادلات دیفرانسیل ارائه می‌دهد، در حالی که برنامه‌ریزی دینامیکی از طریق معادله **هامیلتون–جاکوبی–بل‌من (HJB)**، راه حلی برای کنترل فیدبک ارائه می‌دهد [۱۱]. کنترل بهینه در زمینه‌هایی نظیر هدایت موشک‌ها، بهینه‌سازی فرآیندهای صنعتی و مدیریت منابع، کاربردهای گسترده‌ای یافته است. با این حال، رویکردهای سنتی کنترل بهینه عمدتاً برای سیستم‌هایی طراحی شده‌اند که توسط یک تصمیم‌گیرنده واحد کنترل می‌شوند و هدف آن‌ها بهینه‌سازی یک تابع هزینه منفرد است. در دنیای واقعی، بسیاری از سناریوها شامل تعامل بین چندین تصمیم‌گیرنده هستند که هر یک منافع خود را دنبال می‌کنند، و اینجا است که کنترل بهینه به تنهایی دیگر کافی نیست.

بازی‌های دیفرانسیلی را می‌توان تعمیم کنترل بهینه به محیط‌های چندعاملی دانست [۱]. در کنترل بهینه، یک عامل منفرد به دنبال یافتن سیگنال کنترلی $u(t)$ است تا تابع هزینه خود را با توجه به دینامیک سیستم کمینه کند:

$$\min_{u(\cdot)} J = \int_0^{t_f} L(t, x(t), u(t))dt + \phi(x(t_f))$$

تحت دینامیک:

$$\dot{x}(t) = f(t, x(t), u(t)), \quad x(0) = x_0$$

۱.۶.۱ اصل ماکسیمم پونتریاگین (PMP)

این اصل شرایط لازم برای بهینگی در مسائل کنترل بهینه را فراهم می‌کند [۹]. با تعریف تابع همیلتونی:

$$H(t, x, u, \lambda) = L(t, x, u) + \lambda^T f(t, x, u),$$

شرایط لازم بهینگی عبارتند از:

۱. **معادلات حالت:** $\dot{x}^* = \frac{\partial H}{\partial \lambda}$

۲. **معادلات الحاقی:** $\dot{\lambda}^* = -\frac{\partial H}{\partial x}$

۳. **شرط کمینه‌سازی:** $u^*(t) = \arg\min_{u \in U} H(t, x^*, u, \lambda^*)$

۴. **شرایط اولیه:** $x(0) = x_0$

۷.۱ بازی‌های دیفرانسیلی: تعریف و دسته‌بندی

بازی دیفرانسیلی یک بازی دینامیکی زمان‌ـپیوسته است که در آن وضعیت سیستم بر اساس معادلات دیفرانسیل تکامل می‌یابد و بازیکنان با انتخاب ورودی‌های کنترلی خود در طول زمان، بر این تکامل تأثیر می‌گذارند [۱].

۱.۷.۱ تعریف ریاضی یک بازی دیفرانسیلی n- نفره

- **بازیکنان:** $i = 1, 2, ..., n$
- **متغیر حالت:** $x(t) \in \mathbb{R}^m$ (وضعیت سیستم در زمان t)
- **ورودی کنترلی بازیکن i :** $u_i(t) \in U_i \subset \mathbb{R}^{p_i}$
- **دینامیک سیستم:**

$$\dot{x}(t) = f(t, x(t), u_1(t), u_2(t), ..., u_n(t)), \quad x(t_0) = x_0$$

- **تابع هزینه بازیکن i :** (که به دنبال کمینه کردن آن است)

$$J_i(u_1, u_2, ..., u_n) = \int_{t_0}^{t_f} L_i(t, x(t), u_1(t), ..., u_n(t))dt + \phi_i(x(t_f))$$

۲.۷.۱ دسته‌بندی بازی‌های دیفرانسیلی

۱. **بر اساس افق زمانی:**

- **بازی با افق متناهی:** t_f متناهی است.
- **بازی با افق نامتناهی :** $t_f \to \infty$. معمولاً به دنبال یافتن استراتژی‌های پایدار هستیم.

۲. **بر اساس اطلاعات:**

- **بازی با اطلاعات کامل:** هر بازیکن در هر لحظه از مقدار دقیق متغیر حالت $x(t)$ مطلع است.
- **بازی با اطلاعات ناقص:** بازیکنان تنها مشاهدات نویزی از حالت دارند.
- **بازی با اطلاعات بازخوردی:** استراتژی‌ها تابعی از حالت جاری هستند: $u_i(t) = \gamma_i(t, x(t))$.
- **بازی با اطلاعات حلقه‌باز:** استراتژی‌ها تنها تابعی از زمان هستند: $u_i(t) = \gamma_i(t)$. (معمولاً در عمل کمتر کاربرد دارد زیرا بازیکنان نمی‌توانند به انحرافات پاسخ دهند) [۲].

۳. **بر اساس مجموع پرداخت‌ها:**

- **بازی‌های صفر–جمع :** مجموع توابع هزینه بازیکنان صفر است (یا ثابت). در حالت دو نفره: $J_1 + J_2 = 0$. این بازی‌ها ماهیت کاملاً رقابتی دارند [۷].
- **بازی‌های غیرصفر–جمع :** مجموع توابع هزینه ثابت نیست. این بازی‌ها می‌توانند شامل جنبه‌های رقابتی و همکاری باشند [۲].

۴. **بر اساس تعداد بازیکنان:**

- دو نفره
- چند نفره

۸.۱ تعادل نش در بازی‌های دیفرانسیلی

تعمیم تعادل نش به بازی‌های دیفرانسیلی، مجموعه‌ای از استراتژی‌ها $(u_1^*, ..., u_n^*)$ است به‌گونه‌ای که برای هر بازیکن i و برای هر استراتژی مجاز u_i، داشته باشیم [۲]:

$$J_i(u_i^*, u_{-i}^*) \leq J_i(u_i, u_{-i}^*).$$

۱.۸.۱ شرایط لازم برای تعادل نش (اصل ماکسیمم در بازی‌های دیفرانسیلی)

برای یک بازی دیفرانسیلی غیرصفر–جمع با اطلاعات کامل بازخوردی، شرایط لازم برای وجود تعادل نش، مشابه با شرایط PMP برای هر بازیکن است، با این تفاوت که هر بازیکن همیلتونی مخصوص به خود را دارد [۷، ۲]:
همیلتونی بازیکن i:

$$H_i(t, x, u_1, ..., u_n, \lambda_i) = L_i(t, x, u_1, ..., u_n) + \lambda_i^T f(t, x, u_1, ..., u_n)\,;$$

شرایط:

۱. **معادلات حالت:**

$$\dot{x}^* = f(t, x^*, u_1^*, ..., u_n^*)$$

۲. **معادلات الحاقی برای هر بازیکن i:**

$$\dot{\lambda}_i^* = -\frac{\partial H_i}{\partial x}(t, x^*, u_1^*, ..., u_n^*, \lambda_i^*)$$

۳. **شرط کمینه‌سازی برای هر بازیکن i:**

$$u_i^*(t) = \arg\min_{u_i \in U_i} H_i(t, x^*, u_1^*, ..., u_{i-1}^*, u_i, u_{i+1}^*, ..., u_n^*, \lambda_i^*)$$

۴. **شرایط اولیه:** $x(t_0) = x_0$ و شرایط عرضی $\lambda_i(t_f) = \frac{\partial \phi_i}{\partial x}(x(t_f))$.

دستگاه معادلات حاصل، یک مسئله **مقدار مرزی دو-نقطه‌ای (TPBVP)** است که حل تحلیلی آن معمولاً غیرممکن بوده و نیاز به روش‌های عددی دارد [۳].

۲.۸.۱ معادله همیلتون-ژاکوبی-آیزمن (HJI)

برای بازی‌ها با اطلاعات بازخوردی، می‌توان از برنامه‌ریزی پویا برای استخراج شرایط کافی برای تعادل استفاده کرد. اگر تابع ارزش بازیکن i را به‌صورت $V_i(t,x)$ تعریف کنیم (که نشان‌دهنده حداقل هزینه باقی‌مانده برای بازیکن i از زمان t با حالت x است)، V_i باید در معادله دیفرانسیل جزئی زیر صدق کند:

$$-\frac{\partial V_i}{\partial t} = \min_{u_i \in U_i} \left\{ L_i(t,x,u_1^*,...,u_{i-1}^*,u_i,u_{i+1}^*,...,u_n^*) + \nabla_x V_i^T f(t,x,u_1^*,...,u_{i-1}^*,u_i,u_{i+1}^*,...,u_n^*) \right\},$$

که در آن u_j^* برای $j \neq i$، استراتژی‌های تعادلی سایر بازیکنان هستند. حل این معادله به‌دلیل غیرخطی بودن و ابعاد بالا بسیار دشوار است و به "نفرین ابعاد" معروف است [۲].

۹.۱ بازی‌های دیفرانسیلی خطی-درجه دوم

یک دسته مهم از بازی‌های دیفرانسیلی که قابلیت حل تحلیلی دارند، بازی‌های خطی-درجه دوم هستند [۲]. در این بازی‌ها:

- دینامیک سیستم خطی است: $\dot{x} = Ax + \sum_{i=1}^{N} B_i u_i$
- توابع هزینه درجه دوم هستند:

$$L_i = \frac{1}{2}\left(x^T Q_i x + \sum_{j=1}^{N} u_j^T R_{ij} u_j \right), \quad \phi_i = \frac{1}{2} x^T S_i x.$$

برای بازی‌های خطی-درجه دوم، می‌توان نشان داد که استراتژی‌های تعادلی به‌صورت خطی $u_i^* = -K_i x$ هستند و توابع ارزش به‌صورت درجه دوم $V_i(t,x) = \frac{1}{2} x^T P_i(t) x$ هستند. در این حالت، معادله همیلتون-ژاکوبی-آیزمن به یک معادله دیفرانسیل ریکاتی جفت‌شده تبدیل می‌شود که می‌توان آن را به‌صورت عددی حل کرد.

فصل ۲

مدل‌سازی ریاضی مسئله

۱.۲ مقدمه

در این فصل، چارچوب ریاضی عمومی برای مدل‌سازی مسائل بهینه‌سازی دینامیکی در محیط‌های چندعاملی ارائه می‌دهیم. این چارچوب به‌عنوان پایه‌ای برای تحلیل‌های نظری و پیاده‌سازی‌های عددی در فصول بعدی عمل خواهد کرد. تأکید اصلی بر روی بازی‌های دیفرانسیلی غیرخطی، غیرصفر–جمع و با اطلاعات کامل بازخوردی خواهد بود، اما حالت‌های خاص (مانند بازی‌های صفر–جمع) نیز بررسی خواهند شد [۱، ۲].

۲.۲ تعریف مسئله عمومی

یک بازی دیفرانسیلی با N بازیکن را در نظر بگیرید. مسئله بهینه‌سازی برای بازیکن i به‌صورت زیر تعریف می‌شود [۲]:

$$\min_{u_i(\cdot)\in\mathcal{U}_i} J_i(u_1, u_2, ..., u_N)$$

$$J_i = \int_{t_0}^{t_f} e^{-\rho_i t} L_i(t, x(t), u_1(t), ..., u_N(t))dt + e^{-\rho_i t_f}\phi_i(x(t_f))$$

تحت شرایط زیر

$$\dot{x}(t) = f(t, x(t), u_1(t), ..., u_N(t)), \quad x(t_0) = x_0,$$

$$g_j(t, x(t), u_1(t), ..., u_N(t)) \le 0, \quad j = 1, ..., M \quad \text{(قیود نامساوی)}$$

$$h_k(t, x(t)) \le 0, \quad k = 1, ..., P \quad \text{(قیود حالت)}$$

که در آن

- $\mathcal{U}_i$ مجموعه استراتژی‌های مجاز برای بازیکن i است.
- $\rho_i \ge 0$ نرخ تنزیل بازیکن i است. (نرخ تنزیل عددی است که تعیین می‌کند ارزشِ پول/پاداشِ آینده نسبت به الان چقدر کمتر حساب شود. اگر امروز ۱ واحد ارزش داشته باشیم، ارزش همان ۱ واحد در آینده معمولاً

کمتر در نظر گرفته می‌شود.در مدل‌های تصمیم‌گیری و RLمعمولاً با γ نشان داده می‌شود و در بازهٔ زیر است: $(0 \leq \gamma < 1$

- $L_i : [t_0, t_f] \times \mathbb{R}^m \times U_1 \times ... \times U_N \to \mathbb{R}$ تابع هزینه جاری بازیکن i است.
- $\phi_i : \mathbb{R}^m \to \mathbb{R}$ هزینه نهایی بازیکن i است.
- $f : [t_0, t_f] \times \mathbb{R}^m \times U_1 \times ... \times U_N \to \mathbb{R}^m$ دینامیک سیستم را توصیف می‌کند.
- g_j و h_k قیود مسئله را نشان می‌دهند.

۳.۲ تعادل نش بازخوردی

همان‌طور که اشاره شد، برای بازی‌های پویا، استراتژی‌های بازخوردی (وابسته به حالت) نسبت به استراتژی‌های حلقه‌باز (وابسته به زمان) جذابیت بیشتری دارند، زیرا بازیکنان می‌توانند به انحرافات و شوک‌های وارد شده به سیستم واکنش نشان دهند [۲].

تعریف ۱.۲ (تعادل نش بازخوردی)**.** مجموعه استراتژی‌های بازخوردی $\gamma_i^* : [t_0, t_f] \times \mathbb{R}^m \to U_i$ برای $i = 1, ..., N$ یک تعادل نش بازخوردی نامیده می‌شود اگر برای هر i و برای هر استراتژی بازخوردی مجاز γ_i، و برای تمام مقادیر اولیه $(t, x) \in [t_0, t_f] \times \mathbb{R}^m$، داشته باشیم:

$$J_i(t, x; \gamma_i^*, \gamma_{-i}^*) \leq J_i(t, x; \gamma_i, \gamma_{-i}^*).$$

۴.۲ شرایط لازم برای تعادل نش بازخوردی (اصل ماکسیمم)

اگر توابع L_i و f نسبت به متغیرهای خود مشتق‌پذیر باشند، شرایط لازم برای وجود تعادل نش بازخوردی را می‌توان استخراج کرد، با این تفاوت که متغیرهای الحاقی λ_i اکنون توابعی از زمان و حالت هستند و معادلات الحاقی به‌صورت معادلات دیفرانسیل معمولی در طول مسیر بهینه حل می‌شوند [۷].

برای یک مسیر معین $x^*(t)$ که از استراتژی‌های تعادلی $u_i^*(t) = \gamma_i^*(t, x^*(t))$ ناشی می‌شود، متغیرهای الحاقی $\lambda_i(t)$ باید در معادلات زیر صدق کنند:

۱. **معادلات حالت:**

$$\dot{x}^* = f(t, x^*, u_1^*, ..., u_N^*), \quad x^*(t_0) = x_0$$

۲. **معادلات الحاقی:**

$$\dot{\lambda}_i = -\frac{\partial H_i}{\partial x}(t, x^*, u_1^*, ..., u_N^*, \lambda_i) + \rho_i \lambda_i$$

(توجه: عبارت $+\rho_i\lambda_i$ به‌دلیل وجود عامل تنزیل $e^{-\rho_i t}$ در تابع هزینه اضافه شده است.)

۳. **شرط کمینه‌سازی برای هر t:**

$$u_i^*(t) = \arg\min_{u_i \in U_i} H_i(t, x^*(t), u_1^*(t), ..., u_{i-1}^*(t), u_i, u_{i+1}^*(t), ..., u_N^*(t), \lambda_i(t)),$$

که در آن $H_i = L_i + \lambda_i^T f$.

۴. **شرایط اولیه و عرضی:**

$$x^*(t_0) = x_0, \quad \lambda_i(t_f) = \frac{\partial \phi_i}{\partial x}(x^*(t_f)).$$

۵.۲ شرایط کافی برای تعادل نش بازخوردی معادله معادله همیلتون‌ـ ژاکوبی‌ـ آیزمن

یک روش جایگزین برای یافتن تعادل نش بازخوردی، استفاده از برنامه‌ریزی پویا و حل معادله معادله همیلتون‌ـ ژاکوبی‌ـ آیزمن است [۱]. فرض کنید توابع ارزش $V_i(t, x)$ برای بازیکنان وجود داشته باشند که در معادله زیر صدق کنند:

$$-\frac{\partial V_i}{\partial t} = \min_{u_i \in U_i} \left\{ e^{-\rho_i t} L_i(t, x, u_i, \gamma_{-i}^*(t, x)) + \nabla_x V_i^T f(t, x, u_i, \gamma_{-i}^*(t, x)) \right\},$$

با شرط مرزی $V_i(t_f, x) = e^{-\rho_i t_f} \phi_i(x)$. اگر استراتژی‌های $\gamma_i^*(t, x)$، طرف راست این معادله را برای هر i کمینه کنند، آن‌گاه این استراتژی‌ها یک تعادل نش بازخوردی را تشکیل می‌دهند.

فصل ۳

روش‌های حل پیشنهادی

۱.۳ مقدمه

با توجه به پیچیدگی مسائل معرفی‌شده در فصل قبل، حل تحلیلی آنها به‌جز در موارد خاص مانند بازی‌های خطی‌ـ درجه دوم غیرممکن است. در این فصل، دو دسته روش حل عددی برای تقریب تعادل نش در بازی‌های دیفرانسیلی ارائه می‌دهیم:

(۱) روش ترکیبی تحلیلی‌ـ عددی مبتنی بر اصل ماکسیمم و پرتابی

(۲) روش مبتنی بر یادگیری تقویتی چندعاملی.

در انتها، یک روش ترکیبی، که از هر دو رویکرد بهره می‌گیرد نیز پیشنهاد خواهد شد [۳، ۴].

۲.۳ روش ترکیبی تحلیلی‌ـ عددی

این روش بر پایه تبدیل مسئله بازی دیفرانسیلی به یک مسئله مقدار مرزی دوـ نقطه‌ای با استفاده از شرایط لازم اصل ماکسیمم پونتریاگین استوار است [۳].

۱.۲.۳ ایده اصلی

شرایط لازم اصل ماکسیمم پونتریاگین یک مسئله مقدار مرزی دوـ نقطه‌ای برای متغیرهای حالت $x(t)$ و الحاقی $\lambda_i(t)$ ایجاد می‌کند. هدف، یافتن مقادیر اولیه متغیرهای الحاقی $\lambda_i(t_0)$ است به‌طوری که با انتگرال‌گیری رو به جلو از معادلات حالت و الحاقی، به شرایط نهایی مطلوب $\lambda_i(t_f) = e^{-\rho_i t_f} \cdot \frac{\partial \phi_i}{\partial x}\big(x(t_f)\big)$ برسیم [۹].

۲.۲.۳ الگوریتم روش پرتابی ساده

روش پرتابی ساده یک الگوریتم عددی است که برای حل مسائل کنترل بهینه و بازی‌های دیفرانسیلی استفاده می‌شود، به‌خصوص زمانی که با معادلات دیفرانسیل با شرایط مرزی روبه‌رو هستیم. این روش به‌طور گسترده در حوزه‌هایی مانند رباتیک، اقتصاد و مهندسی کنترل کاربرد دارد [۳].

۳.۲.۳ ایده اصلی روش پرتابی ساده

ایده کلیدی این روش، تبدیل یک مسئله مقدار مرزی که حل آن به‌طور مستقیم دشوار است، به یک یا چند مسئله مقدار اولیه است که حل آنها با روش‌های عددی استاندارد امکان‌پذیر است. این فرآیند شامل مراحل زیر است:

۱. **تبدیل به مسئله مقدار اولیه:** معمولاً دارای شرایط مشخص در نقاط شروع و پایان است. در بسیاری از مسائل، شرایط در نقطه شروع (مثلاً زمان t_0) برای تمام متغیرها مشخص نیست (به‌ویژه متغیرهای کمکی یا متغیرهای هم‌حالت). روش پرتابی این مشکل را با حدس‌زدن مقادیر اولیه نامشخص حل می‌کند.

۲. **حدس‌زدن:** مقادیر اولیه نامشخص (مثلاً $\lambda(t_0)$) را حدس می‌زنیم.

۳. **حل مسئله مقدار اولیه:** با داشتن تمام شرایط اولیه (مقادیر معلوم و حدس‌زده شده)، سیستم معادلات دیفرانسیل را به‌عنوان یک مسئله مقدار اولیه حل می‌کنیم.

۴. **مقایسه نتایج:** نتایج حاصل از حل مسئله مقدار اولیه در نقطه پایانی (مثلاً t_f) را با شرایط مرزی نهایی معلوم مقایسه می‌کنیم.

۵. **اصلاح حدس:** اگر نتایج با شرایط مرزی نهایی مطابقت نداشته باشند، خطای به‌وجود آمده را محاسبه کرده و از آن برای اصلاح حدس اولیه استفاده می‌کنیم (معمولاً با روشی مانند نیوتن–رافسون).

۶. **تکرار:** مراحل ۳ تا ۵ را تکرار می‌کنیم تا زمانی که خطای انطباق با شرایط مرزی نهایی به‌اندازه کافی کوچک شود.

۳.۳ مراحل الگوریتم در بازی‌های دیفرانسیلی

در یک بازی دیفرانسیلی، هدف معمولاً تعیین کنترل‌های بهینه بازیکنان و مسیرهای متناظر متغیرهای حالت و هم‌حالت است. این فرایند به یک مسئله مقدار مرزی منجر می‌شود که شامل معادلات دینامیک سیستم و معادلات مربوط به متغیرهای هم‌حالت است [۳]. فرض کنید متغیر حالت کلی $z(t)$ باشد که شامل متغیرهای حالت سیستم $x(t)$ و ضرایب کمکی $\lambda(t)$ است:

$$\dot{z}(t) = g(z(t), u_1(t), u_2(t))$$

با شرایط مرزی:

$$z(t_0) = z_0 \quad \text{(شرایط اولیه معلوم)}$$

$$z(t_f) = z_f \quad \text{(شرایط نهایی معلوم)}$$

و هدف پیدا کردن استراتژی‌های تعادلی $u_1^*(t)$ و $u_2^*(t)$ است.

۱.۳.۳ گام‌های الگوریتم:

۱. **فرمول‌بندی مسئله مقدار مرزی:** استخراج معادلات دیفرانسیل کامل $\dot{z}(t) = g(z(t), u_1^*(t), u_2^*(t))$ و شرایط مرزی $z(t_0)$ و $z(t_f)$. در این مرحله، $\lambda(t_0)$ معمولاً نامشخص است.

۲. **حدس‌زدن ضرایب کمکی اولیه:** مقادیر اولیه برای ضرایب کمکی، $\lambda(t_0)$، را حدس می‌زنیم:

$$\lambda(t_0) = \lambda_0^{\text{guess}}$$

بردار اولیه $z(t_0)$ را با ترکیب مقادیر معلوم $x(t_0)$ و حدس λ_0^{guess} تشکیل می‌دهیم.

۳. **حل مسئله مقدار مرزی:** با داشتن $z(t_0)$ کامل و با استفاده از استراتژی‌های بهینه $(u_1^*(t), u_2^*(t))$ که از شرایط بهینگی (مانند معادله همیلتون—ژاکوبی—بلمن) به‌دست می‌آیند، سیستم $\dot{z}(t)$ را با روش‌های عددی (مانند اویلر یا رونگه‌ـ کوتا) از t_0 تا t_f حل می‌کنیم. این کار یک مسیر عددی $z(t)$ به ما می‌دهد.

۴. **محاسبه خطای شرط مرزی نهایی:** مقدار نهایی $z(t_f)$ به‌دست آمده را با مقدار مرزی نهایی واقعی z_f مقایسه می‌کنیم:

$$\text{Error} = z_f - z(t_f)$$

۵. **اصلاح حدس اولیه** λ_0: اگر خطا بزرگ باشد، از روش نیوتن‌ـرافسون استفاده می‌کنیم. این روش نیاز به محاسبه ماتریس ژاکوبی $\frac{\partial z(t_f)}{\partial \lambda_0}$ دارد:

$$\lambda_0^{\text{guess new}} = \lambda_0^{\text{guess old}} + \left(\frac{\partial z(t_f)}{\partial \lambda_0}\right)^{-1} \times \text{Error}.$$

محاسبه ژاکوبی، خود نیازمند حل یک دستگاه معادلات دیفرانسیل کمکی دیگر است.

۶. **تکرار:** مراحل ۳ تا ۵ را تکرار می‌کنیم تا زمانی که $\|\text{Error}\| < \epsilon$ برای یک آستانه کوچک ϵ.

۴.۳ مزایا و معایب

۱.۴.۳ مزایا:

- **سادگی مفهومی:** تبدیل مسئله مقدار مرزی به مسئله مقدار اولیه و استفاده از حل‌کننده‌های استاندارد مسئله مقدار اولیه.
- **کارایی در ابعاد کم:** برای مسائلی با تعداد متغیر کم، نسبتاً سریع و کارآمد است.
- **شهودی بودن:** ایده "روش پرتابی" و اصلاح مسیر بر اساس خطا، قابل درک است.

۲.۴.۳ معایب:

- **حساسیت به حدس اولیه:** همگرایی الگوریتم به‌شدت به کیفیت حدس اولیه λ_0^{guess} وابسته است.
- **مشکلات عددی:** در افق‌های زمانی طولانی یا سیستم‌های ناپایدار، خطاهای عددی انباشته‌شده می‌توانند مانع همگرایی شوند.
- **پیچیدگی محاسبه ژاکوبی:** محاسبه ژاکوبی مورد نیاز برای روش نیوتن می‌تواند بسیار دشوار باشد.

- **عدم تضمین یافتن تمام تعادل‌ها:** این روش ممکن است فقط یک تعادل از بین تعادل‌های متعدد احتمالی را پیدا کند.

۵.۳ کاربرد در بازی‌های دیفرانسیلی

روش پرتابی ساده، یکی از روش‌های عددی مؤثر برای یافتن راهبردهای تعادلی در بازی‌های دیفرانسیلی است. این روش به‌طور خاص برای حل دستگاه معادلات حالت و هم‌حالت ناشی از شرایط لازم مرتبه اول که از اصل بیشینگی پونتریاگین به‌دست می‌آیند، به‌کار می‌رود. در این رویکرد، مقادیر اولیه نامعلوم متغیرهای هم‌حالت حدس زده می‌شوند و مسئله مقدار مرزی حاصل به یک مسئله مقدار اولیه تبدیل می‌شود. سپس با حل عددی دستگاه و محاسبه خطای شرایط مرزی در زمان نهایی، حدس‌های اولیه به‌صورت تکراری اصلاح می‌شوند تا شرایط مرزی مورد نظر برآورده شوند. این الگوریتم در برخی مسائل با ساختار مشخص، از جمله بازی‌های دیفرانسیلی خطی—درجه دوم، کاربرد دارد [۳، ۹].

الگوریتم روش پرتابی ساده به‌شرح زیر است:

Algorithm 1 Simple Shooting Method Algorithm for Solving Differential Games

Require: Initial guess $\lambda_i(t_0)$ for all i, initial conditions $x(t_0) = x_0$, time interval $[t_0, t_f]$
Ensure: Optimal strategies $u_i^*(t)$ and state trajectory $x^*(t)$

while Convergence condition not met **do**
  Integrate the following system of differential equations from t_0 to t_f, using $x(t_0) = x_0$ and the guesses $\lambda_i(t_0)$:
    $\dot{x} = f(t, x, u_1^*, ..., u_N^*)$
    $\dot{\lambda}_i = -\frac{\partial H_i}{\partial x} + \rho_i \lambda_i$ for all i
    where $u_i^* = \arg\min_{u_i} H_i(t, x, u_1^*, ..., u_{i-1}^*, u_i, u_{i+1}^*, ..., u_N^*, \lambda_i)$
  Calculate the error in the final condition: $E = [\lambda_i(t_f) - \partial\phi_i/\partial x(x(t_f))]$ for all i
  if $\|E\| < \epsilon$ **then**
    return $u_i^*(t), x^*(t)$
  else
    Update the guess $\lambda_i(t_0)$ using Newton's method or an optimization algorithm
  end if
end while

۱.۵.۳ روش پرتابی چندگانه

روش پرتابی ساده به حدس اولیه بسیار حساس است و ممکن است واگرا شود. روش پرتابی چندگانه با تقسیم بازه زمانی $[t_0, t_f]$ به M زیربازه، این مشکل را کاهش می‌دهد. در هر گره، مقادیر حالت و متغیرهای الحاقی به‌عنوان متغیرهای تصمیم در نظر گرفته می‌شوند و شرایط پیوستگی بین بازه‌ها به‌عنوان قیود مسئله بهینه‌سازی اضافه می‌شوند [۹].

۲.۵.۳ مزایا و معایب روش پرتابی و اصل ماکسیمم پونتریاگین

- **مزایا:** دقت بالا در صورت همگرایی، استفاده از دانش دینامیک سیستم، ارائه مستقیم مسیرهای بهینه.
- **معایب:** نیاز به مشتق‌پذیری توابع، حساسیت به حدس اولیه، دشواری اعمال قیود حالت، افزایش شدید پیچیدگی با تعداد بازیکنان [۳].

۶.۳ روش مبتنی بر یادگیری تقویتی چندعاملی (MADDPG)

برای غلبه بر محدودیت‌های روش کلاسیک، به سراغ روش‌های یادگیری تقویتی می‌رویم. این روش‌ها نیازی به مدل دینامیک (Model-free) ندارند و می‌توانند با ابعاد بالاتر کنار بیایند [۴].

۱.۶.۳ الگوریتم MADDPG

این الگوریتم که توسط لو و همکاران (۲۰۱۷) معرفی شد، یکی از محبوب‌ترین روش‌ها برای یادگیری استراتژی‌های پیوسته در محیط‌های چندعاملی است [۵]. ایده اصلی آن "مرکزی‌سازی یادگیری، غیرمتمرکزسازی اجرا" است.

- **بازیگران:** هر بازیکن i دارای یک شبکه عصبی به‌نام $\mu_i(o_i)$ (بازیگر) است که بر اساس مشاهده محلی o_i از محیط، عمل u_i را انتخاب می‌کند.
- **نقادان:** در مرحله یادگیری، هر بازیکن i یک شبکه نقاد $Q_i^{\mu}(o_1, ..., o_N, u_1, ..., u_N)$ دارد که با مشاهده اعمال و مشاهدات همه بازیکنان، تخمینی از تابع ارزش ارائه می‌دهد [۵].

Algorithm 2 MADDPG Algorithm for Differential Games

Require: Number of players N, actor and critic network architectures, hyperparameters

Ensure: Learned policies μ_i for each player

Initialize actor network parameters θ_i^{μ} and critic network parameters θ_i^{Q} for each player
Initialize target network parameters $\theta_i^{\mu'} \leftarrow \theta_i^{\mu}$, $\theta_i^{Q'} \leftarrow \theta_i^{Q}$
Initialize experience replay buffer $\mathcal{D}$
for episode = 1 to M **do**
    Get initial state x_0
    for t = 1 to T **do**
        For each player i, select action $u_i = \mu_i(o_i) +$ exploration noise
        Execute actions $(u_1, ..., u_N)$ and receive rewards $(r_1, ..., r_N)$ and next state x'
        Store experience $(x, u_1, ..., u_N, r_1, ..., r_N, x')$ in $\mathcal{D}$
        $x \leftarrow x'$
    end for
    for player i = 1 to N **do**
        Sample a random minibatch from $\mathcal{D}$
        Update the critic network by minimizing the TD error:

$$L(\theta_i^{Q}) = \mathbb{E}\left[(Q_i^{\mu}(x, u_1, ..., u_N) - y_i)^2\right]$$

$$y_i = r_i + \gamma Q_i^{\mu'}(x', \mu_1'(o_1'), ..., \mu_N'(o_N'))$$

        Update the actor network using the policy gradient ascent:

$$\nabla_{\theta_i^{\mu}} J \approx \mathbb{E}\left[\nabla_{\theta_i^{\mu}} \mu_i(o_i) \nabla_{u_i} Q_i^{\mu}(x, u_1, ..., u_N)|_{u_i = \mu_i(o_i)}\right]$$

    end for
    Update the target networks with rate τ:

$$\theta_i^{\mu'} \leftarrow \tau\theta_i^{\mu} + (1-\tau)\theta_i^{\mu'}$$

$$\theta_i^{Q'} \leftarrow \tau\theta_i^{Q} + (1-\tau)\theta_i^{Q'}$$

end for

۲.۶.۳ مزایا و معایب روش MADDPG

- **مزایا:** عدم نیاز به مدل دینامیکی، به این معنا که روش از نوع بدون‌مدل بوده و برای یادگیری نیازی به داشتن معادلات دقیق دینامیکی سیستم یا مدل ریاضی صریح از محیط ندارد؛ توانایی کار با ابعاد بالا و توابع غیرخطی؛ و قابلیت اجرای غیرمتمرکز [۴، ۵].
- **معایب:** نیاز به تنظیم دقیق پارامترها، عدم تضمین همگرایی به تعادل نش واقعی، ناپایداری در یادگیری، نیاز به داده‌های زیاد [۵].

۷.۳ روش ترکیبی پیشنهادی

برای بهره‌گیری از نقاط قوت هر دو روش، یک روش ترکیبی جدید پیشنهاد می‌دهیم. ایده اصلی این است که از دانش دینامیک (که در اصل ماکسیمم پونتریاگین استفاده می‌شود) برای هدایت فرآیند یادگیری در MADDPG استفاده کنیم [۳].

۱.۷.۳ مراحل روش ترکیبی

۱. **پیش‌آموزش با روش پرتابی و اصل ماکسیمم پونتریاگین:** برای یک نسخه ساده‌شده از مسئله (با ابعاد کمتر یا دینامیک خطی‌شده)، از روش پرتابی و اصل ماکسیمم پونتریاگین برای یافتن یک تقریب اولیه از استراتژی‌های بهینه استفاده می‌کنیم.

۲. **مقداردهی اولیه شبکه‌ها:** شبکه‌های بازیگر در MADDPG را با استراتژی‌های حاصل از مرحله قبل مقداردهی اولیه می‌کنیم. این کار باعث می‌شود یادگیری از یک نقطه شروع خوب آغاز شود، نه از صفر [۵].

۳. **یادگیری با MADDPG :** سپس فرآیند یادگیری MADDPG را با این مقداردهی اولیه آغاز می‌کنیم. شبکه‌های بازیگر و نقاد با تعامل با محیط (محیط اصلی با دینامیک کامل) به یادگیری ادامه می‌دهند [۴].

۴. **استفاده ازروش پرتابی و اصل ماکسیمم پونتریاگین به‌عنوان تابع پاداش کمکی:** می‌توان از اصل ماکسیمم برای تعریف یک پاداش کمکی استفاده کرد. اگر استراتژی جاری به شرایط روش پرتابی و اصل ماکسیمم پونتریاگین نزدیک باشد (مثلاً شرط کمینه‌سازی همیلتونی به‌طور تقریبی برقرار باشد)، پاداش اضافی به عامل داده شود.

۲.۷.۳ مزایای روش ترکیبی

- **شروع بهتر:** مقداردهی اولیه با روش پرتابی و اصل ماکسیمم پونتریاگین سرعت همگرایی را افزایش داده و از افتادن در بهینه‌های موضعی ضعیف جلوگیری می‌کند.
- **پایداری بیشتر:** دانستن دینامیک به‌عنوان یک "تنظیم‌کننده" عمل کرده و یادگیری را پایدارتر می‌کند.
- **تعمیم‌پذیری:** این روش می‌تواند برای مسائل پیچیده‌تری که روش پرتابی و اصل ماکسیمم پونتریاگین به تنهایی قادر به حل آنها نیست، استراتژی‌های خوبی بیاموزد.

۸.۳ معیارهای ارزیابی

برای مقایسه روش‌ها، از معیارهای زیر استفاده خواهیم کرد:

- **خطای نسبی هزینه:** $\frac{|J_i^{method} - J_i^*|}{|J_i^*|}$ (اگر J_i^* از حل تحلیلی در دسترس باشد).
- **شاخص تعادل نش :** معیاری برای سنجش انحراف از تعادل نش. برای مثال، حداکثر بهبود ممکن برای هر بازیکن در صورت انحراف یک‌جانبه.
- **زمان محاسبات:** زمان لازم برای همگرایی روش.
- **پایداری:** حساسیت نتایج به تغییرات پارامترها و شرایط اولیه.
- **کارایی در ابعاد بالا:** عملکرد روش با افزایش تعداد بازیکنان یا متغیرهای حالت.

فصل ۴

مطالعات موردی و شبیه‌سازی

۱.۴ مقدمه

در این فصل، روش‌های حل معرفی‌شده در فصل قبل بر روی سه مطالعه موردی با ویژگی‌های متفاوت پیاده‌سازی و ارزیابی می‌شوند. هدف، بررسی کارایی، دقت و محدودیت‌های هر روش در مسائل متنوع است [۸، ۱].

۲.۴ مطالعه موردی اول: رقابت در بازار انرژی (بازی غیرصفر–جمع)

۱.۲.۴ شرح مسئله

دو شرکت تولیدکننده برق (بازیکنان ۱ و ۲) را در نظر بگیرید که در یک بازار رقابتی فعالیت می‌کنند. هر شرکت در زمان t میزان تولید خود را $q_i(t)$ (برحسب مگاوات) تعیین می‌کند. قیمت بازار $p(t)$ تابعی از کل عرضه است: $p(t) = a - b(q_1(t) + q_2(t))$. هزینه تولید شرکت i برابر $c_i q_i(t) + \frac{1}{2} d_i q_i(t)^2$ است. همچنین، هر شرکت با سرمایه‌گذاری $I_i(t)$ می‌تواند ظرفیت تولید خود $k_i(t)$ را افزایش دهد. دینامیک ظرفیت تولید به‌صورت زیر است:

$$\dot{k}_i(t) = I_i(t) - \delta k_i(t), \quad k_i(0) = k_{i0}.$$

تولید هر شرکت نمی‌تواند از ظرفیت تولید آن بیشتر شود: $0 \le q_i(t) \le k_i(t)$. هدف هر شرکت، بیشینه کردن سود تجمعی خود است:

$$J_i = -\int_0^T e^{-\rho t} \left[p(t) q_i(t) - \left(c_i q_i(t) + \frac{1}{2} d_i q_i(t)^2 \right) - \frac{1}{2} r_i I_i(t)^2 \right] dt.$$

۲.۲.۴ پارامترهای شبیه‌سازی

$$a = 100, \quad b = 0.1, \quad c_1 = c_2 = 20, \quad d_1 = d_2 = 10, \quad r_1 = r_2 = 0.5, \quad \delta = 0.1,$$
$$\rho = 0.05, \quad T = 10, \quad k_{10} = 50, \quad k_{20} = 50.$$

۳.۲.۴ نتایج و تحلیل

جدول ۱.۴: مقایسه عملکرد روش‌ها در مسئله بازار انرژی

معیار	PMP-SM	MADDPG	روش ترکیبی
سود کل شرکت ۱	۴۲۵۰	۴۱۸۰	۴۲۳۰
سود کل شرکت ۲	۴۱۸۰	۴۱۲۰	۴۲۰۰
زمان محاسبات (ثانیه)	۴۵	۳۸۰	۲۱۰
انحراف از تعادل نش (تقریبی)	۰۲.۰	۱۱.۰	۰۵.۰

روش ترکیبی با مقداردهی اولیه شبکه‌ها با نتایج روش پرتابی و اصل ماکسیمم پونتریاگین، هم دقت را بهبود بخشید و هم زمان یادگیری را نسبت به MADDPG کاهش داد.

۳.۴ مطالعه موردی دوم: مسئله تعقیب و گریز (بازی صفر–جمع)

۱.۳.۴ شرح مسئله

یک پهپاد تعقیب‌کننده و یک پهپاد گریزنده در صفحه حرکت می‌کنند. هدف تعقیب‌کننده رسیدن به فاصله کمتر از l از گریزنده (گرفتن آن) در کوتاه‌ترین زمان ممکن است. هدف گریزنده بیشینه کردن زمان گرفتن یا فرار به یک منطقه امن است [۱].

- متغیرهای حالت: $x = [x_p, y_p, x_e, y_e]^T$ (که در آن موقعیت پهپاد تعقیب‌کننده توسط متغیرهای x_p و y_p و موقعیت پهپاد گریزنده توسط متغیرهای x_e و y_e مشخص می‌شود.)
- دینامیک: $\dot{x}_p = V_p \cos\theta_p, \dot{y}_p = V_p \sin\theta_p, \dot{x}_e = V_e \cos\theta_e, \dot{y}_e = V_e \sin\theta_e$ (که در آن V_p سرعت نزدیک شدن و V_e سرعت فرار است.)
- متغیرهای کنترلی: زوایای حرکت θ_p و θ_e
- تابع هزینه برای تعقیب‌کننده: زمان رسیدن به هدف و کمینه کردن آن، و برای گریزنده به زمان فرار و بیشینه کردن آن
- پارامترها:$V_p = 1.2,\ V_e = 1.0,\ l = 0.5$

۲.۳.۴ نتایج

- **روش پرتابی و اصل ماکسیمم پونتریاگین:** استراتژی بهینه برای هر دو، خطای ناوبری تناسبی را نشان داد [۱]. **ناوبری تناسبی PN** یک قانون هدایت پرکاربرد در موشک‌ها و رهگیرهاست که بر پایه‌ی نرخ تغییر خط دید به سمت هدف عمل می‌کند. به‌جای اینکه موشک فقط به سمت هدف اشاره کند، فرمان جانبی را متناسب با سرعت چرخش خط دید تنظیم می‌کند. هرچه خط دید سریعتر بچرخد، موشک باید شتاب جانبی بیشتری بگیرد تا مسیر برخورد حفظ شود. ناوبری تناسبی را می‌توان به‌عنوان یک قانون هدایت بازخوردی ساده دید که در برخی مدل‌های رهگیری و بازی‌های تعقیب‌وگریز ظاهر می‌شود. در بسیاری از تحلیل‌های نظری، PN

یک تقریب خوب و شهودی برای راهبرد بهینه یا نزدیک به بهینه است. در یک بازی تعقیب و گریز ساده با سرعت ثابت، اگر تعقیب‌کننده از PN استفاده کند و هدف تلاش کند از او فرار کند (با حفظ فاصله خط دید)، ممکن است PN به یک راهبرد نزدیک به تعادل نش منجر شود. اما اگر هدف بتواند مانورهای پیچیده یا غیرقابل پیش‌بینی انجام دهد، PN به تنهایی ممکن است کافی نباشد و نیاز به استراتژی‌های پیشرفته‌تری (مانند استفاده از اطلاعات بیشتر درباره هدف، یا الگوریتم‌های پیشرفته‌تر بازی دیفرانسیلی) باشد.

به‌طور خلاصه، ناوبری تناسبی یک قانون هدایت عملی است که در بسیاری از سناریوهای تعقیب و گریز، به‌ویژه زمانی که ساده‌سازی‌ها معتبر هستند، می‌تواند نمایانگر یا تقریب خوبی از استراتژی بهینه در چارچوب نظریه بازی‌های دیفرانسیلی باشد.

- **روش MADDPG :** گریزنده یاد گرفت که در جهت مخالف تعقیب‌کننده حرکت کند و از دیواره‌ها برای مانور استفاده کند.
- **تحلیل:** در ۸۵% موارد، تعقیب‌کننده آموزش‌دیده با MADDPG موفق به گرفتن گریزنده شد، در حالی که استراتژی روش پرتابی و اصل ماکسیمم پونتریاگین در ۹۵% موارد موفق بود.

۴.۴ مطالعه موردی سوم: مدیریت ترافیک در تقاطع‌های هوشمند

۱.۴.۴ شرح مسئله

یک تقاطع بدون چراغ راهنمایی را در نظر بگیرید که ۴ خودروی خودران از چهار جهت مختلف به آن نزدیک می‌شوند. هر خودرو یک عامل است. هدف هر خودرو عبور از تقاطع در کمترین زمان ممکن و با کمترین شتاب‌گیری (مصرف سوخت) و بدون برخورد با دیگران است.

- متغیرهای حالت: موقعیت و سرعت هر خودرو (x_i, v_i)
- متغیرهای کنترلی: شتاب a_i (محدود بین a_{min} و a_{max})
- دینامیک: $\dot{x}_i = v_i, \dot{v}_i = a_i$
- تابع هزینه برای هر خودرو:

$$J_i = \int_0^{t_f} (w_1 + w_2 a_i(t)^2) dt + w_3 (x_i(t_f) - x_{target})^2.$$

۲.۴.۴ نتایج

پس از حدود ۱۰۰۰۰ اپیزود، خودروها یاد گرفتند که به‌صورت هماهنگ از تقاطع عبور کنند. یک الگوی خودـ سازمان‌دهی پدیدار شد که در آن خودروها سرعت خود را به‌گونه‌ای تنظیم می‌کردند که بدون توقف کامل و با حفظ فاصله ایمن، از تقاطع بگذرند. میانگین زمان عبور ۲۵% و مصرف سوخت ۱۵% نسبت به حالت با چراغ راهنمایی کلاسیک بهبود یافت.

۳.۴.۴ مثال در اقتصاد: رقابت در بازار منابع تجدیدشونده

فرض کنیم دو شرکت ماهی‌گیری در یک دریاچه مشترک ماهی‌گیری می‌کنند. جمعیت ماهی‌ها (حالت سیستم) توسط یک معادله دینامیکی رشد (مثلاً مدل لجستیک) و نرخ ماهی‌گیری هر شرکت (کنترل‌ها) تعیین می‌شوند. هر شرکت به دنبال بیشینه کردن سود خود در طول زمان است (که تابعی از میزان ماهی‌گیری و قیمت ماهی است).

- **مدل‌سازی دینامیکی:**

$$\dot{x} = rx(1 - x/K) - u_1 - u_2,$$

که در آن

x: جمعیت ماهی.

r: نرخ رشد ماهی.

K: ظرفیت حمل محیطی.

u_1, u_2: نرخ ماهی‌گیری شرکت ۱ و ۲.

- **توابع هزینه:** (هدف: بیشینه کردن سود)

$$J_i = \int_{t_0}^{t_f} \left(pu_i - cu_i^2\right)\, dt,$$

که در آن

p: قیمت فروش ماهی.

c: هزینه ماهی‌گیری (یک تابع درجه دوم از نرخ ماهی‌گیری برای نشان دادن بازدهی کاهشی).

- **تحلیل تعادل نش:** شرکت‌ها به‌طور غیرهمکارانه عمل می‌کنند. با استفاده از روش پرتابی و اصل ماکسیمم پونتریاگین تعمیم‌یافته، می‌توان نرخ‌های ماهی‌گیری بهینه $u_1^*(t)$ و $u_2^*(t)$ را پیدا کرد که یک تعادل نش را تشکیل می‌دهند. این تحلیل نشان می‌دهد که در یک تعادل نش غیرهمکارانه، معمولاً بیش‌ازحد بهره‌برداری از منبع مشترک صورت می‌گیرد، یعنی هر شرکت بیش از حد ماهی صید می‌کند و جمعیت ماهی‌ها به سطح پایداری نخواهد رسید. این "تراژدی منابع مشترک" نامیده می‌شود.

- **تحلیل تعادل همکارانه:** اگر شرکت‌ها همکاری کنند، می‌توانند نرخ ماهی‌گیری کل $U = u_1 + u_2$ را طوری تنظیم کنند که سود کل سیستم (جمع سود دو شرکت) بیشینه شود. این منجر به بهره‌برداری پایدارتر از منبع می‌شود، اما سپس باید مکانیزمی برای تقسیم سود حاصل از همکاری تعیین شود.

فصل ۵

بحث و نتیجه‌گیری

۱.۵ تحلیل تطبیقی روش‌ها

بر اساس نتایج فصل قبل، می‌توان مقایسه زیر را بین روش‌ها انجام داد:

جدول ۱.۵: مقایسه کلی روش‌های حل

ویژگی	PMP-SM	MADDPG	روش ترکیبی
نیاز به مدل دینامیکی	دارد (مدل‌مبنا)	ندارد (بدون‌مدل)	دارد (برای مقداردهی اولیه)
دقت در مسائل کوچک	بسیار بالا	متوسط	بالا
کارایی در ابعاد بالا	بسیار پایین	بالا	بالا
اعمال قیود	دشوار	نسبتاً آسان (با جریمه)	آسان‌تر
تضمین همگرایی	دارد	ندارد	نسبتاً خوب
تفسیرپذیری[۱]	بالا	پایین	متوسط
زمان محاسبات	کم	زیاد	متوسط

۲.۵ تحلیل حساسیت

روش پرتابی و اصل ماکسیمم پونتریاگین: این روش به مقداردهی اولیه $\lambda_i(t_0)$ بسیار حساس بود. در مسئله بازار انرژی، خطای ۱۰٪ در حدس اولیه باعث واگرایی روش می‌شد. روش پرتابی چندگانه این حساسیت را کاهش داد اما به‌طور کامل برطرف نکرد [۳].

روش MADDPG : این روش به تنظیم پارامترها (نرخ یادگیری و ساختار شبکه) بسیار حساس بود. نرخ یادگیری نامناسب باعث ناپایداری و واگرایی می‌شد. همچنین، به طراحی تابع پاداش حساس بود.

روش ترکیبی: این روش در برابر تغییر پارامترها مقاوم‌تر بود. مقداردهی اولیه با روش پرتابی و اصل ماکسیمم پونتریاگین باعث شد که یادگیری MADDPG پایدارتر و سریع‌تر باشد [۵، ۳].

[۱] تفسیرپذیری یعنی قابلیت تحلیل شفاف منطق تصمیم‌گیری الگوریتم، به‌گونه‌ای که علت هر دستور کنترلی بر اساس روابط ریاضی قابل بیان باشد.

کتاب‌نامه

[1] Isaacs, R. (1965). *Differential Games: A Mathematical Theory with Applications to Warfare, Pursuits, Control, and Optimization*. John Wiley & Sons.

[2] Bârsâ, T., & Olsder, G. J. (1999). *Dynamic Noncooperative Game Theory* (2nd ed.). SIAM.

[3] Bryson, A. E., Jr., & Ho, Y. C. (1975). *Applied Optimal Control: Optimization, Estimation, and Control*. Hemisphere Publishing.

[4] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.

[5] Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., & Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems (NIPS)*.

[6] Lasry, J. M., & Lions, P. L. (2007). Mean field games. *Japanese Journal of Mathematics*, 2(1), 229-260.

[7] Starr, A. W., & Ho, Y. C. (1969). Zero-sum differential games. *Journal of Optimization Theory and Applications*, 3(3), 184-206.

[8] Jørgensen, S., & Zaccour, G. (2004). *Differential Games in Marketing*. Springer.

[9] Kirk, D. E. (2004). *Optimal Control Theory: An Introduction*. Dover

[10] Bertsekas, D. P. (2017). *Dynamic Programming and Optimal Control*. Athena Scientific.

[11] Lewis, F. L., Vrabie, D., & Syrmos, V. L. (2012). *Optimal Control*. Wiley

واژه‌نامه فارسی به انگلیسی

استراتژی‌ها ..... Strategies
اطلاعات بازخوردی ..... Feedback Information
اطلاعات حلقه‌باز ..... Open-loop Information
اطلاعات کامل ..... Perfect Information
اطلاعات ناقص ..... Imperfect Information
اعتراف (در دوراهی زندانی) ..... Defect
افق متناهی ..... Finite Horizon
افق نامتناهی ..... Infinite Horizon
بازیکنان ..... Players
بازیگران (در یادگیری تقویتی) ..... Actors
بازی‌های خطی–درجه دوم ..... Linear-Quadratic Games
برنامه‌ریزی دینامیکی ..... Dynamic Programming
بردار حالت ..... State Vector
بردار ورودی کنترل ..... Control Input Vector
بیش‌ازحد بهره‌برداری ..... Overexploitation
بهینگی پارتو ..... Pareto Optimality
پایدار (حالت) ..... Steady-state
پیرو ..... Follower
تابع هزینه ..... Cost Function
تراژدی منابع مشترک ..... Tragedy of the Commons
تعادل نش ..... Nash Equilibrium
تعادل نش بازخوردی ..... Feedback Nash Equilibrium
تنظیم‌کننده (در بهینه‌سازی) ..... Regularizer
توابع پرداخت ..... Payoff Functions
رهبر ..... Leader
روش پرتابی چندگانه ..... Multiple Shooting
روش پرتابی ساده ..... Simple Shooting Method
سکوت (در دوراهی زندانی) ..... Cooperate
فرم گسترده ..... Extensive Form

فرم نرمال ..... Normal Form
قانون کنترل ..... Control Law
کنترل بهینه ..... Optimal Control
گره (در روش پرتابی) ..... Node
متغیرهای هم‌حالت ..... Costate variables
مسئله مقدار مرزی دو ـ نقطه‌ای ..... Two-Point Boundary Value Problem (TPBVP)
معادله دیفرانسیل ریکاتی جفت‌شده ..... Coupled Riccati Differential Equation
معادلات تغییراتی ..... Variational equations
معیار عملکرد ..... Performance Index
مرکزی‌سازی یادگیری، غیرمتمرکزسازی اجرا ..... Centralized Learning, Decentralized Execution
ناوبری تناسبی ..... Proportional Navigation
نرخ تنزیل ..... Discount rate
نفرین ابعاد ..... Curse of Dimensionality
نقادان (در یادگیری تقویتی) ..... Critics
نمایش فضای حالت ..... State Space Representation
هزینه جاری ..... Running cost
هزینه نهایی ..... Terminal cost
یادگیری تقویتی ..... Reinforcement Learning

Abstract

This work focuses on the optimization of dynamical systems, with particular emphasis on differential games. In today's complex world, many systems involve multiple decision-making agents, each pursuing distinct and sometimes conflicting objectives. Traditional optimal control methods, which are designed for a single decision-maker, are often insufficient for modeling and controlling such systems. Differential games therefore emerge as a powerful framework for analyzing strategic interactions among agents over time.

The world around us is full of dynamic systems and intricate interactions among decision-makers. Examples include producers and consumers in financial markets, competition among business firms, military engagements between opposing forces, and even urban traffic among drivers. In all of these settings, each agent—whether a human, a company, a robot, or a software system—seeks to optimize its own performance. However, this optimization is not independent, since the outcome for each agent is strongly influenced by the decisions of others.

Game theory, as a branch of applied mathematics, provides powerful tools for analyzing such strategic interactions. In static settings, games are typically represented by payoff matrices. Many real-world problems, however, are inherently dynamic, with the system state evolving over time. To model these processes, differential games, which combine game theory with optimal control, provide an appropriate mathematical framework.

Differential games were first introduced by Rufus Isaacs in the 1950s in the context of military applications, particularly pursuit-evasion problems. Since then, the field has developed significantly and has found applications in economics, management, political science, biology, engineering, and artificial intelligence.

Keywords: Dynamical Systems, Optimal Control, Differential Games, Nash Equilibrium, Dynamic Programming, Pontryagin's Maximum Principle, Multi-Agent Reinforcement Learning