بسم الله الرحمن الرحيم

دانشگاه صنعتی امیرکبیر

دانشکده مهندسی صنایع

پایان‌نامه کارشناسی ارشد

تعین پارامترهای تصمیم‌گیری سیستم‌های چندعامله به‌کار رفته در سیستم‌های تولیدی

نگارش

امیر رجبی‌نسب

استاد راهنما

دکتر سعید منصور

بهمن‌ماه ۱۳۸۷

دانشگاه صنعتی امیرکبیر
(پلی‌تکنیک تهران)

بسمه تعالی

فرم اطلاعات پایان‌نامه
کارشناسی- ارشد و دکترا

تاریخ:
شماره:

معاونت پژوهشی
فرم پروژه تحصیلات تکمیلی ۷

مشخصات دانشجو:

نام و نام خانوادگی: **امیر رجبی نسب** ● دانشجوی آزاد ○ بورسیه ○ معادل

شماره دانشجوئی: **۸۵۱۲۵۰۸۱** دانشکده: **مهندسی صنایع** رشته تحصیلی: **صنایع** گروه:

مشخصات استاد راهنما:

نام و نام خانوادگی: **دکتر سعید منصور** درجه و رتبه: **دانشیار**

نام و نام خانوادگی: درجه و رتبه:

مشخصات استاد مشاور:

نام و نام خانوادگی: **دکتر بهروز کریمی** درجه و رتبه: **دانشیار**

نام و نام خانوادگی: درجه و رتبه:

عنوان پایان‌نامه به فارسی : تعین پارامترهای تصمیم‌گیری سیستم‌های چندهوشوار به‌کار رفته در سیستم‌های تولیدی

عنوان پایان‌نامه به انگلیسی: Multi-agent decision parameters tuning applied to manufacturing systems

نوع پروژه: کارشناسی○ ارشد● دکترا ○ سال تحصیلی: **۸۸-۸۷**

کاربردی ● بنیادی○ توسعه‌ای ● نظری○

تاریخ شروع : **۱۳۸۶/۱۱/۱** تاریخ خاتمه : **۱۳۸۷/۱۱/۱** تعداد واحد : **۶** سازمان تأمین کننده اعتبار :

واژه‌های کلیدی به فارسی: سیستم‌های چندعامله، سیستم‌های تولیدی، هوشمندی جامعه مورچگان، زمانبندی پویا.

واژه‌های کلیدی به انگلیسی : Multi-agent system, Manufacturing system, Ant colony intelligence, Dynamic scheduling.

مشخصات ظاهری	تعداد صفحات ۹۷	تصویر● جدول● نمودار ● نقشه ○ واژه‌نامه ○		تعداد مراجع ۷۴	تعداد صفحات ضمائم
زبان متن	فارسی ●	انگلیسی ○	چکیده	فارسی ●	انگلیسی ●
یادداشت					

نظرها و پیشنهادها به منظور بهبود فعالیت‌های پژوهشی دانشگاه

استاد:

دانشجو:

امضاء استاد راهنما: **تاریخ:**

تقدیم به چشمان مهربان پدرم،

روح پر صلابت مادرم

و

وجود بی آلایش همسرم

که بزرگترین دارایی های من هستند.

چکیده

سیستم‌های چندعامله ابزاری برای گسترش هوش مصنوعی توزیع شده می‌باشد و در عرصه‌های مختلف علمی مورد استفاده و بهره‌برداری قرار گرفته است. یکی از حوزه‌های کاربردی سیستم‌های چندعامله سیستم‌های تولیدی است که با توجه به پیچیدگی این سیستم‌ها وجود ابزار قدرتمندی برای تصمیم‌گیری و حل مسئله همچون سیستم‌های چندعامله به وضوح احساس می‌شود. یکی از مسائل بسیار پیچیده مطرح در سیستم‌های تولیدی، زمانبندی بلادرنگ سیستم‌های تولیدی پویا است که در این پژوهش با استفاده از راهکار چندعامله به حل این مسئله پرداخته شده است.

برای دستیابی به این هدف در این پژوهش ابتدا مفاهیم ضروری سیستم‌های چندعامله و حل مسائل با استفاده از آن در فصل اول تشریح شده و سپس برای روشن شدن اهمیت کاربرد سیستم‌های چندعامله و نیز پژوهش در این حوزه دانشی در فصل دوم گستره وسیعی از کاربردهای سیستم‌های چندعامله به طور مختصر مورد بررسی قرار گرفته است. در فصل سوم مروری بر راهکارهای سنتی حل مسائل زمانبندی پویای تولید و مقایسه آن با راهکار چندعامله صورت پذیرفته که به موجب آن برتری‌های سیستم چندعامله نسبت به راهکارهای پیشین نشان‌داده شده است. در فصل چهارم مدل چندعامله توسعه داده شده در این پژوهش تشریح می‌شود این مدل که مبتنی بر هوشمندی جامعه مورچگان می‌باشد نسبت به سایر راهکارها و مدل‌ها دارای چندین برتری است اول اینکه فرض‌های ساده‌ساز قبلی کنار گذاشته شده و شرایط مسئله به واقعیت نزدیکتر می‌باشد دوم اینکه توابع هدف جامع‌تر و کاراتری مورد استفاده قرارگرفته که بقاء سازمان تولیدی را در بازار رقابتی امروز تضمین می‌نماید سوم اینکه این مدل توانایی پاسخگویی بلادرنگ و پویا به تغییرات و اختلالت محیطی را دارا می‌باشد. در فصل پنجم به منظور بررسی مدل پیشنهادی و اندازه‌گیری کارایی و توانایی آن در ابتدا یک ابزار شبیه‌سازی چندعامله سیستم تولیدی پویا توسعه داده شده که از آن به منظور شبیه‌سازی مدل پیشنهادی و مقایسه کارایی آن با مدل‌های قبلی مورد استفاده قرار گرفته است.این ابزار نیز در نوع خود منحصر بفرد بوده و قادر به شبیه‌سازی کلیه سیستم‌های چندعامله تولیدی پویا می‌باشد. نتایج‌عددی بدست آمده حاکی از برتری مدل پیشنهادی در این پژوهش نسبت به مدل‌های قبلی می‌باشد.

در انتها نیز پیشنهاداتی برای تحقیقات آینده ارائه شده است که از آن جمله می‌توان به موارد زیر اشاره نمود: کنار گذاشتن سایر فرضیات ساده‌ساز از قبیل عدم محدودیت حمل و نقل، تنظیم پویای پارامترهای مدل چندعامله به‌گونه‌ای که بحرانی بودن موعدهای تحویل در تصمیم‌گیری‌ها بیشتر و بهتر دخیل شود و نیز توسعه ابزار شبیه‌سازی به‌گونه‌ای که به یک محیط گرافیکی شبیه‌سازی سیستم‌های چندعامله تولید پویا تبدیل شود.

فهرست مطالب

فصل ۱: مروری بر مفاهیم سیستم‌های چندعامله ۱

۱-۱- مشخصه‌های سیستم‌های چندعامله ۲

۱-۱-۱- طراحی عامل ۲

۱-۱-۲- محیط ۲

۱-۱-۳- ادراک ۳

۱-۱-۴- کنترل ۳

۱-۱-۵- دانش ۴

۱-۱-۶- ارتباطات ۴

۱-۲- عامل‌های منطقی ۴

۱-۲-۱- تعریف عامل ۴

۱-۲-۲- عامل، یک تصمیم‌گیرنده منطقی ۵

۱-۲-۳- جهان مشاهده‌پذیر وخاصیت مارکوف ۶

الف- مشاهده‌پذیری ۷

ب- ویژگی مارکوفی ۸

۱-۲-۴- گذار احتمالی و مطلوبیت ۹

الف- از هدف تا مطلوبیت ۹

ب- تصمیم‌گیری در جهان احتمالی ۱۰

۱-۳- بازیهای استراتژیک ۱۱

۱-۳-۱- تئوری بازی‌ها ۱۱

۱-۳-۲- بازیهای استراتژیک ۱۲

۱-۳-۳- حذف تکراری عمل‌های مغلوب ۱۴

۱-۳-۴- نقطه تعادل نش ۱۵

۱-۴- مشارکت ۱۶

۱-۴-۱- بازیهای مشارکتی ۱۷

۱-۴-۲- کنوانسیون‌های اجتماعی ۱۷

۱-۴-۳- نقش‌ها ۱۸

۱-۴-۴- گرافهای مشارکت ۱۸

الف- مشارکت با استفاده از رد و بدل پیام ۱۹
۱-۵- مشاهده پذیری نسبی ۲۰
۱-۶- طراحی مکانیسم ۲۰
۱-۶-۱- عامل‌های خود غرض ۲۰
۱-۶-۲- مسئله طراحی مکانیسم ۲۱
۱-۷- شالوده‌ای برای عامل‌های فیزیکی هوشمند ۲۲
۱-۷-۱- اهداف و تاریخچه FIPA ۲۲
۱-۷-۲- مفاهیم اساسی FIPA ۲۳
الف- ارتباط عامل‌ها ۲۳
ب- مدیریت عامل‌ها ۲۴

فصل ۲: مروری بر ادبیات کاربرد سیستم‌های چندعامله ۲۶

۲-۱- کاربر سیستم‌های چند عامل در کشاورزی ۲۶
۲-۲- کاربرد سیستم‌های چند عامل در رباتیک ۲۷
۲-۳- کاربرد سیستم‌های چندعامله در مسائل فضایی ۲۸
۲-۴- کاربرد سیستم‌های چندعامله در مدیریت دانش ۲۹
۲-۵- کاربرد سیستم‌های چندعامله در مسائل اجتماعی ۳۰
۲-۶- کاربرد سیستم‌های چندعامله در ساخت و ایجاد خودآموزها ۳۱
۲-۷- کاربرد سیستم‌های چندعامله در سیستم‌های تولیدی ۳۳
۲-۷-۱- الزامات نسل بعدی سیستم‌های تولیدی ۳۴
۲-۷-۲- راهکارهای مبتنی بر عامل در تولید هوشمند ۳۴
۲-۷-۳- سیستم‌های تولیدی هولونیک ۳۶
۲-۷-۴- مروری بر ادبیات سیستم‌های تولیدی عامل‌مدار ۳۸
الف- یکپارچگی سازمانی ۳۸
ب- مشارکت سازمانی ۴۰
ج- زمانبندی و برنامه‌ریزی فرآیندهای تولیدی ۴۱

فصل ۳: کاربرد سیستم چندعامله در زمانبندی پویای تولید ۴۴

۳-۱- پیچیدگی مسائل زمانبندی تولید ۴۵

۳-۲- زمانبندی پویای تولید ۴۷
۳-۲-۱- راهکارهای سنتی زمانبندی پویای تولید ۴۸
الف- قوانین توزیع ۴۸
ب- راهکارهای ابتکاری ۴۹
ج- راهکارهای فراابتکاری ۴۹
د- راهکارهای هوش مصنوعی ۵۰
۳-۲-۲- راهکارهای عامل‌مدار زمانبندی پویای تولید ۵۱
چرا سیستم چندعامله؟ ۵۳

فصل ۴: توسعه مدل چند عامل زمانبندی پویای تولید ۵۶

۴-۱- توصیف مدل ۵۶
۴-۱-۱- مدل عمومی کف کارگاه ۵۷
۴-۱-۲- اختلالات احتمالی سیستم تولیدی ۵۸
۴-۱-۳- معیارهای کارایی ۵۸
۴-۱-۴- توسعه‌های داده‌شده بر مدل ۵۹
۴-۲- سیستم چندعامله مدل عمومی سیستم تولیدی ۶۱
۴-۲-۱- وضعیت عامل‌ها ۶۲
۴-۲-۲- مشارکت عامل‌ها ۶۳
۴-۳- همکاری با استفاده از هوشمندی جامعه مورچگان ۶۵
۴-۳-۱- هوشمندی جامعه مورچگان برای عامل ماشین ۶۶
۴-۳-۲- هوشمندی جامعه مورچگان برای عامل کار ۶۷

فصل ۵: شبیه‌سازی چندعامله سیستم تولیدی ۶۹

۵-۱- مروری بر شبیه‌سازی چندعامله ۶۹
۵-۲- بررسی بسترهای شبیه‌سازی عامل‌مدار ۷۲
۵-۳- معرفی نرم‌افزار و بستر عامل‌مدار JADE ۷۴
۵-۴- معرفی PLASMA و توسعه‌های داده شده بر آن در این پژوهش ۷۷
۵-۴-۱- معماری PlaSMA ۷۸
۵-۴-۲- توسعه‌های داده شده بر PlaSMA ۸۰

۵-۵- پیاده‌سازی مدل زمانبندی پویای تولید و بررسی نتایج عددی ۸۸
۵-۵-۱- تشریح سناریوهای آزمایشی ۸۹
۵-۵-۲- تحلیل نتایج ۹۰
مراجع ۹۴

فصل اول

مروری بر مفاهیم سیستم‌های چندعامله

راهکارهای جدید هوش مصنوعی(AI)[1] بر پایه مفهوم عامل منطقی[2] بنا شده‌اند. یک عامل موجودیتی است که محیط اطراف خود را با استفاده از یک سری حسگر درک کرده و با استفاده از یکسری عملگر به عمل می‌پردازد[۱]. عاملی که همواره سعی دارد یک مقیاس کارایی را بهینه نماید یک عامل منطقی خوانده می‌شود. اینچنین تعریفی از عامل‌های منطقی کاملاً کلی بوده و لذا انسان‌ها(دارای دو چشم به عنوان حسگر دو دست به عنوان عملگر)، رباتها(دارای دوربین‌ها به عنوان حسگر و چند چرخ به عنوان عملگر) و نرم افزارها(دارای واسط گرافیکی کاربر (GUI) به عنوان حسگر ونیز عملگر) را نیز شامل می‌شود. با داشتن این دیدگاه می‌توان AI را علم مطالعه اصول و طراحی عامل‌های منطقی دانست.

با این وجود عامل‌ها به‌ندرت یک سیستم یکتا و مستقل می‌باشند و در اکثر مواقع به طرق مختلف با عامل‌های دیگر رابطه متقابل دارند به این سیستم‌ها، سیستم‌های چندعامله(MAS)[3] و به زیرحوزه‌ای از هوش‌مصنوعی که بر اصول و طراحی سیستم‌های چندعامله تمرکز دارد هوش مصنوعی توزیع شده(DAI)[4] گفته می‌شود.

---

[1] Artificial Intelligence

[2] Rational Agent

[3] Multi Agent System

[4] Distributed Artificial Intelligence

در این فصل ابتدا به بررسی مشخصات سیستم‌های چندعامله پرداخته و سپس مهمترین مباحث مربوط به آن که در فصول بعدی مورد استفاده قرار گرفته است به شکل مختصر تشریح می‌شود. در انتها نیز استاندارد FIPA به عنوان استاندارد پایه‌ای بسیاری از بسترهای نرم‌افزاری چندعامله(از جمله بستر مورد استفاده در این پژوهش) مورد بررسی قرار می‌گیرد.

۱-۱- مشخصه‌های سیستم‌های چندعامله

می‌توان مشخصه‌های زیر را به عنوان ابعادی که یک سیستم چندعامله را از یک سیستم تک عامل متمایز می‌کند بیان نمود[۲]:

۱-۱-۱- طراحی عامل

در بسیاری موارد عامل‌های مختلف تشکیل دهنده سیستم‌های چندعامله به روش‌های متفاوتی طراحی شده‌اند؛ طراحی متفاوت می‌تواند شامل تفاوت در طراحی سخت‌افزاری و یا نرم‌افزاری باشد، مثلاً ربات‌های فوتبالیست می‌توانند با استفاده از سخت‌افزارهای متفاوتی ایجاد شده باشند و یا عامل‌های نرم‌افزاری کدهای متفاوتی را به کار ببرند. عامل‌هایی که بر پایه سخت‌افزارهای متفاوت و یا نرم‌افزارهای متفاوتی ایجاد شده‌اند عامل‌های ناهمگن[1] و عامل‌هایی که به روش واحدی طراحی شده و دارای قابلیت‌های یکسانی می‌باشند عامل‌های همگن[2] نامیده می‌شوند. ناهمگن بودن عامل‌ها می‌تواند تمامی جنبه‌های عملکردی یک عامل را از ادراک عامل گرفته تا تصمیم‌گیری آن تحت تاثیر قرار دهد.

۱-۱-۲- محیط

عامل می‌بایست با محیط اطراف خود در تعامل باشد این محیط می‌تواند ایستا و یا پویا باشد. اغلب تکنیک‌های هوش مصنوعی برای مدل تک‌عامل بر پایه محیط‌های ایستا توسعه یافته‌اند چرا که فرض ایستا بودن محیط حل مسئله را آسانتر کرده و امکان استفاده از محاسبات ریاضی ساده‌تری را فراهم می‌نماید. در یک سیستم چندعامله وجود یک عامل در محیط، شرایط محیط را از دید عامل‌های دیگر پویا می‌کند چرا که هر عملی که عامل انجام می‌دهد محیط را از دید عامل‌های دیگر

[1] Heterogeneous

[2] Homogeneous

تغییر می‌دهد. این امر می‌تواند مسئله‌ساز باشد به طور مثال در مبحث یادگیری هم‌زمان[1] عامل‌ها، هیچ حالت پایداری در محیط قابل مشاهده نیست.

۱-۱-۳- ادراک

اطلاعاتی را که یک عامل توسط حسگرهای خود در یک سیستم چندعامله جمع‌آوری می‌کند معمولاً می‌تواند از جنبه‌های زیر توزیع شده باشد:

- مکانی: عامل ممکن است داده‌ها را از نقاط متفاوتی جمع‌آوری کرده باشد.
- زمانی: داده‌ها می‌توانند در زمان متفاوتی جمع‌آوری شوند.
- معنایی: داده‌ها می‌بایست به روش متفاوتی تفسیر شوند.

این واقعیت که عامل‌ها می‌توانند موارد مختلف و متفاوتی را از محیط اطراف خود درک نمایند جهان را برای عامل‌ها تا حدودی قابل درک می‌نماید این واقعیت تأثیرات متفاوتی را روی تصمیم‌گیری عامل‌ها دارد. مسئله دیگر بحث امتزاج حسگرهاست به این معنی که عامل‌ها چگونه ادراک خود را از محیط اطراف به طور بهینه با هم ترکیب نمایند تا دانش گردآوری شده از موقعیت حاضر افزایش یابد. در بخش ۱-۵ برخی از این مسائل با جزئیات بیشتری مورد باز بینی قرار خواهد گرفت.

۱-۱-۴- کنترل

بر خلاف سیستم‌های تک عامل کنترل در سیستم‌های غیرمتمرکز به صورت توزیع شده می‌باشد. کنترل توزیع‌شده در مقابل کنترل متمرکز مطرح می‌شود و دارای مزایای متعددی است که عمده آن خطاپذیری و پایداری آن می‌باشد. اما با این وجود تمامی پروتکل‌های مربوط به سیستم‌های چندعامله نمی‌توانند به راحتی توزیع شده باشند. این مباحث به طور دقیق‌تر در بخش ۱-۶ مورد بررسی قرار می‌گیرد. در سیستم‌های چندعامله مشارکتی و یا تیمی عامل‌ها علائق و اهداف یکسانی را به اشتراک می‌گذارند و تصمیم‌گیری توزیع شده در آنها محاسبات غیر همزمان اما پر سرعتی را پیشنهاد می‌کند جنبه دیگر این‌گونه سیستم‌ها این است که این سیستم‌ها نیاز به مکانیسم مشارکتی دارند. در بخش ۱-۴ مبحث مشارکت عامل‌ها مورد بررسی قرار خواهد گرفت.

---

[1] Concurrently Learning

۱-۱-۵- دانش

در سیستم تک عامل فرض می‌شود عامل تنها عمل‌های مربوط به خود را می‌داند و نیازی نیست که بداند جهان اطراف چگونه تحت تأثیر این عمل‌ها تغییر می‌کند. اما در سیستم‌های چندعامله این دانش عامل‌ها توسعه یافته تر است و به طور مثال در عامل‌های مشارکتی هر عامل می‌تواند مجموعه عمل‌های عامل دیگر را بداند همچنین عامل‌ها می‌توانند از طریق ارتباطات از درک فعلی یکدیگر با خبر شوند و یا اینکه تصمیمات یکدیگر را تحت تأثیر قرار دهند در بخش ۱-۵ در رابطه با دانش مشترک عامل‌ها بیشتر توضیح داده خواهد شد.

۱-۱-۶- ارتباطات

تعامل میان عامل‌ها اغلب به واسطه ارتباطات میان آن‌ها ایجاد می‌شود. ارتباطات می‌تواند در موارد متعددی مورد استفاده قرار گیرد به طور مثال در ایجاد هماهنگی در بین عامل‌های مشارکتی و یا چانه‌زنی میان عامل‌های رقابتی. این مبحث مسائل جدیدی از قبیل نوع پروتکلی که در تبادل اطلاعات مورد استفاده قرار می‌گیرد و زبان ارتباطی عامل‌ها را به وجود می‌آورد. در ادامه این فصل موارد متعددی از این مسائل مورد بررسی قرار خواهد گرفت.

۱-۲- عامل‌های منطقی

در این بخش تلاش شده که بعد از معرفی عامل منطقی برخی از مشخصه‌های محیط اطراف یک عامل از قبیل مشاهده پذیری[1] و خاصیت مارکوفی[2] مورد بازبینی قرار گیرد، همچنین بررسی شده که برای بهینه عمل کردن یک عامل در محیطی با عدم قطعیت نیاز به چه مواردی است.

۱-۲-۱- تعریف عامل

بنا به تعریف ارائه شده در مرجع [۱] عامل موجودیتی است که محیط اطراف خود را با استفاده از یک‌سری حسگر درک کرده و با استفاده از یک‌سری عملگر به عمل می‌پردازد. واژه خودمختار[3] برای عامل‌هایی به کار گرفته می‌شود که تصمیم‌گیری آن‌ها علاوه بر دانش اولیه‌ای که در زمان طراحی به‌کار گرفته شده بر پایه ادراکشان نیز می‌باشد. در این بخش مسئله تصمیم‌گیری بهینه یک

[1] Observability

[2] Markov Property

[3] Autonomous

عامل مورد بررسی قرار می‌گیرد این مسئله عبارت از چگونگی انتخاب یک عمل از میان عمل‌های ممکن توسط عامل در هر مرحله زمانی با استفاده از دانش موجود عامل از محیط اطراف خود می‌باشد. عامل منطقی عاملی است که همواره عملی را انتخاب می‌کند که یک مقیاس کارایی رابهینه نماید. مقیاس کارایی عموماً توسط کاربر(و در زمان طراحی عامل) تعریف شده و تعیین‌کننده این موضوع است که کاربر چه انتظاری را از فعالیت عامل دارد. به طور مثال یک ربات فوتبالیست می‌بایست به گونه‌ای عمل کند که شانس امتیازگیری تیمش حداکثر شود و یا یک عامل نرم‌افزاری در یک مزایده الکترونیکی می‌بایست به گونه‌ای عمل کند که هزینه‌ها حداقل شود.

در ادامه این بخش به صورت عمده روی عامل‌های محاسباتی تمرکز می‌شود، عامل‌های محاسباتی عامل‌هایی هستند که به صورت مشخص به منظور حل یک مسئله محاسباتی طراحی شده ولذا بر روی یکسری نرم افرازها و ابزارهای محاسباتی پیاده می‌شوند.

۱-۲-۲- عامل، یک تصمیم‌گیرنده منطقی

مسئله تصمیم‌گیری عامل یکی از مباحث کنترل بهینه[1] می‌باشد. در این بخش به منظور ساده‌تر شدن بحث مجموعه‌ای از گام‌های گسسته زمانی $t = 0,1,2, \dots$ در نظر گرفته‌شده که عامل می‌بایست در هریک از این گام‌ها عمل a_t را از مجموعه متناهی عمل‌های A انتخاب نماید. به طور شهودی می‌توان دریافت که به منظور منطقی عمل کردن عامل در زمان t عامل می‌بایست زمان گذشته و نیز زمان آینده را در انتخاب عمل فعلی خود مد نظر قرار دهد. زمان گذشته معرف اطلاعاتی که عامل مشاهده نموده و عمل‌هایی که تا زمان t اتخاذ نموده است، می‌باشد. زمان آینده نیز معرف اطلاعاتی که عامل انتظار دارد مشاهده نماید و عمل‌هایی که انتظار دارد بعد از زمان t اتخاذ نماید می‌باشد.

اگر θ_τ نمایشگر مشاهدات عامل در زمان τ باشد پاراگراف قبلی به این نکته اشاره می‌کند که به منظور انتخاب بهینه یک عمل در زمان t می‌بایست تاریخچه مشاهدات θ_τ و عمل‌های a_τ را برای تمامی زمان‌های $\tau \preceq t$ مد نظر قرار گیرد. تابع(۱-۱)که در واقع تاریخچه مشاهدات-عمل‌های یک عامل را تا زمان t به عمل بهینه a_t نگاشت می‌کند تابع سیاست[2] عامل خوانده می‌شود:

$$\pi(\theta_0, a_0, \dots, \theta_\tau) = a_t \qquad (۱-۱)$$

[1] Optimal Control

[2] Policy Function

لذا با تعریف و استفاده از تابع سیاست بخشی از مسئله تصمیم‌گیری عامل‌ها که همان استفاده از اطلاعات گذشته است حل می‌شود. اما تعریف و پیاده‌سازی این تابع در عمل می‌تواند مشکل‌ساز باشد چرا که تاریخچه کامل می‌تواند حاوی تعداد زیاد وحتی بی‌نهایتی از زوج های مشاهده- عمل بوده که در موارد کاربردی مختلف متغیر است. ذخیره تمامی تاریخچه می‌تواند نیازمند حجم زیادی از حافظه باشد و از طرفی دیگر هزینه محاسبات را به شدت زیاد کند. این واقعیت نیاز به یک تابع سیاست ساده‌تررا مطرح می‌کند. یکی از موارد امکان‌پذیر صرف نظر کردن از تمامی تاریخچه مشاهدات عامل به‌جز مشاهده حال حاضر آن یعنی θ_t می‌باشد. در این حالت تابع سیاست به شکل زیر تغیر می‌کند:

$$\pi(\theta_t) = a_t \qquad (۲-۱)$$

عاملی که به سادگی مشاهده فعلی خود را به عمل تبدیل می‌کند و لذا گذشته را به طور کامل فراموش می‌نماید عامل انفعالی[۱] نامیده می‌شود و تابع سیاست (۱-۲) تابع سیاست انفعالی[۲] ویا بدون حافظه[۳] خوانده می‌شود. سوالی که به طور طبیعی مطرح می‌شود این است که یک عامل انفعالی چگونه وتا چه حدی می‌تواند موفق باشد. همان طورکه در ادامه خواهید دید عامل‌های انفعالی در گروهی از محیط‌ها به خوبی عمل می‌کنند.

۱-۲-۳- جهان مشاهده‌پذیر وخاصیت مارکوف

با توجه به مطالب آورده شده در بخش قبلی واژه‌های "عامل" و "محیط" واژه‌هایی هستند که در کنار هم می‌آیند و لذا نمی‌توان یکی را بدون توجه به دیگری تعریف نمود. از این به بعد فرض می‌شود جهانی شامل یک یا چند عامل وجود دارد این عامل‌ها دارای درک، فکر و عمل می‌باشند. اطلاعات جمع آوری شده از جهان در هر گام زمانی t که مرتبط با فعالیت و هدف در دست نیز می‌باشد حالتی[۴] از جهان است که با s_t نمایش داده می‌شود، مجموعه تمامی حالات جهان با S نمایش داده خواهد شد. به طور مثال در بازی رباتهای فوتبالیست حالت جهان می‌تواند با استفاده از چیدمان زمین بازی، موقعیت و سرعت بازیکنان و توپ مشخص می‌شود علاوه بر این پارامترهای دیگری که در تصمیم‌گیری تأثیر گذار است از قبیل زمان باقی‌مانده تا پایان بازی و غیره و یا در یک

[1] Reflex Agent

[2] Reactive Policy Function

[3] Memory less

[4] State

سیستم تولیدی وضعیت فعلی ماشین‌آلات، سفارشات رسیده و زمان باقیمانده تا تحویل آنها نیز در تعین این حالات موثر می‌باشد. با توجه به طبیعت مسئله، جهان می‌تواند پیوسته و یا گسسته باشد. یک جهان گسسته می‌تواند با استفاده از تعداد متنهای از حالات تعریف شود مانند پیکربندی‌های مختلف در تخته بازی شطرنج. جهان پیوسته دارای تعداد نامحدودی حالت می باشد به طور مثال نقاطی که یک ربات در یک سطح و به طور آزادانه می‌تواند طی کند($S = \mathbb{R}^2$)

الف- مشاهده‌پذیری

یکی از مشخصه‌های اساسی جهان از دیدگاه یک عامل میزان درک عامل از آن می‌باشد. گفته می‌شود جهان(کاملاً) مشاهده‌پذیر است اگر مشاهده حال حاضر عامل(θ_t) حالت فعلی جهان s_t را نتیجه دهد به عبارتی $s_t = \theta_t$. از طرفی در جهان نسبتاً مشاهده پذیر مشاهده حال حاضر عامل θ_t تنها بخشی از اطلاعات مربوط به حالت S_t جهان را به صوررت معین[1] ویا احتمالی[2] و با استفاده از مدل مشاهده(به طور مثال ($P(s_t|\theta_t)$) در اختیار قرار می‌دهد لذا عبارت $P(s_t|\theta_t)$ بیان کننده این است که مشاهده حال حاضر θ_t نمی‌تواند به طور کامل حالت s_t را تعیین نماید بلکه برای هر مشاهده θ_t به یک احتمال مشخص حالت جهان، s_t می باشد(به طوریکه $\sum_{s_t \in S} P(s_t|\theta_t) = 1 , 0 \leq P(s_t|\theta_t) \leq 1$). به عبارتی در اینجا s_t یک متغییر تصادفی است که می تواند تمامی مقادیر مجموعه S را اتخاذ نماید. با توجه به وابستگی احتمالی که بین s_t و θ_t وجود دارد مدل مشاهده می‌تواند به شکل $P(\theta_t|s_t)$ نیز بیان شود که در آن صورت می‌توان $P(s_t|\theta_t)$ را با استفاده از $P(s_t)$ و قانون بیز به صورت زیر محاسبه نمود:

$$P(s_t|\theta_t) = \frac{P(\theta_t|s_t)P(s_t)}{P(\theta_t)} \qquad (۳-۱)$$

نسبتاً مشاهده‌پذیر بودن را می‌توان به دو فاکتور منسوب نمود: اول وجود خطا و اختلال در حسگرهای عامل، به طور مثال وجود نقص در حسگرهای یک عاملی می‌تواند موجب این شود که یک حالت یکتا در زمان‌های مختلف مشاهدات مختلفی را موجب شود. دوم ویژگی‌های ذاتی محیط که از آن با عنوان بدنمایی ادراکی[3] یاد می‌شود، به بیان دیگر در برخی موارد دو حالت مختلف می‌تواند در نظر عامل یکسان به نظر برسد به طور مثال دو در مشابه در دو انتهای راهرو می‌تواند از

[1] Deterministic

[2] Stochastic

[3] Perceptual Aliasing

دید انسان و عامل یکسان به نظر برسد. حل مسائل در یک محیط نسبتاً مشاهده‌پذیر بسیار سخت‌تر از حل مسائل در یک محیط کاملاً مشاهده‌پذیر می‌باشد و الگوریتم‌های تصمیم‌گیری بهینه در محیط نسبتاً مشاهده‌پذیر اغلب بسیار دشوار و پیچیده می‌باشند. در ادامه و در بخش ۱-۵ خواهید دید که مبحث نسبتاً مشاهده‌پذیری نه‌تنها آنچه را که عامل‌ها از جهان می‌دانند را تحت تأثیر قرار می‌دهد بلکه آنچه را که عامل‌ها از یکدیگر می‌دانند را متأثر می‌نماید.

ب- ویژگی مارکوفی[1]

فرض کنید یک عامل انفعالی با تابع سیاست انفعالی $\pi(\theta_t) = a_t$ در یک جهان کاملاً مشاهده‌پذیر وجود دارد. فرض کاملاً مشاهده‌پذیر بودن این نکته را مشخص می‌کند $s_t = \theta_t$ و لذا داریم که:

$$\pi(s_t) = a_t \qquad (۱-۴)$$

لذا در یک محیط کاملاً مشاهده‌پذیر تابع سیاست یک عامل انفعالی عبارت است از نگاشت حالات محیط به عمل. این دستاورد از این واقعیت نشأت می‌گیرد که در بسیاری از مسائل حالت فعلی جهان خود معرف تاریخچه کامل قبل از این زمان می‌باشد. جهانی که در آن حالت فعلی خود خلاصه‌ای از تمامی حالات و اطلاعات موجود قبل از این حالت می‌باشد جهانی با خاصیت مارکوفی خوانده می‌شود. نتیجه‌ای که از این بحث می‌توان گرفت این است که در یک جهان مارکوفی، عامل می‌تواند به راحتی از یک تابع سیاست بدون حافظه (۱-۴) بجای یک تابع سیاست با حافظه زیاد (۱-۲) استفاده کند.

در ادامه این پژوهش و در فصل چهارم خواهید دید که محیط تولیدی را می‌توان به عنوان یک جهان مارکوفی درنظر گرفت و لذا از وضعیت فعلی و حتی از اطلاعات قبلی در تصمیم‌گیری بهره گرفت. درهر صورت آنچه که در بالا آورده شده چگونگی وابستگی تابع سیاست یک عامل به تجارب قبلی و خواص ویژه محیط می‌باشد. اما همان طور که در ابتدای بحث گفته شد به منظوری تصمیم‌گیری بهینه علاوه بر گذشته، آینده را نیز باید در نظر گرفت. این مبحثی است که در قسمت‌های بعدی مورد بررسی قرار می‌گیرد.

[1] Markov Property

۱-۲-۴- گذار احتمالی و مطلوبیت

همان طور که در بخش‌های قبلی گفته شد هر عامل در گام زمانی t عمل a_t را اختیار می‌کند که به موجب آن جهان دچار تغییر می‌شود. یک مدل گذار[1](مدل جهان) تعیین می‌کند چگونه جهان در اثر اجرای یک عمل تغییر می‌نماید. اگر حالت فعلی جهان s_t باشد و عامل عمل a_t را اتخاذ نماید دو مورد زیر را می‌توان در نظر گرفت:

- در یک جهان غیر احتمالی و قطعی، مدل گذار هر جفت حالت–عمل مانند (s_t, θ_t) را به یک حالت جدید s_{t+1} می‌نگارد. به طور مثال در بازی شطرنج هر حرکت، پیکربندی صفحه شطرنج را به یک روش قطعی تغییر می‌دهد.
- در یک جهان احتمالی، مدل گذار هر جفت حالت–عمل مانند (s_t, θ_t) را به یک توزیع احتمالی $P(s_{t+1}|s_t, a_t)$ می‌نگارد. به طور مثال در موارد با جهان نسبتاً مشاهده‌پذیر، s_{t+1} یک متغیر تصادفی است که می‌تواند تمامی مقادیر S را با احتمال $P(s_{t+1}|s_t, a_t)$ اتخاذ کند. بسیاری از کاربردهای جهان واقعی دارای مدل گذار احتمالی می‌باشد. به طور مثال حرکت ربات‌ها به دلیل لیز خوردن چرخ‌ها و یا عوامل دیگر غیر دقیق می‌باشد و یا زمان پردازش قطعات در سیستم‌های تولیدی به دلیل تغییر در کارایی ماشین می‌تواند غیر قطعی و احتمالی باشد.

همان طور که در بخش قبل گفته شد مشاهده‌پذیری نسبی می‌تواند به عدم قطعیت ادراک یک عامل مرتبط باشد. در این بخش مورد دیگری را که عدم قطعیت در آن نقش بازی می‌کند را مشاهده نمودید و آن عبارت بود از عدم قطعیت در تغییر جهان به واسطه اجرای یک عمل خاص توسط عامل. به عبارتی در جهان احتمالی اثر یک عمل خاص عامل از ابتدا مشخص نیست. لذا عدم قطعیت در مدل گذار نیز می‌تواند تصمیم‌گیری بهینه را دشوارتر کند.

الف- از هدف تا مطلوبیت

در هوش مصنوعی یک هدف[2] برای یک فعالیت عبارت است از یک حالت دلخواه از جهان، لذا برنامه‌ریزی عبارت است از جستجو در میان فضای حالت‌ها برای یافتن یک مسیر بهینه به هدف. در حالتی که جهان قطعی است برنامه‌ریزی به یک مسئله جستجوی گراف ساده تبدیل می‌شود که راه‌های زیادی برای آن وجود دارد[۱]. در یک جهان احتمالی برنامه‌ریزی نمی‌تواند با استفاده از

[1] Transition Model

[2] Goal

جستجوی گراف حل شود چرا که گذار یک حالت به حالت دیگرقطعی و مشخص نیست. لذا عامل باید در زمان برنامه‌ریزی احتمالی بودن گذار را مد نظر قرار دهد. برای اینکه ببینید چگونه این امر عملی می‌شود توجه نمایید که در جهان قطعی عامل به صورت پیش فرض میان حالت هدف و حالت غیر هدف برتری قائل می‌شود و به صورت کلی‌تر بین حالت‌های مختلف برتری قائل می‌شود. به طور مثال یک عامل فوتبالیست می‌تواند بین گل‌زدن و بی حرکت ایستادن در مقابل دروازه خالی برتری قائل شود.

یکی از راه‌های فرمول‌بندی ترجیحات اختصاص عدد حقیقی $U(s)$ به هریک از حالات موجود می‌باشد. $U(s)$ مطلوبیت[1] عامل برای حالت S می‌باشد. لذا برای دو حالت s و s' داریم $U(s) > U(s')$ اگر و تنها اگر عامل حالت s را برتر از حالت s' بداند و $U(s) = U(s')$ اگر و تنها اگر عامل حالت s و s' را متفاوت نداند. دقت نمایید که در سیستم‌های چندعامله یک حالت می‌تواند برای تعدادی از عامل مطلوب و برای عامل‌های دیگر در همان زمان غیرمطلوب باشد. در ادامه این پژوهش و در فصل چهارم خواهید دید که این تابع مطلوبیت برای عامل‌های ماشین و قطعه می‌تواند تابع مقدار فرمونی باشد که از طریق هوشمندی جامعه مورچگان[2] بدست آمده است.

ب- تصمیم‌گیری در جهان احتمالی

با وجود تابع مطلوبیت سوالی که در اینجا مطرح می‌شود این است که عامل چگونه می‌تواند از آن برای تصمیم‌گیری استفاده نماید. فرض کنید تنها یک عامل در جهان وجود دارد و جهان احتمالی با مدل گذار $P(s_{t+1}|s_t, a_t)$ باشد فرض کنید حالت فعلی s_t است و عامل به این فکر می‌کند که کدام عمل a_t را اتخاذ نماید فرض کنید $U(s)$ تابع مطلوبیت یک عامل خاص باشد تصمیم‌گیری مطلوبیت محور بر پایه این قضیه بنا شده است که عمل بهینه a_t^* در زمان t می‌بایست مطلوبیت انتظاری را حداکثر نماید:

$$a_t^* = \arg\max_{a_t \in A} \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t) U(s_{t+1}) \qquad (۵-۱)$$

لذا اگر هریک از حالات جهان دارای یک مقدار مطلوبیت باشند عامل می‌تواند در هریک از حالات ممکن محاسبات بالا را انجام داده و عمل a_t^* را اتخاذ می‌نماید. این کار عاملی را فراهم می

[1] Utility

[2] Ant Colony Intelligence

آورد که دارای تابع سیاستی است که در آن هریک از حالات به یک عمل بهینه نگاشته می شود. این تابع سیاست یک تابع یساست حریصانه می باشد که در واقع سیاست بهینه نیز می باشد:

$$\pi^*(s) = arg_a \, max \sum_{s'} P(s' \,|s,a) U(s') \qquad (۶-۱)$$

علاوه بر این راه دیگری نیز برای تعریف سیاست بهینه وجود دارد به این صورت که: برای هر حالت s و هرعمل امکان‌پذیر a می‌توان ارزش بهینه عمل[1] یا ارزش Q^* را که در واقع نشان دهنده درجه خوب بودن عمل a در حالت s برای عامل می‌باشد را تعریف نمود. لذا داریم که $U^*(s) = \max_a Q^*(s,a)$ و لذا سیاست بهینه عامل به صورت زیر محاسبه می شود:

$$\pi^*(s) = arg_a \, max Q^*(s,a) \qquad (۷-۱)$$

این مدل مشابه مدل (۱-۶) می‌باشد با این تفاوت که از مدل گذار استفاده نمی‌کند در بخش ۱-۷ خواهید دید که چگونه می‌توان مقدار بهینه ارزش Q و در نتیجه تابع بهینه سیاست را محاسبه نمود.

۱-۳- بازیهای استراتژیک

در این بخش مسئله تصمیم‌گیری چندعامله‌ها که در آن تعدادی از عامل‌ها در یک محیط با هم در تعامل بوده و به طور همزمان تصمیم‌گیری می‌کنند، بررسی خواهد شد. در این مبحث از تئوری بازیها برای تحلیل مسئله کمک گرفته شده و در کنار آن دو راهکار بنیادی یعنی نقطه تعادل نش[3] و حذف تکراری عمل‌های مغلوب[4] به طور مختصر توضیح داده شده است از آنجا که از این مفاهیم در انجام این پژوهش استفاده نشده، از تشریح مفصل آن خودداری شده است. برای اطلاعات تفصیلی می‌توانید به مرجع [۳] مراجعه نمایید.

۱-۳-۱- تئوری بازی‌ها

همانطور که در بخش ۱-۲ بیان شد، یک عامل عموماً در مورد اثر عمل خود روی محیط اطراف نامطمئن است و لذا عامل می‌بایست این عدم‌قطعیت را در تصمیم‌گیری‌های خود لحاظ نماید. در یک سیستم چندعامله که در آن تعداد زیادی از عامل‌ها به صورت همزمان به تصمیم‌گیری

---

[1] Optimal Action Value

[2] Q-Value

[3] Nash

[4] Iterated Elimination of Dominated Actions