پردیس علوم
دانشکده ریاضی، آمار و علوم کامپیوتر

کاربرد نظریهٔ بازی‌ها در امنیت

نگارنده

سیدمهدی مظلوم

استاد راهنما: دکتر مهدی رضا درویش‌زاده

پایان‌نامهٔ دورهٔ کارشناسی
رشتهٔ ریاضیات و کاربردها

بهار و تابستان ۱۴۰۰

چکیده

هدف اصلی این پروژه، بیان برخی از کاربردهای نظریهٔ بازی‌ها در امنیت است. در این راستا، ابتدا مقدماتی از نظریهٔ بازی‌ها شامل تعریف بازی‌های استراتژیک، استراتژی مخلوط و مفهوم تعادل نش را بیان می‌کنیم. سپس، به طور جداگانه بازی‌های استکلبرگ و بازی‌های بیزی را تعریف می‌کنیم و مفهوم یک بازی بیزی با تعادل استکلبرگ و به دنبال آن، بازی‌های امنیت را شرح می‌دهیم. در بخش اصلی این نوشته، بعد از معرفی سریع‌ترین الگوریتم‌های موجود برای حل بازی‌های بیزی با تعادل استکلبرگ و بررسی میزان کارایی آن‌ها با انجام مقایسه‌های تجربی، دو کاربرد نظریهٔ بازی‌ها در حفاظت از زیرساخت‌های مهم را مطرح می‌کنیم. خواهیم دید که چگونه می‌توان مسئلهٔ امنیت فرودگاه و بنادر مهم را به کمک ابزار نظریهٔ بازی‌ها مدل کرد و با استفاده از آن، امنیت را با وجود منابع محدود در برابر تهدیدها افزایش داد. هم‌چنین، برای هر کدام از این کاربردها مثالی در دنیای واقعی ذکر می‌کنیم و نرم‌افزار ARMOR برای محافظت از فرودگاه بین‌المللی لس‌آنجلس و سیستم PROTECT برای محافظت از بندر بوستون آمریکا را معرفی کرده و عملکرد هر یک را می‌سنجیم.

تقدیم به

همهٔ ذهن‌های روشنی که در اوج آسمان، معصومانه خاموش شدند.

سپاس‌گزاری

خداوند را سپاس‌گزارم که مرا یاری رساند تا بتوانم برگی دیگر از دفتر زندگی خود را ورق زده و با انجام پژوهش پیش رو، این مقطع تحصیلی را به اتمام برسانم. امیدوارم خداوند به بنده توفیق دهد تا بتوانم علم را در راه خدمت به همهٔ انسان‌ها خرج کنم.

از استاد راهنمای عزیزم، جناب آقای دکتر مهدی رضا درویش‌زاده بسیار سپاس‌گزارم که مرا با نظریهٔ بازی‌ها آشنا کردند، الفبای آن را به من آموختند، باعث شدند نخستین گام‌های خود را در این مسیر طی کنم و مرا در انجام این پژوهش و به رشتهٔ تحریر در آوردن آن یاری رساندند.

از خانوادهٔ عزیزم، پدرم، مادرم و خواهرم نیز نهایت قدردانی را دارم که همواره در تمام عرصه‌های زندگی همراهم بوده و در راه رسیدن من به اهدافم از هیچ تلاشی دریغ نکردند. هرچه تا کنون داشته و در آینده خواهم داشت، متعلق به آن‌ها است.

و در پایان قدردان تمام کسانی هستم که برای حفظ امنیت تمام انسان‌ها تلاش می‌کنند و در این راه قدمی هرچند کوچک برمی‌دارند. به ویژه، انسان‌های فداکاری که در برههٔ کنونی برای امنیت جان هم‌نوعان خود در مقابل ویروس کرونا به روش‌های گوناگون می‌کوشند.

پیش‌گفتار

نظریهٔ بازی‌ها شاخه‌ای از ریاضیات است که رقابت میان چند تصمیم‌ساز را در یک محیط استراتژیک، یعنی محیطی که خروجی تصمیم هر تصمیم‌ساز علاوه بر انتخاب خودش متأثر از انتخاب‌های دیگر تصمیم‌سازان است، به صورت ریاضی مدل کرده و به آنان کمک می‌کند تا با توجه به ارجحیت‌هایشان یک تصمیم عقلائی را اتخاذ کنند به گونه‌ای که مطلوبیت حاصل از آن تصمیم، حداقل به اندازهٔ سایر انتخاب‌های در دسترس آنان باشد. در این نوشته، از میان همهٔ کاربردهای نظریهٔ بازی‌ها در زمینه‌های مختلف شامل زیست‌شناسی، علوم اجتماعی، علوم سیاسی و به ویژه اقتصاد، به کاربرد آن در امنیت می‌پردازیم.

حوزه‌های زیادی وجود دارد که ممکن است توسط دشمنان تهدید شود و امنیت‌شان به خطر بیفتد. زیرساخت‌های مهم مثل فرودگاه‌ها و بناهای تاریخی و هم‌چنین مکان‌های با اهمیت سیاسی و اقتصادی از جمله موارد قابل ذکر است که حفظ امنیت آن‌ها همواره یک نگرانی و دغدغهٔ مهم در کل جهان بوده است. از یک طرف، منابع محدود امنیتی امکان پوشش تمام وقت همهٔ مکان‌های قابل دسترس برای دشمنان را به نیروهای امنیتی نمی‌دهد و یک چالش مهم برای آن‌ها چگونگی تخصیص منابع در مکان‌ها و زمان‌های مختلف است. از طرف دیگر، استفاده از الگوهای ثابت و تکراری برای تخصیص منابع باعث می‌شود تا دشمنان به مرور زمان با برنامه‌های امنیتی مسئولان آشنا شده و بتوانند از این آشنایی به نفع خود استفاده کرده و حملات خود را طوری برنامه‌ریزی کنند که به دور از نظارت امنیتی، آسیب‌زننده باشد. در نتیجه، این امر، ما را به راه حل تصادفی‌سازی در تخصیص منابع می‌رساند.

از آنجا که سود نیروهای امنیتی و نیروهای متخاصم از عمل و تصمیم یک‌دیگر تأثیر می‌پذیرد، در محیطی استراتژیک قرار داریم و وارد حوزهٔ نظریهٔ بازی‌ها می‌شویم و برای یافتن بهترین توزیع تصادفی منابع امنیتی برای جلوگیری از نفوذ دشمنان، از آن استفاده می‌کنیم.

این راه حل باید انتظار ما را در پاسخ‌گویی به دو چالش برآورده سازد. ابتدا اینکه، همان‌طور که گفته شد، دشمنان به مرور زمان از برنامه‌های امنیتی پیشنهاد شده آگاهی پیدا می‌کنند. به علاوه، دشمنانی را که ممکن است امنیت جایی را به خطر اندازند، می‌توان در چند دسته شامل تروریست‌ها، قاچاقچیان و یا سارقین مسلح قرار داد که هرکدام دارای انگیزه‌ها و اهداف متفاوتی هستند که به دنبال آن، برای نیروهای امنیتی نیز مقابله با آنان ارزش متفاوتی دارد در حالی که از اینکه با کدام دشمن رو به رو خواهند شد، آگاهی کاملی ندارند. بدین منظور، برای مدل کردن رقابت میان نیروهای امنیتی و دشمنان از بازی‌های بیزی با تعادل استکلبرگ استفاده می‌کنیم. یک بازی استکلبرگ (رهبر - پیرو) به ما این اجازه را می‌دهد تا یک بازیکن را به عنوان رهبر و آغازکنندهٔ بازی در نظر بگیریم که عمل انتخابی‌اش توسط بازیکن دوم قابل مشاهده است و رقیبش (پیرو) با علم به انتخاب رهبر بازی، به دنبال بیشینه کردن سود خود است. به این ترتیب، این دسته از بازی‌ها، مدلی مطابق با عالم واقع را که در آن دشمنان با مشاهدهٔ تصمیم نیروهای امنیتی اقدام به برنامه‌ریزی حملهٔ خود می‌کنند، به ما می‌دهد. هم‌چنین، در یک بازی بیزی قادر هستیم برای یک یا چند بازیکن، انواع مختلفی را در نظر بگیریم که هر نوع یک بازیکن خاص ارجحیت‌های متفاوتی را دارد و رقیب از اینکه با کدام نوع در حال رقابت است، اطلاعات کاملی ندارد. بنابراین، کلاس بازی‌های بیزی نیز به مسئلهٔ متفاوت بودن دشمنان در رقابت با نیروهای امنیتی و عدم آگاهی کامل این نیروها از تفاوت موجود پاسخ می‌دهد.

با توجه به توضیحات داده شده، در فصل اول این نوشته، به بیان مقدمات لازم از نظریهٔ بازی‌ها می‌پردازیم و بعد از بیان چند مفهوم ابتدایی، بازی‌های بیزی و استکلبرگ را مورد بررسی قرار می‌دهیم تا ابزار لازم برای پرداختن به بازی‌های امنیت برایمان فراهم شود. در فصل دوم، روش حل بازی‌های بیزی با تعادل استکلبرگ را بیان می‌کنیم و در این راستا دو

تا از سریع‌ترین و به روزترین الگوریتم‌های به دست آمده برای حل این نوع از بازی‌ها، یعنی الگورتم ASAP و DOBSS را معرفی می‌کنیم و با مقایسهٔ آزمایشات تجربی صورت گرفته، کارایی بهتر این دو الگوریتم نسبت به الگوریتم‌های قبلی را نشان می‌دهیم. در فصل سوم، به کاربرد نظریهٔ بازی‌ها در حفظ امنیت فرودگاه و چگونگی مدل‌سازی به کمک آن می‌پردازیم و این فصل را با معرفی نرم‌افزار ARMOR برای کمک به نیروهای امنیتی فرودگاه بین‌المللی لس‌آنجلس به پایان می‌رسانیم. فصل چهارم، یکی دیگر از کاربردهای نظریهٔ بازی‌ها در امنیت و موضوع حفظ امنیت بنادر مهمی را مطرح می‌کند که امکان تهدید آن‌ها از طریق مسیرهای دریایی وجود دارد. در این فصل نیز سیستم POTECT را برای کمک به گارد ساحلی بندر بوستون آمریکا به عنوان مثالی در دنیای واقعی معرفی می‌کنیم. و در فصل پایانی، یافته‌های این پژوهش را تحت عنوان نتیجه‌گیری بیان می‌کنیم.

در طول انجام کار، از مراجع و منابع مختلفی استفاده شده است که در پایان ذکر شده‌اند. عمدهٔ این مراجع به مقالات منتشر شدهٔ تیمی به رهبری آقای Milind Tambe مربوط می‌شود که در یک دههٔ گذشته به کاربرد نظریهٔ بازی‌ها در امنیت حوزه‌های مختلف پرداخته‌اند و پیشروی موضوعی بودند که در ادامه خواهید خواند.

فهرست مطالب

۱ مفاهیم مقدماتی **۱**

۱.۱ بازی‌های استراتژیک و تعادل نش ۱

۲.۱ استراتژی‌های مخلوط و تعادل نش مخلوط ۶

۳.۱ بازی‌های بیزی با تعادل استکلبرگ ۱۰

۱.۳.۱ بازی‌های استکلبرگ ۱۰

۲.۳.۱ بازی‌های بیزی ۱۳

۳.۳.۱ بازی‌های امنیت ۱۴

۴.۳.۱ تبدیل هارسانی ۱۵

۴.۱ یادداشت‌های بهینه‌سازی خطی ۱۷

۵.۱ یادداشت‌های نظریهٔ گراف ۱۹

۲ الگوریتم‌های حل بازی‌های بیزی با تعادل استکلبرگ **۲۰**

۱.۲ توصیف دامنهٔ امنیتی ۲۱

۲.۲ الگوریتم ASAP ۲۳

۳.۲ الگوریتم DOBSS ۲۹

۴.۲ نتایج تجربی ۳۱

۳ کاربرد نظریهٔ بازی‌ها در امنیت فرودگاه‌ها ۳۷
۱.۳ دامنهٔ امنیتی فرودگاه ۳۷
۲.۳ مدل‌سازی ۳۹
۳.۳ نرم‌افزار $ARMOR$ ۴۰
۴.۳ نتایج تجربی ۴۲

۴ کاربرد نظریهٔ بازی‌ها در امنیت بنادر ۴۵
۱.۴ مدل بازی ۴۶
۲.۴ به کار گیری مدل QR ۴۹
۳.۴ ارزیابی سیستم PROTECT ۵۲

۵ نتیجه گیری ۵۵

فصل اول

مفاهیم مقدماتی

۱.۱ بازی‌های استراتژیک و تعادل نش

بازی استراتژیک مدلی برای رقابت بین تصمیم‌سازان است که از این پس، از آن‌ها به عنوان بازیکنان بازی یاد می‌کنیم. برای هر بازیکن مجموعه‌ای از عمل‌های قابل دسترس موجود است و از آنجا که خروجی بازی برای هر یک از آن‌ها متأثر از عمل‌های انتخاب شده توسط سایر بازیکنان نیز است، هر بازیکن ارجحیت‌هایی روی برداری از عمل‌های انتخاب شده توسط همهٔ بازیکنان شامل خودش دارد. در این نوع از بازی‌ها، بازیکنان عمل‌هایشان را به طور هم‌زمان انتخاب می‌کنند و از تصمیم هم‌دیگر مطلع نیستند. به طور دقیق‌تر، یک بازی استراتژیک به صورت زیر تعریف می‌شود:

تعریف ۱.۱. یک بازی استراتژیک با ارجحیت‌های ترتیبی شامل اجزاء زیر است:

- مجموعه‌ای متناهی از بازیکنان

- برای هر بازیکن، یک مجموعه از عمل‌ها
- متناظر با هر بازیکن، ارجحیت‌هایی روی مجموعه بردارهای از عمل‌ها

موقعیت‌های زیادی وجود دارد که می‌توان آن‌ها را به صورت یک بازی استراتژیک مدل کرد. مثال زیر، یکی از آشناترین مثال‌ها در این مورد است که با نام "معمای زندانی" شناخته می‌شود.

مثال ۲.۱. دو نفر به اتهام ارتکاب جرمی در دو سلول جداگانه قرار داده شده‌اند. شواهدی برای اینکه یکی از این دو متهم حتماً مجرم است، وجود دارد؛ ولی این شواهد برای یافتن مجرم از بین آن‌ها کافی نیست و نیاز به شهادت یکی از مظنونین علیه دیگری است. اگر هیچ‌کدام علیه دیگری شهادت ندهد، هر دو به یک سال زندان محکوم می‌شوند. اگر فقط یکی از آن‌ها علیه زندانی دیگر شهادت بدهد، خودش آزاد می‌شود و متهم دیگر به چهار سال زندان محکوم می‌شود. هم‌چنین در صورتی که هر دو زندانی علیه یک‌دیگر شهادت بدهند، هر کدام به سه سال زندان محکوم خواهند شد. این وضعیت را می‌توان به صورت یک بازی استراتژیک مدل کرد:

مجموعه بازیکنان = { متهم اول، متهم دوم }

مجموعه عمل بازیکنان = { شهادت دادن، سکوت کردن }

ارجحیت‌های بازیکنان:

با توجه به سناریوی بیان شده، برای هر بازیکن، اینکه خودش شهادت بدهد و رقیبش سکوت دارد، مطلوبیت بیشتری دارد. به دنبال آن، دو حالتی که هر دو سکوت کنند و هر دو اعتراف کنند، به ترتیب قرار می‌گیرند. و در انتها، کم‌ترین سود برای هر بازیکن حالتی است که خودش سکوت کند و رقیبش علیه او شهادت بدهد.

به این ترتیب، شکل ۱.۱ فرم ماتریسی این بازی را نشان می‌دهد.

		متهم دوم: سکوت	متهم دوم: شهادت
متهم اول	سکوت	2,2	0,3
	شهادت	3,0	1,1

شکل ۱.۱: بازی معمای زندانی.

توجه داشته باشید که سودهای نوشته شده در ماتریس برای هر بازیکن تنها به صورتی است که ترتیب ارجحیت‌های او را حفظ کند؛ یعنی برای هر بازیکن i که $i = 1, 2$، اگر $u_i(a, b)$ سود حاصل برای او باشد وقتی که خودش عمل a و رقیبش عمل b را انتخاب می‌کند، باید داشته باشیم

$$u_i(F, Q) > u_i(Q, Q) > u_i(F, F) > u_i(Q, F).$$

که در آن Q و F به ترتیب نشان‌دهندهٔ "سکوت کردن" و "شهادت دادن" است.
این بازی ساده بیانگر وضعیتی است که اگرچه سود بازیکنان در همکاری با یک‌دیگر است ولی هر بازیکن، مستقل از عمل انتخاب شده توسط رقیبش، دارای انگیزهٔ "سواری رایگان" است و ترجیح می‌دهد تا با شهادت دادن علیه دیگری، سود بیشتری به دست آورد؛ در نتیجه، هر دو علیه هم‌دیگر شهادت می‌دهند.
اکنون می‌خواهیم به این سؤال پاسخ دهیم که بازیکنان در یک بازی استراتژیک باید چه انتخابی داشته باشند و چگونه تصمیم بگیرند. بر اساس مدل انتخاب عقلانی، هر بازیکن بهترین عمل ممکن خود را از نظر سود انتخاب می‌کند. اما همان‌طور که گفته شد، خروجی بازی علاوه بر تصمیم خودش، به انتخاب سایر بازیکنان نیز وابسته است؛ بنابراین به هنگام انتخاب عمل خود، اعتقادی نسبت به عمل دیگر بازیکنان در ذهن خود دارد و می‌داند که چگونه رفتار می‌کنند. این اعتقاد مبتنی بر تجربیات گذشتهٔ او از انجام چنین بازی‌هایی است و اعتقاد هر بازیکن راجع به دیگر بازیکنان صحیح است. با این فرضیات، تعادل نش به صورت

زیر تعریف می‌شود:

تعریف ۳.۱. یک تعادل نش برداری از عمل‌های بازیکنان مثل a^* است با این ویژگی که هیچ بازیکنی نمی‌تواند با تغییر استراتژی و عمل انتخابی‌اش سود اکیداً بیشتری به دست آورد وقتی که عمل سایر بازیکنان ثابت مانده است.
به طور رسمی، اگر $< N, (A_i)_{i=1}^n, (u_i)_{i=1}^n >$ یک بازی استراتژیک با ارجحیت‌های ترتیب باشد. گوییم

$$a^* = (a_1^*, ..., a_n^*) \in A = A_1 \times ... \times A_n$$

یک تعادل نش برای این بازی است هرگاه برای هر $i \in N$ داشته باشیم

$$u_i(a^*) \geq u_i(a_i, a_{-i}^*) \forall a_i \in A_i$$

که در آن

$$a_{-i} = (a_1, ..., a_{i-1}, a_{i+1}, ..., a_n)$$

و u_i تابع سودی است که ارجحیت‌های بازیکن i ام را نشان می‌دهد.

در مثال "معمای زندانی" زوج استراتژی (F, F) یک تعادل نش این بازی است؛ زیرا هر یک از دو بازیکن با فرض اینکه بازیکن مقابلشان علیه آن‌ها شهادت می‌دهد، اگر به جای "شهادت دادن"، "سکوت کردن" را انتخاب کنند؛ سودشان از ۱ به ۰ کاهش پیدا می‌کند و وضعیت‌شان بدتر خواهد شد. هم‌چنین، به سادگی می‌توان دید که این بازی تعادل نش دیگری ندارد.

اگرچه می‌توانیم در بازی‌های ساده که مجموعه عمل بازیکنان اعضای زیادی ندارد، با بررسی هر بردار از عمل‌ها تعادل‌های نش را طبق تعریف آن بیابیم ولی در بازی‌های پیچیده‌تر این کار دشوار است و بهتر است که از "تابع بهترین پاسخ" استفاده کنیم.

تعریف ۴.۱. فرض کنید $< N, (A_i)_{i=1}^n, (u_i)_{i=1}^n >$ یک بازی استراتژیک با ارجحیت‌های ترتیب باشد. در این صورت تابع بهترین پاسخ برای بازیکن i ام را با B_i نشان داده و به صورت زیر تعریف می‌کنیم.

$$B_i : A_{-i} \to A_i$$
$$B_i(a_{-i}) = \{a_i \in A_i | u_i(a_i, a_{-i}) \geq u_i(a_i^{'}, a_{-i}) \forall a_i^{'} \in A_i\}$$

این تابع که یک تابع مجموعه مقدار است، بهترین عمل هر بازیکن را مشخص می‌کند وقتی که عمل سایر بازیکنان داده شده باشد.

گزاره ۵.۱. برای بازی استراتژیک $< N, (A_i)_{i=1}^n, (u_i)_{i=1}^n >$، بردار عمل $a^* \in A$ یک تعادل نش است اگر و تنها اگر برای هر i داشته باشیم

$$a_i^* \in B_i(a_{-i}^*)$$

در مثال "معمای زندانی" برای زوج استراتژی (F, F) که تنها تعادل نش این بازی بود، داریم

$$F \in B_1(a_2^*), F \in B_2(a_1^*)$$

۲.۱ استراتژی‌های مخلوط و تعادل نش مخلوط

در این بخش قصد داریم استراتژی‌های محض بازیکنان را به استراتژی‌های مخلوط تعمیم دهیم. هدف از این کار، این است که ما در برخی از بازی‌ها هیچ تعادل نش محضی نداریم و بازیکنان نمی‌توانند یک استراتژی محض را به عنوان یک استراتژی تعادلی برگزینند.. بازی زیر را به عنوان مثال در نظر بگیرید.

مثال ۶.۱. دو بازیکن به طور هم‌زمان و جداگانه سکه‌ای را پرتاب می‌کنند. اگر سکه‌ها مطابق باشند؛ یعنی هر دو شیر یا هر دو خط باشند، بازیکن دوم یک سکه به بازیکن اول پرداخت می‌کند و در غیر این صورت، بازیکن اول یک سکه به بازیکن دوم پرداخت می‌کند. فرم ماتریسی این بازی به صورت شکل ۲.۱ است. با بررسی بردار عمل بازیکنان در این بازی

		بازیکن دوم: شیر	بازیکن دوم: خط
بازیکن اول	شیر	1,-1	-1,1
	خط	-1,1	1,-1

شکل ۲.۱: بازی مسابقهٔ پنی.

که به "مسابقهٔ پنی" معروف است، می‌توان دید که هیچ حالت ایستایی برای این بازی وجود ندارد. همین امر انگیزه‌ای است تا استراتژی‌های مخلوط و حالت ایستای تصادفی را تعریف کنیم. بدین منظور، ابتدا باید یک بازی استراتژیک با ارجحیت‌های VNM را تعریف کنیم.

تعریف ۷.۱. یک بازی استراتژیک با ارجحیت‌های VNM از اجزاء زیر تشکیل می‌شود:

- مجموعه‌ای از بازیکنان
- برای هر بازیکن، یک مجموعه از عمل‌ها

- متناظر با هر بازیکن، ارجحیت‌هایی روی توزیع‌های احتمال روی بردارهای عمل بازیکنان که توسط ارزش انتظاری یک تابع سود روی بردارهای عمل بازیکنان نمایش داده می‌شود.

در تعمیم مفهوم تعادل نش به یک حالت ایستای تصادفی در یک بازی استراتژیک با ارجحیت‌های VNM به بازیکنان اجازه می‌دهیم تا یک توزیع احتمال روی مجموعه عمل‌هایشان داشته باشند تا اینکه محدود به انتخاب یک استراتژی محض باشند. از این توزیع احتمال به عنوان یک استراتژی مخلوط یاد می‌کنیم.

تعریف ۸.۱. منظور از استراتژی مخلوط برای یک بازیکن در یک بازی استراتژیک، یک توزیع احتمال روی عمل‌های آن بازیکن است.

اگر یک بازیکن در یک استراتژی مخلوط، احتمال ۱ را به یک عمل و صفر را به بقیهٔ عمل‌هایش نسبت دهد، این استراتژی همان استراتژی محض خواهد بود.
اکنون می‌توانیم تعادل نش مخلوط را مشابه حالت ایستا تعریف کنیم.

تعریف ۹.۱. برداری از استراتژی‌های مخلوط مانند α^* را یک تعادل نش مخلوط گوییم هرگاه برای هر استراتژی مخلوط بازیکن i ام مانند α_i داشته باشیم

$$U_i(\alpha^*) \geq U_i(\alpha_i, \alpha^*_{-i})$$

که در آن $U_i(\alpha)$ سود انتظاری بازیکن i ام نسبت به α است.

هم‌چنین گزارهٔ ۵.۱ عیناً برای استراتژی‌های مخلوط و تعادل نش مخلوط نیز برقرار است؛ یعنی

گزاره ۱۰.۱. برداری از استراتژی‌های مخلوط مانند α^* یک تعادل نش مخلوط است اگر و تنها اگر برای هر i داشته باشیم

$$\alpha_i^* \in B_i(\alpha_{-i}^*)$$

که در آن B_i تابع بهترین پاسخ بازیکن i ام است.

در مثال زیر تعادل‌های نش مخلوط بازی "مسابقهٔ پنی" را با استفاده از توابع بهترین پاسخ به دست می‌آوریم.

مثال ۱۱.۱. فرض کنید بازیکنان اول و دوم به ترتیب به شیر احتمال p و q نظیر می‌کنند. با فرض اینکه استراتژی مخلوط بازیکن دوم داده شده باشد، سود انتظاری بازیکن اول از انتخاب شیر برابر است با

$$q.1 + (1-q).(-1) = 2q - 1$$

و از انتخاب خط برابر است با

$$q.(-1) + (1-q).1 = 1 - 2q$$

بنابراین تابع بهترین پاسخ بازیکن اول عبارت است از و به طور مشابه تابع بهترین پاسخ بازیکن دوم عبارت است از نمودار زیر تلاقی توابع بهترین پاسخ دو بازیکن را نشان می‌دهد. بنابراین این بازی دارای تنها یک تعادل نش مخلوط به صورت زیر و فاقد تعادل نش محض است.

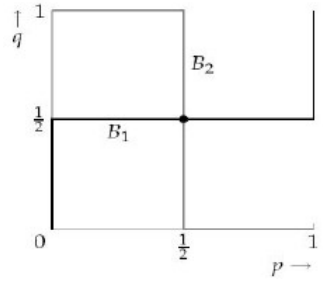
$$((\tfrac{1}{2}, \tfrac{1}{2}), (\tfrac{1}{2}, \tfrac{1}{2})).$$

$$B_1(q) = \begin{cases} \{0\} & q < \frac{1}{2} \text{ اگر}, \\ \{p | 0 \le p \le 1\} & q = \frac{1}{2} \text{ اگر}, \\ \{1\} & q > \frac{1}{2} \text{ اگر} \end{cases}$$

شکل ۳.۱: تابع بهترین پاسخ بازیکن اول

$$B_2(p) = \begin{cases} \{1\} & p < \frac{1}{2} \text{ اگر}, \\ \{q | 0 \le q \le 1\} & p = \frac{1}{2} \text{ اگر}, \\ \{0\} & p > \frac{1}{2} \text{ اگر} \end{cases}$$

شکل ۴.۱: تابع بهترین پاسخ بازیکن دوم



شکل ۵.۱: تلاقی توابع بهترین پاسخ دو بازیکن

۳.۱ بازی‌های بیزی با تعادل استکلبرگ

در فصل‌های بعد از کلاس خاصی از بازی‌ها به نام بازی‌های بیزی با تعادل استکلبرگ [۱] بهره می‌بریم. قبل از پرداختن به این نوع از بازی‌ها، ابتدا مفهوم بازی استکلبرگ را در قالب یک مثال بیان می‌کنیم، با یادآوری این مطلب که در مسئلهٔ امنیت زیرساخت‌ها با منابع محدودی رو به رو هستیم که قصد محافظت از قسمت‌های مختلف یک زیرساخت مثل فرودگاه را با درجه اهمیت متفاوت دارند.

۱.۳.۱ بازی‌های استکلبرگ

یک فرودگاه ساده با تنها دو پایانه را در نظر بگیرید که یک واحد پلیس قصد محافظت از آن‌ها را در برابر یک دشمن دارد و پایانهٔ اول از اهمیت بیشتری برای پلیس برخوردار است. مدل ریاضی رقابت بین نیروهای پلیس و دشمن در شکل ۶.۱ به صورت یک بازی استراتژیک آمده است. در این بازی، سودهای پلیس و دشمن هر دو در بازهٔ $(-5,5)$ قرار گرفته‌اند.

دشمن

پلیس		پایانهٔ ۱	پایانهٔ ۲
	پایانهٔ ۱	5,-3	-1,1
	پایانهٔ ۲	-5,5	2,-1

شکل ۶.۱: بازی استکلبرگ بین پلیس و دشمن.

این سودها بر حسب عواملی مثل آسیب‌های جانی یا مالی برای طرفین، مهم‌تر بودن پایانهٔ ۱ نسبت به ۲ و هم‌چنین احتمالات مربوط به دستگیری دشمن در هر یک از دو پایانه به صورت تجربی به دست آمده‌اند.

اگر پلیس همواره فقط پایانهٔ ۱ را به دلیل اهمیت بیشترش مورد نظارت قرار دهد، یک دشمن عاقل بعد از مدتی و با مشاهدهٔ برنامهٔ امنیتی پلیس، به پایانهٔ ۱ حمله نخواهد کرد و در نتیجه

[۱] Bayesian Stackelberg games

پایانهٔ ۲ را مورد حمله قرار خواهد داد. اگر پلیس همواره از پایانهٔ ۲ مراقبت کند، به طور مشابه دشمن با آگاهی پیدا کردن از این موضوع، برنامهٔ حملهٔ خود را برای پایانهٔ ۱ تنظیم خواهد کرد. اما اگر پلیس به صورت تصادفی عمل کند و به عنوان مثال در ۶۰ درصد زمان روز پایانهٔ ۱ را نظارت کند و در ۴۰ درصد زمان باقی‌مانده، پایانهٔ ۲ را تحت نظر داشته باشد، نتیجهٔ بهتری برای پلیس حاصل خواهد شد؛ زیرا اگرچه دشمن مشاهده می‌کند که پلیس در ۶۰ درصد مواقع در پایانهٔ ۱ و در ۴۰ درصد مواقع در پایانهٔ ۲ قرار دارد، ولی برای روزهای بعد نمی‌تواند از برنامهٔ زمانی پلیس برای امنیت این دو پایانه اطمینان و قطعیت داشته باشد و این نااطمینانی دشمن برای پلیس سود بیشتری خواهد داشت.

به این نوع از بازی‌ها که در آن یک بازیکن مثل پلیس از قبل به یک استراتژی مخلوطی متعهد می‌شود و بازیکن دوم (در اینجا دشمن) بعد از مشاهدهٔ استراتژی پلیس، استراتژی خود را انتخاب می‌کند، یک بازی استکلبرگ گفته می‌شود. (این نام‌گذاری ریشه در کار هنریش استکلبرگ دارد که نخستین بار در سال ۱۹۳۴ روی این نوع از بازی‌ها تحقیق می‌کرد.) توجه کنید که در مثال ما، بازیکن شروع‌کننده (پلیس) که به آن رهبر بازی نیز گویند، می‌تواند به یک استراتژی تصادفی یا مخلوط متعهد شود ولی واکنش بازیکن دوم (دشمن) که به آن پیرو بازی نیز می‌گویند، انتخاب یک استراتژی محض خواهد بود. در این مثال، ما فرض کردیم که دشمن تنها از برنامهٔ امنیتی پلیس در روزهای قبل از حملهٔ خود آگاهی دارد ولی به دلیل تصادفی بودن استراتژی پلیس دقیقاً نمی‌داند این برنامه در روزی که برنامهٔ حملهٔ خود را آماده کرده است، به چه صورت خواهد بود.

بازی‌های استکلبرگ را با نام بازی "رهبر - پیرو" و یا "مهاجم - مدافع" نیز می‌شناسند و ما نیز گاهی اوقات از عبارات "مدافع" و "مهاجم" استفاده خواهیم کرد. همان‌طور که گفته شد، مهاجم از استراتژی مخلوط مدافع آگاهی کامل دارد و بر این اساس در تلاش است تا سود خود را بیشینه سازد. در ادامه هدف ما پاسخ دادن به این سؤال است که کدام استراتژی مخلوط برای مدافع بیش‌ترین سود را به همراه دارد. در مثال بالا که یک بازی ساده با یک واحد پلیس و تنها دو پایانه بود، یافتن بهینه‌ترین حالت تخصیص منابع چندان دشوار نخواهد

بود و با انجام کمی محاسبات دستی می‌توان جواب را یافت ولی در یک مسئله‌ای شامل صدها هدف حمله و چند واحد پلیس، نیازمند ابزارهای محاسباتی هستیم.
مثال زیر نشان می‌دهد که رهبر در یک بازی استکلبرگ از مزیت بیشتری نسبت به یک بازی هم‌زمان برخوردار است.

مثال ۱۲.۱. بازی سادهٔ شکل ۷.۱ را با سودهای داده شده در نظر بگیرید که در آن رهبر، بازیکن سطری و پیرو، بازیکن ستونی است. ابتدا فرض کنید بازی به صورت هم‌زمان باشد و رهبر و پیرو هر دو در یک زمان مشخص استراتژی خود را انتخاب کنند. در این صورت، تنها تعادل نش محض این بازی زمانی است که رهبر a و پیرو c را بازی می‌کنند که برای رهبر سود ۲ را به همراه دارد. حال اگر رهبر به استراتژی مخلوط یکنواختی متعهد شود که در آن هر کدام از عمل‌های a و b را با احتمال برابر $\frac{1}{2}$ انتخاب می‌کند، در این صورت پیرو برای اینکه سود خود را بیشینه سازد، d را انتخاب خواهد کرد که منجر به سود ۳/۵ برای رهبر می‌شود. بنابراین، با متعهد شدن به یک استراتژی، رهبر می‌تواند سود خود را نسبت به حالتی که با رقیبش هم‌زمان بازی می‌کردند، افزایش دهد.

	c	d
a	2,1	4,0
b	1,0	3,2

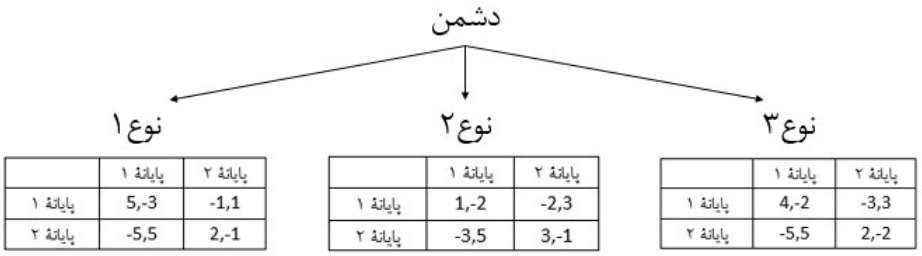
شکل ۷.۱: بازی مثال ۱۲.۱.

در ارتباط با بازی‌های استکلبرگ و تکمیل مثال بالا ذکر دو نکته ضروری است. ابتدا اینکه، جواب تعادل استکلبرگ در بازی‌های منتاهی دو نفره‌ای در فرم نرمال قابل اعمال است که پاسخ پیرو به استراتژی رهبر یگانه باشد؛ در غیر این صورت در پاسخ‌های احتمالی پیرو و در نتیجه در میزان هزینهٔ رهبر ابهام وجود دارد. به علاوه، خروجی جواب استکلبرگ یک بازی همواره برای دو بازیکن بهتر از خروجی تعادل نش نیست. در مثال بالا سود پیرو

از تعادل نش و تعادل استکلبرگ یکسان است ولی مثال‌هایی هم وجود دارد که سود پیرو حاصل از جواب استکلبرگ اکیداً کمتر از سود حاصل از تعادل نش برای او باشد.

۲.۳.۱ بازی‌های بیزی

مسئله‌ای که در واقعیت با آن رو به رو هستیم، فراتر از یک بازی استکلبرگ است. در همان مثال سادهٔ فرودگاه در بخش قبل، دشمن ممکن است گونه‌های متفاوتی داشته باشد. تروریست‌ها، قاچاقچیان مواد مخدر، سارقان و بمب‌گذارانی که امنیت فرودگاه را تهدید می‌کنند، مثال‌هایی از انواع مختلف دشمن هستند که هرکدام از آن‌ها اهداف متفاوتی را دنبال می‌کنند و سودهای متفاوتی را از حمله‌های خود به دست می‌آورند. به عنوان مثال، ممکن است یک نوع دشمن ارزش یکسانی برای حمله به پایانه‌های ۱ و ۲ قائل باشد در صورتی که برای نوع دیگری از دشمن پایانهٔ اول اولویت داشته باشد و یا ممکن است دشمنی قادر به حمله به یکی از پایانه‌ها نباشد. از طرف دیگر، نیروی امنیتی (پلیس فرودگاه) نیز از مقابله با هر یک از این دشمنان ممکن است سود متفاوتی کسب کند. به علاوه، نیروی امنیتی در مورد اینکه در هر حمله با کدام نوع دشمن رو به رو خواهد شد، اطلاعات کاملی ندارد و نامطمئن است؛ بنابراین رقابت بین پلیس و دشمن فقط یک ماتریس سود ندارد و ما به ازای هر نوع دشمن یک ماتریس سود خواهیم داشت. شکل ۸.۱ این موضوع را بهتر برای ما روشن می‌کند. با توجه به مثال بالا، به طور خلاصه، یک بازی بیزی تعمیمی از یک بازی



شکل ۸.۱: یک بازی بیزی با سه نوع پیرو

استراتژیک است که در آن بازیکنان لزوماً دارای اطلاعات کاملی نسبت به بازی نیستند؛ به ویژه هر بازیکن لزوماً ارجحیت‌های دیگر بازیکنان را نمی‌داند.

۳.۳.۱ بازی‌های امنیت

در ادامه، تمرکز ما روی بازی‌های بیزی با تعادل استکلبرگ و به طور جزئی‌تر بازی‌های امنیت خواهد بود. یک بازی امنیت دارای این ویژگی است که آنچه برای مهاجم مطلوب است، برای مدافع نامطلوب است، و برعکس. به عبارت دیگر، سودهای مهاجم و مدافع از یک بردار عمل خاص در تقابل با یک‌دیگر است. هرچند به طور کلی لزومی ندارد این سودها قرینهٔ یک‌دیگر باشند و به اصطلاح یک بازی مجموع صفر داشته باشیم؛ چرا که ارزش یک هدف برای مهاجم لزوماً با ارزش و اهمیت آن هدف برای پلیس برابر نیست و یا ممکن است دشمن از شکست خوردن در یک حمله خروجی منفی‌ای به دست نیاورد؛ زیرا هر حملهٔ ناموفق نیز ممکن است باعث ایجاد ترس عمومی شود و تبعاتی منفی را برای پلیس به همراه داشته باشد. در اصل، در بازی‌های امنیت، اگر مهاجم هدفی را مورد حمله قرار دهد که این هدف توسط پلیس به طور کامل پوشش داده و محافظت شده است، سود کمتری به دست می‌آورد نسبت به حالتی که یک حملهٔ موفقیت‌آمیز به همان هدف داشته باشد. به عنوان مثال، در شکل ۶.۱ ، وقتی مهاجم به پایانهٔ ۱ حمله می‌کند، در حالتی که پلیس نیز همان پایانه را برای محافظت انتخاب کرده، سود -3 به دست می‌اورد اما در حالتی که پلیس پایانهٔ ۲ را برای دفاع انتخاب کرده و پایانهٔ ۱ خالی از نیروی امنیتی است، سود 5 را به دست می‌آورد. این شرایط برای مدافع به طور عکس برقرار است. همان‌طور که گفته شد، هدف ما یافتن استراتژی بهینه برای تخصیص نیروهای امنیتی است که در فصل بعدی به آن‌ها می‌پردازیم و دو تا از سریع‌ترین الگوریتم‌ها در این زمینه را معرفی خواهیم کرد. اما قبل از آن، یکی از رایج‌ترین روش‌های حل بازی‌های بیزی را مورد بررسی قرار می‌دهیم.

۴.۳.۱ تبدیل هارسانی

یافتن پاسخ بهینه برای یک بازی بیزی با تعادل استکلبرگ که انواع زیادی از پیرو وجود داشته باشد، از نظر زمانی دشوار است و کارهای اولیهٔ صورت گرفته در این راستا، الگوریتم‌هایی بودند که برای حل بازی بیزی نیاز به تبدیل آن به فرم نرمال داشتند. این تبدیل را با نام "تبدیل هارسانی" [۲] می‌شناسیم. اگرچه الگوریتم‌هایی که در فصل بعد معرفی می‌کنیم بدون نیاز به تبدیل هارسانی پاسخ بهینهٔ یک بازی بیزی را پیدا می‌کنند و از همین رو، بسیار سریع‌تر و کارآمدتر عمل می‌کنند اما شناخت تبدیل هارسانی و تأثیر آن روی الگوریتم‌های اولیهٔ حل بازی‌های بیزی نیز از اهمیت زیادی برخوردار است.

از طریق یک مثال، تبدیل هارسانی و روش آن برای تبدیل یک بازی بیزی به فرم نرمال را معرفی می‌کنیم. یک بازی استکلبرگ را در نظر بگیرید که در آن دو نوع از پیرو وجود دارد. رهبر با احتمال α با نوع اول پیرو مواجه خواهد شد و با احتمال $1-\alpha$ با نوع دوم رو به رو می‌شود. شکل زیر علاوه بر اطلاعات گفته شده، نشان‌دهندهٔ مجموعه عمل بازیکنان و سودهای آن‌ها نیز است. با به کار بردن تبدیل هارسانی یک جدول ۲ در ۴ حاصل می‌شود که

نوع اول پیرو

	c	d
a	2,1	4,0
b	1,0	3,2

نوع دوم پیرو

	c'	d'
a	1,1	2,0
b	0,1	3,2

شکل ۹.۱: یک بازی بیزی با دو نوع پیرو.

در آن رهبر هم‌چنان دو استراتژی دارد در حالی که یک نوع پیرو با چهار (2×2) استراتژی خواهیم داشت. سودهای بازیکنان نیز با محاسبهٔ ارزش انتظاری حاصل از هر استراتژی برای آن‌ها به دست می‌آید. شکل تغییر یافتهٔ بازی بعد از تبدیل هارسانی به صورت شکل ۱۰.۱ است. برای نمونه، موقعیتی را در بازی تبدیل یافته در نظر بگیرید که رهبر و پیرو به ترتیب استراتژی‌های a و cc' را انتخاب می‌کنند. سود رهبر در بازی جدید به صورت

[۲] Harsanyi transformation

	cc'	cd'	dc'	dd'
a	$2\alpha+(1-\alpha),1$	$2,\alpha$	$4\alpha+(1-\alpha),(1-\alpha)$	$4\alpha+2(1-\alpha),0$
b	$\alpha,1-\alpha$	$\alpha+3(1-\alpha),2(1-\alpha)$	$3\alpha,2\alpha+(1-\alpha)$	3,2

شکل ۱۰.۱: یک بازی بیزی پس از تبدیل هارسانی

مجموع وزن‌داری از سودهایش در دو جدول شکل ۹.۱ به دست می‌آید؛ به عبارت دیگر، α برابر سودی که از تقابل با نوع اول پیرو به دست می‌آورد وقتی که پیرو c را بازی می‌کند با $1-\alpha$ برابر سودی که از تقابل با نوع دوم پیرو به دست می‌آورد وقتی که پیرو c' را انتخاب می‌کند، جمع می‌شود و به عنوان سود حاصل برای او از انتخاب عمل a توسط خودش و cc' توسط رقیبش در نظر گرفته می‌شود. بقیهٔ سودهای نوشته شده در جدول بالا نیز با فرمولی مشابه برای هر دو بازیکن قابل محاسبه هستند.

به طور کلی، برای n نوع پیرو که هر کدام دارای k استراتژی هستند، در بازی تبدیل یافته با تبدیل هارسانی یک نوع پیرو با k^n استراتژی خواهیم داشت و همان‌طور که مشاهده می‌شود، در این روش یک بعد جدول که مربوط به مجموعه عمل‌های پیرو است، به طور نمایی رشد می‌کند؛ به همین علت این روش از نظر محاسباتی در ابعاد بالا زمانبر و انگیزه‌ای است تا به دنبال الگوریتم‌های سریع‌تر برای یافتن جواب در بازی‌های بیزی با تعادل استکلبرگ باشیم.

۴.۱ یادداشت‌های بهینه‌سازی خطی

تعریف ۱۳.۱. یک مسئلهٔ بهینه‌سازی مقید در فضای R^n به صورت رابطهٔ (۱) تعریف می‌شود. در اینجا، $c_1x_1+c_2x_2+...+c_nx_n$ تابع هدف نامیده می‌شود که به دنبال کمینه‌سازی آن هستیم. به ضرایب $c_1,c_2,...,c_n$ "ضرایب هزینه" و متغیرهای $x_1,x_2,...,x_n$ "متغیرهای تصمیم" می‌گوییم. هم‌چنین، نامساوی $\sum_{j=1}^{n} a_{ij}x_j \geq b_i$ متناظر با قید محدودیت i ام است. به ضرایب a_{ij} نیز "ضرایب فنی گوییم."

تعریف ۱۴.۱. اگر مسئلهٔ برنامه‌ریزی خطی (P) را داشته باشیم، دوگان این مسئله به صورت

$$
\begin{aligned}
\text{Minimize} \quad & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\text{Subject to} \quad & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \geqslant b_1 \\
& a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \geqslant b_2 \\
& \vdots \qquad \vdots \qquad\qquad \vdots \quad \vdots \\
& a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \geqslant b_m \\
& x_1, \quad x_2, \quad \ldots, \quad x_n \geqslant 0
\end{aligned}
\tag{1}
$$

(D) تعریف می‌شود.

$$
\begin{aligned}
&\text{P: Minimize} \quad \mathbf{cx} \\
&\text{Subject to} \quad \mathbf{Ax} \geqslant \mathbf{b} \\
&\qquad\qquad\quad \mathbf{x} \geqslant \mathbf{0}
\end{aligned}
$$

$$
\begin{aligned}
&\text{D: Maximize} \quad \mathbf{wb} \\
&\text{Subject to} \quad \mathbf{wA} \leqslant \mathbf{c} \\
&\qquad\qquad\quad \mathbf{w} \geqslant \mathbf{0}
\end{aligned}
$$

گزاره ۱۵.۱. (قضیهٔ جبری KKT) مسئلهٔ برنامه‌ریزی خطی (p) را همانند آنچه در تعریف دوگان یک مسئلهٔ برنامه‌ریزی خطی داشتیم، در نظر بگیرید. در این صورت بردار $x^0 \in R^n$ یک جواب بهینه برای (P) است اگر و تنها اگر داشته باشیم

$$
Ax^0 \geq b, x^0 \geq 0
$$

$$
\exists y^0 \in R^{1\times m}\ s.t. \begin{cases} y^0A \leq c, y^0 \geq 0 \\ y^0(Ax^0 - b) = (y^0A - c)x^0 = 0 \end{cases}
$$

اثبات این گزاره در مرجع (۷) آمده است.

۵.۱ یادداشت‌های نظریهٔ گراف

در این بخش کوتاه به بیان چند تعریف مورد استفاده در فصل چهارم می‌پردازیم.

تعریف ۱۶.۱. یک گراف G از یک مجموعهٔ ناتهی $V(G)$ به نام رأس‌های گراف و یک مجموعهٔ $E(G)$ به نام یال‌های گراف تشکیل می‌شود که هر یال یک زیرمجموعهٔ تک عضوی یا دو عضوی از $V(G)$ است. به علاوه، اگر $\{x, y\} = e \in E(G)$ یک یال باشد، در این صورت x و y را دو سر یا دو انتهای یال e می‌نامیم و گوییم x و y همسایه یا متصل هستند. یالی که دو انتهایش یک باشد را نیز "طوقه" می‌نامیم.
حال فرض کنید G یک گراف باشد. یک "گشت" در G دنباله‌ای مانند $V_0e_1V_1e_2V2...e_kV_k$ از رئوس و یال‌های G است به طوری که به ازای هر $1 \leq i \geq k$، e_i یالی بین $v_{i=1}$ و v_i است. گشتی که یال تکراری نداشته باشد را "گذر" گوییم و اگر ابتدا و انتهای گذری یکی باشد، آن را یک "گذر بسته" می‌نامیم.

فصل دوم

الگوریتم‌های حل بازی‌های بیزی با تعادل استکلبرگ

همان‌طور که در فصل قبل توضیح داده شد، در مدل‌سازی مسئلهٔ امنیت زیرساخت‌ها به کمک نظریهٔ بازی‌ها به دنبال یافتن بهترین استراتژی رهبر در یک بازی بیزی با تعادل استکلبرگ هستیم و اهمیت این موضوع زمانی بیشتر می‌شود که انواع زیادی از پیرو در بازی موجود باشد. در این خصوص کارهایی در گذشته انجام شده که از جملهٔ آن‌ها می‌توان به دو روش "مسائل برنامه‌ریزی خطی چندگانه" و "بیشینه سود تعادل نش بیزی" اشاره کرد که هر دو با استفاده از تبدیل هارسانی ابتدا بازی بیزی را به فرم نرمال تبدیل کرده و سپس تعادل‌های آن را به دست می‌آوردند. اما اشکالاتی که در این دو روش وجود داشت، عاملی برای بهبود الگوریتم‌های حل بازی‌های بیزی با تعادل استکلبرگ شد. اول اینکه، در تبدیل هارسانی، مزیت رهبر بودن در بازی نرمال تغییر یافته در نظر گرفته نمی‌شد. به علاوه، افزایش تعداد نوع‌های پیرو، باعث می‌شود ماتریس فرم نرمال بازی به صورت نمایی بزرگ شده و پیدا کردن تعادل نش از نظر زمانی کاری سخت بود. از این رو، محققان دو الگوریتم جدید با رویکردی تازه را برای حل بازی‌های مورد نظر به دست آوردند که ایرادهای روش‌های قبلی را برطرف

می‌کند و کارایی بهتری دارد. در این فصل، بعد از توصیف دامنهٔ امنیتی‌ای که برای به کار بردن الگوریتم‌ها استفاده می‌شود، دو الگوریتم ASAP [۱] و DOBSS [۲] را به ترتیب معرفی کرده و با ذکر مزایا و معایب هرکدام، به مقایسهٔ آن‌ها با یک‌دیگر و دیگر الگوریتم‌های موجود می‌پردازیم.

۱.۲ توصیف دامنهٔ امنیتی

یادآوری می‌کنیم که در یک محدودهٔ نظارتی، نیروی امنیتی به دلیل کمبود منابع امنیتی و محدودیت‌های زمانی و انرژی برای تأمین سوخت‌رسانی وسایل نقلیه قادر به کنترل همهٔ مناطق در همهٔ زمان‌ها نیست و مجبور است با توجه به عواملی مانند احتمال وقوع جرم، مناطق مختلف را در زمان‌های مختلف نظارت کند. در این شرایط معمولاً سودمند است که نیروهای امنیتی از سیاست گشت تصادفی استفاده کنند تا برای دشمنان غیرقابل پیش‌بینی بوده و به سبب این نااطمینانی از برنامهٔ امنیتی نتوانند به صورت ایمن حملهٔ خود را برنامه‌ریزی کنند. برای نشان دادن کاربرد الگوریتم‌هایی که در بخش‌های آینده معرفی می‌شوند، یک نسخهٔ ساده از دامنهٔ امنیتی مورد نظرمان را به صورت یک بازی بیان می‌کنیم.

ساده‌ترین نسخهٔ بازی ما شامل دو بازیکن، یعنی نیروی امنیتی (رهبر) و دشمن (پیرو)، در محیطی شامل m هدف برای محافظت است. مجموعه استراتژی‌های محض رهبر شامل انتخاب d هدف از بین این m هدف با یک ترتیب مشخص است. نیروی امنیتی می‌تواند از یک استراتژی مخلوط استفاده کند تا برای دشمن غیرقابل پیش‌بینی شود ولی دشمن نیز به مرور زمان از استراتژی مخلوط نیروی امنیتی آگاهی پیدا کرده و سپس یک هدف را برای حمله انتخاب می‌کند. در مدل‌سازی انجام شده، فرض می‌کنیم که دشمن زمانی را صرف حمله به یک هدف می‌کند که در این زمان امکان دستگیری او توسط نیروی امنیتی وجود

[۱] Agent Security via Approximate Policies
[۲] Decomposed Optimal Bayesian Stackelberg Solver

دارد. اگر هدف انتخاب شده توسط دشمن مورد محافظت قرار نگرفته باشد، او با موفقیت به آن هدف حمله می‌کند ولی در صورتی که هدف انتخابی دشمن یکی از اهداف نیروی امنیتی برای محافظت نیز باشد، هرچه این هدف برای نیروی امنیتی اولویت بیشتری داشته باشد و در ترتیب در نظر گرفته شده برای اهداف محافظتی زودتر از بقیهٔ هدف‌ها قرار گرفته باشد، احتمال دستگیری دشمن برای نیروی امنیتی قبل از اینکه با موفقیت حملهٔ خود را به اتمام برساند، بیشتر است. سودهای این بازی را با متغیرهای زیر معرفی می‌کنیم:

- $v_{l,x}$: ارزش هدف l برای نیروی امنیتی
- $v_{l,q}$: ارزش هدف l برای دشمن
- c_x: سود حاصل از دستگیری دشمن برای نیروی امنیتی
- c_q: هزینهٔ دشمن در صورت دستگیری
- p_l: احتمال اینکه نیروی امنیتی بتواند دشمن را در l امین هدف دستگیر کند.

هم‌چنین با توجه به توضیحات گفته شده در مدل داریم: $p_l < p_{l'} \iff l^l < l$
مجموعه استراتژی‌های محض ممکن برای نیروی امنیتی را با X نشان داده که شامل d تایی‌هایی به صورت $i =< w_1, w_2, ..., w_d >$ است که $w_1, ..., w_d = 1, .., m$ و هیچ دو عنصری از آن یکسان نیستند. به عبارت دیگر، نیروی امنیتی با عبور از یک هدف امکان برگشت به آن و نظارت مجدد همان هدف را ندارد. مجموعه استراتژی‌های محض ممکن دشمن را نیز با Q نشان داده که شامل تمام اعداد صحیح $j = 1, ..., m$ است. سودهای نیروی امنیتی و دشمن برای استراتژی‌های محض i, j عبارتند از:

- $-v_{l,x}, v_{l,q}, j = l \notin i.$

$$p_l c_x + (1 - p_i)(-v_v l, x), -p_l c_q + (1 - p_i)(v_{l,q}), j = l \in i. \square$$

با نمادگذاری بالا، می‌توانیم انواع مختلفی از دشمن را که انگیزه‌های متفاوتی دارند، به صورت یک بازی بیزی مدل کنیم مشروط بر اینکه توزیع انواع مختلف دشمن از داده‌های قبلی به دست آمده باشد. در مدل ما، θ_1 مجموعهٔ انواع مختلف نیروی امنیتی و θ_2 مجموعهٔ انواع مختلف دشمن است. از آنجا که تنها یک نوع نیروی امنیتی داریم، θ_1 یک عضو دارد. هم‌چنین، دشمن نوع خودش را می‌داند ولی نیروی امنتی نوع رقیب را نمی‌داند. برای هر بازیکن n، یک مجموعهٔ استراتژی‌های σ_n و یک تابع سود $u_n : \theta_1 \times \theta_2 \times \sigma_1 \times \sigma_2 \to R$ وجود دارد.

۲.۲ الگوریتم ASAP

در این روش که به الگوریتم ASAP معروف است، استراتژی‌های مخلوط ممکن رهبر را به آن دسته از استراتژی‌هایی محدود می‌کنیم که احتمال انتخاب هر عمل مضرب صحیحی از $\frac{1}{k}$ برای یک k از پیش تعیین شده است. بنابراین، ASAP استراتژی‌هایی را فراهم می‌کند که به اصطلاح k - یکنواخت هستند. یک استراتژی مخلوط k - یکنواخت گفته می‌شود اگر توزیع یکنواختی روی یک مجموعه مانند S با اندازهٔ k باشد که اعضای آن ممکن است تکرار شوند. این مجموعه را به اصطلاح یک مجموعهٔ چندگانه گوییم. به عنوان مثال، استراتژی مخلوط متناظر با مجموعهٔ چندگانهٔ $\{1, 2, 3\}$ یک استراتژی است که به عمل اول احتمال $\frac{2}{3}$ و به عمل دوم احتمال $\frac{1}{3}$ را نظیر می‌کند.

یک مزیت الگوریتم ASAP این است که با استفاده از ویژگی مستقل بودن انواع مختلف پیرو به طور مستقیم روی فرم بیزی بازی عمل می‌کند و از تبدیل هارسانی استفاده نمی‌کند. مستقل بودن انواع پیروها باعث دست‌یابی به یک طرح تجزیه در این الگوریتم می‌شود و در نهایت با حل تنها یک مسئلهٔ برنامه‌ریزی خطی الگوریتم جواب مطلوب را به ما می‌دهد.

ابتدا الگوریتم را برای یک نوع پیرو توضیح می‌دهیم. به طور خلاصه، الگوریتم برای این حالت

خاص به ازای یک k مشخص برای هر استراتژی مخلوط رهبر مانند x که متناظر با مجموعهٔ چندگانه‌ای با اندازهٔ k است، سود رهبر از x را در شرایطی که پیرو یک استراتژی با بیش‌ترین سود را انتخاب کرده است، محاسبه می‌کند و در پایان بهترین پاسخ رهبر را مشخص می‌کند. لازم به ذکر است برای دیدن اثبات معادل بودن مسائل برنامه‌ریزی خطی‌ای که در ادامه بیان می‌شوند، می‌توانید مرجع (۲) را نگاه کنید. فرض کنید x و q به ترتیب بردار استراتژی‌های رهبر و پیرو باشند. هم‌چنین، مجموعهٔ استراتژی‌های محض رهبر و پیرو را نیز به ترتیب با X و Q نشان می‌دهیم. به علاوه، ماتریس‌های سود R و C به این صورت تعریف می‌شوند که R_{ij} سود رهبر و C_{ij} سود پیرو است وقتی که رهبر استراتژی محض i و پیرو استراتژی محض j را انتخاب کرده است. اندازهٔ مجموعهٔ چندگانهٔ مورد نیاز الگوریتم را k در نظر بگیرید و یک استراتژی k - یکنواخت مثل x را برای رهبر ثابت فرض کنید. در این صورت مقدار x_i تعداد دفعاتی است که استراتژی محض i در این استراتژی ظاهر شده است که با احتمال $\frac{x_i}{k}$ انتخاب می‌شود. مسئلهٔ برنامه‌ریزی خطی‌ای که پیرو برای یافتن جواب بهینه‌اش به استراتژی x رهبر حل می‌کند، به صورت رابطهٔ (۲) است. تابع هدف این مسئلهٔ برنامه‌ریزی خطی سود انتظاری

$$\begin{aligned} \max \quad & \sum_{j\in Q}\sum_{i\in X}\frac{1}{k}C_{ij}x_i\, q_j \\ \text{s.t.} \quad & \textstyle\sum_{j\in Q} q_j = 1 \\ & q \geq 0. \end{aligned} \tag{2}$$

پیرو از استراتژی x رهبر را بیشینه می‌کند و قیود محدودیت این مسئله استراتژی‌های شدنی q برای پیرو را در نظر می‌گیرند؛ یعنی استراتژی‌هایی که احتمال انتخاب هر استراتژی محض در آن مثبت و جمع احتمال انتخاب استراتژی‌های محض در بردار استراتژی q برابر یک است. دوگان این مسئلهٔ برنامه‌ریزی خطی عبارت است از رابطهٔ (۳). از قضایای کمبود - مکمل ضعیف و جبری KKT می‌دانیم که جواب بهینهٔ مسئلهٔ پیرو معادل سه شرط (PF) [۳] ،

[۳] Primal Feasibility

$$\begin{aligned} \min \quad & a \\ \text{s.t.} \quad & a \geq \sum_{i \in X} \frac{1}{k} C_{ij} x_i \quad j \in Q. \end{aligned}$$
(3)

(DF) [۴] و (WCS) [۵] است که شرط آخر در رابطهٔ (۴) آمده است. این شرایط باید در

$$q_j \left(a - \sum_{i \in X} \frac{1}{k} C_{ij} x_i \right) = 0 \qquad j \in Q.$$
(4)

مسئله‌ای که توسط رهبر حل می‌شود، لحاظ شود تا تنها بهترین پاسخ پیرو طبق الگوریتم در نظر گرفته شود. رهبر به دنبال یک استراتژی k - یکنواخت مثل x است که سود او را بیشینه می‌کند وقتی که پیرو از پاسخ بهینهٔ $q(x)$ استفاده کرده است. بنابرین، رهبر مسئلهٔ (۵) را حل می‌کند. که با اعمال کردن ویژگی‌های استراتژی q_j که وابسته به استراتژی x رهبر است

$$\begin{aligned} \max \quad & \sum_{i \in X} \sum_{j \in Q} \frac{1}{k} R_{ij} q(x)_j \; x_i \\ \text{s.t.} \quad & \textstyle\sum_{i \in X} x_i = k \\ & x_i \in \{0, 1, \ldots, k\}. \end{aligned}$$
(5)

و در نتیجه آن را با $q(x)_j$ نشان می‌دهیم، به مسئلهٔ برنامه‌ریزی خطی (۶) می‌رسیم.

در این مسئله که در آن M یک عدد به قدر کافی بزرگ است، محدودیت‌های اول و چهارم رهبر استراتژی‌های رهبر را به یک استراتژی k - یکنواخت محدود می‌کند و محدودیت‌های دوم و پنجم نیز بیانگر شدنی بودن استراتژی محض پیرو هستند. هم‌چنین، نامساوی‌های سمت چپ و راست در محدودیت سوم به ترتیب از اعمال دو شرط (DF) و (WCS) به دست آمده‌اند.

---

[۴] Dual Feasibility

[۵] Weak Complementary Slackness

$$
\begin{aligned}
\max_{x,q} \quad & \sum_{i\in X}\sum_{j\in Q}\frac{1}{k}R_{ij}x_iq_j \\
\text{s.t.} \quad & \textstyle\sum_i x_i = k \\
& \textstyle\sum_{j\in Q} q_j = 1 \\
& 0 \le (a - \textstyle\sum_{i\in X}\frac{1}{k}C_{ij}x_i) \le (1-q_j)M \\
& x_i \in \{0,1,....,k\} \\
& q_j \in \{0,1\}.
\end{aligned}
\tag{6}
$$

اکنون برای آسان‌تر حل کردن این مسئله نیاز به تبدیل آن به یک مسئلهٔ برنامه‌ریزی خطی داریم. برای خطی‌سازی آن از تغییر متغیر $z_ij = x_iq_j$ استفاده می‌کنیم و به مسئلهٔ (۷) می‌رسیم.

برای تعمیم الگوریتم به بیش از یک نوع دشمن، فرض کنید q^l بردار استراتژی نوع l ام پیرو

$$
\begin{aligned}
\max_{q,z} \quad & \textstyle\sum_{i\in X}\sum_{j\in Q}\frac{1}{k}R_{ij}z_{ij} \\
\text{s.t.} \quad & \textstyle\sum_{i\in X}\sum_{j\in Q} z_{ij} = k \\
& \textstyle\sum_{j\in Q} z_{ij} \le k \\
& kq_j \le \textstyle\sum_{i\in X} z_{ij} \le k \\
& \textstyle\sum_{j\in Q} q_j = 1 \\
& 0 \le (a - \textstyle\sum_{i\in X}\frac{1}{k}C_{ij}(\sum_{h\in Q} z_{ih})) \le (1-q_j)M \\
& z_{ij} \in \{0,1,....,k\} \\
& q_j \in \{0,1\}
\end{aligned}
\tag{7}
$$

باشد و L مجموعهٔ انواع پیرو باشد. هم‌چنین ماتریس‌های سود رهبر و نوع l ام پیرو در تقابل با یک‌دیگر را به ترتیب با R^l و C^l نشان می‌دهیم. با نمادگذاری صورت گرفته و مشابه آنچه برای یک نوع پیرو داشتیم، شرایط بهینگی حاصل برای بهترین پاسخ پیرو نوع l که رهبر آن را در مسئلهٔ خود لحاظ می‌کند، عبارت است از رابطهٔ (۸)

فرض کنید رهبر از استراتژی k - یکنواخت استفاده می‌کند و هم‌چنین احتمال رو به رو شدن با دشمن نوع l ام برابر p^l باشد. در این صورت در حالت کلی رهبر مسئلهٔ (۹) را حل می‌کند.

$$\begin{aligned} \sum_{j \in Q} q_j^l &= 1 \\ a^l - \sum_{i \in X} \frac{1}{k} C_{ij}^l x_i &\geq 0 \\ q_j^l (a^l - \sum_{i \in X} \frac{1}{k} C_{ij}^l x_i) &= 0 \\ q_j^l &\geq 0 \end{aligned} \tag{8}$$

که قیود محدودیت آن مشابه حالت یک نوع پیرو هستند. به علاوه، به طور مشابه با استفاده

$$\begin{aligned} \max_{x,q} \quad & \sum_{i \in X} \sum_{l \in L} \sum_{j \in Q} \frac{p^l}{k} R_{ij}^l x_i q_j^l \\ \text{s.t.} \quad & \textstyle\sum_i x_i = k \\ & \textstyle\sum_{j \in Q} q_j^l = 1 \\ & 0 \leq (a^l - \textstyle\sum_{i \in X} \frac{1}{k} C_{ij}^l x_i) \leq (1 - q_j^l) M \\ & x_i \in \{0, 1, ..., k\} \\ & q_j^l \in \{0, 1\}. \end{aligned} \tag{9}$$

از تغییر متغیر $z_{ij}^l = x_i q_j^l$ می‌توانیم به مسئلهٔ برنامه‌ریزی خطی (۱۰) برسیم که به وسیلهٔ ابزارهای موجود برای حل مسائل با تعداد متغیرهای بالا قادر به یافتن استراتژی بهینهٔ رهبر هستیم.

۳.۲ الگوریتم DOBSS

اگرچه الگوریتم ASAP باعث ایجاد تحولی در یافتن جواب‌های بازی‌های بیزی با تعادل استکلبرگ بود و به دلیل عدم استفاده از تبدیل هارسانی سرعت بیشتری در مقایسه با روش‌های قبلی داشت، اما آزمایش‌های محاسباتی نشان می‌داد که این روش به مشکلاتی

$$
\begin{array}{ll}
\max_{q,z} & \sum_{i\in X}\sum_{l\in L}\sum_{j\in Q}\frac{p^l}{k}R^l_{ij}z^l_{ij} \\
\text{s.t.} & \sum_{i\in X}\sum_{j\in Q}z^l_{ij}=k \\
& \sum_{j\in Q}z^l_{ij}\le k \\
& kq^l_j\le\sum_{i\in X}z^l_{ij}\le k \\
& \sum_{j\in Q}q^l_j=1 \\
& 0\le(a^l-\sum_{i\in X}\frac{1}{k}C^l_{ij}(\sum_{h\in Q}z^l_{ih}))\le(1-q^l_j)M \\
& \sum_{j\in Q}z^l_{ij}=\sum_{j\in Q}z^1_{ij} \\
& z^l_{ij}\in\{0,1,....,k\} \\
& q^l_j\in\{0,1\}
\end{array}
$$

(10)

در یافتن جواب‌های شدنی در ابعاد بالا برخورد می‌کند. از این رو، محققان در جهت بهبود کارایی و هم‌چنین افزایش سرعت، الگوریتم جدیدی به نام DOBSS را معرفی کردند که از آن در سیستم ARMOR که از سال ۲۰۰۸ در فرودگاه بین‌المللی لس‌آنجلس جهت کمک به نیروی امنیتی این فرودگاه مستقر شده است، استفاده می‌شود. در ادامه به معرفی این الگوریتم می‌پردازیم که همانند الگوریتم ASAP به طور مستقیم روی بازی بیزی کار می‌کند و با در نظر گرفتن مزیت رهبر بودن، استراتژی‌های بهینهٔ رهبر را پیدا می‌کند.

همانند بخش قبل، ابتدا الگوریتم را برای حالتی که یک نوع پیرو داشته باشیم، توضیح داده و سپس آن را به حالت کلی تعمیم می‌دهیم. لازم به ذکر است که سناریوی بازی در نظر گرفته برای این الگوریتم و نمادگذاری صورت گرفته در روابط پیش رو دقیقاً همانی هستند که برای الگوریتم ASAP داشتیم. بدین ترتیب، مسئلهٔ بهینه‌سازی حل شده توسط یک نوع پیرو برای یافتن چواب بهینهٔ خودش عبارت است از رابطهٔ (۱۱)

که مجدداً با به دست آوردن دوگان این مسئلهٔ برنامه‌ریزی خطی و اعمال شرایط (PF)،

$$
\begin{array}{ll}
\max_{q} & \sum_{j\in Q}\sum_{i\in X}C_{ij}x_i\,q_j \\
\text{s.t.} & \sum_{j\in Q}q_j=1 \\
& q_j\ge 0.
\end{array}
$$

(11)

(DF) و (WCS) در مسئلهٔ رهبر، مسئله‌ای که او برای یافتن استراتژی بهینهٔ خودش حل می‌کند، به صورت رابطهٔ (۱۲) است.

برای توسعهٔ این روش به حالت کلی، از همان نمادگذاری بخش قبل برای مشخص کردن

$$\begin{aligned} \max_{x,q,a} \quad & \sum_{i\in X}\sum_{j\in Q} R_{ij}x_i q_j \\ \text{s.t.} \quad & \sum_{i\in X} x_i = 1 \\ & \sum_{j\in Q} q_j = 1 \\ & 0 \le (a - \sum_{i\in X} C_{ij}x_i) \le (1-q_j)M \\ & x_i \in [0\dots 1] \\ & q_j \in \{0,1\} \end{aligned} \tag{12}$$

مجموعهٔ انواع پیرو و احتمال رو به رو شدن با هر نوع پیرو استفاده می‌کنیم. بنابراین رهبر در رقابت با هر نوع پیرو مسئله‌ای را با شرایط بهینگی زیر که از جواب بهینهٔ این نوع پیرو، یعنی رابطهٔ (۱۳) به دست آمده است، حل می‌کند.

$$\begin{aligned} \sum_{j\in Q} q_j^l &= 1 \\ a^l - \sum_{i\in X} C_{ij}^l x_i &\ge 0 \\ q_j^l(a^l - \sum_{i\in X} C_{ij}^l x_i) &= 0 \\ q_j^l &\ge 0 \end{aligned} \tag{13}$$

و در حالت کلی، سود انتظاری رهبر از مقابله با یک پیرو با انواع مختلف از بیشینه‌سازی مسئلهٔ (۱۴) حاصل می‌شود.

که با استفاده از همان تغییر متغیر بخش قبل، می‌توانیم آن را به مسئلهٔ برنامه‌ریزی خطی (۱۵) تبدیل کنیم.

$$
\begin{aligned}
\max_{x,q,a} \quad & \sum_{i\in X}\sum_{l\in L}\sum_{j\in Q} p^l R^l_{ij} x_i q^l_j \\
\text{s.t.} \quad & \textstyle\sum_i x_i = 1 \\
& \textstyle\sum_{j\in Q} q^l_j = 1 \\
& 0 \le (a^l - \textstyle\sum_{i\in X} C^l_{ij} x_i) \le (1 - q^l_j)M \\
& x_i \in [0 \dots 1] \\
& q^l_j \in \{0, 1\}
\end{aligned}
\tag{14}
$$

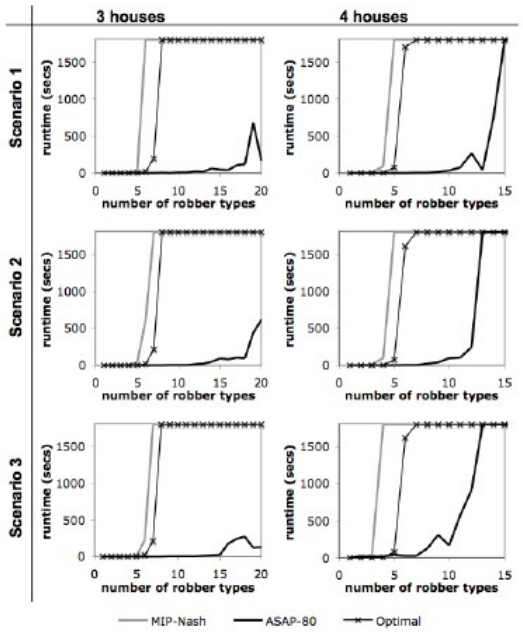
$$
\begin{aligned}
\max_{q,z,a} \quad & \textstyle\sum_{i\in X}\sum_{l\in L}\sum_{j\in Q} p^l R^l_{ij} z^l_{ij} \\
\text{s.t.} \quad & \textstyle\sum_{i\in X}\sum_{j\in Q} z^l_{ij} = 1 \\
& \textstyle\sum_{j\in Q} z^l_{ij} \le 1 \\
& q^l_j \le \textstyle\sum_{i\in X} z^l_{ij} \le 1 \\
& \textstyle\sum_{j\in Q} q^l_j = 1 \\
& 0 \le (a^l - \textstyle\sum_{i\in X} C^l_{ij}(\sum_{h\in Q} z^l_{ih})) \le (1 - q^l_j)M \\
& \textstyle\sum_{j\in Q} z^l_{ij} = \sum_{j\in Q} z^1_{ij} \\
& z^l_{ij} \in [0 \dots 1] \\
& q^l_j \in \{0, 1\}
\end{aligned}
\tag{15}
$$

لازم به ذکر است همانند بخش قبل، برای دیدن اثبات معادل بودن مسائل برنامه‌ریزی خطی‌ای که در در این بخش بیان شدند، می‌توانید مرجع (۱) را نگاه کنید.

۴.۲ نتایج تجربی

در این بخش، ابتدا نتایج آزمایشات صورت گرفته روی روش ASAP و سایر روش‌هایی که از کارهای قبلی به دست آمده بودند را می‌بینیم و به مقایسهٔ آن‌ها می‌پردازیم. چهار روشی که مورد بررسی قرار گرفته‌اند، گشت تصادفی یکنواخت، ASAP، برنامه‌ریزی خطی و بیشینه سود تعادل نش بیزی هستند. تمامی آزمایشات روی ۳ و ۴ هدف با در نظر گرفتن دامنه‌ای امنیتی شامل ۲ هدف انجام گرفته‌اند. به علاوه، تمامی بازی‌ها برای هر نوع دشمن به گونه‌ای

نرمال شده‌اند که کمینه و بیشینه سود بازیکنان به ترتیب ۰ و ۱ باشد.
نمودارهای ۱.۲ الگوریتم‌های مختلف را از نظر زمان اجرا مورد بررسی قرار داده است. هم‌چنین اندازهٔ مجموعهٔ چندگانهٔ مورد استفاده در الگوریتم ASAP برابر ۸۰ در نظر گرفته شده است و تمامی آزمایشاتی که در ۳۰ دقیقه به جواب نرسیدند، قطع شده‌اند. همان‌طور که مشاهده می‌شود، روش (۳) به ازای ۳ و ۴ هدف به ترتیب بیش از ۷ و ۶ نوع دشمن را نمی‌تواند حل کند و روش (۴) حتی برای تعداد کمتری در شرایط مشابه نسبت به روش (۳) جوابگو نیست. اما روش (۲) برای حداقل ۲۰ و ۱۲ نوع دشمن به ازای به ترتیب ۳ و ۴ هدف کارآمد است.
نمودارهای ۲.۲ سود الگوریتم‌های مختلف را نشان می‌دهند که روش (۱) همواره پایین‌ترین

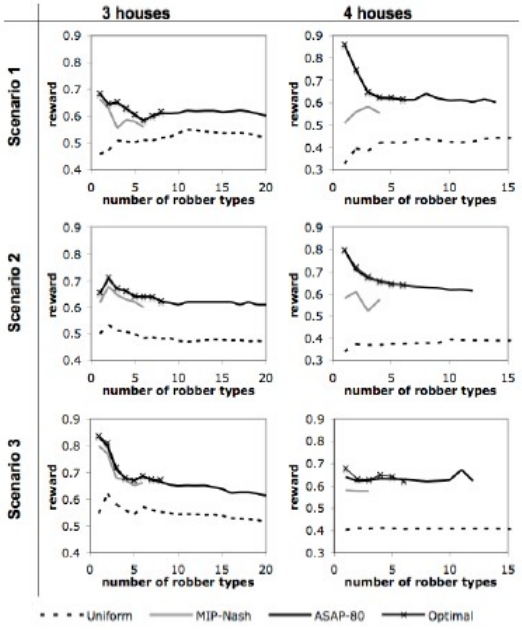


شکل ۱.۲: بررسی الگوریتم‌ها از نظر زمان اجرا.

میزان سود را دارد و روش‌های (۲) و (۳) برای تعداد کم انواع دشمن بسیار نزدیک به هم عمل می‌کنند. روش (۴) نیز اگرچه از (۱) بهتر است ولی نسبت به روش (۲) کارایی کمتری دارد و این اختلاف سود از آنجا ناشی می‌شود که در روش (۴) تنها تعادل نش بازی بیزی

بدون توجه به ترتیب بازی بازیکنان محاسبه می‌شد.

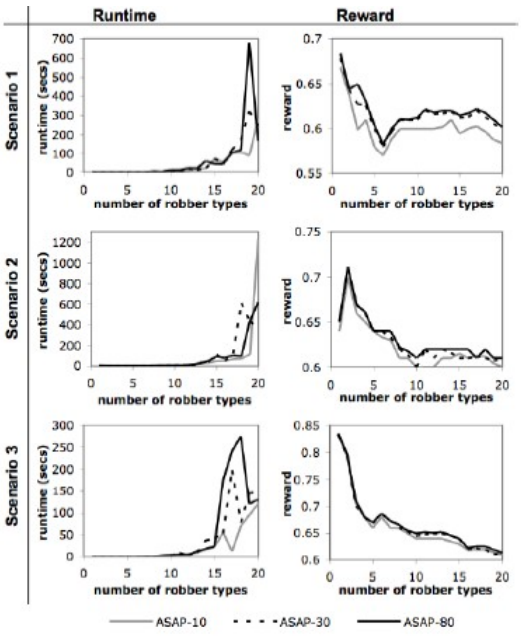
و نمودارهای ۳.۲ منحصراً به سود حاصل از الگوریتم ASAP با توجه به اندازهٔ مجموعهٔ



شکل ۲.۲: بررسی الگوریتم‌ها از نظر سود.

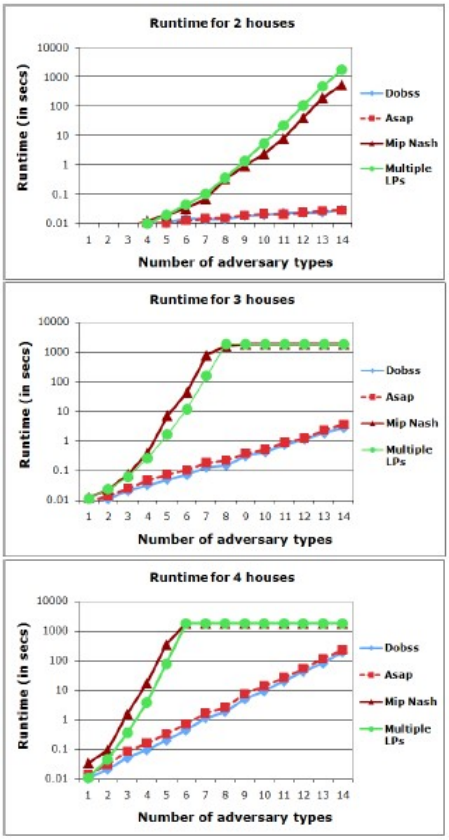
چندگانه می‌پردازد. به طور کلی، هرچه اندازهٔ این مجموعه افزایش پیدا کند، زمان اجرا و سود هر دو افزایش می‌یابند. به علاوه، با افزایش اندازهٔ مجموعهٔ چندگانه افزایش سود ناچیزی خواهیم داشت به طوری که به عنوان مثال، سود حاصل برای مجموعه‌ای با ۱۰ عضو به اندازهٔ ۹۶ درصد سود مجموعه‌ای با اندازهٔ ۸۰ است. زمان اجرا نیز همواره با افزایش اندازهٔ مجموعهٔ چندگانه بیشتر نمی‌شود؛ مثلاً برای سناریوی دوم، با ۲۰ نوع دشمن، $k = 10$ در ۱۲۱۸ ثانیه به نتیجه می‌رسد ولی $k = 80$ در ۶۱۷ ثانیه به نتیجه می‌رسد.

حال الگوریتم DOBSS را هم به مقایسهٔ انجام شده اضافه می‌کنیم. سه نمودار ۴.۲ زمان اجرای چهار الگوریتم نام برده به ترتیب برای ۲، ۳ و ۴ هدف را نشان می‌دهد که نتیجه می‌شود الگوریتم DOBSS از نظر زمان اجرا اندکی بهتر از الگوریتم ASAP عمل می‌کند و



شکل ۳.۲: بررسی الگوریتم ASAP بر اساس اندازهٔ مجموعهٔ چندگانه

دلیل این امر در نظر نگرفتن جواب‌های ناشدنی در الگوریتم DOBSS است. در جدول ۵.۲



شکل ۴.۲: مقایسه با الگوریتم DOBSS.

می‌توانیم تعداد جواب‌های نشدنی تولید شده توسط الگوریتم ASAP را مشاهده کنیم که در آن ستون‌ها نشان‌دهندهٔ تعداد انواع دشمن و سطرها بیانگر تعداد هدف‌های محافظتی برای نیروی امنیتی هستند.

	2	3	4	5	6	7
1	0	0	0	0	0	10
2	0	0	0	0	5	0
3	0	5	5	15	5	10
4	0	10	15	20	20	20
5	0	10	10	10	5	10
6	0	10	10	10	15	15
7	0	15	20	15	10	20
8	0	5	15	10	30	45
9	0	5	10	15	20	30*
10	5	10	20	5	35*	35*
11	0	5	20	10	35*	40*
12	0	10	20	10	30*	30*
13	0	20	20	25*	25*	0
14	0	20	20	30*	20*	20*

شکل ۵.۲: تعداد جواب‌های ناشدنی تولید شده توسط الگوریتم ASAP.

فصل سوم

کاربرد نظریهٔ بازی‌ها در امنیت فرودگاه‌ها

در این فصل به کاربرد نظریهٔ بازی‌ها در برنامهٔ امنیتی فرودگاه‌ها با استفاده از مدل بازی‌های بیزی با تعادل استکلبرگ می‌پردازیم. بدین منظور یکی از مثال‌های عینی موفق در این زمینه یعنی فرودگاه بین‌المللی لس‌آنجلس را مطرح کرده و کارکرد نرم‌افزار ARMOR را در این خصوص بررسی می‌کنیم.

۱.۳ دامنهٔ امنیتی فرودگاه

فرودگاه لس‌آنجلس یکی از شلوغ‌ترین و پر رفت و آمدترین فرودگاه‌های آمریکا است که به طور متوسط سالانه ۶۰-۷۰ میلیون مسافر از آن استفاده می‌کنند. از آنجا که این فرودگاه به یکی از اهداف تروریست‌هایی که قصد حمله به آن را داشتند، تبدیل شده بود، پلیس فرودگاه یک سیستم امنیتی طراحی کرده بود که از چند حلقهٔ محافظتی شامل بازرسی وسایل نقلیه، واحد گشت پلیس برای جاده‌های منتهی به فرودگاه و واحد گشت سگ‌ها برای

داخل فرودگاه و هم‌چنین بررسی مسافران و کیف آن‌ها تشکیل می‌شد. اما از یک طرف، منابع کافی (نیروهای پلیس) برای نظارت تمام اتفاقات فرودگاه وجود نداشت و از طرف دیگر، با توجه به اندازهٔ فرودگاه و تعداد مسافران آن، این سطح از نظارت باعث تأخیر برای مسافران می‌شد. بنابراین، در ابتدا تنها تعدادی از ایست‌های بازرسی و پایانه‌ها را برای نظارت انتخاب کردند ولی چالش پیش روی نیروی امنیتی فرودگاه این بود که فعال کردن برخی از ایست‌های بازرسی در دسترس و واحد سگ‌ها با یک برنامهٔ زمانی مشخص و قطعی به دشمنان این اجازه را می‌داد تا با الگو برداری از مشاهدهٔ برنامهٔ زمانی، حملهٔ خود را به گونه‌ای طراحی کنند که از نظارت پلیس در امان مانده و برنامه‌های نظارتی قطعی را ناکارآمد کنند. در نتیجه، پلیس فرودگاه به راه حل تصادفی‌سازی رسید و به طور خاص، از میان همهٔ اقدامات امنیتی که می‌تواند تصادفی‌سازی برایشان اعمال شود، پلیس دو مطلب مهم را مطرح کرد. اول اینکه، با توجه به تعداد زیاد جاده‌های منتهی به فرودگاه، در چه مکان‌هایی و زمان‌هایی باید ایست‌های بازرسی را برای بررسی ماشین‌ها قرار دهند. برای مثال، اگر قرار است دو تا ایست بازرسی برای بازهٔ زمانی ۸ تا ۱۱ صبح قرار داده شود، یک برنامهٔ پیشنهادی به پلیس ممکن است این باشد که در روز دوشنبه، ایست‌های بازرسی در مسیرهای اول و دوم قرار بگیرند اما برای روز سه‌شنبه و در همان بازهٔ زمانی، باید در مسیرهای اول و سوم قرار بگیرند. دوم اینکه، به یک برنامه برای واحد گشت سگ‌ها در پایانه‌های داخل فرودگاه دست پیدا کند. به عنوان نمونه، اگر سه واحد گشت سگ در دسترس است، یک برنامه می‌تواند این باشد که در روز اول این واحد در پایانه‌های ۱، ۳ و ۶ قرار بگیرند و در روز دوم، در پایانه‌های ۲، ۴ و ۶ استقرار پیدا کنند. مشکلات پلیس فرودگاه در خصوص برقراری امنیت این فرودگاه، سه چالش مهم را نشان می‌داد: (۱) حمله‌کننده‌های بالقوه می‌توانند با علم به برنامه‌های زمانی نیروهای امنیتی به مرور زمان، استراتژی حملهٔ خود را انتخاب کنند. (۲) اطلاعات ناشناخته و نامشخصی از انواع دشمنانی که ممکن است با آن‌ها رو به رو شود، وجود دارد. (۳) اگرچه تصادفی‌سازی به حذف الگوهای قطعی کمک می‌کرد اما باید هزینه‌ها و سودهای یک هدف خاص نیز در گرفته می‌شد. از همین رو، مسئلهٔ تصمیم‌گیری قرار دادن ایست‌های بازرسی یا واحد گشت

سگ‌ها در فرودگاه لس‌آنجلس را به عنوان یک بازی بیزی با تعادل استکلبرگ مدل می‌کنیم.

۲.۳ مدل‌سازی

برای مدل‌سازی مسئلهٔ امنیت فرودگاه بین‌المللی لس‌آنجلس به عنوان یک بازی بیزی با تعادل استکلبرگ، تنها روی قرار دادن ایست‌های بازرسی تمرکز می‌کنیم و برای واحد گشت سگ‌ها به طور مشابه قابل بیان است.

بازی ما دو بازیکن، یعنی پلیس فرودگاه (رهبر) و دشمنان (پیرو)، دارد و در محیطی شامل k مسیر ورودی به فرودگاه قرار داریم. مجموعه استراتژی‌های محض پلیس فرودگاه یک زیرمجموعه از k مسیر در دسترس است و مجموعه استراتژی‌های محض دشمن انتخاب یک مسیر از بین k مسیر برای حمله است. پلیس فرودگاه می‌تواند یک استراتژی مخلوط را انتخاب کند تا دشمنان دارای عدم قطعیت نسبت به محل قرار گرفتن ایست‌های بازرسی باشند، هرچند که دشمن از استراتژی مخلوط پلیس آگاهی پیدا خواهد کرد. فرض می‌کنیم m نوع دشمن داشته باشیم که هرکدام برنامه‌ها و قابلیت‌های حملهٔ متفاوتی با یک‌دیگر دارند. چون هر نوع دشمن بعد از مشاهدهٔ برنامهٔ امنیتی پلیس اقدام به انتخاب هدف حملهٔ خود می‌کند، مدل‌سازی با بازی استکلبرگ به رهبریت پلیس فرودگاه انجام می‌شود. در این حالت فرض کنید X مجموعه عمل‌های ممکن برای پلیس باشد که شامل ترکیبی از انتخاب‌های چندتایی ایست‌های بازرسی است. برای مثال، اگر پلیس قصد داشته باشد تنها یک ایست بازرسی قرار دهد، آنگاه $X = \{1, ..., k\}$ است ولی اگر قصد داشته باشد که دو تا ایست بازرسی قرار دهد، $X = \{(1,2), (1,3), ..., (k-1,k)\}$ است. هر نوع $l \in L\{1, ..., m\}$ دشمن می‌تواند تصمیم بگیرد که کدام k جاده را برای حمله انتخاب کند و یا هیچ‌کدام را انتخاب نکند. لذا، مجموعه عمل‌های او دارای $k+1$ عضو است که با Q نشان می‌دهیم. حال اگر پلیس فرودگاه مسیر i ام را برای محافظت و دشمن نوع l ام مسیر j ام را برای حمله انتخاب کند، سود پلیس برابر R^l_ij و سود دشمن برابر C^l_ij خواهد بود که مقادیر این

سودها بر اساس قابلیت دشمن در حمله، آسیبی که می‌تواند وارد کند و احتمال دستگیری او متفاوت است.

با مدل‌سازی بالا، ورودی‌های الگوریتم $DOBSS$ را در اختیار داریم و رهبر می‌تواند به کمک این الگوریتم که در فصل قبل معرفی شد، بیشینه سود خود را محاسبه کرده و استراتژی بهینهٔ خود را برای قرار دادن ایست‌های بازرسی پیدا کند. این الگوریتم در طراحی نرم‌افزار ARMOR به کار گرفته است که در بخش‌های بعدی به معرفی آن و سنجش میزان کارایی‌اش می‌پردازیم.

۳.۳ نرم‌افزار $ARMOR$

دو نسخهٔ متفاوت برای ARMOR [۱] وجود دارد که یکی برای ایست‌های بازرسی و دیگری برای واحد گشت سگ‌ها استفاده می‌شود. هر دو از نظر ساختار و عملکرد یکسان هستند و تنها در ورودی برنامه تفاوت دارند؛ لذا در ادامه روی نسخه‌ای از ARMOR که برای ایست‌های بازرسی استفاده می‌شود، تمرکز می‌کنیم. این نرم‌افزار از چهر مؤلفهٔ مهم تشکیل شده است: (۱) یک بخش قابل مشاهده برای کاربر که نقش رابط را دارد. (۲) یک روش برای ساختن ماتریس‌های بازی بیزی با تعادل استکلبرگ که در الگوریتم DOBSS استفاده می‌شود. (۳) نحوهٔ پیاده‌سازی الگوریتم DOBSS و (۴) روشی برای ساخت برنامهٔ پیشنهادی به کاربر. هم‌چنین، این برنامه دو دسته اطلاعات به عنوان ورودی دریافت می‌کند. یک دسته اطلاعات مستقیمی است که کاربر از طریق رابط وارد می‌کند و دستهٔ دیگر اطلاعات مرتبط با ایست‌های بازرسی یا واحد گشت سگ‌ها است که می‌تواند روی اقدامات نیروی امنیتی تأثیرگذار باشد؛ مثل حجم ترافیک مسافران فرودگاه در یک روز خاص.

در بخش رابط کاربر برنامه، کاربر با دادن اطلاعات ورودی فضای عمل برنامه را تعریف می‌کند. این اطلاعات شامل تعداد ایست‌های بازرسی قابل دسترس در یک بازهٔ زمانی مشخص، زمان

[۱] assistant for randomized monitoring over routes

شروع و پایان هر بازهٔ زمانی و تعداد روزهایی که قرار است برای آن‌ها برنامه‌ریزی صورت بگیرد، می‌شود. به ازای هر بازهٔ زمانی داده شده، سیستم یک بازی جدید طراحی می‌کند. با توجه به اطلاعاتی نظیر تعداد مسیرهای ورودی و تعداد ایست‌های بازرسی در دسترس، فضای عمل بازی تعریف می‌شود و سیستم می‌تواند یک بازی بیزی با تعادل استکلبرگ را طراحی کرده و با حل آن، برنامهٔ پیشنهادی را خروجی دهد. این برنامهٔ پیشنهادی خود از سه بخش تشکیل شده است. (۱) ایست‌های بازرسی اجباری: نشان‌دهندهٔ ایست‌های بازرسی‌ای است که حتماً باید در یک روز و زمان مشخص فعال باشند. (۲) ایست‌های بازرسی ممنوعه: نشان‌دهندهٔ آن دسته از ایست‌های بازرسی است که نباید از آن‌ها استفاده شود. (۳) ایست‌های بازرسی کمینه: ایست‌های بازرسی‌ای را نشان می‌دهد که حداقل در یکی از بازه‌های زمانی داده شده باید فعال باشند.

در بخش تولید ماتریس‌های الگوریتم DOBSS با توجه به اطلاعاتی که کاربر وارد کرده است، برای هر نوع دشمن یک ماتریس بازی بیزی با تعادل استکلبرگ ساخته و به بخش پیاده‌سازی الگوریتم فرستاده می‌شود تا استراتژی مخلوط بهینهٔ نیروی امنیتی محاسبه شود. برای نشان دادن این فرایند، فرض کنید سه ایست بازرسی A، B و C در دسترس است و برای یک بازهٔ زمانی مشخص می‌خواهیم دو تا را فعال کنیم. هم‌چنین فرض کنید که دو ایست بازرسی A و B کارایی بالایی دارند در حالی که C کارایی پایینی دارد. بر این اساس، استراتژی مخلوط تولید شده توسط الگوریتم DOBSS احتمال بالایی را به استفادهٔ هم‌زمان از ایست‌های بازرسی A و B نظیر می‌کند و دو ترکیب دوتایی دیگر از این سه ایست بازرسی با احتمال کمتری مورد استفاده قرار می‌گیرند؛ به عبارت دیگر، در زمان‌های کمتری فعال خواهند بود. برای مثال، استفاده از (A, B)، (A, C) و B, C به ترتیب ۷۰، ۱۵ و ۱۵ درصد خواهد بود. بعد از اینکه کاربر برنامهٔ پیشنهادی را مشاهده کرد، می‌تواند در صورت نیاز به طور دستی تغییراتی را در برنامه اعمال کند؛ زیرا ممکن است در یک زمانی استفاده از ایست بازرسی A امکان‌پذیر نباشد در حالی که بیشترین کارایی را دارد. کاربر می‌تواند این محدودیت را به طور دستی به نرم‌افزار بدهد. در این حالت، نرم‌افزار با نظیر

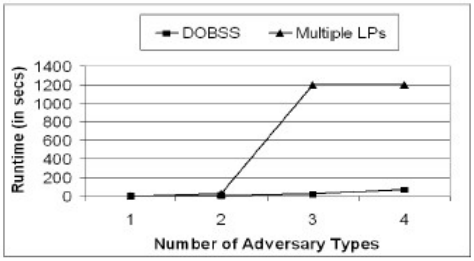
کردن احتمال صفر به ایست بازرسی A یک برنامهٔ جایگزین پیشنهاد می‌دهد و در ضمن آن، به کاربر هشدار می‌دهد که برنامهٔ جایگزین سود کمتری دارد. در بخش بعدی، میزان کارایی نرم‌افزار ARMOR از نظر زمان اجرا و کیفیت جواب‌های تولیدی بررسی می‌کنیم.

۴.۳ نتایج تجربی

اگرچه در فصل قبل، به مقایسهٔ الگوریتم DOBSS و سایر رقیب‌هایش پرداختیم و بهتر بودن آن را از نظر زمان اجرا و سود نشان دادیم، اما در اینجا بار دیگر برای دامنهٔ امنیتی مورد نظرمان، یعنی فرودگاه بین‌المللی لس‌آنجلس نتایج آزمایشات صورت گرفته روی الگوریتم‌های مختلف را بررسی می‌کنیم.

نمودار ۱.۳ الگوریتم DOBSS را با روش برنامه‌ریزی خطی از نظر زمان اجرای الگوریتم مقایسه می‌کند. همان‌طور که مشاهده می‌شود، برای ۱ و ۲ نوع دشمن این دو الگوریتم بسیار نزدیک به هم عمل می‌کنند اما برای تعداد بیش از ۲ نوع دشمن الگوریتم DOBSS به مراتب سریع‌تر است به طوری که برای ۴ نوع دشمن در کمتر از ۱۰۰ ثانیه جواب می‌دهد در حالی که الگوریتم Multiple LPs به بیش از ۱۲۰۰ ثانیه زمان نیاز برای یافتن جواب دارد.

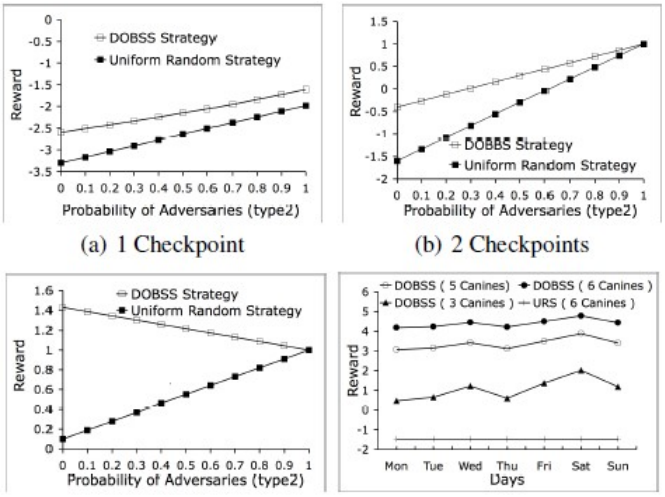
نمودارهای ۲.۳ به بررسی دو الگوریتم ARMOR و تکنیک تصادفی‌سازی یکنواخت و



شکل ۱.۳: مقایسهٔ DOBSS با روش برنامه‌ریزی خطی از نظر زمان اجرا.

مقایسهٔ آن‌ها با هم از نظر میزان سود می‌پردازد. تمامی آزمایشات این قسمت بر روی دو

نوع دشمن انجام شده و محور افقی نشان‌دهندهٔ احتمال مواجه شدن با نوع دوم دشمن است. هم‌چنین، محور عمودی سود حاصل برای پلیس فرودگاه را نشان می‌دهد. با توجه به نمودارها، برای تعداد ۱ تا ۳ ایست بازرسی همواره استراتژی DOBSS سود بیشتری به همراه دارد. هم‌چنین، برای واحد گشت سگ‌ها نیز استفاده از ۶ و یا حتی ۳ واحد گشت سگ که توسط الگوریتم DOBSS تصادفی‌سازی شده‌اند، از ۶ واحد گشت سگ به طور یکنواخت تصادفی‌سازی شده‌اند، سود به مراتب بیشتری دارد.



شکل ۲.۳: بررسی الگوریتم‌ها از نظر سود.

فصل چهارم

کاربرد نظریهٔ بازی‌ها در امنیت بنادر

در این فصل به کاربرد نظریهٔ بازی‌ها در امنیت بنادر و سواحل می‌پردازیم و بدین منظور، سیستم PROTECT [1] را معرفی می‌کنیم که به گارد ساحلی آمریکا برای محافظت از بندر بوستون در حال استفاده است. سیستم PROTECT از الگوریتمی به نام PASAQ استفاده می‌کند که مبنای آن، ساختار بازی استکلبرگ (مهاجم - مدافع) است و گارد ساحلی به عنوان مدافع در مقابل دشمنانی قرار دارد که قبل از آغاز هر حمله می‌توانند برنامهٔ گشت نظارتی را مشاهده کنند. جواب PROTECT به دست آوردن یک استراتژی مخلوط یا یک الگوی نظارتی تصادفی با توجه به اهداف دشمن و واکنش پیش‌بینی شدهٔ دشمنان به گشت ساحلی آمریکا است. به علاوه، این سیستم بر اساس مدل QR رفتار دشمنان را مدل می‌کند و فرض عقلانیت کامل دشمنان را ندارد که با دنیای واقعی سازگارتر است.

[1] port tesilience operational/tactical enforcement to combat terrorism

۱.۴ مدل بازی

برای مدل کردن محدودهٔ امنیتی گارد ساحلی بندر به عنوان یک بازی استکلبرگ، نیاز به تعریف مجموعه استراتژی‌های مهاجم، مدافع و سود اهداف موجود در بندر داریم. همان‌طور که قبلاً گفته شد، در بازی استکلبرگ مهاجم با آگاهی از استراتژی مدافع می‌تواند حملهٔ خود را برنامه‌ریزی کند؛ بنابراین تمام هدف‌های ممکنی که دشمن می‌تواند به آن‌ها حمله کند، به عنوان استراتژی‌های محض او در نظر می‌گیریم. برای تعریف مجموعه استراتژی‌های گارد ساحلی، ابتدا اهداف مهم بندر که نزدیک به یک‌دیگر هستند را در یک دسته قرار می‌دهیم و به عنوان یک موقعیت جغرافیایی در نظر می‌گیریم که هدف گارد ساحلی محافظت از موقعیت‌های جغرافیایی مختلف شامل اهداف گوناگون است. به علاوه، فرض می‌کنیم که گارد ساحلی می‌تواند برای هر موقعیت جغرافیایی یک فعالیت امنیتی مخصوصی در نظر بگیرد که این فعالیت‌های امنیتی از نظر مدت زمانی که طول می‌کشند، با یک‌دیگر متفاوت هستند. در این صورت، برای آن دسته از فعالیت‌هایی که برای مدافع رمان زیادتری می‌خواهد، سود بیشتری در نظر می‌گیریم. بی‌تفاوت نبودن نسبت به زمان فعالیت‌های امنیتی از آن جهت اهمیت دارد که ممکن است در یک برنامهٔ امنیتی، یک واحد گشت از موقعیت‌های جغرافیایی کمتری محافظت کند ولی زمان بیشتری را به فعالیت‌های امنیتی اختصاص دهد و در مقابل، یک واحد گشت مکان‌های بیشتری را در مدت زمان کمتری نظارت کند. با این توضیحات، گراف $G = (V, E)$ را به این صورت تعریف می‌کنیم که مجموعه رئوس V مکان‌هایی هستند که باید از آن‌ها محافظت شود و مجموعه یال‌های E شامل تمام مسیرهای موجود بین هر دو موقعیت جغرافیایی است. برنامهٔ پیشنهادی PROTECT برای محافظت از رئوس گراف، یک گشت بسته از G است که ابتدا و انتهای آن، موقعیت جغرافیایی $b \in V$ است که آن را مکان امنیتی پایه برای گارد ساحلی می‌نامیم. هم‌چنین، به ازای هر رأس گراف، یک فعالیت امنیتی مشخص با حداکثر زمان t به آن نظیر می‌شود. (لازم به ذکر است که حتی گذر از یک موقعیت جغرافیایی نیز به عنوان یک فعالیت امنیتی در نظر گرفته

می‌شود.) توجه کنید که گارد ساحلی در یک برنامهٔ امنیتی می‌تواند چند بار از یک موقعیت جغرافیایی بازدید کند. برای مثال، نظارت از مکان امنیتی پایه بیش از یک‌بار اتفاق می‌افتد؛ زیرا در یک گشته بسته قرار داریم. گراف G به همراه محدودیت‌های b و t به ما کمک می‌کند تا استراتژی‌های گارد ساحلی را به طور صریح تعریف کنیم. مجموعه استراتژی‌های مدافع در این بازی، شامل دنباله‌ای از رئوس گراف است که هر عضو این دنباله یک زوج مرتب است که مؤلفهٔ اول آن موقعیت جغرافیایی و مؤلفهٔ دوم آن فعالیت امنیتی مورد نظر برای آن موقعیت جغرافیایی است. جدول زیر مثالی ملموس را به ما ارائه می‌دهد.

سطرها و ستون‌های جدول به ترتیب متناظر با استراتژی‌های مدافع و مهاجم هستند. در

برنامهٔ گشت‌زنی	هدف ۱	هدف ۲	هدف ۳	هدف ۴
$(1:k_1),(2,k_1),(1,k_1)$	50,-50	...	15,-15	...
$(1:k_2),(2,k_1),(1,k_1)$	...	60,-60	...	...
$(1:k_1),(2,k_1),(1,k_2)$	...	...	...	-20,20
$(1:k_1),(3,k_1),(2,k_1),(1,k_1)$	50,-50	...	...	...
$(1:k_1),(2,k_1),(3,k_1),(1,k_1)$	...	...	15,-15	...

این مثال، دو فعالیت دفاعی k_1 و k_2 موجود است که k_2 سود بیشتری دارد؛ زیرا مدت زمان بیشتری طول می‌کشد. هم‌چنین، محدودیت زمانی t به اندازهٔ دو برابر زمان لازم برای فعالیت k_2 در نظر گرفته شده است. به علاوه، موقعیت جغرافیایی اول شامل دو هدف و موقعیت‌های جغرافیایی دوم و سوم هرکدام یک هدف دارند؛ یعنی در مجموع ۳ موقعیت جغرافیایی و ۴ هدف برای حملهٔ دشمن وجود دارد. در تمام استراتژی‌های نوشته شده در جدول برای مدافع، مکان اول یک مکان امنیتی پایه است. مثلاً، گشت امنیتی سطر دوم با فعالیت k_2 روی مکان اول آغاز می‌شود. سپس، گارد ساحلی به مکان امنیتی دوم سفر کرده و فعالیت k_1 را در آنجا انجام می‌دهد و دوباره با فعالیت k_1 به مکان آغازین برمی‌گردد و گشت تمام می‌شود.

برای سودها، اگر هدف i ام انتخاب دشمن باشد و شکست بخورد، مدافع سود R_i^d را کسب می‌کند و مدافع ضرری به اندازهٔ P_i^a متحمل می‌شود. در غیر این صورت، مدافع به اندازهٔ P_i^a

ضرر می‌کند و سود مهاجم نیز R^a است. به علاوه، فرض کنید که G^d_{ij} سود مدافع باشد وقتی که مهاجم هدف i را انتخاب کرده و مدافع از مکان j ام محافظت می‌کند. G^d_{ij} را می‌توان به صورت ترکیب خطی سود و زیان مدافع برای هدف i بیان کرد که A_{ij} بیانگر احتمال کارآمدی فعالیت دفاعی برای هدف i در موقعیت جغرافیایی j است. در صورتی که هدف i در مکان j قرار نداشته باشد، A_{ij} برابر صفر است و در صورتی که مکان j شامل فقط یک فعالیت دفاعی روی هدف i باشد، می‌توانیم سود حاصل را با استفاده از رابطهٔ زیر به دست آوریم.

$$G^d_{ij} = A_{ij}R^d_i + (1 - A_{ij})P^d_i$$

برای مثال، در استراتژی پنجم جدول، هدف ۱ با فعالیت دفاعی k_1 پوشش داده شده است. اگر $A_{ij} = 0.5$، $R^d_i = 150$ و $P^d_1 = -50$ باشد، آنگاه

$$G^d_{15} = 0.5(150) + (1 - 0.5)(-50) = 50$$

هم‌چنین، G^a_{ij} را نیز به طور مشابه می‌توان محاسبه کرد. توجه کنید که سودهای متناظر شده با هر هدف بر اساس اندازه‌گیری‌های یک شرکت تحلیل ریسک بر اساس معیارهایی مثل آسیب مالی یا جانی است. به علاوه، برخلاف الگوریتم DOBSS، سیستم PROTECT با فرض مجموع صفر بودن بازی بین مدافع و مهاجم کار می‌کند.

سیستم PROTECT بعد از تعیین استراتژی‌های مدافع، با ترکیب استراتژی‌ها و حذف استراتژی‌های مسلط شده، یک فرم فشرده‌ای از استراتژی‌ها را برای ماتریس بازی فراهم می‌کند. در این فشرده‌سازی، ابتدا از بین استراتژی‌های معادل، یعنی استراتژی‌هایی که تنها در جایگشت موقعیت‌های جغرافیایی تفاوت دارند، تنها یکی را نگه می‌داریم؛ زیرا همگی سود برابر دارند. سپس، اگر در یک استراتژی، برای یک موقعیت جغرافیایی، بیش از یک فعالیت امنیتی مشخص شده باشد، آن فعالیتی که بیش‌ترین کارآمدی را دارد حفظ کرده و بقیه را از استراتژی حذف می‌کنیم. دلیل این امر، کمک به انجام محاسبات است بدون اینکه خللی

به کلیت وارد شود؛ زیرا فعالیت‌های امنیتی جانبی آنچنان تفاوتی از نظر میزان سود ندارند. و در انتها استراتژی‌های مسلط شده از ماتریس بازی حذف می‌شوند. جدول زیر را در نظر بگیرید.

در این جدول، استراتژی فشردهٔ $\gamma_2 = \{(1, k_2), (2, k_1)\}$ متناظر با سطرهای دوم و سوم

استراتژی فشرده	هدف ۱	هدف ۲	هدف ۳	هدف ۴
$\Gamma_1 = \{(1, k_1), (2, k_1)\}$	50,-50	30,-30	15,-15	-20,20
$\Gamma_2 = \{(1, k_2), (2, k_1)\}$	100,-100	60,-60	15,-15	-20,20
$\Gamma_3 = \{(1, k_1), (2, k_1), (3, k_1)\}$	50,-50	30,-30	15,-15	10,-10

جدول قبل است. به علاوه، تنها تفاوت استراتژی‌های γ_1 و γ_2 در فعالیت امنیتی مربوط به موقعیت جغرافیایی اول است و طبق فرض k_2 سود بیشتر دارد؛ به همین علت استراتژی مسلط شدهٔ γ_1 نیز می‌تواند حذف شود. در نهایت، ماتریس تولید شده برای بازی ماتریسی شامل استراتژی‌های فشرده است.

۲.۴ به کار گیری مدل QR

"تعادل پاسخ کمی" یک مدل مهم در نظریهٔ بازی‌ها است که به سبب توانایی‌اش در مدل‌سازی رفتار انسانی در بازی‌های با حرکات هم‌زمان بسیار مورد توجه قرار گرفته است و مدل پایهٔ بسیاری از مطالعات بوده است. در این مدل، بازیکنان به جای بیشینه کردن سود خود، به طور تصادفی در بازی عمل می‌کنند و احتمال انتخاب یک استراتژی غیر بهینه افزایش می‌یابد وقتی که هزینهٔ خطای آن کاهش یابد. با این وجود، تاکنون این مدل برای بازی‌های استکلبرگ به کار نرفته بود. در بازی‌های استکلبرگ، فرض می‌کنیم که مهاجم دارای عقلانیت محدود است و مدافع به کمک نرم‌افزار استراتژی منطقی بهینهٔ مدافع را محاسبه می‌کند. با توجه به استراتژی مدافع، بهترین پاسخ کمی مهاجم به صورت زیر تعریف می‌شود.

$$q_i = \frac{e^{\lambda G_i^a(x_i)}}{\sum_{j=1}^{T} e^{\lambda G_i^a(x_i)}}$$

پارامتر λ نشان‌دهندهٔ مقدار خطا در استراتژی مهاجم است و می‌تواند از صفر تا بی‌نهایت تغییر کند. مقدار صفر نشان‌دهندهٔ کاملاً تصادفی انتخاب شدن استراتژی‌های مهاجم و مقدار بی‌نهایت به معنای کاملاً منطقی انتخاب کردن استراتژی‌ها توسط او است. q_i بیانگر احتمال حملهٔ مهاجم به هدف i ام است. $G_i^a(x_i)$ متناظر با سود انتظاری مهاجم از حمله به هدف i است وقتی که مدافع با احتمال x_i از هدف i دفاع می‌کند و در نهایت، T تعداد کل اهداف است.

یک بازی استکلبرگ با سودهای جدول زیر را در نظر بگیرید.

فرض کنید رهبر بازی به استراتژی b متعهد شده است. در این صورت، اگر پیرو c را انتخاب

	c	d
a	3,1	5,0
b	2,0	4,2

کند، سود صفر و اگر d را انتخاب کند، صود ۲ را به دست می‌آورد. یک مهاجم منطقی d را بازی می‌کند تا سود خود را بیشینه سازد اما بهترین پاسخ کمی به ما می‌گوید که مهاجم با احتمال

$$\frac{e^0}{e^0+e^{2\lambda}}$$

استراتژی c را انتخاب می‌کند و با احتمال

$$\frac{e^{2\lambda}}{e^0+e^{2\lambda}}$$

استراتژی d را بازی می‌کند.

در حالی که در کاربردهای قبلی نظریهٔ بازی‌ها در امنیت مهاجمان با عقلانیت کامل فرض

شذه بودند، سیستم PROTECT با در نظر گرفتن فرض عقلانیت محدود برای مهاجمان و استفاده از QR [۲] یک گام رو به جلو برداشته است. بر اساس این مدل، در مدل بازی استکلبرگ، بهترین پاسخ مهاجم با توجه به مدل QR داده می‌شود و مدافع با آگاهی از این موضوع، استراتژی بهینهٔ خود را محاسبه می‌کند. برای به کار بردن مدل QR در ساختار بازی استکلبرگ، سیستم PROTECT از الگوریتمی به نام PASAQ استفاده می‌کند که کار آن، محاسبهٔ استراتژی بهینهٔ مدافع با توجه به مدل QR برای دشمنان و حل مسئلهٔ برنامه‌ریزی غیرخطی P با نمادگذاری جدول داده شده است.

خط اول مسئله متناظر با محاسبهٔ سود انتظاری مدافع است. $QR(i|\lambda)$ احتمال استفادهٔ

$$\mathrm{P}:\begin{cases}\max_{a} \sum_{i=1}^{T} QR(i\,|\,\lambda)((R_i^d - P_i^d)x_i + P_i^d) \\ x_i = \sum_{j=1}^{J} a_j A_{ij}, \forall i \\ \sum_{j=1}^{J} a_j = 1 \\ 0 \le a_j \le 1, \forall j\end{cases}$$

t_i	هدف i
$R^d i$	سود مدافع از پوشش هدف t_i وقتی که مورد حمله قرار می‌گیرد
$P^d i$	ضرر مدافع از عدم پوشش هدف t_i وقتی که مورد حمله قرار می‌گیرد
$R^a d$	سود مهاجم از حمله به هدف t_i وقتی که محافظت نشده است
$P^a d$	ضرر مهاجم از حمله به هدف t_i وقتی که محافظت شده است
A_{ij}	احتمال کارآمد بودن انتخاب استراتژی فشردهٔ Γ_i برای هدف t_i
a_j	احتمال انتخاب استراتژی فشردهٔ Γ_i
J	تعداد کل استراتژی‌های فشرده

مهاجم از مدل QR برای حمله به هدف i است. هم‌چنین، در خط دوم، مقدار محافظت از هدف با توجه به A_{ij} است که کارآمدی فعالیت امنیتی را اندازه می‌گیرد. به علاوه، احتمال انتخاب استراتژی فشردهٔ γ_j با a_j نشان داده شده است. برای مدل ما، λ در بازهٔ $(0.5, 4)$

[۲] quantal response

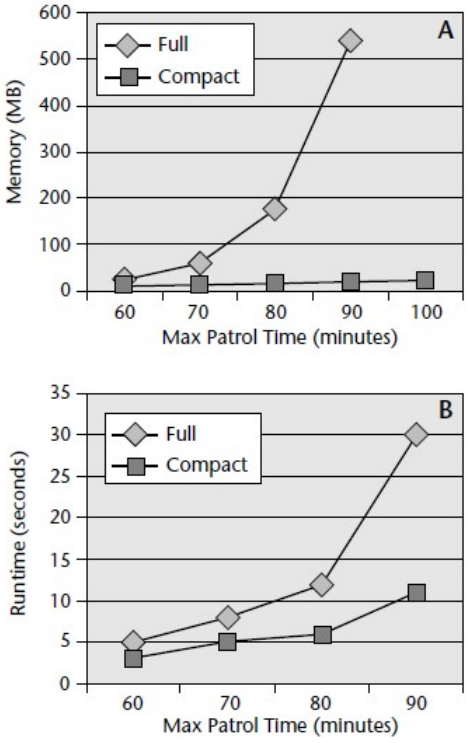
قرار دارد که ابتدای بازه به معنای انتخاب کاملاً تصادفی و انتهای بازه به معنای انتخاب کاملاً منطقی است. اندازه‌گیری‌ها نیز نشان داده است که بیشینه سود متوسط برای مدافع به ازای $\lambda = 1.5$ حاصل می‌شود.

۳.۴ ارزیابی سیستم PROTECT

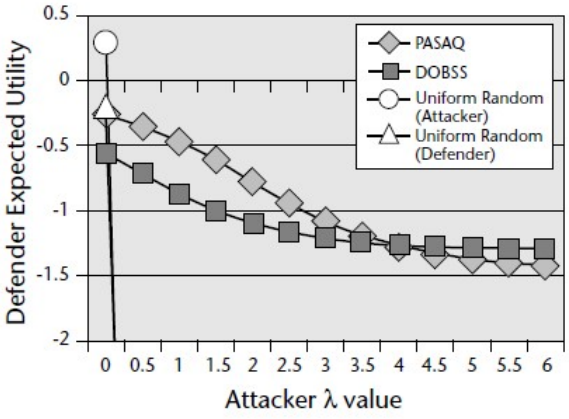
در این بخش، به ارزیابی‌های انجام شدهٔ سیستم $PROTECT$ بر اساس شبیه‌سازی‌های صورت گرفته و اطلاعات دریافتی از گارد ساحلی بندر در دنیای واقعی می‌پردازیم. در آزمایش‌های انجام شده، بازی مجموع صفر فرض شده و سود مدافع در بازهٔ $(-10, 5)$ و سود مهاجم در بازهٔ $(-5, 10)$ قرار داده شده است و مقدار λ نیز 1.5 فرض شده است.

نمودارهای ۱.۴ تأثیر فشرده‌سازی استراتژی‌ها در زمان اجرای الگوریتم و حافظهٔ مورد نیاز آن را نشان می‌دهد. وقتی که بیشینه زمان گشت‌زنی ۹۰ دقیقه باشد، الگوریتم با استراتژی‌های کامل ۵۴۰ مگابایت حافظه مصرف کرده و ۳۰ ثانیه طول می‌کشد تا به جواب برسد در حالی که با استفاده از استراتژی‌های فشرده به عنوان ورودی، زمان مورد نیاز به ۱۱ ثانیه و حافظهٔ مصرفی به تنها ۲۰ مگابایت کاهش پیدا می‌کنند.

وقتی سیستم PROTECT از الگوریتم PASAQ با $\lambda = 1.5$ استفاده می‌کند، نسبت به سایر مقادیر λ از مزیت بیشتری برخوردار است. هرچند که مقایسهٔ الگوریتم‌ها در حالتی که $\lambda = \infty$ است، اهمیت زیادی دارد؛ زیرا الگوریتم‌های قبلی مثل DOBSS عقلانیت کامل پیرو را فرض کرده بودند. نمودار ۲.۴ ، سود الگوریتم‌های نام برده را بر حسب مقادیر مختلف λ نشان می‌دهد.



شکل ۱.۴: تأثیر فشرده‌سازی استراتژی‌ها.



شکل ۲.۴: بررسی الگوریتم‌ها از نظر سود.

فصل پنجم

نتیجه گیری

در این پروژه، به دو کاربرد مهم نظریهٔ بازی‌ها در امنیت پرداختیم. ابتدا، با استفاده از بازی‌های بیزی با تعادل استکلبرگ مسئلهٔ تخصیص تصادفی بهینهٔ منابع امنیتی برای محافظت از فرودگاه بین‌المللی لس‌آنجلس را مدل کردیم. از پیش مشخص بودن استراتژی‌های نیروی امنیتی برای دشمنان و تصمیم‌گیری آنان با توجه به آگاهی داشتن از برنامهٔ امنیتی فرودگاه و هم‌چنین، ناشناخته بودن نوع دشمنان برای نیروهای امنیتی ما را به استفاده از بازی‌های بیزی با تعادل استکلبرگ سوق دادند. در این قسمت، دو فرض محدود کننده را برای مدل‌سازی در نظر داشتیم. اول اینکه، دشمن همواره از استراتژی محض استفاده خواهد کرد و از میان اهداف موجود برای حمله، یکی را انتخاب می‌کند. دوم اینکه، در بازی بیزی طراحی شده، نیروی امنیتی دارای تنها یک نوع بود و دشمن نیز در این مورد اطلاعات کامل داشت.

به عنوان تعمیمی از مدل‌سازی انجام شده، می‌توان در کارهای بعد، استراتژی‌های مخلوط را نیز به استراتژی‌های دشمنان هنگام برنامه‌ریزی برای حمله اضافه کرد. در واقع، سناریویی را در نظر بگیرید که دشمن نیز همانند نیروی امنیتی از استراتژی مخلوط استفاده می‌کند تا سود خود را افزایش دهد. به عنوان مثال، فرودگاهی با دو پایانه را در نظر بگیرید که یکی از برنامه‌های دشمن برای حمله می‌تواند این باشد که با احتمال ۷۰ درصد به پایانهٔ اول حمله

کند و با احتمال ۳۰ درصد پایانهٔ دوم را برای حمله انتخاب کند و یا ۷۰ درصد نیروهای دشمن از پایانهٔ اول حمله کنند و ۳۰ درصد باقی‌مانده پایانهٔ دوم را برای نفوذ انتخاب کنند. سؤالی که می‌توان مطرح کرد، این است که در این شرایط، نیروی امنیتی باید چگونه تصمیم بگیرد تا بتواند با متحمل شدن کم‌ترین آسیب از هر دو پایانه محافظت کند و الگوریتم اشاره شده در فصل سوم با چه تغییراتی مواجه خواهد شد.

به علاوه، اگر در بازی بیزی بین نیروی امنیتی و دشمن، نیروی امنیتی نیز دارای چند نوع باشد، مدل‌سازی انجام شده مجدداً نیاز به تغییر خواهد داشت. برای ملموس‌تر شدن انواع مختلف نیروی امنیتی، فرض کنید نیروی امنیتی از چند راهکار برای حفظ امنیت فرودگاه استفاده می‌کند و به جز موارد اشاره شده در متن شامل ایست‌های بازرسی وسایل نقلیه و واحد گشت سگ‌ها، از دوربین‌های مدار بسته نیز بهره می‌برد که کل اتفاقات فرودگاه را پوشش می‌دهد. حال اگر دشمنان از برنامهٔ امنیتی فرودگاه که به طور تصادفی توسط نرم‌افزار ARMOR به دست می‌آید، بتوانند عبور کنند، امنیت فرودگاه تا حدی به خطر می‌افتد ولی همچنان با توجه به وجود دوربین‌های مدار بسته در فرودگاه و مشاهدهٔ سوژه‌های مشکوک به مرور زمان، نیروی امنیتی برای مقابله با دشمنان شانس خواهد داشت و احتمال موفقیتش بسته به اینکه چه زمانی اطلاعات از دوربین‌های مدار بسته دریافت شوند، متفاوت خواهد بود و طبیعتاً هر چه دیرتر متوجه عملیات تهاجمی دشمن شوند، با احتمال کمتری می‌توانند جلوی آنان را بگیرند. در این وضعیت، نوع نیروی امنیتی نیز برای دشمن ناشناخته خواهد بود؛ زیرا در صورتی که حملهٔ آن‌ها در ایست بازرسی یا توسط واحد گشت سگ‌ها متوقف شود، بلافاصله دستگیر خواهند شد؛ اما اگر بتوانند از این دو عبور کرده و عملیات خود را ترتیب اثر دهند، وابسته به اینکه چه مدت بعد از اقدام برای تهدید امنیت فرودگاه توسط دوربین‌های مدار بسته رؤیت شده و پلیس برای دستگیری آن‌ها اقدام خواهد کرد، می‌توانند امیدوار به فرار از دست نیروهای امنیتی باشند. بدین ترتیب، با در نظر گرفتن دوربین مدار بسته به عنوان یکی از انواع نیروی امنیتی، از طرفی سود دشمن در مقابله با هر نوع نیروی امنیتی متفاوت خواهد بود و از طرف دیگر، اطلاعاتی راجع به اینکه با کدام نوع نیروی امنیتی

مقابله می‌کند، ندارد؛ در نتیجه، یک بازی بیزی خواهیم داشت که هر دو بازیکن دارای بیش از یک نوع هستند.

در فصل چهارم، به کاربرد نظریهٔ بازی‌ها در بنادر به منظور حفظ سرمایه‌های مهم یک بندر اشاره شد. با توجه به رقابت بین گارد ساحلی و دشمنان و توضیحاتی که در بند قبلی بیان شد، باز هم از ساختار بازی استکلبرگ (مهاجم - مدافع) برای مدل‌سازی استفاده کردیم. سیستم PROTECT که از الگوریتم PASAQ بهره می‌برد، دو تفاوت عمده با نرم‌افزار ARMOR و الگوریتم مورد استفاده در این نرم‌افزار، یعنی DOBSS داشت. اولاً، در سیستم PROTECT برای دشمن انواع مختلفی در نظر نگرفتیم و مهاجم نیز همانند مدافع تنها یک نوع داشت؛ بنابراین از بازی بیزی استفاده نکردیم. می‌توان ساختار مدل‌سازی این قسمت را نیز با وجود انواع مختلف برای دشمن بررسی کرد. ثانیاً، در این سیستم یک گام رو به جلو نسبت به الگوریتم‌های قبلی در مسئلهٔ امنیت برداشته شد و آن، استفاده از مدل QR برای پاسخ‌دهی دشمن به استراتژی گارد ساحلی بود که ما را با واقعیت و مدل رفتاری بازیکنان در تصمیم‌گیری‌ها نزدیک‌تر می‌کرد. مدل QR را می‌توانیم برای الگوریتم‌های قبلی نظیر DOBSS نیز به کار ببریم. هرچند که با در نظر گرفتن عقلانیت کامل برای دشمنان در الگوریتم PASAQ باز هم مقایسه‌های انجام شده، برتری این الگوریتم نسبت به رقبایی مثل DOBSS و روش برنامه‌ریزی خطی را نشان می‌داد. دلیل این امر، استفاده از فشرده‌سازی استراتژی‌ها و حذف استراتژی‌های مسلط شده برای گارد ساحلی بود که زمان اجرای الگوریتم را کوتاه‌تر می‌کرد و به دادن خروجی سرعت می‌بخشید. هم‌چنان، یافتن الگوریتم‌های سریع‌تر از PASAQ برای بهبود عملکرد دفاعی نیروهای امنیتی در حوزه‌های مختلف، می‌تواند جزو موضوعات مورد مطالعه در بحث کاربرد نظریهٔ بازی‌ها در امنیت باشد.

به علاوه، در دنیای امروز و زمان کنونی، همان اندازه که احتمال تهدید زیرساخت‌های مهم از طریق راه‌های زمینی و با توجه به موقعیت مکانی، احتمالاً از طریق راه‌های دریایی وجود دارد، راه‌های هوایی نیز تهدید محسوب می‌شوند و مقابله با حملات هوایی که از راه دور به وسیلهٔ موشک یا هواپیماهای جنگنده صورت می‌پذیرد، از اهمیت زیادی برخوردار است. در

این راستا، در ادامهٔ کارهای با ارزش انجام شده در گذشته که بخش اعظم آن توسط تیم آقای Milind Tambe صورت گرفته و به دو تا از مهم‌ترین کارهای آن‌ها در این پژوهش اشاره کردیم، مدل‌سازی رقابت بین نیروی امنیتی و دشمنان در مرزهای هوایی نیز پیشنهاد می‌شود.

کتاب‌نامه

[1] P. Parchuri, M. Tambe, J. P. Pearce, F. Ordonez, *Playing games for security: An efficient exact algorithm for solving Bayesian Stackelberg games*, Conference Paper, January 2008.

[2] P. Parchuri, M. Tambe, J. P. Pearce, F. Ordonez, *An Efficient Heuristic for Security against Multiple Adversaries in Stackelberg Games*, Conference Paper, January 2007.

[3] J. Pita, M. Tambe, F. Ordonez, M. Jain, *Using Game Theory for Los Angeles Airport Security*, Article in AI Magazine, March 2009.

[4] B. An, E. Shieh, R. Yang, M. Tambe, C. Baldwin, J. Direnzo, B. Maule, G. Meyer, *PROTECT- A Deployed Game-Theoretic System for Strategic Security Allocation for the United States Coast Guard*, Article in AI Magazine, December 2012.

[5] J. Pita, M. Tambe, F. Ordonez, M. Jain, *Using Game Theory for Los Angeles Airport Security*, Article in AI Magazine, March 2009.

[6] M. J. Osborne, *An Introduction to Game Theory*, Oxford University Press, 2004.

[7] M. S. Bazaraa, J. J. Jarvis, *Linear Programming and Network Flows*, Georgia Institute of Technology, 1977.

[8] G. Chartrand, P. Zhang, *A First Course in Graph Theory*, Western Michigan University, 2012.

Abstract

The main purpose of this project is to express some of the applications of game theory in the security and protection of important infrastructures. In this regard, we first make an introductory statement of game theory, including the concept of equilibrium. Then, we define Stackelberg games and Bayesian games separately. In the primary section of this note, we examine two applications of game theory in security. First, we express how to use game theory to protect Los Angeles International Airport and modeling with Bayesian Stackelberg games which leading to software has been provided to assist this airport security forces. In the following, we explain how to protect the port of Boston in United States and a system which assist the United States Coast Guard. In the end, we also analyze the experimental results of each of them in order to find out better the efficiency of algorithms used.

College of Science

School of Mathematics, Statistics, and Computer Science

Applications of Game Theory in Security

Seyed Mahdi Mazloum

Supervisor: Dr. Mahdi Reza Darvishzadeh

A thesis submitted to Undergraduate Studies Office

in partial fulfillment of the requirements for the degree of B.Sc.

Mathematics and its Applications

Spring and Fall 2021