به کارگیری اتوماتای یادگیر[1] برای ساختن استراتژی در Iterated Prisoner's Dilemma

محمدرضا آیت اله زاده شیرازی **محمدرضا میبدی**

آزمایشگاه محاسبات نرم

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی امیرکبیر

تهران- ایران

چکیده

در این مقاله، مسئلۀ به کارگیری اتوماتای یادگیر برای ساختن استراتژی در Iterated Prisoner's Dilemma (IPD) مورد بررسی و ارزیابی قرار می گیرد. Prisoner's Dilemma (PD) مدلی برای مطالعۀ همکاری، تضاد و تصمیم گیری می باشد. هم اکنون، بیشتر تحقیقات در رابطه با PD در زمینۀ ساختن استراتژیهای خوب برای آن است. پرسش اصلی این مقاله این است که آیا اتوماتای یادگیر می تواند به استراتژی مطلوب در IPD همگرا گردد؟ و آیا بازیکنی که از اتوماتای یادگیر برای تصیمم گیری در بازی استفاده می کند، می تواند به شکل گروهی یا فردی معقول رفتار کند؟ نتایج حاصل از ارزیابیها نشان می دهند که اتوماتای یادگیر با ساختار متغیر مورد استفاده در ارزیابی در بازی با اتوماتای یادگیر دیگر به موازنۀ Nash همگرا می شود. همچنین در مسابقه های تکی با استراتژیهای دیگر قطعی یا احتمالاتی طراحی شده برای IPD اتوماتای یادگیر با انتخاب مقادیر مناسب برای پارامتر پاداش به نتایج مطلوب گروهی و فردی دست می یابد.

کلمات کلیدی: اتوماتای یادگیر با ساختار متغیر، یادگیری تقویتی، Iterated Prisoner's Dilemma

۱. مقدمه

مکاترونیک یکپارچه سازی و ادغام مهندسی مکانیک با الکترونیک و کنترل هوشمند کامپیوتر در طراحی و ساخت محصولات صنعتی و فرآیند ها می باشد. مدل سازی و طراحی، یکپارچه سازی سیستم، کنترل هوشمند، روباتیک، ساخت و تولید، کنترل حرکت، کنترل نویز و لرزش، سیستمهای نوری-الکتریکی از جمله موضوعاتی هستند که در حوزۀ مکاترونیک مطرح می گردند. هم اکنون روشهای هوشمند و ریاضی مانند نظریه بازیها، یادگیری ماشین، شبکه های عصبی، منطق فازی، نظریه آشوب، بازشناسی الگو، بازنمایی دانش، روشهای استنتاج هوشمند و غیره یا ترکیب این روشها در مکاترونیک به منظور برنامه ریزی، طراحی، کنترل فرآیند، یکپارچه سازی سیستم، بهبود مدل سازی سیستمها و ارائۀ راه حلهای جدید برای مسائل موجود در مکاترونیک استفاده می شود. در این مقاله، استفاده از یک روش یادگیری تقویتی به نام اتوماتای یادگیر در حل یکی از مسائل مهم در نظریۀ بازیها یعنی ساخت استراتژی بازی مورد بررسی و ارزیابی قرار می گیرد. نتایج حاصل در تحقیقات مکاترونیک به خصوص در زمینۀ بهبود مدل سازی و تحلیل سیستمهای مکاترونیکی قابل استفاده می باشد.

Prisoner's Dilemma (PD) مدلی برای مطالعۀ همکاری و تضاد می باشد[1,2,3,9]. بر روی این مدل، مطالعات بسیاری شده است. هم اکنون بیشتر تحقیقات در این زمینه بر روی ساختن استراتژیهای خوب برای آن است و می توان گفت که ساختن این استراتژیها بسیار سخت تر از ساختن

[1] Learning Automata

اســتراتژی بــرای بازیهــای کلاســیک اســت[2,4,5]. در زمینــۀ ســاخت اســتراتژی بــرای ایــن مــسئله تلاشــهای بــسیاری صــورت گرفتــه اســت [1,2,4,5,6,7,8,10,11] که می توان این فعالیتها را در زمینۀ ساخت استراتژیهای قطعی، احتمالاتی و تکاملی برای PD تقسیم نمـود.در ایـن مقالـه، به کارگیری اتوماتای یادگیر با ساختار متغیر به عنوان یک استراتژی احتمالاتی برای IPD مورد بررسی قـرار مـی گیـرد. اتوماتـای یـادگیر در محیطـی تصادفی عمل می کند و استراتژی خودش برای انتخاب اعمال را براساس پاسخ دریافتی از محیط بهنگام سازی می کند. اتوماتـا دارای تعـداد محـدودی عمل می باشد و متناظر با هر عمل، پاسخ محیط با درجه ای از اطمینان می تواند مطلوب یا نامطلوب باشد. اتوماتا با به کارگیری استراتژیهای قطعی یا تصادفی می تواند به مقاصد متفاوتی دست پیدا کند. اتوماتای یادگیر به دو گروه عمدۀ اتوماتای یادگیر با ساختار ثابت و اتوماتای یادگیر با ساختار متغیر تقسیم می گردد. در این مقاله، استفاده از اتوماتای یادگیر با ساختار متغیر به عنوان استراتژی در بازی IPD مورد ارزیابی قرار می گیرد.

در ادامه مقاله، ابتدا در بخش ۲، اتوماتای یادگیر، اتوماتای یادگیر با ساختار متغیر و شماهای تقویتی در محیطهای S مورد بررسـی قـرار مـی گیرنـد. بخش ۳، به معرفی بازی Prisoner's Dilemma و مروری بر روی فعالیتهای انجام شده در رابطه با تعریف استراتژی برای بازی PD می پـردازد. بخـش ۴، فعالیت انجام شده در راستای ساختن استراتژی در IPD توسط اتوماتای یادگیر را توضیح می دهد. بخش ۵ و ۶ به بیان نتـایج حاصـل از ارزیابیهـای انجام شده برروی اتوماتای یادگیر در IPD می پردازد. در بخش ۷ نتیجه گیری ارائه می شود.

۲. اتوماتای یادگیر

اتوماتای یادگیر یک مدل انتزاعی است که تعداد معدودی عمل را می تواند انجام دهد. هر عمل انتخاب شده توسط محیطی احتمالی ارزیـابی شـده و پاسخی به اتوماتای یادگیر داده می شود. اتوماتای یادگیر از این پاسخ استفاده نموده وعمل خود را برای مرحله بعد انتخاب می کند[12,13] . شـکل ۱ ارتباط بین اتوماتای یادگیر و محیط را نشان می دهد.

شکل ۱: ارتباط بین اتوماتای یادگیر و محیط

محیط را می توان توسـط سـه تـایی $E \equiv \{\alpha, \beta, c\}$ نـشان داد کـه در آن $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ مجموعـه ورودیهـا، $\beta \equiv \{\beta_1, \beta_2, ..., \beta_m\}$ مجموعه خروجیها و $c \equiv \{c_1, c_2, ..., c_r\}$ مجموعه احتمالهای جریمه می باشد. هر گاه β مجموعه دو عضوی باشد ، محیط از نوع P می باشـد. در چنین محیطی $\beta_1 = 1$ به عنوان جریمه و $\beta_2 = 0$ به عنوان پاداش در نظر گرفته می شود. در محیط از نوع Q ، $\beta(n)$ می تواند به طور گسسته یـک مقدار از مقادیر محدود در فاصله [0,1] و در محیط از نـوع S، $\beta(n)$ متغیـر تـصادفی در فاصـله [0,1] اسـت. c_i احتمـال اینکـه عمـل α_i نتیجـه نامطلوب[۲] داشته باشد، می باشد. در محیط ایستا[۳] مقادیر c_i بدون تغییر می مانند ، حال آن که در محیط غیر ایستا[۴] این مقـادیر در طـی زمـان تغییـر می کنند. اتوماتای یادگیر به دو گروه با ساختار ثابت و با ساختار متغیر تقسیم می گردد. با توجه به این کـه در ایـن مقالـه از اتوماتـای سـاختار متغیـر استفاده شده است، در ادامه توضیحاتی در رابطه با اتوماتای ساختار متغیر داده می شود. برای مطالعه بیشتر در رابطه با اتوماتاهای ساختار ثابت و متغیر می توان به [12,13,14,15,16] مراجعه نمود.

اتوماتـای یـادگیر بـا سـاختار متغیـر[۵] توسـط ۴ تـایی $\{\alpha, \beta, p, T\}$ نـشان داده مـی شـود کـه در آن $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ مجموعـه عملهـای اتوماتــا، $\beta \equiv \{\beta_1, \beta_2, ..., \beta_m\}$ مجموعــه ورودیهــای اتوماتــا و $p = \{p_1, p_2, \ldots p_n\}$ بــردار احتمــال انتخــاب هــر یــک از اعمــال و $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری می باشد. در این نوع از اتوماتاها، اگر عمل α_i در مرحله n ام انتخاب شود و این عمل، پاسخ مطلوب از محیط دریافت نماید، احتمال $p_i(n)$ افزایش یافته و سایر احتمالها کاهش می یابند. برای پاسخ نامطلوب احتمال $p_i(n)$ کـاهش یافتـه و سایر احتمالها افزایش می یابند. در هر حال، تغییرات به گونه ای صورت می گیرد تا حاصل جمع $p_i(n)$ ها همواره ثابـت و مـساوی یـک بـاقی بمانـد. الگوریتم (۱) نمونه ای از الگوریتمهای یادگیری خطی در اتوماتای با ساختار ثابت است[13].

الف- پاسخ مطلوب برای عمل i:

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)] \qquad (۱)$$

---

[۲] Unfavorable

[۳] Stationary

[٤] Non-Stationary

[5] Variable Structure Learning Automata

$$p_j(n+1) = (1-a)\,p_j(n) \qquad j \neq i \quad \forall j$$

ب- پاسخ نامطلوب برای عمل i:

$$P_i(n+1) = P_i(n) - (1-b)\,P_i(n) \qquad p_j(n+1) = \frac{b}{r-1} + (1-b)\,p_j(n) \quad \forall j \quad j \neq i$$

در روابط فوق، پارامتر پاداش و a پارامتر پاداش و b پارامتر جریمه می باشد. با توجه به مقادیر a و b سه حالت زیر را می توان در نظر گرفت. زمانی که a و b با هم برابر باشند، الگوریتم را L_{R-P} [6] می نامیم. زمانیکه b از a خیلی کوچکتر باشد، الگوریتم را $L_{R-\varepsilon P}$ [7] می نامیم و زمانی که b مساوی صفر باشد، الگوریتم را L_{R-I} [8] می نامیم.

با درنظر گرفتن طبیعت پاسخ محیط، مدلهای P، Q و S برای محیطی که اتوماتای یادگیر در آن عمل می کند،درنظرگرفته می شود [13]. پاسخ در محیطهای مدل P دارای مقدار دودویی می باشد. در مدل Q متناظر با عمل α_i ، خروجی محیط ممکن است تعداد متناهی از مقادیر اختیار کند. با نرمال سازی مقادیر خروجی، هر مدل Q با مقادیر متناهی از خروجیهای محیط در فاصلهٔ واحد [0,1] مشخص می گردد. تعداد این مقادیر خروجی از عملی به عمل دیگر متفاوت است و با m_i برای عمل α_i $(i=1,2,\ldots,\gamma)$ بیان می شود. در مدل S، پاسخها می توانند مقادیری پیوسته در یک فاصله مشخص را اختیار کنند. با نرمال سازی پاسخها، می توان آنها را در فاصله [0,1] درنظرگرفت. اگر پاسخ محیط در مدل Q برای عمل α_i با $\beta^i_1, \beta^i_2, \ldots, \beta^i_{mi}$ مشخص شود که در آن $\beta^i_1 < \beta^i_2 < \ldots < \beta^i_{mi}$ ، مجموعه نرمال شدهٔ پاسخها $\{\beta^i_j\}$ به شکل $\beta^i_j = \beta^i_j - a / b\text{-}a$ $j=1,2,\ldots,m_i$ تعریف می گردد که در آن $a = \min_i \{\beta^i_j\}$ و $b = \max_i \{\beta^i_{mi}\}$ است. نرمال سازی مشابهی را نیز می توان در مدل S انجام داد. نگارشهای مدل S و Q برای شمای L_{R-I} و L_{R-P} به صورت زیر می باشند. باید توجه داشت که مدل S بازنمایی عمومی تری ازدو نگارش قبلی است. با داشتن مدل S می توان مدلهای Q و P را نیز به دست آورد. بهنگام سازی احتمالات در شمای SL_{R-I} براساس معادلهٔ (۲) بیان می شود:

$$P_i(n+1) = P_i(n) - a\,(1-\beta(n))\,P_i(n) \qquad \alpha(n) <> \alpha_I \qquad (۲)$$
$$P_i(n+1) = P_i(n) + a\,(1-\beta(n)) \textstyle\sum_{j \neq i} P_j(n) \qquad \alpha(n) = \alpha_i$$

شمای SL_{R-P} برای مدلهای Q و S براساس معادلهٔ (۳) بیان می شود:

$$P_i(n+1) = P_i(n) + \beta(n)\,[(a/r\text{-}1) - a\,P_i(n)] - [1-\beta(n)]\,a\,P_i(n) \qquad \alpha(n) <> \alpha_I \qquad (۳)$$
$$P_i(n+1) = P_i(n) + \beta(n)\,a\,P_i(n) + (1-\beta(n))\,a\,(1-P_i(n)) \qquad \alpha(n) = \alpha_i$$

برای مطالعه بیشتر در باره اتوماتاهای یادگیر می توان به [12,13,14,15,16] مراجعه کرد.

۳. Iterated Prisoner's Dilemma: یک بازی جمع غیرصفر

بازی Prisoner's Dilemma (PD) مدلی سنتی برای مطالعهٔ همکاری و تضاد می باشد[1,2,3,4,5,6,9]. گونه های مختلف PD در [1]بررسی شده اند. همچنین بحث جالبی در رابطه با PD در [3] یافت می شود. با وجود این که PD مدل ساده ای دارد، ولی می توان پدیده های پیچیدهٔ بسیاری را براساس آن مطالعه کرد. یکی از مسائل مهم در PD ساختن استراتژیهای خوب برای آن است که ساختن این استراتژیها یا همان الگوریتمهایی که توسط عاملها در DAI پیاده سازی می شوند، بسیار مشکل تر از ساختن استراتژی برای بازیهای کلاسیک است [2,4,5]. وضعیتهایی که در آنها تضاد به وجود می آید، نه تنها نیروی محرکه ای در طبیعت و جامعه هستند، بلکه نقطهٔ شروعی برای بسیاری از تحقیقات در حوزهٔ هوش مصنوعی توزیع شده (DAI) می باشند. بررسی نمونه های PD در طبیعت در [4] یافت می شود.

PD در سال ۱۹۵۲ توسط Merill M. Flood و Melvin Dresher در شرکت RAND معرفی گردد. آنها سعی کردند تا مسئلهٔ نامعقولیت[9] را در نظریهٔ بازیهای جان فون نویمان و اسکار مورجنسترن وارد کنند. PD بر مبنای داستان ساده ای که توسط آلبرت تاکر بیان شد، به شرح زیر عمل می کند:

دو مرد به خاطر نقض قانون جداگانه در زندان نگهداری می شوند. به هرکدام گفته شده است که:

۱) اگر یکی اعتراف کند و دیگری اعتراف نکند، اولی آزاد می شود و دومی برای ۲۰ سال به زندان می رود. اگر هر دو اعتراف کنند، هر دو ده سال به زندان می روند.

۲) اگر هیچکدام اعتراف نکنند، هر دو یکسال زندانی می شوند.

---

[6] Linear Reward Pealty

[7] Linear Reward Epsilon Penalty

[8] Linear Reward Inaction

[9] irrationality

چگونه می توان این بازی را حل کرد؟ اگر هر دو نفر بخواهند زمانی را که در زندان به سر می برند، حداقل کنند چه استراتژیهایی معقول هستند؟ یکی از این دو نفر ممکن است این چنین استنتاج کند که: دو حالت ممکن است اتفاق بیافتد: نفر دیگر اعتراف کند یا ساکت بماند. اگر اعتراف کند، اگر من اعتراف نکنم، ۲۰ سال به زندان می روم اما اگر من هم اعتراف کنم، ۱۰ سال به زندان می روم، پس در این حالت بهتر است که اعتراف کنم. از سوی دیگر اگر نفر دیگر اعتراف نکند و من هم اعتراف نکنم، یک سال زندانی می شوم، اگر من اعتراف کنم، آزاد می شوم. در هر حالت، بهترین حالت اعتراف است، پس من اعتراف می کنم. اما نفر دیگر هم به همین شکل استنتاج می کند به طوری که هر دو اعتراف می کنند و برای ۱۰ سال به زندان می روند. هنوز هم اگر هر دو نامعقول عمل کرده بودند و ساکت می ماندند، هر دو یکسال زندانی می شدند.

معضلی که در پیش روی این دو زندانی وجود دارد این است که هرکاری که دیگری انجام می دهد، اعتراف برای هر کدام بهتر از سکوت است. اما نتیجه ای که حاصل از اعتراف هر دو می باشد، برای هریک بدتر از نتیجه ای است که در حالت سکوت هر دو نفر حاصل می گردد. این معما، تضادی را بین *معقولیت گروهی و فردی* نشان می دهد. گروهی که اعضایش علاقهٔ شخصی معقول را دنبال می کنند از گروهی که اعضایش برخلاف علاقه شخصی معقول عمل می کنند، به نتیجهٔ بدتری دست می یابد. این نتیجه، بزرگترین تاثیر را برروی علوم مدرن اجتماعی گذاشته است. تعاملات بسیاری در دنیای مدرن وجود دارند که بسیار شبیه به این وضعیت هستند. به عنوان مثال می توان به مسئلهٔ ازدحام ترافیک، آلودگی هوا، کاهش مراکز حمل ماهی و استخراج بیش از حد منابع زیرآبی اشاره کرد. جزییات این تعاملات بسیار متفاوت است، اما در تمام این تعاملات عمل معقول فردی برای هر فرد نتیجهٔ کمی به همراه دارد و PD به بررسی این تعاملات می پردازد. البته PD مدلی بسیار ساده و مجرد از این تعاملات است.PD بازی دونفره است، اما بسیاری از کاربردهای این ایده، تعاملات چندنفره هستند. البته انواع چندنفرهٔ این بازی نیز در [10] مورد بحث قرار گرفته اند. چند نکته در رابطه با PD وجود دارد که در ادامه به آنها اشاره می شود:

- فرض کرده ایم که هیچ ارتباطی بین دو نفر وجود ندارد. اگر ارتباط وجود داشته باشد و هر دو با هم هماهنگ شوند، نتایج بسیار متفاوت است.
- در PD دو نفر یکبار با یکدیگر تعامل دارند. تکرار تعاملات در IPD ممکن است به نتایج کاملا متفاوتی منتهی گردد.
- استنتاج بیان شده، تنها راه استنتاج در این مسئله نمی باشد. شاید معقول ترین جواب نیز نباشد.

در نظریهٔ بازیها، PD را می توان یک بازی دونفرهٔ *جمع غیرصفر* درنظرگرفت. در بازی جمع غیرصفر، بازیکنان به طور کامل مخالف و متضاد یکدیگر نیستند و ممکن است که دو بازیکن از یک نتیجه راضی تر از نتیجهٔ دیگر باشند. در بازی جمع غیرصفر موارد زیر را می توان بیان کرد [19,20]: الف) یک زوج استراتژی حداکثر لزوما نقطهٔ موازنه نمی باشد و برعکس، ب) نقاط موازنه لزوما دارای پاداشهای یکسانی نمی باشند و ج) راه حل واضحی برای بازی وجود ندارد. به شکل رسمی می توان بازی PD را به شکل زیر توصیف کرد:

اگر S_i مجموعهٔ استراتژیهای در دستری بازیکن i باشد و n تابع نتیجه $u_1, u_2 \dots u_n$ داشته باشیم که در آن تابع (۴) تابعی باشد که ترکیبی از استراتژیها را بگیرد و پاداش بازیکن i را بازگرداند.

$$u_i : S_1 \times S_2 \times \dots \times S_n \rightarrow P \qquad (۴)$$

آنگاه بازی IPD به شکل زیر بیان می گردد که در آن هر بازیکن دارای مجموعهٔ استراتژیهای زیر است:

$S_1 = \{C,D\} \qquad S_2 = \{C,D\}$

یعنی، هر بازیکن باید بین دو حرکت زیر انتخاب انجام دهد: همکاری (Cooperate) که آن را با C نمایش می دهیم و پشت کردن (Defect) که آن را با D نمایش می دهیم. تابع نتیجه براساس ماتریس نتیجهٔ بیان شده در جدول ۱ محاسبه می گردد.

جدول ۱. ماتریس نتیجهٔ PD

	Cooperate	Defect
Cooperate	R = 3, R =3 Reward for mutual cooperation	S = 0, T = 5 Sucker's payoff Temptation to Defect
Defect	T = 5, S = 0 Temptation to defect Sucker's payoff	P = 1, P = 1 Punishment for mutual defect

براساس این ماتریس، توابع نتیجهٔ زیر به دست می آیند:

u1(C,C) = 3   u2(C,C) = 3
u1(C,D) = 0   u2(C,D) = 5
u1(D,C) = 5   u2(D,C) = 0
u1(D,D) = 1   u2(D,D) = 1

برای رسیدن به معضل[10] نامساوی (۵) باید برقرار باشد:

$$S < P < R < T \qquad (۵)$$

که برای این نمونه S = 0, P =1, R=3, T=5 می باشد.

در بازی N نفره، استراتژی N تایی نقطه موازنه است با فرض این که هیچکدام از بازیکنان دیگر قصد تغییر دادن استراتژیهایشان را نداشته باشند و یا به بیان دیگر هیچ بازیکنی دلیل مثبتی برای تغییر دادن استراتژی خود را نداشته باشد[19]. خروجی متناظر با این مجموعه از استراتژیها، خروجی موازنه نامیده می شود.Nash ثابت کرد که تمام بازیهایی با N بازیکن با مجموعه های استراتژی محدود، حداقل یک استراتژی موازنهٔ خالص یا ترکیبی دارند[20]. در بازی دو نفرهٔ جمع صفر، ممکن است که بیش از یک نقطهٔ موازنه وجود داشته باشد اما همهٔ آنها دارای خروجی یکسانی هستند. اما این موضوع لزوما در بازیهای جمع غیرصفر صحیح نمی باشد. همانگونه که بیان شد، مشکل در این جاست که علائق فردی با علائق گروهی متفاوت است.
در حالتی که بازی فقط یکبار تکرار شود، بازیPD با موازنهٔ Nash حل می شود. در این حالت، همیشه باید طرف مقابل را لو داد. در این بازی زوج عمل (D,D) موازنهٔ Nash است زیرا با فرض این که بازیکن دوم D را انتخاب کند، بهتر است که بازیکن اول D را به جای C انتخاب کند و با فرض این که بازیکن اول D را انتخاب کند، بهتر است که باز هم بازیکن دوم D ر به جای C انتخاب کند. هیچ زوج عمل دیگری نیز موازنهٔ Nash نمی باشد.

هنگامی که این مدل توسعه می یابد، در حالت تکرار این بازی، بازیکنها به طور تکراری با یکدیگر رو به رو می شوند، بدون این که بدانند آیا این بار، بار آخر است یا خیر. پاداش هر بازیکن در این حالت، جمع پاداشهای دریافتی در هر رویارویی می باشد. برای ارزش قائل شدن برای همکاری و برای این که تفاوت بین علاقهٔ فردی و جمعی حفظ شود، نامساوی (۶) نیز باید برقرار باشد:

$$S + T < 2R \qquad (۶)$$

البته در حالت بازی تکراری، آنچه ظرف مقابل در حرکتهای گذشته انجام داده است، ممکن است برروی انتخاب راهی که در حرکت بعد پیش گرفته می شود، تاثیر بگذارد. در این بازی زوج استراتژی (D,D) با نتیجهٔ (1,1) نقطهٔ موازنه است. اما این نقطه برای هر دو بازیکن از (3,3) بدتر است. در اینجا، استراتژی پشت کردن یا D بر استراتژی همکاری یا C غالب است. البته اغلب مردم زوج استراتژی (C,C) را بهترین راه حل می دانند.

۳-۱. مروری بر کارهای جاری در زمینهٔ ساخت استراتژی برای بازی IPD

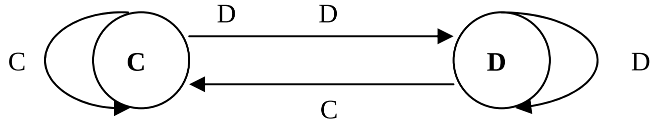
منظور از استراتژی، قانونی است که رفتار (C یا D) را براساس تاریخچهٔ تعاملات انجام شده تغییر می دهد. برای انجام بازی IPD استراتژیهای زیادی طراحی شده است [1,2,3,4,5,6,7,8,9,10,11]. براساس مطالعات انجام شده، می توانیم این استراتژیها را به سه دسته استراتژیهای قطعی، استراتژیهای احتمالاتی و استراتژیهای تکاملی تقسیم کنیم. در ادامه، بعضی از این استراتژیها توضیح داده می شوند.

استراتژیهای قطعی، استراتژیهایی هستند که در آنها در هر مرحله، هر بازیکن براساس یک قانون قطعی عمل می کند. مثالهایی از این استراتژیها در ادامه توضیح داده می شوند:

ALL-C: همیشه همکاری می کند. *(C)

ALL-D: همیشه همکاری نمی کند. *(D)

TIT-for-TAT: در حرکت اول همکاری می کند و سپس بازی قبلی طرف مقابل را انجام می دهد. این استراتژی به شکل یک اتوماتای قطعی محدود (DFA) در شکل ۲ بازنمایی شده است.



شکل ۲. بازنمایی استراتژی TFT توسط اتوماتای قطعی محدود

Per-CD: به شکل متناوب C و سپس D را بازی می کند، یعنی *(CD)

Soft-Majo: همکاری می کند، سپس بیشترین حرکت طرف مقابل را بازی می کند. اگر حرکتهای C و D طرف مقابل مساوی بودند، C را بازی می کند.

---

[10] dilemma

Prober: (DCC) را بازی می کند، سپس اگر طرف مقابل در حرکت ۲ و ۳ همکاری کرده باشد، در تمام حرکتهای دیگر D را بازی می کند، در غیر اینصورت TFT را بازی می کند.

Spiteful: همکاری می کند تا زمانی که با وی مقابله (D) شود، آنگاه همیشه مقابله (D) می کند.

Mistrust: مقابله می کند، آنگاه بازی طرف مقابل را انجام می دهد.

Per_Nasty: به شکل متناوب (DDC) را بازی می کند.

Per_Kind: به شکل متناوب (CCD) را بازی می کند.

Pavlov: هنگامی که به خاطر همکاری پاداش داده می شود یا به خاطر عدم همکاری تنبیه می شود، همکاری می کند و در غیراینصورت همکاری نمی کند [11].

استراتژیهای تصادفی یا احتمالاتی نیز برای بازی IPD وجود دارد[6]. یکی از این استراتژیها REASON می باشد. این استراتژی به شرح زیر است[5]:

- در دور اول با احتمال ۰/۵۶۶۹۶ C را بازی می کنم و با احتمال ۰/۴۳۳۰۴ D را بازی می کنم، آنگاه
- اگر دور اول [C D] باشد، *(DC) را بازی می کنم.
- اگر دور اول [D C] باشد، *(CD) را بازی می کنم.

استراتژی جالب دیگر، ترکیب REASON با TFT می باشد که به آن REASON-TFT [a,1-a] می گویند. این استراتژی به شرح زیر است:

- در دور اول، C را با احتمال a بازی می کنم و در موارد دیگر D را بازی می کنم.
- آنگاه TFT را بازی می کنم.

گروه دیگر از استراتژیهای موجود، استراتژیهای تکاملی هستند. در این استراتژیها، استراتژیهای مختلف امتحان می شوند و بررسی می شود که کدامیک در جمعیت استراتژیهای حریف بهتر عمل می کند. دو پیاده سازی برای این استراتژیها وجود دارد. در پیاده سازی اول، یک بازیکن استراتژیهای مختلف را بازی می کند و نتیجه کار آنها را ضبط می کند. بعد از جمع آوری اطلاعات کافی، آنها بهترین استراتژی را برای بازیهای آینده انتخاب می کنند. در پیاده سازی دوم، گروهی از بازیکنان یکی از استراتژیهای موجود را بازی می کنند. بعد از تعداد کافی انجام بازی، بازیکنی با پایین ترین امتیاز در گروه، استراتژی فعلی خود را کنار می گذارد و استراتژی بالاترین امتیاز در گروه را پی می گیرد. درنهایت، تمام بازیکنان در گروه به یک استراتژی همگرا می شوند[9].

در [17] از یادگیر تقویتی چندعامله[11] در بازی IPD استفاده شده است. در این تحقیق، توانایی تنوعی از عاملهای یادگیرندهٔ Q برای بازی IPD در برابر حریف ناشناخته بررسی می شود. در بعضی از آزمایشات، حریف استراتژی قطعی مانند Tit-for-Tat بوده است و در بعضی دیگر یک یادگیرندهٔ Q بوده است. نتیجهٔ آزمایشات به عمل آمده این بوده است که تمام یادگیرنده های Q یادگرفتند که به شکل بهینه در برابر TFT بازی کنند. بازی کردن در برابر یادگیرندهٔ دیگر مشکل تر بوده است، زیرا تطبیق یادگیرندهٔ دیگر یک محیط غیرثابت ایجاد کرده است و بازیکن دیگر دانش پیش زمینه ای دربارهٔ IPD مانند سیاست طراحی شده برای تشویق همکاری در اختیار نداشته است. در این بازی، یادگیرنده هایی با حافظهٔ بیشتر، جداول جستجو و زمان بندی exploration در بازیهای IPD بهترین عملکرد را داشتند.

۳- ۲. ارزیابی استراتژیهای IPD

به منظور مقایسهٔ استراتژیهای IPD باید بتوان آنها را با یکدیگر مقایسه نمود. بدین منظور سه راه وجود دارد[2,5]:

۱- رو در رویی دو استراتژی (Single confrontation). در انتهای رودر رویی دو استراتژی (به عنوان مثال، بعد از ۱۰۰ دور) امتیازات حاصل توسط هر بازیکن جمع زده می شود و برنده بازیکنی است که امتیاز بیشتری دارد.

۲- تورنمنت: k استراتژی را انتخاب می کنیم، هر استراتژی در برابر تمام استراتژیهای دیگر (از جمله خودش) بازی می کند. امتیازات بازیها جمع زده می شوند. برنده بازیکنی است که بیشترین امتیاز را دارد. استراتژیهای خوب در تورنمنت به خوبی با محیطشان تطابق می یابند، اما اغلب در برابر تغییرات محیطی چندان مستحکم نمی باشند.

۳- تکامل اکولوژیکی (ecological evolution). در این روش ارزیابی، تقلیدی از فرآیند انتخاب طبیعی صورت می گیرد. در نظر بگیرید که جمعیتی از N بازیکن داریم که هرکدام براساس استراتژی به خصوصی بازی می کند. در ابتدا درنظر می گیریم که جمعیت هر استراتژی در اکولوژی با بقیه یکسان است. سپس تورنمنتی برگزار می شود و استراتژیهای خوب تقویت و استراتژیهای بعد تضعیف می شوند. این کار با توزیع مجدد متناسب جمعیتها انجام می شود. این فرآیند که توزیع مجدد نسل نامیده می شود، تا زمانی که ایستایی جمعیت مشاهده گردد (یعنی هیچ تغییری بین دو نسل نباشد) ادامه می یابد. استراتژی خوب، استراتژیی می باشد که برای زمان طولانی تری در جمعیت پایدار بوده و بزرگترین جمعیت را دارد. به عنوان مثال، سه استراتژی A، B و C را در نظر بگیرید که هرکدام دارای ۱۰۰ نفر جمعیت هستند. تورنمنتی انجام می شود (هر بازیکن در برابر ۲۹۹ بازیکن دیگر)، امتیازات هر استراتژی محاسبه می گردد (جمع امتیازات هر عضو جمعیت

---

[11] Multiagent reinforcement learning

استراتژی معین). جمعیت جدید برای هر استراتژی محاسبه می گردد. این جمعیت متناسب با امتیاز به دست آمده می باشد. این نسل دوم است. محاسبه تکرار می شود تا زمانی که جمعیتها ایستا شوند.

در [8] از الگوریتم ژنتیک برای تکامل استراتژیهای بهینه برای بازی PD استفاده شده است. در این تحقیق نشان داده می شود که جمعیتهایی که تکامل می یابند، از خود دو رفتار را نشان می دهند: توانایی دفاع در برابر بازیکنانی که D بازی می کنند و توانایی همکاری با بازیکنانی که همکاری می کنند.

۴. ساختن استراتژی در IPD توسط اتوماتای یادگیر

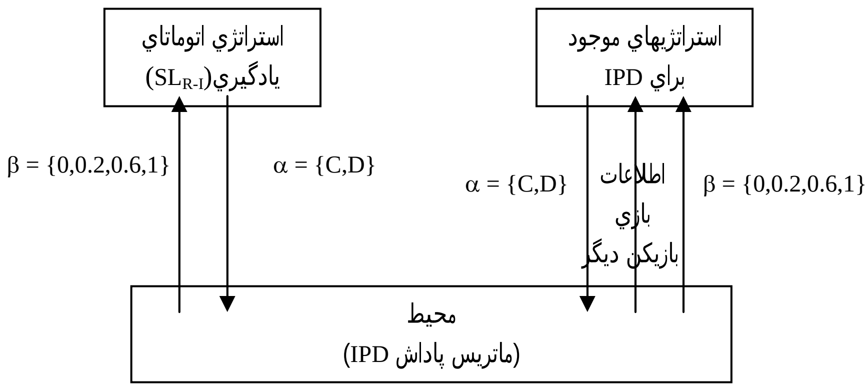
یکی از مسائل مهم تحقیقاتی در زمینه IPD، انتخاب بهترین استراتژی می باشد. در این مقاله، هدف استفاده از اتوماتای یادگیر در ساختن استراتژی برای IPD و ارزیابی این استراتژی می باشد. سوالی که در اینجا مطرح است این است که *آیا الگوریتم یادگیر می تواند به استراتژی مطلوب در بازی IPD همگرا شود؟* بدین منظور ابتدا لازم است که مسئله را به شکل رسمی بازنمایی کنیم. پس در ابتدا چند تعریف ابتدایی بیان می شود و سپس شکل رسمی مسئله و نتایج ارزیابی بیان می گردد. همانگونه که اشاره شد، یک اتوماتا را می توان با پنج تایی $\{\beta,\Phi,F_j,\alpha,G\}$ نمایش داد: مجموعهٔ ورودی $\beta = \{ \beta_1, \beta_2,\dots, \beta_m\}$، مجموعهٔ وضعیت $\Phi = \{\Phi_1, \Phi_2, \dots, \Phi_j\}$، تابع انتقال وضعیت Fj به طوری که $\Phi_j(n+1) = F_j[\Phi_j(n), \beta_j(n)]$، مجموعهٔ عمل $\alpha j = \{\alpha_1,\alpha_2,\dots,\alpha_j\}$ و تابع خروجی G_j به طوری که در آن $\alpha_j(n) = G_j[\Phi_j(n)]$ می باشد.

بازی $\alpha(n)$ مجموعه ای از استراتژیها می باشد که توسط اتوماتا در مرحلهٔ n انتخاب می شود. بازی $\alpha(n)$ به شکل بردار N تایی $\alpha (n) = [\alpha_1(n), \alpha_2(n),\dots\alpha_N(n)]$ بیان میشود. خروجی بازی $\beta(n)$ برداری N تایی می باشد که عناصر آن $\beta_1(n)$، $\beta_2(n)$، ....، $\beta_N(n)$ می باشد که $\beta_i(n)$ متناظر با اتوماتای A_j می باشد. N اتوماتا در بازی Γ شرکت می کنند، اگر احتمال خروجی $\beta(n)$ به خاطر $\alpha(n)$ باشد. اتوماتای Aj می تواند یکی از انواع اتوماتاها باشد. به عنوان مثال، می توان از اتوماتای با ساختار ثابت مانند اتوماتای Tseltin و Krinsky استفاده نمود یا از شماهای L_{R-I} یا L_{R-P} استفاده نمود.

به منظور ارزیابی اتوماتای یادگیر در IPD از اتوماتای یادگیر با ساختار متغیر با شمای L_{R-I} که در محیط مدل S عمل می کند، استفاده شد. بدین ترتیب اگر اتوماتا را با سه تایی (α,T,β) نشان دهیم. این اتوماتا در هر دور از بازی می تواند یکی از دو عمل همکاری (C) یا عدم همکاری (D) را انجام دهد و در نتیجه مجموعهٔ اعمال این اتوماتا $\alpha = \{C,D\}$ میباشد. مجموعهٔ پاسخهای نرمال شدهٔ محیط که محیطی از نوع S می باشد، برابر است با $\beta = \{0,0.2,0.6,1\}$. شمای تقویتی این اتوماتا، شمای SL_{R-I} است که احتمالات انجام دو عمل این اتوماتا را براساس معادلهٔ (۷) بهنگام سازی می کند:

$$P_2(n+1) = P_2(n) - a\,(1- \beta(n))\, P_2(n) \qquad \alpha = D \qquad (۷)$$
$$P_1(n+1) = P_1(n) + a\,(1- \beta(n))\, P_2(n) \qquad \alpha = C$$

استراتژی یا استراتژیهای دیگری می توانند از انواع دیگر یا از نوع اتوماتای یادگیر وجود داشته باشند که با اتوماتا بازی کنند.



شکل ۳: بازی اتوماتای یادگیری با استراتژیهای دیگر IPD

استراتژی اتوماتای یادگیر بدین شکل عمل می کند که در هر مرحله از بازی، بازیکن اتوماتا براساس احتمال اعمال، یکی از دو عمل C یا D انتخاب می کند و انجام می دهد. براساس پاسخ دریافتی از محیط که براساس ماتریس نتیجه محاسبه می شود، احتمالات انجام اعمال براساس شمای تقویتی SL_{R-I} بهنگام سازی می گردد. به منظور ارزیابی این استراتژی، براساس روشهای موجود برای ارزیابی استراتژی IPD، استراتژی اتوماتای یادگیر در تقابلهای فردی و همچنین در تورنمنت با استراتژیهای دیگر IPD شرکت کرد که نتایج حاصله در بخش بعد بیان می گردد.

۵. نتایج حاصل از مسابقه های تکی استراتژی اتوماتای یادگیر

به منظور مقایسهٔ استراتژی اتوماتای یادگیر با استراتژیهای دیگر یکی از راههای ممکن رو در رویی دو استراتژی (Single confrontation) است. در انتهای رودر رویی دو استراتژی (به عنوان مثال، بعد از ۱۰۰ دور) امتیازات حاصل توسط هر بازیکن جمع زده می شود و برنده بازیکنی است که امتیاز بیشتری دارد. در این ارزیابی استراتژی اتوماتای یادگیر با استراتژیهای ALL-C، ALL-D، Per-CD، Soft-Majo ، Prober، Spiteful، Mistrust، Per_Nasty و Per_Kind مسابقه تکی داد. در این مسابقه ها سعی شد تا هر مسابقه با استفاده از مقادیر مختلف پارامتر a در شمای تقویتی SL_{R-I} نتایج بررسی گردد. در این بازیها، به ازاء مقادیر مختلف a نتیجهٔ مسابقه متفاوت بود و در یک مقدار مشخص از پارامتر تقویتی a اتوماتا به نتیجهٔ مناسبی در برابر استراتژی دیگر دست می یافت. در جدول ۲ بهترین نتایج حاصل از مسابقه ها دیده می شود.

استراتژی اتوماتای یادگیر در بازی مقابل ALL-D به نقطهٔ موازنه در PD یعنی (D,D) با نتیجهٔ (1,1) همگرا گردید. با توجه به این که استراتژی اتوماتای یادگیر یک استراتژی احتمالاتی می باشد، در صورتی که اندکی دیر به استراتژی مناسب همگرا شود، با تفاضل زیادی به استراتژی ALL-D می بازد. بعد از آزمایش مقادیر مختلف پارامتر a این استراتژی با مقدار ۰/۵۷ پارامتر a به نتیجهٔ مساوی دست یافت.

در مقابله های تکی، یکی از نکات قابل توجه اهمیت تنظیم پارامتر پاداش a در نتیجهٔ حاصل از بازی می باشد. در بعضی از مقادیر a، در اکثر موارد استراتژی اتوماتای یادگیر بازنده بود به همین منظور بازیها با مقادیر مختلف a انجام شد تا نتیجهٔ مناسب بدست آید. در این جا بازهم به مسئلهٔ نتیجهٔ فردی و نتیجهٔ گروهی می توان اشاره کرد. در بعضی از مقادیر a بدون توجه به نتیجهٔ حاصل برای دو طرف، مجموع امتیازاتی که دو طرف کسب می کردند از حالت برد اتوماتای یادگیر بیشتر بود.

جدول ۲. بهترین نتایج حاصل از بازی اتوماتای یادگیر با استراتژیهای دیگر در ۱۰۰۰ تکرار

ردیف	استراتژی حریف	مقدار پارامتر a	امتیاز SLR-I	امتیاز حریف
۱	ALL-C	۰/۲۵	۵۰۰۰	۰
۲	ALL-D	۰/۵۷	۱۰۰۰	۱۰۰۰
۳	TFT	۰/۲	۱۰۱۳	۱۰۰۸
۴	PerCD	۰/۴	۳۰۰۰	۵۰۰
۵	Spiteful	۰/۵۶۵	۱۰۰۰	۱۰۰۰
۶	Mistrust	۰/۲	۱۰۰۹	۱۰۰۹
۷	SoftMajo	۰/۴۸	۱۰۰۴	۹۹۹
۸	Prober	۰/۳	۱۰۱۱	۱۰۰۱

با توجه به این که L_{R-I} با مقدار ویژه ای از پارامتر پاداش a نتیجهٔ تقریبا مناسبی را در برابر هر استراتژی به دست می آورد، می توان مسئلهٔ مدل نمودن حریف[12] را در این بازی معرفی نمود. در این حالت بازیکن می تواند بدون اطلاع از استراتژی حریف، استراتژی وی را مدل کند و براساس آن مقدار مناسب a را تنظیم نماید. در حالتی که هر دو بازیکن از استراتژی L_{R-I} استفاده می کنند و یک بازی اتوماتا به وجود می آید، هر دو به بازی (C,C) همگرا می شوند که این نتیجه نمی تواند نشان دهندهٔ معقولیت گروهی باشد.

۶. ارزیابی استراتژی اتوماتای یادگیر در تورنمنت

یکی دیگر از روشهای ارزیابی استراتژی اتوماتای یادگیر در بازی IPD شرکت دادن آن در تورنمنتی با حضور سایر استراتژیها می باشد. در این تورنمنت استراتژی LA به همراه ۱۰ استراتژی دیگر به رقابت پرداخت. هر استراتژی در برابر تمام استراتژیهای دیگر (از جمله خودش) بازی می کند. امتیازات مسابقات جمع زده می شود و برنده بازیکنی است که بیشترین امتیازات را دارد. بعد از پیاده سازی تورنمنت فوق به ازا دو مقدار پارامتر پاداش a نتایج جدول ۳ و ۴ به دست آمد.

در این تورنمنت نیز پارامتر پاداش اتوماتا نقش مهمی را در نتیجه دارد. اما برخلاف مسابقه های تکی، اتوماتا با مقدار پایین پارامتر پاداش نتیجهٔ بهتری را کسب می کند. می توان گفت که استراتژی ALLD که بهترین نتیجه را در این تورنمنت کسب کرده است، تطابق بیشتری با محیط می یابد ولی نسبت به تغییرات محیطی استحکام کمتری دارد. برای این که استراتژی SL_{R-I} نتایج بهتری کسب کند، می توان گفت که عواملی مانند طولانی تر شدن زمان تورنمنت برای یادگیر بهتر و تنظیم سازگار پارامتر پاداش a در برابر هر استراتژی، می تواند منجر به نتایج بهتری شود.

۷. نتیجه گیری

---

[12] oponent modelling

در این مقاله، مسئلهٔ به کارگیری اتوماتای یادگیر به عنوان الگوریتم مورد استفاده برای تصمیم گیری بازیکنان در بازی IPD مورد بررسی و ارزیابی قرار گرفت. پرسش اصلی این بود که آیا این الگوریتم یادگیر می تواند به استراتژی مطلوب در IPD همگرا شود؟ و آیا بازیکنی که از اتوماتای یادگیر برای تصمیم گیری در بازی استفاده می کند، می تواند به شکل گروهی یا فردی معقول رفتار کند؟ به منظور ارزیابی اتوماتای با ساختار متغیر که از شمای L_{R-I} استفاده می کند، به کارگرفته شد. بازیکنی که از این الگوریتم استفاده می کرد در مسابقه های تکی با استراتژیهای دیگر و در تورنمنتی با حضور استراتژیهای دیگر شرکت نمود. نتایج حاصل از ارزیابیها نشان می دهند که اتوماتای یادگیر ساختار متغیر مورد استفاده در ارزیابی در بازی با اتوماتای یادگیر دیگر به موازنهٔ Nash همگرا می شود. همچنین در مسابقه های تکی با استراتژیهای دیگر قطعی یا احتمالاتی طراحی شده برای IPD این شمای یادگیر با انتخاب پارامترهای مناسب به نتایج مطلوب گروهی و فردی دست می یابد. در مسابقات انجام شده، انتخاب پارامتر پاداش a در شمای تقویتی بهنگام سازی احتمالات نقش بسیار مهمی را در به دست آوردن نتایج فردی یا گروهی مناسب بازی می کرد. اما در تورنمنت این استراتژی رتبهٔ مناسبی را به دست نیاورد. در این تورنمنت نیز انتخاب پارامتر پاداش در شمای تقویتی نقش مهمی را ایفا می کرد. در تورنمنت برخلاف مسابقه های تکی، اتوماتا با مقدار کمتر پارامتر پاداش a نتیجهٔ بهتری را کسب کرد. نتیجهٔ مقدماتی حاصل از عملکرد اتوماتا در تورنمنت می تواند این باشد که استراتژی اتوماتا نسبت به تغییرات محیطی استحکام بیشتری دارد. در تورنمنت عواملی مانند طولانی تر شدن بازی و تطبیق سازگار پارامتر پاداش دربرابر استراتژیهای مختلف شرکت کننده باعث بهتر شدن عملکرد اتوماتا می گردد.

جدول ۳. نتیجهٔ تورنمنت با مقدار پارامتر پاداش a=0.0001 برای استراتژی اتوماتای یادگیر

رتبه	نام استراتژی	امتیاز
۱	ALLD	۲۹۹۸۴
۲	PerCD	۲۷۹۸۸
۳	Prober	۲۴۸۸۷
۴	PerNasty	۲۳۲۰۶
۵	SoftMajo	۲۳۱۶۱
۶	TFT	۲۱۵۴۷
۷	Spiteful	۲۰۹۴۲
۸	**SL_{R-I}**	**۲۰۳۶۴**
۹	Perkind	۱۹۲۹۳
۱۰	Mistrust	۱۸۵۴۸
۱۱	ALLC	۱۸۰۳۳

جدول ۴. نتیجهٔ تورنمنت با مقدار پارامتر پاداش a=0.3 برای استراتژی اتوماتای یادگیر

رتبه	نام استراتژی	امتیاز
۱	ALLD	۳۲۰۰۴
۲	PerCD	۲۹۴۹۷
۳	Prober	۲۸۱۶۵
۴	PerNasty	۲۴۵۱۵
۵	SoftMajo	۲۲۸۲۶
۶	Spiteful	۲۲۴۹۷
۷	TFT	۲۲۱۶۸
۸	Perkind	۲۰۶۷۰
۹	ALLC	۱۹۴۷۶
۱۰	Mistrust	۱۹۰۱۸
۱۱	**SL_{R-I}**	**۱۶۵۲۱**

مراجع

1- Axelrod, A: “The Evolution of Cooperation”. New York, Basic Books (1984)

2- Beaufils, B., J. Delahaye, and Mathieu, P., "Complete Classes of Strategies for the Classical Iterated Prisoner’s Dilemma", Evolutionary Programming VII Proceedings, Lecture Notes in Computer Science 1447 (1998)

3- Hofstadter, D.R: “Metamagical Themas”, New York, Basic Books (1985)

4- Brembs, B., "Chaos, cheating and cooperation: potential solutions to the Prisoner’s Dilemma", In: Proc. of the OIKOS 76:1, Copenhagen (1996)

5- Delahaye J., Mathieu, P., "Complex Strategies in the Iterated Prisoner’s Dilemma", In: Proc. of the Chaos and Society 94 (1994)

6- Delahaye J. and Mathieu, P., "The Iterated Lift Dilemma or How to Establish Meta-Cooperation with your opponent?", In: Proc. of the Chaos and Society (1996)

7- Delahaye J. and Mathieu, P., "Random Strategies in a Two Levels Iterated Prisoner’s Dilemma: How to avoid conflicts?", In: Proc. of the ECAI’96 (1996)

8- Golbeck, J., "Evolving Strategies for the Prisoner's Dilemma", Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation (2002) (299-306)

9-Ternasky, J.:” Prisoner’s Dilemma: Game Overview and Strategies”, http://windowsxp.devx.com/ PD/articles (2001)

10- Beaufils, B., J. Delahaye, and Mathieu, P., "Our Meeting with Gradual: A Good Strategy for the Classical Iterated Prisoner’s Dilemma", In: Proc. of the Artificial Life V (1997)

11- Kraines, D. and Kraines, V., "Evolution of Learning among Pavlov Strategies in a Competitive Environment with Noise," Journal of Conflict Resolution, (1995) (439-466)

12- Mars, P., Chen, J. R. and Nambir, R., **“**Learning Algorithms: Theory and Applications in Signal Processing”, Control and Communications, CRC Press, Inc(1996).

13- Narenrdra, K., S. and M. A. L. Thathachar: “Learning Automata: An Introduction”, Prentice Hall (1989).

14- Lakshmivarahan, S., “Learning Algorithms: Theory and Applications”, New York, Springer Verlag, (1981).

15- Meybodi, M. R. and S. Lakshmivarahan: “Optimality of a Generalized Class of Learning Algorithm”, Information Science, Vol. 28 (1982) 1-20

16- Meybodi, M. R. and S. Lakshmivarahan: “On a Class of Learning Algorithms which have a Symmetric Bahavior under Success and Failure”, Lecture Notes in Statistics, Springer Verlag (1984) 145-155.

17- Sandholm T. S. and Robert H. Crites, “Multiagent Reinforcement Learning in the Iterated Prisoner's Dilemma”, Biosystems Journal, 37 (1995) 147-166

18- Kuhn, S. T.," Prisoner’s Dilemma", Stanford Encyclopedia of Philosophy (2000)

19- Osborne, M. J.: “An introduction to game theory”, Oxford University Press (2001)

20- Principia Cybernetica Web: “Game Theory”, http:// pespmc1.vub.ac.be/ ASC/ GAME_THEOR.html (1999)