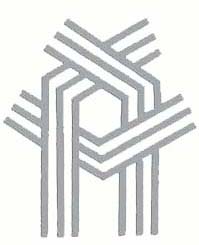
1ـ اولين صفحه: سفيد

2ـ دومين صفحه: بسم الله الرحمن الرحيم (در وسط صفحه)



**دانشگاه علوم اقتصادي**

**دانشکده مدیریت نهادهای اقتصادی**

**گروه مهندسی دانش و علوم تصمیم**

#### پايان نامه کارشناسي ارشد رشته­ مهندسی دانش وعلوم تصمیم

کنترل هوشمند موجودی در زنجیره تأمین مبتنی بر سیستم‌های چند عامله

**استاد راهنما:**

دکتر سید امیررضا ابطحی

**استاد مشاور:**

**دکتر کاوه خلیلی دامغانی**

**پژوهشگر:**

**احمدرضا ناظمی سجزیی**

**اسفند ماه 1392**

**اظهارنامه دانشجو**

موضوع پایان‌نامه : کنترل هوشمند موجودی در زنجیره تأمین مبتنی بر سیستم‌های چند عامله

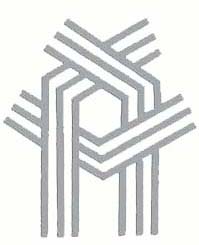
استاد راهنما: دکتر سید امیررضا ابطحی

این‌جانب احمدرضا ناظمی دانشجوي دوره كارشناسي ارشد رشته علوم کامپیوتر گرايش مهندسی دانش و علوم تصمیم دانشگاه علوم اقتصادي به شماره دانشجويي 901517010 گواهي می‌نمایم كه تحقيقات ارائه‌شده در اين پایان‌نامه توسط شخص این‌جانب انجام‌شده و صحت و اصالت مطالب نگارش شده مورد تأیید می‌باشد و در موارد استفاده از كار ديگر محققان به مرجع مورد استفاده اشاره شده است. به علاوه گواهي می‌نمایم كه مطالب مندرج در پایان‌نامه تاكنون براي دريافت هيچ نوع مدرك يا امتيازي توسط این‌جانب يا فرد ديگري ارائه نشده است و در تدوين متن پایان‌نامه چارچوب مصوب دانشکده را به طور كامل رعايت کرده‌ام.

**امضاء دانشجو:**

**تاريخ:**

**کليه حقوق مادی و معنوی مترتب بر نتايج مطالعات، ابتکارات و نوآوری‌های ناشی از تحقيق، همچنين چاپ و تکثير، نسخه‌برداری، ترجمه و اقتباس از اين پایان‌نامه کارشناسی ارشد، برای دانشگاه علوم اقتصادی محفوظ است.نقل مطلب با ذکر منبع بلامانع است.**



**دانشگاه علوم اقتصادي**

**دانشکده مدیریت نهادهای اقتصادی**

**پایان‌نامه کارشناسي ارشد رشته مهندسی دانش و علوم تصمیم آقاي احمدرضا ناظمی تحت عنوان**

کنترل هوشمند موجودی در زنجیره تأمین مبتنی بر سیستم‌های چند عامله

در تاريخ …… توسط هیئت داوران زير بررسي و به تصويب نهايي رسيد

اعضای هیئت داوران نام و نام خانوادگی رتبه علمی امضاء

1. استاد راهنما دکتر ………… .................. امضا
2. استاد مشاور دکتر ………… ................. امضا
3. استاد داور دکتر ………… ................... امضا

**امضاء نماينده تحصيلات تکميلي**

6-‌ هفتمين صفحه: سپاسگزاري (اختياري)

7- هشتمين صفحه: تقديم اثر (اختياري)

چکیده

چالش بزرگ مدیریت سفارش دهی در زنجیره تأمین اتخاذ یک راهبرد سفارش دهی مناسب در هر رده از زنجیره تأمین برای کاهش هزینه انبار است. در این پایان‌نامه عامل‌های هوشمند نرم‌افزاری وظیفه تصمیم‌گیری پویا را برای پیدا کردن یک راهبرد مناسب سفارش دهی بر عهده می‌گیرند. در گام اول ما به سراغ الگوریتم یادگیری Q[[1]](#footnote-1) می‌رویم و ضمن نشان دادن برتری آن نسبت به الگوریتم‌هایی که سیاست ثابت سفارش دهی را اتخاذ می‌کنند به بیان ضعف‌های آن در مدیریت سفارش دهی با فضای حالت بزرگ همچون سیستم‌های چند عامله می‌پردازیم .در گام بعد راهکاری را برای عمومیت بخشیدن به الگوریتم یادگیری Q با استفاده از تقریب زدن بیان می‌کنیم و نشان می‌دهیم که استفاده از الگوریتم بهینه‌شده با غلبه بر مشکل فضای حالت وسرعت یادگیری پایین، می‌تواند به خوبی روش‌های قبلی راهبرد بهینه را برای مدیریت سفارش دهی پیدا کند. این دست آورد سبب میشود که راه برای استفاده از این الگوریتم در مسایل دنیای واقعی که شامل شبکه ای از زنجیره های تامین با رده های مختلف سفارش دهی هستند هموار شود.

**کلمات کلیدی:** کنترل هوشمند موجودی، تعمیم یادگیری تقویتی، تخمین تابع ارزش، رگرسیون وزن دار محلی.

فهرست­ مطالب  
**عنوان**   **صفحه**

[پيشگفتار:..................................................................................................................................................................................................................................2](#_Toc392242392)

[1 کلیّات 5](#_Toc392242393)

[1.1 مقدمه...................................................................................................................................................................................................5](#_Toc392242394)

[1.2 بیان مسئله...........................................................................................................................................................................................5](#_Toc392242395)

[1.1.1 معضل ابعاد بالا 7](#_Toc392242396)

[1.3 اهمیت موضوع و مرور ادبیات...................................................................................................................................................8](#_Toc392242397)

[1.3.1 اهمیت موضوع 8](#_Toc392242398)

[1.3.2 مرور ادبیات.......................................................................................................................................................................10](#_Toc392242399)

[1.4 ضرورت و کاربرد تحقیق..........................................................................................................................................................13](#_Toc392242400)

[1.5 فرضیه‌ها...........................................................................................................................................................................................14](#_Toc392242401)

[2 مرور ادبیات تحقیق 14](#_Toc392242402)

[2.1 مقدمه..................................................................................................................................................................................................14](#_Toc392242403)

[2.2 زنجیره تأمین...................................................................................................................................................................................15](#_Toc392242404)

[2.2.1 مدیریت زنجیره تأمین 16](#_Toc392242405)

[2.2.2 کنترل موجودی 18](#_Toc392242406)

[**2.2.3** مدیریت سفارش در زنجیره تأمین 20](#_Toc392242407)

[2.3 یادگیری تقویتی...............................................................................................................................................................................20](#_Toc392242408)

[2.3.1 سیاست یا راهبرد بهینه 22](#_Toc392242409)

[2.3.2 محاسبه پاداش در بلندمدت 22](#_Toc392242410)

[2.3.3 خاصیت مارکوف 23](#_Toc392242411)

[2.3.4 تابع پاداش: .........................................................................................................................................................................23](#_Toc392242412)

[2.3.5 مسئله مارکوف 24](#_Toc392242413)

[2.4 الگوریتم یادگیری Q ..................................................................................................................................................................25](#_Toc392242414)

[2.4.1 شرح الگوریتم ..................................................................................................................................................................28](#_Toc392242415)

[2.5 یادگیری تقویتی در مسئله مدیریت سفارش 34](#_Toc392242416)

[2.5.1 حالت سیستم............................................................................................. 34](#_Toc392242417)

[2.5.2 پاداش ..............................................................................................................36](#_Toc392242418)

[2.5.3 تابع ارزش.......................................................................................................................................................................37](#_Toc392242419)

[2.5.4 قانون سفارش دهی 37](#_Toc392242420)

[2.5.5 راهبرد عامل‌ها 37](#_Toc392242421)

[3 مدل ارایه شده 39](#_Toc392242422)

[3.1 مقدمه.................................................................................................................................................................................................39](#_Toc392242423)

[3.2 پیش‌فرض‌ها.................................................................................................................................................................................40](#_Toc392242424)

[3.3 موانع موجود.................................................................................................................................................................................41](#_Toc392242425)

[3.3.1 هوشمند سازی عامل 41](#_Toc392242426)

[3.3.2 کمبود داده‌های آموزشی 42](#_Toc392242427)

[3.3.3 دانش اولیه کم راجع به محیط 42](#_Toc392242428)

[3.3.4 فضای حالت بسیار بزرگ 42](#_Toc392242429)

[3.4 تخمین تابع ارزش..........................................................................................................................................................................43](#_Toc392242430)

[3.5 تخمین زدن جدول حالت- عمل..................................................................................................................................................44](#_Toc392242431)

[3.5.1 کاهش خطای تقریب زدن 45](#_Toc392242432)

[3.6 رگرسیون وزن دار محلی..........................................................................................................................................................48](#_Toc392242433)

[3.6.1 تابع کرنل..............................................................................................................................................................................50](#_Toc392242434)

[3.6.2 محاسبه فاصله در رگرسیون وزن دار محلی 52](#_Toc392242435)

[3.7 الگوریتم پیش‌بینی کننده HEDGER....................................................................................................................................54](#_Toc392242436)

[3.8 الگوریتم آموزشی HEDGER................................................................................................................................................56](#_Toc392242437)

[4 پیاده‌سازی الگوریتم 59](#_Toc392242438)

[4.1 بهبود وپیاده سازی الگوریتم های ژنتیک و یادگیری Q 59](#_Toc392242439)

[4.1.1 بیان جزیات مساله 59](#_Toc392242440)

[4.1.2 پیاده‌سازی با الگوریتم ژنتیک 60](#_Toc392242441)

[4.1.3 پیاده‌سازی با یادگیری Q 61](#_Toc392242442)

[4.1.4 نتایج حل مساله به روش ژنتیک و الگوریتم Q 64](#_Toc392242443)

[4.1.5 معایب این دو روش 66](#_Toc392242444)

[4.2 الگوریتم ارایه شده برای برطرف کردن معایب روشهای قبلی 70](#_Toc392242445)

[4.2.1 بهینه‌سازی الگوریتم HEDGER برای بازی نوشابه 70](#_Toc392242446)

[4.3 نتایج..................................................................................................................................................................................................71](#_Toc392242447)

[5 نتیجه‌گیری 78](#_Toc392242448)

[5.1 کارهای آینده....................................................................................................................................................................................78](#_Toc392242449)

[6 منابع 80](#_Toc392242450)

[7 پیوست 84](#_Toc392242451)

فهرست شکل‌ها

[شکل 1 عامل های موجود و جریان داده ، کالا 15](file:///E:\UES\Course\MyWorkMehr92\Doc\Final.docx#_Toc392242288)

[شکل 2 - زنجیره تأمین چند سطحی 20](#_Toc392242289)

[شکل 3 نحوه تعامل عامل و محیط 21](#_Toc392242290)

[شکل 4 خانه‌ای با 5 اتاق که ربات در یکی از اتاق‌های آن قرار دارد و می‌خواهد با استفاده از یادگیری تقویتی مسیر خارج شدن از اتاق را بیابد 31](#_Toc392242291)

[شکل 5 حالت‌های مسئله پیدا کردن مسیر خروج و پاداش هر عمل در هر حالت 31](#_Toc392242292)

[شکل 6 شمای برنامه برای یافتن راهبرد بهینه خروج از اتاق 33](#_Toc392242293)

[شکل 7 نحوه کد کردن حالت انبار 37](#_Toc392242294)

[شکل 8 تابع کرنل Gaussian با پهنای باند متفاوت 52](#_Toc392242295)

[شکل 9 نمودار کاهش مقدار آلفا 62](#_Toc392242296)

[شکل 10-نمودار هزینه زنجیره طی 35 هفته باراهبرد‌های سفارش دهی متفاوت 64](#_Toc392242297)

[شکل 11-نمودار هزینه خرده‌فروش در طول 35 هفته در بازی نوشابه 66](#_Toc392242298)

[شکل 12- نمودار هزینه انبار عمده‌فروش در طول 35 هفته در بازی نوشابه 67](#_Toc392242299)

[شکل 13 نمودار هزینه کارخانه در طول 35 هفته در بازی نوشابه 68](#_Toc392242300)

[شکل 14 هزینه کل زنحیره وقتی عاملها به صورت مستقل تصمیم میگیرند 72](#_Toc392242301)

[شکل 15 نمودار هزینه خرده فروش بصورت کنترل غیر متمرکز 73](#_Toc392242302)

[شکل 16 نمودار هزینه عمده فروش بصورت کنترل غیر متمرکز 73](#_Toc392242303)

[شکل 17 نمودار هزینه کارخانه بصورت کنترل غیر متمرکز 74](#_Toc392242304)

[شکل 18 نمودار وضعیت انبار خرده فروش بصورت کنترل غیر متمرکز 75](#_Toc392242305)

[شکل 19نمودرا وضعیت انبار عمده فروش بصورت کنترل غیر متمرکز 75](#_Toc392242306)

[شکل 20 نمودار وضعیت انبار کارخانه بصورت کنترل غیر متمرکز 76](#_Toc392242307)

# 

[الگوریتم 1 یادگیری تقویتی 30](#_Toc392240800)

[الگوریتم 2 رگرسیون وزن دار محلی 49](#_Toc392240801)

[الگوریتم 3 رگرسیون وزن دار محلی با استفاده از k نزدیکترین همسایه 55](file:///E:\UES\Course\MyWorkMehr92\Doc\Final.docx#_Toc392240802)

[الگوریتم 4 تخمین زننده HEDGER 56](file:///E:\UES\Course\MyWorkMehr92\Doc\Final.docx#_Toc392240803)

[الگوریتم 5 q-learning با استفاده ازHEDGER 57](#_Toc392240804)

[الگوریتم 6شبه کد یادگیری تقویتی استفاده‌شده در (Chaharsooghi, Heydari, & Zegordi, 2008) 63](#_Toc392240805)

# پيشگفتار:

استفاده از الگوریتم یادگیری تقویتی در مسایل مدیریت سفارش دهی در زنجیره تامین به علت فضای حالت بزرگ و همچنین زمان یادگیری به ندرت مورد توجه محققان قرار گرفته بود. از این رو نیاز بود تا این مشکلات به نحوی برطرف شود تا زمینه استفاده از این الگوریتم پویا در مسایل سفارش دهی دنیای واقعی فراهم شود .در این پایان‌نامه ما به معرفی یک راهکار جدید برای سفارش دهی هوشمند در مسئله کنترل زنجیره تأمین توسط عامل‌های نرم‌افزاری در یک سیستم چند عامله شامل سه رده که در هر رده یک عامل خودمختار[[2]](#footnote-2) وجود دارد می‌پردازیم.در ابتدا با معرفی زنجیره تأمین به اهمیت مسئله مدیریت سفارش دهی برای کنترل موجودی در زنجیره تأمین اشاره می‌کنیم و پس از آن مروری کوتاه بر راهکارهای ارائه‌شده برای هوشمند سازی مدیریت سفارش دهی می‌پردازیم، همچنین در این فصل تکنیک یادگیری تقویتی را معرفی می‌کنیم و یادگیری Q را به عنوان الگوریتمی برای یافتن بهترین ارزش حالت-اقدام معرفی می‌کنم در **فصل سوم** مسئله بازی نوشابه را با استفاده از الگوریتم ژنتیک و الگوریتم یادگیری Q حل می‌کنیم و ضمن نشان دادن کارایی بالای الگوریتم Q نسبت به ژنتیک در مسایل مدیریت سفارش دهی هوشمند به بیان ضعف این الگوریتم در مسایل با فضای حالت بزرگ به خصوص سیستم‌های چند عامله می‌پردازیم همچنین راهکارهای برطرف کردن ضعف الگوریتم Q را درقالب مدل ارایه شده نشان می دهیم، در این مدل از الگوریتم HEDGER برای تخمین تابع ارزشQ در یادگیری Q استفاده می‌کنیم و آن را برای مدیریت سفارش­دهی در سیستم‌های چند عامله بهینه می‌کنیم و در چهارم نتیجه حاصل از استفاده از الگوریتم HEDGER بهینه‌شده را با نتیجه حاصل از الگوریتم یادگیری Q مقایسه می‌کنیم.در پایان خلاصه ای از کاهای انجام شده ونتایج حاصل را بیان می کنیم.

**فصل اول**

**کلیات**

# کلیّات

## مقدمه

**در این فصل کلیاتی که در این پایان‌نامه مطرح می‌شود را بیان می‌کنیم، در ابتدا به بیان مسئله مدیریت سفارش****[[3]](#footnote-3) در زنجیره تأمین[[4]](#footnote-4) می‌پردازیم سپس انجام این وظیفه را توسط عامل هوشمند بررسی می‌کنیم، بعد از آن اهمیت موضوع را بیان می‌کنیم و به معرفی تحقیقاتی که در این زمینه انجام‌شده می‌پردازیم، ضرورت و کاربرد این مسئله و همچنین فرضیاتی که در حل این مسئله مطرح می‌شود در پایان این فصل بیان می‌شود.**

## بیان مسئله

**مسئله کنترل موجودی[[5]](#footnote-5) شامل رویکرد یک پارچه برنامه‌ریزی و کنترل انبار در شبکه‌ای از سازمان‌های همکار از مبدأ تهیه مواد اولیه تا رسیدن محصول به دست مشتری نهایی است. در این پایان‌نامه قصد داریم مسئله کنترل موجودی در زنجیره تأمین را به صورت مجموعه‌ای از عامل‌های هوشمند[[6]](#footnote-6) در یک سیستم چند عامله[[7]](#footnote-7) بررسی کنیم که در آن عامل‌ها سعی می‌کنند با همکاری یکدیگر و استفاده از تکنیک‌های یادگیری ماشین[[8]](#footnote-8) مقدار بهینه‌ای برای سفارش کالا به بالاسری خود در زنجیره پیدا کنند. به عنوان مثال در یک زنجیره تأمین خطی (بر خلاف شبکه‌ای) با سه سطح، سه عامل تصمیم‌گیرنده و جود دارد[[9]](#footnote-9) که هر کدام مسئول سفارش دادن مقدار بهینه برای پر کردن (جایگزین کردن)[[10]](#footnote-10) موجودی انبار خود است. هدف بلندمدت همه عامل‌ها در این سیستم افزایش میانگین پاداش و همچنین نگه‌داشتن وضعیت انبار نزدیک سطح صفر است، به طوری که عامل با رسیدن تقاضای جدید از پایین‌دستی دچار کمبود[[11]](#footnote-11) و همچنین موجودی اضافه[[12]](#footnote-12) در انبار نشود. سفارش بهینه در چنین سیستمی به علت عدم قطعیت نهفته در پارامترهای آن (مقدار تقاضای محصول توسط مشتری، زمان رسیدن مشتری، مدت زمان سفارش محصول تا رسیدن آن و اثر شلاقی[[13]](#footnote-13)** (Lee & Whang, 1999)(Lee & Wu, 2006) **بسیار سخت و پیچیده است** (Chaharsooghi, Heydari, & Zegordi, 2008)**. در این مسئله ما از الگوریتم یادگیری تقویتی[[14]](#footnote-14) برای هوشمند سازی عامل‌های تصمیم گیر استفاده می‌کنیم.**

**هدف این الگوریتم پیدا کردن نگاشت بهینه از** حالت سیستم **به** عمل بهینه **به این معنی که در پایان اجرای الگوریتم از هر حالت (وضعیت انبار) عمل مناسب(مقدار بهینه سفارش کالا) بدست می آید. و اساس آن جستجوی فضای حالت با ترکیبی از** حرکات تصادفی**[[15]](#footnote-15) واستفاده از** عمل بهینه یافته شده تا کنون**[[16]](#footnote-16) برای رسیدن به جواب است.** (Q-learning)

**یکی از بحث‌های چالش‌برانگیز که در استفاده از این الگوریتم در سیستم‌های چند عامله وجود دارد این است که هر عامل باید وضعیت[[17]](#footnote-17) همه عامل‌های موجود را در فضای حالتشان[[18]](#footnote-18) در نظر بگیرد (که این عمل باعث افزایش شدید فضای حالت برای این نوع یادگیری می‌شود) وجود یک محیط پویا این مشکل را دو چندان می‌کند، چرا که هر عامل باید دیگر عامل‌ها را به عنوان جزیی از محیط در نظر بگیرد. و بنابراین هر عامل نیاز دارد برای در نظر گرفتن سودمندی هر حالت جدولی بزرگ از فضای حالت خود و دیگر عوامل را در حافظه نگهداری کند[[19]](#footnote-19). یکی از راه‌هایی که در کتاب** (Sutton, 1998) **برای حل این مشکل ارائه‌شده استفاده از توابع تقریب زن[[20]](#footnote-20) به جای نگهداری کل فضای حالت در حافظه عامل است.**

### معضل ابعاد بالا

**در مسایل دنیای واقعی با فضای حالت بسیار بزرگ وعدم قطعیت موجود در آن روش‌های سنتی یادگیری تقویتی معمولا نمایش ضعیفی از کارایی دارند. علت این مسئله همانطور که در بخش قبل عنوان شد پیچیدگی فضای حالت یا ببه عبارت دیگر معضل ابعاد بالا (نفرین ابعاد****[[21]](#footnote-21)) است.**

**در دهه‌ی اخیر شیوه‌های مبتنی بر تجرید[[22]](#footnote-22) برای غلبه بر این موضوع مورد توجه قرار گرفته است. این شیوه‌ها به صورت طبیعی منجر به معماری کنترلی سلسله مراتبی و الگوریتم‌های یادگیری مرتبط با این ساختار می‌شوند. در یادگیری تقویتی سلسله مراتبی[[23]](#footnote-23) وظیفه اصلی به یک سری از زیر وظایف که عموماً برای حل ساده‌تر هستند فرموله بندی می‌شود. راه‌حل‌های سلسله مراتبی اغلب جواب‌های نزدیک به بهینه‌ای ارائه می‌دهند و نسبت به تکنیک‌های یادگیری تقویتی هزینه مناسب تری را در زمان اجرا، زمان یادگیری و فضای مورد نیاز برای حل مسایل محض دارند.** (Driessens & Dˇzeroski, 2002a)

**یک راه دیگر برای بهبود کارایی این الگوریتم تخمین زدن خط مشی[[24]](#footnote-24) (تابع ارزش[[25]](#footnote-25)) عامل در یادگیری تقویتی است.** در یادگیری تقویتی عمده تمرکز بر روی طراحی یک تابع تخمین زننده با قابلیت بالا در تخمین زدن و کارایی محاسباتی خوب است.(Fazel Zarandi, 2013)**.**

**استفاده از شبکه‌های عصبی مصنوعی[[26]](#footnote-26) راه حل دیگری برای نگهداری ارزش هر عمل به جای استفاده از جدول می‌باشد.**

**فضای حالت بزرگ دو چالش برای مسئله یادگیری تقویتی دارد یکی از آن‌ها فضای ذخیره‌سازی است به طوری که به ازای هر حالت، عمل[[27]](#footnote-27) بهینه باید ذخیره شود و دیگری عمومیت بخشیدن[[28]](#footnote-28) است؛ یعنی تمام این حالت‌ها طی فرایند یادگیری باید چندین بار تست شوند تا مقدار بهینه برای آن حالت به دست آید. به طوری که** تجربه کم ممکن است داده مورد نیاز را برای هر حالت فراهم نکند**[[29]](#footnote-29). این مسئله باعث می‌شود الگوریتم یادگیری تقویتی کارایی لازم را نداشته باشد (فضای حالت بزرگ و یادگیری کند). چالش دیگر سرعت یادگیری پایین این الگوریتم است چرا که باید ابتدا به طور کامل فضای حالت مساله را جستجو کند و راهکار بهینه را از آن استخراج کند.**

**با توجه به مطالب بیان‌شده حل مسئله معضل ابعاد**21 **بالا و همچنین سرعت بخشیدن به فرایند یادگیری این الگوریتم در مسئله کنترلِ موجودیِ چند عامله در یادگیری تقویتی می‌تواند هردو مشکل را حل کند. حال سؤال این جاست که چگونه می‌توان با بهبود الگوریتم یادگیری تقویتی در سیستم‌های چند عامله، فرایند یادگیری را در چنین سیستم‌هایی بهبود بخشید.**

## اهمیت موضوع و مرور ادبیات

### اهمیت موضوع

**نقاط ضعف کنترل موجودی سنتی که روزبه‌روز بیشتر مشکل‌آفرین می‌شدند از یک سو و نیاز به توسعه مکانیزم‌های هماهنگ‌کننده در زنجیره تأمین از سوی دیگر زمینه را برای به وجود آمدن و رشد رویکردهای جدیدی در زمینه‌ی کنترل موجودی در زنجیره‌ی تأمین فراهم نمودند. در این رویکردهای جدید، اطلاعات تقاضای مشتری نهایی در کل زنجیره به اشتراک گذاشته می‌شود و تصمیمات مربوط به کنترل موجودی حلقه‌های مختلف زنجیره تأمین به صورت یکپارچه و توسط یک سیستم مرکزی اتخاذ می‌شوند.**

**در سال‌های اخیر با توجه به رقابت موجود در بازار و افزایش تنوع کالاها، شرکت‌ها به این نتیجه رسیده‌اند که برای پیروزی در این رقابت نیاز دارند تا با شرکت‌هایی که با آن‌ها همکاری می‌کنند اعم از تأمین‌کنندگان مواد و یا خرده‌فروشان و مشتریان نهایی تعامل برقرار کنند. از این رو زنجیره تأمین در مدل‌های موجودی بروز بیشتری پیدا کرد. زنجیره تأمین در کنترل موجودی، ابتدا مدل‌های ساده دو سطحی که شامل یک فروشنده و چند خریدار است مورد مطالعه قرار گرفت. سپس در سال‌های اخیر مدل‌های زنجیره تأمین چند سطحی مورد بررسی قرار گرفت که معمولا شامل تأمین‌کنندگان، تولیدکنندگان، مرکز توزیع و خرده‌فروشان است. در مدل‌های زنجیره تأمین دو سطحی نیز حالت‌های مختلفی مطالعه شده است که عبارت‌اند از:**

1. مدل‌های شامل یک فروشنده و یک خریدار
2. مدل‌های شامل یک فروشنده و چند خریدار
3. مدل‌های شامل چند فروشنده و یک خریدار
4. مدل‌های شامل چند فروشنده و چند خریدار

در تاریخچه کنترل موجودی، پیش از این که موضوع کنترل موجودی در زنجیره تأمین بررسی شود، مقالات بسیاری روی رفتار تقاضا و تاثیر آن در برآورد هزینه‌ها کارکرده‌اند. آشکار است که تقاضا در دنیای واقعی مقداری ثابت و قطعی نبوده و معمولا متغیر و احتمالی است. مطالعات تجربی (M.P. Baganha, 1998)این نکته را بیان می‌کند که سیاست کنترل موجودی با افزایش در خواست‌های نامنظم مشتری ناپایدار می‌شود. اثر شلاقی (J.W. Forrester, 1961) یکی از موضوعات مهم در کنترل موجودی است (Lee & Wu, 2006)،به این مفهوم که یک تغییر کوچک در درخواست پایین‌دستی منجر به یک سفارش بسیار زیاد به بالادستی می‌شود. اعوجاج در خواست‌های مشتری مهم‌ترین عامل در به وجود آمدن اثر شلاقی است. عامل‌های دیگری همچون تأخیر در ارسال اطلاعات بین عامل‌ها و همچنین عدم پشتیبانی مناسب طرف اعضای زنجیره از دیگر عوامل به وجود آمدن اثر شلاقی در زنجیره تأمین هستند (Sheu, 2005).

در دهه‌های گذشته تحقیقات بسیاری برای بهبود سیاست سفارش دهی در کنترل موجودی انجام‌گرفته که نشان از اهمیت این بحث در زنجیره تأمین دارد (Strozzi, Bosch, & Zaldívar, 2007) (S.O. Kimbrough, 2002) (Mahadevan, Marchalleck, Das, & Gosavi, 1997) (Chaharsooghi, Heydari, & Zegordi, 2008). سیاست سفارش دهی نقش مهمی در کاهش هزینه انبار (هزینه کمبود و مازاد بودن موجودی) و افزایش سطح سرویس‌دهی به مشتری دارد. در مواقعی که تقاضا از طرف مشتری مقدار ثابتی است سیاست بهینه در سفارش دهی **یک برای یک** یا **پاس دادن سفارش****[[30]](#footnote-30)** است. بر اساس این راهبرد هر عامل مقداری را سفارش می‌دهد که توسط پایین‌دستی به عنوان تقاضا رسیده است (S.O. Kimbrough, 2002).این خط مشی برای محیط‌های قطعی و معین مناسب است. برای تصمیم‌گیری در باره مقدار سفارش مناسب در محیط‌های غیرقطعی و احتمالی روش‌هایی وجود دارد ک در تحقیق‌های (S.O. Kimbrough, 2002)و (Giannoccaro & Pontrandolfo, 2002) معرفی‌شده‌اند. در سیستم زنجیره تأمین چند رده‌ای**[[31]](#footnote-31)** هدف بلندمدت همه عامل‌ها افزایش میانگین پاداش و همچنین نگه‌داشتن وضعیت انبار در نزدیک سطح صفر است، به طوری که عامل با رسیدن تقاضای جدید از پایین‌دستی دچار کمبود و همچنین موجودی اضافه در انبار نشود. سفارش بهینه در چنین سیستمی به علت عدم قطعیت نهفته در پارامترهای آن )مقدار تقاضای محصول توسط مشتری، زمان رسیدن مشتری،مدت زمان سفارش محصول تا رسیدن آن و اثر شلاقی) بسیار سخت و پیچیده است.. (Chaharsooghi, Heydari, & Zegordi, 2008)

### مرور ادبیات

مدیریت سفارش دهی در زنجیره تأمین (SCOM) **[[32]](#footnote-32)** که بحث اصلی این پایان نامه است یک روش تجمیعی یا متمرکز**[[33]](#footnote-33)** برای مشخص کردن مقدار سفارش کالای هر عامل به عامل بالادست خود، در زنجیره تأمین است. هدف نهایی مدیریت سفارش دهی کاهش هزینه انبار در کل زنجیره تأمین است.

تمرکز 32SCOM روی در خواستهای زنجیره برای کاهش هزینه انبار ، کاهش پس افت، بالا بردن سطح سرویس‌دهی به مشتری و افزایش سودمندی در کل زنجیره تأمین است (Chaharsooghi, Heydari, & Zegordi, 2008). **مشاهدات نشان داده که هنگامی که انسان‌ها مدیریت سفارش دهی را در زنجیره تأمین به دست می‌گیرند (هم در مسایل واقعی و هم در آزمایشگاه‌های شبیه‌سازی) نتایج حاصله از نقطه بهینه سیستم دورتر است** (Lee & Whang, 1999) (Petrovic, Xie, Burnham, Burnham, & Petrovic, 2008) این نتایج ممکن است به دلایل نبود محرک‌هایی برای اشتراک‌گذاری اطلاعات، عقلانیت نسبی و گاهی اوقات به خاطر تصمیم‌گیری‌های فردی**[[34]](#footnote-34)** است.

در تاریخچه بهبود سیاست سفارش دهی، بیشتر تمرکز محققان بر هماهنگی و ترکیب سیاست‌های تصمیم‌گیری بر روی سیستم‌های بیش تر از سه عامل بوده (S.O. Kimbrough, 2002) (Mahadevan, Marchalleck, Das, & Gosavi, 1997). هنگامی که هیچ هماهنگی بین اعضای زنجیره تأمین **نباشد** هر عامل بر اساس معیارهای خودش سفارش می‌دهد، که نتیجه آن یک بهینگی محلی در مقابل بهینگی کل زنجیره تأمین است. (کنترل غیر متمرکز)

**بیشتر کارهایی که جهت بهبود سفارش دهی در زنجیره تأمین انجام‌شده از الگوریتم ژنتیک برای بهینه‌سازی استفاده شده است . استفاده از الگوریتم ژنتیک دارای درجه آزادی بیشتری نسبت به سفارش دهی** یک بر یک30 **است. در این نوع بهینه‌سازی هر عامل برای کمینه کردن هزینه انبار در کل زنجیره تأمین با همکاری دیگر عامل‌ها قانون اختصاصی سیاست بهینه مخصوص به خود را یاد می‌گیرد .**

**یکی از محدودیت‌های الگوریتم‌های مبتنی بر ژنتیک قانون سفارش دهی ثابت****[[35]](#footnote-35) برای هر عامل است، به عنوان مثال در تحقیق** (S.O. Kimbrough, 2002) **با عنوان "آیا عامل هوشمند می‌تواند زنجیره تأمین را مدیریت کند؟" که یکی از اولین الگوریتم‌های پیشنهادی برای بهینه‌سازی میزان هزینه زنجیره تأمین است. پس از تکرارهای زیاد برای بهینه‌سازی با استفاده از ژنتیک ، قانون‌های ثابتی را برای سفارش دهی به عنوان قوانین بهینه پیشنهاد می‌کند، به این صورت که برای سه عامل قانونی به صورت مشخص می‌کند و اندازه درخواست پایین‌دستی است. درخواستی که به بالادستی سفارش داده می‌شود برابر است .** مقدار 𝒚 توسط الگوریتم ژنتیک مشخص می‌شود و **به عنوان یک مقدار ثابت همواره به درخواست پایین‌دستی اضافه می‌شود** و به عنوان سفارش به بالادستی ارسال می‌شود. تحقیق فوق نشان داده که استفاده از چنین راهبردی نتایج بهتری نسبت به راهبرد پاس دادن سفارش یا1-1دارد.

**قانون سفارش دهی ثابت**35 **برای هر عامل در الگوریتم‌های مبتنی بر ژنتیک از نقاط ضعف آن محسوب می‌شود به خصوص در محیط‌های پویا که توزیع آماری درخواست‌های مشتری به درستی مشخص نیست. یکی از اولین تلاش‌هایی که برای ایجاد قانون سفارش دهی به صورت پویا انجام‌شده مربوط به** (Giannoccaro & Pontrandolfo, 2002) **، که در آن از تکنیک میانگین پاداش نیمه مارکوف[[36]](#footnote-36) برای مسئله کنترل موجودی در زنجیره تأمین استفاده شده است.** (Chaharsooghi, Heydari, & Zegordi, 2008) **نیز از یادگیری تقویتی برای بهینه‌سازی مدل بازی نوشابه[[37]](#footnote-37) استفاده کرده است. مدل‌سازی** مسئله به این صورت عامل‌ها را قادر می‌سازد که قوانین مختلفی در حالت‌های مختلف انبار اتخاذ کنند.**[[38]](#footnote-38)**

**مدل‌سازی بر مبنای یادگیری تقویتی عامل‌ها را قادر می‌سازد از قوانین مختلفی در سفارش دهی کالا استفاده کنند که به نوبه‌ی خود این روش انعطاف‌پذیری بیشتری نسبت به بهینه‌سازی‌های مبتنی بر ژنتیک دارد. در تحقیق** (Fazel Zarandi, 2013)**که از یادگیری تقویتی فازی استفاده‌شده زنجیره تأمین را به صورت ترکیبی از چندین عامل هوشمند در نظر گرفته است که وظیفه این عامل‌ها سفارش دادن است و در آن هر عامل سعی میکند با بررسی کل زنجیره مقدار سفارش بهینه را بیابد . عامل‌ها بر اساس هدف مشترکی با یکدیگر تعامل دارند در تحقیق فوق آن‌ها یک سیستم خطی زنجیره تأمین را در نظر گرفته‌اند که در آن چهار عامل سفارش‌دهنده وجود دارد که هر کدام از آن‌ها مسئول سفارش دهی در یک قسمت خاصی از زنجیره است هدف اصلی عامل‌ها کاهش هزینه نهایی سیستم است که در بلندمدت محاسبه می‌شود. همچنین در این تحقیق عدم قطعیت نیز وجود دارد.**

**ادبیات مدیریت زنجیره تأمین شاخه‌های مختلفی را شامل پیش‌بینی، تهیه و تدارک، تولید، توزیع، کنترل موجودی، حمل‌ونقل و سرویس‌دهی به مشتری پوشش می‌دهد، مدیریت سفارش دهی**3**(کنترل موجودی) که بحث اصلی این تحقیق است، رویکرد یک پارچه برنامه‌ریزی و کنترل انبار در شبکه‌ای از سازمان‌های همکار از مبدأ تهیه مواد اولیه تا رسیدن محصول به دست مشتری نهایی است.**

## ضرورت و کاربرد تحقیق

**در حوزه آکادمیک خلأ موجود در طراحی یک عامل یادگیر کنترل‌کننده انبار در سیستم چند عامله مبتنی بر الگوریتم یادگیری تقویتی با در نظر گرفتن محدودیت‌های الگوریتم یادگیری تقویتی و همچنین ارائه راهکار برای بهبود همزمان این محدودیت‌ها حس می‌شود. مقالات متعددی که در بخش** Error! Reference source not found.**معرفی شد هر کدام به نحوی سعی در بهبود کارایی مسئله کنترل موجودی زنجیره تأمین داشته‌اند ولی هیچ یک تاکنون به طراحی چنین عاملی نپرداخته‌اند. لذا در این پایان‌نامه قصد داریم امکان طراحی چنین سیستمی را بررسی کنیم. در بخش کاربرد عملیاتی نیز هوشمند سازی سیستم‌های سنتی یکی از راهکارها برای بالا بردن کارایی و سودآوری زنجیره تأمین است. با استفاده از این رویکرد می‌توانیم یک مکانیزم تصمیم‌گیری هوشمند بدون دخالت عامل انسانی در یک شبکه زنجیره تأمین داشته باشیم. استفاده از مفاهیم سیستم‌های توزیع‌شده و چند عامله و همچنین استفاده از مفاهیم تصادفی و احتمالی در این مسئله امکان پیاده‌سازی واقعی مسئله کنترل موجودی در زنجیره تأمین را به ما می‌دهد.**

**استفاده از سیستم‌های چند عامله در حل مسایل زنجیره تأمین به علت ماهیت توزیع‌شده بودن آن مورد توجه بسیاری از پژوهشگران قرارگرفته ولی آنچه کمتر به آن توجه شده استفاده از عامل‌های خودمختار و هوشمند در این زمینه است و بیشتر به جای استفاده از سیستم‌های چند عامله از سیستم‌های مبتنی بر عامل[[39]](#footnote-39) استفاده شده است** Jiao, 2006) ;D. Mele, 2007 (Liang,2006; Hanafizadeh, 2009; Jiang, 2009 **اگر چه شباهت‌هایی در استفاده از یادگیری تقویتی در این پایان‌نامه و تحقیق‌های ذکرشده در بالا و همچنین بخش** Error! Reference source not found. **وجود دارد ولی تفاوت‌هایی از قبیل حل مسئله کنترل موجودی ، یادگیری حین اجرا ، ارائه راهکار جدید برای بهبود کارایی این الگوریتم در محیط چند عامله و استفاده از عامل‌های هوشمند مستقل وجود دارد. در جدول زیر به طور موردی این تفاوت‌ها بیان شده است.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | کنترل موجودی | یادگیری حین اجرا | تسریع یادگیری | تخمین زدن جدول ارزش | چند عامله |
| (Fazel Zarandi, 2013) | **■** | **■** | **×** | **×** | **×** |
| (Lin, 2011) | **×** | **■** | **×** | **×** | **×** |
| (Hanafizadeh, 2009) | **×** | **■** | **×** | **×** | **■** |
| (Jiang, 2009) | **■** | **■** | **×** | **×** | **multi-echelon** |
| (Chaharsooghi, Heydari, & Zegordi, 2008) | **■** | **■** | **×** | **×** | **×** |
| (Vengerov, 2007) | **×** | **■** | **×** | **×** | **■** |
| (D. Mele, 2007) | **×** | **×** | **×** | **×** | **■** |
| (Jiao, 2006) | **×** | **×** | **×** | **×** | **■** |
| (Liang, 2006) | **×** | **×** | **×** | **×** | **multi-echelon** |
| (Giannoccaro & Pontrandolfo, 2002) | **■** | **■** | **×** | **×** | **×** |
| هدف از این پایان‌نامه | **■** | **■** | **■** | **■** | **■** |

## فرضیه‌ها

**در این مسئله ما فرض می‌کنیم یک عامل غیرانسانی کنترل‌کننده موجودی در محیط** w **[[40]](#footnote-40)وجود دارد؛ که ادراکات و تصمیمات آن به چند حالت محدود ختم می‌شود. عامل می‌تواند درخواست فعلی و همچنین وضعیت انبار را مشاهده کند، بر اساس این مشاهده عامل می‌تواند چندین عمل محدود[[41]](#footnote-41) را انجام دهد که باعث می‌شود حالت محیط تغییر کند، همچنین ما فرض می‌کنیم که محیط (حالت انبار ، درخواست مشتری) به بازه‌های زمانی گسسته تقسیم می‌شود[[42]](#footnote-42) که در این حالت عامل با مشاهده محیط کار**t **[[43]](#footnote-43) را انجام می‌دهد.**

**هدف ما این است که در نهایت یک نگاشتی از مشاهده[[44]](#footnote-44) به عمل را ایجاد کنیم که کار** T **را به بهترین نحو انجام دهد (تا جایی که امکان دارد مقرون به صرفه)، چنین نگاشتی به وسیله عبارت زیر بیان می‌شود.**

**اگر کار** T **هیچ ابهامی نداشته باشد، عامل با گذشت زمان و با استفاده از تجربه کسب‌شده می‌تواند سیاست بهینه را برای انجام این کار پیدا کند، بنابراین در بازه زمانی t عامل وضعیت محیط را بررسی می‌کند و عمل مناسب را برای انجام دادن کار** T بر **اساس سیاستی که یاد گرفته انجام می‌دهد.انجام عمل** at**باعث می‌شود که حالت محیط از stبه** st+1**تغییر کند.که منجر به مشاهده جدید از محیط می‌شود (**Ot+1**).**در این پایان‌نامه ما فرض می‌کنیم که می‌خواهیم سیاست کنترل انبار بهینه را به صورت بر خط[[45]](#footnote-45) به دست آوریم.

**با توجه به اینکه این مسئله در زنجیره تأمین بررسی می‌شود ما سه نمونه از این عامل‌ها را در سه سطح به صورت خطی شامل یک خرده‌فروش، یک عمده‌فروش و یک کارخانه در نظر می‌گیریم شکل 1**

جریان کالا

شکل 1 عامل های موجود و جریان داده ، کالا

**سفارش مشتری در بازه زمانی ثابت و با مقادیر متفاوت[[46]](#footnote-46)**بین 0تا 15 واحد **به دست خرده‌فروش می‌رسد. سفارش خرده‌فروش، عمده‌فروش و کارخانه در بازه زمانی ثابت و با مقادیر متفاوت بین 0 تا 18[[47]](#footnote-47) واحد به دست بالاسری می‌رسد.زمان تدارک[[48]](#footnote-48) ثابت و قطعی است. از لحاظ فضای انبارها محدودیتی وجود ندارد. پس افت[[49]](#footnote-49)در این مسئله امکان‌پذیر است و هزینه ثابتی برای هر یک از اجزا دارد. هزینه انبارش کالا در انبار مقادیر ثابتی است و برای همه سطوح یکسان است. زمان رسیدن در خواست (جریان اطلاعات) به بالاسری صفر واحد زمانی است. ظرفیت ناوگان حمل و نقلی برای توزیع محصول محدود است (یک کامیون با حداکثر ظرفیت 18 واحد کالا). یک نوع محصول در زنجیره تولید و توزیع می‌شود. هر عامل وظیفه بهبود سطح موجودی خود را با توجه به وضعیت انبار خود دیگر دارد.**

**فصل دوم**

**مرور ادبیات تحقیق**

# 

# مرور ادبیات تحقیق

## مقدمه

**در دهه 1980 شرکت‌ها به دنبال تکنیک‌ها و راهبردهایی بودند که با استفاده از آن‌ها بتوانند هزینه‌های تولیدی خود را کاهش داده و در بازارهای مختلف رقابت کنند. بعضی از این تکنیک‌ها عبارت بودند از: سیستم زمان‌بندی به هنگام (JIT[[50]](#footnote-50))، سیستم کانبان، تولید ناب، مدیریت کیفیت جامع و غیره. شرکت‌ها توانستند با استفاده از این تکنیک‌ها هزینه‌های تولید خود را در حد امکان کاهش دهند. ولی شرکت‌های رقیب نیز با استفاده از همین تکنیک‌ها، هزینه‌های تولید خود را در حد امکان کاهش داده‌اند. در بازار رقابتی امروز، تولیدکنندگان به این نتیجه رسیده‌اند که برای رسیدن به یک حالت بهینه و کم کردن هزینه‌ها و از طرفی بالا بردن رضایت مشتری نیاز به یک تعامل خوب با تأمین‌کنندگان دارند. در واقع داشتن دید یک طرفه برای بهینه کردن تابع هزینه جوابگوی دنیای رقابتی امروز نیست، پس برای کاهش هزینه‌ها و ماندن در بازار رقابتی باید سایر فرصت‌های بالقوه برای کاهش هزینه را پیدا کرد. یکی از این زمینه‌ها که فرصت‌های بالقوه زیادی برای کاهش هزینه در آن وجود دارد، زنجیره تأمین است.در این فصل مروری بر ادبیات زنجیره تأمین و کنترل موجودی در آن خواهیم داشت.** و در ادامه به معرفی یادگیری تقویتی به عنوان الگوریتم بدون ناظر[[51]](#footnote-51) (که نشان داده می‌شود برای حل مسئله کنترل موجودی توسط عامل‌های هوشمند مناسب است) می‌پردازیم. تکنیک‌های یادگیری تقویتی به طور مستقیم از تجربه‌ای که نسبت به محیط به دست می‌آورند، می‌آموزند. ما در این فصل دربخش ‏2.3 ابتدا چارچوب یادگیری تقویتی را معرفی می‌کنیم و سپس به معرفی یادگیری Q به عنوان یک الگوریتم و تکنیک پرطرفدار برای حل مسایل یادگیری تقویتی می‌پردازیم و در انتها یک مسئله ساده را برای درک چگونگی حل مساله توسط این الگوریتم و نشان دادن کارایی و روند یادگیری این الگوریتم حل می‌کنیم.

## زنجیره تأمین

**زنجیره تأمین را می‌توان یک شبکه جهانی از تأمین‌کنندگان، کارخانجات، انبارها، مراکز توزیع و خرده‌فروشان دانست که موارد اولیه را گردآوری و تبدیل نموده و محصول نهایی را به دست مشتریان می‌رسانند. یک زنجیره تأمین شامل همه مراحل (اعضای زنجیره ) است که چه مستقیم و چه غیرمستقیم، در برآورده سازی درخواست یک مشتری نقش دارند. در یک زنجیره تأمین معمولی، مواد خام از انبارهای میانی و انبارهای توزیع‌کننده‌ها ارسال می‌شوند و از آنجا نیز به سمت خرده‌فروش‌ها و در نهایت به دست مشتری نهایی یا همان مصرف‌کننده می‌رسند. در بعضی از این مراحل، کالا انبارش می‌شود و در بعضی دیگر حمل می‌شود.** یعنی یک زنجیره تأمین مجموعه‌ای از انبارش ها و حمل‌ونقل ‌هاست**. اعضای زنجیره تأمین معمولی عبارتنداز: تأمین‌کنندگان و انبارهای مواد. فعالیت‌های زنجیره تأمین با سفارش مشتری شروع می‌شود و وقتی که مشتری پول خرید کالا و خدمات دریافتی خود را پرداخت می‌کند، خاتمه می‌یابد.** اختلاف بین پولی که مشتری می‌پردازد با کل هزینه‌های متحمل شده توسط زنجیره برای تولید و توزیع کالا، میزان سوددهی زنجیره را نشان می‌دهد**. بر همین اساس موفقیت یک زنجیره بر حسب میزان سوددهی آن تعریف می‌شود.**

**بعضی از شرکت‌ها کوشش می‌کنند تا کنترل زنجیره تأمین خود را با کنترل عمومی عمودی و با استفاده از مالکیت یکپارچگی تمام اجزای مختلف در امتداد زنجیره تأمین از تهیه مواد و خدمات تا تحویل محصول نهایی و خدمت به مشتری، به دست آورند.[[52]](#footnote-52) اما حتی این نوع ساختار سازمانی، فعالیت‌های مختلف و واحدهای عملیاتی ممکن است ناهماهنگ باشند.**

**شرایط نوین حاکم بر بازارهای امروزی از قبیل افزایش انتظارات مشتریان در زمینه‌های قیمت، کیفیت، تنوع محصولات، تحویل به موقع و ... با در نظر گرفتن** جریان آزاد اطلاعات که به معنی افزایش سطح آگاهی مشتریان نسبت به محصولات رقیب می‌باشد**؛ همچنین پیشرفت‌های قابل‌ملاحظه‌ای که در فنآوریهای تولیدی و سیستم‌های حمل‌ونقل و روش‌های سفارش گذاری حاصل گردیده و درعین‌حال کاهش هزینه‌های مستقیم نیروی کار، بنگاه‌های اقتصادی را وادار نموده تا جهت پوشش نیازهای جدید، فراهم آوردن رضایت مشتریان و باقی ماندن در صحنه رقابت از یک سو با زنجیره تأمین و از سوی دیگر با مشتریان پیوند یابند. این امر موجب شده است که مدیران در صدد استفاده از روشی جهت اداره زنجیره تأمین برآیند تا به شکلی کاملاً موثر در صحنه رقابت حاضر شوند. در این میان استفاده از** مدیریت زنجیره عرضه **به عنوان روشی جهت پوشش نیازهای تأمین مد نظر قرار گرفت و با توسعه علوم ارتباطات الکترونیکی توسعه دو چندانی یافت.**

**در همین راستا مفهوم مدیریت زنجیره تأمین برای بالا بردن سود زنجیره مطرح می‌شود که** مستلزم مدیریت جریان‌های بین مراحل و درون هر یک از مراحل زنجیره برای بیشینه کردن کل سود دهی آن است. **در بخش بعد مدیریت زنجیره تأمین را معرفی می‌کنیم.**

### مدیریت زنجیره تأمین

**افزایش سطح توقع مشتریان و افزایش رقابت میان تولیدکنندگان باعث ظهور و رشد ایده‌هایی نظیر مدیریت زنجیره تأمین شده است. مدیریت زنجیره تأمین به معنای برنامه‌ریزی، سازمان‌دهی و کنترل فعالیت‌ها در زنجیره تأمین است و این عملکرد در حال حاضر منبع اصلی ایجاد مزایای رقابتی و مالی در شرکت‌های تولیدی و خدماتی است، به همین جهت مدیران ارشد شرکت‌های تولیدی از روش‌ها و ابزارهای مختلفی جهت تحقق اهداف و طرح‌های تجاری خود، مبتنی بر کسب مزیت رقابتی و دستیابی به سهم بیشتری از بازار استفاده می‌نمایند. در این راستا استراتژی‌هایی که منجر به کاهش هزینه‌ها، افزایش سرعت تحویل محصول(کاهش Lead Time تحویل)، ارائه محصول سفارشی به مشتریان، ارائه محصولات باکیفیت بالاتر، قیمت پایین تر و دسترس‌پذیری بالاتری می‌گردند از اولویت بالاتری برخوردار خواهند بود.**

**برای مدیریت موثر زنجیره تأمین ضروری است که تأمین‌کنندگان و مشتریان با یکدیگر در یک روش هماهنگ و با مشارکت و ارتباطات اطلاعاتی و گفتگو با یکدیگر کارکنند. این امر یعنی جریان سریع اطلاعات در میان مشتریان و عرضه‌کنندگان،** مراکز توزیع و سیستم‌های حمل‌ونقل**، شرکت‌ها را قادر می‌سازند که زنجیره‌های عرضه بسیار کارایی را ایجاد نمایند و بر دستیابی به اهداف کلی شرکت تمرکز کند.**

#### تعاریف و مفاهیم اولیه در مدیریت زنجیره تأمین

**مدیریت زنجیره تأمین در برگیرنده مدیریت مواد و اطلاعات در طول زنجیره تأمین از تأمین‌کنندگان تا تولیدکنندگان قطعات و مونتاژ کنندگان نهایی و در مرحله بعد توزیع‌کنندگان و نهایتاً مصرف‌کنندگان می‌باشد. تعریف‌های زیادی درباره مدیریت زنجیره تامین وجود دارد که به برخی از آن‌ها اشاره می‌کنیم.**

* (Handfield, Ernest, & Jr, 2004)، زنجیره تأمین و مدیریت آن را به صورت زیر تعریف می‌کند: زنجیره تأمین، همه فعالیت‌های مرتبط با جریان و تبدیل کالاها از مرحله ماده خام (استخراج) به حالت نهایی (برای مصرف) و نیز جریان‌های اطلاعاتی مرتبط با آن‌ها را شامل می‌شود. یکپارچه‌سازی فعالیت‌های مذکور به منظور دستیابی به یک موقعیت مطلوب را مدیریت زنجیره تأمین می‌نامند.
* مدیریت زنجیره تأمین یک فرایند تصمیم‌گیری در سطوح عملیاتی، تاکتیکی و راهبردی است که راندمان زنجیره تأمین را بهینه‌سازی می‌کند (گلپایگانی, 1389).
* تعریف: مدیریت زنجیره تأمین مجموعه‌ای از راهکارها جهت یکپارچه‌سازی اعضای زنجیره (تأمین‌کنندگان، تولیدکنندگان، توزیع‌کنندگان، خرده‌فروشی‌ها و مشتری نهایی) که هدف آن کاهش هزینه‌های سیستم و نیز افزایش سطح خدمت دهی به مشتریان است.
* بر اساس تعریف (Harland, 1996)مدیریت زنجیره تأمین عبارت است از:
* مدیریت فعالیت‌های کسب‌وکار
* مدیریت روابط داخلی سازمان
* مدیریت روابط با تأمین‌کنندگان رده اول و دوم در طول زنجیره و نهایتاً با کل زنجیره
* (New & Payne, 1995)زنجیره تأمین را به عنوان ارتباط‌دهنده همه فرایندهای ساخت و تأمین (از مواد خام تا مشتری) و متشکل از چندین سازمان معرفی می‌نمایند. (Baatz, 1995) بازیافت مواد را نیز بخشی از حوزه فعالیت‌های مدیریت زنجیره تأمین می‌داند.

#### تاریخچه مدیریت زنجیره تأمین

**ایده‌ی همکاری و شراکت در سیستم‌های تولیدی بعد از اولین انقلاب صنعتی وجود داشت. نگرش آن زمان گرچه به عنوان رویکرد زنجیره‌ی تأمین کامل نبود. ولی با توجه به شرایط اقتصادی حاکم کامل به نظر می‌رسید. در آن زمان هر بخش سازمان یک جزیره جداگانه بود و میان سازمان و سازمان‌هایی که طرف تجاری آن بودند مانند تأمین‌کنندگان و عمده‌فروشان، روابط غیر دوستانه‌ای حاکم بود. اما پس از دومین انقلاب صنعتی با سیستم تولید تویوتا در ژاپن، شرایط تغییر کرد. تغییرات دیگری نیز با ظهور سیستم‌هایMRP [[53]](#footnote-53)و MRP-II[[54]](#footnote-54) و در نهایت سیستم ERP[[55]](#footnote-55) به وقوع پیوست. سازمان‌ها به شکل یک موجودیت درآمده و زنجیره‌های تأمین داخلی شروع به رشد نمودند. روابط میان سازمان‌های تجاری بهبود یافت و با نام شراکت گسترش پیدا کرد. این تغییرات در نهایت منجر به پدید آمدن ایده مدیریت زنجیره تأمین شد. در واقع ایده زنجیره تأمین از زمان‌های بسیار قدیم وجود داشته ولی ایده‌ی مدیریت زنجیره تأمین ایده‌ی جدیدی است.**

**مدیریت زنجیره تأمین با رشد و ترکیب عملکردهای مدیریت ساخت و مدیریت تدارکات به وجود آمد. ایده‌ی مدیریت تدارکات نیز خود از به هم پیوستن مدیریت مواد و مدیریت توزیع و فروش به وجود آمد.**

**ادبیات مدیریت زنجیره تأمین شاخه‌های مختلفی از جمله پیش‌بینی، تهیه و تدارک، تولید، توزیع، کنترل موجودی، حمل‌ونقل و سرویس‌دهی به مشتری پوشش می‌دهد. در بخش بعد کنترل موجودی در زنجیره تأمین را که بحث اصلی این پایان نامه است معرفی می‌کنیم.**

### کنترل موجودی

**در جامعه‌ی صنعتی تولیدکنندگان کالا، یکی از مشکلات اساسی شرکت‌ها مدیریت موجودی انبار است. اگر موجودی انبار شرکتی بسیار زیاد باشد. شرکت می‌بایست، پیشاپیش هزینه‌ی تولید و استهلاک مازاد تولید را بپردازد. اگر موجودی انبار شرکت به هنگام افزایش تقاضای مصرف‌کننده بسیار کم باشد، شرکت فرصت فروش به مشتریانی را از دست خواهد داد که منتظر نمی مانند و ممکن است به سراغ رقیبی بروند که کالاهای مشابه دم دست دارد. هر چه مدیریت انبار بهتر باشد، سود بیشتری عاید شرکت می‌شود.کنترل موجودی همواره یک ابزار بسیار مهم و کارآمد برای شرکت‌ها است. به همین دلیل یکی از موضوعاتی که محققان همواره روی آن متمرکز داشته‌اند کنترل موجودی است. در گذشته مدیریت موجودی معمولا برای محصولات تمام‌شده در انتهای زنجیره به کار گرفته می‌شد. اما با گذشت زمان شرکت‌ها کم‌کم درک نمودند که** هزینه‌ی مازاد موجودی غیرضروری در هر کجای زنجیره تأمین که ذخیره باشد، خواه شرکت نسبت به آن مسئولیت مستقیم داشته باشد یا نه، باز هم در هزینه‌های پایین خطی شرکت تاثیر گذار است. **بنابراین امروزه موجودی نگهداری شده در هر یک از حلقه‌های زنجیره تأمین با یکدیگر مرتبط در نظر گرفته‌شده و به سطوح نگهداری مواد خام و اجزا با دید عناصر مرتبط به هم و به وجود آورنده‌ی فرصت برای کاهش هزینه‌ها نگریسته می‌شود.**

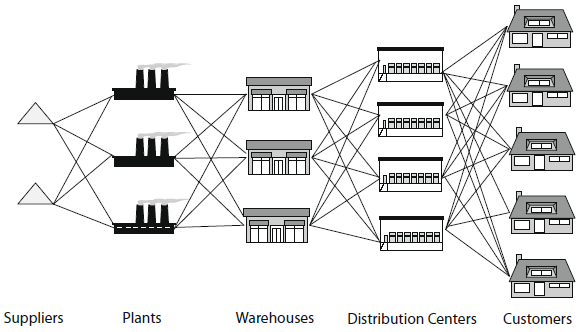
#### تعریف موجودی:

**موجودی، انباشته یا مجموعه‌ای از کالا که به منظور برآورده ساختن نیاز آینده در نظر گرفته می‌شود. از دلایل نگهداری موجودی می‌توان به تغییرات تقاضا در طول زمان، تغییرات قیمت و هزینه‌ها، عدم اطمینان درباره موعد تحویل، بهبود خدمت دهی به مشتریان و رسیدن به سطح مطلوبی از انعطاف‌پذیری در خط تولید اشاره کرد.**

#### تعریف مدیریت انبار(کنترل موجودی):

**مدیریت انبار در زنجیره تأمین عبارت است از برنامه‌ریزی برای کنترل موجودی انبار از مبدأ تولید تا مصرف‌کننده نهایی در طول شبکه‌ای از سازمان‌هایی که باهم همکاری می‌کنند** (Alev Taskin Gumus, 2009)**. کنترل موجودی روی درخواست مشتری نهایی تمرکز می‌کند و هدف آن بالا بردن سطح سرویس‌دهی به مشتری، افزایش تنوع محصول و کاهش هزینه‌هاست** (Giannoccaro I. P., 2003)**، اکثر شرکت‌های بزرگ به صورت شبکه‌ای شامل کارخانه‌های تولید مواد خام اولیه ،کارخانه‌های ایجاد محصول و مراکز توزیع محصول برای رساندن محصول به دست مشتری هستند.**شکل 2 **. واژه "چند رده‌ای" یا "چند سطحی" معمولا هنگامی برای نام‌گذاری چنین شبکه‌های تولید و توزیع(یا زنجیره تأمین) به کار می رود که یک محصول در طول بیش از یک گام تا رسیدن به دست مشتری نهایی حرکت کند** (Ganeshan, 1999)(Rau, 2003)**.** شکل 2 **یک سیستم چند سطحی شامل تعدادی تولیدکننده، کارخانه، انبار، مراکز توزیع و مشتری را نشان می‌دهد** (Andersson, 2001)(Axsater, 2003)**.**

شکل 2 - زنجیره تأمین چند سطحی

****

وظیفه سیستم مدیریت موجودی حصول اطمینان از موجود بودن مواد و قطعات در مکان و زمانی است که به آن‌ها نیاز است. به طوری که هزینه کل سیستم مذکور در حداقل ممکن نگه داشته شود. **به این منظور باید با منابع تولید و عرضه کالا، قیمت‌ها، روش‌های حمل‌ونقل، میزان تخفیف‌های اعطائی، سیستم انبارداری و بازرسی آن آشنایی کامل داشت.**

### مدیریت سفارش در زنجیره تأمین[[56]](#footnote-56)

مدیریت سفارش در زنجیره تأمین که نقش مهمی در کنترل موجودی در زنجیره تأمین دارد عبارت است از یک رویکرد ترکیبی[[57]](#footnote-57) برای مشخص کردن مقدار سفارش هر عامل زنجیره تأمین که به عامل بالادست [[58]](#footnote-58) می‌دهد و هدف آن کمینه کردن هزینه انبار در کل زنجیره تأمین است.این رویکرد تمرکزش بر درخواست‌های زنجیره برای رسیدن به اهداف کاهش هزینه انبار کاهش کمبود کالا[[59]](#footnote-59) بالا بردن سطح سرویس به مشتری و افزایش سودمندی کل زنجیره است.

## یادگیری تقویتی

یادگیری تقویتی در یک بیان کلی یعنی با توجه به شناختی که از محیط داریم و بر اساس نتایج تعاملاتی که با محیط داشته‌ایم و سودها و زیان‌هایی که در نتیجه انجام عمل‌های مختلف به دست آورده‌ایم راهبُردی**[[60]](#footnote-60)** را پیدا کنیم که با عمل به آن در بلندمدت سود خود را بیشینه کنیم. این نوع یادگیری یک چهارچوب سرراست برای یادگیری در یک محیط و دستیابی به یک هدف را فراهم می‌کند. این روش یک شبیه‌سازی تصادفی است که اثبات‌شده کارایی خوبی در مسایل با سایز بزرگ دارد.

تصمیم‌گیرنده و کسی که یاد می‌گیرد را عامل[[61]](#footnote-61) و چیزی که عامل با آن تعامل می‌کند را محیط[[62]](#footnote-62) می‌نامیم این تعامل بین عامل و محیط به صورت پی در پی صورت می‌گیرد بدین ترتیب که در هر گام عامل حالت () محیط را دریافت می‌کند و بر مبنای این حالت عمل را (از بین مجموعه اعمالی که می‌تواند انتخاب کند ) انجام می‌دهد و به حالت جدید می رود. محیط نیز در پاسخ به این عمل به او پاداش می‌دهد.در **Error! Reference source not found.** نحوه این تعامل نشان داده شده است.

شکل 3 نحوه تعامل عامل و محیط



پاداش آنی[[63]](#footnote-63) که عامل از محیط دریافت می‌کند بیان‌کننده این است که آخرین تصمیمی که عامل گرفته تا چه اندازه خوب بوده است. هدف نهایی عامل این است که راهبرد بهینه که نگاشتی از مشاهدات[[64]](#footnote-64) به اقدامات[[65]](#footnote-65) است را پیدا کند، اگر فرض کنیم محیط کاملاً قابل مشاهده[[66]](#footnote-66) است، در این حالت حل مسئله یعنی، پیدا کردن نگاشتی از حالت‌ها به تصمیمات.

### سیاست یا راهبرد بهینه

سیاست یا راهبُرد عامل که با π نشان داده می‌شود تابع احتمالی است که احتمال انتخاب‌شدن هر عمل را در هر حالت و با توجه به گام زمانی می‌دهد. به عنوان مثال این مفهوم را بیان می‌کند که اگر عامل درزمان در حالت قرار گرفته باشد با احتمال عمل را انتخاب می‌کند.

هدف عامل در این نوع یادگیری بیشینه کردن پاداش‌هایی است که در بلندمدت به دست می‌آورد. در هر مرحله زمانی، این پاداش به صورت عددی ساده بیان می‌شود. **منظور از بیشینه کردن پاداش نگاه بلندمدت به مجموع پاداش‌ها است و به معنی بیشینه کردن این پاداش‌ها در هر مرحله نیست**. علامت پاداش کانال ارتباطی ما با عامل است و به واسطه آن ما میزان خوب بودن حرکت را به عامل بیان می‌کنیم و به این وسیله به او می‌گوییم که **به چه هدفی برسد نه اینکه چگونه به هدف برسد.** به عنوان مثال در مسئله کنترل سفارش دهی هر گاه عامل مقداری را به **بالاسری سفارش داد ما باید پاداشی که به او می‌دهیم بیانگر این باشد که حرکت تا چه اندازه در راستای هدف بوده**، نه اینکه مسیر رسیدن به هدف را به آن نشان دهیم[[67]](#footnote-67).

روش‌های تقویتی نشان می‌دهند که چگونه یک عامل بر اساس تجربه‌ای که از تعامل با محیط به دست می‌آورد. راهبرد خود را تغییر می‌دهد.

### محاسبه پاداش در بلندمدت

اگر ترتیب پاداش‌هایی که عامل بعد از زمان می‌گیرد به صورت باشد، وی به دنبال بیشینه کردن امید ریاض پاداش کل خواهد بود. پاداش کل به صورت جمع پاداش‌های هر مرحله تعریف می‌شود:

|  |  |  |
| --- | --- | --- |
|  |  |  |

که در آن آخرین مرحله می‌باشد. فرمول فوق برای مسایلی مفید است که فرایند در مرحله مشخصی پایان می‌پذیرد. و به اصطلاح دارای حالت پایانی می‌باشد، اما مسایل فراوانی همانند مسئله کنترل موجودی وجود دارند که تعامل با محیط تا بی‌نهایت ادامه دارد، در این صورت فرمول فوق با قرار دادن واگرا می‌شود. برای حل این مشکل از فاکتور تنزیل[[68]](#footnote-68) به صورت زیر استفاده می‌شود.

|  |  |  |
| --- | --- | --- |
|  |  |  |

فاکتور تنزیل به این صورت عمل می‌کند که وزن بیشتری به پاداش‌هایی که زودتر دیده‌شده‌اند می‌دهد؛ یعنی پاداش‌هایی که t کوچک‌تری دارند.

اگر ما این فاکتور را برابر صفر قرار دهیم الگوریتم به صورت حریصانه عمل می‌کند یعنی فقط پاداشی که الآن گرفته را در نظر می‌گیرد و بهترین انتخاب انتخابی است که بیش‌ترین پاداش را بدهد. مقادیر بزرگ‌تر از صفر این نکته را بیان می‌کنند که ما چه مقدار به پاداش‌هایی که در آینده خواهیم گرفت توجه می‌نماییم.

### خاصیت مارکوف[[69]](#footnote-69)

مسایل یادگیری تقویتی معمولا به عنوان فرایندهای تصمیم‌گیری مارکوف[[70]](#footnote-70) در نظر گرفته می‌شوند. در این نوع فرایندها مجموعه محدودی از حالت‌ها ، مجموعه محدودی از انتخاب‌ها یا اعمال وجود دارد همچنین در این فرایند زمان به صورت گسسته در نظر گرفته می‌شود.

### تابع پاداش:

|  |  |  |
| --- | --- | --- |
|  |  |  |

یک اندازه‌گیری سریع از مقدار خوب بودن حرکت نمایش می‌دهد.حالتی که عامل بعد از انجام حرکت در آن قرار می‌گیرد بسته با تابع انتقال زیر می‌تواند تغییر کند.

|  |  |  |
| --- | --- | --- |
|  |  |  |

این تابع یک تابع توزیع احتمال انتخاب حالت‌های ممکن از حالت فعلی است ،یک خاصیت مهم در فرایندهای مارکوف این است که حالت بعدی در تابع انتقال فقط بسته به آخرین حالت دارد این خاصیت به عنوان خاصیت مارکوف[[71]](#footnote-71) شناخته می‌شود.

### مسئله مارکوف

به مسئله يادگيري تقویتی كه در آن خاصيت ماركوف برقرار باشد، فرآيند تصمیم‌گیری ماركوف گويند. این مسئله به این صورت است که باید یک راه حل برای نگاشت حالت به عمل بر مبنای پاداش آنی به دست آوریم که مجموع پاداش به دست آمده را در طول زمان بیشینه کند. اگر ما توابع TوR را بدانیم آنگاه می‌توانیم تابع ارزش بهینه هر حالت را همانند زیر حساب کنیم.

|  |  |  |
| --- | --- | --- |
|  |  |  |

این تابع به هر حالت یک ارزش نسبت می‌دهد که بر مبنای بهترین پاداشی است که می‌توانیم با انتخاب یک حرکت در آن حالت به دست آوریم به اضافه **بهترین** مقداری که ما می‌توانیم در حالت بعد به دست آوریم (این مقدار با احتمال رفتن به آن حالت وزن دار شده است). اگر ما این تابع را به دست آوریم می‌توانیم از روی آن را به دست آوریم.

این راهبرد به سادگی در هر حالت حرکتی را انتخاب می‌کند که مقدار بیشینه را بدهد، وانتخاب حرکت به حرکات قبلی آن بستگی ندارد :

|  |  |  |
| --- | --- | --- |
|  |  |  |

روش‌های شناخته‌شده خوبی برای محاسبه این مقدار بهینه وجود دارد (همانند polic‌y‌‌‌\_iteration وvalue\_iteration (Sutton, 1998))که روال ساده‌ای را برای یادگرفتن تابع ارزش‌‌‌‌ و در نتیجه راهبرد بهینه معرفی می‌کنند. در قسمت بعد ما یادگیری Q[[72]](#footnote-72) را به عنوان الگوریتمی که تلاش می‌کند به صورت تکراری مقدار بهینه تابع ارزش را پیدا کند معرفی می‌کنیم.

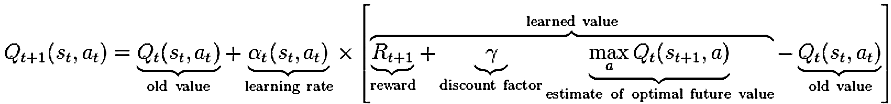
## الگوریتم یادگیری Q

این الگوریتم یک تکنیک **یادگیری تقویتی مستقل از مدل است**، همان طور که گفته شد این روش قادر است یک راهبرد انتخاب عمل در حالت را برای هر فرایند مارکوف ارائه دهد، این روش تابع عمل ارزش را یاد می‌گیرد وبر اساس آن بهترین عمل که همان عملی است که بیش‌ترین ارزش را در هر حالت دارد به عنوان راهبرد بهینه در هر حالت انتخاب می‌شود. یکی از مزایای این الگوریتم مستقل از مدل بودن آن است به این معنی که برای حل مسئله نیاز به مدل کلی مسئله ندارد و فقط به حالت فعلی و حالت بعدی نیاز دارد. همچنین این الگوریتم می‌تواند مسایلی با توابع انتقال احتمالی و همچنین پاداش‌های احتمالی را حل کند (بدون نیاز به تغییر الگوریتم) همچنین اثبات‌شده برای هر فرایند محدود مارکوف الگوریتم Q در نهایت راهبرد بهینه را پیدا خواهد کرد (Sutton, 1998)

مدل‌سازی مسئله شامل یک عامل، حالتی که عامل در آن قرار دارد و مجموعه‌ای از عمل‌ها به ازای هر حالت است. با انتخاب یک عمل عامل می‌تواند حالت خود را تغییر دهد هر حالت جدید به عامل یک پاداش می‌دهد که این پاداش سیگنالی است که میزان خوب بودن آن حالت را بیان می‌کند، هدف عامل بیشینه کردن مجموع پاداش‌هایی است که در هر مرحله (حالت) در یافت می‌کند، در این حالت ممکن است **عامل حرکتی را انتخاب کند که پاداش آنی زیادی نداشته باشد ولی در نهایت منجر شود که مجموع پاداش‌های دریافتی آن بیشینه شود.**

مقدار ارزش هر عمل در هر حالت با مشخص می‌شود و در هر مرحله با معادله 1 به‌روز آوری می‌شود. (Q-learning)

معادله 1 به‌روز آوری ارزش هر عمل در الگوریتم Q



که در آن پاداشی است که عامل در حالت جدید دریافت می‌کند، به عبارت دیگر با انتخاب عمل در حالت عامل به حالت می رود و پاداش آنی را دریافت می‌کند. ضریب یادگیری نام دارد و به این معنی است که مقدار جدید به دست آمده تا چه اندازه می‌تواند مقدار قبلی را تغییر دهد.

**اگر این مقدار برابر صفر باشد عامل هیچ‌چیزی یاد نمی‌گیرد و مقدار ارزش عمل اولیه خود را حفظ می‌کند و اگر این مقدار برابر یک قرار گیرد مقدار قبلی همیشه به وسیله مقدار جدید جایگزین می‌گردد** در این صورت معادله به صورت زیر در می‌آید:

|  |  |  |
| --- | --- | --- |
|  |  |  |

در این رابطه فاکتور تنزیل68 نام دارد و نقش آن در رابطه این است که مهم بودن پاداش‌هایی که در آینده به دست می‌آیند را مشخص می‌کند، همانطور که در ‏2.3.2 گفته شد، اگر این مقدار نزدیک به صفر باشد عامل فقط زمان حال را در نظر می‌گیرد (نزدیک‌بین) و اگر نزدیک به یک باشد عامل سعی می‌کند یک افق بلندمدت را در نظر بگیرد اگر این مقدار برابر یک یا بیشتر باشد، ممکن است به هیچ‌گاه به مقدار مناسب همگرا نشود و عامل نتواند بهترین حرکت را در آن حالت انتخاب کند.

برای هر حالت ما مجموعه‌ای از عمل‌ها داریم که بهترین را ه برای ذخیره کردن آن استفاده از جدول به صورت زیر است:

جدول 1 نحوه نگهداری ارزش هر عمل در یادگیری Q

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | عمل/حالت |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

بنابراین وقتی داده‌های این جدول توسط

به‌روز آوری شد، راهبرد بهینه به این صورت است که در هر حالت ما عملی که بیش‌ترین ارزش را دارد انتخاب می‌کنیم.

با توجه به معادله 1 بدیهی است که مقدار در هر مرحله نیازمند مقدار اولیه[[73]](#footnote-73) است که در اکثر مراجع یادگیری تقویتی این مقدار برابر صفر در نظر گرفته میشود.

اکنون که نحوه به‌روز آوری در مشخص شد نوبت به معرفی خود الگوریتم رسیده است.

### شرح الگوریتم

این الگوریتم به این صورت است که در ابتدا مقدار اولیه همه را برابر صفر قرار می‌دهد و در یک دوره طولانی سعی در آموزش عامل برای به‌روز آوری مقدار با استفاده از

دارد و دریافت بیش‌ترین پاداش دارد.[[74]](#footnote-74)

در ادامه برای مشخص شدن نحوه عملکرد این الگوریتم یک مثال ساده را بیان می‌کنیم.

#### مثالی ساده برای نمایش نحوه عملکرد و کارایی الگوریتم یادگیری Q

فرض کنید که یک ربات[[75]](#footnote-75) داریم که می‌خواهد با استفاده از یادگیری تقویتی مسیر خارج شدن از اتاق شکل 4 را بیابد، هدف این مسئله این است که اگر عامل در یکی از اتاق‌های این خانه قرار گرفت بتواند استراتژی بهینه برای خروج را یاد بگیرد. اگر بخواهیم این مسئله را به یک مسئله یادگیری تقویتی مدل کنیم ابتدا باید حالت‌های مسئله، عمل‌هایی که عامل می‌تواند در هر حالت انجام دهد و پاداش هر عمل را مشخص کنیم.

همچنین باید حالت نهایی که جواب مسئله است را مشخص کنیم. بنابراین باید برای هر درب (ورود و خروج از آن) پاداشی در نظر بگیریم .

گراف مسئله و پاداش هر تصمیم در

شکل 5 نشان داده شده است. در این شکل یک مسیر اضافه برای اتاق F قرار داده‌شده که نشان‌دهنده پایان کار است ،بنابراین اگر عامل در این حالت قرار گرفت ،در این حالت باقی می‌ماند(اگر به صورت حریصانه عمل کند یا اگر فقط سعی کند از آموخته‌های خود استفاده کند.)با توجه به شکل 5 جدول پاداش‌ها به صورت (8-1)است.

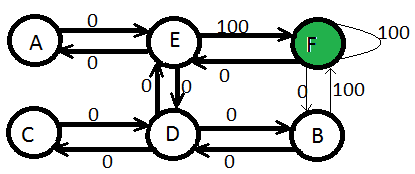
الگوریتم 1 یادگیری تقویتی

|  |
| --- |
| 1. **پارامترها از قبیل را مقداردهی کنید همچنین میزان پاداش هر حالت-عمل را مشخص کنید.** 2. **جدول 1 را تشکیل دهید و مقادیر اولیه آن را برابر صفر قرار دهید** 3. **کارهای زیر را به تعداد باری که می‌خواهید الگوریتم آموزش ببیند انجام دهید**  * **یک حالت اولیه تصادفی انتخاب کنید** * **تا زمانی که به هدف نرسیده‌اید کارهای زیر را انجام دهید** * **یک حرکت را از میان تمام حرکت‌های ممکن از این حالت انتخاب کنید** * **با انتخاب این حرکت شما به حالت جدید میروید** * **بیش‌ترین مقداری که می‌توانید از حالت به دست بیاورید را محاسبه کنید** * **عبارت زیر را به‌روز آوری کنید** * **حالت ر را عنوان حالت فعلی در نظر بگیرید** * **پایان**  1. **پایان** |

شکل 4 خانه‌ای با 5 اتاق که ربات در یکی از اتاق‌های آن قرار دارد و می‌خواهد با استفاده از یادگیری تقویتی مسیر خارج شدن از اتاق را بیابد



شکل 5 حالت‌های مسئله پیدا کردن مسیر خروج و پاداش هر عمل در هر حالت



|  |  |  |
| --- | --- | --- |
|  | R = |  |

در ‏(8-1) سطر اول یعنی از خانه A فقط می‌توانیم با پاداش 0 به خانه E برویم و از خانه E می‌توانیم با پاداش‌های 0 به A،0 به Dو 100 به F برویم. علامت به این معنی است که هیچ راهی از اتاق x به اتاق y وجود ندارد.

حال که جدول پاداش‌ها مشخص شد نوبت به ایجاد جدول ارزش هر عمل و مقداردهی آن اولیه آن با عدد صفر است.این جدول در‏(9-1) نشان داده شده است.

|  |  |  |
| --- | --- | --- |
|  | Q= |  |

اکنون با استفاده از الگوریتم 1، گام به گام مسئله را با در نظر گرفتن پیش می بریم.

فرض کنید عامل در ابتدا در اتاقB قرار دارد، در این اتاق با توجه به ‏(8-1) تنها دو اقدام می‌توان انجام داد. رفتن به اتاق D یا F، با یک انتخاب تصادفی از بین این دو اقدام خانه F انتخاب می‌شود. حال باید جدول اقدام-عمل با توجه معادله 1 به‌روز آوری شود.این به‌روز آوری به صورت زیر انجام می‌شود:

|  |  |  |
| --- | --- | --- |
|  | Q= |  |

چون حالتF هدف است، بنابراین عامل در این حالت می‌ماند و یک **بخش**[[76]](#footnote-76) از حل مسئله پایان یافته است. برای **بخش** بعدی، یک حالت ابتدایی دیگر را به صورت تصادفی در نظر می‌گیریم، فرض کنید این حالت تصادفی اتاق D باشد.

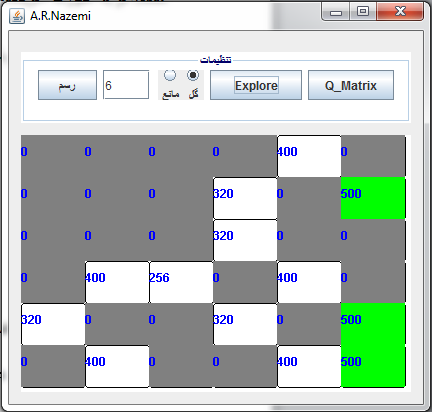
از اتاق D به سه اتاق B,C,E راه وجود دارد(با توجه ‏(9-1)) که یکی را به صورت تصادفی انتخاب می‌کنیم . این انتخابB است، بنابراین برای این حالت به صورت زیر به‌روز آوری می‌شود.

|  |  |  |
| --- | --- | --- |
|  | Q= |  |

چون حالت Bپایانی نیست بنابراین **بخش**76 تمام نشده و باید مجدد از B اتاقی را به تصادف انتخاب کنیم، فرض کنیم این انتخابF باشد بنابراین به صورت زیر به‌روز آوری می‌شود.

که همان مقدار قبلی است .اگر بخش‌ها را به تعداد زیاد تکرار کنیم جدولQ به ‏(12-1) زیر همگرا می‌شود ،که با استفاده از برنامه‌ای که نوشته‌شده این همگرایی در قسمت بعد نشان داده می‌شود:

شکل 6 شمای برنامه برای یافتن راهبرد بهینه خروج از اتاق



|  |  |  |
| --- | --- | --- |
|  | **Q=** |  |

همان طور که در رابطه ‏(6-1) مشخص‌شده راهبرد بهینه در هر حالت، حالتی است که بیش‌ترین ارزش را داشته باشد. بنابراین کافی است ربات در هر حالت با استفاده از ‏(12-1) حرکتی را انتخاب کند که بیش‌ترین ارزش را داشته باشد. که این کار منجر به در پیش گرفتن راهبرد بهینه در تصمیم‌گیری عامل می‌شود.

## یادگیری تقویتی در مسئله مدیریت سفارش

همان‌گونه که در فصل اول گفته شد مدیریت سفارش یک رویکرد ترکیبی[[77]](#footnote-77) برای مشخص کردن مقدار سفارش هر عامل زنجیره تأمین است و هدف آن کمینه کردن هزینه انبار در کل زنجیره تأمین است. این رویکرد تمرکزش بر درخواست‌های زنجیره برای رسیدن به اهداف کاهش هزینه انبار، کاهش کمبود کالا[[78]](#footnote-78)، بالا بردن سطح سرویس به مشتری و افزایش سودمندی کل زنجیره است. برای اینکه بتوانیم این مسئولیت را به عهده عامل هوشمند واگذار کنیم ، باید ابتدا بتوانیم این مسئله را به فرم مسایل مارکوف در بیاوریم. در ادامه مسئله یادگیری تقویتی را در قالب یک مسئله مدیریت سفارش بیان می‌کنیم و در آن متغییرهای حالت، تابع پاداش، تابع ارزش هر عمل و اینکه راهبرد بهینه چه خصوصیاتی باید داشته باشد بیان می‌کنیم.

### حالت سیستم

همان طور که در ‏2.3.3 بیان شد اگر سیستم دارای خاصیت مارکوف باشد با یک گام حرکت می‌تواند گام بعدی و مقدار پاداشی که به ازای آن گام می‌گیرد را پیش‌بینی کنی (یعنی حالت بعدی فقط به حالت فعلی بستگی دارد) با این وجود هنگامی که حالت سیستم کاملاً مارکوف نیست بهتر است آن را به عنوان نیمه مارکوف [[79]](#footnote-79)در نظر بگیریم.

فرآیندهای تصمیم‌گیری نیمه مارکوف حالت توسعه‌یافته70MDPs می‌باشد. در واقع، تفاوت از آنجایی ناشی می‌شود که درMDPs تصمیم‌گیری تنها در نقاط گسسته از پیش تعیین‌شده (در زمان) مجازاست، درحالی‌که در79SMDPs تصمیم‌گیرنده می‌تواند یک اقدام را زمانی انتخاب نماید که حالت سیستم تغییر کند.

علاوه بر این،SMDPs تکامل سیستم را در زمان پیوسته مدل می‌کند همچنین زمان صرف شده توسط سیستم در یک حالت خاص از یک توزیع احتمال پیروی می‌کند. انتخاب عمل درSMDPs تنها عامل تعیین توزیع احتمال مشترک حالت پس از آن نمی‌باشد، بلکه به زمان بین دوره‌های تصمیم‌گیری نیز بستگی دارد. به طور کلی، حالت سیستم ممکن است چندین بار بین دوره‌های تصمیم‌گیری تغییر کند، اما تنها حالت سیستم در دوره تصمیم‌گیری مربوط به تصمیم‌گیرنده مد نظر است).آنچه که بین دوره‌های تصمیم‌گیری اتفاق می‌افتد حائز اهمیت نیست.(

با توجه به این مطلب ما حالت سیستم را به صورت زیر در نظر می‌گیریم.

|  |  |  |
| --- | --- | --- |
|  |  |  |

یعنی حالت کل سیستم مجموعه‌ای از حالت‌های هر کدام از عامل‌هاست که این حالت در زمان تصمیم‌گیری برای سفارش دهی مشخص می‌شود، ممکن است چندین بار بین دو تصمیم‌گیری حالت سیستم عوض شود که ما به آن اعتنایی می‌کنیم.

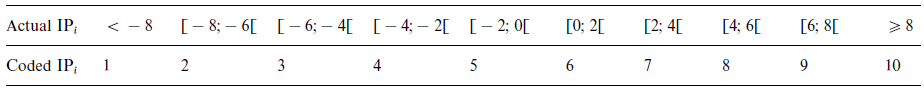
حال که با حالت سیستم آشنا شدیم باید حالت هر عامل را مشخص کنیم. در این مسئله مقدار کالای موجود در انبار هر عامل را به عنوان حالت آن عامل در نظر می‌گیریم که می‌تواند صفر مقداری مثبت یا مقداری منفی[[80]](#footnote-80) باشد. بنابراین هر انبار بینهایت حالت می‌تواند داشته باشد! با توجه به اینکه ما برای ذخیره کردن ارزش هر حالت از جدول استفاده می‌کنیم، باید برای این مشکل راه‌حلی پیدا کنیم.

در مقاله‌هایی که در این زمینه کارشده است حالت انبار را کد بندی می‌کنند یعنی چندین حالت را به عنوان یک حالت در نظر می‌گیرند. و از آن به عنوان نماینده آن حالت‌ها استفاده می‌کنند. درشکل 7 نمونه‌ای از این کد بندی نشان داده شده است.

### پاداش

در مسئله مدیریت سفارش دهی در زنجیره تأمین هدف سیستم کاهش هزینه‌های انبار است.بنابراین سیگنال پاداش باید متناسب با هزینه انبار در نظر گرفته شود ما به تبعیت از بازی نوشابه هزینه انبارش، کمبود موجودی و سفارش هر واحد کالا را در نظر می‌گیریم و این مقدار را به عنوان پاداش با علامت منفی به عامل در هر مرحله تصمیم‌گیری می‌دهیم. علت اینکه این مقدار را منفی در نظر می‌گیریم این است که عامل سعی می‌کند تصمیمی را بگیرد که کمترین پاداش منفی یا بیش‌ترین سود را داشته باشد.

شکل 7 نحوه کد کردن حالت انبار



بنابراین تابع پاداش هر عامل به صورت زیر است.

|  |  |  |
| --- | --- | --- |
|  |  |  |

### تابع ارزش

محاسبه ارزش هر عمل و به‌روز آوری آن در طول اجرای الگوریتم با استفاده از

انجام می‌شود.

### قانون سفارش دهی

برای سفارش دهی کالا به بالادستی ما از قانون X+Y استفاده می‌کنیم در این قانون X مقدار سفارشی است که از طرف خریدار رسیده و ما مقدار Y واحد به آن اضافه می‌کنیم و به بالاسری سفارش می‌دهی مقدار Y می‌تواند مثبت منفی یا صفر باشد عامل در طول یادگیری سعی می‌کند مقدار بهینه را برای Y پیدا کند.

### راهبرد عامل‌ها

در این روش راهبرد عامل‌ها این است که با توجه به جدول ارزش هر عمل،که بعد از آموزش دیدن به دست می‌آید عملی را انتخاب کنند که بیش‌ترین ارزش را در بلندمدت داشته باشد.

**فصل سوم**

**مدل ارایه شده**

# مدل ارایه شده

## مقدمه

در این فصل ما مسئله مدیریت سفارش دهی در زنجیره تأمین را در بازی نوشابه توسط الگوریتم یادگیری تقویتی حل می‌کنیم و ضمن مقایسه نتایج حاصله با الگوریتم‌های فرا ابتکاری به مشکلات موجود در توسعه این الگوریتم در سیستم‌های چند عامله اشاره خواهیم کرد در پایان این فصل بعضی از راه‌حل‌های موجود برای از میان بر داشتن این موانع را بررسی می‌کنیم.تا کنون **بیشتر کارهایی که جهت هوشمند سازی عامل‌ها در زنجیره تأمین انجام‌شده از الگوریتم ژنتیک برای بهینه‌سازی استفاده شده است . در این نوع بهینه‌سازی هر عامل برای کمینه کردن هزینه انبار در کل زنجیره تأمین با همکاری دیگر عامل‌ها قانون اختصاصی راهبرد بهینه مخصوص به خود را یاد می‌گیرد. محدودیت الگوریتم‌های مبتنی بر ژنتیک قانون سفارش دهی ثابت[[81]](#footnote-81) برای هر عامل است قانون سفارش دهی ثابت در محیط‌های پویا که توزیع آماری درخواست‌های مشتری به درستی مشخص نیست، باعث می‌شود این الگوریتم ناکارآمد جلوه دهد، بنابراین تلاش‌های محققان در راستای استفاده از الگوریتم‌هایی است که بتوانند قانون سفارش دهی به صورت پویا را در طول اجرای بازی ارائه دهند. یادگیری تقویتی یکی از این نوع الگوریتم‌هاست. مدل‌سازی بر مبنای یادگیری تقویتی عامل‌ها را قادر می‌سازد از قوانین مختلفی در سفارش دهی کالا استفاده کنند که به نوبه‌ی خود این روش انعطاف‌پذیری بیشتری نسبت به بهینه‌سازی‌های مبتنی بر ژنتیک دارد.**

## پیش‌فرض‌ها

برای اینکه این مسئله را حل کنیم پیش‌فرض‌هایی را در نظر می‌گیریم که بیشتر آن‌ها پیش‌فرض‌هایی است که در بازی نوشابه در نظر گرفته می‌شود، ولی چون قرار است که این مسئله توسط عامل‌های کامپیوتری شبیه‌سازی شود نیاز است که پیش‌فرض‌های دیگری نیز به مسئله افزوده شوند. **در این مسئله ما فرض می‌کنیم یک عامل غیرانسانی کنترل‌کننده موجودی در محیط** W**[[82]](#footnote-82)** **وجود دارد؛ که ادراکات و تصمیمات آن به چند حالت محدود ختم می‌شود. عامل می‌تواند درخواست فعلی و همچنین وضعیت انبار را مشاهده کند، بر اساس این مشاهده عامل می‌تواند چندین عمل محدود[[83]](#footnote-83) را انجام دهد، انجام این اعمال باعث تغییر در حالت محیط می‌شود، همچنین ما فرض می‌کنیم که محیط (حالت انبار،درخواست مشتری) به بازه‌های زمانی گسسته تقسیم می‌شود[[84]](#footnote-84) که در این حالت عامل با مشاهده محیط کار** T**[[85]](#footnote-85) را انجام می‌دهد.**

**هدف ما این است که در نهایت نگاشتی از مشاهده به عمل را ایجاد کنیم که کار** T **را به بهترین نحو انجام دهد (تا جایی که امکان دارد مقرون به صرفه)، چنین نگاشتی به وسیله عبارت زیر بیان می‌شود.**

|  |  |  |
| --- | --- | --- |
|  |  |  |

**ما سه نمونه از این عامل‌ها را در سه سطح به صورت خطی شامل یک خرده‌فروش، یک عمده‌فروش و یک کارخانه در نظر می‌گیریم. پیش‌فرض‌های دیگر که عمدتاً در بازی نوشابه وجود دارند به قرار زیر هستند.**

**سفارش مشتری در بازه زمانی ثابت و با مقادیر متفاوت[[86]](#footnote-86)بین 0 تا 15 واحد به دست خرده‌فروش می‌رسد. سفارش خرده‌فروش، عمده‌فروش و کارخانه در بازه زمانی ثابت و با مقادیر متفاوت بین 0 تا 18 واحد به دست بالاسری می‌رسد. زمان تدارک[[87]](#footnote-87) ثابت و قطعی است. از لحاظ فضای انبارها محدودیتی وجود ندارد. پس افت[[88]](#footnote-88)در این مسئله امکان‌پذیر است و هزینه ثابتی برای هر یک از اجزا دارد. هزینه انبارش کالا در انبار مقدار ثابتی است و برای همه سطوح یکسان است. زمان رسیدن در خواست (جریان اطلاعات) به بالاسری صفر واحد زمانی است. ظرفیت ناوگان حمل و نقلی برای توزیع محصول محدود است (یک کامیون با حداکثر ظرفیت 18 واحد کالا). یک نوع محصول در زنجیره تولید و توزیع می‌شود. هر عامل وظیفه بهبود سطح موجودی خود را با توجه به وضعیت سایر انبارها دارد.**

## موانع موجود

حال که با فرضیات مسئله آشنا شدیم، می‌توانیم راجع به مسایلی که در این پروژه مطرح می‌شود و باید بر آن غلبه کنیم تا بتوانیم یک سیستم هوشمند کنترل موجودی را طراحی کنیم صحبت می‌کنیم

### هوشمند سازی عامل

یک رویکرد منطقی این است که از یک سری داده اولیه شامل جفت (مشاهده-اقدام) برای آموزش عامل استفاده کنیم تا عامل نگاشت بهینه‌ی از حالت به عمل را یاد بگیرد نگاشتی که در این فرایند یاد گرفته‌شده بعد می‌تواند به عنوان سیاست کنترلی عامل برای تعمیم به حالت‌هایی که تابه حال ندیده است مورد استفاده قرار گیرد.

این رویکرد یادگیری با ناظر[[89]](#footnote-89) نام دارد و تا زمانی که ما بخواهیم یک راهبرد نیمه بهینه را برای عامل یاد بگیریم باید اقدامات درست را برای هر حالت در داده‌های آموزشی فراهم کنیم. با توجه به اینکه ما بهترین تصمیم را برای هر حالت نمی‌دانیم، بنابراین استفاده از این نوع یادگیری برای عامل کنترل‌کننده انبار مفید نمی‌باشد. مشکل دیگر این نوع یادگیری این است که یادگیری با ناظر فقط قادر به یادگیری مسایلی است که توزیع داده‌های آن را قبلاً دیده باشد. به عنوان مثال اگر درخواست‌های مشتری از توزیع نرمال پیروی کند و ما داده‌های آموزشی را بر اساس توزیع نرمال ایجاد کرده باشیم آنگاه اگر از عامل درخواست‌هایی خارج از آن توزیع داشته باشیم این نوع یادگیری نمی‌تواند راهبرد بهینه را برای یادگیری پیدا کند.

بنابراین انتخاب الگوریتمی که بتواند در چنین محیط پویایی راهبرد بهینه را پیدا کند یکی از مشکلات موجود در حل این مسئله است.

### کمبود داده‌های آموزشی

تا زمانی که ما بخواهی داده‌های موجود را با تعامل با محیط واقعی به دست بیاوریم نرخ داده‌هایی که برای آموزش به دست می‌آوریم محدود است. دانشی که عامل از هر بار مشاهده محیط به دست می‌آورد شامل مقدار تقاضای مشتری و همچنین وضعیت انبار است، با توجه به اینکه ما دوست داریم عامل به صورت آنلاین آموزش ببیند وجود این داده‌های کم که به صورت گام به گام (گام‌های زمانی) به دست می‌آیند در مراحل ابتدایی یادگیری یکی دیگر از موانع ما در حل این مسئله است.

### دانش اولیه کم راجع به محیط

بسیاری از سیستم‌های یادگیر سعی می‌کند که بدون هیچ دانش اولیه شروع به یادگیری کنند، اگرچه این مسئله بسیار جالب توجه است ولی وقتی در مسئله عامل هوشمند کنترل‌کننده موجودی، می‌خواهیم از الگوریتم یادگیر استفاده کنیم عدم شناخت محیط مشکلاتی را برای فرایند یادگیری ایجاد می‌کند، نبود دانش اولیه عامل را مجبور می‌کند در ابتدای فرایند یادگیری به صورت تصادفی اقدام کرده و در حالت‌های مختلف حرکت‌های تصادفی انجام دهد، با توجه به تعداد حالت‌های زیادی که یک عامل می‌تواند طی فرایند یادگیری داشته باشد (رجوع کنید به ‏3.3.4) اگر عامل به این شیوه اقدام کند بسیاری از وقت عامل صرف حرکت‌های تصادفی می‌شود و به تبع آن از کارایی الگوریتم در یادگیری آنلاین می‌کاهد.

### فضای حالت بسیار بزرگ

در مسئله کنترل موجودی با توجه به اینکه وضعیت انبار نامتناهی است در بیشتر پژوهش حالت‌های انبار را کد بندی می‌کنند نمونه‌ای از این کد بندی در جدول نمایش داده شده است، استفاده از این کد بندی که نوعی گسسته سازی است علاوه بر اینکه باید دقت بالایی در انتخاب طول بازه‌ها و نقاط مرزی داشته باشد، موجب ادغام حالت‌ها با یکدیگر و باعث می‌شود حالت‌های مخفی[[90]](#footnote-90) را در فرایند آموزش در نظر نگیریم.

علاوه بر این چون ما این مسئله را به صورت متمرکز[[91]](#footnote-91) حل می‌کنیم بنابراین عامل باید علاوه بر این که حالت انبار خود را بداند باید از حالت انبار دیگر عامل‌ها نیز باخبر باشد که در مسئله بازی نوشابه این حالت برای هر عامل به صورت زیر است.

حالت فعلی (مراجعه کنید به ‏(13-1)) که با توجه به نامحدود بودن حالت انبارها (و یا به صورت کد درآمدن آن‌ها) فضای حالت بسیار بزرگی می‌شود به این فضا چندین عامل در شبکه‌ای از زنجیره تأمین‌ها با انبارهای متفاوت را اضافه کنید.

## تخمین تابع ارزش

در فصول قبل یادگیری تقویتی را معرفی کردیم و یادگیری Q را به عنوان الگوریتمی برای یادگیری تابع ارزش شرح دادیم، همچنین نشان دادیم یادگیری Q چطور از جدول حالت-عمل برای رسیدن به پاداش در بلند مدت استفاده می‌کند، باتوجه به اینکه جدول حالت –عمل می تواند محدود باشد وبا بزرگتر شدن فضای حالت استفاده از جدول از لحاظ زمان وفضای مورد نیاز مقرون به صرفه نمی باشد بنابراین نیاز است که ما جدول ارزش را با تابع ارزش تخمین بزنیم در این فصل به معرفی تابع ارزش می پردازیم, سپس مسئله بازی نوشابه را با تخمین جدول حالت-عمل حل می‌کنیم ونتایج آنرا با استفاده از روش سنتی در یادگیری Q که همان استفاده از جدول است مقایسه می‌کنیم.

## تخمین زدن جدول حالت- عمل

همان طور که گفتیم محاسبه تابع ارزش به این صورت است که کافی است الگوریتم نگاهی به جدول حالت عمل بیندازد که با توجه به شماره حالت بهترین عمل را (ماکزیمم گیری روی ردیف جدول) را انتخاب کند، این‌طور نمایش تابع Q همیشه امکان‌پذیر نیست. هرگاه فضای حالت- عمل بسیار بزرگ شود دیگر امکان ذخیره آن در حافظه وجود ندارد. اگر ما بتوانیم جدول را به نحو خوبی تعمیم دهیم و سپس از یک تابع تقریب زن برای بیان تابع ارزش استفاده کنیم آنگاه می‌توانیم از الگوریتم یادگیری تقویتی در مسایل با فضای حالت بزرگ یا پیوسته استفاده کنیم. همان طور که در فصل قبل نشان دادیم این الگوریتم توانایی بسیار بالایی در پیدا کردن راهبرد بهینه سفارش دهی دارد.

در فصل قبل دیدیم که گسسته سازی یکی از راهکارهای موجود برای کاهش فضای حالت مسئله بود. شاید این سؤال به ذهن برسد که چرا گسسته سازی برای حالت‌های پیوسته نمی‌تواند در کاهش فضای حالت موثر باشد؟ در حل مسئله کنترل موجودی با استفاده از یادگیری تقویتی که از جدول برای نمایش ارزش هر عمل استفاده می‌کرد، ما با کد بندی (گسسته سازی) حالت انبار به 10 حالت برای سه عامل 1000 حالت ایجاد شد که در هر حالت 343 عمل می‌توانست انجام شود برای آموزش این الگوریتم نیاز بود که عامل در هر حالت بیش از یک بار قرار بگیرد تا الگوریتم همگرا شود. (Singh, 1999) نشان دادند که برای همگرا شدن الگوریتم یادگیری تقویتی با n حالت یک راهبرد نزدیک به بهینه نیاز به O() جفت حالت عمل برای آموزش دارد.) بنابراین در مسئله کنترل موجودی با توجه به کد کردن حالت انبار که باعث می‌شود فضای حالت کاهش پیدا کند (با در نظر گرفتن حالت‌های پنهان که با کد کردن مجموعه‌ای از حالت‌ها ممکن است در نظر گرفته نشوند) و با توجه به اینکه ما یک سیستم کوچک شامل 3 عامل را شبیه‌سازی کردیم، 1000 حالت برای ذخیره کردن در حافظه، بسیار زیاد است و باعث می‌شود روند آموزش به کندی پیش برود. به این پیچیدگی 343 حرکت برای هر حالت را نیز اضافه کنید. بنابراین متوجه می‌شویم که استفاده از الگوریتم یادگیری تقویتی در مسئله کنترل موجودی بدون استفاده از تخمین تابع کاری زمان بر و غیرکارا است. بنابراین لازم است که برای مسایل پیوسته و فضای حالت‌های بزرگ از تابعی استفاده کنیم که جدول حالت – عمل را تخمین بزند.

به نظر می‌رسد استفاده از تابع تقریب زن به جای جدول حالت-عمل یک راه حل منطقی برای مسئله‌های با فضای حالت بزرگ است. با این حال استفاده از چنین تابعی مسایل جدیدی را ایجاد می‌کند که برای استفاده از این تابع در ابتدا باید بر این مسایل جدید غلبه کنیم.

بنابراین استفاده از تابع تخمین زننده ارزش این‌گونه نیست که یک الگوریتم یادگیری با ناظر را به راحتی جایگزین جدول کنیم. هر الگوریتم یادگیری با ناظر هنگامی که روی داده‌های واقعی تست می‌شود دارای خطای تخمین است. هرگاه از چنین تخمین زننده‌هایی به جای تابع تخمین ارزش استفاده کنیم این خطاهای کوچک به سرعت انباشته می‌شوند و تابع تخمین را غیرقابل استفاده می‌کنند. در یادگیری Q انتخاب عمل در حالت باعث می‌شود که عامل به حالت برود و یک پاداش دریافت کند و باعث میشود در جدول مقدار به صورت زیر به روز آوری شود.

|  |  |  |
| --- | --- | --- |
|  |  |  |

مقدار حالت-عمل جفت بر اساس نرخ یادگیری و فاکتور نزولی و مقدار تخمین ما از بیش‌ترین مقداری که در حالت بعدی نصیب ما می‌شود به دست می‌آید[[92]](#footnote-92). نکته‌ای که در اینجا باید به آن اشاره شود این است که مقدار جدید برای بر مبنای مقدار فعلی است که از تخمین زدن به دست آمده و همچنین ماکزیمم مقداری که در حالت به دست می‌آید به این معنی که خطاهای تخمین این دو مقدار در به‌روز آوری تاثیر دارد. تاثیر این مقدار برای به‌روز آوری حالت دیگر مورد استفاده بگیرد ، خطا همین طور بزرگ و بزرگ تر می‌شود تا اینکه خطا بر تقریب چیره می‌شود و باعث می‌شود تابع تقریب به درستی جواب ندهد.

### کاهش خطای تقریب زدن

برای کاهش خطا در تخمین زدن باید به دنبال منبع خطا بگردیم، (Thrun, 1993), (Gordon, 1995) در مقالاتشان یکی از منابع خطا را هنگام استفاده از تابع تقریب زن، بیش از اندازه تخمین زدن[[93]](#footnote-93) است یکی از مهم‌ترین دلایل بیش از اندازه تخمین مربوط به مشکلی است که در آمار با نام نقاط مخفی برون‌یابی[[94]](#footnote-94) شناخته می‌شود، تخمین تابع تقریب زن به طور کلی برای همه نقاط Query مناسب تا زمانی که ما نتوانیم یک فرض قوی در باره تابعی که می‌خواهیم یاد بگیریم داشته باشیم ما فقط می‌توانیم تخمین را در ناحیه‌ای که داده‌های آموزشی قرار دارند انجام دهیم، به عبارت دیگر ما از تابع تقریب زن فقط می‌توانیم روی‌داده‌های که درون بازه داده‌های آموزشی هستند استفاده کنیم نه داده‌های خارج از آن.

یک راه حل برای این مشکل این است که یک پوسته محدب اطراف داده‌های آموزشی بسازیم و فقط به نقاطی که درون این پوسته قرار می‌گیرند پاسخ دهیم. حال مسئله این است که چطور این پوسته را بسازیم؟

(Cook, 1979) چنین ساختاری را پوسته مستقل از متغییر (IVH)[[95]](#footnote-95) نام و پیشنهاد داد که بهترین تعادل بین کارایی و ایمنی استفاده از چنین پوسته‌ای در تخمین زدن است به این صورت که اگر نقطه داخل پوسته قرار گیرد ما آن نقطه را برای تخمین زدن می‌پذیریم در غیر این صورت از تخمین زدن آن نقطه اجتناب می‌کند. ماتریس X که ردیف‌های آن متناظر داده‌های آموزشی است ماتریس hat[[96]](#footnote-96) به صورت زیر محاسبه می‌شود

یک نقطه دلخواه درون پوسته قرار دارد اگر که

که در آن عناصر روی قطر اصلی ماتریس V هستند. بنابراین برای اینکه ما از ایجاد خطا در تخمین جلوگیری کنیم این آزمون را روی نقطه Query برسی می‌کنیم، تا اینجا لزوم استفاده از توابع تقریب زن و همچنین موانع استفاده از این نوع توابع را در یادگیری تقویتی بیان کردیم ،همچنین روش (Cook, 1979) را برای کاهش اثرات خطای ایجادشده در تخمین بیان کردیم در ادامه به معرفی یکی از روش‌هایی که در یادگیری ماشین از تقریب زدن استفاده می‌کنند می‌پردازیم.

# یادگیری مبتنی بر نمونه

یادگیری مبتنی بر نمونه در مرحله آموزش فقط داده‌های آزمون را ذخیره می‌کند و تعمیم آن‌ها را وقتی انجام می‌دهد که داده‌ی جدید وارد شود، هرزمان که یک داده جدید وارد شود،رابطه آن با داده‌هایی که قبلاً ذخیره‌شده‌اند بررسی می‌شود تا یک خروجی مناسب برای داده فعلی تولید شود. یادگیری مبتنی بر نمونه از روش‌های نزدیک‌ترین همسایه و رگرسیون وزن دار محلی[[97]](#footnote-97) که هر نمونه را مانند نقطه‌ای در فضای اقلیدسی در نظر می‌گیرند استفاده می‌کند، همچنین استدلال مبتنی بر نمونه[[98]](#footnote-98) که پیچیده‌تر نیز هستند و هر نمونه را مانند یک نماد[[99]](#footnote-99) در نظر می‌گیرند. روش‌های مبتنی بر نمونه گاهی با نام یادگیری "تنبل[[100]](#footnote-100)" شناخته می‌شوند، چرا که در مرحله آموزش هیچ کاری انجام نمی‌دهند (فقط داده‌های آموزشی ذخیره می‌شوند) و یادگیری تا زمانی که داده جدید وارد نشده به تعویق می‌افتد. نحوه کار این الگوریتم به این صورت است که هنگامی که تقاضای تخمین نقطه‌ای داده می‌شود این الگوریتم از نقاطی که قبلاً ذخیره‌شده و به داده فعلی نزدیک‌تر هستند(بر اساس معیارهای اندازه‌گیری فاصله) برای ایجاد خروجی مطلوب استفاده می‌کند، به عنوان مثال ساده‌ترین نوع از یادگیری مبتنی بر نمونه "1-نزدیک‌ترین همسایه" است که مقدار نزدیک‌ترین نقطه به نقطه درخواستی را برمی‌گرداند، بار محاسباتی این نوع الگوریتم‌ها برخلاف مرحله آموزش در مرحله تخمین بالا است .

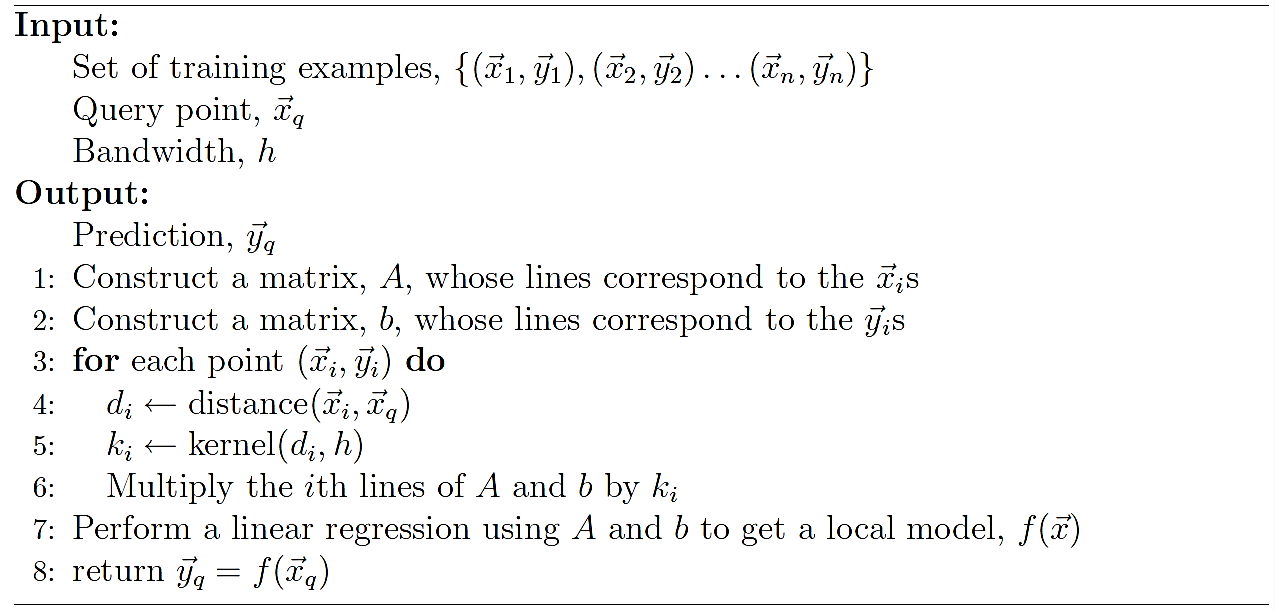
الگوریتم‌های مبتنی بر نمونه قادرند با تعداد کمی داده آموزشی تخمین‌های خوبی بزنند ولی مشکل عمده آن‌ها این است که محاسبات زیادی هنگام تخمین یک نقطه انجام می‌دهند، به خاطر اینکه آن‌ها نقطه درخواستی را با همه داده‌های آموزشی مقایسه می‌کنند که این مقایسه از درجه پیچیدگی O(n) است به علاوه پیچیدگی تخمین واقعی بر اساس نقاط ذخیره‌شده

برای تخمین تابع ارزش در یادگیری تقویتی ما زا این نوع الگوریتم‌ها استفاده می‌کنیم وسعی داریم به نحوی معایب ذکرشده در بالا را برای تخمین بهتر بر طرف کنیم

## رگرسیون وزن دار محلی

رگرسیون وزن دار محلی یک شاخه از رگرسیون استاندارد خطی است که در آن نقاط آموزشی نزدیک‌تر به نقطه مورد در خواست تاثیر بیشتری بر خط(پوسته) رگرسیون دارند. با داشتن یک سری داده آموزشی رگرسیون خطی یک مدل خطی را منطبق بر داده‌های آموزشی می‌سازد که کمترین مربع خطای پیش‌بینی را روی همه داده‌های آموزشی داشته باشد، مدل ایجادشده این ادعا را دارد که مدل کلی داده‌های آموزشی را ایجاد کرده است. LWR از جهت دیگر فقط تابع پیش‌بینی را به طور محلی ایجاد می‌کند، بدون هیچ فرضی در باره داده‌های کلی. فرایند آموزش LWR شامل ذخیره کردن داده‌های وزن دهی به داده ها و استفاده از رگرسیون استاندارد برای تخمین داده‌های جدید است. درالگوریتم 2 نحوه تخمین یک نقطه توسط این الگوریتم نمایش داده شده است.

الگوریتم 2 رگرسیون وزن دار محلی



نقاط آموزشی و Query به صورت جفت بردارهای واقعی () در نظر گرفته می‌شوند. این الگوریتم ماتریسA را شبیه رگرسیون خطی به صورت زیر می‌سازد

ردیف‌های ماتریس A متناظر با نقطه هستند. ماتریس دیگر یعنی b نیز به صورت زیر است که ردیف‌های آن متناظر با است.

یک رگرسیون استاندارد معادله Ax=b را حل می‌کند. و معادله کلی خط(پوسته) را برای تخمین بقیه نقاط ایجاد می‌کند. ولی در LWR ما ابتدا به داده(نقاط) های آموزشی وزن می‌دهیم و پس از آن معادله را با نقاط جدید حل می‌کنیم ، بنابراین نقاطی که به Query نزدیک‌ترند تاثیر بیشتری بر رگرسیون دارند.

در الگوریتم 2 برای هر نقطه در مجموعه آموزشی ، فاصله آن تا داده Query محاسبه می‌شود(خط 4)، همچنین یک تابع کرنل[[101]](#footnote-101) با پهنای باند[[102]](#footnote-102) h، برای وزن دادن به داده‌های آموزشی به‌کاربرده می‌شود، پس از وزن دار کردنِ دو ماتریس، از رگرسیون استاندارد خطی برای ایجاد یک مدل محلی استفاده می‌شود (خط 7 الگوریتم 2).

LWR برای هر نقطه Queryیک مدل محلی ایجاد می‌کند بنابراین با رگرسیون استاندارد تفاوت دارد چرا که در رگرسیون استاندارد یک مدل کلی برای همه داده ها ایجاد می‌شود. این خاصیت به LWR اجازه مدل کردن تابع‌های بسیار پیچیده را می‌دهد. همچنین بیانگر این نکته است که LWR بسیار محاسبات بیشتری نسبت به رگرسیون استاندارد دارد. رگرسیون وزن دار محلی به عبارت دیگر برای هر نقطه Query یک خط می‌سازد (fit new line) و این خط فقط برای پاسخ دادن به این نقطه ایجاد می‌شود وبری نقطه query دیگر یک خط متفاوت دیگر می‌سازد

در الگوریتم 2 از دو تابع استفاده شده است ، یکی برای محاسبه فاصله Query از داده‌های آموزشی و دیگری برای وزن دار کردن نقطه‌ها در ادامه این دو تابع را توضیح می‌دهیم.

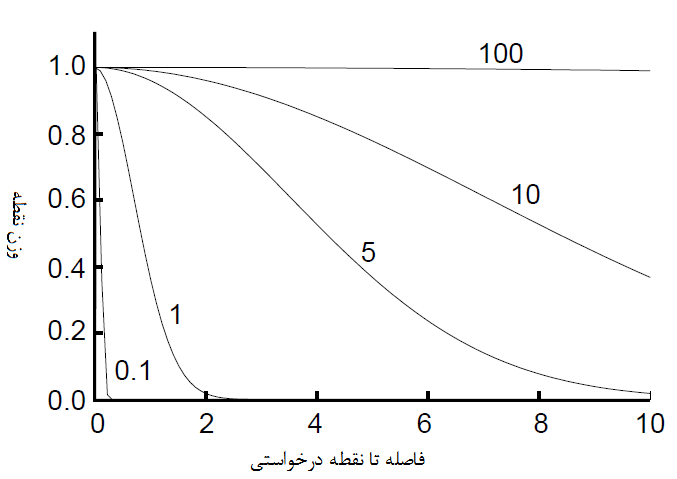
### تابع کرنل

تعریف:تابع کرنل در الگوریتم 2 برای تبدیل فاصله هر نقطه آموزشی به وزن مناسب استفاده شده است ،این تابع دارای پارامتری به نام پهنای باند است که میزان تاثیر نقاط آموزشی را بر نقطه Query کنترل می‌کند. تابعی که عموماً به عنوان کرنل استفاده می‌شود تابع Gaussian است و به صورت زیر تعریف می‌شود.

که در آن d فاصله نقطه query از نقطه فعلی است و h پهنای باند نامیده می‌شود. ما از این تابع برای تخمین تابع ارزش در یادگیری تقویتی استفاده می‌کنیم. علت اینکه ما از Gaussian استفاده می‌کنیم این است که طیف وسیعی از داده‌ها زا پوشش می‌دهد و هیچ‌گاه وزن صفر را برنمی‌گرداند( به عنوان خروجی). از توابع دیگر نیز می‌توان با رعایت سه شرط زیر به عنوان کرنل استفاده کرد.

شکل 8 چند مثال از تابع کرنل Gaussian را به همراه پهنای باند نشان می‌دهد

شکل 8 تابع کرنل Gaussian با پهنای باند متفاوت



پهنای باند یک معیار اندازه‌گیری است که ما مشخص می‌کنیم چقدر تابعی که می‌خواهیم مدل شود نسبت به داده‌های اطراف حساس باشد. اگر این مقدار برابر یک باشد همان طور که از نمودار مشخص است به همه نقاط اطراف وزن یکسان می‌دهد ولی اگر این مقدار کمتر از یک و به صفر نزدیک باشد تاثیر نقاط نزدیک‌تر به نقطه query در مدل ایجادشده بیشتر است. به عبارت دیگر h زیاد الگوریتم را به رگرسیون استاندارد نزدیک می‌کند و h کوچک الگوریتم را به "1-نزدیک‌ترین همسایه" تبدیل می‌کند.

### محاسبه فاصله در رگرسیون وزن دار محلی

LWR از فاصله هر نقطه تا query برای وزن دادن به داده‌های آموزشی استفاده می‌کند. به همین دلیل انتخاب روش اندازه‌گیری فاصله تاثیر زیادی روی نحوه تخمین زدن تابع دارد . در این الگوریتم عموماً فاصله اقلیدسی استفاده می‌شود.

که در آن وزنی است که به هر بعد نقطه داده می‌شود. به عنوان مثال در حل مسئله کنترل موجودی ما هر حالت انبار را یک نقطه در نظر می‌گیریم چون این مسئله به صورت متمرکز[[103]](#footnote-103) حل می‌شود. بنابراین هر حالت ما شامل وضعیت انبار 3 عامل است مثلاً یکی از حالت‌ها می‌تواند به صورت بیان شود، که بیانگر این است که خرده‌فروش دارای سفارش عقب‌افتاده، عمده‌فروش و کارخانه به ترتیب دارای 4 و6 واحد کالا در انبار هستند. حال اگر ما بخواهیم تاثیر وزنی خرده‌فروش را در رگرسیون وزن دار محلی بالا ببریم می‌توانیم وزن انبار خرده‌فروش را نسبت به دیگر انبارها بیشتر در نظر بگیریم. در این مسئله ما برای ساده کردن مسئله وزن همه انبارها را یکسان در نظر می‌گیریم، بدیهی است تاثیر وزن انبارها می‌تواند به عنوان کارهای آینده این پژوهش در نظر گرفته شود.

همچنین از پارامتر وزن می‌توان برای نرمال سازی استفاده کرد.علاوه بر فاصله اقلیدسی ، مقیاس‌های دیگری همچون فاصله مینکوفسکی[[104]](#footnote-104) و فاصله منهتن[[105]](#footnote-105) نیز می‌تواند به عنوان مقیاس اندازه‌گیری فاصله استفاده شود.

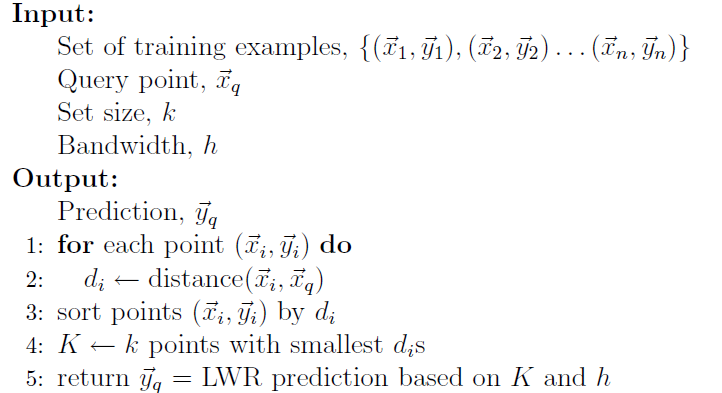
در ادامه الگوریتمHEDGER را که با استفاده از رگرسیون وزن دار محلی و همچنین پوسته مستقل از متغیر( که در‏3.5.1 معرفی شد)در تز دکترای William D. Smart [[106]](#footnote-106) با عنوان “Making Reinforcement Learning Work on Real Robots” معرفی شده است. را بیان می‌کنیم، این الگوریتم در مسئله مسیریابی ربات که در محیطی پیوسته قرار دارد نتایج قابل قبولی در تخمین تابع ارزش در یادگیری تقویتی داشته است.

## الگوریتم پیش‌بینی کننده HEDGER

HEDGER یک الگوریتم مبتنی بر نمونه است که از رگرسیون وزن دار محلی برای تخمین زدن استفاده می‌کند، این الگوریتم توسعه‌یافته از الگوریتم رگرسیون خطی استاندارد است که در آن نقاط نزدیک‌تر به نقطه query نسبت به نقاط دیگر تاثیر بیشتری در خط (پوسته) رگرسیون دارند. برای هر نقطه Query یک رگرسیون جدید در نظر گرفته می‌شود.این مدل یک مدل خطی نمی‌باشد ولی به سادگی مدل خطی است. مدل‌های خطی محلی می‌توانند با تکنیک‌های شناخته‌شده‌ای تقریب‌های خوبی را از مدل کلی بزنند(برای مطالعه بیشتر به (Atkeson, 1997)مراجعه کنید)

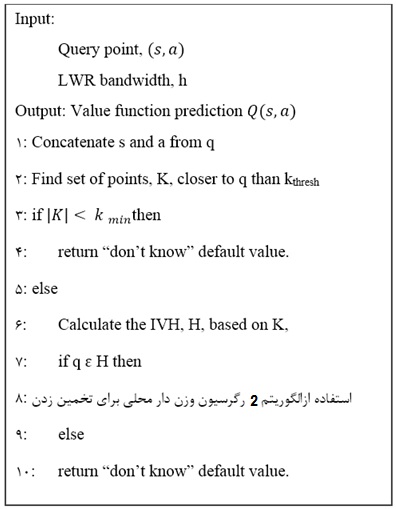
نقاط آموزشی در LWR بر اساس تابعی که به نسبت فاصله‌شان از نقطه Query به آن‌ها رتبه می‌دهد، وزن می‌گیرند. این تابع عموماً یک تابع کرنل از قبیل Gaussian است که در ‏3.6.1و‏3.6.2 معرفی شدند، این تابع یک پارامتر "عرض" دارد که پهنای باند نامیده می‌شود و هرچه بزرگ‌تر باشد به این معنی است که نقاط دورتر نیز در خط(پوسته) رگرسیون تاثیر گذارند که نتیجه آن ایجاد یک رگرسیون کلی و نرم تر[[107]](#footnote-107) است وبینهایت بودن آن رگرسیون محلی را به یک رگرسیون کلی(عمومی) تبدیل می‌کند.برعکس پهنای باند کوچک باعث می‌شود تخمین زدن بسیار حساس به نقاط اطرافش باشد.

یکی از مشکلاتی که استفاده از LWR به عنوان تخمین زننده در یادگیری تقویتی دارد هزینه بالای محاسبات در آن است. برای n داده آموزشی که هر کدام d خصوصیت (بُعد) دارند اگر از همه داده‌های آموزشی برای تخمین استفاده کنیم پیچیدگی محاسباتی از درجه O(nd2+ d3) است . (Smart, 2002) با وجود اینکه بسیاری از داده‌های آموزشی که در فاصله دور تر از query قرار دارند تاثیر بسیار کمی روی تخمین دارند. اگر ما بخواهیم در تخمین محلی خوبی بزنیم در و همچنین محاسبات را کاهش دهیم بهتر است که از نقاط دوردست را که تاثیر کمی بر جواب نهایی دارند چشم‌پوشی کنیم.استفاده از مجموعه نقاط نزدیک به نقطه Query دز تخمین زدن هزینه محاسبات را تا حد زیادی کاهش می‌دهد.

برای انتخاب یک زیرمجموعه مناسب از نقاط همسایه نقطه Query ما تعدادk تا از نزدیک‌ترین همسایه‌های نقطه query را انتخاب می‌کنیم،به این صورت که فاصله نقطه query را از داده‌های آموزشی محاسبه می‌کنیم و k کمترین آن را به عنوان k نزدیک‌ترین همسایه انتخاب می‌کنیم[[108]](#footnote-108). در الگوریتم 3 استفاده از K نزدیک‌ترین همسایه برای رگرسیون وزن دار محلی نشان داده شده است.

الگوریتم 3 رگرسیون وزن دار محلی با استفاده از k نزدیکترین همسایه

**Error! Reference source not found.** تخمین زننده HEDGER را نشان می‌دهد.این الگوریتم با ابتدا زوج (عمل-حالت) را به یک بردار تبدیل می‌کند(خط یک) سپس k تا از نزدیک‌ترین همسایه‌ها را برای تخمین زدن نقطه query طبق خط 1 تا 4 **Error! Reference source not found.** انتخاب می‌کند اگر موفق به انتخاب تعداد حداقل همسایه نشد مقدار پیش‌فرض را برمیگرداند، در غیر این صورت پوسته IVH را اطراف داده‌های kنزدیک‌ترین همسایه ایجاد می‌کند(خط6) و آزمون موجود بودن query داخل پوسته را انجام می‌دهد اگر نقطه داخل پوسته بود رگرسیون محلی را همانند **Error! Reference source not found.** انجام می‌دهد وگرنه مقدار پیش‌فرض را به عنوان نتیجه تخمین بر می‌گرداند . مقدار پیش‌فرض شبیه حالتی است که ما جدول حالت- عمل را قبل از فرایند آموزش با مقداری(عموماً صفر) پر می‌کنیم

الگوریتم 4 به این منظور طراحی‌شده که یک تخمین مطمئن را برای نقطه Query بزند .هر جا این الگوریتم نسبت به توانایی تخمین زدن نامطمئن باشد مقدار پیش‌فرض یا عبارت"نمی‌دانم" را به عنوان پاسخ بر می‌گرداند.

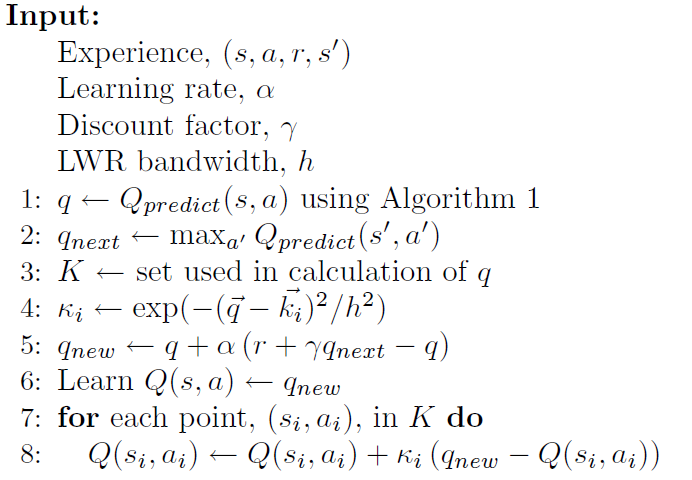
الگوریتم 4 تخمین زننده HEDGER

به نظر می‌رسد که الگوریتم HEDGER قابل جایگزینی با جدول state-action می‌باشد در مرحله بعد ما الگوریتم یادگیری Q که با استفاده از الگوریتم HEDGER مقدار Q\_value را تخمین و به‌روز آوری می‌کند معرفی می‌کنیم.

## الگوریتم آموزشی HEDGER

در این بخش ما چگونگی تقریب زدن الگوریتم HEDGER را در یادگیری تقویتی شرح می‌دهیم ، ونشان می‌دهیم این الگوریتم چطور از تجربه‌های قبلی برای تخمین ارزش حالت فعلی استفاده می‌کند. آنچه عامل در هر حالت می‌تواند ادراک کند با چهارتایی مشخص می‌شود.به این صورت که عامل با انجام حرکت a در حالت s پاداش آنی r را دریافت می‌کند و به حالت s’ می‌رود. الگوریتم 5 نشان می‌دهد که HEDGER چگونه از تجربه‌اش برای بروز آوری تابع تقریب زن استفاده می‌کند. ابتدا ما باید مقدار Q-value حالت فعلی . ماکزیمم مقداری که از حالت جدید s’ نصیب عامل می‌شود را تخمین بزنیم.

الگوریتم 5 q-learning با استفاده ازHEDGER



در خط یک ما Q\_value حالت فعلی را با استفاده از الگوریتم 5 تخمین می‌زنیم،پس از آن در همسایه‌های نقطه s جستجو می‌کنیم و با تخمین زدن Q\_value همسایه‌ها بیش‌ترین مقدار آن را به عنوان qnext مشخص می‌کنیم[[109]](#footnote-109)(خط 2) در خط 3 همسایه‌های جستجو شده در مرحله قبل را به عنوان مجموعه Kنزدیک‌ترین همسایه در نظر می‌گیریم،در خط 4 به هر یک از اعضای مجموعه K وزن مناسب را با توجه به فاصله‌شان از نقطه s با استفاده از تابع کرنل می‌دهیم ، در خط 5 مقدار تخمین شده را با توجه به قانون استاندارد به‌روز آوری Q\_learning مشخص می‌کنیم.ما از این مقدار برای به‌روز آوری همسایه‌های نقطه s در خط 8 استفاده می‌کنیم.به این صورت که نقاط نزدیک‌تر به s تاثیر پذیری بیشتری نسبت به دیگر همسایه‌ها از به‌روز آوری Q\_value خواهند داشت.

به‌روز آوری همسایه‌ها به این شکل به ما امکان می‌دهد که تابع تقریب زن نرم داشته باشیم و به‌روز آوری نقطه‌ای که درون داده‌های آموزشی نیست با به‌روز آوری همسایه‌های آن در داده‌های آموزشی که در حافظه ذخیره‌شده‌اند انجام می‌شود.

بنابراین تا به اینجا با محدود کردن دامنه نقاط پیش‌بینی و همچنین بهبود الگوریتم رگرسیون و انتخاب تعداد محدودی از همسایه‌ها برای تخمین زدن تابع ارزش محدودیت‌های موجود بر سر راه تابع تخمین تا حدود زیادی بر طرف شده است. در بخش بعد ما با استفاده از این الگوریتم مدل کنترل موجودی در زنجیره تأمین را طراحی می‌کنیم و توانایی این الگوریتم در تصمیم‌گیری داینامیک در محیط تصادفی و پویا را مورد بررسی قرار می‌دهیم

**فصل چهارم**

**اجرای مدل ونتایج**

# پیاده‌سازی الگوریتم

## بهبود وپیاده سازی الگوریتم های ژنتیک و یادگیری Q

در این قسمت ما ابتدا این مسئله را با الگوریتم ژنتیک حل می‌کنیم و سپس نحوه پیاده‌سازی آن را با الگوریتم یادگیری Q بیان می‌کنیم. به علت اینکه هدف این پایان‌نامه حل مسئله با الگوریتم‌های فرا‌‌ ابتکاری نیست بنابراین نحوه پیاده‌سازی و جزییات الگوریتم ژنتیک بررسی نمی‌شود.

### بیان جزیات مساله

ما در این مسئله سه عامل خرده‌فروش، عمده‌فروش و کارخانه را در نظر گرفته‌ایم که هر کدام مسئول بهینه‌سازی مقدار سفارش خود با توجه به وضعیت دیگر عامل‌ها هستند، به عبارت دیگر ما این مسئله را به صورت متمرکز centralized حل می‌کنیم. حالت‌هایی که هر عامل می‌تواند با توجه به مقدار کالا در انبار داشته باشد به صورت زیر کد شده‌اند.

جدول 1 موجودی به اعدادی بین 0 تا 9 کد می‌شود

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  | موجودی |
| 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | کد |

بنابراین هر عامل 10 حالت دارد ، حال باید تصمیماتی که عامل می‌تواند در هر حالت بگیرد نیز مشخص شود. در ‏2.5.4 قانونX+Y را معرفی کردیم طبق این قانون اگر **اندازه درخواست پایین‌دستی باشد درخواستی که به بالادستی سفارش داده می‌شود برابر است.** مقدار Y می‌تواند مثبت منفی یا صفر باشد.

عامل در طول یادگیری سعی می‌کند مقدار بهینه را برای Y پیدا کند **. ما در این پیاده‌سازی از الگوریتم یادگیری Q مقدار** Y **را در بازه [-3,3] در نظر می‌گیریم. بنابراین عامل در هر حالت باید یکی از اعداد 3- تا 3 را برای اضافه کردن به در خواست پایین‌دستی انتخاب کن و با افزودن آن به X آن را برای بالادستی ارسال کند .مقدار پاداشی که ما به عامل به ازای هر تصمیم می‌دهیم برابر با هزینه انبار در آن هفته است(با علامت منفی) طبق بازی نوشابه هزینه انبارش هر واحد کالا در انبار به ازای هر هفته 0.5 واحد پولی و هزینه پس افت 1 واحد پولی است، همچنین هزینه سفارش دهی برابر صفر در نظر گرفته شده است بنابراین مجموع این هزینه‌ها سیگنال پاداش را مشخص می‌کند.**

### پیاده‌سازی با الگوریتم ژنتیک

روشی که ما در این پایان‌نامه استفاده کرده‌ایم مبتنی بر الگوریتمی است که اولین بار در (S.O. Kimbrough, 2002)پیشنهاد شد.علاقمندان جهت مطالعه بیشتر می‌توانند به این مقاله مراجعه کنند. برای پیاده‌سازی الگوریتم ژنتیک در مسئله مدیریت سفارش دهی ما از پروژه متن باز Jenetics که یک کتابخانه‌ای[[110]](#footnote-110) از پیاده‌سازی الگوریتم ژنتیک(GA) به زبان جاوا است استفاده کرده‌ایم ، این پروژه وابستگی بسیار کمی به دیگر کتابخانه‌های زبان جاوا دارد و تنها وابستگی آن به کتابخانه JScience است که همراه بسته دانلود شده وجود دارد، این پروژه از نشانی

https://sourceforge.net/projects/jenetics

به صورت رایگان قابل دانلود است. لازم به ذکر است که ما از نسخه 1.3.0—2013/06/14 استفاده کردیم.

پارامترهایی که مادر این مسئله در نظر گرفتیم به صورت زیر است

PopulationSize=1000

Mutator= 0.3

Evolve= 5000

SinglePointCrossover=0.5

نتایج حاصل از این روش در قسمت‏4.1.4 آمده است.

### پیاده‌سازی با یادگیری Q

یادگیریQ حرکت بهینه را در حالت فعلی بر مبنای تابع ارزش(رجوع کنید به ‏(5-1)) به دست می‌آورد، بعد از اجرای این حرکت و در یافت پاداش ، تابع ارزش بروز آوری می‌شود. اگر در هر مرحله ما فقط بهترین اقدام را انتخاب کنیم ،امکان این وجود د دارد که بعضی از حرکات در حالت‌ها اصلاً انتخاب نشوند. این اتفاق زیاد خوشایند نیست زیرا ما نمی‌توانیم مطمئن باشیم که استراتژی که الآن عامل انتخاب می‌کند بهترین استراتژی است. بنابراین ما باید بعضی از نقاط غیر بهینه را برای به دست آوردن اطلاعات راجع به محیط نیز جستجو کنیم. به هر حال اگر ما تعداد زیادی حرکت اکتشافی انجام دهیم .کارایی الگوریتم کاهش پیدا می‌کند این مسئله با نام exploration/exploitation در یادگیری Q مطرح می‌شود.سؤالی که مطرح است این است که آیا ما باید محیط را کاوش کنیم تا بیشتر راجع به محیط بدانیم یا از راه‌حلی که پیداکرده‌ایم بهره‌برداری کنیم تا همیشه از پاداش بهترین حرکت پیداشده تا آن زمان بهره‌مند شویم ؟

تعدادی را ه حل برای این مسئله در مقاله‌هایی که راجع به یادگیریQ آمده بیان‌شده ، که تقریباً در همه آن‌ها بر توازن بین exploration وexploitation تاکید شده. ما در این پایان‌نامه از تابعی استفاده می‌کنیم که در آغاز فرایند یادگیری بیشتر از exploration بهره ببرد و هرچه به پایان الگوریتم نزدیک می‌شویم از explorationکاسته شود و بیشتر وexploitation انجام شود، ما برای این کار از یک تابع که نموداری کسینوسی در بازه[0,] ربع اول دایره استفاده می‌کنیم، شمای این تابع در آورده شده است.

شکل 9 نمودار کاهش مقدار آلفا

در این روش ما یک عدد تصادفی بین صفر و یک در هر دور اجرای الگوریتم تولید می‌کنیم و اگر این عدد از آلفا کوچک‌تر بود، exploration انجام می‌دهیم در غیر این صورت وexploitation را دنبال می‌کنیم.به مرور زمان که **α** کوچک‌تر می‌شود مقدار explorationها کاسته و exploitation ها زیادتر می‌شوند.

در الگوریتم 6 روند یادگیری Q نمایش داده شده است ، تفاوت این الگوریتم با الگوریتمی که ما در پیاده‌سازی واقعی استفاده کرده‌ایم این است در این الگوریتم α همچنین برابر یک در نظر گرفته‌شده که ما در پیاده‌سازی واقعی را کمتر از یک در نظر گرفته‌ایم همچنین α یعنی ضریب یادگیری را با زمان اجرای الگوریتم متغیر در نظر گرفته‌ایم یعنی هرچه از زمان الگوریتم می‌گذرد ضریب یادگیری نیز کاهش می‌یابد، این نکته در اکثر مقالات مرتبط در نظر گرفته شده بود. نمودار کاهش مقدار αدر طول زمان همانند است

الگوریتم 6شبه کد یادگیری تقویتی استفاده‌شده در (Chaharsooghi, Heydari, & Zegordi, 2008)



چون ما این مسئله را به صورت متمرکز در نظر می‌گیریم بنابراین حالت همه سیستم را باید علاوه بر حالت عامل در نقطه تصمیم‌گیری در نظر بگیریم بنابراین جدول ارزش- اقدام ما دارای حالت و در هر حالت انتخاب وجود دارد.یعنی اگر سیستم در حالت[4,3,9] تصمیم[-3,2,-1] بگیرد به این معنی است که در حالت تصمیم‌گیری وضعیت انبار خرده‌فروش در کد4، عمده‌فروش در کد3 و کارخانه در کد 9 قرار دارد و در این حالت خرده‌فروش 3- واحد به درخواست مشتری اضافه می‌کند و به عمده‌فروش به عنوان سفارش این هفته ارسال می‌کند.و به همین ترتیب مقدار سفارش عمده‌فروش و کارخانه به بالاسری مشخص می‌شود.

### نتایج حل مساله به روش ژنتیک و الگوریتم Q

در این بخش ما نتایج حاصل از اجرای الگوریتم یادگیری Q در بازی نوشابه ارائه می‌کنیم .اجرای بازی برای هر سه قسمت مسئله به عامل‌های هوشمند سپرده شده است و تصمیم‌گیری در هر مرحله با توجه به وضعیت دیگر انبارها انجام می‌شود به عبارت دیگر مسئله به صورت کنترل متمرکز در نظر گرفته شده است. تقاضای مشتری در بازه زمانی 35 هفته بررسی می‌شود. این تقاضاها از (Chaharsooghi, Heydari, & Zegordi, 2008)گرفته شده است. که ما از آنها برای نشان دادن کارایی الگوریتم استفاده کرده‌ایم. هزینه هر عامل در طول هر هفته در در محیط‌های پویا همچون این مسئله در بلندمدت و با تغییر راهبرد سفارش دهی مشتری می‌تواند اثرات بدی بر هزینه کل سیستم داشته باشد در ادامه نمودار هزینه هر عامل در طول 35 هفته با اتخاذ راهبرد‌های متفاوت آورده شده است مشخص شده است.

شکل 10-نمودار هزینه زنجیره طی 35 هفته باراهبرد‌های سفارش دهی متفاوت

لازم به ذکر است که حالت اولیه عامل‌ها همان حالت اولیه بازی نوشابه بوده یعنی در ابتدا درون انبار هر یک از عامل‌ها 4 واحد کالا وجود دارد. در شکل 10 نمودار هزینه کل سیستم طی 35 هفته به تفکیک الگوریتم‌های مورد استفاده آورده شده است.

### معایب این دو روش

همان طور که از شکل 10 پیداست الگوریتم یادگیری Q با توجه به اتخاذ راهبرد پویا در هر هفته توانسته هزینه کمتری نسبت به دیگر راهبردها داشته باشد. و الگوریتم ژنتیک با در نظر گرفتن راهبرد ثابت سفارش دهی اگرچه هزینه کمتری نسبت به راهبرد پاس دادن سفارش دارد ولی مشکل اتخاذ راهبرد ثابت در محیط‌های پویا همچون این مسئله در بلندمدت و با تغییر راهبرد سفارش دهی مشتری می‌تواند اثرات بدی بر هزینه کل سیستم داشته باشد در ادامه نمودار هزینه هر عامل در طول 35 هفته با اتخاذ راهبرد‌های متفاوت آورده شده است.

شکل 11-نمودار هزینه خرده‌فروش در طول 35 هفته در بازی نوشابه

همان طور که از نمودار شکل 10 پیداست استفاده از یادگیری Q توانست بهبود نسبتاً خوبی را در مسئله بازی نوشابه نسبت به روش‌های فرا ابتکاری و پاس دادن سفارش انجام دهد ولی نکته‌ای که در این نمودارها به چشم نمی‌آید. فضای حالت نسبتاً بزرگی است که در الگوریتم یادگیری Q مورد نیاز است همان طور که در‏2.4 اشاره کردیم، ‏0برای نگه‌داری بهترین اقدام در هر حالت نیاز به یک جدول داریم که به تعداد حالت‌ها ردیف و به تعداد اقدام‌ها ستون داشته باشند. اگر ما مدیریت سفارش دهی در بازی نوشابه را به صورت متمرکز در نظر بگیریم یعنی هر عامل تلاش کند هزینه کل سیستم را کاهش دهد بنابراین نیاز است که حالت کنونی عامل‌های دیگر را نیز بداند و بر اساس آن اقدام بهینه را انتخاب کند.که این عمل باعث می‌شود فضای حالت بیشتری نیز نیاز باشد، در این مسئله کوچک که فقط سه عامل با تصمیمات محدود وجود داشت نیاز به فضای جدولی با 343000 خانه برای ذخیره‌سازی ارزش هر حالت-اقدام هستیم این مسایل در سیستم‌های چند عامله که در آن شبکه‌ای از زنجیره‌های تأمین با تعداد زیادی رده در هر شبکه وجود دارد باعث می‌شود که الگوریتم یادگیری Q با وجود کارایی بسیار خوب در سیستم‌های کوچک عملاً در سیستم‌های چند عامله ناکارآمد باشد.

شکل 12- نمودار هزینه انبار عمده‌فروش در طول 35 هفته در بازی نوشابه

شکل 13 نمودار هزینه کارخانه در طول 35 هفته در بازی نوشابه

نکته دیگری که در اینجا لازم است بیان کنیم حالت‌های پنهانی است که ما آنها را در نظر نمیگیریم.همانطور که در‏4 وجدول نشان داده شد ما مجموعه ای از حالتها را به یک حالت کد می‌کنیم که این امر باعث میشود بسیاری از حالتها که قالب یک حالت فشرده شوند.که این عمل که گسسته سازی فضای حالت وجود دارد باعث میشود که ما مقدار زیادی از حالت‌ها را نادیده بگیریم. در فصل بعد ما به معرفی روشی برای بهبود این مشکلات در یادگیری Q میپردازیم.

جدول 2- هزینه هر عامل در هر هفته توسط سه راهبرد متفاوت، پاس دادن سفارش، ژنتیک، یادگیری Q

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **هزینه انبار** | | یادگیری Q | | | ژنتیک | | | **پاس دادن سفارش** | | |
| **شماره هفته** | **تقاضا** | خرده‌فروش | عمده‌فروش | کارخانه | خرده‌فروش | عمده‌فروش | کارخانه | خرده‌فروش | عمده‌فروش | کارخانه |
| 1 | 15 | 7 | 4 | 8 | 7 | 4 | 8 | **7** | **4** | **8** |
| 2 | 10 | 24 | 17 | 10.5 | 24 | 15 | 11.5 | **24** | **15** | **12** |
| 3 | 8 | 41 | 31.5 | 22.5 | 41 | 31.5 | 21 | **41** | **32** | **19** |
| 4 | 14 | 69 | 50.5 | 45.5 | 71 | 52 | 44 | **72** | **53** | **40** |
| 5 | 9 | 98 | 81.5 | 67.5 | 103 | 82.5 | 64.5 | **104** | **84** | **57.5** |
| 6 | 3 | 130 | 113.5 | 100 | 137 | 113.5 | 94 | **139** | **116.5** | **84** |
| 7 | 13 | 155 | 134 | 127.5 | 168 | 134 | 121 | **172** | **139.5** | **108** |
| 8 | 2 | 169 | 156 | 144.5 | 190 | 157.5 | 139 | **197** | **166.5** | **123** |
| 9 | 13 | 182 | 166 | 164 | 216 | 171 | 161.5 | **227** | **184.5** | **142.5** |
| 10 | 11 | 187 | 174 | 175 | 238 | 185 | 177.5 | **254** | **204** | **155.5** |
| 11 | 3 | 190 | 179.5 | 187 | 253 | 203 | 195.5 | **275** | **228.5** | **170.5** |
| 12 | 4 | 194.5 | 187 | 197.5 | 268 | 215 | 222 | **297** | **248** | **194** |
| 13 | 6 | 198 | 196 | 215.5 | 275 | 222.5 | 238.5 | **312** | **264** | **207.5** |
| 14 | 11 | 199 | 202.5 | 225.5 | 290 | 229.5 | 252.5 | **336** | **278.5** | **220** |
| 15 | 15 | 204 | 209.5 | 231.5 | 308 | 236.5 | 264.5 | **362** | **292.5** | **228.5** |
| 16 | 12 | 208 | 219.5 | 246 | 327 | 248 | 280 | **390** | **314.5** | **241.5** |
| 17 | 15 | 215 | 228.5 | 265 | 354 | 265.5 | 306 | **429** | **344.5** | **264.5** |
| 18 | 4 | 215.5 | 239.5 | 279 | 380 | 292 | 336.5 | **468** | **384.5** | **292** |
| 19 | 12 | 216 | 247.5 | 296.5 | 411 | 313 | 371.5 | **513** | **420** | **324** |
| 20 | 3 | 218 | 259 | 316.5 | 433 | 332 | 399 | **550** | **454.5** | **348.5** |
| 21 | 13 | 220 | 264 | 329 | 452 | 339.5 | 423.5 | **585** | **478.5** | **370** |
| 22 | 10 | 222 | 275 | 342.5 | 468 | 347.5 | 443 | **618** | **502** | **386.5** |
| 23 | 15 | 224 | 282 | 358.5 | 485 | 354 | 462 | **651** | **526** | **402.5** |
| 24 | 15 | 229 | 290.5 | 372.5 | 510 | 365.5 | 487.5 | **695** | **557** | **425** |
| 25 | 3 | 231 | 299 | 387.5 | 525 | 383.5 | 515.5 | **730** | **595.5** | **450** |
| 26 | 11 | 233.5 | 307 | 397.5 | 547 | 395.5 | 552 | **773** | **629** | **483.5** |
| 27 | 1 | 236.5 | 321.5 | 416.5 | 556 | 406 | 577.5 | **804** | **662** | **506** |
| 28 | 13 | 239.5 | 332 | 427.5 | 567 | 414 | 601.5 | **838** | **683.5** | **527** |
| 29 | 10 | 240.5 | 349 | 440.5 | 582 | 427 | 617.5 | **867** | **703** | **540** |
| 30 | 10 | 242.5 | 361 | 450 | 594 | 435.5 | 633.5 | **891** | **723.5** | **553** |
| 31 | 0 | 250.5 | 375.5 | 467.5 | 596 | 445 | 657.5 | **912** | **747** | **574** |
| 32 | 0 | 262 | 391 | 484.5 | 600 | 450.5 | 679.5 | **922** | **764.5** | **593** |
| 33 | 8 | 270.5 | 413 | 507 | 600 | 463 | 696 | **939** | **775** | **606.5** |
| 34 | 0 | 280 | 435.5 | 526 | 600 | 481 | 710 | **943** | **782** | **620.5** |
| 35 | 14 | 287 | 455 | 540.5 | 606 | 497.5 | 715 | **951** | **783** | **621.5** |

## الگوریتم ارایه شده برای برطرف کردن معایب روشهای قبلی

### بهینه‌سازی الگوریتم HEDGER برای بازی نوشابه

برای تخمین زدن جدول حالت- عمل در بازی نوشابه ما تغییراتی در الگوریتم HEDGER لحاظ کرده‌ایم و آن را برای مسئله بازی نوشابه بهینه کرده‌ایم .در ابتدا ما برای سادگی بیشتر تصمیم‌گیری را از حالت متمرکز به حالت غیرمتمرکز تغییر داده‌ایم. در این حالت هر عامل دارای یک جدول به عنوان جدول حالت – عمل است. که ما 10 حالت را به عنوان داده‌های آموزشی برای تخمین زدن در ابتدا در نظر می‌گیریم.که این 10 حالت دقیقاً برابر موجودی انبار هر عامل است و مانند مسئله فصل قبل این حالت‌ها نماینده کد شده از مقدار موجودی نیستند به عبارت دیگر ما ~~جدول~~ را در این راه حل استفاده نمی‌کنیم.

علاوه بر این ما از پوسته مستقل از متغیر استفاده نمی‌کنیم و بجای آن برای موجودی انبار کران بالا و پایین در نظر می‌گیریم تا تخمین سفارش در خارج از بازه آموزشی نباشد هرگاه وضعیت انبار از مقدار ماکزیمم بیشتر شد ما سفارش Y-3 را به بالاسری ارسال می‌کنیم و هر گاه از مینیمم کمتر شد ما سفارش Y+3 را به بالاسری ارسال می‌کنیم.

ما برای تخمین زدن از همه نقطه‌ها استفاده می‌کنیم و با انتخاب مقدار مناسب برای در ‏3.6.1 بسیاری از نقاطی که در فاصله دورتری از نقطه Query هستند وزن صفر می‌گیرند تاثیر بر نتیجه تخمین ندارند.

در الگوریتم 5 بعد از تکرارهای فراوان در بازی نوشابه مشاهده شد خط 7و8 تاثیر چندانی بر تخمین زدن‌های بعدی ندارد و فقط بار محاسباتی برنامه را افزایش می‌داد.یکی از علت‌های آن گسسته بودن حالت‌های انبار که فقط می‌توانستند اعداد صحیح باشند بود، چون مار هر بار که نقطه جدید را تخمین می‌زدیم آن را به مجموعه داده‌های آموزشی اضافه می‌کردیم بنابراین نیاز نبود که نقاط اطراف آن را با مقادیر جدید به‌روز آوری کنیم.

برای محاسبه رگرسیون ما از کتابخانه رایگان Commons Math که توسط بنیاد Apache  ارائه‌شده استفاده کردیم نسخه استفاده‌شده این ابزار از آدرس <http://commons.apache.org/proper/commons-math/> قابل دانلود است.

بقیه مواردی که در الگوریتم 5 آمده است در پیاده‌سازی بازی نوشابه توسط عامل‌های هوشمند لحاظ شده است.در قسمت بعد نتایج حاصل از این الگوریتم آورده شده است.

## نتایج

برای پیاده‌سازی مسئله بازی نوشابه با استفاده از عامل‌ها هوشمند از JADE استفاده کرده‌ایم،JADE یک نرم‌افزار برای توسعه عامل‌هایی است که به زبان جاوا پیاده‌سازی شده‌اند، سیستم‌های مبتنی بر JADE هماهنگی بین عامل‌های FIPA را پشتیبانی می‌کنند و پیاده‌سازی استانداردی از زبان ارتباط بین عامل‌ها FIPA-ACL را فراهم کرده‌اند، این عمل باعث می‌شود ارتباط بین عامل‌ها راحت تر انجام شود. استفاده از JADE پیاده‌سازی سیستم‌های چند عامله را به خصوص در سخت‌افزارها و سیستم‌عامل‌های مختلف سهولت می‌بخشد، محل قرار گرفتن عامل‌ها می‌تواند در ماشین‌های مختلف و در مکان‌های مختلف باشد و کنترل عامل‌ها توسط رابط گرافیکی که برای این کار فراهم‌شده امکان‌پذیر است، عامل‌ها با استفاده از این سیستم می‌توانند از ماشینی به ماشین دیگر انتقال یابند.

استفاده از JADE به ما این امکان را می‌دهد که بدون در گیر شدن با جزییات سیستم‌های چند عامله بتوانیم بستر مناسب را برای حل مسئله به صورت توزیع‌شده فراهم آوریم

در این مسئله ما 3 عامل خودمختار در سه رده خرده‌فروش، عمده‌فروش و کارخانه ‌داریم که مسئولیت سفارش دهی به عهده خودشان واگذار شده است این عامل‌ها به دو صورت می‌توانند اقدام کنند یکی اینکه هنگام تصمیم‌گیری فقط وضعیت انبار خود را بررسی کنند و مقدار سفارش را بر اساس سودمندی خود، به رده‌بالاتر ارسال کنند ما به این حالت کنترل غیرمتمرکز میگوییم. حالت دیگر به این صورت است که عامل‌ها با در نظر گرفتن حالت انبار خود و دیگر عامل‌ها در سایر رده‌های زنجیره با در نظر گرفتن سودمندی اقدام برای کل زنجیره سفارش خود را به رده‌بالاتر ارسال می‌کنند این حالت کنترل متمرکز نامیده می‌شود.

شکل 14 هزینه کل زنحیره وقتی عاملها به صورت مستقل تصمیم میگیرند

درشکل 15و شکل 16 و شکل 17 هزینه انبار به تفکیک رده سفارش دهی آورده شده است.

شکل 15 نمودار هزینه خرده فروش بصورت کنترل غیر متمرکز

شکل 16 نمودار هزینه عمده فروش بصورت کنترل غیر متمرکز

برای سادگی بیشتر ونشان دادن کارایی عمومیت بخشیدن به یادگیری Q[[111]](#footnote-111) ما در این فصل از کنترل غیر متمرکز در هنگام تصمیم گیری استفاده میکنیم، به این معنی که هر عامل برای خودش یک جدولِ ارزش(تابعِ ارزش) حالت-اقدام دارد که فقط حالت ها ی انبار خود را نگه داری میکند.

علاوه بر این سه عامل هوشمند یک عامل دیگر هم برای ارسال سفارش و همچنین کنترل زمان اجرای برنامه وجود دارد.

نتایج استفاده از رگرسیون وزن دار محلی به عنوان تابع تخمین زن در مسئله مدیریت سفارش دهی در زنجیره تأمین در شکل 14 تا شکل 20 آمده است.

همان طور که از نمودارها پیداست استفاده از تخمین تابع به جای استفاده از جدول به خوبی توانسته مدیریت سفارش دهی چند عامله را در بازی نوشابه را بر عهده بگیرد. همان طور که از شکل 14 مشخص است هزینه مدیریت انبار در طی 35 هفته با استفاده از داده‌هایی که در (Chaharsooghi, Heydari, & Zegordi, 2008) و فصل قبل استفاده‌شده با تقریب بسیار خوبی به هزینه مدیریت انبار با استفاده از یادگیری Q سنتی نزدیک است.

شکل 17 نمودار هزینه کارخانه بصورت کنترل غیر متمرکز

در مدیریت سفارش دهی در زنجیره تأمین هدف ایده آل عامل‌ها نگه‌داشتن وضعیت انبار در حالت صفر یا به عبارت دیگر no Inventory است. در شکل 18 تا شکل 20نمودار وضعیت انبارها طی 35 هفته نشان داده شده است.

شکل 18 نمودار وضعیت انبار خرده فروش بصورت کنترل غیر متمرکز

شکل 19نمودرا وضعیت انبار عمده فروش بصورت کنترل غیر متمرکز

شکل 20 نمودار وضعیت انبار کارخانه بصورت کنترل غیر متمرکز

**فصل پنجم**

# 

**خلاصه و نتیجه گیری**

# نتیجه‌گیری

در این پایان‌نامه ما ابتدا به بررسی استفاده از عامل هوشمند مدیریت سفارش دهی زنجیره تأمین پرداختیم، و راهکارهای هوشمند سازی را توسط الگوریتم‌های فرا ابتکاری و یادگیری ماشین معرفی کردیم، در ادامه به با نمایش تفاوت‌های استفاده از الگوریتم ژنتیک و یادگیری Q در مسئله بازی نوشابه، برتری الگوریتم یادگیری Q را نسبت به الگوریتم ژنتیک در مدیریت سفارش دهی به صورت کنترل متمرکز نشان دادیم.این برتری ناشی از اتخاذ راهبرد تصمیم‌گیری پویا (برخلاف راهبرد تصمیم‌گیری ثابت در تمام دوره انجام بازی نوشابه) بود. پس از آن با توجه به ضعف الگوریتم یادگیری Q در فضای حالت بزرگ، سعی در پیدا کردن راهی برای حل این مشکل نمودیم، استفاده از یادگیری مبتنی بر نمونه برای تخمین زدن جدول حالت-اقدام راهکاری بود که در بسیاری از مراجع یادگیری تقویتی پیشنهاد شده بود و همچنین با توجه به پیچیدگی این الگوریتم بسیاری از راهکارهای تخمین زننده با شبکه‌های عصبی مصنوعی با شکست مواجه شده بود. بنابراین پیدا کردن یک تخمین زننده خوب نیازمند مطالعات فراوانی در این حوزه بود. در این پژوهش ما با تغییر دادن الگوریتم یادگیری HEDGER و مناسب ساختن آن با مسئله بازی نوشابه توانستیم یک تخمین زننده خوب که مبتنی بر رگرسیون وزن دار محلی است معرفی کنیم. برای کاهش پیچیدگی الگوریتم ما ابتدا مسئله را از حالت متمرکز به حالت محلی یا غیرمتمرکز تغییر دادیم و الگوریتم بهینه‌شده HEDGER را به عنوان الگوریتم یاد گیر بازی نوشابه استفاده کردیم، نتایج حاصله نشان از موفقیت این الگوریتم در تخمین تابع ارزش یادگیری Q داشت. که این امر راه را برای پژوهش‌های بعدی در این حوزه باز می‌کند.

## کارهای آینده

* در این پژوهش پارامترهای زیادی از قبیل ضریب یادگیری، فاکتور تنزیل ،دامنه تابع کرنل، تعداد همسایه‌ها برای تخمین زدن و... وجود دارد که با بهینه‌سازی این پارامترها می‌توان کارایی این روش را بهبود بخشید.
* بررسی مسئله کنترل موجودی به روش متمرکز و غیرمتمرکز با استفاده از این روش
* با توجه به اینکه تخمین تابع حالت- اقدام باعث می‌شود مشکل فضای حالت الگوریتم یادگیری Q برطرف شود ، بنابراین راه برای نشان دادن کارایی این الگوریتم در شبکه‌ای از زنجیره‌های تأمین با تعداد زیادی عامل در رده‌های مختلف باز شده است.
* با توجه به اینکه نرم‌افزار به خصوصی برای تخمین تابع به روش رگرسیون وزن دار محلی وجود ندارد، ارائه چنین نرم‌افزاری می‌تواند کمک زیادی به پژوهش گران نماید، در آینده قصد داریم چنین نرم‌افزاری را به صورت رایگان در اختیار علاقمندان قرار دهیم.

# منابع

Alev Taskin Gumus, A. F. (2009). A multi-echelon inventory management framework for stochastic and fuzzy supply chains. *Expert Systems with Applications 36, 36*, 5565–5575.

Andersson, J. &. (2001). A two-echelon inventory model with lost sales. *International Journal of Production Economics, 69(3)*, 307–315.

Atkeson, C. G. (1997). Locally weighted learning. . *AI Review, 11*, 11–73.

Axsater, S. (2003). Approximate optimization of a two-level distribution inventory system. *International Journal of Production Economics*, 545–553.

Baatz, E. (1995). CIO100-best practices: the chain gang. *CIO, Vol.8 No.19*, 46-52.

Chaharsooghi, S. K., Heydari, J., & Zegordi, S. H. (2008). A reinforcement learning model for supply chain ordering management: An application to the beer game. *Decision Support Systems 45*, 949-959.

Cook, R. D. (1979). Influential observations in linear regression. *Journal of the American Statistical Association, 74*, 169–174.

D. Mele, G. G. (2007). An agent-based approach for supply chain retrofitting under uncertainty. *Computers and Chemical Engineering 31*, 722–735.

Driessens, K., & Dˇzeroski, S. (2002a). (). Integrating experimentation and guidance in relational reinforcement learning. *Proceedings of the Nineteenth International Conference on Machine Learning* (pp. 115-122). Morgan Kaufmann Publishers, Inc.

Fazel Zarandi, S. V. (2013). A fuzzy reinforcement learning algorithm for inventory control in supply chains. *The International Journal of Advanced Manufacturing Technology (65)*, 557-569.

Ganeshan, R. (1999). Managing supply chain inventories: A multiple retailer, one warehouse, multiple supplier model. *International Journal of Production Economics, 59(1-3)*, 341–354.

Giannoccaro, I. P. (2003). A fuzzy echelon approach for inventory management in supply chains Research. *European Journal of Operational*, 149(1), 185–196.

Giannoccaro, I., & Pontrandolfo, P. (2002). Inventory management in supply chains: a reinforcement learning approach. *International of Journal Production Economics 78*, 153-161.

Gordon, G. J. (1995). Stable function approximation in dynamic programming. *Proceedings of the Twelfth International Conference on Machine Learning. .* San Francisco: Morgan Kaufmann.

Hanafizadeh, M. H. (2009). Designing fuzzy-genetic learner model based on multi-agent systems in supply chain management. *Expert Systems with Applications 36*, 10120–10134.

Handfield, R. B., Ernest, L., & Jr, N. (2004). Key issues in global supply base management. *Industrial Marketing Management*, 29–35.

Harland, C. (1996). Supply network strategies the case of health supplies. *European Journal of Purchasing & Supply Management*, 183-192.

Jiang, Z. S. (2009). Case-based reinforcement learning for dynamic inventory control in a multi-agent supply-chain system. *Expert Systems with Applications 36*, 6520–6526.

Jiao, X. Y. (2006). An agent-based framework for collaborative negotiation in the global manufacturing supply chain network. *Robotics and Computer-Integrated Manufacturing 22*, 239–255.

Lee, H. T., & Wu, J. C. (2006). A study on inventory replenishment policies in a two-echelon supply chain system. *Computers & Industrial Engineering 51*, 257–263.

Lee, H., & Whang, S. (1999). Decentralized multi-echelon supply chains: incentives and information. *Management Science, Vol. 45(5)*, 633– 640.

Liang, C.-C. H. (2006). Agent-based demand forecast in multi-echelon supply chain. *Decision Support Systems 42*, 390– 407.

Lin, C. C. (2011). Nonlinear system control using self evolving neural fuzzy inference networks with reinforcement evolutionary learning. *Appl Soft Comput J 11*, 5463-5476.

M.P. Baganha, M. C. (1998). The stabilizing effect of inventor in supply chainsy. *Operations Research 46 (3)* , 72-73.

Mahadevan, S., Marchalleck, N., Das, K. T., & Gosavi, A. (1997). Self-improving factory simulation using continuous-time average-reward reinforcement learning. *Proceedings of the 14th International Conference on Machine Learning*, 202–210.

New, S. J., & Payne, P. (1995). Research Frameworks in Logistics: Three Models, Seven Dinners and a Survey. *International Journal of Physical Distribution & Logistics Management*, 60-77.

Petrovic, D., Xie, Y., Burnham, K., Burnham, K., & Petrovic, R. (2008). Coordinated control of distribution supply chains in the presence of fuzzy customer demand. *European Journal of Operational Research, Vol. 185* , 146–158.

*Q-learning*. (n.d.). Retrieved from Wikipedia, the free encyclopedia: http://en.wikipedia.org/wiki/Q-learning

Rau, H. W.-Y.-M. (2003). Integrated inventory model for deteriorating items under a multi-echelon supply chain environment. *International Journal of Production Economics, 86(2)*, 155–168.

S.O. Kimbrough, D. W. (2002). Computers play the beer game: can artificial agents manage supply chains? *Decision Support Systems 33*, 323–333.

Sheu, J. (2005). A multi-layer demand-responsive logistics control methodology for alleviating the bullwhip effect of supply chains. *European Journal of Operational Research 161* , 797-811.

Singh, M. K. (1999). Finite-sample convergence rates for Q learning and indirect algorithms. *Advances in Neural Information Processing Systems, volume 11. MIT press*.

Smart, W. D. (2002). Making Reinforcement Learning Work on Real Robots. *Ph.D. thesis, Department of Computer Science, Brown University*.

Sterman, J. (1989). Modeling managerial behavior: misperceptions of feedback in a dynamic decision making experiment. *Management Science, Vol. 35 (3)*, 321-339.

Strozzi, F., Bosch, J., & Zaldívar, M. J. (2007). Beer game order policy optimization under changing customer demand. *Decision Support Systems, Vol. 42*, 2153-2163.

Sutton, R. B. (1998). *Reinforcement learning: an introduction.* Cambridge: MIT Press.

Thrun, S. &. (1993). Issues in using function approximation for reinforcement learning. *Proceedings of the Fourth Connectionist Models Summer School,Hillsdale, NJ: Lawrence Erlbaum.*

Vengerov. (2007). A reinforcement learning approach to dynamic resource allocatin. *Eng Appl Artif Intel 20*, 283-390.

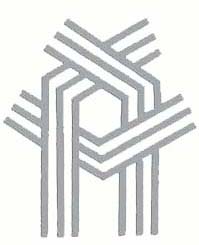
Watkins, C. (1989). *Learning from Delayed Rewards. PhD thesis.* Cambridge, England: Cambridge University.

*زنجیره\_مارکوف*. (n.d.). Retrieved from wikipedia The Free Encyclopedia: http://simple.wikipedia.org/wiki/Markov\_chain

گلپایگانی, ع. ص. (1389). *مدیریت زنجیره تامین مبتنی بر عامل.* تهران: ناقوس.

**Abstract**

In this study we introduce a framework to decide on ordering policies of supply chain members based on generalization of ­Reinforcement Learning (RL) in supply chain. In the real world, applications with very large state and action spaces and unknown state transition probability, classical RL algorithms usually show poor performance. Based on this investigation, the main goal of this study is to present a generalization of RL ordering management (GRLOM) to work well in real word Supply Chain Inventory Management problems. We show that our frame work, works well in a traditional learning setting. Using a GRLOM removes the need to visit every state-action pair, since we can generalize to pairs that we have not yet seen. The study presented in this work is fundamentally an empirical analysis, intended to show the feasibility of applying the GRLOM algorithm to reinforcement-learning benchmark tasks.



**University of Economic Sciences**

**Faculty of Management**

**M.S. Thesis**

**Title of the Thesis**

**Intelligent inventory management in supply chain based on multi agent systems**

**Supervisor:**

**Sayed AmirReza Abtahi, PhD**

**Advisor:**

**Kave Khalili Damghani, PhD**

**By:**

**AhmadReza Nazemi Sejzi**

**Date**

**March 2014**

1. QLearning- [↑](#footnote-ref-1)
2. - Autonomous [↑](#footnote-ref-2)
3. - ordering management [↑](#footnote-ref-3)
4. - Supply Chain [↑](#footnote-ref-4)
5. - inventory management [↑](#footnote-ref-5)
6. - intelligent agent [↑](#footnote-ref-6)
7. **-** Multi Agent System(MAS) [↑](#footnote-ref-7)
8. - Machine Learning [↑](#footnote-ref-8)
9. -کارخانه، عمده فروش، خرده فروش [↑](#footnote-ref-9)
10. - replenishment [↑](#footnote-ref-10)
11. - backlog [↑](#footnote-ref-11)
12. - inventory iolding [↑](#footnote-ref-12)
13. bull wipe Or forrester effect - [↑](#footnote-ref-13)
14. - Reinforcement Learning [↑](#footnote-ref-14)
15. exploration - [↑](#footnote-ref-15)
16. exploitation - [↑](#footnote-ref-16)
17. - state [↑](#footnote-ref-17)
18. - state space [↑](#footnote-ref-18)
19. - این جدول در واقع نگاشتی از حالت به عمل(Space🡪Action) را برای عامل در نظر می گیرد و در طول اجرای الگوریتم با محاسبه پاداش بلند مدت و کوتاه مدت به ازای انجام آن عمل ارزش آن را در جدول ذخیره می کند. به این روش یادگیری تقویتی مبتنی بر جدول گویند [↑](#footnote-ref-19)
20. - function approximation [↑](#footnote-ref-20)
21. cures of dimensionality - [↑](#footnote-ref-21)
22. abstraction - [↑](#footnote-ref-22)
23. Hierarchical Reinforcement Learning(HRL) - [↑](#footnote-ref-23)
24. - policy [↑](#footnote-ref-24)
25. - Value Function [↑](#footnote-ref-25)
26. - Artificial Neural Network [↑](#footnote-ref-26)
27. - State-Action [↑](#footnote-ref-27)
28. - Generalization [↑](#footnote-ref-28)
29. - علت این مساله این است که، برای اینکه الگوریتم نگاشت بهینه را پیدا کند لازم است که چندین بار در هر حالت قرار بگیرد(در این مساله به عنوان مثال چندین بار وضعیت انبار با مقدار کالای 10=X را مشاهده کند)تا بر اساس تجربه ای که کسب میکند مقدار مناسب سفارش دهی را پیداکند. عمده تمرکز این الگوریتم استفاده از تجربه برای یادگیری است. [↑](#footnote-ref-29)
30. pass order - [↑](#footnote-ref-30)
31. - multi echelon [↑](#footnote-ref-31)
32. Supply chain ordering management (SCOM)- [↑](#footnote-ref-32)
33. Centralized - [↑](#footnote-ref-33)
34. درمقابل تصمیم گیری های گروهی - [↑](#footnote-ref-34)
35. Fixed Rule Ordering - [↑](#footnote-ref-35)
36. Semi Markov Average Reward(SMART)- [↑](#footnote-ref-36)
37. beerGame (Sterman, 1989) یک شبیه سازی شناخته شده در زنجیره تامین است که بسیاری از علاقه مندان را همچون محققان دانشگاهی برای شبیه سازی سیستم های سفارش دهی به خود جذب کرده است. پارامترهای بهینه در سیاست سفارش دهی بازی نوشابه هنگامی که درخواست های مشتری افزایش می یابد در دو حالت متفاوت تجزیه وتحلیل می شود .(سیاست سفارش دهی ثابت برای همه عاملها وسیاست متفاوت). نشان داده می شود که کمترین هزینه برای زنجیره (در شرایط بازی نوشابه) زمانی بدست می آید که بازیکنان سیاست های متفاوتی داشته باشند (برخلاف سیاست یکسان برای همه بازیکنان) (Strozzi, Bosch, & Zaldívar, 2007)- [↑](#footnote-ref-37)
38. در‏2.4 خواهید دید که الگوریتم یادگیری تقویتی چگونه از تجربه فعلی برای سفارش دهی در آینده استفاده میکند.این روش به الگوریتم این امکان را میدهد که سیاست سفارش دهی پویا داشته باشد. - [↑](#footnote-ref-38)
39. Agent-Based **-**  [↑](#footnote-ref-39)
40. world - [↑](#footnote-ref-40)
41. مشاهده حالت انبار، دریافت تقاضای مشتری، تصمیم به سفارش دادن کالای جدید و مشخص کردن مقدار سفارش - [↑](#footnote-ref-41)
42. به این صورت که ما محیط را به صورت پیوسته در نظر نمیگیریم و هر گاه عامل وضعت انبار و در خواست مشتری را مشاهده میکند محیط را تغییر میدهد و بین دو زمان مشاهده انبار ودرخواست مشتری تغییر محیط در تصمیم گیری تاثیری ندارد- [↑](#footnote-ref-42)
43. Task - [↑](#footnote-ref-43)
44. Observation - [↑](#footnote-ref-44)
45. online - [↑](#footnote-ref-45)
46. بازه های زمانی احتمالی و از توزیع های شناخته شده آماری پیروی می کنند.- [↑](#footnote-ref-46)
47. علت این امر این است که ما از قانون X+Y با استفاده می کنیم.- [↑](#footnote-ref-47)
48. Lead-time - [↑](#footnote-ref-48)
49. Backorder - [↑](#footnote-ref-49)
50. Just In Time - [↑](#footnote-ref-50)
51. unsupervised - [↑](#footnote-ref-51)
52. Centralized Supply Chain Management [↑](#footnote-ref-52)
53. Material requirements planning (MRP) - [↑](#footnote-ref-53)
54. Manufacturing resource planning (MRP II)- [↑](#footnote-ref-54)
55. Enterprise resource planning (ERP)- [↑](#footnote-ref-55)
56. (Supply chain ordering management(SCOM - [↑](#footnote-ref-56)
57. integrated approach - [↑](#footnote-ref-57)
58. *upstream actor* - [↑](#footnote-ref-58)
59. lower slacks - [↑](#footnote-ref-59)
60. Policy - [↑](#footnote-ref-60)
61. - Agent [↑](#footnote-ref-61)
62. - Environment [↑](#footnote-ref-62)
63. Immediate reward - [↑](#footnote-ref-63)
64. - Observations [↑](#footnote-ref-64)
65. - Actions [↑](#footnote-ref-65)
66. fully observable - [↑](#footnote-ref-66)
67. در واقع مسیر رسیدن به هدف در این مساله برای خودمان هم نامشخص است. - [↑](#footnote-ref-67)
68. discount factor - [↑](#footnote-ref-68)
69. زنجیره مارکوف که به افتخار آندری مارکوف ریاضی‌دان اهل روسیه این‌گونه نام‌گذاری شده یک سیستم ریاضی است که در آن انتقال از یک حالت به حالت دیگر صورت می‌گیرد که البته تعداد این حالات قابل شمارش است. زنجیره مارکوف یک [فرایند تصادفی](http://fa.wikipedia.org/wiki/%D9%81%D8%B1%D8%A7%DB%8C%D9%86%D8%AF_%D8%AA%D8%B5%D8%A7%D8%AF%D9%81%DB%8C) بدون حافظه است بدین معنی که [توزیع احتمال شرطی](http://fa.wikipedia.org/wiki/%D8%AA%D9%88%D8%B2%DB%8C%D8%B9_%D8%A7%D8%AD%D8%AA%D9%85%D8%A7%D9%84_%D8%B4%D8%B1%D8%B7%DB%8C) حالت بعد تنها به حالت فعلی بستگی دارد و به وقایع قبل از آن وابسته نیست. این نوع بدون حافظه بودن [خاصیت مارکوف](http://fa.wikipedia.org/wiki/%D8%AE%D8%A7%D8%B5%DB%8C%D8%AA_%D9%85%D8%A7%D8%B1%DA%A9%D9%81) نام دارد- [↑](#footnote-ref-69)
70. Markov Decision Process - [↑](#footnote-ref-70)
71. MarkovProperty- [↑](#footnote-ref-71)
72. -Q Learning [↑](#footnote-ref-72)
73. initial value - [↑](#footnote-ref-73)
74. -long-term high reward - [↑](#footnote-ref-74)
75. Robot - [↑](#footnote-ref-75)
76. episode - [↑](#footnote-ref-76)
77. integrated approach - [↑](#footnote-ref-77)
78. lower slacks - [↑](#footnote-ref-78)
79. Semi-Markov(SMDPs) - [↑](#footnote-ref-79)
80. مقدار منفی بیانگر این نکته است که عامل نتوانسته در خواست قبلی مشتری را بطور کامل پاسخ دهد .- [↑](#footnote-ref-80)
81. Fixed Rule Ordering - [↑](#footnote-ref-81)
82. world - [↑](#footnote-ref-82)
83. مشاهده حالت انبار، دریافت تقاضای مشتری، تصمیم به سفارش دادن کالای جدید و مشخص کردن مقدار سفارش - [↑](#footnote-ref-83)
84. به این صورت که ما محیط را به صورت پیوسته در نظر نمیگیریم و هر گاه عامل وضعت انبار و در خواست مشتری را مشاهده میکند محیط را تغییر میدهد و بین دو زمان مشاهده انبار ودرخواست مشتری تغییر محیط در تصمیم گیری تاثیری ندارد- [↑](#footnote-ref-84)
85. task - [↑](#footnote-ref-85)
86. بازه های زمانی احتمالی و از توزیع های شناخته شده آماری پیروی می کنند.- [↑](#footnote-ref-86)
87. Lead-time - [↑](#footnote-ref-87)
88. Backorder - [↑](#footnote-ref-88)
89. supervise learning - [↑](#footnote-ref-89)
90. hidden state - [↑](#footnote-ref-90)
91. centralize - [↑](#footnote-ref-91)
92. - این مقدار با جستجو روی همه حالت های بعدی ممکن که از این حالت به آن میرسیم وانتخاب بیشترین مقدار ممکن بدست می آید [↑](#footnote-ref-92)
93. over-estimation - [↑](#footnote-ref-93)
94. hidden extrapolation - [↑](#footnote-ref-94)
95. independent variable hull - [↑](#footnote-ref-95)
96. hat matrix - [↑](#footnote-ref-96)
97. locally weighted regression - [↑](#footnote-ref-97)
98. case-based reasoning - [↑](#footnote-ref-98)
99. symbolic representation - [↑](#footnote-ref-99)
100. lazy - [↑](#footnote-ref-100)
101. kernel function - [↑](#footnote-ref-101)
102. bandWith- [↑](#footnote-ref-102)
103. Centralized - [↑](#footnote-ref-103)
104. Minkowski - [↑](#footnote-ref-104)
105. Manhattan - [↑](#footnote-ref-105)
106. - William D. Smart [wds@cs.brown.edu](mailto:wds@cs.brown.edu) Computer Science Department, Box 1910, Brown University, Providence, RI 02912, USA [↑](#footnote-ref-106)
107. smoother - [↑](#footnote-ref-107)
108. - البته محاسبه فاصله نقطه از همه داده های آموزشی کار زمانبری است که در (Smart, 2002)روشی را برای نگهداری یک درخت با دسترسی سریع به k نزدیکترین همسایه معرفی شده است، ما در حل مساله کنترل موجودی برای سادگی بیشتر از این روش استفاده نکرده ایم . [↑](#footnote-ref-108)
109. - با توجه به اینکه ممکن است نقاط بعدی که از این مرحله قابل دسترسی هستند هنوز مشاهده نشده باشند و همچنین جز داده های آموزشی نباشند.تخمین زدن آن نقاط و پیدا کردن بیشترین سودمندی در این حالت بسیار پیجیده است، ما از همسایه های مو جود در داده های آموزشی برای پیدا کردن بیشترین سوذمندی استفاده میکینم در (Smart, 2002)روشی با عنوان pairwise bisection معرفی شده که یک روش دو مرحله ای برای پیدا کردن نقاطی که احتمال وجود ماکزیمم در آنها وجود دارد به کار میرود. [↑](#footnote-ref-109)
110. library - [↑](#footnote-ref-110)
111. Q-Learning Generalization- [↑](#footnote-ref-111)