مجله علمی-پژوهشی «علوم و فناوری‌های پدافند غیرعامل»
سال سوم، شماره ۱، بهار ۱۳۹۱: ص ۷۱-۵۷

برنامه‌ریزی عملیات پویش و مین‌یابی برای چند روبات در محیط‌های ناشناخته

علی میردار هریجانی[۱]، الیس مسیحی*[۲]

۱- کارشناسی ارشد، ۲- استادیار، گروه مهندسی صنایع، دانشگاه تربیت مدرس

(دریافت: ۱۳۹۰/۰۶/۲۰، پذیرش: ۱۳۹۱/۰۳/۱۵)

چکیده

به موازات پیشرفت سخت‌افزاری روبات‌ها، علم برنامه‌ریزی حرکت آنها نیز روزبه روز پیچیده‌تر و ضروری‌تر شده و روبات‌ها را قادر می‌سازد قابلیت‌های خود را در کاربردهایی که حضور انسان‌ها در آن ناممکن یا خطرناک هستند ارتقا دهند. مثالی از چنین کاربردهایی، عملیات جستجو و جمع‌آوری مین‌های دفن شده در محیط‌های باز می‌باشد. مقاله حاضر به حل مسئله پویش و مین‌یابی در محیطی ناشناخته توسط چند روبات متحرک می‌پردازد که در آن روبات‌ها با یک‌دیگر تعاملی هم‌کارانه دارند. برای پویش محیط و یافتن اقلام پراکنده در آن، روش جدیدی به نام MSRT توسعه داده شده، و برای برنامه‌ریزی جهت دسترسی و جمع‌آوری اقلام پراکنده توسط روبات‌ها مدلی ریاضی شبیه به مسئله فروشنده دوره‌گرد چندگانه (mTSP) پیشنهاد شده است که با تلفیق آن با تکنیک‌هایی نظیر خوشه‌بندی k-means، الگوریتم جستجوی A* و گراف دیدنگار، مسیریابی برای هر روبات انجام می‌شود. مقایسه روش جدید با نتیجه‌های حاصل از بهینه‌سازی ریاضی مسئله نشان داد که روش پیشنهادی جواب‌های نزدیک به بهینه مطلق را در زمان‌های بسیار کوتاه‌تری تولید می‌کند.

کلید واژه‌ها: پویش، مین‌یابی، برنامه‌ریزی حرکت روبات‌های چندگانه، مسئله فروشنده دوره‌گرد، خوشه‌بندی.

Planning of Covering and Mine Sweeping Operations for Multiple Robots in Unknown Environments

A. Mirdar Harijani, E. Masihian*

Industrial Engineering Dept., Faculty of Engineering, Tarbiat Modares University

(Received: 09/11/2011, Accepted: 05/29/2012)

Abstract

In parallel with advances in robots hardware, the motion planning science has become increasingly indispensable and complicated, and enables robots to enhance their capabilities in applications where the presence of humans is impractical or hazardous. The covering and minesweeping operations of outdoor areas are examples of such applications. This paper deals with the problem of covering and minesweeping of unknown environments by multiple mobile robots which have cooperative interactions. For searching the space, the new method of multi sensor-based random trees (MSRT) is developed, and for planning the robots' paths toward the scattered items and their collection, a mathematical model similar to the multiple traveling salesman problem is proposed, which together with the k-means clustering, the A search, and the visibility graph techniques plans the route of each robot. Comparisons with the counterpart mathematic model solved to optimality showed that the proposed method produces suboptimal solutions in much shorter computational times.*

Keywords: Covering, Minesweeping, Multi Robot Motion Planning, TSP, Clustering

* Corresponding Author E-mail : masehian@modares.ac.ir

۱. مقدمه

هم گام با پیشرفت علم و فناوری، روبات‌هایی ساخته می‌شوند که نسبت به نمونه‌های قبلی از قابلیت‌های بیشتری جهت انجام طیف وسیعی از کارها برخوردار هستند. به کارگیری روبات‌ها در کاربردهایی که حضور انسان‌ها در آن ناممکن یا خطرناک هستند و یا انجام آنها مستلزم همکاری گروهی است اهمیت بیشتری پیدا کرده است. از جمله‌ی این کاربردها می‌توان به خاموش کردن آتش، یافتن مین، جستجو و نجات در محیط‌های خطرناک، جاسوسی و نقشه-برداری از مناطق دشمن، حمل یک شیء بزرگ، تمیز کردن و جمع-آوری ضایعات در محیط‌های خطرناک، کاوش در سیارات دیگر، مأموریت در نیروگاه‌های اتمی، کارخانجات صنعتی، معادن و محیط-هایی که از سلاح‌های شیمیایی و میکروبی استفاده شده اشاره کرد.

در ادبیات برنامه‌ریزی حرکت روبات، عمل جستجوی محیط و یافتن اقلام پراکنده در محیط (شامل ضایعات، اشیاء، مین‌ها، پوکه و فشنگ، بسته‌های غذا و غیره) به کنکاش[1] معروف است، که برای انجام آن یک یا چند روبات باید سطح را پویش[2] کنند تا اشیاء و اقلام اقلام پراکنده در محیط را پیدا کنند.

در مقاله حاضر روشی جدید برای حل مسئله پویش و کنکاش چند روباتی توسعه داده شده است که در آن چند روبات خودمختار با همکاری و هماهنگی با یک‌دیگر به کنکاش و جستجوی اقلام پراکنده در یک محیط باز می‌پردازند. اقلام پس از یافته‌شدن به روبات‌ها تخصیص داده شده و توسط آنها جمع آوری می‌شوند. محیط کاری برای روبات‌ها ناشناخته است و روبات‌ها بدون اطلاع از موقعیت اقلام موجود در محیط، شروع به جستجو کرده و با حرکت در محیط و شناسایی آن و تبادلات اطلاعات با روبات‌های دیگر، در صدد انجام مأموریت خود هستند. جهت شناسایی این اقلام فرض می‌شود که روبات‌ها مجهز به حسگرهایی[3] هستند که توانایی تشخیص آنها و نیز موانع موجود را دارند. روبات‌ها بایستی به گونه‌ای در محیط حرکت کنند که ترکیبی از معیارهایی مانند درصد پوشش محیط، ارتباط بین روبات‌ها، کنترل سطح به وسیله‌ی تمام روبات‌ها و پویش مناطق جدید به بهینه‌ترین وجه ممکن حاصل شود.

یک الگوریتم کاوش بر مبنای مرز[4] را برای مسئله پویش چند روباته[5] روباته[5] ارائه شده است[۱]. اکثر تکنیک‌های موجود در زمینه کاوش، از نوع کاوش بر مبنای مرز هستند. منطق این رویکرد عبارت است از اینکه روبات بایستی به سمت مرز بین منطقه کاوش شده‌ی امن و قلمروی کاوش نشده حرکت کند تا اطلاعات ناشی از ادراکات جدید را به حداکثر برساند [۲].

با استفاده از الگوریتم ساخت درخت پوشا[6]، حل مسئله پوشش غیربه‌هنگام[7] محیط کاری انجام شده است[۳]. در این مقاله فرض شده که روبات‌ها نسبت به محیط کاری از دانش قبلی برخوردارند؛ یعنی به نقشه کامل محیط کار مجهزند و محل موانع را می‌دانند. جهت ایجاد پوشش، هر روبات از کپی مسیر پوشش درخت پوشا و مکان اولیه‌اش روی مسیر استفاده می‌کند تا یک زیر مسیر تخصیص داده شده به آن را به‌وجود آورد.

در یکی از الگوریتم‌های ارائه شده، دو روبات در یک محیط ناشناخته و با شروع از یک نقطه‌ی نامشخص، هم‌دیگر را در محیط ملاقات می-کنند[۴]. چنین مسئله‌ای، *مسئله قرار ملاقات*[8] روبات نامیده می‌شود. شود. دو روبات ناهمگن مجهز شده به دو حسگرند که یک جهت نگاشت و دیگری برای محاسبه موقعیت در محیط به کار می‌روند.

محققان با استفاده از برنامه‌ریزی خطی عدد صحیح مختلط به حل مسئله پوشش چند روبات در محیط ناشناخته پرداخته‌اند [۵]. فرض شده است که روبات‌ها از قابلیت ارتباط بی‌سیم برخوردارند و در صورتی می‌توانند با هم ارتباط داشته باشند که در دامنه‌ی ارتباطی هم دیگر باشند. معیارهای بهینه‌سازی عبارتند از: پوشش تمام مناطق خاص، فراهم‌کردن امکان ارتباط بین روبات‌ها تا حد ممکن، کنترل حداکثر سطح به وسیله تمام روبات‌ها، و کاوش مناطق جدید. هدف مسئله نیز عبارت است از بیشینه‌سازی این معیارها، در حالی که محدودیت‌هایی از قبیل اجتناب از برخورد، سرعت ممکنه برای هر روبات، ارتباط بین روبات‌ها و حس‌کردن موانع و مناطق خاص زمانی که در دامنه حسگر روبات هستند را ارضا کند [۵].

تعمیمی از مسئله مین‌یابی توسط روبات‌ها به مسئله رسیدن چند هواپیمای بدون سرنشین به نقاط هدف مورد بررسی قرارگرفته است [۶]. مکان بعضی از اهداف از پیش معلوم بوده و بعضی دیگر در حین مانور هواپیماها شناسایی می‌شوند. ماموریت هواپیماهای بدون سرنشین این است که تمامی اهداف را کشف کنند، به گونه‌ای که احتمال به خطر افتادن هر یک از هواپیماها کمینه شود. احتمال به خطر افتادن، با بیشینه‌کردن فاصله هواپیماها تا نقاط تهدید (دشمن) و نیز با تعقیب هم زمان اهداف با چند هواپیما کم می‌شود. در این مقاله، هر هواپیما بر مبنای خوانش رادار نقشه‌ای از مکان‌های اهداف را ایجاد و با کمک دیاگرام وُرونویی[9] مسیر خود را برنامه‌ریزی می‌کند. نتیجه اینکه کل فضای در دسترس توسط هواپیماها پویش و شناسایی شده است.

با استفاده از روش تجزیه ذوزنقه‌ای و استفاده از گراف هم‌جواری و با حرکات رفت و برگشت روبات در هر سلول به کار پوشش محیط‌های غیربه‌هنگام پرداخته شده است [۷]. متأسفانه تجزیه‌ی ذوزنقه‌ای مسیر کارایی جهت پوشش ایجاد نمی‌کند چرا که به‌طورمعمول سلول‌های

[1] Foraging
[2] Sweeping
[3] Sensor
[4] Frontier-Based Exploration Algorithm
[5] Multi-Robot Exploration
[6] Spanning Tree
[7] Offline Coverage
[8] Rendezvous Problem
[9] Voronoi Diagram

باریک و طویلی ایجاد می‌کند. البته می‌توان با ادغام سلول‌های تجزیه‌ی ذوزنقه‌ای مسیر پوشش کاراتری را به‌وجود آورد، مانند روش تجزیه‌ی سلولی Boustrophedon که برای پوشش محیط‌های غیربه‌هنگام ارائه شده است [۸]. پوشش محیط‌های به‌هنگام[1] توسط روش تجزیه‌ی سلولی Boustrophedon نیز ارائه شده‌اند [۹].

از جمله موضوعات دیگری که می‌توان آن را به عنوان مسئله پویش و پوشش محیط قلمداد کرد، کار مشاهده‌ی چند هدف متحرک است که در گونه‌ای از آن، محققان با مدل‌سازی ریاضی مسئله و بیشینه-سازی آن به حل مسئله پرداخته‌اند [۱۰]. محققان دیگری نیز از الگوریتمی بر مبنای نمونه‌گیری برای بیشینه‌سازی کوتاه‌ترین مسیر برای فرار استفاده کرده‌اند[۱۱]. از جمله روش‌های دیگر ارائه شده بر مبنای نمونه‌گیری تصادفی می‌توان به روش درخت تصادفی جستجوی سریع (RRT[2]) اشاره کرد که با گسترش یک درخت، کل فضای پیکربندی مورد پوشش قرار می‌گیرد [۱۲].

روش SRT[3]، که می‌توان آن را تغییر یافته‌ی روش RRT نیز دانست، یک الگوریتم مبتنی بر نمونه‌گیری است که به ویژه برای پوشش محیط‌های به‌هنگام کارایی دارد. رویکرد SRT نوعی برنامه-ریزی حرکت احتمالی است که در آن نقشه‌ی مسیرِ فضای پیکربندی آزاد به‌تدریج با تولید پیکربندی‌های تصادفی روبات در داخل ناحیه‌ی ایمنِ محلیِ شناسایی شده، توسط حسگرها ساخته می‌شود [۱۳]. SRT همانند RRT دارای ساختار درختی است که نقشه‌ی مسیر فضای پیکربندی آزاد روبات را ارائه می‌کند. هر گره SRT شامل یک پیکربندی (موقعیت) عاری از برخورد q است که روبات به آن رسیده است. این گره توصیفی از ناحیه‌ی ایمن محلی S اطراف q را به همراه دارد که از طریق سیستم حسگر دریافت می‌شود. درخت به صورت افزایشی و با گسترش ساختار به سمتی که به صورت تصادفی انتخاب شده گسترش می‌یابد، به گونه‌ای که پیکربندی جدید و مسیر رسیدن به این موقعیت جدید در داخل ناحیه‌ی ایمن محلی[4] (LSR) باشد. روش SRT برای روبات‌های غیرهولونومیک نیز به کار رفته است [۱۴]. ترکیبی از روش SRT و استراتژی بر مبنای مرز[5] نیز برای برای حل مسئله پوشش به کار رفته است [۱۵].

در مقاله‌ی حاضر برای پویش روبات‌ها در محیط و یافتن اقلام پراکنده در محیط، روش MSRT[6] که مبتنی بر SRT می‌باشد توسعه توسعه داده شده است. پس از یافتن اقلام، با توسعه‌ی مدلی شبیه به مسئله فروشنده دوره‌گرد چندگانه، به حل مسئله جمع‌آوری اقلام پرداخته شده است. از طریق الگوریتم خوشه‌بندی k-means، کل اقلام شناسایی شده در محیط به تعداد روبات‌ها خوشه‌بندی شده و مسئله فروشنده دوره‌گرد چندگانه به چند مسئله فروشنده دوره‌گرد محلی تبدیل می‌شود. با حل مسئله فروشنده دوره‌گرد توسط الگوریتم جستجوی *A، ترتیب جمع‌آوری اقلام برای هر روبات به-دست آمده و در نهایت اقلام اختصاص یافته به هر روبات توسط روبات مربوطه جمع‌آوری می‌شود که جهت انجام این کار مسیر حرکت هر روبات از روی گراف دیدنگار به‌دست می‌آید.

سازمان‌دهی مقاله بدین صورت است: در بخش ۲ به معرفی مفروضات مسئله و در بخش ۳ به تشریح چگونگی پویش روبات‌ها در محیط و جزئیات روش MSRT پرداخته شده است. تشریح چگونگی یافتن ترتیب جمع‌آوری اقلام با استفاده از مسئله فروشنده دوره‌گرد چندگانه و حل این مسئله توسط الگوریتم خوشه‌بندی k-means و الگوریتم جستجوی *A در بخش ۴ آمده است و در بخش ۵ نتیجه‌های شبیه‌سازی و مقایسه‌های عددی با حل دقیق مسئله ارائه شده‌اند. نتیجه‌گیری و تحقیقات آینده نیز در بخش پایانی تشریح شده‌اند.

۲. مفروضات مسئله

سیستم چند روباتی که در این مقاله مطرح شده است دارای ویژگی‌های زیر است:

- سیستم، خودمختارِ توزیع شده است؛ یعنی هر روبات مستقلاً تصمیم می‌گیرد و در محیط به صورت خودمختار عمل می‌کند.
- روبات‌ها صلب و دیسک شکل بوده و فضای کاری آنها ۲ بُعدی است و در راستای محورهای مختصات X و Y حرکت می‌کنند.
- موانع در محیط کاری ایستا بوده و تغییر شکل و مکان نمی‌دهند.
- روبات‌ها هولونومیک هستند، بدین معنی که تعداد درجات آزادی قابل کنترل آنها از تعداد کل درجات آزادی آنها کم‌تر نیست و در نتیجه می‌توانند در جهات مختلف مستقیماً حرکت کنند.
- روبات‌ها هیچ دانش قبلی از محیط نداشته و محیط برای روبات‌ها ناشناخته است. به عبارت دیگر، مسئله مورد تحقیق، یک مسئله برنامه‌ریزی حرکت به‌هنگام[7] است.
- روبات‌ها مجهز به حسگرهای فاصله یاب هستند که در هر پیکربندی q، فضای آزاد اطرافشان را (شامل موانع موجود در محیط مانند دیوار، درخت، صخره و غیره) تا شعاع مشخصی شناسایی می‌کنند. فضای اطراف عاری از مانع، ناحیه ایمن محلی (LSR) نامیده شده و با S نشان داده می‌شود. همچنین، روبات‌ها دارای حسگرهای مین یاب نیز هستند که آنها را قادر به تشخیص اقلام پراکنده مانند مین‌های مستتر در محیط می‌نمایند.
- روبات‌ها تمامی اقلام اختصاص یافته به خود را یک‌جا جمع‌آوری و در قرارگاه (دپو) تخلیه می‌کنند.

[1] Online
[2] Rapidly-Exploring Random Tree (RRT)
[3] Sensor-Based Random Tree (SRT)
[4] Local Safe Region (LSR)
[5] Frontier-Based
[6] Multi Sensor-Based Random Tree (MSRT)
[7] Online Motion Planning

روبات‌ها از ظرفیت‌های حمل یکسانی برخوردار بوده و می‌توانند به چند محل سر بزنند و اقلام را جمع‌آوری کنند. آنها به محض پر شدن ظرفیت به قرارگاه (دپو) برگشته و اقلام را تخلیه می‌کنند.

۳. پویش روبات‌ها در محیط

هنگامی که چند روبات مأمور به انجام کاری در یک محیط مشترک می‌شوند، بحث هماهنگی و همکاری بین آنها اهمیت به سزایی پیدا می‌کند. در چنین شرایطی روبات‌ها بایستی از برخورد با یکدیگر اجتناب کرده و در کار یکدیگر تداخل ایجاد نکنند. بنابراین روبات‌ها بایستی با یکدیگر در ارتباط باشند و اطلاعات دریافتی خود را با همدیگر به اشتراک بگذارند.

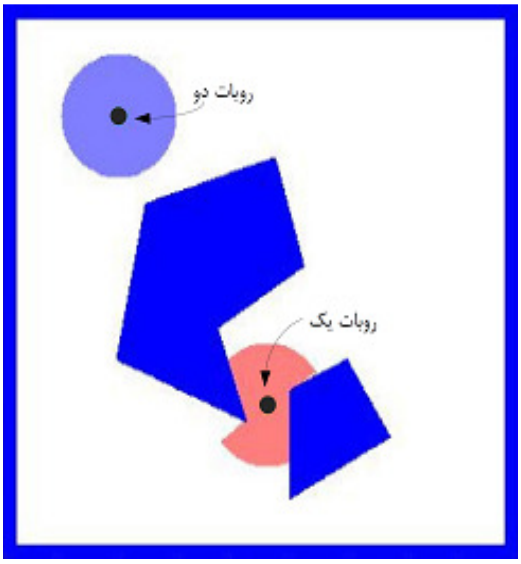
برای تبادل اطلاعات بین روبات‌ها فرض شده که آنها به سیستم موقعیت‌یاب جهانی[1] مجهز هستند و قادرند مختصات مکانی خود را تعیین کرده و از طریق یک سیستم اِترنت[2] رادیویی (بی‌سیم) با دیگر روبات‌ها ارتباط برقرار کرده و تبادل اطلاعات نمایند. این امر باعث افزایش سطح اطلاعاتی روبات‌ها شده و در نتیجه امکان هماهنگی و همکاری بین آنها نیز بیشتر می‌شود.

همان‌طور که قبلاً اشاره شد، مسئله مورد بررسی عبارت است از برنامه‌ریزی پویش روبات‌ها در یک محیط ناشناخته جهت جمع‌آوری مین‌های پراکنده در آن. بدین منظور، روشی جدید بنام MSRT برای پویش کامل سطح توسط چند روبات توسعه داده شده است که چند روبات را قادر می‌سازد از طریق هماهنگی و همکاری با یکدیگر، به طور سیستماتیک به کاوش در محیط بپردازند.

۳-۱. روش پویش چند روباتی MSRT

این روش چند روبات را قادر می‌سازد تا به طور سیستماتیک به پویش در یک محیط ناشناخته بپردازند. در الگوریتم جدید، برنامه-ریزی حرکت هر روبات منفرد مطابق با روش SRT (که در بخش ۱ معرفی شد [۱۳])، انجام می‌گیرد که با اشتراک گذاشتن اطلاعات اکتسابی روبات‌های دیگر، هر روبات تنها بخشی از محیط را پویش می‌کند، به طوری که اشتراک فضاهای پویش شده توسط روبات‌ها برابر با تهی بوده و اجتماع این فضاها برابر با کل محیط می‌باشد.

جهت محاسبه‌ی ناحیه‌ی ایمن محلی، هر روبات با فرستادن امواجی با طول موج مشخص از حسگرهای فاصله یاب خود از محیط اطرافش اطلاعات کسب می‌کند. اگر مانعی در دامنه‌ی مؤثر این امواج قرار گیرد موج بازتابیده شده و جسم شناسایی می‌شود. در غیر این صورت، روبات به اندازه‌ی دامنه‌ی مؤثر قابل اندازه‌گیری از محیط اطرافش اطلاع کسب می‌کند. نمونه‌ای از محیط کاری روبات شامل موانع چندضلعی در شکل (۱)آورده شده است که در آن روبات‌ها به شکل دیسک‌های کوچک نشان داده شده‌اند. همچنین، نواحی ایمن محلی هر روبات که توسط حسگرها محاسبه شده‌اند نیز در اطراف هر روبات رسم شده‌اند.



شکل ۱. نواحی ایمن محلی برای دو موقعیت مختلف روبات

در شکل (۱) ناحیه‌ی ایمن محلی روبات ۱ تحت تأثیر موانع به شکل دایره کامل نبوده و در دو منطقه با موانع فصل مشترک پیدا کرده است، در حالی که برای روبات ۲ هیچ مانعی در دامنه‌ی حسگر روبات قرار نداشته و در نتیجه شکل ناحیه‌ی ایمن محلی به صورت یک قرص کامل با شعاع دامنه‌ی مؤثر ایجاد شده است.

به طور کلی، گام‌های الگوریتم MSRT در هر تکرار آن برای هر روبات به شرح زیر می‌باشد، که شبه کد کلی آن در شکل ۲ آمده است:

۱. کسب اطلاعات از محیط اطراف نقطه‌ی فعلی q_{curr} روبات (ناحیه S) توسط تابع PERCEPTION(q_{curr}) و ذخیره‌ی آن در ماتریس درخت تصادفی τ_j توسط تابع ADD(τ_j, (q_{curr}, S)).

۲. تبادل اطلاعات کسب شده با روبات‌های دیگر توسط تابع SHARE(all τ_j).

۳. انتخاب جهت تصادفی θ_{rand} برای پویش توسط تابع RANDOM_DIR و محاسبه‌ی گام حرکتی $\alpha \cdot r$ در جهت θ_{rand} توسط تابع RAY و انتخاب نقطه کاندید جدید q_{cand}.

۴. ارزیابی q_{cand} توسط توابع بولی VALID1 و VALID2.

[1] Global Positioning System (GPS)
[2] Ethernet

```
BUILD_MSRT(q_init, K_max, I_max, α, β, n, range, d_min)
    q_curr = q_init
    For k =1 to K_max
        j ← 1
        Loop
            S ← PERCEPTION(q_curr)
            ADD(τ_j, (q_curr, S))
            j = j + 1
        Until (j = number of robots)
        τ ← SHARE(all τ_j)
        j ← 1
        Loop
            i ← 0
            Loop
                θ_rand ← RANDOM_DIR
                τ_j ← RAY(S, θ_rand)
                q_cand ← DISPLACE(q_curr, θ_rand, α×r)
                i ← i + 1
                VALID1 (q_cand, d_min, other LSR, τ_j)
                VALID2 (q_cand, β, range, τ − τ_j)
                VALID = VALID1 ∩ VALID2
            Until (VALID = 1 OR i = I_max)
            If VALID = 1
                MOVE_TO(q_cand)
                q_curr ← q_cand
            Else
                MOVE_TO(q_curr.parent)
                q_curr ← q_curr.parent
        Until (j = number of robots)
    Return τ
```

شکل ۲. شبه کد روش MSRT

```
PERCEPTION(q_curr, q, sensors, range)
    For t =1 to sensors
        Y← RAY_EQU(t)
        For ob =1 to max(q)
            [XI, YI] = INTSEC_P_L(q(ob),Y)
            ADD(Rays, [XI, YI])
        END
        [mindis, closestpoint] ←
                FIND_MIN_DIS(Rays)
        ADD(vis, [t, mindis, closestpoint])
    END
    PERCEPTION ← REVISE(vis)
    Return PERCEPTION
```

شکل ۳. شبه کد تابع PERCEPTION

در گام اول در هر تکرار k از الگوریتم، یک فرآیند ادراک توسط تابع PERCEPTION(q_{curr}) برای هر روبات j ($j = 1, \ldots, n$) رخ می‌دهد؛ به این معنی که اطلاعات حسگرها پردازش‌شده و فضای آزاد اطراف روبات در پیکربندی فعلی (q_{curr}) که ناحیه‌ی S نامیده می‌شود به دست می‌آید. حال یک گره جدید شامل پیکربندی q_{curr} روبات و ناحیه‌ی ایمن محلی S مربوطه به درخت τ_j اضافه می‌شود. در واقع، هر روبات j درختی تصادفی ایجاد می‌کند که گره‌های آن نقاطی هستند که روبات در هر مرحله از الگوریتم در آنجا قرار می‌گیرد. به طور دقیق‌تر، گام‌های الگوریتم فرآیند ادراکی در هر تکرار به شرح زیر می‌باشد، که شبه کد آن در شکل ۳ نشان داده شده است:

۱. محاسبه‌ی معادله‌ی خط پرتو ساطع شده از حسگر t-ام توسط تابع RAY_EQU[1] (sensors تعداد کل حسگرهای روبات است).

۲. محاسبه‌ی نقطه‌ی تلاقی معادله خط فوق با تمامی موانع موجود در محیط (q) توسط تابع INTSEC_P_L[2] و ذخیره‌ی نقاط تلاقی در ماتریس Rays.

۳. به‌دست آوردن فاصله و مختصات نزدیک‌ترین نقطه‌ی برخورد توسط تابع FIND_MIN_DIS و ذخیره آنها به همراه شماره‌ی حسگر t در ماتریس vis.

۴. اصلاح محدوده‌ی ادراکی روبات (ماتریس vis) توسط تابع REVISE با توجه به حداکثر دامنه‌ی قابل اندازه‌گیری توسط حسگرهای روبات (Range).

در گام دوم، پس از اینکه درخت τ_j روبات j به روز رسانی شد، تابع SHARE اطلاعات دریافتی این روبات، از قبیل مختصات نقاط کاوش شده و سایر اطلاعات دریافتی از محیط را در اختیار روبات‌های دیگر قرار می‌دهد. این کار جهت تضمین عدم برخورد روبات‌ها و عدم دخالت آنها در کار یکدیگر و ایجاد هماهنگی کامل بین روبات‌ها انجام می‌گیرد.

در گام سوم بایستی یک نقطه کاندید (q_{cand}) برای تعیین موقعیت جدید روبات تعیین شود. لذا یک جهت کاوش به صورت تصادفی (θ_{rand}) توسط تابع RANDOM_DIR برای هر روبات تولید می‌شود و تابع RAY فرا خوانده می‌شود که این تابع شعاع r از S را در جهت θ_{rand} محاسبه می‌کند که برابر است با مینیمم بین فاصله‌ی روبات از نزدیک‌ترین مانع و حداکثر دامنه‌ی قابل اندازه‌گیری توسط حسگرهای روبات در جهت θ_{rand}. به عبارت دیگر r برابر است با طول دامنه حرکتی روبات در جهت θ_{rand}، که اگر مانعی در این جهت و در محدوده‌ی حداکثر دامنه‌ی قابل اندازه‌گیری توسط حسگرها وجود نداشته باشد، طول دامنه حرکتی برابر با حداکثر دامنه‌ی قابل اندازه‌گیری توسط حسگرها است، در غیر این صورت طول دامنه حرکتی برابر با فاصله‌ی بین روبات و مرز مانع در جهت θ_{rand} است.

[1] Ray Equation
[2] Intersection of Polygon and Line

جهت حفظ فاصله‌ی ایمن از موانع، نقطه‌ی جدید کاندید q_{cand} به فاصله $\alpha \times r$ از q_{curr} و در جهت θ_{rand} انتخاب می‌شود، که در آن $\alpha < 1$ است (در این مقاله $\alpha = 0.8$ در نظر گرفته شده است). چند مثال از پیکربندی‌های کاندید قابل‌قبول در شکل (۴- الف و ب) آورده شده است. شکل (۴- ج) جزئیات ایجاد نقطه‌ی کاندید برای (۴- الف) را نشان می‌دهد.

در گام چهارم، پس از اینکه q_{cand} هر روبات به صورت تصادفی در ناحیه‌ی ایمن S تولید شد، آن توسط توابع بولی VALID1 و VALID2 مورد ارزیابی قرار می‌گیرد. تابع بولی VALID1 نقاط کاندید را از دو لحاظ مورد ارزیابی قرار می‌دهد: اول اینکه نقطه‌ی کاندید جدید بایستی در فاصله‌ای بزرگ‌تر از d_{min} نسبت به q_{curr} قرار گیرد، و دوم اینکه نقطه‌ی کاندید جدید در داخل ناحیه‌ی ایمن محلی (LSR) هر گره دیگر متعلق به τ قرار نگیرد. تابع بولی VALID2 در صورتی نقطه‌ی کاندید جدید را قابل‌قبول ارزیابی می‌کند که فاصله‌ی این نقطه از نقاطی که روبات‌های دیگر در آن قدم گذاشته‌اند کمتر از $\beta \times range$ باشد، که $\beta < 1$ بوده و range همان حداکثر دامنه‌ی قابل اندازه‌گیری توسط حسگرها است. بنابراین VALID2 مانع از آن خواهد شد تا روبات‌ها وارد منطقه‌ی کاوش‌شده‌ی روبات دیگر شوند، و به عبارتی منطقه‌ی کاوش شده توسط روبات‌های دیگر را مانعی غیر قابل عبور برای روبات تلقی می‌کند.

در صورتی که هر دو تابع بولی VALID1 و VALID2 نقطه‌ی کاندید را قابل‌قبول ارزیابی کنند آنگاه روبات به پیکربندی q_{cand} حرکت کرده و این چرخه تکرار می‌شود. در غیر این‌صورت، الگوریتم یک پیکربندی تصادفی دیگر از q_{curr} ایجاد می‌کند تا زمانی که یا یک پیکربندی قابل قبول به دست آید یا تعداد دفعات کوشش جهت یافتن یک پیکربندی جدید برابر با I_{max} شود. اگر پس از I_{max} بار تکرار، الگوریتم موفق به یافتن یک پیکربندی جدید معتبر برای روباتی نشود، آن روبات در امتداد آخرین یال عبوری خود یک عمل برگشت به عقب انجام داده و به گره والدِ q_{curr} (گره ای که q_{curr} از آن به وجود آمد) برمی‌گردد و در تکرار بعد، چرخه‌ی کاوش را از این گرهِ والد آغاز می‌کند.

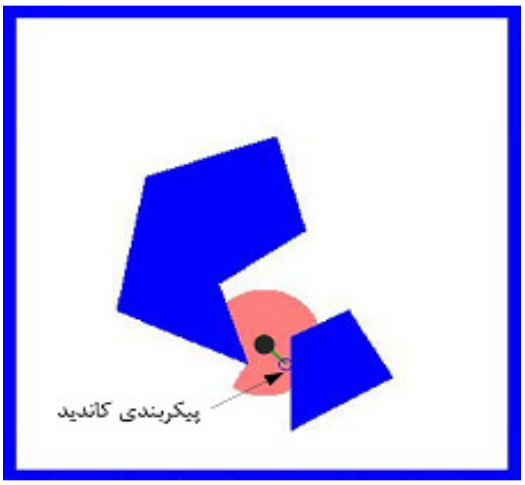
به عنوان مثال در شکل(۵) روبات از نقطه‌ی s شروع به حرکت کرده و پس از رسیدن به گره ۱۴، دیگر نمی‌تواند پیکربندی کاندید قابل-قبولی را در تعداد تکرار لازم پیدا کند. بنابراین عمل برگشت انجام داده و به گره والد گره ۱۴ (گره ۱۳) برگشته که در این گره هم نمی‌تواند در تعداد تکرار لازم پیکربندی کاندید قابل‌قبولی ایجاد کند. در نتیجه به گره ۱۲ برگشته و در این گره پیکربندی کاندید قابل-قبول ۱۵ را ایجاد می‌کند.

در صورتی که در تکرارهای بعدی نیز یک پیکربندی جدید معتبر برای آن روبات پیدا نشود (این حالت به ویژه هنگامی اتفاق می‌افتد که حرکت روبات توسط موانع یا نقاط کاوش شده توسط روبات‌های دیگر مسدود شود)، آن روبات طی یک برگشت به عقب‌های متوالی به گره شروع اولیه خود برمی‌گردد. لازم به ذکر است که روبات‌ها مستقل از یکدیگر عمل می‌کنند و ممکن است برخی از آنها در یک تکرار مشخص موفق به یافتن پیکربندی کاندید جدید شوند. شکل (۶) نمونه‌هایی از پویش محیط توسط دو، سه، چهار، و پنج روبات را نشان می‌دهد. آنچه تشریح شد مختصری از روش MSRT برای پویش محیط بود، ولی همان‌طور که گفته شد هدف هر روبات در حین پویش، شناسایی اقلام پراکنده در محیط یا همان کنکاش در محیط است. در عمل، روبات‌ها به حسگرهای حساس به اقلام مورد جستجو مجهز هستند که به محض قرار گرفتن در دامنه‌ی حسگر، اقلام شناسایی شده و اقدامات لازم انجام می‌گیرد. در الگوریتم حاضر نیز از تکنیک مشابهی استفاده شده است؛ بدین ترتیب که زمانی که اقلام در محدوده‌ی حداکثر دامنه‌ی قابل اندازه‌گیری توسط حسگرها قرار گرفتند، روبات آنها را شناسایی کرده و مختصات مکانی آنها را ذخیره می‌کند. با توجه به اینکه هر روبات در کار روبات‌های دیگر تداخل ندارد و وارد منطقه‌ی کاوش شده توسط آنها نمی‌شود، هر یک از اقلام تنها توسط یک روبات شناسایی می‌شود و هر روبات، توسط تابع SHARE از اقلام شناسایی شده توسط روبات‌های دیگر مطلع می‌شود.

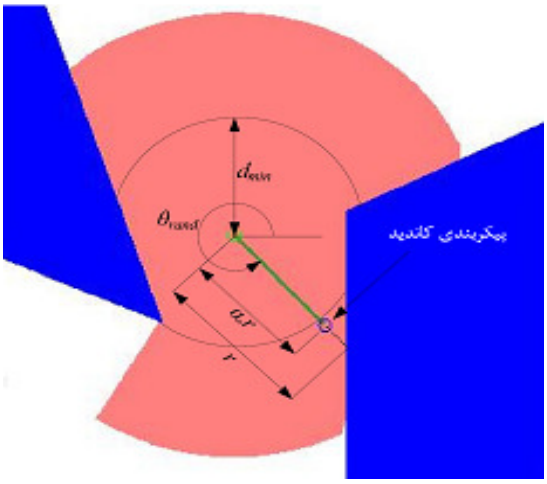
۴. تخصیص و جمع‌آوری اقلام یافته شده

لحظه‌ی اتمام کار کنکاش وقتی است که همه‌ی روبات‌ها حداکثر ناحیه‌ی آزاد مجاور خود را کاوش کرده و به مکان شروع اولیه‌ی خود برگشته‌اند. در این حالت، روبات‌ها از وضعیت کل محیط (شامل محل موانع و محل‌اقلام) اطلاع دارند و در نتیجه، مد کاری روبات‌ها از وضعیت به‌هنگام (آنلاین) به وضعیت غیربه‌هنگام (آفلاین) تبدیل شده است، چرا که هر روبات در طول فرآیند کنکاش اطلاعات خود را در اختیار روبات‌های دیگر قرار داده است. اگرچه تا این مرحله هر روبات تعدادی از اقلام را یافته است، لکن زمان صرف شده یا مسافت پیموده شده توسط هر روبات ممکن است با روبات‌های دیگر اختلاف زیادی داشته و به عبارتی کار کنکاش محیط به نحو متعادلی میان روبات‌ها تقسیم نشده باشد.

بنابراین، روبات‌ها بایستی با لحاظ کردن معیارهای بهینگی مطلوب مانند (۱) کمینه‌سازی مسافت کل پیموده شده‌ی روبات‌ها، و (۲) کمینه‌سازی زمان کل انجام کار، به جمع‌آوری اقلام یافته شده بپردازند. یک عامل کلیدی جهت بهینه‌سازی معیارهای فوق، تخصیص مناسب اقلام به روبات‌ها است به نحوی که مسافت کل پیموده شده توسط روبات‌ها هنگام جمع‌آوری اقلام به حداقل ممکن برسد.

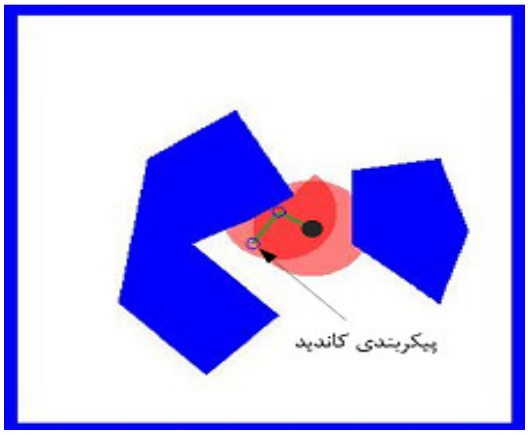


(الف)

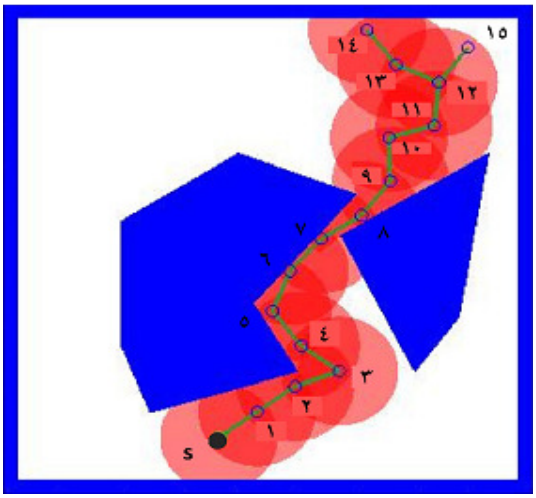


(ج)

شکل ۴. (الف) و (ب) تولید پیکربندی کاندید در روش MSRT. در این پیکربندی‌ها q_{cand} قابل قبول هستند چرا که در فاصله‌ای بزرگ‌تر از d_{min} نسبت به q_{curr} قرار گرفته‌اند و در ناحیه‌ی ایمن محلی گره دیگر واقع نشده‌اند. (ج) جزئیات ایجاد نقطه‌ی کاندید در شکل (الف)



(ب)



شکل ۵. چگونگی پویش و برگشت به عقب روبات

مسئله مورد بحث را می‌توان نوعی از مسئله فروشنده دوره‌گرد چندگانه[1] در نظر گرفت، که در آن چند فروشنده و چندین شهر وجود دارند و فروشندگان بایستی در کوتاه‌ترین زمان (و مسیر) ممکن به شهرها سر زده و کالای خود را بازاریابی کنند به طوری که کلیه شهرها به‌طور دقیق یک بار ملاقات شوند. البته هر فروشنده از شهر متفاوتی سفر خود را شروع کرده و بایستی با همان شهر آن را به پایان برساند. انواع، کاربردها و روش‌های فرمول بندی مسئله فروشنده دوره‌گرد چندگانه در [۱۶] مرور شده است.

از آنجایی که مسئله فروشنده دوره‌گرد چندگانه تعمیم یافته مسئله فروشنده دوره‌گرد می‌باشد، در دسته‌ی مسائل NP-Complete قرار می‌گیرد [۱۷] و بنابراین زمان لازم برای حل دقیق ریاضی آن با افزایش تعداد شهرها و فروشنده‌ها، رشد نمایی می‌یابد و در نتیجه الگوریتم‌های ابتکاری مختلفی جهت حل آن پیشنهاد شده‌اند، مانند [۱۸] که توسط الگوریتم ژنتیک، و [۱۹] که با شبیه سازی تبرید[2] به حل مسئله پرداخته‌اند.

[1] Multiple Traveling Salesman Problem (Multiple-TSP)

[2] Simulated Annealing (SA)

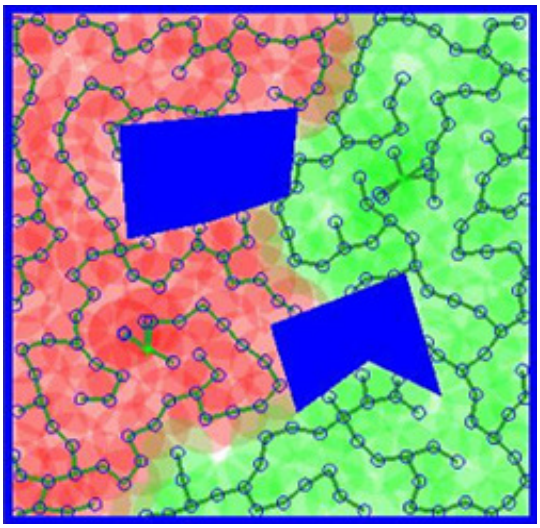
رویکرد دیگری برای حل این مسئله، تجزیه آن به چند مسئله فروشنده دوره گرد تکی می‌باشد که معمولاً از طریق روش‌های خوشه‌بندی شهرها و تخصیص هر خوشه به یک فروشنده انجام می‌گیرد. به عنوان نمونه، تکنیک خوشه بندی k-means برای بخش بندی و تخصیص شهرها به فروشندگان استفاده شده است [۲۰] و سپس هر مسئله فروشنده دوره‌گرد تکی با روش‌های جستجوی ممنوع[1] و شبیه سازی تبرید حل و مقایسه شده‌اند.

در تحقیق حاضر، مسئله تخصیص اقلام (که معادل مسئله فروشنده دوره‌گرد چندگانه است) ابتدا با روش خوشه‌بندی به چند زیر مسئله تجزیه شده و هر یک از آنها نیز توسط روش جستجوی A* حل می‌شوند. در نتیجه اقلام طوری به روبات‌ها تخصیص داده می‌شوند که مسیرهای پیمایش آنها بهینه شوند.

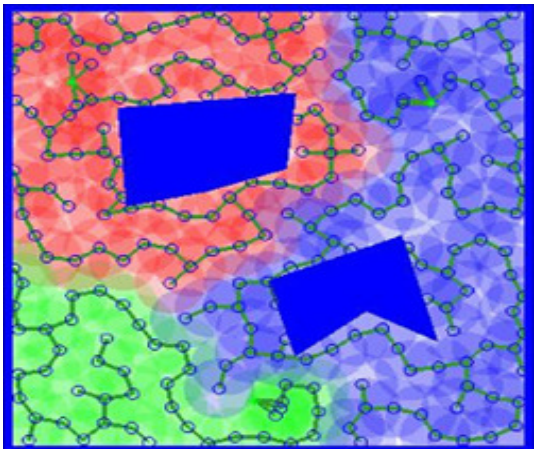
ممکن است این سؤال پیش آید که چرا روبات‌ها هم‌زمان با کنکاش محیط و شناسایی اقلام به جمع‌آوری آنها نمی‌پردازند؟ چند دلیل می‌تواند برای این موضوع وجود داشته باشد:

(۱) جمع آوری اقلام (مین‌ها و غیره) تنها پس از فراهم شدن زیرساخت‌های این کار و در زمانی غیر از پویش محیط انجام شود؛

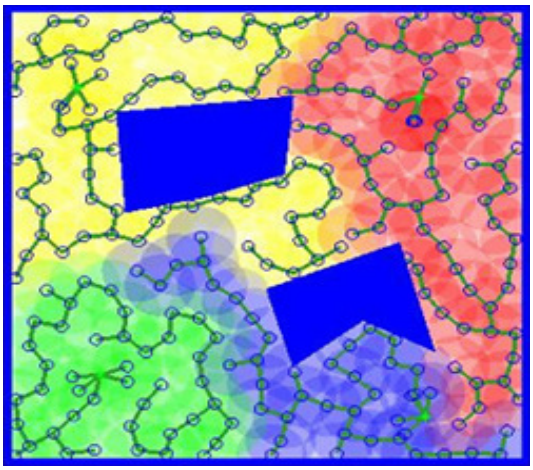
(۲) روبات‌های جستجوگر امکان و ظرفیت برداشتن (و احیاناً خنثی کردن) اقلام‌ها را نداشته باشند و در نتیجه انواع دیگری از روبات مورد نیاز باشد.



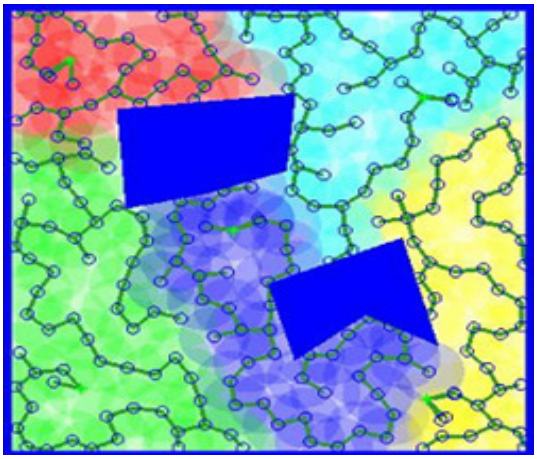
(الف)



(ب)



(ج)



(د)

شکل ۶. نمونه‌هایی از پویش محیط توسط چند روبات: (الف) پویش محیط توسط ۲ روبات؛ (ب) پویش محیط توسط ۳ روبات؛ (ج) پویش محیط توسط ۴ روبات؛ (د) پویش محیط توسط ۵ روبات

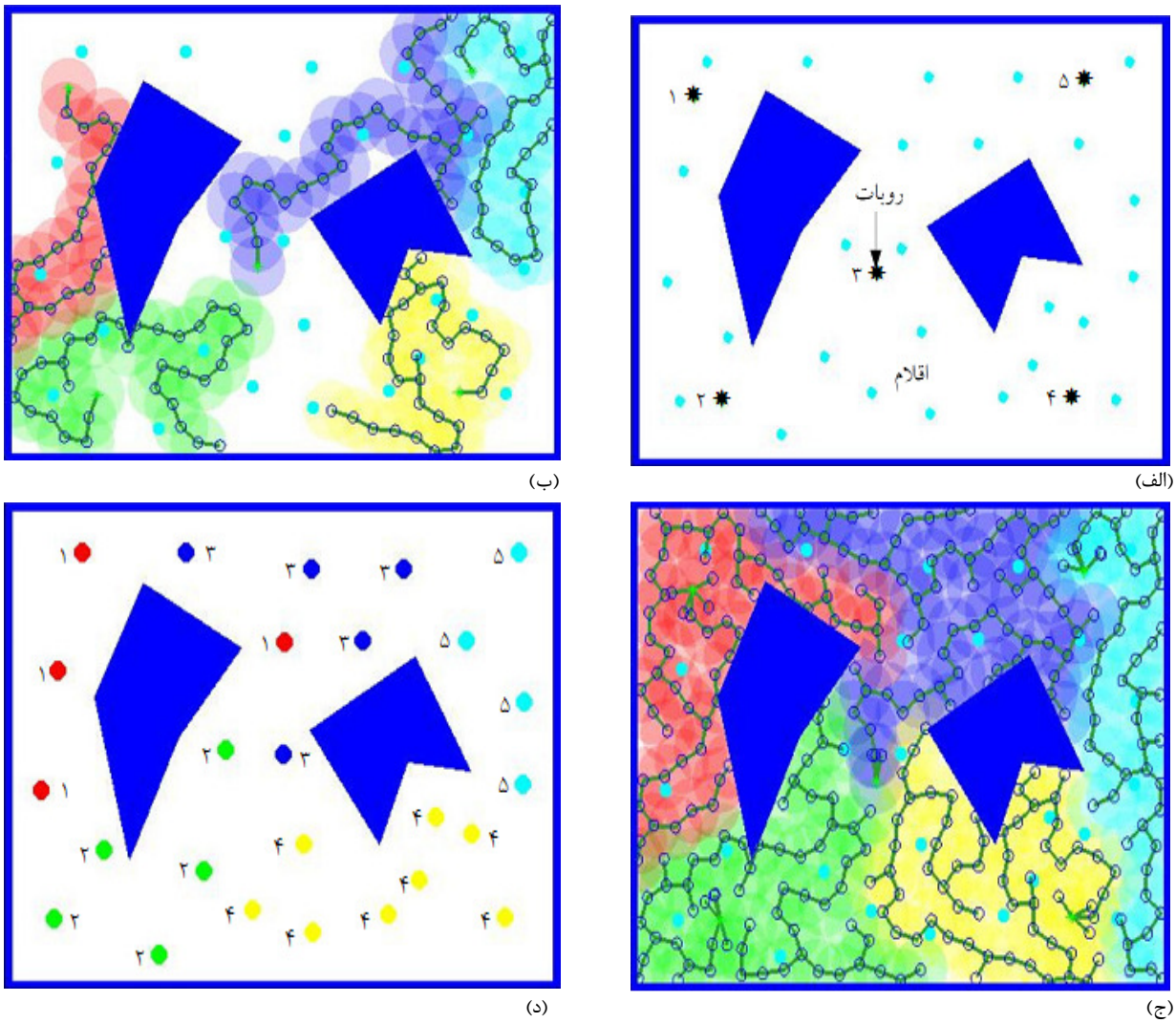
[1] Tabu Search (TS)

۴-۱.خوشه‌بندی اقلام یافته‌شده

از آنجایی که روبات‌ها هنگام شروع کار در قرارگاه‌های خود واقع هستند و به‌طورطبیعی بهتر است که اقلام پراکنده در اطراف خود را جمع آوری نمایند، مناسب است که عمل خوشه‌بندی اقلام به گونه‌ای صورت گیرد که خوشه‌هایی به تعداد روبات‌ها و با مرکزیت هر روبات ایجاد شود. از این نظر، الگوریتم خوشه‌بندی k-means الگوریتم مناسبی جهت انجام این کار تشخیص داده شد چرا که در آن، هدف کمینه‌سازی مجذور خطاها حول مرکز خوشه‌ها است. لذا می‌توان با انتخاب مکان استقرار (قرارگاه) روبات‌ها به عنوان مرکز هر خوشه، اقلام را با توجه به معیار بهینگی به تعداد روبات‌ها خوشه‌بندی کرد. به طور خلاصه الگوریتم k-means شامل گام‌های زیر است:

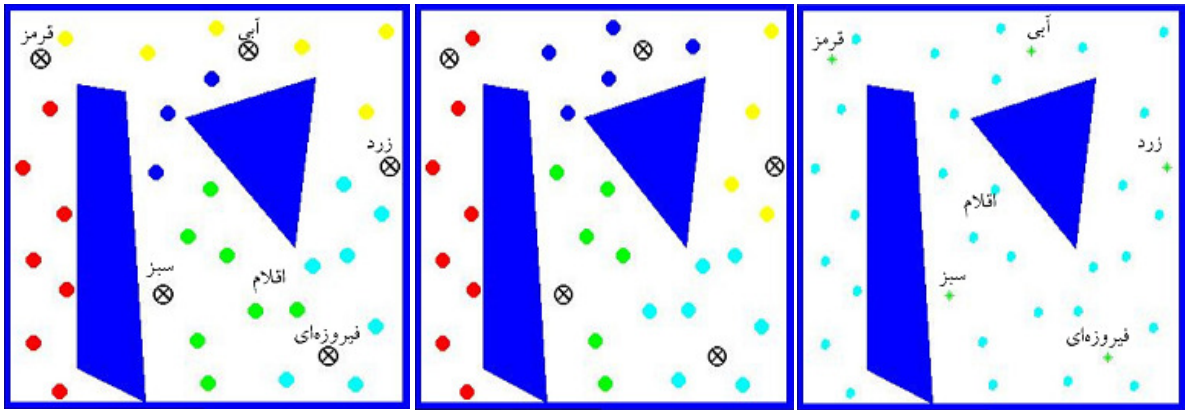
۱- k داده به عنوان مرکز جرم اولیه انتخاب می‌شود.
۲- تک‌تک اعضای مجموعه‌ی داده‌ها به نزدیک‌ترین مرکز جرم اختصاص می‌یابند.
۳- مرکز جرم هر خوشه مجدداً محاسبه می‌شود.
۴- مرحله ۲ و ۳ آنقدر تکرار می‌شود که دیگر مکان مرکز جرم‌ها تغییر نکند. در این صورت، الگوریتم k خوشه‌ی نهایی را به دست می‌دهد.

شکل (۸- ب) نتیجه‌ی عملیات کنکاش به روش MSRT را نشان می‌دهد که هر یک از روبات‌ها به نقطه‌ی شروع خود برگشته و اقلام شناسایی شده توسط هر روبات نیز به رنگ همان روبات نشان داده شده‌اند.



(الف) (ب) (ج) (د)

شکل ۷. محیط کاری به همراه مین‌های پراکنده در محیط که در آن پنج روبات با کنکاش در محیط به شناسایی اقلام پراکنده در محیط می‌پردازند: (الف) محیط کاری به همراه اقلام پراکنده در محیط؛ (ب) مراحل میانی کنکاش؛ (ج) اتمام کار کنکاش؛ (د) اقلام یافته شده توسط هر روبات



(الف) (ب) (ج)

شکل ۸. خوشه‌بندی k-means: (الف) محیط کاری و اقلام؛ (ب) اقلام پیدا شده توسط هر روبات؛ (ج) اقلام اختصاص یافته به هر روبات در نتیجهٔ خوشه‌بندی k-means

جدول ۱. درصد مساحت پوشش داده شده و تعداد اقلام شناسایی شده توسط هر روبات در مسائل پنج، چهار، سه و دو روباتی

مسائل پنج روباتی

شرح	مساحت (٪)		تعداد (٪)	
	M	*SD*	*M*	*SD*
روبات ۱	۲۰٫۰	۴٫۹	۱۹٫۳	۵٫۰
روبات ۲	۱۹٫۰	۴٫۳	۱۶٫۴	۶٫۹
روبات ۳	۲۳٫۸	۴٫۴	۲۶٫۰	۵٫۸
روبات ۴	۲۰٫۶	۵٫۸	۲۱٫۰	۶٫۹
روبات ۵	۱۶٫۶	۶٫۵	۱۷٫۳	۹٫۲

مسائل چهار روباتی

مساحت (٪)		تعداد (٪)	
M	*SD*	*M*	*SD*
۲۴٫۰	۴٫۹	۲۴٫۲	۵٫۷
۲۵٫۶	۵٫۰	۲۶٫۱	۸٫۶
۲۳٫۶	۶٫۴	۲۰٫۶	۶٫۴
۲۶٫۸	۵٫۲	۲۹٫۱	۶٫۷

مسائل سه روباتی

شرح	مساحت (٪)		تعداد (٪)	
	M	*SD*	*M*	*SD*
روبات ۱	۳۱٫۰	۷٫۸	۳۰٫۶	۱۱٫۴
روبات ۲	۳۶٫۴	۸٫۴	۳۸٫۸	۱۱٫۴
روبات ۳	۳۲٫۶	۷٫۷	۳۰٫۶	۱۰٫۶

مسائل دو روباتی

مساحت (٪)		تعداد (٪)	
M	*SD*	*M*	*SD*
۴۷٫۸	۱۳٫۰	۴۹٫۷	۱۴٫۲
۵۲٫۲	۱۲٫۷	۵۰٫۳	۱۴٫۲

نتیجه‌های خوشه‌بندی با الگوریتم k-means در شکل(۸) آورده شده است. در شکل (۸-الف) محیط کاری و روبات‌ها نشان داده شده‌اند، که در آن ۵ روبات آبی، قرمز، زرد، سبز، و فیروزه‌ای در موقعیت‌های شروع خود قرار گرفته و اقلام جمع-آوری شونده نیز به شکل دوایر کوچک در محیط پراکنده‌اند. این روبات‌ها اطلاعات دریافتی از محیط (از قبیل محل موانع و

اقلام شناسایی شده) را با یکدیگر به اشتراک می‌گذارند، ولی چون هنگام کنکاش محیط به روش MSRT نواحی به صورت تصادفی پویش می‌شوند و معیار کوتاه‌ترین فاصله پیمایش مد نظر قرار نمی‌گیرد، بایستی با عمل خوشه‌بندی، اقلام یافته شده مجدداً به روبات‌ها تخصیص داده شوند. در شکل ۸(ج) خوشه‌بندی k-means انجام شده است که در این تخصیص، از آنجایی که قرارگاهِ هر روبات مرکز جرم یک خوشه منظور می‌شود، به طور طبیعی اقلام نزدیک به هر قرارگاه به همان خوشه تخصیص داده می‌شوند.

۴-۲. تعیین ترتیب جمع‌آوری اقلام

پس از اینکه خوشه‌های مناسبی به دست آمد و اقلام با توجه به فواصل واقعیشان از روبات‌ها تخصیص یافتند، هر روبات بایستی به سمت اشیاء اختصاص یافته به خود رفته و آنها را برداشته و به قرارگاه برگردد. همان‌طور که قبلاً اشاره شد این مسئله شبیه به مسئله فروشنده دوره گرد بوده و لذا برای به دست آوردن ترتیب جمع‌آوری اشیاء مدل مسئله فروشنده دوره‌گرد برای هر روبات تشکیل و حل شده است. جهت حل از الگوریتم جستجوی A* استفاده می‌شود.

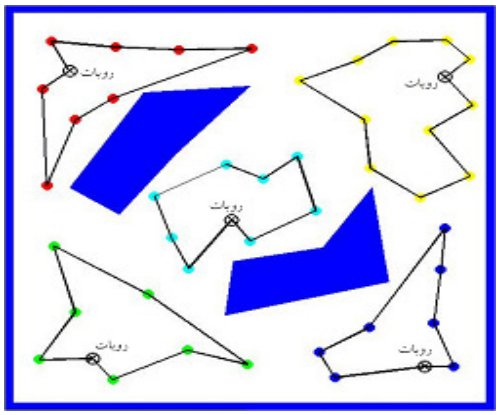
روش A* یک روش بهینه و کارای جستجوی گراف است که به طور گسترده در یافتن مسیر بهینه بین دو گره‌ی گراف مورد استفاده قرار می‌گیرد و به علت دقت و عملکرد پالایش کاربردهای فراوانی یافته است. A* گسترش یافته‌ی روش جستجوی Dijkstra است که با استفاده از یک هیوریستیک (معیاری تخمینی) عملکرد بهتری نسبت به آن پیدا کرده است. A* از یک جستجوی اول- بهترین [1] استفاده می‌کند و مسیری با کم‌ترین هزینه میان دو گره شروع و هدف پیدا می‌کند. جهت مشخص کردن مسیر از یک تابع هزینه‌ی ترکیبی f(x) استفاده می‌کند که مجموع دو تابع است:

(۱) تابع هزینه مسیر g(x) که برابر است با هزینه‌ی گره شروع تا گره فعلی x، و

(۲) یک تخمین از فاصله گره x تا گره هدف که به‌طورمعمول با h(x) نشان داده می‌شود.

بخش h(x) از تابع f(x) بایستی یک هیوریستیک قابل‌قبول [2] باشد، یعنی نباید فاصله گره x از تابع هدف را بیش از مقدار واقعی آن برآورد کند. بنابراین برای کاربردهایی نظیر مسیریابی، h(x) می‌تواند فاصله خط مستقیم تا هدف را اختیار کند چرا که کوتاه‌ترین مسیر ممکن بین دو گره است. اگر h(x) شرط اضافه $h(x) \le d(x, y) + h(y)$ را برای هر یال x و y گراف ارضا کند (d طول یال است)، آنگاه h سازگار [3] نامیده می‌شود. در چنین شرایطی A* می‌تواند بسیار کاراتر اجرا شود و هیچ گرهی بیش از یک بار به کار گرفته نمی‌شود. در واقع، A* معادل با الگوریتم جستجوی Dijkstra با هزینه‌ی کاهش یافته‌ی $d(x,y) := d(x,y) - h(x) + h(y)$ خواهد شد.

مشابه تمامی الگوریتم‌های جستجوی مطلع [4]، الگوریتم A* ابتدا به جستجوی مسیری می‌پردازد که احتمال رسیدن آن به هدف بیشتر است. اما نکته‌ای که این الگوریتم را از الگوریتم‌های جستجوی حریصانه‌ی دیگر متمایز می‌کند آن است که مسیری که از گره شروع تا گره فعلی x پیموده شده را نیز مدنظر قرار می‌دهد. یعنی، بخش g(x) هیوریستیک هزینه‌ی محلی از گره بسط یافته‌ی قبلی نیست بلکه هزینه‌ی گره x از گره شروع است. طرز کار A* چنین است: از گره اولیه شروع می‌کند و مجموعه‌ای از گره‌هایی که دارای اولویت پیمایش هستند را بنام مجموعه OPEN به وجود می‌آورد. گرهی از این مجموعه که هزینه‌ی f(x) کمتری دارد از اولویت بالاتری جهت پیمایش برخوردار است. در هر تکرار الگوریتم، گره با f(x) کمتر از مجموعه حذف می‌شود و گره‌های همسایه‌ی آن به مجموعه اضافه می‌شود و مقادیر h و f این گره محاسبه می‌شود. الگوریتم تا جایی تکرار می‌شود که گره هدف مقدار f کمتری نسبت به دیگر گره‌های مجموعه داشته باشد. مقدار f گره هدف همان طول کوتاه‌ترین مسیر است.همانند جستجوی عرضی [5]، A* یک الگوریتم کامل [6] است و در صورت وجود جواب، آن را خواهد یافت. الگوریتم Dijkstra می‌تواند یک حالت ویژه از روش A* تلقی شود که در آن h(x) = 0 است[۲۱]. جواب حاصل از روش A*، ترتیب جمع‌آوری اشیای هر خوشه توسط هر روبات را مشخص می‌کند که روبات‌ها بایستی آنها را جمع‌آوری کرده و در قرارگاه تخلیه کنند شکل (۹).



شکل ۹. یک نمونه از حل مسئله ترتیب جمع‌آوری اقلام توسط روش A*

[1] Best-First Search
[2] Admissible Heuristic
[3] Consistent
[4] Informed Search
[5] Breadth-First Search
[6] Complete

۴-۳. جمع‌آوری اقلام توسط روبات‌ها

پس از مشخص شدن ترتیب جمع‌آوری اقلام روبات‌ها به جمع آوری آنها می‌پردازند، اما برای انجام این کار بایستی مسیر حرکت روبات‌ها جهت جمع‌آوری اقلام و تخلیه آنها (یعنی حرکت روبات‌ها از قرارگاه تا شیء، از شیء تا شیء، و از شیء تا قرارگاه) در کوتاه‌ترین مسیر ممکن برنامه‌ریزی شود. واضح است که اگر مانعی در مسیر مستقیم بین دو نقطه مبدأ و مقصد وجود نداشته باشد، کوتاه‌ترین مسیر بین دو نقطه، خط مستقیم واصل آنها خواهد بود. در غیر این صورت، می‌توان کوتاه‌ترین مسیر بین دو نقطه‌ی مبدأ و مقصد را از روی گراف دیدنگار[1] به دست آورد.

گراف دیدنگار که در فضای دو بعدی کوتاه‌ترین مسیر بین دو نقطه دلخواه را در حضور موانع به دست می‌دهد گرافی است که گره‌هایش شامل آن دو نقطه و رئوس همه‌ی موانع بوده و یال-هایش کلیه‌ی خطوط راستی هستند که بدون داشتن تقاطع با موانع، هر دو گره قابل رؤیت را به همدیگر وصل می‌کنند [۲۲]. روبات‌ها با استفاده از گراف دیدنگار می‌توانند با طی کوتاه‌ترین مسیر ممکن و طبق ترتیبی که در بخش ۴-۲ به دست آمد به جمع‌آوری اقلام بپردازند. شکل (۱۰) مراحل میانی و پایانی جمع‌آوری اقلام توسط روبات‌ها را نشان می‌دهد.

۵. شبیه سازی و مقایسات عددی

همان‌گونه که در بخش قبل اشاره شد، مسئله جمع‌آوری اقلام توسط روبات‌ها با استفاده از روش‌های ابتکاری جدید برای جمع‌آوری اقلام برنامه‌ریزی شد. از آنجایی که ساختار مسئله مشابه مسئله فروشنده دوره‌گرد چندگانه می‌باشد، جهت اعتبار سنجی روش توسعه داده شده جواب حاصله از آن بایستی با نتایج حاصل از حل دقیق ریاضی مسئله روشنده دوره‌گرد چندگانه مقایسه شود.

مدل ریاضی توسعه داده شده برای مسئله جمع‌آوری اقلام توسط روبات‌ها (فروشنده دوره‌گرد چندگانه) به شکل زیر است:

$$\text{Minimize} \quad \sum_{i=1}^{n+m}\sum_{j=1}^{n+m} C_{ij}X_{ij} \qquad (۱)$$

$$St.$$

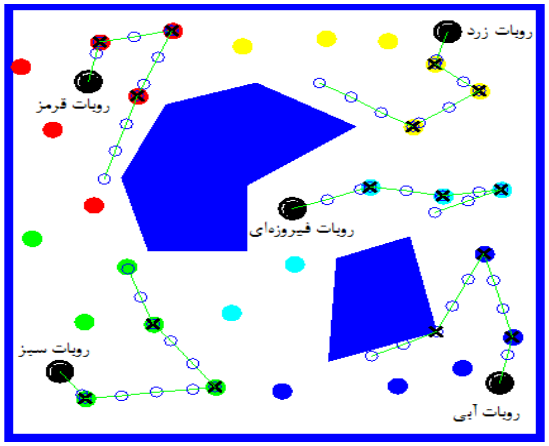
$$\sum_{i=1}^{n+m} X_{ij} = 1, \qquad \forall j = 1,2,\ldots,n+m,\ i \neq j \qquad (۲)$$

$$\sum_{j=1}^{n+m} X_{ij} = 1, \qquad \forall i = 1,2,\ldots,n+m,\ i \neq j \qquad (۳)$$

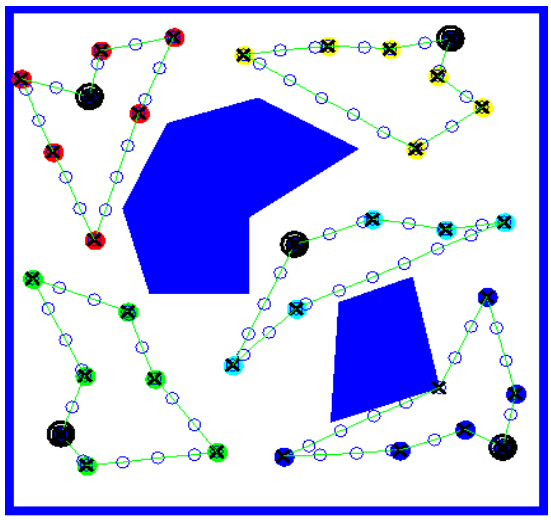
$$Y_i - Y_j + (n+m)X_{ij} \leq n+m-1, \qquad 1 \leq i \neq j \leq n \qquad (۴)$$

$$X_{ii} \in \{0,1\} \qquad \forall i,j \qquad (۵)$$

[1] Visibility Graph



(الف)



(ب)

شکل ۱۰. عملیات جمع‌آوری اقلام توسط روبات‌ها: (الف) مراحل میانی عملیات؛ (ب) پایان عملیات جمع‌آوری اقلام.

در این مدل، i و j اندیس اقلام (مین‌ها) و روبات‌ها هستند (در بازه ۱ تا $n+m$)؛ n تعداد اقلام است؛ m تعداد روبات‌ها است؛ C_{ij} مسافت بین مین‌ها یا روبات‌های i و j است؛ Y_i عددی حقیقی است که از به‌وجود آمدن زیردور[2] جلوگیری می‌کند؛ X_{ij} برابر با ۱ است چنانچه j بلافاصله پس از i در مسیر دور روباتی باشد، و در غیر این صورت برابر با صفر است. در مدل، تابع هدف (۱) عبارت است از کمینه‌سازی مجموع مسافت پیموده شده توسط تمامی روبات‌ها. محدودیت (۲) تضمین می‌کند که تنها یک حرکت از کلیهٔ مین‌های دیگر به سمت مین یا روبات j صورت گیرد. محدودیت (۳) تضمین می‌کند که تنها یک حرکت از مین یا روبات i به سمت تمامی مین‌های دیگر انجام شود. محدودیت (۴) از به‌وجود آمدن زیردور جلوگیری می‌کند و

[2] Subtour

محدودیت (۵) الزام می‌کند که متغیرها مقادیر صحیح صفر یا ۱ را اختیار کنند.

جواب دقیق مسئله جمع‌آوری اقلام با حل مدل فوق توسط نرم افزار LINGO بر روی پردازندهٔ Intel 2 Core×2 GHz و 4 GB RAM به دست آمده است. جواب حاصل، دور کامل بهینه‌ای را برای m روبات به‌وجود می‌آورد که هر دور مشخص‌کنندهٔ ترتیب جمع‌آوری اقلام است و از موقعیت استقرار روبات شروع و به همان ختم می‌شود. نتایج حل چند مسئله در جدول(۲) آورده شده است.

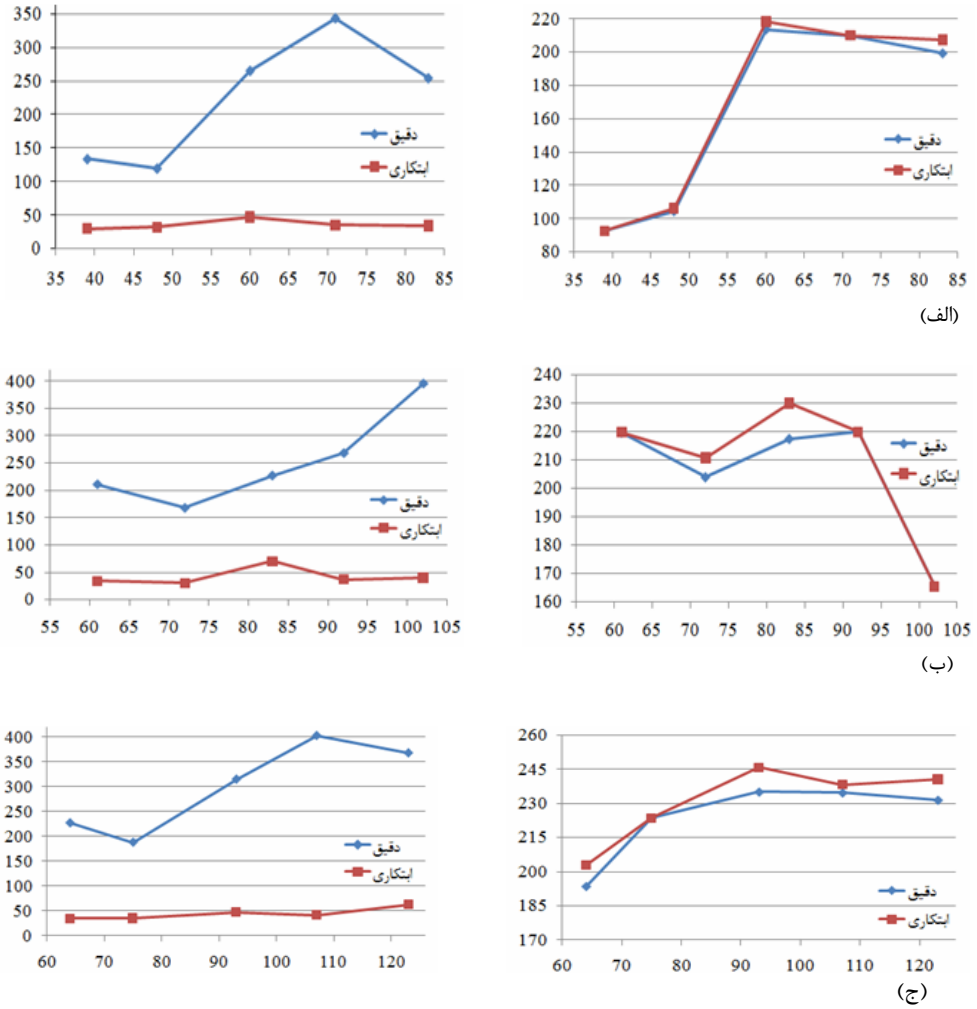
همان‌گونه که از جدول پیداست روش ابتکاری جدید در زمان بسیار کوتاه‌تری از روش دقیق جواب‌های نزدیک به بهینه ارائه می‌کند. شکل‌های ۱۱(الف)-(ج) نتایج مقایسه بین جواب‌ها و زمان‌های حل حاصل از روش ابتکاری و روش دقیق را برای ۳، ۴ و ۵ روبات به طور نموداری نشان می‌دهد.

جدول ۲. مقایسه‌ی جواب حاصل از روش ابتکاری و روش دقیق

تعداد روبات	تعداد اقلام	روش ابتکاری		روش دقیق		شکاف جواب‌های دو روش
		مقدار جواب	زمان (s)	مقدار جواب	زمان (s)	
۳	۳۹	۹۲٫۶۴	۲۹٫۶۴	۹۲٫۶۴	۱۳۴	٪۰٫۰
	۴۸	۱۰۶٫۲۱	۳۱٫۹۹	۱۰۴٫۳۸	۱۲۰	٪۱٫۷
	۶۰	۲۱۸٫۲۴	۴۷٫۰۲	۲۱۳٫۶۶	۲۶۵	٪۲٫۱
	۷۱	۲۱۰/۰۰	۳۵٫۲۳	۲۱۰/۰۰	۳۴۳	٪۰٫۰
	۸۳	۲۰۷٫۳۶	۳۴٫۳۳	۱۹۹٫۳۰	۲۵۴	٪۳٫۹
۴	۶۱	۲۱۹٫۶۸	۳۳٫۹۱	۲۱۹٫۶۸	۲۱۱	٪۰٫۰
	۷۲	۲۱۰٫۷۰	۳۰٫۷۴	۲۰۳٫۹۷	۱۶۸	٪۳٫۲
	۸۳	۲۲۹٫۸۸	۷۰٫۲۰	۲۱۷٫۴۸	۲۲۷	٪۵٫۴
	۹۲	۲۱۹٫۸۲	۳۶٫۸۶	۲۱۹٫۸۲	۲۶۸	٪۰٫۰
	۱۰۲	۱۶۵٫۴۸	۴۰٫۰۹	۱۶۵٫۴۸	۳۹۵	٪۰٫۰
۵	۶۴	۲۰۲٫۸۹	۳۴٫۶۰	۱۹۳٫۴۰	۲۲۷	٪۴٫۷
	۷۵	۲۲۳٫۵۷	۳۵٫۲۰	۲۲۳٫۵۷	۱۸۸	٪۰٫۰
	۹۳	۲۴۵٫۸۲	۴۷٫۴۱	۲۳۵٫۱۷	۳۱۵	٪۴٫۳
	۱۰۷	۲۳۸٫۲۱	۴۱٫۹۴	۲۳۴٫۸۱	۴۰۳	٪۱٫۴
	۱۲۳	۲۴۰٫۶۴	۶۲٫۹۰	۲۳۱٫۵۵	۳۶۸	٪۳٫۸

۵. نتیجه‌گیری

در این مقاله مسئله پویش و کنکاش چند روبات در محیط‌های ناشناخته جهت شناسایی و جمع آوری اقلام پراکنده مانند مین مورد بررسی قرار گرفته است. برای پویش روبات‌ها در محیط و یافتن اقلام پراکنده در آن روش جدید MSRT توسعه داده شده است، که روبات‌ها را قادر می‌سازد کل فضا را بدون تداخل در کار یکدیگر پویش نمایند. پس از یافتن اقلام، با توسعه‌ی مدلی شبیه به مسئله فروشنده دوره‌گرد چندگانه، اقلام به روبات‌ها تخصیص داده می‌شوند. از طریق الگوریتم خوشه‌بندی k-means، کل اقلام شناسایی شده در محیط به تعداد روبات‌ها خوشه‌بندی شده و مسئله فروشنده دوره‌گرد چندگانه به چند مسئله فروشنده دوره‌گرد جزئی تبدیل می‌شود، که با حل هر یک توسط الگوریتم جستجوی A*، ترتیب جمع‌آوری اقلام برای هر روبات به‌دست آورده می‌شود.



شکل ۱۱. مقایسات کیفیت جواب (راست) و زمان حل (چپ) دو روش ابتکاری و دقیق: (الف) تا (ج) به ترتیب عملیات ۳، ۴، و ۵ روبات.

۵. مراجع

[1] Rooker, M. N.; Birk, A. "Multi-Robot Exploration Under The Constraints of Wireless Networking."; Control Engineering Practice 2007, 15, 435-445.

[2] Yamauchi, B. "A Frontier-Based Approach For Autonomous Exploration."; IEEE Int. Conf. Rob. & Automation 1997, 146-151.

[3] Hazon, N.; Kaminka, G. A., "On Redundancy, Efficiency, and Robustness in Coverage for Multiple Robots."; Robotics and Autonomous Systems 2008, 56, 1102-1114.

[4] Roy, N.; Dudek, G., "Collaborative Robot Exploration and Rendezvous: Algorithms, Performance Bounds and Observations."; Autonomous Robots, 2001, 11, 117-136.

[5] Atay, N.; Bayazit, B. "Mixed-Integer Linear Programming Solution to Multi-Robot Task Allocation Problem."; Technical Report WUCSE-2006-54, Washington Univ. In Saint Louis, 2006.

[6] Beard, R. W.; Mclain, T. W.; Goodrich, M. A.; Anderson, E. P. "Coordinated Target Assignment and Intercept for Unmanned Air Vehicles."; IEEE Trans. on Rob. and Automation 2002, 18, 911-922.

[7] Acar, E. U.; Choset, H.; Rizzi, A. A.; Atkar, P. N.; Hull, D. "Morse Decompositions for Coverage Tasks."; International Journal of Robotics Research 2002, 21, 331-344.

[8] Choset, H. "Coverage Of Known Spaces: The Boustrophedon Cellular Decomposition."; Autonomus Robots 2000, 9, 247-253.

[9] Acar, E. U.; Choset, H. "Sensor-Based Coverage of Unknown Environments: Incremental Construction of Morse Decomposition" International Journal Of Robotics Research 2002, 21, 345-366.

[10] Parker, L. "Distributed Algorithms for Multi-Robot Observation of Multiple Moving Targets."; Autonomous Robots 2002, 12, 231-255.

[11] Murrieta, R.; Tovar, B.; Hutchinson, S. "A Sampling-Based Motion Planning Approach to Maintain Visibility of Unpredictable Targets."; Autonomous Robots 2005, 19, 285–300.

[12] Kuffner, J. J.; Lavalle, S. M. "RRT-Connect: An Efficient Approach to Single Query Path Planning."; IEEE Int. Conf. Rob.& Autom. 2000, 995-1001.

[13] Oriolo, G.; Venditelli, M.; Freda, L.; Troso, G. "The SRT Method: Randomized Strategies for Exploration."; IEEE International Conference on Robotics and Automation 2004, 4688-4694.

[14] Espinoza, J. L.; Sanchez, A. L.; Osorio, M. "Exploring Unknown Environments with Mobile Robots Using SRT-Radial."; IEEE Int. Conf. on Intelligent Robots and Systems 2007, 2089-2094.

[15] Freda, L.; Oriolo, G. "Frontier-Based Probabilistic Strategies for Sensor-Based Exploration."; IEEE International Conference on Robotics and Automation 2005, 3881-3887.

[16] Bektas, T. "The Multiple Traveling Salesman Problem: An Overview of Formulations and Solution Procedures."; International Journal Of Management Science 2006, 34, 209-219.

[17] Garey, M.; Johnson, D. S. "Computers and Intractability: A Guide to the Theory of NP-Completeness."; W. H. Freeman & Company Pub., New York, 1979.

[18] Tang, L.; Liu, J.; Rong, A.; Yang, Z. "A Multiple Traveling Salesman Problem Model for Hot Rolling Scheduling in Shangai Baoshan Iron & Steel Complex."; European Journal of Operational Research 2000, 124, 267-282.

[19] Song, C.; Lee, K.; Lee, W.D. "Extended Simulated Annealing for Augmented TSP and Multi-Salesmen TSP."; Proc. International Joint Conference on Neural Networks, 2003, 3, 2340-2343.

[20] Nallusamy, R.; Duraiswamy, K.; Dhanalaksmi, R.; Parthiban, P. "Optimization of Non-Linear Multiple Traveling Salesman Problem Using K-Means Clustering, Shrink Wrap Algorithm and Metaheuristics."; Int. J. Nonlinear Science 2009, 8, 480-487.

[21] Hart, P. E.; Nilsson, N. J.; Raphael, B. "A Formal Basis for Heuristic Determination of Minimum Cost Paths."; Transactions of Systems Science and Cybernetics 1968, 4, 100-107.

[22] Latombe, J. C. "Robot Motion Planning."; Kluwer Academic Publishers, London, 1991.