نشـــریه پـــژوهـــش‌های مهنـــدســـی صنـــــایع
در سیستـــم‌هـــای تولیـــــد

نشـــریه پـــژوهـــش‌های مهنـــدســـی صنـــــایع
در سیستـــم‌هـــای تولیـــــد

ISSN: 2345-2269

سال پنجم، شماره دهم، بهار و تابستان ۱۳۹۶، صفحـــــه ۳۱-۵۱
www.ier.basu.ac.ir

DOI: 10.22084/ier.2017.8001.1392

پویش محیط ناشناخته و تخصیص کار توسط ربات‌های چندگانه با استفاده از یک معماری دو سطحی و الگوریتم ژنتیک

حسن یوسفی حلوایی[۱]، الیپس مسیحی[۲،*]

۱. کارشناسی ارشد، دانشکده مهندسی صنایع و سیستم‌ها، دانشگاه تربیت مدرس تهران، ایران.
۲. دانشیار، دانشکده مهندسی صنایع و سیستم‌ها، دانشگاه تربیت مدرس تهران، ایران.

اطلاعات مقاله

تاریخچه مقاله:
دریافت ۱۳۹۵/۰۶/۵
پذیرش ۱۳۹۶/۰۲/۱۹

کلمات کلیدی:
ربات های چندگانه
پویش محیط
مسیریابی بهنگام
تخصیص کار
الگوریتم ژنتیک

خلاصه

سیستم ربات‌های چندگانه جهت انجام کارهایی که ماهیت توزیع شده در فضا، زمان و یا عملکرد دارند، بسیار مناسب می‌باشند. برای مسائلی نیز که قابلیت تفکیک به زیرمسائل مستقل از هم را دارند، استفاده از سیستم ربات‌های چندگانه به دلیل کاهش زمان تکمیل کار به صرفه خواهد بود. به کارگیری چنین سیستم‌هایی مستلزم تخصیص مناسب کارها در بین ربات‌ها است که از جمله مسائل NP-hard می‌باشد. در این مقاله، یک معماری دو سطحی جهت حل مسئله پویش محیط و یافتن کارهای موجود در آن توسط ربات‌های غیر همگن ارائه گردیده و پس از تخصیص کار بین ربات‌ها، مسیریابی ربات به سمت کارهای مورد نظر صورت می‌پذیرد. در سطح اول که از رویکرد توزیع شده استفاده شده است، ربات‌ها به صورت غیرمتمرکز و با استفاده از الگوریتم Multi SRT به پویش محیط پرداخته و مسیریابی آنها به سمت کارهای مورد نظر با استفاده از الگوریتم Multi Tangent Bug انجام می‌گیرد. در سطح دوم، به صورت متمرکز و با استفاده از الگوریتم ژنتیک ارائه شده (FGA)، ربات‌ها به کارهای مورد نظر تخصیص می‌یابند. انجام هر یک از کارها باعث ایجاد مطلوبیت برای سیستم شده و انجام تمامی کارها هدف غائی سیستم تلقی می‌شود. در ادامه، نتایج حاصل از شبیه‌سازی مسائل آزماینده مختلف، قابلیت اطمینان معماری ارائه شده در سطح اول و همچنین دقت و سرعت جواب‌های به دست آمده در سطح دوم را به اثبات رسانده و در پایان جهت اعتبارسنجی الگوریتم ژنتیک ارائه شده نتایج آن با الگوریتم ژنتیک NSGA-II مقایسه شده است.

۱- مقدمه

یکی از کاربردهای سیستم‌های چندرباتی، پویش محیطهای پیچیده و خطرناک می باشد. در چنین محیط‌هایی، ربات‌ها بایستی توانایی جمع‌آوری اطلاعات از محیط و به اشتراک گذاشتن آن با سایر هم‌تیمی‌هایشان را داشته باشند تا در نهایت با هماهنگی و همکاری با یکدیگر برنامه‌ریزی لازم جهت دستیابی به اهداف تعیین شده را انجام دهند. همکاری و هماهنگی برای برنامه‌ریزی و انجام وظایف مختلف از معیارهای مهم سیستم‌های چند رباتی به شمار می‌رود که در سال‌های اخیر مورد توجه شمار زیادی از محققان حوزه‌ی رباتیک و بهینه‌سازی قرار گرفته است. در برخی شرایط که علاوه بر انجام وظایف مختلف معیار بهینگی نیز مطرح می‌شود اهمیت ایجاد هماهنگی دوچندان می‌گردد. می‌توان گفت هماهنگی و همکاری هسته مرکزی بسیاری از کارهای انجام شده روی سیستم‌های چندرباتی هستند [۱].

با پیشرفت فناوری و به وجود آمدن مسائل جدید در حوزه‌های مختلف، کاربرد سیستم‌های چندرباتی بیش از پیش مورد استقبال واقع شده است. از آن جمله می‌توان به استفاده از این سیستم‌ها در

* نویسنده مسئول: الیپس مسیحی
تلفن: ۸۲۸۸۴۹۳۹-۰۲۱؛ پست الکترونیکی: masehian@modares.ac.ir

حوزه‌های لجستیک و انبارداری [۲]، بازیابی اطلاعات در محیط‌های طبیعی [۳]، تشکیل و آرایش تیم‌های ورزشی [۴]، مأموریت کاوش و جستجو در سیارات دیگر [۵] و غیره اشاره نمود. یکی از جنبه‌های مهم همکاری و هماهنگی در ربات‌های چندگانه تخصیص کار بین آن‌ها است که عمدتاً به صورت توزیع شده و یا متمرکز درنظر گرفته می‌شود [۶]. هدف از تخصیص کار بین ربات‌های چندگانه، پاسخ به این سؤال است که "هر یک از ربات‌ها باید کدامیک از کارها را انجام دهد؟". دو رویکرد مختلفی که در مسائل تخصیص کار بین ربات‌ها در نظر گرفته می‌شوند، رویکردهای *متمرکز* و *توزیع شده* می‌باشند [۷]. در مسائلی که از رویکرد متمرکز استفاده می‌کنند، جهت بیشینه‌سازی کارایی کل سیستم و نیز کمینه‌سازی هزینه، معیارهایی از قبیل سود، هزینه، منابع و زمان باید در طول تخصیص کار در نظر گرفته شوند. درنتیجه می‌توان گفت که ماهیت تخصیص کار در ربات‌های چندگانه حالت بهینه‌سازی با معیارهای چندگانه را دارد. تخصیص کار در چنین حالتی تبدیل به مسئله‌ی NP-hard می‌گردد [۸]. این مسئله با توجه به قابلیت تبدیل MRTA[1] به مسائل مختلف NP-hard از قبیل قبیل مسئله‌ی فروشنده‌ی دوره‌گرد چندگانه [۹]، مسئله‌ی کمینه–بیشینه مسیریابی وسایل نقلیه [۱۰] به راحتی قابل اثبات است. در این رویکرد، کل ربات‌ها به صورت یک واحد در نظر گرفته می‌شوند که از طریق کنترل مرکزی قابل کنترل بوده و اطلاعات به طور همسان بین تمامی ربات‌ها به اشتراک گذاشته می‌شوند.

رویکرد دیگری که مورد توجه بسیاری از محققان قرار گرفته است رویکرد توزیع شده است. در این رویکرد، کنترل کننده‌ی مرکزی وجود نداشته و ربات‌ها فقط از طریق ارتباطات محلی که با یکدیگر برقرار می‌کنند از محیط آگاهی پیدا کرده و اقدام به انجام کارهای مختلفی می‌کنند. در این رویکرد هر کدام از ربات‌ها تنها می‌تواند با تعداد اندکی از ربات‌های همسایه‌ی خود ارتباط برقرار ساخته و به تبادل اطلاعات بپردازد [۵].

در سال ۲۰۱۳ یک طبقه بندی برای تخصیص وظایف در سیستم ربات‌های چندگانه ارائه شده است [۱۱]، که در آن سه معیار مهم برای مقایسه مسائل تخصیص وظایف پیشنهاد می‌شود:

جنبه اول، ربات‌های تک کاره (ST)[2] در مقابل ربات‌های چند کاره (MT)[3] است. ربات‌های تک کاره در یک زمان خاص فقط یک کار را می‌توانند انجام دهند و ربات‌های چند کاره می‌توانند چند کار را همزمان انجام دهند.

جنبه دوم، وظایف تک رباتی (SR)[4] در مقابل وظایف چند رباتی (MR)[5]. وظایف تک رباتی به این معنی است که هر وظیفه نیازمند یک ربات است و وظایف چند رباتی به این معنی است که بعضی وظایف می‌توانند به چند ربات نیاز داشته باشند.

1. Multi-Robot Task Allocation
2. Single-Task
3. Multi-Task
4. Single-Robot
5. Multi-Robot

جنبه سوم، تخصیص فوری (IA)[6] در مقابل تخصیص زمانبر (TA)[7]. در تخصیص فوری شرایط محیط به گونه ای است که اطلاعات مختص اکنون است و تخصیص باید فوراً صورت گیرد. در تخصیص زمانبر اطلاعات کافی نیست و در طول زمان تخصیص صورت می‌گیرد.

۱-۱- هماهنگی در سیستم‌های چند رباتی

در سیستم‌های چند رباتی، اولین نوع هماهنگی تسهیم فضای کاری بین ربات‌های مختلف است. هدف از ایجاد این نوع هماهنگی اجرای برنامه‌های درنظر گرفته شده توسط ربات‌ها به صورت ایمن و منسجم با درنظرگیری سایر ربات‌ها است. دومین عامل مهم، هماهنگی زمانی است. این نوع از هماهنگی ربات‌ها را قادر می‌سازد تا کارهای مختلف را قبل، بعد و به صورت همزمان نسبت به سایر ربات‌ها انجام دهند. این ویژگی در به‌کارگیری تعداد زیادی از ربات-ها بسیار مفید است.

ایجاد هماهنگی یکی از ضرورت‌های سیستم‌های چندرباتی است. ربات‌ها بایستی بتوانند از رفتار هم تیمی‌های خود آگاه شده و با توجه به آنها برنامه‌ریزی کنند. علاوه بر هماهنگی در مسائل چند رباتی توجه به منابع و محدودیت‌ها نیز از عوامل مهم می‌باشند. مثلاً در مسئله‌ی جستجو و نجات، رسیدن به هدف در یک زمان خاص ارزشمند است. در مسائل دیگری نظیر تشکیل یک فرم خاص و یا حرکت با یک فرم ثابت توسط ربات‌ها، یک گروه از ربات‌های متحرک یک فرم هندسی خاص به خود گرفته و جهت حرکتشان را با هدف مورد نظر همسو می‌کنند، و هر ربات باید با ربات‌های دیگر هماهنگ باشد و به درستی عمل کند. هماهنگی بین ربات‌ها می‌تواند در قالب محدودیت‌های متقابلشان بیان شود [۱۲].

۱-۲- همکاری در سیستم ربات‌های چندگانه

یکی دیگر از مؤلفه‌های مهم سیستم‌های چندرباتی همکاری است که به عنوان رفتار گروهی جهت دستیابی به هدف مخصوص تعریف می‌شود. همکاری را می‌توان به هم پیوستن ربات‌ها برای انجام یک فعالیت دانست که در نتیجه آن کارایی بیشتر شده و یا در منابع صرفه جویی حاصل می‌شود. غالباً بدون همکاری هم می‌توان یک مأموریت را به پایان رساند ولی نتیجه بهینه نخواهد بود [۱۳].

۱-۳- مسئله پویش محیط و تخصیص کار

مسئله‌ی پویش محیط یکی از شاخه‌های مهم در تحقیقات سیستم‌های چند رباتی به شمار می‌رود. هدف اصلی در این مسئله جستجو و پویش حداکثر محیط در کمترین زمان ممکن با استفاده از تعداد مشخصی ربات است. استفاده از الگوریتم‌های هماهنگی در مأموریت پویش برای بالابردن قابلیت اطمینان و بهره‌وری بسیار موثر هستند. قابلیت اطمینان به این معنی است که تیم در برابر از کار افتادن ربات‌ها مقاوم بوده و کارایی کل سیستم با خرابی چند جزء

6. Instantaneous Assignment
7. Time-extended Assignment

تحت تاثیر قرار نگیرد. به دلیل اینکه با خرابی یک جزء کارایی سیستم متمرکز به شدت کاهش می‌یابد، استفاده از الگوریتم نامتمرکز برای بالا بردن بهره‌وری مطلوبیت فراوانی دارد. لازمه بهره‌وری نیز اتمام پویش محیط با کمترین هزینه، زمان و سایر منابع مهم است.

هنگامی که از یک تیم ربات درمسئله‌ی پویش استفاده می‌شود مسئله‌ی اصلی هماهنگ کردن حرکت ربات‌هاست. اگر هماهنگی نباشد ممکن است همه‌ی ربات‌ها یک مسیر را دنبال کنند و به اندازه یک تک ربات برای پویش محیط زمان صرف شود. بنابراین مسئله‌ی اصلی انتخاب حرکت‌های مختلف برای ربات‌های متفاوت است تا همزمان نواحی مختلفی را کاوش کنند. یکی از جدیدترین روش‌های پیشنهادی جهت پویش و پوشش کل محیط کاری، استفاده از روش boustrophedon و مکانیزم پس‌رو به نام الگوریتم BoB می‌باشد [۱۴]. در این روش ابتدا ربات‌ها در سیستم مورد نظر مطابق رویکرد مبتنی بر بازار طراحی می‌شوند. سپس بدون هیچگونه کنترل کننده مرکزی ربات‌ها با استفاده از تعاملات محلی نواحی بدون همپوشانی را به صورت تدریجی از طریق حرکات رفت و برگشتی ایجاد می‌کنند. جهت پویش و پوشش کل محیط کاری، هر یک از ربات‌ها مجهز به مکانیزم هوشمند پس‌رو هستند که با استفاده از روش جستجوی A^* ربات را به سمت نزدیکترین نواحی کشف نشده هدایت می‌کند. هنگامی که گره‌ای جهت حرکت پس‌رو وجود نداشته باشد، کل محیط کاری پویش شده است. از نقاط مثبت الگوریتم BoB کارایی این روش در معیارهای نرخ پویش محیط، طول مسیر پویش و حجم کاری توزیع شده در بین ربات‌ها می باشد. به دلیل استفاده از روش تجزیه سلولی جهت بخش‌بندی فضای کاری به سلول‌های مجزا، زمان اجرای این الگوریتم و پیچیدگی آن بالا خواهد بود.

یکی دیگر از روش‌های پویش محیط استفاده از اصول ارتباطی مبتنی بر فرومون می باشد [۱۵]. این رویکرد که StiCo نامیده شده است بر مبنای ارتباط Stigmergic می‌باشد که توسط آقای مارکو دوریگو [۱۶] معرفی شده است. اساس کار StiCo تقسیم‌بندی محیط کاری به نواحی دایره‌ای مساوی (قلمرو) می‌باشد که هر ربات مسئول حرکت و پویش یکی از این نواحی است. هر ربات بدون نیاز به ارتباط مستقیم با سایر ربات‌ها با استفاده از فرومونی که بر روی مرزهای قلمرو خود برجای می‌گذارد به سایر ربات‌ها این مفهوم را می‌رساند که وارد قلمرو نشده و سایر نواحی را پوشش دهند. هر ربات همانند مورچه‌ها دارای دو سنسور جهت تشخیص فرومون سایر ربات‌ها می‌باشد. از مزایای الگوریتم فوق می‌توان به قابلیت اطمینان بالا، پیچیدگی و زمان اجرای پایین اشاره نمود. به دلیل استفاده از روش فراابتکاری مبتنی بر فرومون در پویش محیط، دقت پایین جواب‌های حاصل شده از معایب این الگوریتم در مقایسه با روش‌های پویش محیط مبتنی بر گراف می باشد.

به کارگیری الگوریتم حرکت دسته جمعی پرندگان[1] جهت تخصیص پویای کار در بین ربات‌های چندگانه روشی است همچنین مورد توجه قرار گرفته است [۱۷]. در این روش ابتدا مسئله تخصیص پویای کار مدل سازی شده و سپس هر یک از ربات‌ها به عنوان یک ذره (Particle) درنظر گرفته می‌شود که بدون وجود کنترل کننده مرکزی بایستی به صورت مجزا الگوریتم منحصر به خود را اجرا نماید تا در نهایت به صورت توزیع شده کارهای موجود در محیط انجام گیرند. الگوریتم پیشنهادی بر روی ربات های واقعی ELISA III پیاده‌سازی شده و کارایی و عملکرد روش فوق مورد تحلیل قرار گرفته است. نوآوری تحقیق فوق منحصر به ارائه روشی جهت تخصیص کار در بین ربات بوده و در حوزه پویش محیط و مسیریابی ربات‌ها نوآوری خاصی ارائه نشده است.

از دیگر روش‌های پیشنهادی که از الگوریتم‌های فرابتکاری مبتنی بر جمعیت استفاده می‌کنند می‌توان به کاربرد الگوریتم ژنتیک چند هدفه جهت تخصیص کار در ربات‌های چندگانه اشاره نمود [۱۸]. رویکرد استفاده شده در این روش تلفیقی از تخصیص کار مبتنی بر مزایده و همچنین الگوریتم ژنتیک چند هدفه (NSGA-II)[2] می‌باشد که جهت تخصیص ربات‌های چندگانه به کارهای موجود در محیط به صورت توزیع شده استفاده می‌شود. در این روش درجه پیچیدگی الگوریتم پیشنهادی با روش‌های قبلی موجود مقایسه شده و کارایی آن بررسی می‌گردد. دقت زیاد و زمان بالا از ویژگی‌های الگوریتم ارائه شده است.

یکی دیگر از رویکردهای موجود جهت ایجاد هماهنگی در پویش محیط استفاده از روش اولویت‌بندی [۱۹] و ارائه روشی جهت بهینه سازی آن در سیستم نامتمرکز می‌باشد. در این رویکرد ربات‌ها براساس اهمیت وظایف و یا براساس منابع مهم اولویت بندی می‌شوند؛ به طوری که مسیر برای ربات‌ها یکی پس از دیگری بر اساس اولویت محاسبه می‌شوند. هنگامی که حرکت یک ربات با اولویت P_j برنامه‌ریزی می‌شود باید این نکته را در نظر گرفت که ربات با اولویت $P_i < P_j$ قبلاً بایستی برنامه‌ریزی شده باشد. یافتن اولویت‌های مطلوب به نوبه خود یک مساله‌ی مهم است. یکی از کارهای تحقیقاتی صورت گرفته در این مورد استفاده از یک الگوریتم جستجوی تصادفی با استفاده از تپه نوردی برای یافتن راه حل و کمینه کردن کلی طول مسیر می‌باشد [۲۰].

در [۱۹] مساله‌ی هماهنگی مسیر با استفاده از الگوریتمی به نام شبکه‌بندی حل شده و در آن نشان داده شده است که می‌توان یک مسیر را به قسمت‌هایی تقسیم کرد تا ربات‌ها با هم برخوردی نداشته باشند. این الگوریتم برای ۱۵۰ ربات اجرا شده که از آن میان فقط ۱۰ مسیر با هم برخورد داشته و زمان محاسبه بر حسب دقیقه بوده است. در [۲۱] مساله‌ی هماهنگی با تغییر دادن زمان شروع هر ربات به نتایج خوبی دست پیدا کرده است. در این روش وقتی رباتی شروع به

1. Particle Swarm Optimization
2. Non-dominated Sorting Genetic Algorithm- II

حرکت می‌کند تا زمانی که به هدف خود نرسد اصلا متوقف نخواهد شد.

روش جالب دیگری برای هماهنگ کردن ربات‌های چندگانه توسط پارکر [۲۲] به نام ALLIANCE معرفی شده است. ALLIANCE یک چارچوب بوده که برای هماهنگ کردن سیستم‌های چند رباتی غیر مشابه با استفاده از روش‌های رفتاری بکار می‌رود. از روش هایی که از الگوریتم‌هایی مبتنی بر بازار استفاده می‌کنند می‌توان [۲۳] را نام برد. در روش پیشنهادی جهت هماهنگی ربات‌ها در مسئله‌ی پویش محیط و تخصیص کار در بین آنها از الگوریتم تخصیص کار مبتنی بر مزایده به صورت توزیع شده استفاده شده و نتایج مطلوبی حاصل شده است.

۲- معرفی مسئله

مسئله‌ی در نظر گرفته شده در این مقاله پویش محیط ناشناخته توسط ربات‌های چندگانه و تخصیص کارهای پیدا شده به ربات‌ها است. فضای کاری در نظر گرفته شده به صورت بهنگام (برخط) بوده و ربات‌ها هیچ اطلاعی از محیط در بدو شروع به جستجو ندارند. همچنین موانع متعددی در محیط در نظر گرفته شده‌اند که ربات‌ها در حین پویش و همچنین حرکت به سمت کارهای پیدا شده بایستی از برخورد با آن‌ها اجتناب ورزند. از کاربردهای واقعی این مسئله می‌توان به پویش محیط برای مین‌یابی، جستجوی غذا در محیط، رسیدگی به نیازهای بیماران در یک بخش بیمارستانی، و عملیات امداد و نجات در هنگام سوانح اشاره کرد. در ذیل مفروضات در نظر گرفته شده برای هر کدام از مؤلفه‌های مسئله ذکر می‌شوند.

۲-۱- ربات‌های چندگانه

از لحاظ شکل ظاهری ربات‌ها یکسان بوده و به صورت دایره‌ای می‌باشند. ولی از سایر جهات ربات‌ها ناهمگن بوده و دارای قابلیت‌ها و توانایی‌های متفاوتی می‌باشند. در این مسئله فرض بر این است که یک حافظه‌ی کلی وجود داشته و اطلاعاتی که ربات‌ها از محیط کسب می‌کنند در آن حافظه ثبت شده و بر اساس آنها تخصیص کار به ربات‌ها صورت می‌گیرد. این حافظه به عنوان یک سیستم مرکزی عمل نموده و تصمیم‌گیری‌های لازم را انجام می‌دهد. همچنین هر یک از ربات‌ها دارای ویژگی‌هایی از قبیل شعاع حسگر، سرعت حرکت و حافظه‌ی محدود نیز می‌باشند. لازم به توضیح است که خطاهایی مانند خطای حسگر، لغزش و همچنین اتمام سوخت درنظر گرفته نشده‌اند.

۲-۲- معیارهای بهینگی

در این مسئله هدف بهینه کردن معیارهای مختلف با در نظرگیری محدودیت‌های موجود است. در زیر به معیارهای مختلف بهینگی به همراه محدودیت‌های آن‌ها اشاره می‌شود.

تابع مطلوبیت: هر یک از کارهای موجود در محیط دارای یک مطلوبیت و همچنین یک پنجره‌ی زمانی می‌باشند. چنانچه کاری در بازه‌ی زمانی مشخص شده‌اش انجام نگیرد، مطلوبیت آن کار کاهش یافته تا اینکه به صفر برسد. هدف مسئله بیشینه نمودن مطلوبیت حاصل از انجام کارها در بازه‌های زمانی مربوطه می‌باشد.

پویش حداکثر محیط: معیار دیگری که بایستی بیشینه گردد مقدار پوشش محیط کاری است. ربات‌ها بایستی طوری حرکت کنند تا همواره در محیط‌های جدید به جستجو پرداخته و وارد قسمت‌هایی که قبلاً توسط خود ربات یا سایر ربات‌ها پویش شده‌اند نشوند. در صورت در نظرگرفتن این عامل، حرکت ربات‌ها به گونه‌ای خواهد بود که بتوانند حداکثر مقدار محیط را جستجو کنند.

مقدار مسافت طی شده توسط ربات‌ها: یکی دیگر از معیارهای در نظر گرفته شده جهت کمینه‌سازی، مقدار مسافت طی شده توسط مجموعه‌ی ربات‌ها است. در صورت تخصیص مناسب کار در بین ربات‌ها، مسافت طی شده توسط ربات‌ها جهت رسیدن به کار مورد نظر کوتاه‌تر شده و در نتیجه مجموع مسافت طی شده توسط کل ربات‌ها کمینه می‌گردد.

علاوه بر معیارهای بهینگی، محدودیت‌های مختلفی نیز در مسئله‌ی مورد نظر لحاظ شده‌اند که در زیر به آن‌ها اشاره می‌شود:

پرهیز از برخورد با موانع: یکی از اصلی‌ترین محدودیت‌های پویش محیط و حرکت به سمت کارها، اجتناب از برخورد با موانع توسط ربات‌ها است. این محدودیت یکی از معیارهای مهم در مسائل مربوط به مسیریابی ربات‌ها است. انتخاب الگوریتم مناسب جهت مسیریابی و پرهیز از برخورد با موانع در محیط به‌هنگام با توجه به ساختار مسئله، تأثیر به سزایی بر روی عملکرد کل سیستم و در نتیجه بهینه نمودن معیارهای مختلف دارد.

پرهیز از برخورد ربات‌ها با یکدیگر: از دیگر محدودیت‌هایی موجود در محیط واقعی که در مسائل چند رباتی همواره مطرح است اجتناب از برخورد ربات‌ها با یکدیگر است. این محدودیت که مربوط به ایجاد هماهنگی مناسب در بین ربات‌ها است لازم است که در طول اجرای الگوریتم مدنظر قرار گیرد تا مسئله‌ی شبیه‌سازی مورد نظر برآوردی صحیح از مسائل محیط واقعی باشد.

محدودیت‌های مرتبط با تخصیص کار: مسئله‌ی تخصیص کار در بین ربات‌های چندگانه جزء مسائل NP-Hard بوده و دارای توابع هدف و محدودیت‌های متعددی است. این محدودیت‌ها از قبیل توانایی/ عدم توانایی انجام یک کار توسط ربات، در نظر گرفتن فاصله بین ربات‌ها و کارها و انتخاب ربات‌های مناسب بایستی در حل مسئله مورد توجه قرار گیرند.

۲-۳- جایگاه مسئله

بر اساس طبقه‌بندی صورت گرفته شده توسط [۱]، از لحاظ سطح دانش ربات‌ها آگاهی نسبی از همدیگر دارند. این آگاهی هم در مرحله‌ی پویش محیط و تخصیص کار و هم در مرحله‌ی مسیریابی ربات‌ها در نظر گرفته شده‌اند. از لحاظ هماهنگی و سازمان‌بندی نیز سیستم مورد نظر به ترتیب هماهنگ و کم متمرکز (حالت بین

متمرکز و توزیع شده) است. از نقطه نظر هماهنگی [۲۴] نیز ربات‌ها دارای هماهنگی نامتمرکز می‌باشند. این نوع هماهنگی ترکیبی از دو حالت هماهنگی اتوماتیک و هماهنگی از طریق ارتباط است و ربات در حالی که با سایر ربات‌ها تعامل دارد بر اساس کنترل محلی‌اش عمل می‌کند و اطلاعات کسب شده از دو منبع را با هم ادغام و استفاده می‌کند که احتمال خطا را کاهش می‌دهد.

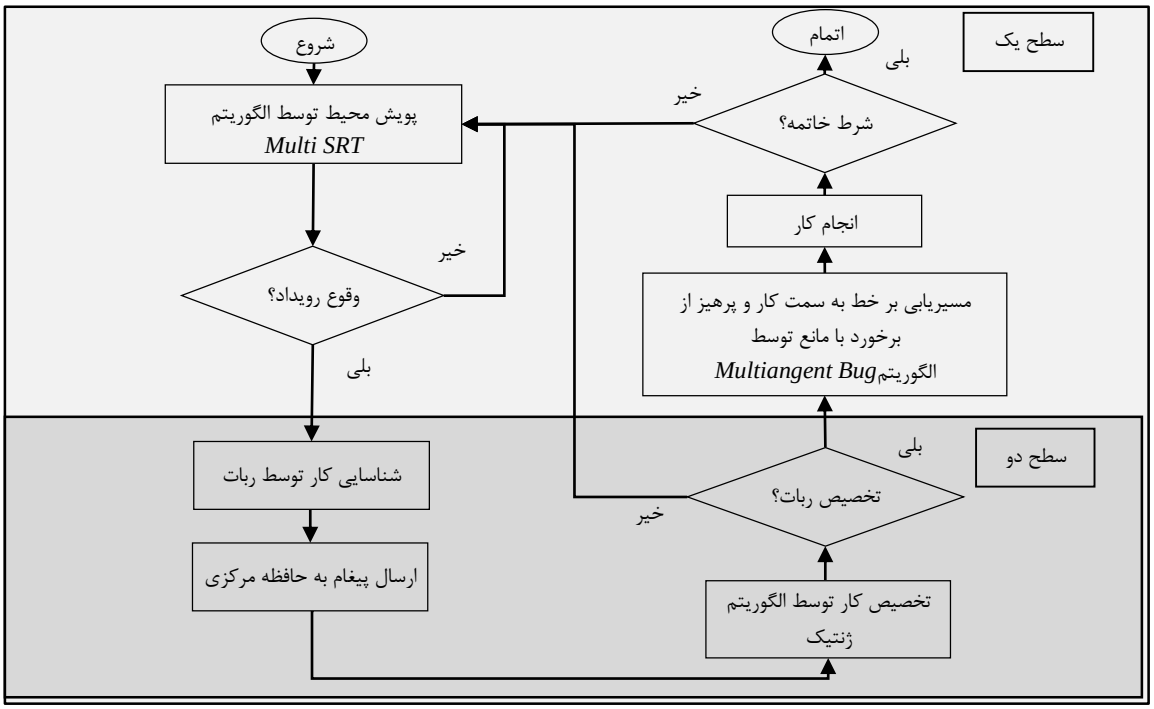
۳- رویکرد پیشنهادی

در این مقاله جهت حل مسئله‌ی پویش محیط و تخصیص کار در ربات‌های چندگانه یک معماری دو سطحی ارائه شده است. اجزای مختلفی از جمله پویش محیط و مسیریابی به هنگام به صورت توزیع شده در سطح اول و تخصیص کار بین ربات‌ها و انجام کارها توسط ربات‌های چندگانه به صورت متمرکز در سطح دوم درنظر گرفته شده‌اند. در قسمت‌های پیش رو به توضیح معماری ارائه شده به همراه مؤلفه‌های مختلف آن پرداخته می‌شود.

حالات مختلف ربات‌ها و کارها: برای سهولت در به‌کارگیری روش‌های مختلف به صورت موازی، حالت‌های مختلفی با در نظرگیری سطوح معماری برای هریک از ربات‌ها و همچنین کارها در نظر گرفته شده‌اند. هر یک از ربات‌ها با توجه به حالت خود روش مورد نظر را اجرا می‌نماید. همچنین رویدادهایی هم تعریف شده‌اند که در صورت وقوع آن‌ها حالت ربات‌ها تغییر یافته و وظیفه‌ی دیگری را انجام می‌دهند. این رویدادها می‌توانند پیدا کردن کار جدید، دریافت و ارسال پیغام از طریق ربات‌ها و همچنین خروجی الگوریتم تخصیص کار باشد که ربات‌ها را جهت انجام کارهای مختلف تخصیص داده و در نتیجه وضعیت ربات‌ها تغییر می‌یابد.

۳-۱- معماری ارائه شده جهت پویش محیط و تخصیص کار

شکل ۱ معماری دو سطحی ارائه شده در این مقاله را برای یک ربات نشان می‌دهد. همانطور که مشاهده می‌شود، در شروع الگوریتم در سطح یک، ربات در وضعیت پویش محیط قرار داشته و با به‌کارگیری رویکرد توزیع شده به جستجوی محیط می‌پردازند. این وضعیت می‌تواند با وقوع یک رویداد تغییر یابد. چنانچه هر یک از ربات‌ها کاری را در محیط پیدا کند، در حافظه‌ی مرکزی ثبت شده و در سطح دوم به صورت متمرکز با توجه به اطلاعات به دست آمده، ربات‌های مناسب جهت انجام کارها تخصیص داده می‌شوند. پس از این مرحله، حالت ربات‌های تخصیص یافته تغییر کرده و وضعیت حرکت به سمت کار بدون برخورد به موانع را به خود می‌گیرند. در این حالت با استفاده از الگوریتم Multi Tangent Bug، که به صورت توزیع شده اجرا می‌گردد، هر یک از ربات‌ها به سمت کارهای مورد نظر حرکت می‌کنند. سایر ربات‌های تخصیص داده نشده نیز حالت پویش قبلی خود را ادامه داده تا اینکه رویداد جدیدی صورت گیرد. پس از رسیدن ربات‌ها به محل کارها، وضعیت آن‌ها دوباره تغییر یافته و تبدیل به حالت انجام کار می‌گردد. پس از اتمام کار نیز ربات‌ها تغییر وضعیت داده و دوباره شروع به پویش مجدد محیط جهت یافتن کار جدید می‌نمایند. جهت فهم بهتر الگوریتم ارائه شده، شبه کد مربوطه در شکل ۲ نشان داده شده است.



شکل (۱): فلوچارت معماری دو سطحی پیشنهادی.

Function Multi_Robot (*n*, *m*)
Inputs: *n* = Number of Robots
m = Number of Tasks

```
Set the initial parameters: Robot_Mode ← Zeros(n, 1); Task_Mode ← Zeros(m, 1)
While iteration ≤ Stopping_Criterion
    For r = 1 to n
        IF Robot_Mode=0
            // The robot is in exploration mode
            [x_new, y_new] ← Multi_SRT(x_old, y_old, Obstacles)
            If [x_new, y_new] − [x_task, y_task] ≤ ε
                Task_Mode(t) ← 1
            End
        ElseIF Robot_Mode = 1
            // The robot moves toward an assigned task
            [x_new, y_new] ← Multi_Tangent_Bug(x_old, y_old, Obstacles)
            If [x_new, y_new] = [x_task, y_task]
                Robot_Mode(r) ← 2 , Task_Mode(t) ← 3
            End
        ElseIF Robot_Mode = 2
            If Time_Task > 0
                [x_Robot, y_Robot] ← [x_task, y_task]
            ElseIf Time_Task = 0
                Robot_Mode(r) ← 3
                Task_Mode(t) ← 4
            End
        End
    End
    Task_index ← Find(Task_Mode = 1)
    Robot_index ← Find(Robot_Mode = 0 OR Robot_Mode = 3)
    [Robot_assign, Task_assign] ← Task_Allocation(Task_index, Robot_index)
    Robot_Mode(Robot_assign) ← 1
    Task_Mode(Task_assign) ← 2
End
```

Level 1 / Level 2

شکل (۲): شبه کد معماری دو سطحی پیشنهادی.

پویش محیط توسط الگوریتم Multi SRT

در این تحقیق جهت پویش محیط به صورت توزیع شده و پرهیز از برخود با موانع از الگوریتمMulti SRT[1] استفاده شده است. این الگوریتم حالت توسعه یافته‌ی الگوریتم SRT است.

الگوریتم SRT از یک روش جستجو که ساختار داده‌ای به نام درخت تصادفی بر اساس حسگر ربات تولید کرده، استفاده می‌کند. روش SRT درختی است که نقشه مسیر فضای پیکربندی آزاد ربات را نمایش می‌دهد. هر یک از گره‌های این درخت نشانگر یک پیکربندی q (محل جدید ربات) عاری از برخورد با موانع است که ربات می‌تواند به عنوان محیط محلی شناخته شده اطراف q توسط حسگر، به آن دسترسی داشته باشد. سپس با توسعه‌ی ساختار داده‌ای از طریق انتخاب تصادفی جهت‌های حرکت و دستیابی به نقطه‌ی جدید (q_{cand}) که در داخل شعاع حسگر قرار دارد، درخت جستجو به تدریج ساخته می‌شود.

در هر تکرار k از الگوریتم، فرایند حس کردن محیط (جمع‌آوری اطلاعات از طریق حسگرها) جهت به دست آوردن ناحیه ستاره‌ای شکل (S) که تخمینی از فضای اطراف ربات قرار گرفته شده در پیکربندی q_{curr} است، انجام می‌گیرد. سپس گره جدید شامل پیکربندی q_{curr} به درخت τ اضافه می‌شود. پس از این مرحله زاویه-ی جستجوی θ_{rand}، توسط تابع RANDOM_DIR به صورت تصادفی انتخاب شده و سپس تابع RAY جهت محاسبه‌ی شعاع حرکت r ، در راستای θ_{rand} فراخوانی می‌شود.

پیکربندی جدید q_{cand} با طی گامی به طول $\alpha.r$ در راستای زاویه θ_{rand} به دست می‌آید. پارامتر $\alpha<1$ تضمین می‌کند که q_{cand} در داخل فضای S قرار گیرد. مقادیر α نزدیک به ۱ موجب تقویت ویژگی اکتشافی الگوریتم شده و مقادیر کم آن نیز حاشیه امنیت الگوریتم را تقویت خواهد کرد. بعد از تولید پیکربندی q_{cand} در داخل ناحیه‌ی S، تابع VALID جهت بررسی شدنی بودن آن فراخوانی می‌شود، که مطابق آن q_{cand} بایستی دارای دو ویژگی زیر

1. Multi Sampling-Based Roadmap of Trees

باشد:

۱) فاصله‌ی آن تا q_{curr} بیشتر از مقدار d_{min} باشد.

۲) در داخل محوطه امن محلی LSR[1] مربوط به سایر گره‌های متعلق به τ قرار نگیرد.

در صورتی که پیکربندی q_{cand} ویژگی‌های فوق را داشته باشد، به عنوان گره جدید به نقشه مسیر اضافه شده و ربات به سمت آن نقطه حرکت می‌کند و این چرخه در تکرارهای بعد نیز اجرا می‌شود.

روش SRT به تنهایی قابل پیاده‌سازی در سیستم ربات‌های چندگانه نبوده و نیاز است تا با ساختار این چنین سیستم‌هایی تطبیق داده شود. اصلاحات صورت گرفته شده در این الگوریتم، مربوط به اعتبارسنجی تابع *VALID* در ارتباط با مکان‌های جدیدی است که به صورت تصادفی ایجاد می‌شوند. این تابع، شدنی بودن پیکربندی جدید q_{cand} را بررسی کرده و در صورت تائید، پیکربندی جدید به درخت τ اضافه می‌گردد. با در نظر گیری ربات‌های متعدد این تابع بایستی همپوشانی ربات‌ها با یکدیگر را بررسی کند. در این صورت پیکربندی جدید q_{cand} جهت شدنی بودن بایستی در داخل محیطهای محلی شناخته شده مربوط به سایر ربات‌های موجود در محیط نیز قرار نگیرد. بدین منظور لازم است ساختار درختی τ برای تمامی ربات‌ها ایجاد شود، تا از طریق آن، همپوشانی ربات‌ها با یکدیگر تشخیص داده شده و از تولید پیکربندی جدید ناشدنی جلوگیری شود.

جهت پیاده سازی الگوریتم SRT به صورت توزیع شده در ربات‌های چندگانه، علاوه بر تغییرات فوق، پردازش‌های صورت گرفته شده به صورت موازی در نظر گرفته می‌شوند. رویکردهای توضیح داده شده در قسمت قبل لازم است برای هر یک از ربات‌ها به صورت جداگانه به کار گرفته شود. این امر نیازمند طراحی چارچوبی جهت اجرای این رویکرد در بین تمامی ربات‌ها به صورت موازی است. در شکل ۴ شبه کد الگوریتم مورد نظر که از روش SRTMulti استفاده می‌کند نشان داده شده است. فرض بر این است که تعداد ربات‌ها برابر N بوده و $R = \{1, 2, \ldots, N\}$ می‌باشد.

تخصیص کار در ربات‌های چندگانه

در سطح دوم مربوط به معماری ارائه شده، پس از پیدا شدن کارها اطلاعات به دست آمده در حافظه مرکزی جمع شده و بر اساس آن تخصیص کار به صورت متمرکز انجام می‌گیرد. این نوع تخصیص بر اساس اطلاعات موجود صورت گرفته و در هنگام وقوع هر رویداد تکرار می‌گردد. از ویژگی‌های این تخصیص موضعی بودن آن است.

مسئله‌ی تخصیص کار در این تحقیق بر اساس دسته‌بندی صورت گرفته توسط [۱۱]، جزو مسائل ST-MR-IA طبقه بندی میشود. در این نوع مسائل، هر کدام از ربات‌ها در یک لحظه قادر به انجام تنها یک کار است. برخی از کارهای موجود در محیط نیز به بیش از یک ربات نیاز دارند. هر یک از کارها بسته به ویژگی‌هایشان ممکن است به یک یا چند ربات نیاز داشته باشند. تخصیص صورت گرفته نیز از نوع تخصیص فوری است، یعنی شرایط محیطی به گونه‌ای است که اطلاعات مختص اکنون بوده و تخصیص باید فوراً صورت گیرد. همچنین اطلاعات موجود به صورت موضعی بوده و نسبت به کل محیط و تمامی کارها آگاهی کامل وجود ندارد.

الگوریتم ژنتیک FGA[2]: با توجه به ماهیت مبتنی بر جمعیت بودن الگوریتم ژنتیک که امکان تولید جواب‌های مختلف کارا را در یک بار اجرا به دست می‌دهد، می‌توان با طراحی ساختار مناسب از این ویژگی به همراه سرعت بالای اجرای الگوریتم بهره‌برداری نمود. در این مقاله الگوریتم تکاملی ژنتیک برای حل مسئله‌ی تخصیص کار در ربات‌های چندگانه با در نظر گیری رویکرد متمرکز ارائه شده است که در زمان‌های ایده‌آل جواب‌هایی با کیفیت مناسب را ارائه می‌نماید. نحوه‌ی نمایش هر یک از جواب‌ها در الگوریتم به کارگرفته شده به صورت زیر است. همچنین کارها و ربات‌های موجود به دو نوع $Type_1$ و $Type_2$ طبقه‌بندی شده‌اند که هر یک ویژگی‌های خاصی را دارا هستند.

$$Robot\begin{cases}Type_1: r_1, r_3, r_4, r_5, r_8\\ Type_2: r_2, r_6, r_7\end{cases} \quad Task\begin{cases}Type_1: t_2, t_4\\ Type_2: t_1, t_3\end{cases} \qquad (۱)$$

۱	۲	۳	۴	۵	۶	۷	۸
۳	۰	۱	۱	۰	۲	۴	۳

شکل(۳): نحوه نمایش کروموزوم

مطابق شکل ۳ طول کروموزوم برابر با تعداد کارهای مورد نظر است. اعداد نوشته شده در داخل هر یک از سلول‌ها بیانگر کارتخصیص یافته است. عدد صفر بدین معنی است که ربات مربوطه به کار خاصی تخصیص نیافته و وضعیت آن در حالت پویش محیط است.

در شکل ۳، تعداد کارهای موجود ۴ عدد است که بایستی توسط ۸ عدد ربات انجام گیرند. تخصیص نمایش داده شده بدین صورت است که ربات‌های ۱ و ۸ به کار سوم، ربات‌های ۲ و ۵ در حال جستجوی محیط، ربات‌های ۳ و ۴ به کار اول و ربات‌های ۶ و ۷ به ترتیب به کارهای دوم و چهارم تخصیص یافته‌اند. لازم به توضیح است که کارهای اول و سوم به دو ربات نیاز دارند. همچنین در تخصیص به نوع ربات‌ها و کارها نیز باید توجه شود. به طور مثال در تخصیص فوق، کار اول جهت انجام به دو ربات ۳ و ۴ نیازمند است، ولی همین کار توسط ربات ۲ به تنهایی نیز می‌تواند انجام گیرد. کارهایی که از نوع اول می‌باشند می‌تـوانند توسط همه‌ی ربات‌ها انجام گیرند ولی بهتر است جهت صرفه‌جویی در هزینه توسط ربات‌های نوع ۱ انجام شوند. کارهای نوع ۲ نیز جهت انجام به یک ربات از نوع دوم و یا دو ربات از نوع اول نیاز دارند که این تخصیص با توجه به شرایط مسئله صورت می‌پذیرد.

همچنین در این الگوریتم عملگر تقاطع تک نقطه‌ای مورد

1. Local Safe Region

2. Fast Genetic Algorithm

استفاده قرار گرفته است. عملگر جهش مورد استفاده نیز بدین صورت است که دو تا از کارهایی که به ربات‌های متفاوت تخصیص داده شده‌اند به صورت تصادفی انتخاب شده و با یکدیگر تعویض می‌شوند.

پس از به‌کارگیری عملگرهای تقاطع و جهش جهت ایجاد جواب‌های جدید، ممکن است جواب‌های به دست آمده نشدنی باشند. به طور مثال ممکن است یک ربات از نوع اول به تنهایی به کاری از نوع دوم تخصیص یابد و یا اینکه ربات‌های تخصیص یافته به یک کار بیشتر از نیاز آن باشند. در این صورت با استفاده از تابع اصلاح، جواب‌های نشدنی اصلاح می‌شوند. در ادامه عملکرد تابع اصلاح توضیح داده می‌شود.

تابع اصلاح: در این رویکرد کارها بر اساس مطلوبیت و پنجره‌ی زمانی که دارند به صورت نزولی مرتب می‌شوند. تابع اهمیت کارها به صورت زیر محاسبه می‌شود.

$$Importance(Task_j) = \alpha \cdot value(Task_j) + \frac{\beta}{Time(Task_j)} \quad (۲)$$

هرچقدر کارها دارای مطلوبیت بیشتر و همچنین پنجره‌ی زمانی کمتری باشند، اولویت بالاتری خواهند داشت. پس از مرتب نمودن کارها، بررسی شدنی بودن آن‌ها به ترتیب اولویتشان انجام می‌گیرد. برای تعیین شدنی بودن، تعداد و نوع ربات‌های تخصیص یافته به کار مورد نظر بررسی می‌شوند. حالات گوناگون نشدنی بودن یک جواب به همراه نحوه‌ی اصلاح آن‌ها در ادامه ذکر می‌شود:

تعداد ربات‌های تخصیص یافته به یک کار بیش از نیاز آن باشند: به صورت تصادفی تعداد ربات‌های اضافی حذف شده و وضعیت آن‌ها به صورت آزاد (تخصیص نیافته) در نظر گرفته می‌شود تا جهت تخصیص به کارهای بعدی مورد استفاده قرار گیرند.

تعداد ربات‌های تخصیص یافته به یک کار کمتر از نیاز آن باشند: از بین سایر ربات‌ها به صورت تصادفی و با توجه به نوع آن‌ها ربات‌های مورد نیاز جهت تخصیص انتخاب می‌شوند.

با توجه به رتبه‌بندی کارها بر اساس اولویت و اصلاح آن‌ها بر اساس این رتبه بندی، اصلاحات فوق عملی بوده و در پایان جواب شدنی به دست می‌دهد. همچنین در صورتی که هر یک از ربات‌ها به کاری تخصیص نیابند، بدین معنا است که ربات مورد نظر وضعیت پویش و جستجوی خود را حفظ خواهد نمود.

Function Multi_SRT(q_{init} ,$[x,y]_{obstacles}$, n)
Input: q_{init} = The first configuration of the Robot
$[x,y]_{obstacles}$ = The location of the Obstacles
n = Number of Robots

1. Set the initial parameters: α , d_{min} , I_{max} //I_{max} : The Maximum number of trials to generate new configuration
2. **For** $k = 1 : K_{max}$
3. **For** $i = 1$ to n
4. $q_{curr} = q_{init}$; $S_i \leftarrow$ PERCEPTION (q_{curr}) //Sensor data are gathered to obtain regionS_i
5. Add (τ_i , (q_{curr} , S_i)) , $j \leftarrow 0$; // The q_{curr} and associated S added to tree τ
6. **While** $j \leq I_{max}$
7. **For** $j = 1$ to I_{max}
8. $\theta_{rand} \leftarrow$ RANDOM_DIR ; $r_i \leftarrow$ RAY (S_i , θ_{rand})
9. $q_{cand} \leftarrow$ DISPLACE (q_{curr} , θ_{rand} , α , r_i)
10. **If** VALID (q_{cand} , d_{min} , τ_i , $\tau_{j=1:N\,,\,J\neq i}$) = 1
11. $q_{curr} \leftarrow q_{cand}$
// Move to the new q_{cand}
12. **Break While**
13. **End**
14. $j \leftarrow j+1$
15. **End**
16. $q_{curr} \leftarrow q_{curr}.\ parent$
//Back to the pervious q_{curr}
17. **End**
18. $i \leftarrow i+1$
19. **End**
20. $k \leftarrow k + 1$
21. **End**

شکل (۴): شبه کد الگوریتم Multi SRT

این کار به دلیل اینکه ممکن است در آینده کار دیگری با اولویت بالاتر در محیط ظاهر شود منطقی به نظر می‌رسد و در این صورت ربات‌های تخصیص نیافته، به کار جدید کشف شده تخصیص خواهند یافت. همچنین عدم تخصیص ربات به یک کار با توجه به تابع هدف کمینه نمودن هزینه‌ی تخصیص، منطقی است.

بدنه اصلی الگوریتم ژنتیک: در ابتدا جمعیت اولیه P_0 (به تعداد P جواب) به صورت تصادفی ایجاد شده و سپس جواب‌های دوتایی والد به صورت تصادفی انتخاب شده و با استفاده از عملگر تقاطع و جهش جمعیت جدید Q_0 به تعداد P ایجاد می‌گردد. در ادامه جمعیت R_0 به اندازه‌ی $2P$ از ترکیب P_0 و Q_0 ایجاد می‌شود.

پس از ایجاد جمعیت $R_t = Q_t \cup P_t$، جمعیت R_t بر اساس تابع برازندگی $F = w_1 f_1 + w_2 f_2 + w_3 f_3$ بصورت نزولی مرتب می‌شود. سپس P جواب اول به عنوان جمعیت جدید P_{t+1} انتخاب می‌شود. در شکل ۵ شبه کد مربوط به الگوریتم ژنتیک ارائه شده است.

توابع هدف الگوریتم ژنتیک: در الگوریتم ژنتیک ارائه شده تابع هدف مورد نظر جمع وزنی سه تابع برآزندگی می‌باشد که در جدول (۱) توضیح داده شده‌اند. هر یک از این توابع به صورت جداگانه در الگوریتم ژنتیک به ازای هر جواب موجه محاسبه شده و جمع جبری به دست آمده مبنای تخصیص کار قرار می‌گیرد.

تنظیم پارامترهای الگوریتم ژنتیک: یکی از مسائل مهم در طراحی الگوریتم ژنتیک تنظیم و انتخاب مقادیر مناسب برای هر یک از پارامترها می باشد. انتخاب مقادیر مناسب باعث کارایی و عملکرد بهتر الگوریتم شده و در نهایت می‌تواند منجر به ارائه جواب‌های بهینه گردد. روش طراحی آزمایشات یکی از بهترین روش‌های تنظیم پارامتر می‌باشد که مورد استفاده بسیاری از محققین قرار گرفته است. در این مقاله روش تاگوچی به عنوان یکی از روش‌های DOE جهت تنظیم پارامترهای مورد استفاده در الگوریتم ژنتیک، استفاده شده است. جهت دستیابی به مقادیر بهتر پارامترهای مورد نظر مطابق جدول (۲) برای هر یک از پارامترها ۳ سطح درنظر گرفته شده است.

با توجه به اجرای آزمایشات در شکل ۶ مشاهده می‌شود که عملگر تقاطع در سطح سوم خود یعنی ۰/۹۵ بیشترین مقدار تابع هدف را به دنبال دارد. همچنین برای عملگر جهش نیز سطح دوم یعنی مقدار ۰/۳ مطلوبیت بیشتری دارد. برای مقدار جمعیت اولیه و همچنین تعداد تکرار الگوریتم نیز به ترتیب مقادیر ۵۰ کروموزوم و ۴۰ تکرار در سطوح دوم و سوم در نظر گرفته می‌شوند.

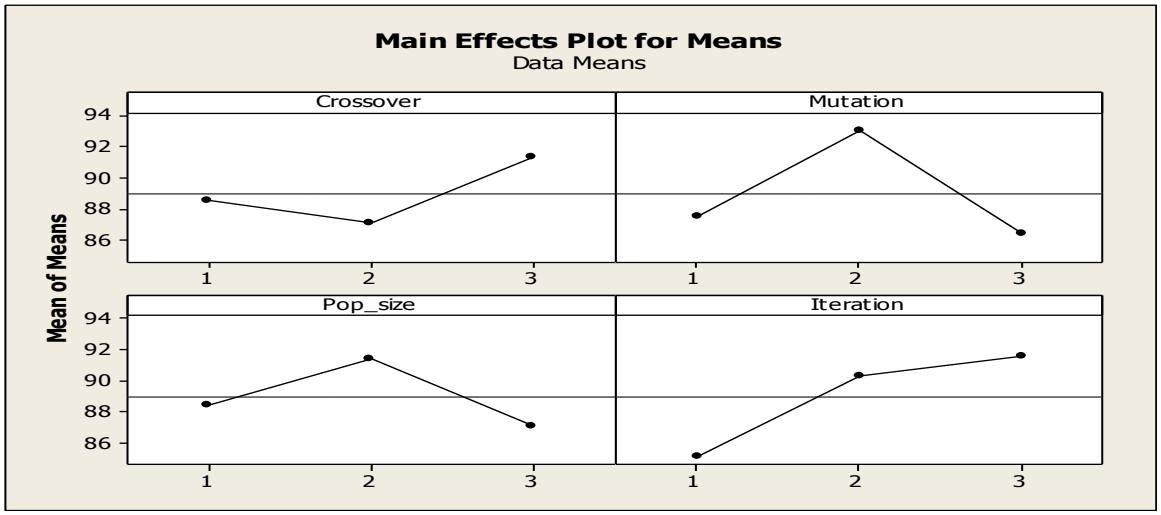
Function Fast Genetic Algorithm(R , T)
Input: R = The Robots
T = The Tasks

1. Set the initial parameters: $P_0 \leftarrow$ rand (*Population_Size* , *number of Tasks*);
   $R_0 \leftarrow$ zeros(2 × *Population_Size* , *number of Tasks*)
2. **While** (*Generation* ≤ *Generation_Size*)
3. **For** i =1 to *Population_Size* //Step size = 2
4. (S_1^1, S_2^1)← Select_Random (P_i) //Randomly select two chromosomes from P_i
5. S_1^2, S_2^2← Crossover (S_1^1, S_2^1)
6. S_1^*, S_2^*← Mutation (S_1^2, S_2^2)
7. (S_1^*, S_2^*) → Q_i (i , i+1) //Insert Offspring into the new population Q_i
8. **End**
9. $R_i = P_i \cup Q_i$
10. R_i^* = Sort(R_i , F) //$F= w_1 \cdot f_1 - w_2 \cdot f_2 - w_3 \cdot f_3$
11. $P_{i+1} = R_i^*$ (1 to *Population_Size*)
12. **End**
13. Report the best solution as a final solution

شکل (۵): شبه کد الگوریتم ژنتیک ارائه شده

جدول (۱): توابع برازندگی الگوریتم ژنتیک ارائه شده.

توابع برازندگی	شرح توابع
ارزش کارها	کارهای با ارزش بالاتر دارای اولویت بالا جهت تخصیص به ربات‌ها می باشند.
مقدار مسافت	تخصیص کار باید به صورتی باشد تا ربات‌ها با طی کمترین مسافت ممکن کارهای موجود را انجام دهند.
ارزش ربات‌ها	در تخصیص صورت گرفته ابتدا باید ربات‌هایی با ارزش پایین تخصیص یابند تا در صورت پیدا شدن کار جدید ربات‌های تخصیص نیافته بتوانند آن کار را انجام دهند.



شکل (۶): نتایج حاصل از اجرای آزمایشات طراحی شده به روش تاگوچی

جدول (۲): جدول طراحی آزمایشات تاگوچی و بهترین مقادیر به‌دست آمده برای هر پارامتر (نمایش داده شده با زیرخط)

پارامتر	سطح ۱	سطح ۲	سطح ۳
احتمال اجرای عملگر تقاطع (P_c)	۰/۷۵	۰/۸۵	**۰/۹۵**
احتمال اجرای عملگر جهش (P_m)	۰/۲۰	**۰/۳۰**	۰/۴۰
اندازه جمعیت اولیه (کروموزوم)	۴۰	**۵۰**	۶۰
شرط خاتمه (تکرار)	۲۰	۳۰	**۴۰**

مدل‌سازی ریاضی تخصیص کار: با توجه به مسئله تخصیص کار به‌وجود آمده در این رویکرد، مدل ریاضی تخصیص فوق به صورت زیر می‌باشد:

$$F = \max\left(w_1 \cdot f_1 + w_2 \cdot f_2 + w_3 \cdot f_3\right) \quad (۳)$$

$$f_1 = \sum_{j=1}^{m} y_j \, u_j \quad (۴)$$

$$f_2 = \sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij} d_{ij} \quad (۵)$$

$$f_3 = \sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij} v_{ij} \quad (۶)$$

$$st: \quad 0 \le \sum_{j=1}^{m} x_{ij} \le 1 \qquad \forall i = 1:n \quad (۷)$$

$$1 \le \sum_{i=1}^{n} x_{ij} \le 2 \qquad \forall j = 1:m \quad (۸)$$

$$y_j = \left(\bigcup_{i=1}^{n} x_{ij}\right) \qquad \forall j = 1:m \quad (۹)$$

$$x_{ij} \in \{0,1\} \quad , \quad y_j \in \{0,1\} \quad (۱۰)$$

در مدل فوق y_j متغیر دودویی است که نشان دهنده انجام کار j-ام می‌باشد. در صورتی که رباتی به کار j-ام تخصیص یابد متغیر فوق ۱ و در غیر اینصورت صفر خواهد بود. u_j نشان دهنده مطلوبیت انجام کار j-ام است، و متغیر x_{ij} نشانگر تخصیص ربات i-ام به کار j-ام می‌باشد. این متغیر نیز دودویی بوده و در صورت تخصیص ربات i-ام به کار j-ام برابر ۱ و در غیر این صورت صفر می‌باشد. لازم به توضیح است که تعداد ربات ها n و تعداد کارها m در نظر گرفته شده است. معادله (۳) تابع هدف کلی را بیان می‌دارد که شامل جمع جبری سه تابع دیگر است که بایستی بیشینه گردند: در تابع f_1 میزان مطلوبیت حاصل از انجام کارها بیان می‌گردد. تابع f_2 بیانگر میزان مسافت طی شده توسط ربات‌ها جهت انجام کارها است که بایستی کمینه گردد، که در آن d_{ij} میزان فاصله بین ربات i ام و کار j ام است. تابع f_3 بیان می‌دارد که ابتدا باید ربات‌هایی با ارزش پایین تخصیص یابند. v_{ij} نشان دهنده ارزش ربات i-ام جهت انجام کار j-ام است.

معادله (۷) نشان می دهد که هر ربات در هر لحظه تنها می تواند به یک کار تخصیص یابد، همچنین ممکن است رباتی به هیچ یک از کارها تخصیص داده نشود. تعداد ربات‌های تخصیص یافته به هر کار می‌تواند یک یا دو ربات باشد که در معادله (۸) نمایش داده می‌شود. رابطه بین متغیر y و متغیر x در معادله (۹) نشان داده شده است. در صورتی که هر یک از ربات‌ها به کار j ام تخصیص یابد متغیر y_j برابر ۱ و در صورتی که هیچ رباتی به کار j ام تخصیص نیابد متغیر فوق برابر صفر می‌گردد. معادله (۱۰) نیز بیانگر دودویی بودن متغیرهای x و y است.

مسیریابی بهنگام ربات‌های چندگانه توسط الگوریتم Multi Tangent Bug

در محیط‌های ناشناخته که ربات هیچ گونه اطلاعی از محیط ندارد و یا اطلاعات محدودی از محیط دارد، برای مسیریابی نیاز است که از روش‌های مناسب استفاده شود. روش‌هایی از قبیل رسم دیاگرام‌های ورونوئی، تجزیه سلولی، گراف دیدنگار و روش پتانسیل میدانی به دلیل نیاز به اطلاعات کامل از محیط، در محیط‌های ناشناخته و مسیریابی به هنگام کاربرد چندانی ندارند. از جمله روش‌های مناسب جهت مسیریابی به هنگام، الگوریتم‌های Bug می‌باشند.

الگوریتم‌های Bug1 و Bug2 [۲۵] از ساده‌ترین و قدیمی‌ترین روش‌های مسیریابی ربات بر اساس حسگر می‌باشند. این الگوریتم‌ها ربات را به صورت نقطه که دارای حسگرهایی جهت تشخیص مانع می‌باشند، در نظر می‌گیرند. همچنین می‌توان در این روش دامنه حسگر ربات را صفر در نظر گرفت. هنگامی که دامنه‌ی حسگر ربات غیر صفر در نظر گرفته شود، الگوریتم TangentBug [۲۶] به

عنوان یکی از روش‌های بهبود یافته Bug2 که از اطلاعات به دست آمده از حسگر جهت یافتن کوتاه‌ترین مسیر تا هدف استفاده می‌کند، به کار گرفته می‌شود. الگوریتم‌های Bug و شبیه آن‌ها، روش‌های سر راست و ساده‌ای جهت استفاده می‌باشند که موفقیت آن‌ها در رسیدن به هدف تضمین شده است. دو نوع رفتاری که در این الگوریتم‌ها مورد استفاده قرار می‌گیرد حرکت بر روی خط مستقیم و دنبال کردن لبه مانع است. اساس کار این الگوریتم‌ها بدین صورت است که در شروع، ربات در راستای خط واصل به هدف به طور مستقیم حرکت نموده تا اینکه به هدف برسد و یا اینکه مانعی را بر سر مسیر خود تشخیص دهد. در این حالت ربات محیط مانع را پیمایش کرده تا اینکه دوباره شرایط حرکت مستقیم به هدف فراهم شود.

در الگوریتم‌های Bug1 و Bug2 فرض بر این است که حسگرهای ربات از نوع لمسی بوده و در صورت تماس با مانع قادر به تشخیص مانع می‌باشند، در صورتی که در روش TangentBug فرض می‌شود که ربات دارای حسگر است که دامنه‌ی آن از صفر بزرگ‌تر بوده و قبل از تماس با مانع می‌تواند آن را تشخیص دهد. روش کار در این الگوریتم به این صورت است که ربات به طور مستقیم به سمت هدف حرکت می‌کند تا اینکه به مانعی برخورد کند. در این صورت ربات به سمت نقطه‌ی برخورد بر روی مانع که دارای کمترین هزینه ابتکاری است، حرکت می‌کند. تابع هزینه ابتکاری حاصل جمع فاصله‌ی بین ربات و نقطه‌ی روی مانع که توسط ربات تشخیص داده شده است و فاصله‌ی بین آن نقطه و هدف است. در صورتی که تابع هزینه شروع به افزایش نماید ربات حالت حرکت مستقیم به سمت هدف را به اتمام رسانده و حالت حرکت بر روی لبه‌ی مانع را آغاز می‌کند. این حالت تا موقعی که فاصله‌ی مستقیم بین ربات و هدف کمتر از فاصله‌ی بین نقطه‌ی آغازین حرکت بر روی لبه و هدف باشد ادامه خواهد یافت. پس از اتمام این حالت ربات دوباره حالت حرکت مستقیم به سمت هدف را آغاز کرده تا اینکه به هدف برسد. جهت پیاده‌سازی الگوریتم Tangent Bug در ربات‌های چندگانه لازم است تا ساختار این الگوریتم ماهیت توزیع شده داشته و با استفاده از اطلاعات محلی که ربات‌ها کسب می‌کنند مسیریابی ربات صورت گیرد. در این روش حافظه‌ی مرکزی وجود نداشته و ربات‌ها به صورت مستقل و با استفاده از اطلاعات محدود خود به سمت کارهای یافت شده حرکت نموده و از برخورد با موانع اجتناب می‌ورزند. در شکل ۷ شبه کد مربوط به الگوریتم Multi Tangent Bug ارائه شده است.

۴- نتایج شبیه سازی

در این قسمت نتایج حاصل از شبیه سازی روش پیشنهادی بر روی مسائل آزماینده مختلف گزارش می‌شود. همانطور که در شکل ۸ مشاهده می‌شود محیط کاری دارای چهار مانع می‌باشد که همراه با لبه‌های محیط کاری با رنگ آبی نمایش داده شده‌اند. هر یک از ربات‌ها در حین پویش محیط و همچنین حرکت به سمت کارها بایستی از برخورد با موانع و لبه‌های محیط کاری اجتناب ورزد. ابعاد محیط کاری نیز (cm) ۱۰×۱۰ در نظر گرفته شده است.

Function Multi_Tangent_Bug (R, q_{goal} , x, m)
Input: R_i = A range of robot's sensor i
q_{goal} = The location of the target
x_i= The location of the robot i
m = Number of Robots

For i = 1 to m
  $T_i \leftarrow$ The point which intersect the $d(x_i , q_{goal})$ //$T_i \in \{R_i\}$
  $O_j \leftarrow$ The point which intersect the (R_i , WO_j) //WO_j is an obstacle
  While $x \neq q_{goal}$
    $x \leftarrow n \in (T , O_i)$ with minimizes $d(x , n) + d(n , q_{goal})$
    If $d(x_i , n_i) + d(n_i, q_{goal}) < d(x_{i+1}, n_{i+1}) + d(n_{i+1}, q_{goal})$ //The robot detects a "local minimum"
      $\theta \leftarrow \{ y \in \partial WO_f : \lambda x + (1-\lambda) y \in Q_{free} \forall \lambda \in [0,1] \}$
      $d_{reach} \leftarrow min\ d(q_{goal} , c)\ ;\ c \in \theta$
      while $d_{followed} \leq d_{reach}$
        $x_i \leftarrow n \in (O_j)$ //Chose a boundary following direction
        update d_{reach} , $d_{followed}$ and (O_j)
      End
    End
  End
End

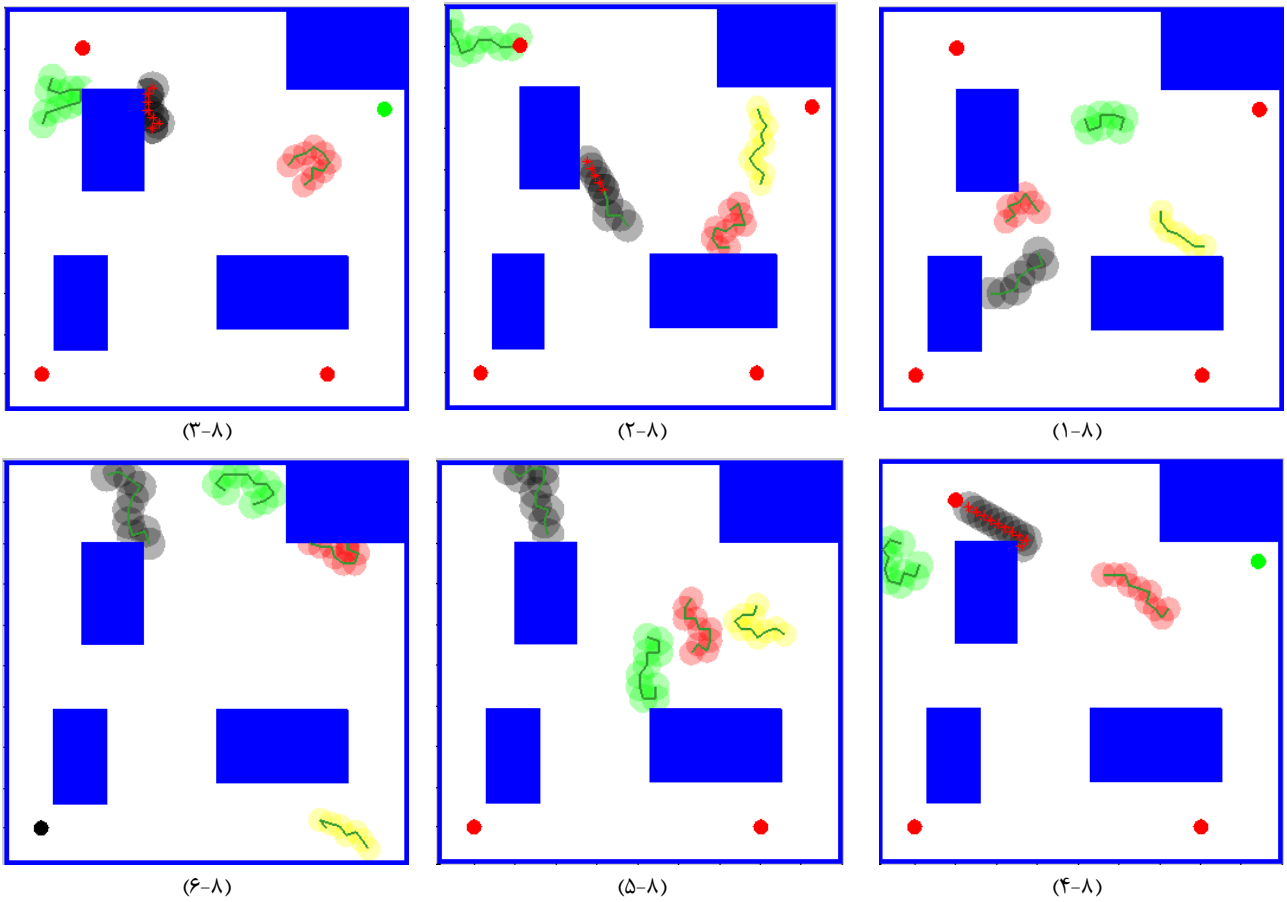
شکل (۷):شبه کد مربوط به الگوریتم Tangent Bug

کارهای موجود در محیط نیز به شکل دایره های قرمز رنگ نمایش داده می‌شوند. هر یک از کارها دارای پنجره زمانی می‌باشد که با اتمام زمان فوق از ارزش کارها کاسته شده تا این‌که به صفر برسد. ربات‌هایی که پنجره زمانی آنها به اتمام رسیده است با رنگ زرد و ربات‌های که دارای ارزش صفر می‌باشند با رنگ سیاه نمایش داده می‌شوند.

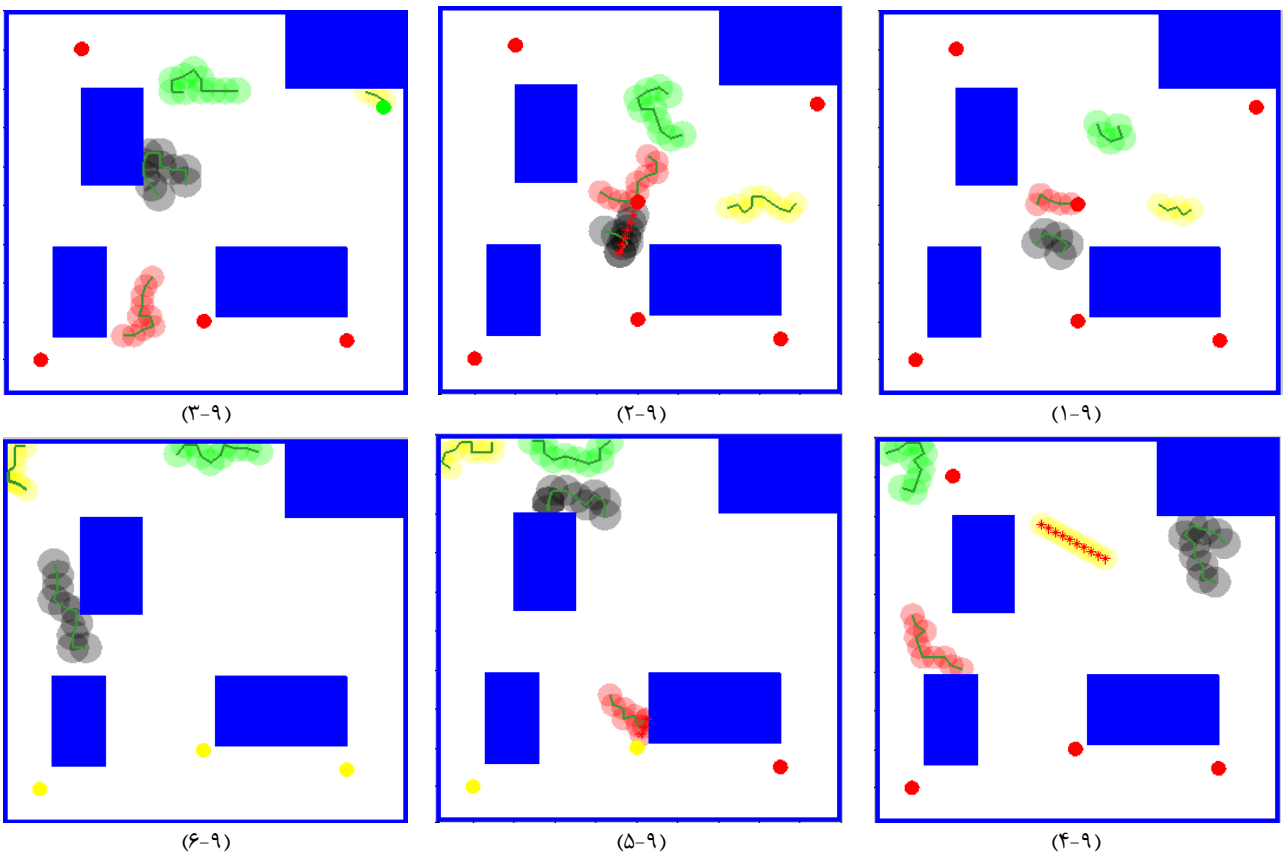
همچنین بسته به حالات مختلفی که هر یک از کارها در ارتباط با ربات‌ها می‌توانند داشته باشند با رنگ‌های مختلفی نشان داده می‌شوند. چنانچه هر یک از کارها در محیط در وضعیت ناشناخته قرار داشته باشد با رنگ قرمز و در صورتی که توسط یک یا چند ربات انجام گیرد رنگ آن به سبز تغییر می‌یابد. پس از اتمام کارها توسط هر یک از ربات‌ها کارهای موجود از محیط کاری حذف خواهند شد.

ربات‌های استفاده شده در شبیه‌سازی مسائل آزماینده به صورت ربات‌های دایره‌ای با شعاع حسگرهای متفاوت درنظر گرفته شده‌اند. هر یک از ربات ها دارای حافظه‌های محدودی هستند که مکان‌های قبلی پویش شده در آن حافظه نگه‌داری می‌شود. پس از حذف هر یک از مکان‌های بازدید شده قبلی از حافظه ربات، مکان فوق به صورت یک مکان جدید درنظر گرفته می‌شود که بایستی مجددا بازدید شود. هر یک از ربات‌های دارای سرعت و توانایی های متفاوتی می‌باشند که این ویژگی‌ها بایستی در تخصیص ربات به کارها درنظر گفته شود. تعداد سنسورهای موجود بر روی هر یک از ربات‌ها برابر ۲۰ عدد درنظر گرفته شده است. بدین مفهوم که در هر حرکتی که ربات‌ها جهت پویش و یا حرکت به سمت کارها انجام می‌دهند هر یک از سنسورها که در فاصله‌ی زاویه‌ای 18° نسبت به هم قرار دارند اشعه‌ای را به طول مشخص جهت تشخیص مانع ساطع می‌کنند. این اشعه‌ها با رنگ‌های متفاوت در شکل‌های ۸ الی ۱۰ نشان داده شده است.

همانطور که در شکل (۸-۱) مشاهده می‌شود در ابتدا تمامی کارها ناشناخته می‌باشند که با رنگ قرمز نشان داده شده‌اند. پس از اینکه ربات‌ها در محیط به جستجو پرداختند یکی از کارها توسط ربات سبز پیدا شده و در نتیجه الگوریتم ژنتیک جهت تخصیص ربات مناسب اجرا شده که ربات سیاه ربات منتخب می‌باشد که به ترتیب در اشکال (۸-۲)، (۸-۳) و (۸-۴) به سمت کار مورد نظر حرکت می‌نماید. پس از پویش محیط و تخصیص ربات به کارهای یافت شده کارهای موجود از محیط حذف می‌شود و در نهایت مطابق شکل (۸-۶) پس از اتمام زمان اجرای الگوریتم تنها یک کار کشف نشده در محیط وجود دارد که به دلیل پایان یافتن پنجره زمانی و صفر شدن ارزش آن با رنگ سیاه نمایش داده شده است. از دیگر حالات کارها می‌توان به حالتی اشاره نمود که در آن پنجره زمانی به اتمام رسیده و مطلوبیت آن شروع به کاهش می‌کند که در شکل (۹-۵) با رنگ زرد نمایش داده می‌شود.



شکل(۸): نتایج حاصل از شبیه سازی مسئله ۱



شکل(۹): نتایج حاصل از شبیه سازی مسئله ۲

نتایج حاصل از شبیه‌سازی الگوریتم پیشنهادی و همچین اجرای الگوریتم ژنتیک بر روی مسائل آزماینده مختلف در قالب جداول و نمودارهای مختلف در بخش بعد نمایش داده شده و تحلیل‌های لازم ارائه شده است.

۵- مقایسات و تحلیل‌ها

به منظور اعتبارسنجی الگوریتم، سناریوهای مختلفی طراحی گردیده‌اند. توسعه این سناریوها به طوری بوده است که از لحاظ تعداد ربات‌ها (N) و کارها (M) با یکدیگر متفاوت باشند. در هر یک از سناریوها پس از ارائه‌ی نتایج به دست آمده، فاصله‌ی هر یک از نتایج با کران بالا نیز گزارش می‌شوند. تأثیر پارامترهایی نظیر مقدار حافظه‌ی هر یک از ربات‌ها و تعداد تکرار الگوریتم نیز بر روی عملکرد الگوریتم ارائه می‌شوند. همچنین جهت تحلیل و اعتبارسنجی الگوریتم ژنتیک ارائه شده نتایج آن با الگوریتم ژنتیک NSGA-II مقایسه شده و تحلیل های لازم انجام می‌گیرد.

۵-۱- نتایج حاصل از اجرای الگوریتم ژنتیک

در راستای اعتبارسنجی الگوریتم ژنتیک ارائه شده، نتایج به دست آمده با الگوریتم ژنتیک NSGA-II مقایسه می‌گردد. الگوریتم NSGA-II از مهم‌ترین الگوریتم‌های ژنتیک چند هدفه می‌باشد که در سال‌های اخیر ارائه شده است[۲۷]. در این الگوریتم با استفاده از رویکرد مرتب‌سازی و حفظ گوناگونی و همچنین در نظرگیری توابع مختلف بهینه سازی، یک عده جواب‌های ناچیره در پایان به دست می‌آیند. شکل‌های ۱۱ و ۱۲ به ترتیب مقدار تابع هدف و زمان اجرای هر دو الگوریتم را نمایش می‌دهد.

مقدار انحراف تابع مطلوبیت الگوریتم FGA از الگوریتم NSGA-II به ازای مسائل آزماینده مختلف در شکل ۱۱ نمایش داده می‌شود. همانطور که مشاهد می‌شود مقدار انحراف ناچیز بوده و کیفیت جواب‌های هر دو الگوریتم تقریبا یکسان می‌باشد ولی مطابق شکل ۱۲ زمان اجرای الگوریتم FGA نسبت به NSGA-II به مراتب پایین‌تر می‌باشد که با توجه به ساختار مسئله و معماری ارائه شده و نیاز به زمان پایین در تخصیص کار بین ربات‌ها الگوریتم FGA از کارایی بهتری برخوردار می‌باشد.

۵-۲- نتایج حاصل از اجرای معماری ارائه شده

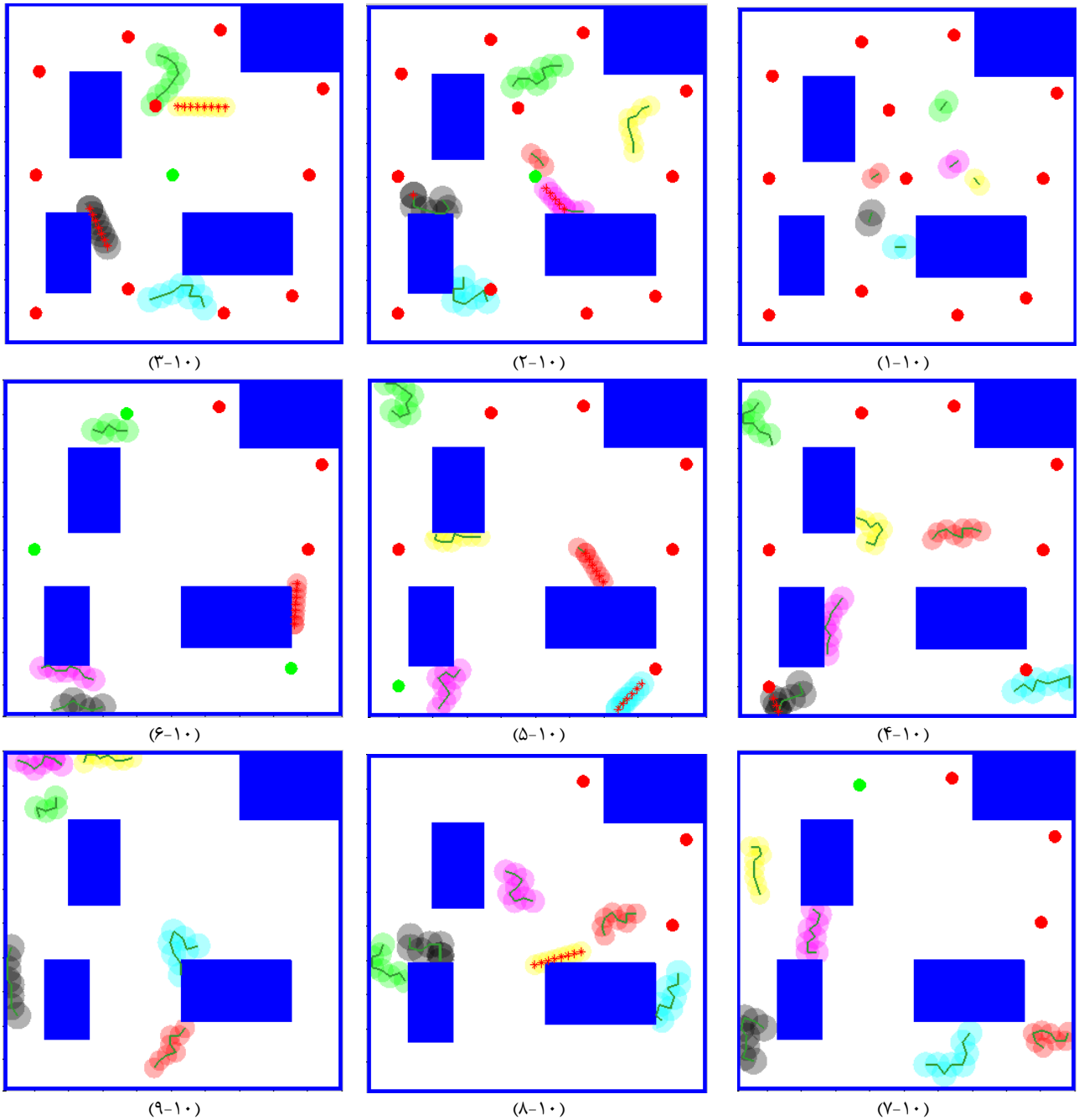
جهت سنجش عملکرد و کارایی معماری ارائه شده، مسائل متنوع با تعداد ربات‌ها و کارهای مختلف به همراه پارامترهای گوناگون در نظر گرفته شده‌اند. جدول (۳) مسائل در نظر گرفته شده جهت حل با معماری پیشنهادی را نشان می‌دهد. همان‌طور که مشاهده می‌شود تعداد ۸ مسئله با انواع مختلف کارها و ربات‌ها و همچنین پنجره‌های زمانی متفاوت در نظر گرفته شده است.

جدول (۴) نتایج حاصل از اجرای معماری ارائه شده برای ۸ مسئله‌ی در نظر گرفته شده را نشان می‌دهد. مقدار حافظه‌ی در نظر گرفته شده برای تمامی ربات‌ها ۱۰ واحد زمانی بوده و نتایج به دست آمده به ازای تعداد تکرار ۴۰۰ گزارش شده‌اند. نتایج گزارش شده در این جدول شامل متوسط مسافت طی شده توسط تمامی

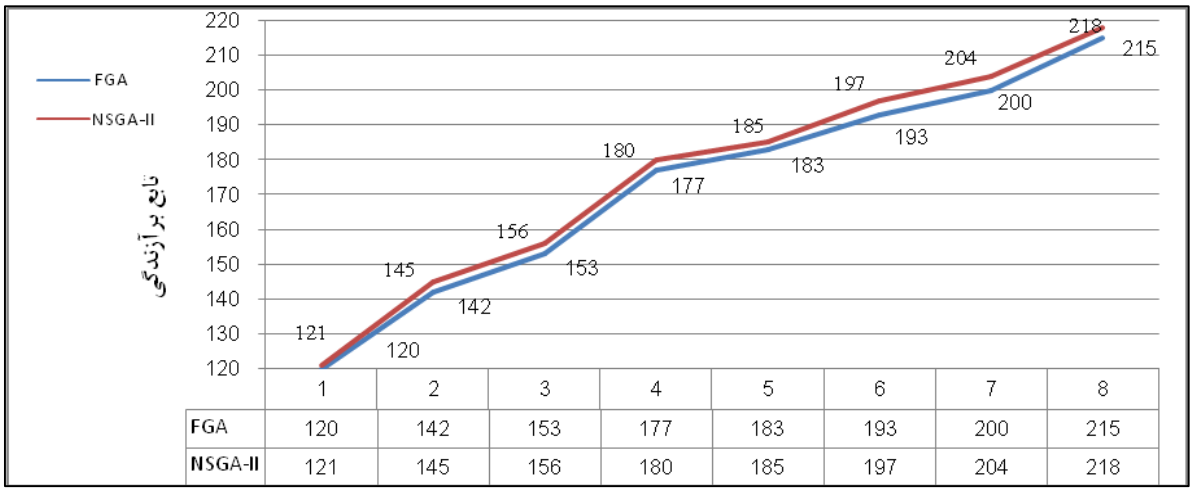
ربات‌ها جهت رسیدن به کارهای مورد نظر، متوسط مقدار مطلوبیت به دست آمده از انجام کارها در مدت زمان مشخص و همچنین درصد فاصله از کران بالا برای مطلوبیت به دست آمده است. شکل ۱۳ نتایج حاصل از اجرای الگوریتم پیشنهادی را در مقایسه با نتایج کران بالای هر یک از مسائل نشان می‌دهد. در این شکل برای هر مسئله کران بالای مجموع ارزش کل کارها به همراه مطلوبیت حاصل از اجرای الگوریتم بر روی آن مسئله نمایش داده می‌شود.

شکل ۱۴ درصد انحراف مطلوبیت به دست آمده از کران بالا را برای سناریوهای مختلف نشان می‌دهد. به طور مثال برای مسئله‌ی اول مقدار فاصله‌ی مطلوبیت به دست آمده تا کران بالا ۲۲٪ و برای مسئله‌ی هشتم برابر ۰/۸٪ است.

درصد فاصله از کران بالا برای مطلوبیت به دست آمده بدین معنی است که ربات‌ها در طول اجرای الگوریتم موفق به کشف و انجام درصد معینی از کارهای موجود در محیط شده‌اند و بقیه‌ی کارها ناشناخته باقی مانده و یا به طور ناقص انجام شده‌اند. به طور مثال عدد ۱۰٪ گویای این واقعیت است که مطلوبیت حاصل از انجام کارهای کشف شده توسط ربات‌ها در طول اجرای الگوریتم، به اندازه ۹۰٪ از مطلوبیت کل کارهای موجود در محیط است. با توجه به ساختار مسئله، این معیار می‌تواند شاخص مناسبی جهت سنجش عملکرد و اعتبارسنجی الگوریتم ارائه شده، باشد.

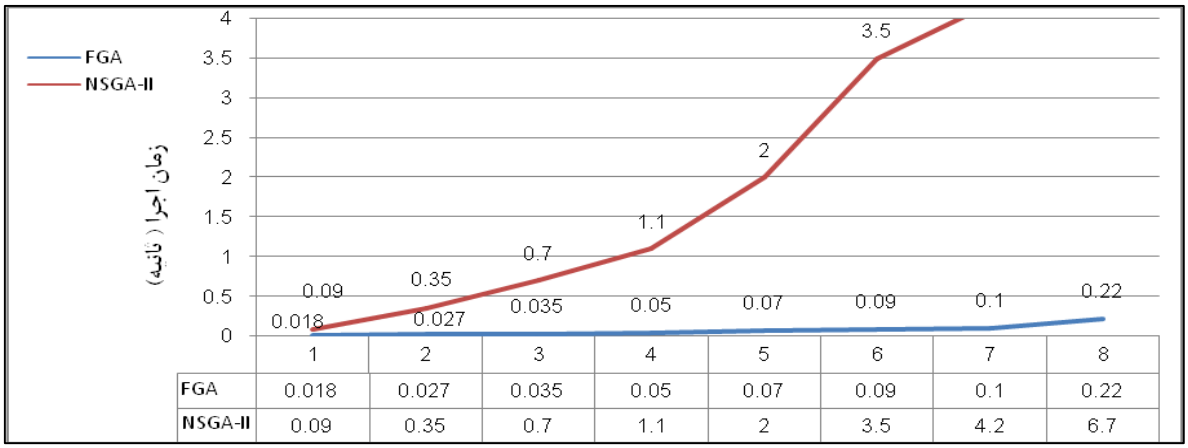


شکل(۱۰): نتایج حاصل از شبیه سازی مسئله ۷



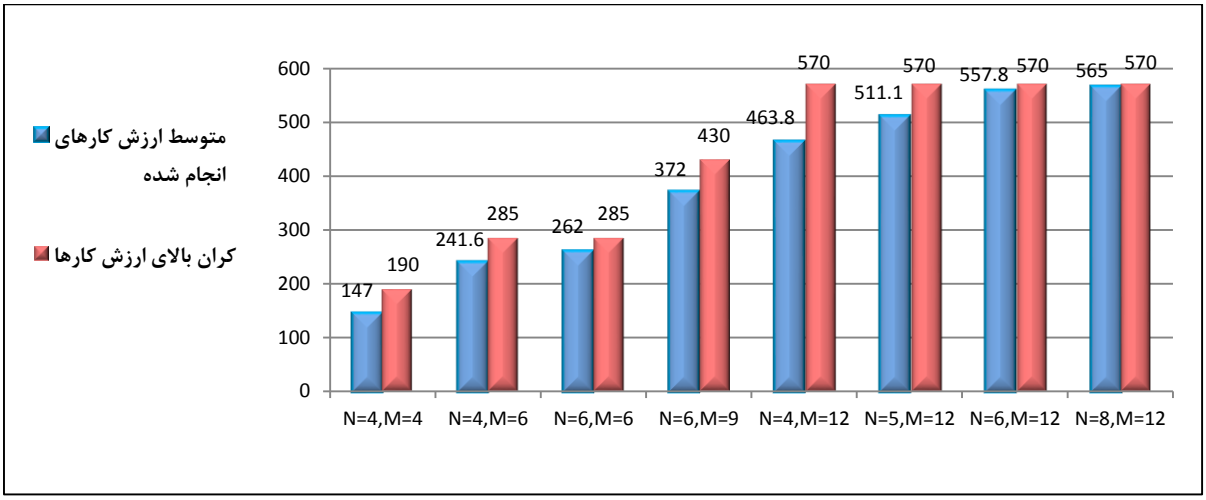
	1	2	3	4	5	6	7	8
FGA	120	142	153	177	183	193	200	215
NSGA-II	121	145	156	180	185	197	204	218

شکل (۱۱):مقایسه نتایج الگوریتم FGA با NSGA-II



	1	2	3	4	5	6	7	8
FGA	0.018	0.027	0.035	0.05	0.07	0.09	0.1	0.22
NSGA-II	0.09	0.35	0.7	1.1	2	3.5	4.2	6.7

شکل(۱۲): مقایسه زمان اجرای الگوریتم FGA با NSGA-II



شکل (۱۳): نتایج حاصل از اجرای معماری پیشنهادی در مقایسه با کران بالای مطلوبیت کل کاره

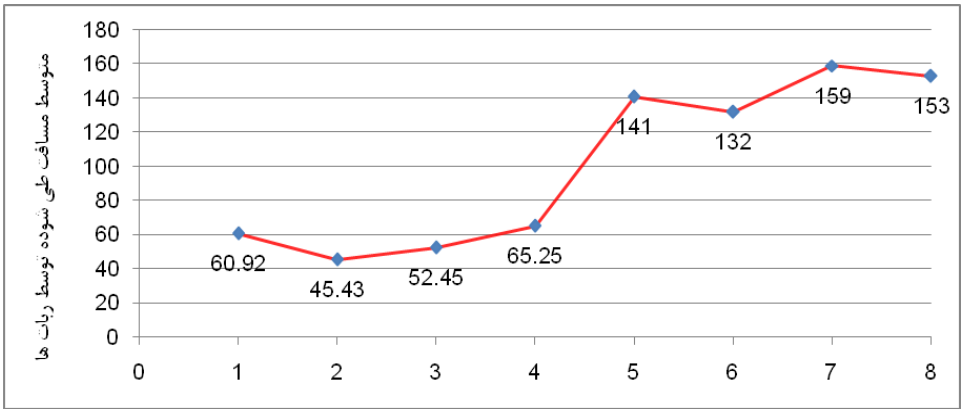
یکی دیگر از خروجی‌های در نظر گرفته شده برای الگوریتم، متوسط مسافت طی شده توسط ربات‌ها جهت رسیدن به اهداف مورد نظر است. الگوریتم مورد نظر بایستی به طوری تخصیص کار بین ربات‌ها و همچنین مسیریابی به هنگام ربات‌ها را انجام دهد که مسافت کل پیموده شده توسط ربات‌ها کمینه گردد. شکل ۱۵ متوسط مسافت طی توسط ربات‌ها برای دستیابی به کارهای موجود در محیط را به ازای سناریوهای مختلف نشان می‌دهد. برای سناریوهای پنجم تا هشتم، با توجه به افزایش تعداد ربات‌ها و همچنین تعداد کارهای موجود در محیط، افزایش مقدار مسافت طی شده توسط ربات‌ها منطقی به نظر می‌رسد.

از دیگر معیارهای تأثیر گذار در عملکرد الگوریتم، تعداد ربات‌های موجود در محیط است. به نظر می‌رسد با توجه به ساختار

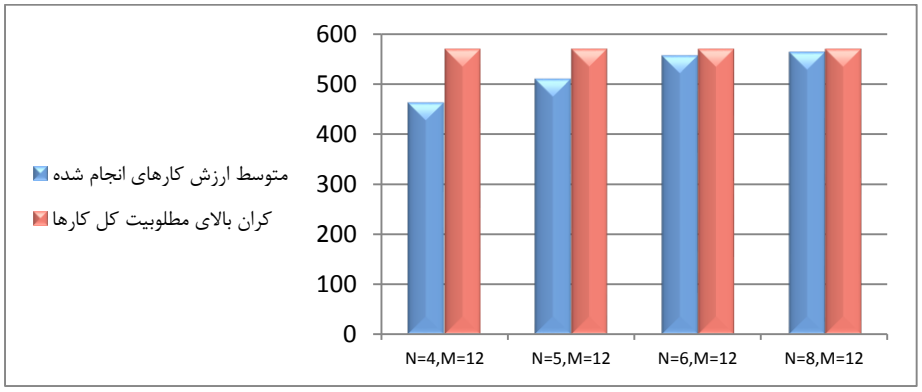
مسئله افزایش تعداد ربات‌ها باعث بهبود عملکرد الگوریتم گردد. جدول (۵) تأثیر افزایش تعداد ربات‌ها در محیط به ازای تعداد ثابت کارها را نشان می‌دهد. در این جدول تعداد کارهای موجود در محیط عدد ثابت ۱۲ و تعداد ربات‌ها متغیر در نظر گرفته شده است. همچنین حافظه‌ی ربات‌ها برای تمامی سناریوها ۱۰ بوده و تعداد تکرارها ۴۰۰ در نظر گرفته شده‌اند. شکل ۱۶ تأثیر افزایش تعداد ربات‌ها بر مقدار درصد انحراف از کران بالا را نمایش می‌دهند.

همان‌طور که در شکل ۱۶ نیز نشان داده می‌شود، با افزایش تعداد ربات‌ها، مطلوبیت به دست آمده افزایش یافته و در نتیجه درصد انحراف از کران بالای مطلوبیت کاهش می‌یابد. با توجه به افزایش تعداد ربات‌ها، می‌توان نتیجه گرفت که نواحی ناشناخته‌ی بیشتری پویش شده و در نتیجه تعداد کارهای بیشتری کشف و انجام می‌شوند که در نهایت منجر به افزایش کارایی الگوریتم می‌شود.

با توجه به شکل ۱۷، در صورتی که در حین اجرای الگوریتم یکی از ربات‌ها خراب شود، کارایی سیستم تغییری نکرده و عملکرد آن همانند حالت بدون خرابی خواهد بود. در صورتی که دو تا از ربات‌ها خراب شوند عملکرد سیستم به میزان ۱/۰۲ درصد کاهش می‌یابد. این کاهش هنگامی که تعداد ربات‌های خراب به عدد ۴ افزایش یابد، برابر ۱۹/۳۸ درصد خواهد بود.



شکل(۱۵): متوسط مسافت طی شده توسط هر یک از ربات‌ها



شکل (۱۶): تاثیر افزایش ربات‌ها بر عملکرد الگوریتم

جدول (۳): مسائل آزماینده طراحی شده

	مسئله	نوع کار	ارزش کارها	پنجره‌ی زمانی کارها
۱	$N=۴, M=۴$	{۱،۲،۱،۲}	{۵۰، ۷۵، ۲۰، ۴۵}	{۲۰۰،۲۲۰،۱۹۰،۱۹۰}
۲	$N=۴, M=۶$	{۱،۲،۱،۲،۱،۱}	{۵۰،۷۵،۲۰،۴۵،۴۰،۵۵}	{۲۰۰،۲۲۰،۱۹۰،۱۹۰،۲۵۰،۱۹۸}
۳	$N=۶, M=۶$	{۱،۲،۱،۲،۱،۱}	{۵۰،۷۵،۲۰،۴۵،۴۰،۵۵}	{۲۰۰،۲۲۰،۱۹۰،۱۹۰،۲۵۰،۱۹۸}
۴	$N=۶, M=۹$	{۱،۲،۱،۲،۱،۱،۲،۱،۱}	{۵۰،۷۵،۲۰،۴۵،۴۰،۵۵،۵۰،۷۵،۲۰}	{۲۰۰،۲۲۰،۱۹۰،۱۹۰،۲۵۰،۱۹۸،۲۲۰،۱۹۰،۱۹۰}
۵	$N=۴, M=۱۲$	{۱،۲،۲،۱،۲،۲،۱،۲،۱،۲،۱،۱}	{۵۰،۷۵،۲۰،۴۵،۴۰،۵۵،۹۰،۵۵،۲۰،۴۵،۴۰،۳۵}	{۲۷۰،۳۰۰،۲۶۰،۲۶۰،۳۱۰،۲۷۰،۲۷۵،۲۶۵،۲۸۶،۲۶۰،۲۶۵،۲۷۰}
۶	$N=۵, M=۱۲$	{۱،۲،۲،۱،۲،۲،۱،۲،۱،۲،۱،۱}	{۵۰،۷۵،۲۰،۴۵،۴۰،۵۵،۹۰،۵۵،۲۰،۴۵،۴۰،۳۵}	{۲۷۰،۳۰۰،۲۶۰،۲۶۰،۳۱۰،۲۷۰،۲۷۵،۲۶۵،۲۸۶،۲۶۰،۲۶۵،۲۷۰}
۷	$N=۶, M=۱۲$	{۱،۲،۲،۱،۲،۲،۱،۲،۱،۲،۱،۱}	{۵۰،۷۵،۲۰،۴۵،۴۰،۵۵،۹۰،۵۵،۲۰،۴۵،۴۰،۳۵}	{۲۷۰،۳۰۰،۲۶۰،۲۶۰،۳۱۰،۲۷۰،۲۷۵،۲۶۵،۲۸۶،۲۶۰،۲۶۵،۲۷۰}
۸	$N=۹, M=۱۲$	{۱،۲،۲،۱،۲،۲،۱،۲،۱،۲،۱،۱}	{۵۰،۷۵،۲۰،۴۵،۴۰،۵۵،۹۰،۵۵،۲۰،۴۵،۴۰،۳۵}	{۲۷۰،۳۰۰،۲۶۰،۲۶۰،۳۱۰،۲۷۰،۲۷۵،۲۶۵،۲۸۶،۲۶۰،۲۶۵،۲۷۰}

جدول (۴): نتایج حاصل از اجرای الگوریتم بر روی مسائل آزماینده، با تعدادتکرار ۴۰۰ و حافظه هر ربات ۱۰

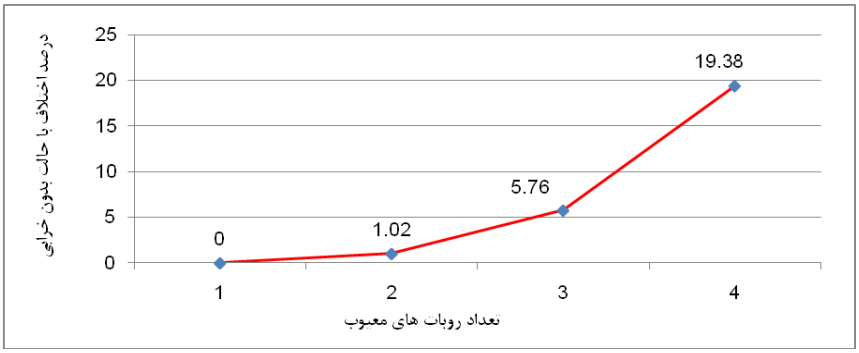
پارامترهای مسئله		متوسط مسافت طی شده توسط ربات‌ها	متوسط ارزش کارهای انجام	ارزش کل کارها	درصد انحراف تا کران بالا
مسئله	نوع ربات	جهت انجام کار (سانتی‌متر × ۱۰)	گرفته شده توسط ربات‌ها		
۱	{۱،۲،۱،۲}	۶۰/۹۲	۱۴۷/۰	۱۹۰	۲۲/۰
۲	{۱،۲،۱،۲}	۴۵/۴۳	۲۴۱/۶	۲۸۵	۱۵/۰
۳	{۱،۲،۱،۲،۱،۱}	۵۲/۴۵	۲۶۲/۰	۲۸۵	۷/۹
۴	{۱،۲،۱،۲،۱،۱}	۶۵/۲۵	۳۷۲/۰	۴۳۰	۵/۰
۵	{۱،۲،۱،۲}	۱۴۱/۰۰	۴۶۳/۸	۵۷۰	۱۸/۰
۶	{۱،۲،۱،۲،۱}	۱۳۲/۰۰	۵۱۱/۱	۵۷۰	۱۰/۰
۷	{۱،۲،۱،۲،۱،۱}	۱۵۹/۰۰	۵۵۷/۸	۵۷۰	۲/۰
۸	{۱،۲،۱،۲،۱،۱،۲،۱،۲}	۱۵۳/۰۰	۵۶۵/۰	۵۷۰	۰/۸

جدول (۵): تاثیر افزایش ربات‌ها بر عملکرد الگوریتم با تعداد تکرار ۴۰۰ و حافظه هر ربات ۱۰

پارامترهای مسئله		متوسط مسافت طی شده توسط ربات‌ها	متوسط ارزش کارهای انجام	ارزش کل کارها	درصد انحراف تا کران بالا
مسئله	تعداد ربات‌ها و کارها	جهت انجام کار (سانتی‌متر × ۱۰)	گرفته شده توسط ربات‌ها		
۵	N = ۴ , M= ۱۲	۱۴۱	۴۶۳/۸	۵۷۰	٪۱۸
۶	N = ۵ , M= ۱۲	۱۳۲	۵۱۱/۱	۵۷۰	٪۱۰
۷	N = ۶ , M= ۱۲	۱۵۹	۵۵۷/۸	۵۷۰	٪۲
۸	N = ۷ , M= ۱۲	۱۵۳	۵۶۵	۵۷۰	٪۰/۸

جدول (۶): تاثیر خرابی ربات‌ها بر عملکرد الگوریتم با تعداد تکرار ۴۰۰ و حافظه هر ربات ۱۰

پارامترهای مسئله		متوسط ارزش کارها	متوسط ارزش کارها	درصد اختلاف
تعداد ربات‌ها و کارها	تعداد ربات خراب شده	(بدون خرابی)	(با در نظر گیری خرابی)	
N = ۹ , M= ۱۲	۱	۵۶۵	۵۶۴	٪ ۰/۰۰
N = ۹ , M= ۱۲	۲	۵۶۵	۵۵۹/۲	٪ ۱/۰۲
N = ۹ , M= ۱۲	۳	۵۶۵	۵۳۲/۴	٪ ۵/۷۶
N = ۹ , M= ۱۲	۴	۵۶۵	۴۵۵/۵	٪ ۱۹/۳۸



شکل (۱۷): تاثیر خرابی ربات‌ها بر عملکرد الگوریتم

۶- جمع‌بندی و نتیجه گیری

از نوآوری‌های این مقاله ارائه معماری دوسطحی جهت حل مسئله‌ی پویش محیط و انجام کارهای موجود در آن توسط ربات‌های غیر همگن می‌باشد. این معماری در سطح اول شامل ارائه الگوریتم توزیع شده Multi SRT جهت پویش محیط ناشناخته و Multi Tangent Bug جهت مسیریابی ربات‌ها به سمت کارهای موجود بوده و در سطح دوم شامل به‌کارگیری الگوریتم ژنتیک جهت تخصیص متمرکز ربات‌ها می‌باشد. در این روش ربات‌ها دارای حافظه‌ی محدودی بوده و باید تا حد امکان مکان‌های بازدید نشده را جستجو کنند. به دلیل استفاده از قابلیت توزیع شده در سطح اول، معماری ارائه شده دارای قابلیت اطمینان بالایی است که با خرابی یک یا چند جزء کارایی کل سیستم از بین نمی‌رود. همچنین به دلیل استفاده از ربات‌های ساده دارای هزینه پایین‌تر در جنبه-های کاربردی می‌باشد. در سطح دوم نیز به دلیل استفاده از رویکرد متمرکز در تخصیص کار، دقت بالا در ارائه جواب از معیارهای مهم معماری ارائه شده می‌باشد.

از ویژگی‌های بارز الگوریتم ژنتیک ارائه شده نیز سرعت بالا به همراه کیفیت مناسب در ارائه جواب‌های شدنی می‌باشد، بدین

ترتیب که بایستی پس از پیدا شدن کارها الگوریتم ژنتیک اجرا شده و تخصیص کار به سرعت انجام گیرد تا زمان کافی برای پویش مجدد محیط در مدت زمان مشخص شده وجود داشته باشد. ضمناً پارامترهای مهم الگوریتم ژنتیک توسعه داده شده، احتمال اجرای عملگر تقاطع، احتمال اجرای عملگر جهش، جمعیت اولیه، و تعداد تکرار هستند که با استفاده از روش تاگوچی تنظیم شده‌اند.

با توجه به نتایج به دست آمده از اجرای الگوریتم می‌توان گفت میانگین درصد انحراف از کران بالا برای مسائل آزماینده در نظر گرفته شده در حدود ۱۰ درصد می‌باشد که مسئله ۱ با ۲۰ درصد و مسئله ۸ با ۰/۸ درصد به ترتیب دارای بیشترین و کمترین انحراف هستند. این بدان معنی است که مطلوبیت حاصل از انجام کارهای کشف شده توسط ربات‌ها در طول اجرای الگوریتم، به اندازه‌ی ۱۰٪ کمتر از مطلوبیت کل کارهای موجود در محیط است. از دیگر معیارهای مهم محاسبه شده می‌توان به تاثیر خرابی ربات‌ها بر عملکرد الگوریتم اشاره نمود. نتایج به دست آمده برای مسئله ۸ نشان می‌دهد که در صورت خرابی یکی از ربات‌ها کارایی سیستم کاهش پیدا نمی‌کند ولی برای خرابی ۳ و ۴ ربات عملکرد سیستم به ترتیب به اندازه ۵/۷۶ و ۱۹/۳۸ درصد کاهش می‌یابد.

همان‌گونه که در بخش‌های قبل اشاره شد، در سطح دوم معماری ارائه شده اطلاعات به دست آمده در یک حافظه مرکزی ثبت شده و تخصیص کار به صورت متمرکز در این سطح انجام می‌گیرد. از جمله تحقیقات آتی می‌توان به حذف این سیستم مرکزی و ارائه‌ی چارچوب کاملاً توزیع شده اشاره نمود که در آن ربات‌ها تنها از طریق ارتباطات محلی و اطلاعات محدود به دست آمده توسط خودشان، تصمیمات لازم را اتخاذ می‌کنند. میزان تبادل اطلاعات بین ربات‌ها و کاهش سطح اطلاعات تبادلی بین ربات‌ها به کمترین مقدار ممکن از دیگر کارهای تحقیقاتی آتی در این زمینه به شمار می‌رود. درنظرگیری ویژگی‌های مختلف برای محیط کاری و همچنین ربات‌ها و کارها می‌تواند به عنوان توسعه‌ی چارچوب ارائه شده در این تحقیق درنظر گرفته شود. به فرض مثال، درنظرگیری محیط کاری با موانع سیار، وجود محدودیت‌های حرکتی ربات‌ها از جمله غیرهولونومیک بودن آنها از جمله پیشنهادات آتی به شمار می‌روند.

مراجع

[1] Farinelli, A., Iocchi, L., Nardi, D., (2004). "Multirobot systems: a classification focused on coordination. Syst", Man Cybern.Part B Cybern. IEEE Trans. 34: 2015–2028.

[2] Cardarelli, E., Sabattini, L., Secchi, C., Fantuzzi, C., (2014). "Multisensor data fusion for obstacle detection in automated factory logistics", International Conference on. Intelligent Computer Communication and Processing (ICCP), 2014: 221–226.

[3] Wood, A., Virone, G., Doan, T., Cao, Q., Selavo, L., Wu, Y., Fang, L., He, Z., Lin, S., Stankovic, J., (2006). "Wireless sensor networks for assisted-living and residential monitoring". Univ. Va. Comput. Sci. Dep. Tech. Rep.

[4] Fabro, J., Reis, L.P., Lau, N., others, (2014). "Using Reinforcement Learning Techniques to Select the Best Action in Setplays with Multiple Possibilities in Robocup Soccer Simulation Teams", in IEEE Joint Conference on Robotics: SBR-LARS Robotics Symposium and Robocontrol (SBR LARS Robocontrol), 2014: 85–90.

[5] De Lope, J., Maravall, D., Quiñonez, Y., (2015). "Self-organizing techniques to improve the decentralized multi-task distribution in multi-robot systems", Neurocomputing 163: 47–55.

[6] Liu, L., Shell, D.A., (2012). "Large-scale multi-robot task allocation via dynamic partitioning and distribution", Auton. Robots 33: 291–307.

[7] Shi, Z., Chen, Q., Li, S., Cai, H., Shen, X., (2010). "Cooperative task allocation for multiple mobile robots based on multi-objective optimization method, in: 3rd IEEE International Conference on", Computer Science and Information Technology (ICCSIT), 2010: 484–489.

[8] Sarker, M.O.F., Dahl, T.S., Arcaute, E., Christensen, K., (2014). "Local interactions over global broadcasts for improved task allocation in self-organized multi-robot systems". Robot. Auton. Syst. 62: 1453–1462.

[9] Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., Shmoys, D.B., (1985). "The traveling salesman problem. A guided tour of combinatorial optimization". Wiley.

[10] Applegate, D., Cook, W., Dash, S., Rohe, A., (2002), "Solution of a min-max vehicle routing problem". Inf. J. Comput. 14: 132–143.

[11] Korsah, G.A., Stentz, A., Dias, M.B., (2013). "A comprehensive taxonomy for multi-robot task allocation". Int. J. Robot. Res. 32, 1495–1512.

[12] Cao, Y., Yu, W., Ren, W., Chen, G., (2013). "An overview of recent progress in the study of distributed multi-agent coordination". IEEE Trans. On Ind. Inform. 9: 427–438.

[13] Capitan, J., Spaan, M.T., Merino, L., Ollero, A., (2013). "Decentralized multi-robot cooperation with auctioned POMDPs". Int. J. Robot. Res. 32: 650–671.

[14] Viet, H.H., Dang, V.H., Choi, S., Chung, T.C., (2015). "BoB: an online coverage approach for multi-robot systems", Appl. Intell. 42: 157–173.

[15] Ranjbar-Sahraei, B., Weiss, G., Nakisaee, A., (2012). "A multi-robot coverage approach based on stigmergic communication, in: Multiagent System Technologies", Springer, pp. 126–138.

[16] Dorigo, M., (1992). "Optimization, learning and natural algorithms. PhD Thesis", Politec. Milano, Italy.

[17] Nedjah, N., de Mendonc, R.M., de Macedo Mourelle, L., others, (2015). "PSO-based Distributed Algorithm for Dynamic Task Allocation in a Robotic Swarm", Procedia Comput. Sci. 51: 326–335.

[18] Tolmidis, A.T., Petrou, L., (2013). "Multi-objective optimization for dynamic task allocation in a multi-robot system", Eng. Appl. Artif. Intell. 26:1458–1468.

[19] Siméon, T., Leroy, S., Lauumond, J.P., (2002). "Path coordination for multiple mobile robots: A resolution-complete algorithm. IEEE Trans", Robot. Autom. 18: 42–49.

[20] Bennewitz, M., Burgard, W., Thrun, S., (2001). "Optimizing schedules for prioritized path planning of multi-robot systems", Proceedings of IEEE International Conference on Robotics and Automation, ICRA 271–276.

[21] Akella, S., Hutchinson, S., (2002). "Coordinating the motions of multiple robots with specified trajectories", Proceedings of IEEE International Conference on Robotics and Automation ICRA, 624–631.

[22] Parker, L.E., (1998). "ALLIANCE: An architecture for fault tolerant multirobot cooperation", IEEE Trans. On Robot. Autom. 14: 220–240.

[23] Chao-xia, S., Bing-rong, H., Yan-qing, W., (2007). "Cooperative exploration by multi-robots without global localization", Int. J. Adv. Robot. Syst. 4: 339–348.

[24] Iocchi, L., Nardi, D., Piaggio, M., Sgorbissa, A., (2003). "Distributed coordination in heterogeneous multi-robot systems". Auton. Robots 15: 155–168.

[25] Lumelsky, V.J., Stepanov, A.A., (1987). "Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape". Algorithmica 2: 403–430.

[26] Kamon, I., Rivlin, E., Rimon, E., (1996). "A new range-sensor based globally convergent navigation algorithm for mobile robots", in Proceedings of IEEE International Conference on Robotics and Automation, 1996: 429–435.

[27] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., (2002). "A fast and elitist multiobjective genetic algorithm: NSGA-II", IEEE Trans. on Evol. Comput. 6: 182–197.

IERPS Journal of Industrial Engineering Research in Production Systems

ISSN: 2345-2269

Journal of Industrial Engineering Research in Production Systems

Spring and Summer 2017 Volume 5. No. 10. pp.31-51
www.ier.basu.ac.ir

DOI: 10.22084/ier.2017.8001.1392

Multi Robot Exploration and Task Allocation through a Two-layer Architecture and Genetic Algorithm

H. Youssefi Halvaei[1], E. Masihi[1,*]

[1] Department of Industrial Engineering, Tarbiat Modares University, Tehran, Iran.

ARTICLE INFO

***Article history*:**

Received 27 August 2015
Accepted 9 May 2017

***Keywords*:**

Multi Robots
Environment Covering
Online Path Planning
Task Allocation
Genetic Algorithm

ABSTRACT

Multi-robot systems are preferable for tasks that are inherently distributed in space, time, or functionality. For the problems that can be decomposed into independent subproblems, using a multi-robot system offers a potential for reducing the overall task completion time. For effective employment of multi robot systems, it is necessary to properly implement Task Allocation, which is an NP-hard problem. In this paper, a two-layer architecture for exploring and covering an unknown environment by multiple heterogeneous robots is developed. At the first layer of the architecture, the Multi-SRT algorithm is developed for exploration and covering of the environment and the Multi-Tangent-Bug is used for online path planning and obstacle avoidance in a distributed manner. In the second layer, by means of a centralized approach, a Fast Genetic Algorithm (FGA) is proposed for solving the multi-robot task allocation problem. Performing each task enhances the utility of the system, and completing all tasks is the ultimate goal of the system. For evaluating the efficiency of the FGA, a number of scenarios were run and the results were compared to NSGA-II algorithm. Simulation results showed the reliability of the developed architecture at the first layer and the precision and quality of the task allocation at the second layer.

* Corresponding author. Ellips Masihi
Tel.: 021-82884939; *E-mail address: masehian@modares.ac.ir*