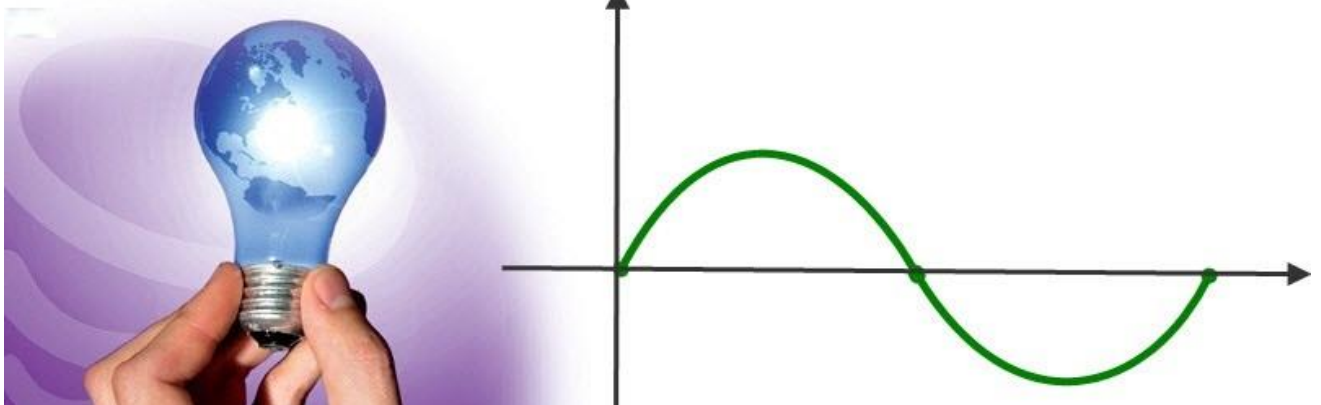


برای دریافت فایل Word پروژه به سایت **ویکی پاور** مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم



برای دریافت فایل Word پروژه به سایت **ویکی پاور** مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

موضوع پروژه:

رباتیک

PATH PLANNING FOR A
MOBILE ROBOT
USING GENETIC ALGORITHMS

برای خرید فایل word این پروژه [اینجا کلیک کنید](#).

(شماره پروژه = ۵۳۹)

پشتیبانی: ۰۹۳۵۵۴۰۵۹۸۶

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

چکیده

این مقاله الگوریتمی جدید برای مسئله برنامه ریزی مسیر کلی به یک هدف ، برای ربات متحرک را با استفاده از الگوریتم ژنتیک ارائه می دهد. الگوریتم ژنتیک برای یافتن مسیر بهینه برای ربات متحرک جهت حرکت در محیط استاتیک که توسط نقشه ای با گره ها و لینک ها بیان شده است ، بکار گرفته شده است. موقعیت هدف و موانع برای یافتن یک مسیر بهینه در محیط دو بعدی داده شده است. هر نقطه اتصال در شبکه ژنی است که با استفاده از کد باینری ارائه شده است. تعداد ژن ها در یک کروموزوم تابعی از تعداد موانع در نقشه (نمودار) می باشد.

بنابراین از یک کروموزوم با طول ثابت استفاده کردیم. مسیر ربات ایجاد شده ، در مفهوم کوتاهترین مسیر ، بهینه است. ربات دارای محل آغاز و محل هدف تحت فرضیه ای است که ربات از هر محل فقط یکبار می گذرد یا اصلا نمی گذرد. نتایج بدست آمده در شبیه سازی ؛ قدرت الگوریتم پیشنهادی را تایید می نماید.

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

مقدمه

مسئله طراحی مسیر ربات متحرک را می توان بصورت ذیل بیان کرد:

داده های مسئله (محل شروع، محل هدف، نقشه ای دو بعدی مسیرهاکه شامل موانع ساکن می باشد). هدف بدست آوردن یک مسیر بدون تصادم بین دو نقطه خاص در ایفای معیار بهینه سازی با در نظر گرفتن محدودیت ها (به احتمال زیاد: کوتاهترین مسیر) می باشد. مسئله طراحی مسیر از نظر محاسباتی بسیار پر هزینه است.

با اینکه حجم زیادی از تحقیقات برای حل بیشتر این مسائل انجام شده است، با این وجود، روش های معمول، غیر قابل انعطاف می باشند.

۱. اهداف مختلف بهینه سازی و تغییرات اهداف

۲. عدم قطعیت ها در محیط ها

۳. محدودیت های متفاوت برای منابع محاسباتی

مرور و بازنگری روش های موجود برای حل مسئله طراحی مسیر، در [1] ارائه شده است. روش های زیادی برای ایجاد یک مسیر بهینه از قبیل برنامه ریزی دینامیک و روش های تبدیل مسافت گزارش شده است. در روش برنامه ریزی دینامیک اگر نقطه ی شروع SP و نقطه ی هدف PG باشد، نقطه ی زیر هدف PI است. و روش تولید مسیر، نحوه تعیین توالی زیر اهداف است که زیر اهداف خود از مجموعه PI ($l=1,2,3,\dots$) انتخاب می شوند. ما باید تمام مسیرهای ممکن را بررسی کرده و مسیر با کمترین مقدار هزینه را به عنوان مسیر بهینه انتخاب نمائیم. توان محاسباتی بسیار فراوانی بویژه در محیط های دارای زیر اهداف فراوان مورد نیاز است. در روش تبدیل مسافت، کار طراحی مسیر، محیطی را با شبکه یکنواخت می پوشاند و فواصل را از طریق فضای خالی، از سلول هدف، منتشر می کند. قسمت پیشین موج مسافت، حول

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

موانع و در نهایت از طریق تمامی فضاهای آزاد در محیط جریان می یابد. برای هر نقطه شروع در محیط نمایانگر محل اولیه ربات متحرک، کوتاهترین مسیر به مقصد، از طریق رفتن به قسمت پائین و از طریق شیب دارترین مسیر نزولی رسم شده است. با این وجود به هنگام وجود دو سلول یا بیشتر جهت گزینش با همان حداقل تبدیل فاصله ابهام مسیرهای بهینه وجود دارد. دو روش مذکور ملزم توان محاسباتی بسیار بالا در محیطی است که دارای تعداد زیاد اهداف فرعی (زیر اهداف) و موانع است.

محققان روش های فراوان را برای حل مسائل طراحی مسیر ربات های متحرک با وجود موانع ایستا و متحرک بر مبنای soft computing، بیان کرده اند. soft computing متشکل از منطق فازی، شبکه های عصبی و محاسبات تکاملی است (الگوریتم های ژنتیک و تکاملی GA & EA). تاکنون تلاش های زیادی در استفاده از منطق فازی برای طراحی و برنامه ریزی حرکت ربات متحرک وجود داشته است. اخیراً استفاده از محاسبات تکاملی رواج فراوانی پیدا کرده و در واقع روشی است که به منظور بکارگیری در موقعیت هایی که دانش اولیه راجع حل مسئله وجود نداشته و یا اطلاعات محدود می باشد، قابلیت استفاده به گونه ای موثرتر، عمومی تر و راحت تر را داراست.

الگوریتم های ژنتیکی و تکاملی نیازمند اطلاعات اشتقاقی یا برآوردهای فرمال اولیه از راه حل نیستند و از آنجائیکه طبیعتاً تصادفی می باشند دارای قابلیت جستجوی کل فضای جواب با احتمال بیشتر پیدا کردن بهینه عمومی می باشند.

می توان تحقیق قبلی راجع طراحی مسیر را به صورت یکی از دو روش مقابل طبقه بندی کرد: مبتنی بر مدل و مبتنی بر سنسور .

در حالت مبتنی بر مدل، مدل های منطقی از موانع شناخته شده، برای تولید تصادم بدون مسیر بکار گرفته می شوند. در حالیکه در روش مبتنی بر سنسور، کشف و اجتناب از موانع ناشناخته است. در این مقاله الگوریتمی جدید جهت بدست آوردن مسیر بهینه بر مبنای مدل پیشنهاد شده است.

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

ادامه مطالب مقاله بصورت ذیل مرتب شده اند :

در بخش ۲، مقدمه ای مختصر راجع الگوریتم ژنتیک ارائه شده است. در بخش ۳، فرمول سازی مسئله مورد بررسی واقع شده، در بخش ۴ الگوریتم پیشنهادی، معرفی و در بخش ۵ نتایج شبیه سازی نشان داده شده است.

۱. مسیریابی

مسئله مسیریابی ربات در چند حالت قابل بررسی است :

در یک مفهوم می توان مسیریابی ربات را در قالب تعقیب خط (عموما مسیری از پیش تعیین شده با رنگ متفاوت از زمینه) معرفی نمود. ربات هایی با این کاربرد تحت عنوان مسیریاب شناخته می شوند. یکی از کاربردهای عمده این ربات، حمل و نقل وسایل و کالاهای مختلف در کارخانجات، بیمارستان ها، فروشگاه ها، کتابخانه ها و ... میباشد. ربات تعقیب خط تا حدی قادر به انجام وظیفه کتاب داری کتابخانه ها می باشد. به این صورت که بعد از دادن کد کتاب، ربات با دنبال کردن مسیری که کد آن را تعیین میکند، به محلی که کتاب در آن قرار گرفته می رود و کتاب را برداشته و به نزد ما می آورد. مثال دیگر این نوع ربات در بیمارستان های پیشرفته است، کف بیمارستان های پیشرفته خط کشی هایی به رنگ های مختلف به منظور هدایت ربات های مسیریاب به محل های مختلف وجود دارد. (مثلا رنگ قرمز به اتاق جراحی یا آبی به اتاق زایمان). بیمارانی که توانایی حرکت کردن و جا به جا شدن را ندارند و باید از ویلچر استفاده کنند، این ویلچر نقش ربات تعقیب خط را دارد، و بیمار را از روی مسیر مشخص به محل مطلوب می برد.

با توجه به وجود موانع (استاتیک و دینامیک) در محیط، مسیریابی ربات در مفهومی کاربردی تر، پیمودن مسیر مبدا تا مقصد بدون برخورد با موانع می باشد. مسلما با وجود تعداد زیاد موانع، تعداد مسیرهای قابل عبور ربات بسیار زیاد خواهد بود و یقینا انتخاب کوتاه ترین مسیر توسط ربات برای حرکت از مبدا به

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

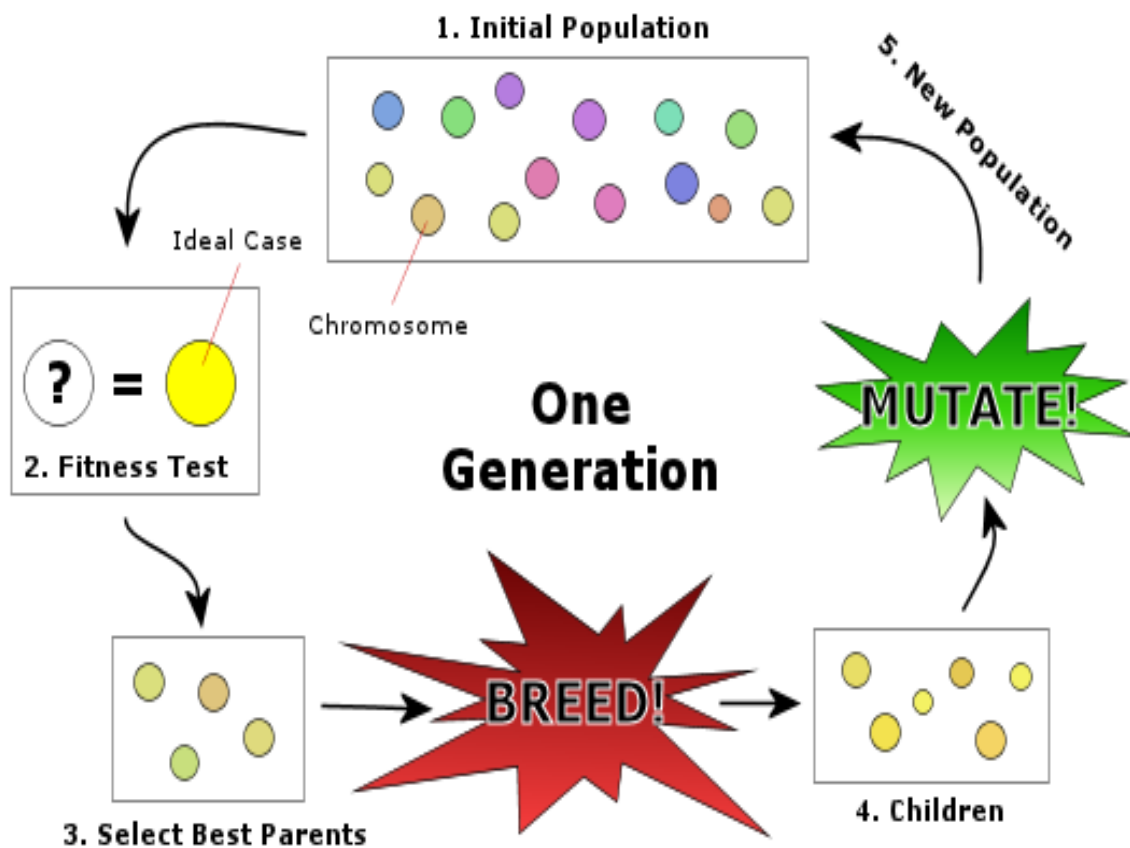
مقصد، دارای ارزش اجرایی بالایی خواهد بود. در این مقاله چنین مسئله ای مورد بررسی واقع شده است. نقاط مبدا و مقصد و نیز محل موانع به عنوان ورودی داده شده است، نیز می دانیم موانع ایستا می باشند (در حالت وجود موانع پویا در عین نزدیکی بیشتر به شرایط واقعی، روش های مورد استفاده بسیار پیچیده خواهند بود) و مسئله در حالت دو بعدی بررسی می شود (روبات بر روی صفحه حرکت می نماید). برای این منظور الگوریتم های مسیریابی با هدف انتخاب کوتاهترین مسیر قابل استفاده می باشند، الگوریتم هایی که به منظور مسیریابی در شبکه ها قابلیت استفاده دارند. با این وجود در این بررسی از الگوریتم ژنتیک استفاده شده است. همچنین الگوریتم های ژنتیک و نیز دیگر روش های مشابه به منظور بهینه سازی مصرف انرژی روبات، مسیر تغییر زاویه ازوی روبات، زمان حرکت روبات و... قابل استفاده می باشند.

۲. الگوریتم ژنتیک

GA در سال ۱۹۷۵ توسط Holland بر پایه تقلیدی از تکامل طبیعی یک جمعیت پایه ریزی شد به نحوی که کروموزوم ها به منظور خلق نسل جدید اجازه تولید مجدد داشته و جهت بقاء در نسل آینده به رقابت می پردازند. با گذشت زمان، بر روی نسل ها، fitness بهبود می یابد و در نهایت بهترین راه حل قابل حصول است. اولین جمعیت $p(0)$ به طور تصادفی با n و m کد می شود در هر نسل t ، مناسبترین عناصر برای حضور در mating pool انتخاب می شوند و با سه عملگر پایه ای ژنتیک؛ تولید مثل، ادغام و جهش؛ جهت تولید نسل جدید تکامل می یابند. بر پایه بقاء بهترین هابی توان نتیجه گرفت کروموزوم های بدست آمده با استفاده از روشی منتخب بهترین کروموزوم ها قابل حصول می باشند.

برای دریافت فایل Word پروژه به سایت **ویکی پاور** مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

از جمله مزایای GA که این روش را جهت بکارگیری آن در مورد انتخاب متغیر مناسب می نماید می توان به توانایی پیدا کردن بهینه عمومی با سرعت بالا، امکان جستجو موازی چند نقطه و نیز فرار از بهینه های محلی اشاره نمود.



Procedure

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

GA

Begin

t=0

initialize p(t)

evaluate p(t)

while not satisfy stopping rule do

begin

t=t+1

select p(t) from p(t-1)

alter(t)

evaluate p(t)

end

end

چنانچه بیان شد عموماً تکامل از یک نسل به نسل بعد، شامل سه مرحله است: ارزیابی تناسب، گزینش و بازآفرینی.

ابتدا، جمعیت کنونی با استفاده از تابع تکامل تناسب ارزیابی شده و سپس بر اساس مناسب بودنشان طبقه بندی می شوند و در واقع نسل جدید با هدف بهبود و ارتقاء تناسب بوجود می آید.

روش بکار بردن عملگرهای، تولیدمثل، جهش و ادغام توسط الگوریتم ژنتیک به شکل زیر است:

در آغاز، باز آفرینی منتخب، بر روی جمعیت کنونی بنحوی بکار گرفته می شود که رشته، تعدادی کپی، بر اساس مناسب بودن آنها تهیه می کند. این عمل منجر به تولید جمعیت میانی خواهد شد. سپس دوما، الگوریتم ژنتیک والدین را از جمعیت کنونی با احتمال بیشتر در انتخاب کروموزوم های بهتر گزینش می نماید. این عمل همراه با کمیت تناسب و دسته بندی کروموزوم خواهد بود و نهایتاً (سوما)، این الگوریتم فرزندان (رشته های جدید) را از والدین منتخب با استفاده از اپراتورهای ادغام یا جهش بازآفرینی می نماید. اساساً ادغام، شامل تبادلی تصادفی بیت هابین دو رشته جمعیت میانی می باشد. در نهایت عملگر جهش

برای دریافت فایل Word پروژه به سایت **ویکی پاور** مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

به طور تصادفی تعدادی از بیت های بین رشته های جدید را تعویض می نماید. این الگوریتم زمانی پایان می یابد که راه حل قابل قبول پیدا شود یا معیار همگرایی ایفا شود و یا وقتی که به تعداد محدود و از پیش تعیین شده تکرار دست یابیم. مشخصه های اصلی الگوریتم های ژنتیک این است که آنها می توانند فضای جستجو را به طور برابر جستجو کنند و نیازی به بهینه سازی تابع برای تمایز گذاشتن یا هرگونه ویژگی یکنواخت ندارند. دقت راه حل اکتسابی به تعداد کد مورد استفاده برای کدگذاری متغیر خاص (طول کروموزوم) بستگی دارد.

۳. فرمول سازی مسئله

مسئله به صورت زیر بیان شده است: ربات متحرک در مسیری بسته حرکت می نماید. این ناحیه به صورت نموداری دو بعدی توصیف می شود. چنین نموداری شامل تعداد محدودی موانع ایستا و یک سری نقاط به منظور جلوگیری از برخورد رباتها به موانع است. هدف، طراحی مسیر بهینه بدون هیچ گونه تصادم بین نقطه شروع و نقطه هدف می باشد. به این معنا که باید معیار هدف را با محدودیت عاری از تصادم بهینه سازی کرد. اگر نمودار دو بعدی شامل نقاط اتصال باشد، فضای کل راه حل های احتمالی، بسیار بزرگ است. در واقع فضای C هر ربات، فضای کل پیکربندی های مسیر ربات است.

مسئله طراحی مسیر، معادل با طراحی حرکت یک نقطه از ربات (نقطه مرجع) در فضای C می باشد. فضای C بدون برخورد، اشاره به مکان هندسی تمام نقاط فضای C دارد که نمایانگر پیکربندی های عملی و عاری از تصادم است. به این معنا که باید مسیر بهینه عملی و میسر باشد. به طور کل برای پیدا کردن مسیر بهینه بدون تصادم، سه هدف بهینه سازی متفاوت وجود دارد:

۱. به حداقل رساندن مسافت طی شده (کوتاهترین مسیر)

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

۲. حفظ مسیر یکنواخت

۳. ایفای ملزومات مشخص (ربات نباید خیلی به موانع نزدیک شود)

شکل ۱ موقعیت های مختلف را نشان می دهد

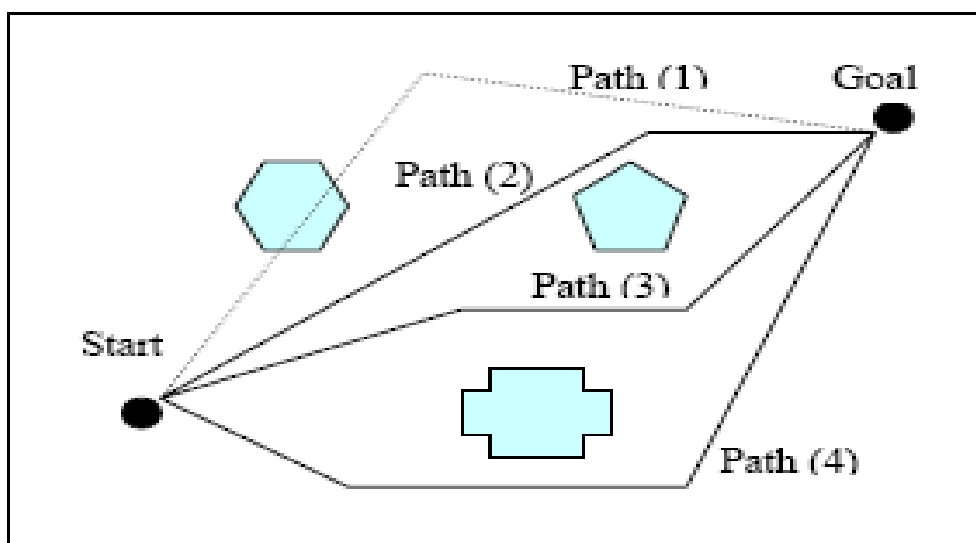


Figure 1
Path Planning Objectives

مسیر ۱. مسیری محتمل در نظر گرفته نمی شود (تصادم مانع)

مسیر ۲. کوتاهترین مسافت و مسیر محتمل است

مسیر ۳. میسر بوده و مطمئن تر است (دورترین مسیر از مانع)

مسیر ۴. میسر است ولی معیار بهینگی را ایفا نمی کند

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

می توان این اهداف را با عوامل متفاوت، نسبت به اهمیت شان در مسئله بهینه سازی ایستامورد آزمایش قرار داد. در این مقاله بر ارائه کوتاهترین مسیر بدون تصادم، بین نقطه شروع (PS) و نقطه هدف (PG) با استفاده از الگوریتم ژنتیک متمرکز می شویم.

۴. الگوریتم طراحی مسیر پیشنهادی

در این بخش، برای پیدا کردن مسیرهای بهینه برای یک ربات، جهت حرکت در محیط ایستای بیان شده از طریق یک نمودار با گره ها و لینک ها، الگوریتم ژنتیک بکار رفته است. از موارد کاربرد این مسئله می توان به استفاده از چنین ربات هایی در نمایشگاه ها، بیمارستان ها، موزه ها و لابراتوارها برای انتقال مواد اشاره داشت. در الگوریتم پیشنهادی، هر مسیر از نقطه شروع تا هدف، جواب است. در آغاز جمعیت های تصادفی از رشته، تولید شده که نشان دهنده جواب های مجاز (میسر) یا غیر مجاز (غیر میسر) می باشد. جواب های غیر مجاز، رشته هایی هستند که نمی توانند به هدف برسند. به این معنا که جواب رشته، ربات را به حرکت در تصادم با موانع، هدایت می کند. جواب های مجاز، رشته هایی هستند که می توانند به هدف برسند. برای حرکات ربات متحرک به سمتی که از برخورد با موانع اجتناب نماید باید بهترین رشته ای را بیابیم که هدف را در کوتاهترین مسیر می رساند.

راه حل غیرمجاز دارای کمترین تناسب است (تناسب صفر). نمودار $x-y$ که در مقاله مورد بررسی واقع شده است چنانچه در شکل 2 نشان داده شده است ناحیه ای با مساحت $15m^2$ می باشد.

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

ربات دارای یک نقطه شروع و نقطه هدف روی نمودار ، تحت این فرضیه است که در طراحی مسیر ، ربات از هر نقطه فقط یکبار می گذرد یا اصلا نمی گذرد. هر گره دارای یک عدد در شکل ۲ است و گره ها برای رمزگذاری مسیر ، به صورت رشته بیان شده طبق ترتیب اعداد ، بکار می روند. برای مثال 0-4-5-10-9-15 یک مسیر با $PS=0$ و $PG=15$ می باشد.

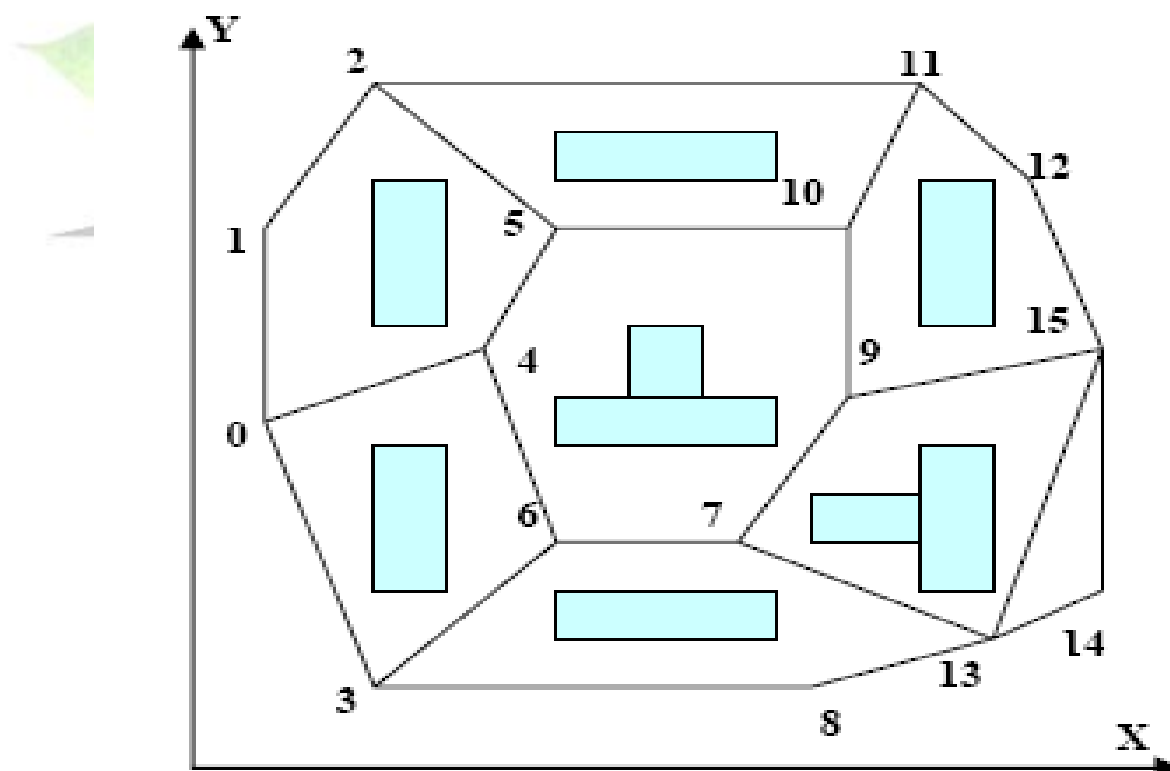


Figure 2
Workspace Map

برای دریافت فایل Word پروژه به سایت **ویکی پاور** مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

با استفاده از الگوریتم ژنتیک، نقاط اتصال، در آغاز بطور تصادفی انتخاب شده اند، حال آنکه می بایستی اعداد مجاور، با یک لینک روی نمودار، بهم متصل باشند. در این زمینه، اکثر محققان، رشته مبنی بر ترتیب و دارای طول متغیر (کروموزوم ها) را بکار برده اند. این کروموزوم، از طریق رشته بیت ها، بازآفرینی شده است (یعنی، نمایش طبیعی و نه دودویی). بنابراین باید عملگر ادغام و جهش خاصی را اتخاذ نمود. هنگام ادغام، رشته ای که مناسب تشخیص داده شده است، به طور تصادفی به عنوان یک والد انتخاب می شود. اگر والد دوم شامل عدد مشترکی در رشته اول باشد. یکی از دو رشته ها، بخشی از رشته هایشان را بعد از عدد مشترک مبادله می کنند در غیر اینصورت، رشته ای دیگر بصورت والدین دوم انتخاب می شود و همین فرآیند دنبال می شود برای مثال با استفاده از نمودار شکل ۲:



هر والد دارای نقطه مشترک ۱۰ است، بخش زیر خط دار هر رشته مبادله شده و دو مولود بصورت زیر ایجاد می شود

مولود ۱: 0-1-5-10-9-15

مولود ۲: 0-3-6-4-5-10-11-12-15

برای دریافت فایل Word پروژه به سایت **ویکی پاور** مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

بعد از ادغام، برای مشخص کردن تکرار یا عدم تکرار عدد در هر رشته، مولودها (فرزندان) بررسی می شوند. در این صورت بخشی از رشته بین ارقام تکراری حذف می شود. سپس اصلاحاتی مورد نیاز است زیرا ممکن است حالتی پیش آید که مولودها راه حل مجاز نیستند. این روش، الگوریتم را پیچیده تر می سازد. مسئله مهم، نحوه تولید کدهای دودویی به گونه ای که استفاده از کروموزوم هایی با طول ثابت عملی و آسان است. یکی از چالش آورترین جنبه های روش پیشنهادی، کد گذاری هر رشته در کد دودویی با طول ثابت است. مسیر ربات متحرک با استفاده از اعداد دودویی کدگذاری شده که در آن هر ژن در یک کروموزوم، از طریق ۴ بیت باینری به گونه ای که در جدول ۱ نشان داده شده است، کدگذاری شده است.



برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

Table I
Map Via Points

Point index	Point code	X-value (m)	Y-value (m)
0	0000	1	7
1	0001	1	11
2	0010	3	14
3	0011	3	1
4	0100	5	8
5	0101	6	11
6	0110	6	4
7	0111	8	4
8	1000	10	1
9	1001	10	7
10	1010	10	11
11	1011	11	14
12	1100	13	12
13	1101	12	2
14	1110	14	3
15	1111	14	8

WikiPower.ir

A. کروموزوم ها و جمعیت اولیه

یک کروموزوم، مطابق با راه حل عملی مسئله بهینه سازی است. بنابراین هر کروموزوم نشان دهنده مسیر است که متشکل از قطعات خط راست می باشد، به صورت توالی گره ها با اولین گره که نشان دهنده نقطه شروع اولین قطعه، به دنبال گره های یانی بین قطعات و قطعه آخری نشان دهنده نقطه پایانی قطعه آخری، هدف، می باشد. حداکثر طول کروموزوم پیش فرض، برابر با تعداد نقاط نمودار در طول ژن است. در این حالت، ۱۶ نقطه داری که هر نقطه در ۴ بیت دودویی کدگذاری شده است. به این معناکه طول کروموزوم

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

برابر با $4 \times 16 = 64$ بیت می باشد. این نتیجه ساده است زیرا می توانیم طول موثر کروموزوم را برای کاستن توان محاسباتی لازم، کاهش دهیم. از توپولوژی نمودار مشاهده کردیم که مسیر بهینه برای رسیدن به نقطه هدف مرتبط با تعداد موانع است.

اگر تعداد موانع ایستا برابر با m باشد، کوتاهترین مسیر متشکل از حداکثر نقاط $(m+2)$ یا تعداد $(m+1)$ تکه خط است. این رابطه می پذیرد که شکل مانع پیچیده نیست و می تواند به صورت نقطه حجم در نظر گرفته شود.

برای مثال اگر هیچگونه مانعی وجود نداشته باشد ($m=0$) کوتاهترین مسیر متشکل از یک قطعه خطی از نقطه شروع تا هدف است (تعداد نقاط ۲) در شکل ۱، تعداد موانع ایستا $m=3$ ؛ کوتاهترین مسیر متشکل از ۳ نقطه؛ کمتر از ۵ ($m+2$) می باشد. در شکل ۲، $m=7$ ، کوتاهترین مسیر از نقاط ۰ تا نقطه ۱۵ $\{0-4-6-7\}$ (۶ نقطه) می باشد که کمتر از ۹ است ($m+2$). بنابراین معادله زیر برای تعریف طول کروموزوم بکار خواهد رفت. $C_n = m+2$ که m تعداد موانع ایستا و C_n تعداد ژن ها در کروموزوم است. برای نمودار داده شده در شکل ۲، طول کروموزوم $4 \times (7+2) = 44$ بیت دودویی، می توان جمعیت اولیه کروموزوم ها را به طور تصادفی چنان تولید کرد که هر کروموزوم دارای m ژن تصادفی باشد حال آنکه نقاط آغازین و هدف در جمعیت ثابت شده اند.

B. ارزیابی

معمولا انتخاب تابع ارزیابی بر مسئله تحت شرایط بسیار خاص است. تابع ارزیابی یک کروموزوم، برای محاسبه تعیین مناسب بودن آن بکار می رود. زیرا تناسب باید به صورت کاهش فاصله، افزایش یابد. بنابراین تابع تناسب F از یک مسیر محتمل بصورت زیر ارزیابی شده است.

برای دریافت فایل Word پروژه به سایت **ویکی پاور** مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

$$F = \begin{cases} \frac{1}{\sum_{i=1}^{m+1} d(P_i, P_{i+1})} & ; \text{ Feasible path} \\ 0 & ; \text{ Infeasible path} \end{cases}$$

که در آن $d(p_i, p_{i+1})$ طول قطعه بین دو نقطه p_i, p_{i+1} در کروموزوم با تعداد نقاط $m+2$ است.

Table II
Linked Points

Point index	Linked points
0	1, 3, 4
1	0, 2
2	1, 5, 11
3	0, 6, 8
4	0, 5, 6
5	2, 4, 10
6	3, 4, 7
7	6, 9, 13
8	3, 13
9	7, 10, 15
10	5, 9, 11
11	2, 10, 12
12	11, 15
13	7, 8, 14, 15
14	13, 15
15	9, 12, 13, 14

می

توان طول قطعه را با استفاده از مقادیر مختصات X-Y در جدول ۱ به صورت زیر محاسبه نمود

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

$$d(P_{i+1}, P_i) = \sqrt{(X_{i+1} - X_i)^2 + (Y_{i+1} - Y_i)^2}$$

اگر مسیر محتمل نباشد، مقدار تناسب آن صفر می باشد. الگوریتم پیشنهادی قادر است نقاط مسیر را برای آشکار سازی عملی بودن آن یا عدم عملی بودنش در جدول ۲، ترسیم نماید. نقاط اتصالی، نقاط مجاز مجاور به نقاط کنونی در راه حل برای اجتناب از موانع اند.

C. عملگرها

در الگوریتم، ازدو عملگر ژنتیکی رایج استفاده شده است. ادغام و جهش. ادغام، دو مسیر "والد" را برای تولید دومسیر جدید "مولود" در نسل بعدی دوباره ترکیب می کند. از ادغام دو نقطه ای استفاده شده است. هر دو مسیر والد به طور تصادفی به سه بخش تقسیم شده و دوباره ترکیب شده اند. بخش وسطی اولین مسیر بین موقعیت های بیت هی ادغام و بخش میانی دومین مسیر (والد دوم) برای ایجاد فرزندان جدید تبادل می شوند. موقعیت های بیت ادغام به طور تصادفی در طول کروموزوم بین موقعیت های ۵،۳۲ انتخاب شده اند. این محدودیت ها، برای عدم اعمال تغییر نقاط اولی و هدف، طی فرآیند ادغام، انتخاب شده اند. برای مثال، با مشخص ودن والدین (۳۶ بیت) به صورت زیر، باید به طور تصادفی در موقعیتهای ۱۵ و ۲۳ ادغام داده شوند. همچنین فرآیند جهش، برای تغییر تصادفی موقعیت یک بیت در کروموزوم کار برده شده است (بین موقعیت ۵ و ۳۲). شبه کد الگوریتم پیشنهادی بصورت ذیل داده شده است:

positions 15 and 23:

```
000000000111111100000000011000001111
0000000000000000011111111000000001111
```

The new two children are:

```
0000000001111111011111110011000001111
0000000000000000100000000000000001111
```

برای دریافت فایل Word پروژه به سایت **ویکی پاور** مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

BEGIN

Initialize the starting and goal points

Generate randomly the initial population using m via points in each chromosome

While NOT (convergence condition) DO

Evaluate the fitness for each chromosome in current population using equation (2)

Rank the population using the fitness values

Eliminate the lowest fitness chromosome

Duplicate the highest fitness chromosome

Apply randomly crossover process between current parents using the given probability, while keeping the start and end points without change in the population

Apply the mutation process with the given probability

Generate the new population

END

Output the best individual found

END

برای دریافت فایل Word پروژه به سایت **ویکی پاور** مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

۵. نتایج شبیه سازی

الگوریتم بالا به منظور شبیه سازی با استفاده از Matlab آزمایش شده است. محیط شبیه سازی دارای دستورات دستکاری رشته ای بسیار موثر است که تبدیل متغیرهای عددی به رشته ای و عکس این حالت را به گونه ای موثر میسر می سازد. در نتیجه عملگرهای ادغام و جهش به راحتی قابل پیاده سازی می باشند.

مثال شبیه سازی شده :

نقطه ی شروع ۰ و نقطه ی پایان ۱۵ است شکل ۲ به صورت تولید کننده مسیر شبیه سازی شده است. بیت آغازین متشکل از ۶ کروموزوم ۶۳ بیتی است. احتمال ادغام ۱۰۰ درصد در نظر گرفته شده و احتمال جهش ۰,۰۰۵ درصد می باشد. بهترین نتایج با استفاده از دو نقطه ادغام بدست آمده است. نتایج شبیه سازی بعد از ۵۰ نسل بصورت زیر داده شده است :

```
000001000110011001100111100111111111
000001000110011011110111100111111111
000001000110011001100111100111111111
000001000110011001100111100111111111
000001000110011001100111100111111111
000001000110011001100111100111111111
```

نتیجه ی بالا باینری معادل با جمعیت مسیرها به صورت زیر می باشد.

برای دریافت فایل Word پروژه به سایت **ویکی پاور** مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

{0-4-6-6-6-7-9-15-15}
 {0-4-6-6-15-7-9-15-15}
 {0-4-6-6-6-7-9-15-15}
 {0-4-6-6-6-7-9-15-15}
 {0-4-6-6-6-7-9-15-15}
 {0-4-6-6-6-7-9-15-15}

کروموزوم دومی به علت فرآیند جهش، تاحدودی متفاوت است. بخشی از اعداد تکراری، جداشده و نتیجه نهایی کوتاهترین مسیر منفرد {0-4-6-7-9-15} یا مسیر بهینه منفرد با طول 17.975 همگرا شده است. این مسئله، با سایر نقاط آغازین تست شده و نقاط نهایی و همگرایی آن، برای بدست آوردن مسیر بهینه در هر مورد ضمانت شده است.

الگوریتم ژنتیک ساده با طول ثابت کروموزوم، برای یک ربات متحرک به منظور دریافت کوتاهترین مسیر در محیط ایستای دو بعدی با M مانع، شکل گرفته است. الگوریتم توسعه یافته از طرح کدگذاری موثر با رشته دودویی دارای طول ثابت استفاده می نماید. طول کروموزوم بستگی به تعداد موانع ساکن دارد. هر کروموزوم دارای طول ثابت $(m+2)$ ژن است. این الگوریتم نیازمند توان محاسباتی کمتر از سایر الگوریتمهای اخیرا توسعه یافته است. در محیط های دینامیک لازم است طول کروموزوم متغیر در نظر گرفته شود.

بررسی بیشتر

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

چنانچه در بخش ۱ اشاره گردید نسخه های زیادی از مسأله طراحی مسیر وجود دارد. در این بخش با عنوان بررسی بیشتر، هدف بررسی اجمالی یک نمونه مطالعه موردی در بهینه سازی حرکت بازوی روبات و نیز مروری بر روشهای قبلی اعمال شده می باشد. دسته بندی دقیقی از این مسائل و روشهایی برای حل آنها را می توان در بررسی انجام شده توسط Hwang و Ahuja در سال ۱۹۹۲ پیدا کرد. برای روشن شدن بحث، مورد ویژه ای را بررسی می نمائیم. یک بازوی ربات در طول مجموعه از موانع قرار گرفت. موقعیت های ابتدایی و انتهایی بازوی ربات داده شدند و مسأله پیدا کردن مجموعه حرکاتی است که روبات را راهنمایی می کند تا بین دو موقعیت بدون برخورد با موانع حرکت کند.

برای حرکت دادن روبات از بین موانع، روشهای قبلی (Brooks 1983) مستقیماً بکار رفتند تا مدل‌های 3D CAD برای روبات و موانع برای یافتن راه حلی بکار آیند،

یعنی «فضای سه بعدی اختیاری» مدنظر آنها بود. در این فضا مسأله طراحی مسیر شامل یافتن حرکاتی از ساختار مجتمع سه بعدی روبات در فضایی درهم ریخته ۳ بعدی می باشند.

پیشرفت اساسی، بیان مسأله در فضای شناخته شده دیگر به صورت فضای پیکربندی شده بود که توسط Lozano perez بیان گردید. در این فضا موقعیت (پیکربندی) روبات کاملاً توسط نقطه ای منفرد که دارای n پارامتر مستقل به صورت هماهنگ است مشخص شده بود. موقعیت هایی که از لحاظ فیزیکی غیرقانونی اند (به دلیل تصادفها) توسط نواحی خاصی بیان و نامیده می شوند. در فضای پیکربندی شده، مسأله طراحی مسیر، شامل یافتن یک منحنی پیوسته است (بیان و نمایش مسیر برای یک نقطه منفرد هندسی) است که:

(i) نقاط بیان کننده پیکربندی اولیه و نهایی روبات را به هم وصل می کند

(ii) یکدیگر را قطع نمی کنند

این روش سبب تسهیل مسأله طراحی مسیر می گردد (یک مسیر را برای یک نقطه تنها جستجو می کند) در مقابل جستجوی یک فضای چند بعدی (ابعاد به تعداد DOF روبات است) و در برابر شکل‌های پیچیده

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

تر موانع (موانع فیزیکی خیلی ساده ممکن است در نتیجه خیلی پیچیده و مجتمع باشند). به عنوان مثال ، اجازه دهید بازوی طراحی شکل ۱ را در نظر بگیریم . موقعیتش در طول موانع ، یکبار که مقادیر زاویه های بین اتصالاتش شناخته شوند، کاملاً مشخص می شود. بنابراین ، برای هر زوج ، امکان دارد که تصمیم گرفت که کجا روبات با موانع محیطی برخورد کند . این چیزی است که ما در شکل ۲ ، برای نمایش نقشه بین موانع فیزیکی در فضای کاری انجام دادیم .

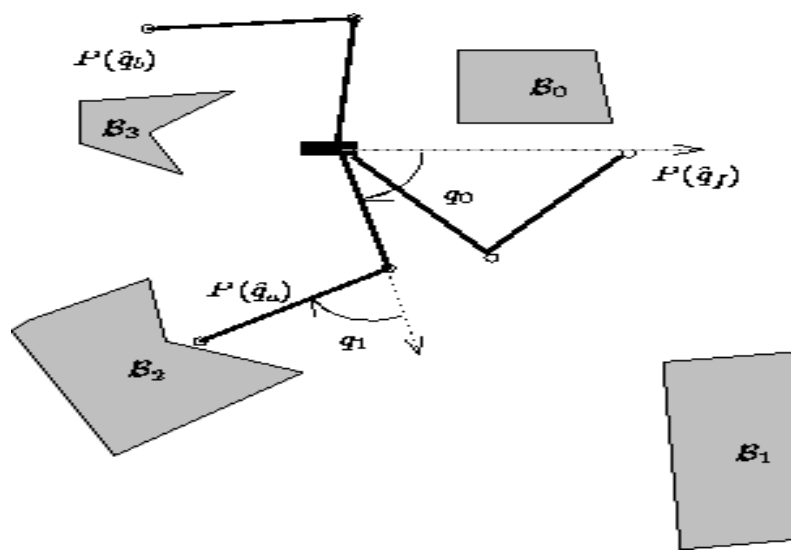
اکنون با حرکت یک نقطه در طول پیوستگی منحنی برای بازوی طراحی بین نقاط مربوطه و در فضای اختیاری یک حرکت آزاد از برخورد تعریف خواهد شد . این منحنی یک راه حل برای مسأله ی طراحی این مسیر خاص می باشد.

روش جدیدی در این زمینه ، به «فضای مسیر گلوله» توجه دارد که در آن تمام مسیر به عنوان یک نقطه ی تنها نمایش داده می شود . عامل هماهنگ کردن این نقطه ، مقادیر پارامترهایی است که حرکات موفق روبات را تعریف می کند.

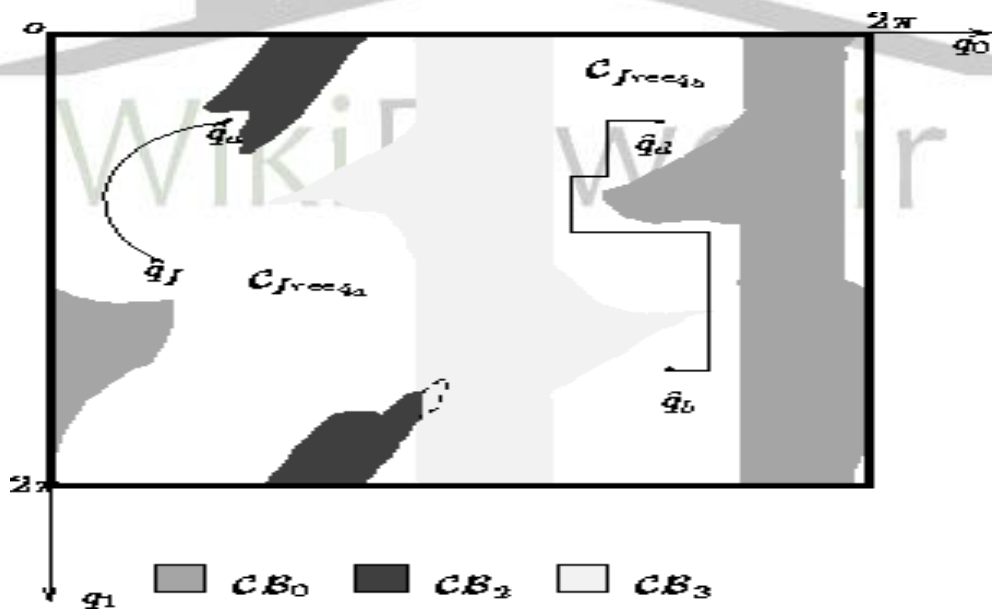
برای نمونه ، فهرست فرمانهای موفق ارسال شده به کنترل کننده ی روبات ، عمیقاً مسیر ورودی روبات را کشف می کند. در این فضا، مسأله طراحی مسیر ، به جستجو برای یک نقطه تنها ، کاسته می شود . یکبار دیگر ما ، ساده سازی مسأله طراحی مسیر را (جستجو برای یک نقطه) در برابر جستجوی فضای چند بعدی (ابعاد فضای مسیر گلوله ، به تعداد پارامترهای مورد نیاز برای ساده سازی کامل تمام مسیر است) عوض کردیم.

برای مثال ، در شکل ۲ ، مسیر مابین ، میتواند توسط نقطه ای در یک فضای هفت بعدی ساده شده که از توجه به طول هفت قطعه بدست آمده نمایش داده می شود.

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم



شکل ۱: یک بازوی DOF که در طول موانع در یک فضای کاری قرار دارد.



شکل: ساختار فضایی مربوط به شکل ۱. توجه کنید.

(۲) به دو ناحیه تقسیم شده است و نمی تواند با یک مسیر پیوسته متصل شده باشد.

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

(۳) در اینجا مانعی وجود ندارد، چرا که مزاحمتی برای بازو ندارد.

روش های کلی :

روش های کلی تر به دو کلاس اصلی تقسیم می شوند:

(i) روشهای قبض و جمع شدن

(ii) روشهای بسط و تجزیه

در روشهای قبض ، سعی می شود ابعاد مسأله آغازین را با ایجاد پیوستگی مجدد روی زیر مجموعه های متعدد فضای شکل گرفته ، کاهش داد. در روشهای تجزیه سعی می شود نواحی فضای ساخته شده را که رها از موانع اند تشخیص دهند و بشناسند. در نهایت هر دو روش با یک گراف کلاسیک که بیش تر از جستجوی یک فضای نظری است تمام می شوند. در اصل این روشها کاملند، چرا که اگر یک مسیر موجود باشد ، مسیر را خواهند یافت و اگر مسیری موجود نباشد در یک زمان محدود تمام می شوند. متأسفانه قبض مجدد یا گراف تجزیه یک مسأله کاملاً NP می باشد: همچنانکه تعداد DOF افزایش می یابد ، پیچیدگی این وظیفه هم بصورت نهایی (Exponentially) رشد می کند.

در نتیجه ، این طراحیها فقط برای رباتهایی به کار می روند که تعداد محدودی DOF (۳ یا ۴) داشته باشند. علاوه بر آن، آنها آهسته اند و فقط می توانند بصورت Offline بکار روند. طراح مدلی از محیط اطراف را به کمک می طلبد . طرحی را ایجاد می کند که به کنترل کننده ربات می رود و در واقع آن را اجرا می کند. معمولاً زمان مورد نیاز برای رسیدن به این هدف به اندازه کافی کوتاه نیست تا اجازه دهد که ربات در یک محیط دینامیک حرکت کند.

طراحی مسیر با طراحیهای محلی:

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

یک راه برای مبارزه با مسئله ، جابجایی کامل شدن در برابر اجرا شدن می باشد . برای انجام این عمل، طراحیهای محلی توسط شیب یک تابع هزینه (معمولاً فاصلهٔ Euclidean به هدف) راهنمایی می شود و شروع به شمارش فشارهایی می کنند که موانع را معرفی می نماید تا از آنها پرهیز کند. از آنجا که مسئله طراحی مسیر NP کامل می باشد با اطلاع از تابع هزینه، معمولاً طراحی یک محیط فریبنده امکان پذیر می گردد که در آن روش در یک حداقل محلی گیر خواهد افتاد. به هر حال این روشها ، کاربردهای صنعتی زیادی دارند ، چرا که می توانند توسط رباتهای مجتمع و مدل‌های محیطی که دارای هزاران صورت هستند به کار گرفته شود، که اغلب زمان زیادی برای روشهای کلی صرف می کند. طراحی مسیر با تکنیک های تصادفی و نشانه های رمزی:

تماس و دسترسی رندوم یا تصادفی ، اولین بار توسط Latombe & Barraquand معرفی شد و بعداً توسط Overmars و اخیراً توسط Kavyaki به کار رفته است . ایدهٔ اصلی در پشت این الگوریتم ها ساخت گرافیکی در فضای شکل گرفته می باشد. گراف به صورت افزایشی به صورت زیر، بدست می آید:

یک طرح محلی به کار گرفته می شود و سعی می کند به هدف برسد. حرکت باید در یک حداقل محلی متوقف شود و یک نود جدید (Node) یا نشانه های مرزی با تولید یک حرکت رندوم که از آن حداقل محلی شروع می شود ، ایجاد می گردد. روش، این دو مرحله را تا تشکیل هدفی که از یکی از این موقعیت های واسطه به آن رسیده است با یک حرکت شیب نزولی تکرار می کند. این الگوریتم ها ، با یک نمایش دلخواه و محتاطانه از فضای تشکیل یافته کار می کنند. این ها به عنوان نمونه های احتمالی کامل شناخته می شوند ، چرا که احتمال دارد با یک راه حل به پایان برسند (یک مسیر یافته شود یا مسیری موجود نباشد) و در یک نقطه به هم می رسند بصورتی که زمان مجاز به سمت بی نهایت افزایش می یابد.

روشهای دیگری هم طراحی شده اند که نشانه های مرزی استفاده می کنند برای مثال ، Sandro که توسط Hwang و Chen معرفی شده است که از نشانه های مرزی استفاده می کنند تا فضای آزاد را تقریبی نماید. این روش مشابه «طراحی سلسله مراتبی» است که در AI به کار رفته است :

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

روش باید در رسیدن به یک هدف دچار خطا شود و هدفهای جزئی تر تولید شوند تا زمانیکه مسئله برای حل شدن کاملاً آسان شود. در سایر روش های آنها، ابتدا یک طراح محلی به کار می رود تا به موقعیت نهایی برسد. طراح محلی باید خطا کند تا فضای شکل گرفته به دو زیر فضا تقسیم شود، که یکی از آنها حاوی هدف و دیگری شامل هدف جزئی تر است. لذا مسئله به دو مسئله کوچکتر تقسیم می شود:

(i) از موقعیت آغازین به هدف جزئی تر می رود.

(ii) از هدف جزئی تر به موقعیت نهایی می رود.

Sandros نشان داده است که برای یافتن مسیر به طرز استادانه به خوبی سازگار شده است این روش برای مسیرهای طراحی شده برای ربات های Puma و Adept تکمیل و تست شده است

طراحی مسیر در فضای مسیر گلوله:

روشهای قبلی اساساً بر اساس فضای شکل گرفته استوار بودند، قبض یا جمع شدن، تجزیه یا بسط، یا بهینه سازی در این فضا ساخته می شود. روش دیگر توجه به «فضای مسیر گلوله» می باشد. به عنوان مثال، Ferbach در روش خود VDP، با توجه به موضوع خط راستی که در شکل آغازین و نهایی را پیوند می دهد شروع کرد. این مسیر بصورت تصاعدی تغییر می کند بطوریکه، نواحی ممنوع برای عبور کاهش می یابد.

در هر تکرار، یک زیر مجموعه چند برابری که شامل مسیر صحیح است، به صورت رندوم تولید میشود. سپس با استفاده از یک روش برنامه ریزی دینامیک، تجدید نظر و جستجو می گذرد که از طول ناحیه ممنوعه به عنوان تابع هزینه برای دستیابی به حداقل استفاده میکند. نتایج جستجو در مسیر گلوله ای

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

جدیدی که با نواحی ممنوعه برخورد میکند از مسیر گلوله ای اصلی کوچکتر است، فرایند تکرار می شود تا یک مسیر گلوله ای مناسب پیدا شود.

همانطوریکه در بخش قبل دیدیم ، همچنین ممکن است محیط های ساده ای را طراحی کرد که این نوع از الگوریتم ها آهسته تر از یک روش رندم خالص می سازد . طرز کار Michalewicz and و Xiao و Lin مشابه این روش است . همانطور که در نسخه قبلی الگوریتم اشاره شد، الگوریتم های ژنتیک برای بهینه سازی از فضای مسیر گلوله به کار می رود.

مسیرهای گلوله ای با استفاده از هماهنگی های نقطه های واسطه پارامتر بندی می شود . یک الگوریتم تکاملی برای بهینه سازی تابع هزینه بکار می رود که بر اساس طول مسیر گلوله و عرض نواحی ممنوعه قرار دارد . اپراتورهای استاندارد الگوریتم های ژنتیک تغییر یافته اند و بعداً برای ایجاد متغیر های بزرگ مسیرها توسعه یافته اند. تعداد نقاط واسطه ثابت شده است و با استفاده از یک روش ذهنی و غیر مستدل انتخاب می شوند . این تعداد داده شده ، ممانعتی برای طراحی یک مسئله ایجاد نمی کنند که برای حل شدن نیاز به نقاط واسطه بیشتری خواهد داشت و الگوریتم را به خطامی کشاند و قتی که یک راه حل وجود دارد.

۷. منابع

[1] J. C. Latombe, "Robot Motion Planning", Norwell, MA: Kluwer, 1991.

[۲] Memetic Algorithm Based Path Planning for a Mobile Robot

Neda Shahidi, Hadi Esmailzadeh, Marziye Abdollahi, Caro Lucas

INTERNATIONAL JOURNAL OF INFORMATION TECHNOLOGY VOLUME 1 NUMBER 3

2004 ISSN 1305-2403

[۳] GENETIC ALGORITHM FOR ROBOT PATH ESTIMATION

Ćurković, P.; Jerbić, B. & Vranješ, B.

THE 17TH INTERNATIONAL DAAM SYMPOSIUM

"INTELLIGENT MANUFACTURING & AUTOMATION: FOCUS ON MECHATRONICS & ROBOTICS"

8-11TH NOVEMBER 2006

[۴] C. Hocaoglu, A.C. Sanderson, "Planning Multiple Paths with Evolutionary Speciation", IEEE Trans.on Evolutionary Computation, Vol.5, No. 3, JUNE2001.

[۵] A Comparison of Genetic Programming and Genetic Algorithms for Auto-tuning Mobile Robot Motion Control

M. Walker, C. H. Messom

Institute of Information and Mathematical Sciences,

Massey University, Albany Campus, Auckland, New Zealand.

fM.G.Walker, C.H.Messomg@massey.ac.nz

[۶] ROBOT PATH PLANNING IN UNSTRUCTURED ENVIRONMENTS USING A KNOWLEDGE-BASED GENETIC ALGORITHM

Simon X. Yang **, Yanrong Hu**

Institute of Computer Science and Technology, Chongqing

University of Posts and Telecommunications, China School of Engineering, University of Guelph, Canada

[۷] D. E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley Publishing Company, 1989.

[۸] "OBSTACLES AVOIDANCE IN A SELF PATH PLANNING

OF A POLAR ROBOT" JUAN CARLOS ROSETE FONSECA¹, EFRÉN GORROSTIETA HURTADO², JOAQUIN PEREZ MENESES³.

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

[۹] *Intelligent Automation and Soft Computing, Vol. 10, No. 1, pp. 51-64, 2004 Copyright © 2004, TSI®Press Printed in the USA. All rights reserved 51*

MOBILE ROBOT PATH PLANNING USING HYBRID GENETIC ALGORITHM AND TRAVERSABILITY VECTORS METHOD

C.-K. LOO^{1,2} M. RAJESWARI² E. K. WONG¹ M. V. C. RAO¹

¹
Faculty of Engineering and Technology Multimedia University (Melaka)75450 Melaka, Malaysia

e-mail: ckloo@mmu.edu.my

²
School of Industrial Technology Universiti Sains Malaysia Penang, Malaysia e-mail: mandava@cs.usm.my

[1۰] A Comparison of Genetic Programming and Genetic Algorithms for Auto-tuning Mobile Robot Motion Control

M. Walker, C. H. Messom Institute of Information and Mathematical Sciences, Massey University, Albany Campus, Auckland, New Zealand. {M.G.Walker, C.H.Messom}@massey.ac.nz

شبه کد Matlab

```

clc
clear all
c=0;
b1=0;
coordinate=zeros(2,16);
chromosome=char(1,36);
population=char(6,36);
childs=char(2,36);
childs_dec=zeros(2,9);
fit_childs=zeros(2,1);
result=char(1,36);
fitness=zeros(9,1);
result_dec=zeros(1,9);
population_dec=zeros(6,9);
con_str_point=char(1,4);
[R C]=size(chromosome);
chromosome_length=36;
b=char(1,4);

```

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

```

k=char(chromosome);
coordinate=[1 1 3 3 5 6 6 8 10 10 10 11 13 12 14 14
            7 11 14 1 8 11 4 4 1 7 11 14 12 2 3 8 ];

asscipoint={'0000','0001','0010','0011','0100'
            ',0101','0110','0111','1000','1001','1010','1011','1100','1101','1110','1111'};

[R1 C1]=size(coordinate);
m=7;
population_num=6;
p_cross=100;
p_mutation=0.005;
iteration_num=50;
point_start=0;
point_goal=15;
ps = dec2bin(point_start,4);
pg = dec2bin(point_goal,4);
ps_str = num2str(ps);
pg_str = num2str(pg);
chromosome(1,1:4)=ps_str(1:4);
chromosome(1,33:36)=pg_str(1:4);
loc1=99;

%#####initaial
population#####

Sw=0;
While (sw < population_num)
for i=1:population_num

for j=5:32
n=randint;
str_n=num2str(n);
chromosome(1,j)=str_n;
end

```


برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

```
%#####create first
population#####
```

```
for l1=1:36
population(i,l1)=chromosome(1,l1);
end
```

```
%#####calculate fitness
#####
```

```
l3=2;
```

```
for l2=5:4:32
con_str_point(1,1:4)=population(i,l2:l2+3);
b=con_str_point;
```

```
for ii=1:16
vc=asscpoint(ii);
c=strncmpi(b,vc,4);
if (c(1,1)==1)
loc1=ii-1;
end
end
%loc1
```

```
% ##### put loc1 in population_dec
#####
```

```
population_dec(i,1)=0;
```

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

```

    population_dec(i,9)=15;
    population_dec(i,l3)=loc1;
    l3=l3+1;
end % for l2 5:32

% #####change the repeat
point#####

for l7=1:6
    for l15=1:8
        for l16=l15+1:9
            if (population_dec(l7,l15)==population_dec(l7,l16))

                for l17=l15:l16
                    population_dec(l7,l17)=population_dec(l7,l15);
                end % for l17

                for l18=l15:l16-1
                    for l19=1:4
                        loc_l15=l18*4+l19;
                        loc_l16=(l16-1)*4+l19;
                        population(l7,loc_l15)=population(l7,loc_l16);
                    end % for l19
                end %for l18
            end %if ==

        end% forl16
    end % for l15
end %for l7

% ##### calculate distance and fitness
#####

swit=0;
for l4=1:m+1
    cur_op=population_dec(i,l4);
    next_op=population_dec(i,l4+1);

switch cur_op

```

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

```

case 0
  if ((next_op~=1)&(next_op~=3)&(next_op~=4)) swit=-1; end
case 1
  if ((next_op~=0)&(next_op~=2)) swit=-1; end
case 2
  if ((next_op~=1)&(next_op~=5)&(next_op~=8)) swit=-1; end
case 3
  if ((next_op~=0)&(next_op~=6)&(next_op~=8)) swit=-1; end
case 4
  if ((next_op~=0)&(next_op~=5)&(next_op~=6)) swit=-1;end
case 5
  if ((next_op~=2)&(next_op~=4)&(next_op~=10)) swit=-1; end
case 6
  if ((next_op~=3)&(next_op~=4)&(next_op~=7)) swit=-1; end
case 7
  if ((next_op~=6)&(next_op~=9)&(next_op~=13)) swit=-1;end
case 8
  if ((next_op~=3)&(next_op~=13)) swit=-1;end
case 9
  if ((next_op~=7)&(next_op~=10)&(next_op~=15)) swit=-1;end
case 10
  if ((next_op~=5)&(next_op~=9)&(next_op~=11)) swit=-1;end
case 11
  if ((next_op~=2)&(next_op~=10)&(next_op~=12)) swit=-1;end
case 12
  if ((next_op~=11)&(next_op~=15)) swit=-1; end
case 13
  if ((next_op~=7)&(next_op~=8)&(next_op~=14)&(next_op~=15)) swit=-1;end
case 14
  if ((next_op~=13)&(next_op~=15)) swit=-1;end
case 15
  if ((next_op~=9)&(next_op~=12)&(next_op~=13)&(next_op~=14)) swit=-1;end
end %switch
swit
end % end l4

if (swit==0)

```

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

```

sum=0;
  for q=1:m+1
    xq1=coordinate(1,population_dec(i,q)+1);
    xq2=coordinate(1,population_dec(i,q)+1);
    yq1=coordinate(2,population_dec(i,q)+1);
    yq2=coordinate(2,population_dec(i,q)+1);
    dis=sqrt(((xq1-xq2)^2 )+((yq1-yq2)^2));
    sum=sum+dis;
  end
  fitness(i,1)=1/sum ;
sw=sw+1

elseif (swit==-1)
  fitness(i,1)=0;
end %if
  fitness
end % for i polpulation
end %while

%#####iteration#####
#####

iterate=0;

for ite=1:iteration_num

% #####selection two chromosome with roulette_wheel
#####

[index_parent_1,index_parent_2]=mselection_roulette_wheel(population,fitness,population_num);

% #####crossover 2-point
#####

```

برای دریافت فایل Word پروژه به سایت **ویکی پاور** مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

```
p_cross=100;
rand_c=rand;
if rand_c<p_cross
[child_1,child_2]=mcrossover(fitness,population,chromosome_length,index_parent_1,i
ndex_parent_2);
else
continue;
end
```

```
% #####mutation
#####
```

```
[child_1,child_2]=mmutation(chromosome_length,child_1,child_2);
```

```
for l8=1:36
childs(1,l8)=child_1(1,l8);
childs(2,l8)=child_2(1,l8);
end
```

```
%#####fitness
#####
```

```
for l7=1:2
```

```
loc1=0;
l3=2;
```

```
for l5=5:4:32
```

```
con_str_point(1,1:4)=childs(l7,l5:l5+3);
b=con_str_point;
```

```
for ii=1:16
```

```
vc=asscipoint(ii);
c=strncmpi(b,vc,4);
if (c(1,1)==1)
```

برای دریافت فایل Word پروژه به سایت **ویکی پاور** مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

```
loc1=ii-1;
end
end
%loc1
```

```
% #####
population_dec##### put loc1 in
```

```
childs_dec(i,1)=0;
childs_dec(i,9)=15;
```

```
childs_dec(i,l3)=loc1;
l3=l3+1;
end % for l5 5:32
% #####change the repeat
point#####
```

```
for l15=1:8
for l16=l15+1:9
```

```
if (childs_dec(l7,l15)==childs_dec(l7,l16))
fff=l15;
fff1=l16;
for l17=l15:l16
childs_dec(l7,l17)=childs_dec(l7,l15);
end % for l17
```

```
for l18=l15:l16-1
for l19=1:4
loc_l15=l18*4+l19;
loc_l16=(l16-1)*4+l19;
childs(l7,loc_l15)=childs(l7,loc_l16);
end % for l19
end %for l18
```

```
end %if ==
```

```
end% forl16
```

برای دریافت فایل Word پروژه به سایت **ویکی پاور** مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

end % for l15

```
% #####
fitness##### calculate distance and
```

```
swit=0;
for l4=1:m+1
    cur_op=childs_dec(i,l4);
    next_op=childs_dec(i,l4+1);

switch cur_op
case 0
    if ((next_op~=1)&(next_op~=3)&(next_op~=4)) swit=-1; end
case 1
    if ((next_op~=0)&(next_op~=2)) swit=-1; end
case 2
    if ((next_op~=1)&(next_op~=5)&(next_op~=8)) swit=-1; end
case 3
    if ((next_op~=0)&(next_op~=6)&(next_op~=8)) swit=-1; end
case 4
    if ((next_op~=0)&(next_op~=5)&(next_op~=6)) swit=-1;end
case 5
    if ((next_op~=2)&(next_op~=4)&(next_op~=10)) swit=-1; end
case 6
    if ((next_op~=3)&(next_op~=4)&(next_op~=7)) swit=-1; end
case 7
    if ((next_op~=6)&(next_op~=9)&(next_op~=13)) swit=-1;end
case 8
    if ((next_op~=3)&(next_op~=13)) swit=-1;end
case 9
    if ((next_op~=7)&(next_op~=10)&(next_op~=15)) swit=-1;end
case 10
    if ((next_op~=5)&(next_op~=9)&(next_op~=11)) swit=-1;end
case 11
    if ((next_op~=2)&(next_op~=10)&(next_op~=12)) swit=-1;end
case 12
```

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

```

    if ((next_op~=11)&(next_op~=15)) swit=-1; end
case 13
    if ((next_op~=7)&(next_op~=8)&(next_op~=14)&(next_op~=15)) swit=-1;end
case 14
    if ((next_op~=13)&(next_op~=15)) swit=-1;end
case 15
    if ((next_op~=9)&(next_op~=12)&(next_op~=13)&(next_op~=14)) swit=-1;end
end %switch
swit;
end % end l4

```

```

if (swit==0)
    sum=0;
    for q=1:m+1
        xq1=coordinate(1,childs_dec(l7,q+1)+1);
        xq2=coordinate(1,childs_dec(l7,q)+1);
        yq1=coordinate(2,childs_dec(l7,q+1)+1);
        yq2=coordinate(2,childs_dec(l7,q)+1);
        dis=sqrt(((xq1-xq2)^2)+(yq1-yq2)^2);
        sum=sum+dis;
    end
    fit_childs(l7,1)=1/sum ;
else
    fit_childs(l7,1)=0;
end %if

```

```
end %end l7
```

```

%#####accepting &
Replace#####

```

```

[population,fitness]=maccepting_Replace(population,fitness,index_parent_1,index_par
ent_2,child_1,child_2,fit_childs);

```


برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

```

end %end for num_iteration
fitness
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%          RESULT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[min_fit,index]=max(fitness);
% display(population(index,:))

index;

for l9=1:36
    result(1,l9)=population(index,l9);
end
result;
loc1=0;
l3=2;

for l5=5:4:32
    con_str_point(1,1:4)=result(1,l5:l5+3);
    b=con_str_point;

    for ii=1:16
        vc=asscpoint(ii);
        c=strncmpi(b,vc,4);
        if (c(1,1)==1)
            loc1=ii-1;
        end
    end
    %loc1

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%          put          loc1          in
population_dec%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

result_dec(1,1)=0;

```

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

```
result_dec(1,9)=15;
```

```
result_dec(1,l3)=loc1;
l3=l3+1;
end % for l5 5:32
```

```
display(result(1,:));
display(result_dec(1,:));
```

```
%@@@@@@@@@@@@@
@@@@@@@@@@@@@
```

FUNCTIONS

```
function [index_parent_1,index_parent_2]=
mselection_roullet_wheel(population,fitness,population_num)
```

```
fit=fitness;
s=zeros(population_num,1);
```

```
sum_fit=sum(fit);
for i=1:population_num
b(i,1)=fit(i,1)./sum_fit;
```

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

```
end
t=0;
for i=1:population_num
    t=b(i,1)+t;
    sb(i,1)=t;
end
for i=1:population_num
    random_num(i,1)=rand;
end
for i=1:population_num
    for j=1:population_num
        if random_num(i,1)<sb(j,1)
            s(i,1)=j;
            break
        end
    end
end
end

iteration=zeros(population_num,1);
for i=1:population_num
    for j=1:population_num
        if (s(j,1)==i)
            iteration(i,1)=iteration(i,1)+1;
        end
    end
end
end

[max_1,index_parent_1]=max(iteration)
iteration(index_parent_1,1)=0;

[max_2,index_parent_2]=max(iteration)
```

برای دریافت فایل Word پروژه به سایت **ویکی پاور** مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

```
function[child_1,child_2]=mcrossover(fitness,population,chromosome_length,index_pa
rent_1,index_parent_2)
temp=0;
point1 = randint(1,1,[5,chromosome_length-4])
point2 = randint(1,1,[5,chromosome_length-4])

parent_1=population(index_parent_1,:)
parent_2=population(index_parent_2,:)

if (point1>point2)
    temp=point1;
    point1=point2;
    point2=temp;
end

for i=1:point1
    child_1(1,i)=parent_1(1,i);
    child_2(1,i)=parent_2(1,i);
end

for i=point1+1:point2
    child_1(1,i)=parent_2(1,i);
    child_2(1,i)=parent_1(1,i);
end

for i=point2+1:chromosome_length
    child_1(1,i)=parent_1(1,i);
    child_2(1,i)=parent_2(1,i);
end
```

برای دریافت فایل Word پروژه به سایت **ویکی پاور** مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

```
function [child_1,child_2]=mmutation(chromosome_length,child_1,child_2(

p_mutation=0.005;

for i=5:chromosome_length-4
    rand_m1=rand;
    if rand_m1<p_mutation
        child_1(1,i)=randint;
    end

    rand_m2=rand;
    if rand_m2<p_mutation
        child_2(1,i)=randint;
    end
end
function[population,fitness]=maccepting_Replace(population,fitness,index_parent_1,in
dex_parent_2,child_1,child_2,fit_childs);
fitness_parent_1=fitness(index_parent_1,1);
fitness_parent_2=fitness(index_parent_2,1);
check_min_fit(1,1)=fitness_parent_1;
check_min_fit(1,2)=fitness_parent_2;
check_min_fit(1,3)=fit_childs(1,1);
check_min_fit(1,4)=fit_childs(2,1);
[min_1,index_min_1]=max(check_min_fit);
check_min_fit(1,index_min_1)=min(check_min_fit)+1;
[min_2,index_min_2]=max(check_min_fit);

if index_min_1 == 1
    population(index_parent_1,:)= population(index_parent_1,:);
    fitness(index_parent_1,1)=fitness(index_parent_1,1);
end

if index_min_1 == 2
    population(index_parent_1,:)= population(index_parent_2,:);
    fitness(index_parent_1,1)=fitness(index_parent_2,1);
end
```

برای دریافت فایل Word پروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازم

```
if index_min_1 == 3
    population(index_parent_1,:)= child_1;
    fitness(index_parent_1,1)=fit_childs(1,1);
end
```

```
if index_min_1 == 4
    population(index_parent_1,:)= child_2;
    fitness(index_parent_1,1)=fit_childs(2,1);
end
```

```
if index_min_2 == 1
    population(index_parent_2,:)= population(index_parent_1,:);
    fitness(index_parent_2,1)=fitness(index_parent_1,1);
end
```

```
if index_min_2 == 2
    population(index_parent_2,:)= population(index_parent_2,:);
    fitness(index_parent_2,1)=fitness(index_parent_2,1);
end
```

```
if index_min_2 == 3
    population(index_parent_2,:)= child_1;
    fitness(index_parent_2,1)=fit_childs(1,1);
end
```

```
if index_min_2 == 4
    population(index_parent_2,:)= child_2;
    fitness(index_parent_2,1)=fit_childs(2,1);
end
```