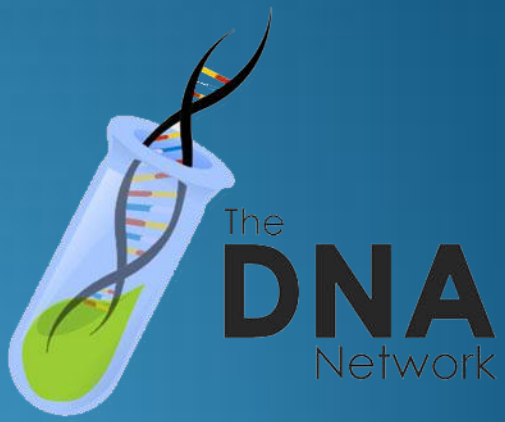
هوش مصنوعی

مدرس : احمد ابدالی

فصل چهارم

جستجوی آگاهانه یا جستجوی هیورستیک



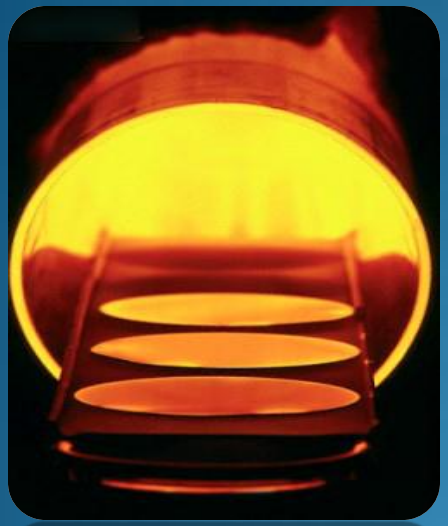
Ahmad Abdali Mohamadi

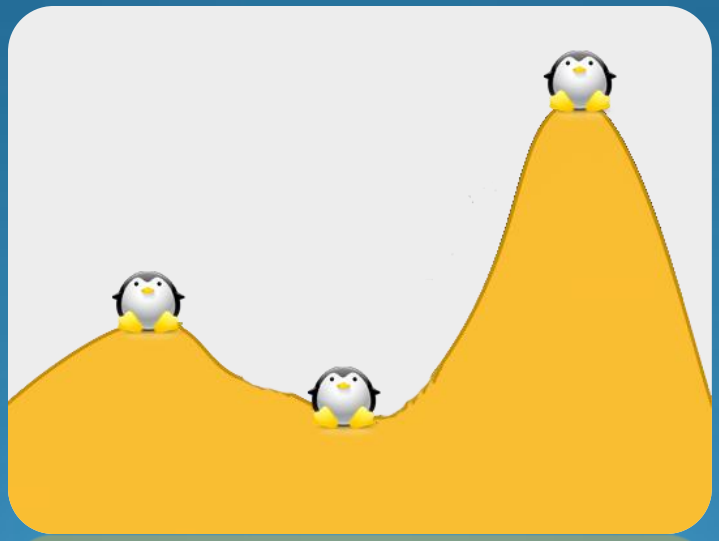
فصل چهارم قسمت دوم

الگوریتم های جستجوی محلی

الگوریتم تکاملی

الگوریتم های جست و جوی محلی و بهینه سازی





Ahmad Abdali Mohamadi

انواع روش های جستجو

جستجوی نا آگاهانه یا کور

جستجوی آگاهانه یا هیورستیک

جستجوی متاهیوریستیک

جستجوی متا هیورستیک

مسائلی وجود دارند که فضای جستجوی آنها به اندازه ای بزرگ می باشد که استفاده از روش های جستجوی ناآگاهانه و هیوریستیکی را برایمان غیرممکن می سازند.

در این موارد از روش های متاهیوریستیکی یا فرا اکتشافی استفاده می کنیم .

Ahmad Abdali Mohamadi

الگوريتم هاي اصلاح تكراري يا جستجوی محلی

دسته اي از مسائل وجود دارد كه در آنها مسير رسيدن به جواب مهم نيست بلكه فقط جواب نهايي اهميت دارد همانند مسئله ۸ وزير. ايده اصلی در اين الگوريتم ها اين است كه وقتي از حالت قبلي به حالت فعلي برسيم ديگر حالت قبلي را فراموش كنيم لذا در اين مسائل درخت نداريم بلكه تنها يك وضعيت فعلي داريم كه خودش شامل تمام اطلاعات مورد نياز مسئله است كه در اين الگوريتم ها از يك ساختار كامل شروع شده و سپس با انجام تغييراتي در آن سعي در اصلاح كيفيت آن ساختار داريم.

دو مزیت اصلی این الگوریتم ها :

۱) از حافظه کمی استفاده می کنند .

۲) در فضای بزرگ بر خلاف الگوریتم های سیستماتیک راه حل خوبی پیدا می کنند .

Ahmad Abdali Mohamadi

اصول کلی الگوریتم هاي اصلاح تکراري یا جستجوی محلی

ایده اصلی به این صورت است وقتی از حالت قبلی به حالت فعلی رسیدیم حالت قبلی را فراموش کنیم . پس در این مسائل چیزی مثل درخت نداریم .
بلکه تنها یک وضعیت داریم که
شامل تمام اطلاعات مورد نیاز مسله است .

شعار این الگوریتم ها :

از یک حالت شروع کن و با انجام تغیراتی در آن سعی در اصلاح کیفیت کن (تکامل داروینی)

نتیجه گیری اخلاقی !!

الگوریتم های قبلی ، فضای جست و جو را به طور سیستماتیک بررسی میکنند

تا رسیدن به هدف یک یا چند مسیر نگهداری میشوند
مسیر رسیدن به هدف ، راه حل مسئله را تشکیل میدهد

در الگوریتم های محلی مسیر رسیدن به هدف مهم نیست

مثال: مسئله ۸ وزیر

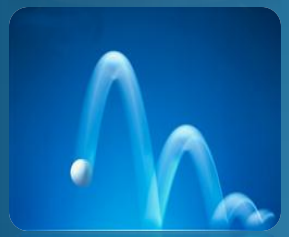
دو امتیاز عمده جست و جوهای محلي

استفاده از حافظه کمکی
ارائه راه حلهای منطقي در فضاهای بزرگ و نامتناهی

انواع الگوريتم هاي جستجوی محلی

تپه نوردی

گرم و سرد



جستجوی پرتویی

ژنتیک

مسائل بهینه سازی

هدف یافتن بهترین حالت در بین چند حالت است

Ahmad Abdali Mohamadi

فضای حالت الگوریتم های جستجوی محلی

Ahmad Abdali Mohamadi

الگوریتم تپه نوردی

الگوریتم تپه نوردی بدین صورت است که ابتدا جوابی به شکل تصادفی برای مساله تولید
می شود. سپس در یک حلقه و تا زمانی که شرط توقف الگوریتم برقرار نشده است ، به طور مکرر تعدادی همسایه برای حالت فعلی تولید شده و از میان حالت های همسایه بهترین آن ها انتخاب شده و جایگزین حالت فعلی می شود .
در حالت کلی بهینگی جوابی که این الگوریتم پیدا می کند محلی است. لازم به ذکر است که منظور از بهینگی مینیمم سازی یا ماکزیمم سازی یک مسئله بهینه سازی می باشد.

Ahmad Abdali Mohamadi

الگوریتم تپه نوردی

به عبارت دیگر : در الگوریتم تپه نوردی ابتدا حالت اولیه به شکل تصادفی شروع می شود سپس پس از ایجاد حالت های ممکن بهترین حالت را انتخاب می کند و به آن حالت می رود .

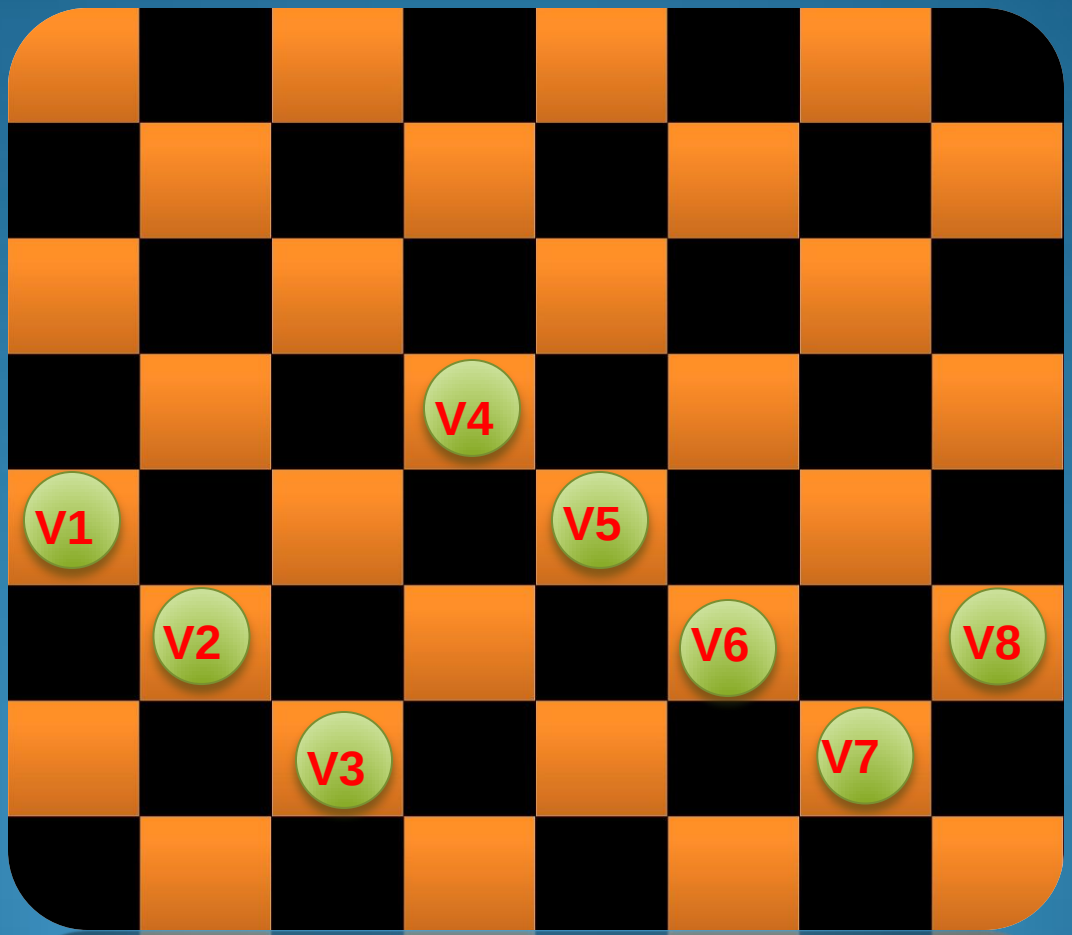
Ahmad Abdali Mohamadi

الگوریتم تپه نوردی مانند پیدا کردن قله کوه اورست در مه غلیظ است !

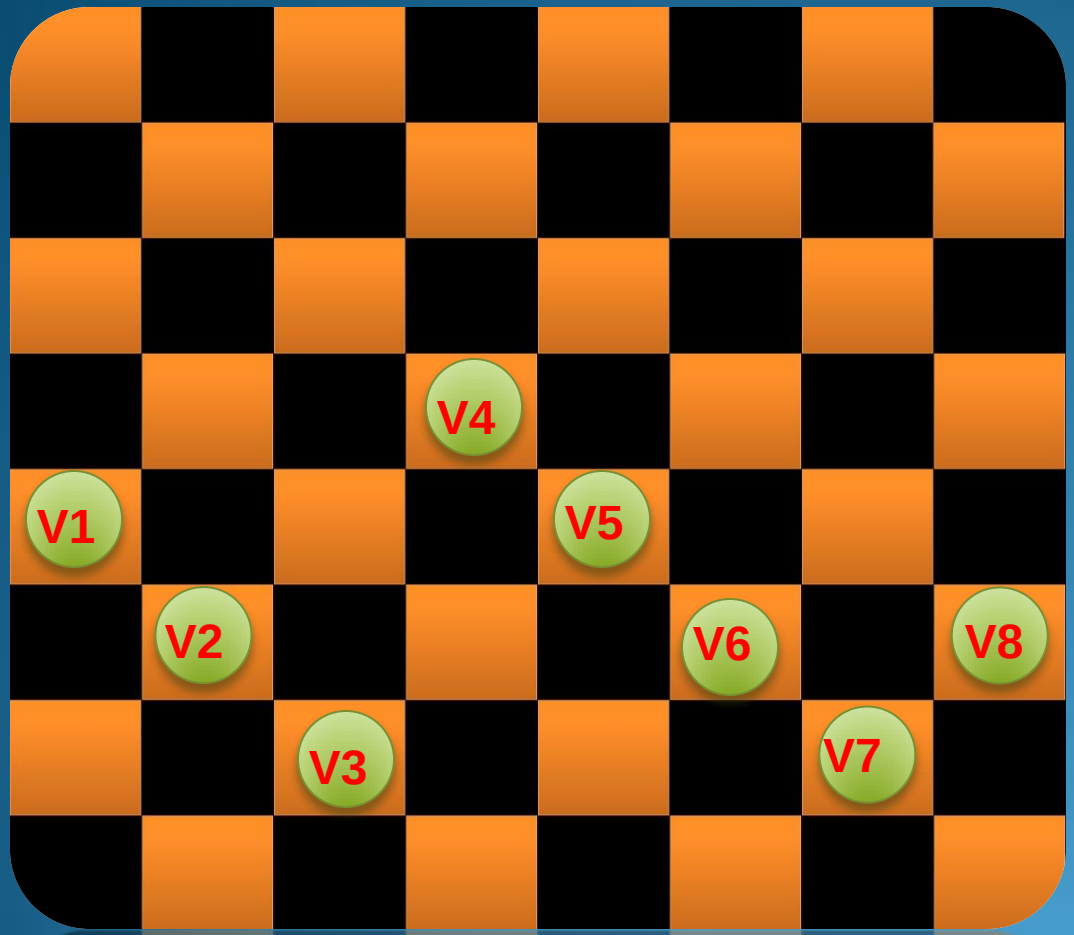
هشت وزیر با الگوریتم تپه نوردی

در ابتدا سعی میکنیم یک تابع هیورستیک را برای مسله تعریف کنیم .

تابع هیورستیک = تعداد برخوردهای مستقیم و غیر مستقیم بین جفت وزیر ها



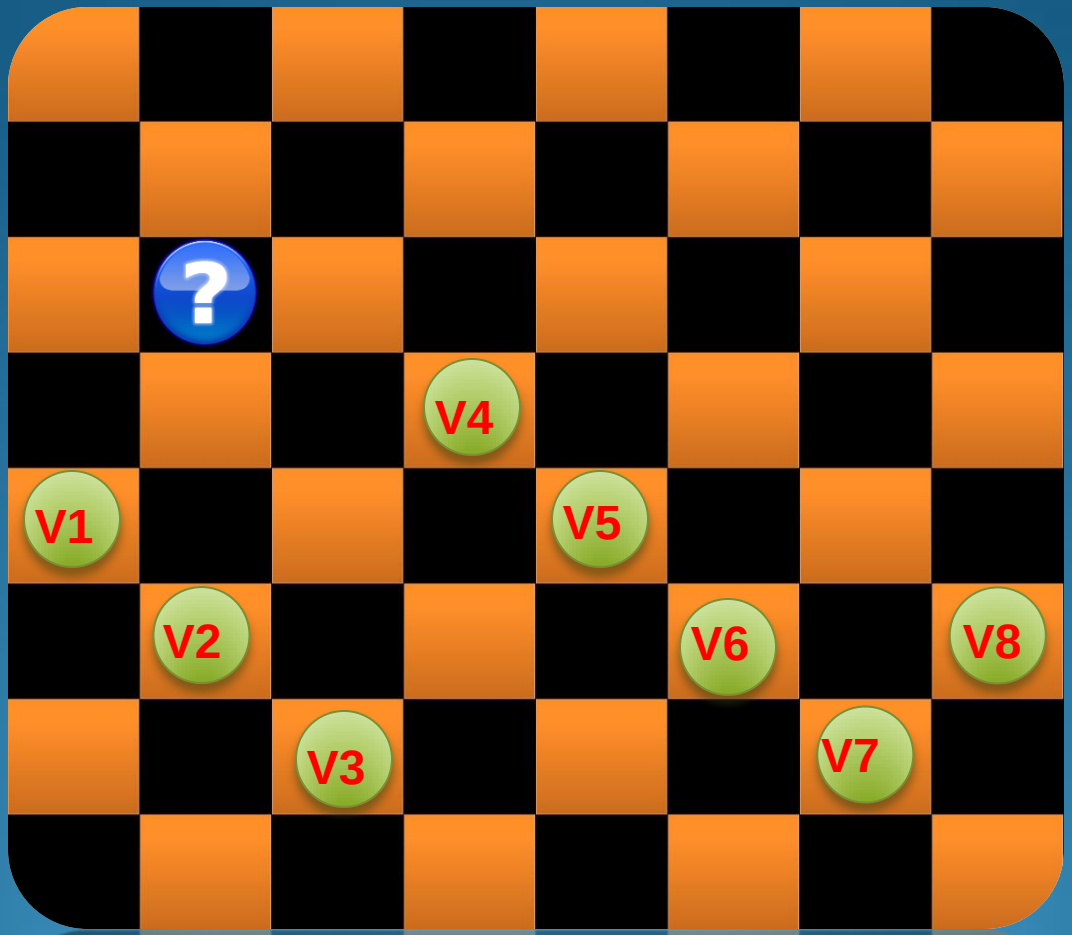
تابع هیورستیک برای هشت وزیر



h(این حالت) = 17

{ V1 , V2 }
{ V1 , V3 }
{ V1 , V5 }
{ V2 , V3 }
{ V2 , V4 }
{ V2 , V6 }
{ V2 , V8 }
{ V3 , V5 }
{ V3 , V7 }
{ V4 , V5 }
{ V4 , V6 }
{ V4 , V7 }
{ V5 , V6 }
{ V5 , V7 }
{ V6 , V7 }
{ V6 , V8 }
{ V7 , V8}

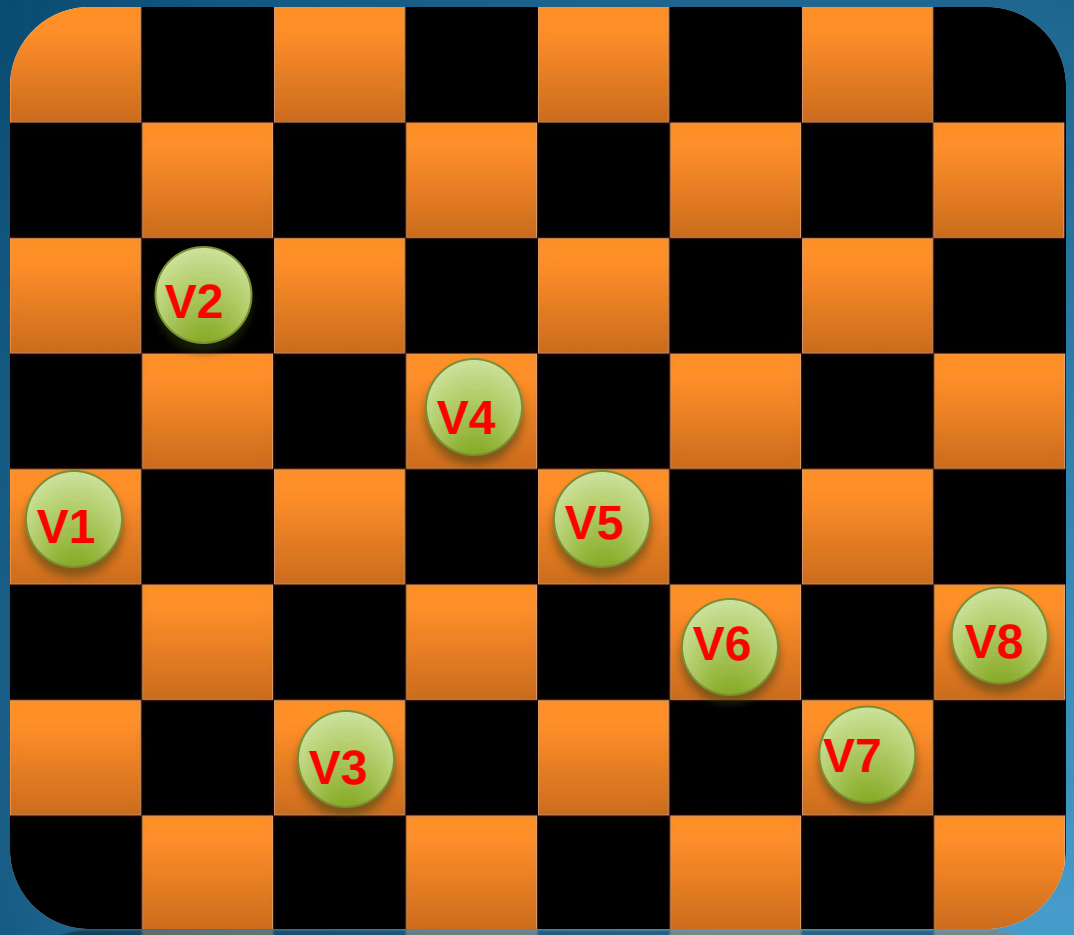
تابع هیورستیک برای هشت وزیر



h(این حالت) = ?

Ahmad Abdali Mohamadi

تابع هیورستیک برای هشت وزیر



h(این حالت) = 12

{ V1 , V3 }

{ V1 , V5 }

{ V3 , V5 }

{ V3 , V7 }

{ V4 , V5 }

{ V4 , V6 }

{ V4 , V7 }

{ V5 , V6 }

{ V5 , V7 }

{ V6 , V7 }

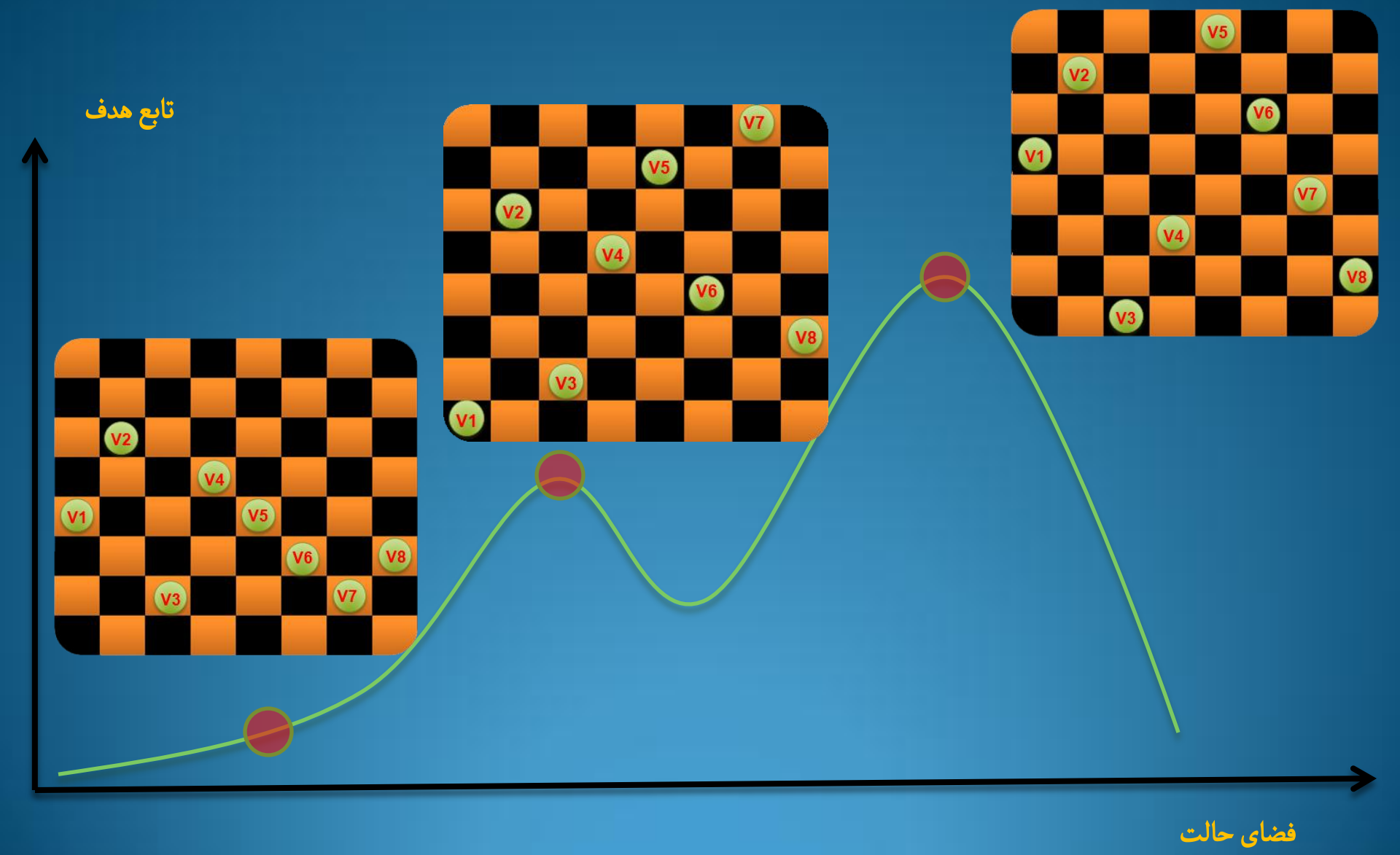
{ V6 , V8 }

{ V7 , V8 }

تابع هیورستیک برای هشت وزیر

Ahmad Abdali Mohamadi

نمودار قسمتی از فضای حالت هشت وزیر با الگوریتم تپه نوردی



Ahmad Abdali Mohamadi

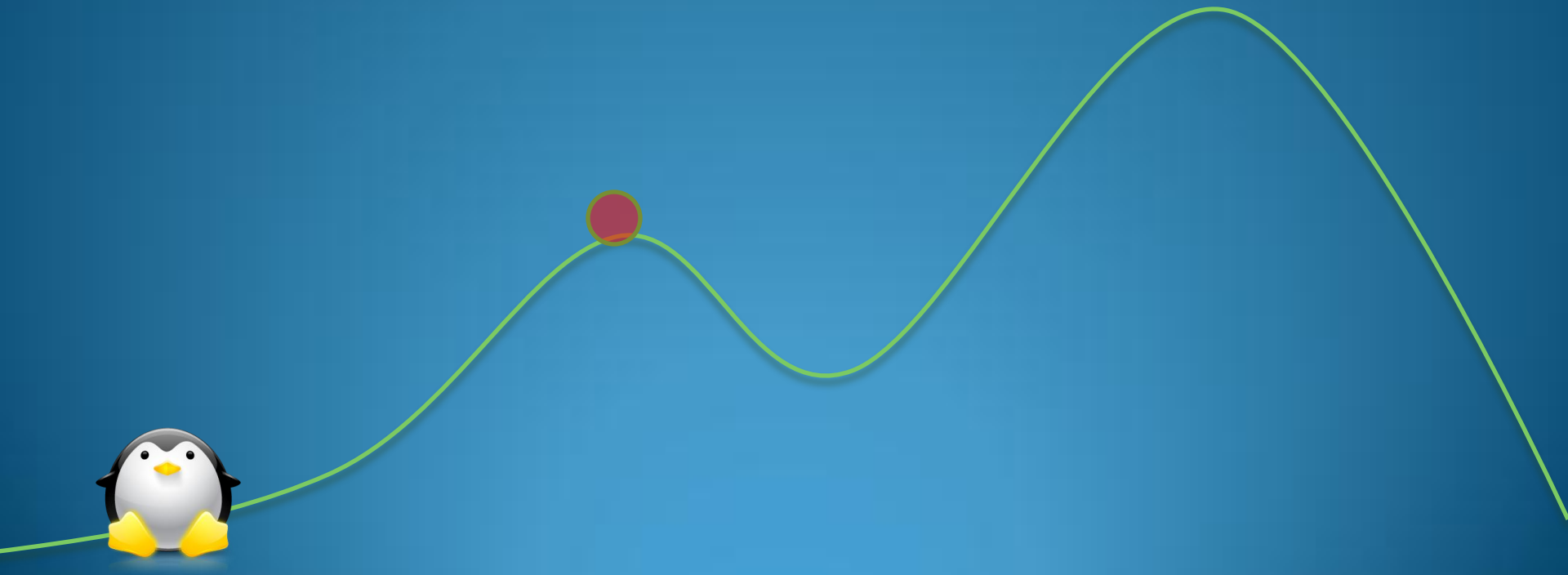
خبر های بد !!

الگوریتم تپه نوردی در سه مورد زیر با مشکل مواجه می شود :

تیغه　　فلات　　ماکزیمم محلی

علل ناکامی الگوریتم تپه نوردی

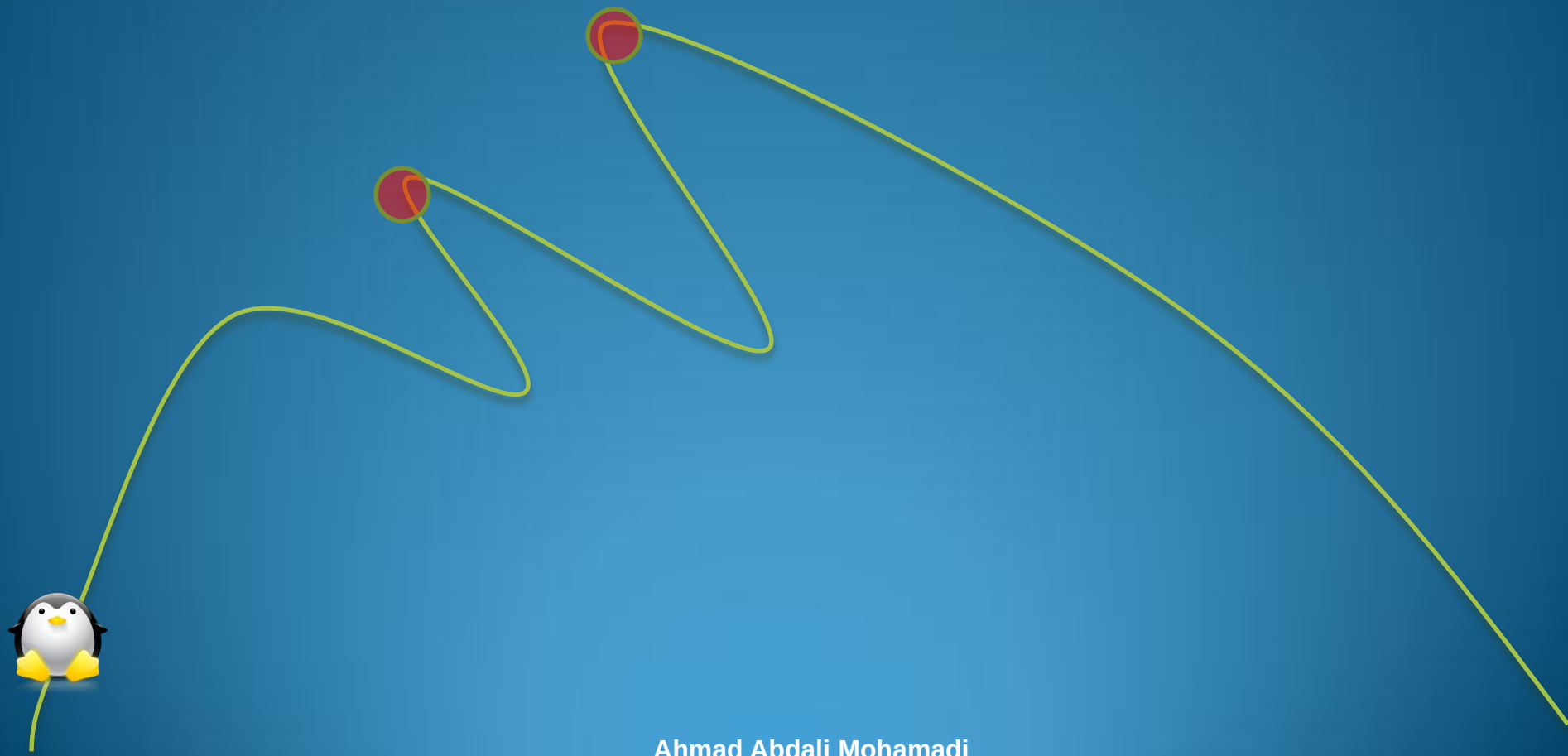
ماکزیمم محلی : حالتي است كه از تمامي حالات همسايه خود بهتر است (حالت جاري از همسايه هاي هم سطح خود بهتر است) اما از چند حالت بعدتر از اين همسايه ها بهتر نيست. در يك ماكزيمم محلي هر حركتي به نظر اشتباه به نظر مي آيد معمولاً در نزديكي حالت نهايي اين حال اتفاق مي افتد



Ahmad Abdali Mohamadi

علل ناکامی الگوریتم تپه نوردی

تیغه : حالتي است که باعث ایجاد چند ماکزیمم محلی می شود و عبور از آنها دشوار است .



علل ناکامی الگوریتم تپه نوردی

فلات : ناحیه ای است که مقدار تابع ارز یابی در آن ثابت است .

دو نوع فلات داریم :

۱) ماکزیمم محلی صاف

۲) شانه

بر طرف کردن ایرادات الگوریتم تپه نوردی

روش هايي براي برخورد با اين موانع ذكر شده وجود دارد اگرچه هيچ يك تضميني براي حل قطعي مشكل نمي دهند.

روش اول : يك پرش بلند به يك جهت ديگر و بررسي حالات جديد.

روش دوم : به صورت تصادفي نقطه ديگري از فضاي حالت را انتخاب نموده و كار را ادامه مي دهيم اين روش براي حل مشكل نوك قله موثر است.

روش سوم : عقبگرد به مراحل قبلي و تلاش براي حركت در يك جهت ديگر. براي پياده سازي اين روش لازم است كه همواره ليستي از مسيرهاي مطلوب ديگر ذخيره كنيم و چنانچه مسير فعلي به يك نقطه كور رسيد به يكي از آنها عقبگرد كرده و جستجو را ادامه دهيم. اين روش براي برخورد با مشكل ماكزيمم محلي نسبتاً خوب است.

نکات مهم الگوریتم تپه نوردی

نکته ۱ : الگوريتم تپه نوردي در صورتي كامل است كه درگير سه وضعيت ذكر شده نگردد.

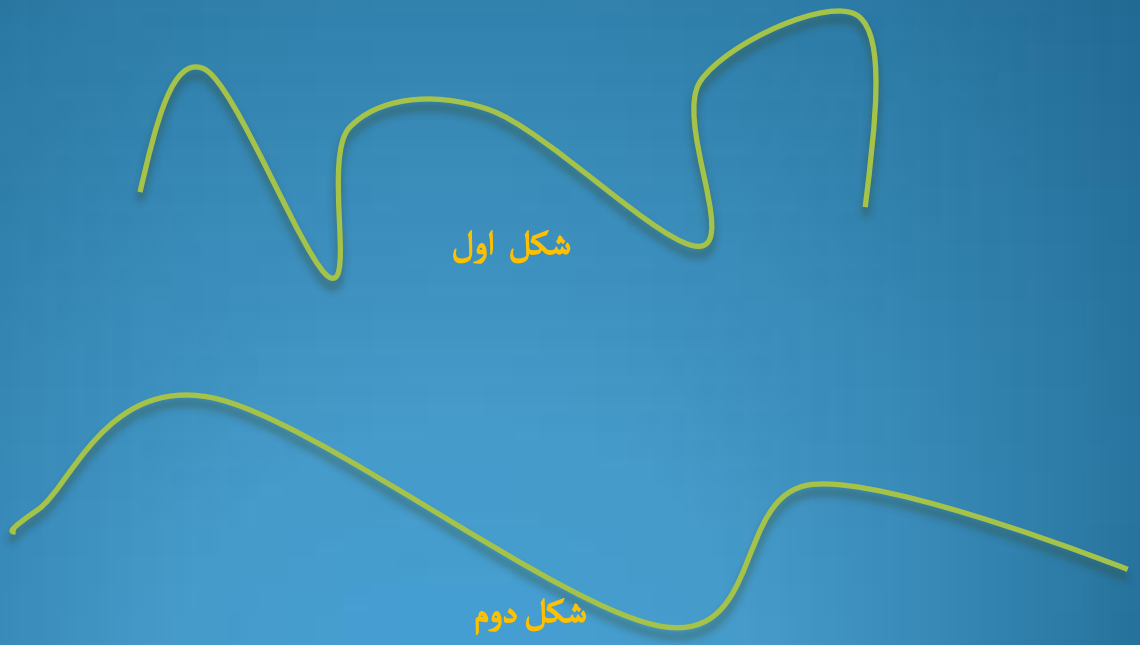
نکته ۲ : الگوريتم تپه نوردي ممكن است در يافتن پاسخ مسئله شكست بخورد بدين شكل كه هيچ كدام از حالات بعدي بهتر از حالت جاري نبوده و حالت جاري نيز يك حالت هدف نباشد

نکته ۳ : الگوريتم تپه نوردي همانند الگوريتم حريصانه است با اين تفاوت كه امكان بازگشت به عقب وجود ندارد.

نکات مهم الگوریتم تپه نوردی

نکته ۴ : موفقیت الگوریتم تپه نوردی فضای حالت مسله بستگی دارد در حالت کلی هر چقدر تعداد ماکزیمم محلی و فلات کمتر باشد تپه نوردی در
شروع تصادفی موفق تر عمل می کند .

در شکل دوم تپه نوردی موفق تر است



شبه کد الگوریتم تپه نوردی

```
Function HillClimbing()
{
  Generate a solution ( Next_State )
  Best = Next_State
  Loop
    Start = Best
    Next_States = Neighbors (Start)
    Best = SelectBest (Next_States)
  Until stop criterion satisfied
}
```

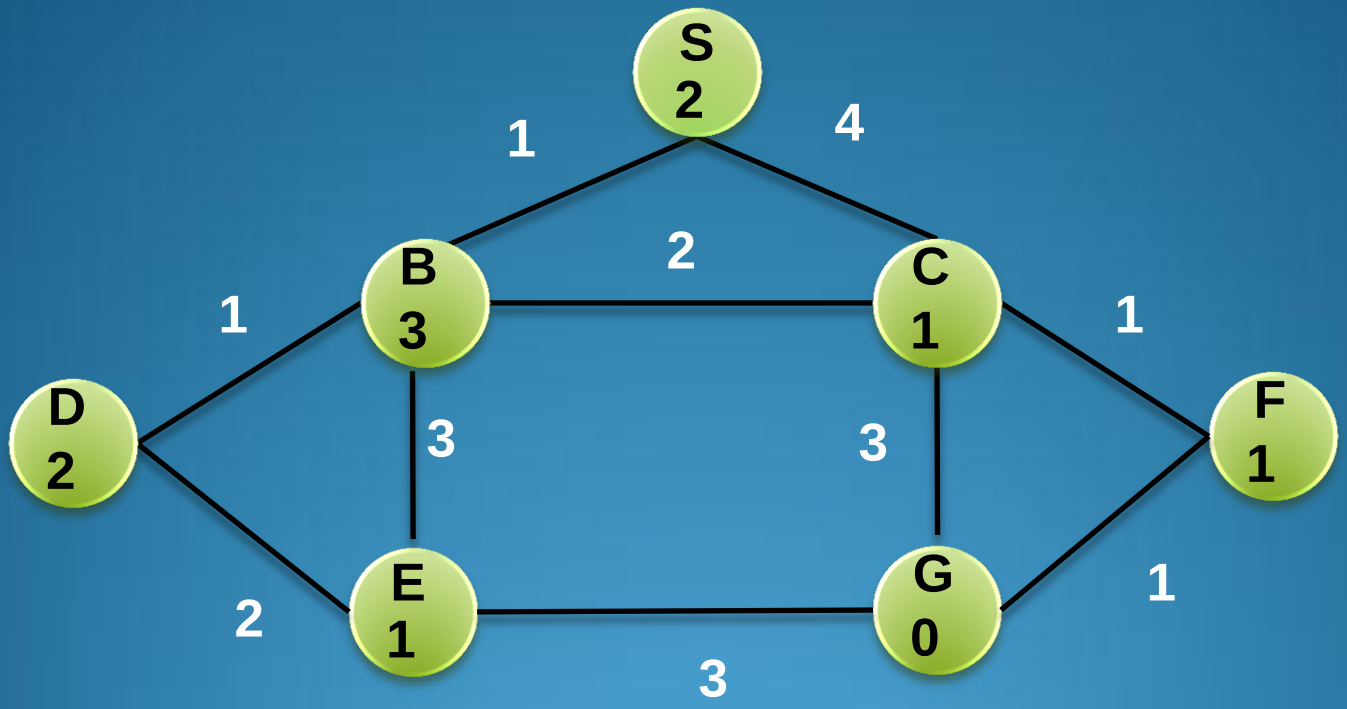
الگوریتم تپه نوردی بهینه شده (داخل کتاب نیست)

همانطور که در الگوریتم تپه نوردی بررسی کردیم ، این روش جستجوی محلی دارای مشکل قرار گرفتن در بهینگی محلی است. این مشکل تا حدی است که حتی در مورد مسائل ساده ای همچون مسئله ۸ وزیر نیز این روش جستجو از درصد موفقیت بسیار پایینی برخوردار بود. با این حال می توان تغییر کوچکی در این الگوریتم داد و تا حدودی میزان موقیت الگوریتم تپه نوردی را در حل مسائله بهینه سازی بالا برد. مشکل الگوریتم تپه نوردی این بود که هنگام قادر به تغییر حالت از یک محل بهینگی به محل بهینگی دیگر نبود. اما برای گریز از بهینگی های محلی می توانیم ترتیبی اتخاذ کنیم که هنگام قرار گرفتن در بهینگی های محلی به بخش دیگری از فضای حالت جهش کنیم و سپس در آنجا به جستجوی جواب با روش تپه نوردی بپردازیم و این عمل را تا رسیدن به یک ماکزیمم سراسری تکرار کنیم.

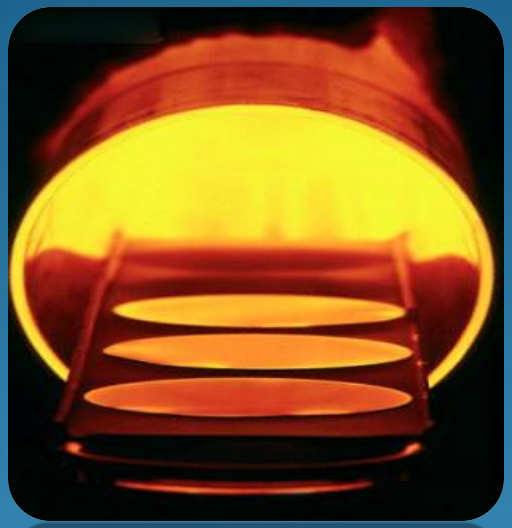
Ahmad Abdali Mohamadi

مثالی از الگوریتم تپه نوردی

مثال : الگوریتم تپه نوردی را روی گراف زیر پیاده سازی کنید ؟



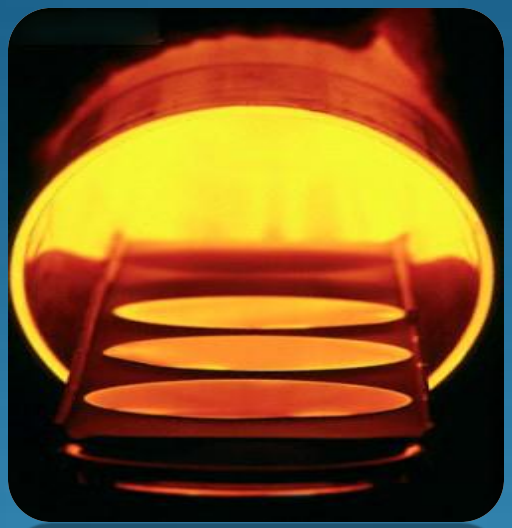
Simulated Annealing الگوریتم گرم و سرد



روش SA یکی از روش های متاهیوریستیکی احتمالی است که ایده آن از عمل سرد کردن تدریجی فلزات برای استحاکم بیشتر آن ها نشات گرفته است. همانند روش های تپه نوردی و تپه نوردی تعمیم یافته ، در این روش نیز مسئله از یک حالت مانند S در فضای حالت مسئله شروع کرده و با گذر از حالتی به حالت دیگر به جواب بهینه مسئله نزدیک می شود. انتخاب حالت شروع هم می تواند به صورت تصادفی انجام پذیرد و هم می تواند بر اساس یک قاعده حالت اولیه مسئله را انتخاب کنیم. در اینجا نیز تابع Objectiveمیزان بهینگی حالت فعلی را محاسبه کرده و Neighbor یک حالت همسایه برای حالت فعلی تولید می کند.

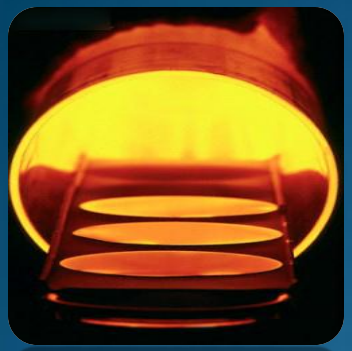
Ahmad Abdali Mohamadi

روش کار الگوریتم گرم و سرد



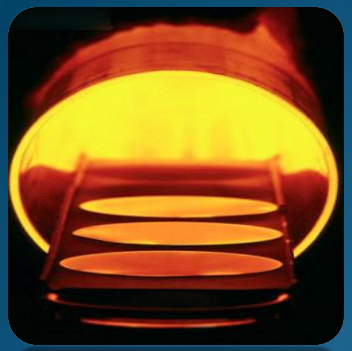
روش کلی کار به این صورت است که در هر تکرار ، الگوریتم SA حالت همسایه ای مانند 's ایجاد می کند و بر اساس یک احتمال ، مسئله از حالت s به حالت 's می رود و یا اینکه در همان حالت s باقی می ماند . این روند تا زمانی تکرار می شود که به یک جواب نسبتا بهینه برسیم یا اینکه ماکزیمم تعداد تکرار ها را انجام داده باشیم. همانطور که قبلا نیز بررسی کردیم نحوه تولید حالت همسایه از اهمیت به سزایی برخوردار است.

Ahmad Abdali Mohamadi



روش کار الگوریتم گرم و سرد

در الگوریتم SA انتخاب دما نقس بسیار زیادی در موفقیت الگوریتم دارد. انتخاب دمای اولیه بسیار بزرگ موجب کندی الگوریتم و انتخاب دمای کم موجب قرار گرفتن در نقاط بهینگی محلی می شود. استفاده از دمای بسیار زیاد در صورتی که مدت زمان کافی برای حل مساله داشته باشیم ، بهتر به نظر می رسد. با این حال هنگام پیاده سازی الگوریتم SAبرای یک مساله خاص بهتر آن است که دمای اولیه الگوریتم با توجه به بزرگی مساله تعیین گردد. با اینکه مشکل بهینگی های محلی در الگوریتم SAحل شده است ، با این حال در برخی مسائل هنگام استفاده از این الگوریتم به مشکلاتی بر خواهیم خورد. در الگوریتم SAممکن است هنگام تولید حالت همسایگی به حالتی گذر کنیم که قبلا یکبار این حالت را مشاهده کرده ایم. البته نمی توان برگشت به حالت تکراری را مشکلی برای الگوریتم تلقی کرد. چرا که در برخی موارد همین برگشت به حالت قبلی ممکن است موجب بهبود الگوریتم گردد.



الگوریتم گرم و سرد در یک جمله

هر وقت در ماکزیمم محلی گیر کردیم برخلاف تپه نوردی که حالت شروع
را تصادفی آغاز می کند اجازه می دهیم جستجو چند قدم به عقب برود تا از ماکزیمم محلی فرار کند .

این الگوریتم تضمین نمی کند حتماً بهترین جواب را می دهد بلکه ممکن است در ماکزیمم محلی گیر کند .

Ahmad Abdali Mohamadi

الگوریتم جستجوی پرتو محلی

در الگوریتم تپه نوردی ابتدا یک جواب تصادفی برای مسئله تولید می کردیم و سپس سعی بر آن داشتیم که با تولید حالت های همسایه حالت فعلی و انتخاب بهترین حالت همسایه ، به سمت جواب بهینه حرکت کنیم. همچنین با تغییر کوچکی که بر روی الگوریتم تپه نوردی انجام دادیم ، مشکل قرار گرفتن الگوریتم در بهینگی محلی را حل کردیم

(این روش تنها در مورد مسائل کوچک جوابگو خواهد بود).

Ahmad Abdali Mohamadi

الگوریتم جستجوی پرتو محلی

روش جستجوی دیگری بر پایه جستجوی تپه نوردی به نام جستجو پرتو محلی نیز وجود دارد که در حل مسائل از قدرت بسیار بیشتری نسبت به جستجوی تپه نوردی برخوردار می باشد. در این روش برخلاف روش تپه نوردی ابتدا kجواب برای مساله تولید می کنیم. سپس برای هریک از این K حالت همسایه های آن ها را تولید کرده و از میان همه همسایه ها K تا از بهترین همسایه ها را انتخاب می کنیم که این فرآیند تا رسیدن به شرط توقف ادامه می یابد.

هرچند که الگوریتم جستجوی پرتو محلی در مقایسه با الگوریتم تپه نوردی از کارآیی بیشتری برخوردار است ، با این حال هر دو این روش ها از قرار گرفتن در بهینگی های محلی مصون نیستند. از اینرو نیاز به روش های دیگری داریم که بتوانند از بهینگی های محلی گذر کرده و به سمت بهینگی سراسری حرکت کنند.

Ahmad Abdali Mohamadi

الگوریتم جستجوی پرتو محلی

به جای یک حالت ، k حالت را نگهداری میکند

حالت اولیه: k حالت تصادفی

گام بعد: جانشین همه k حالت تولید میشود

اگر یکي از جانشین ها هدف بود ، تمام میشود

وگر نه بهترین جانشین را انتخاب کرده ، تکرار میکند

تفاوت عمده با جستجوی شروع مجدد تصادفی

در جست و جوی شروع مجدد تصادفی ، هر فرایند مستقل از بقیه اجرا میشود

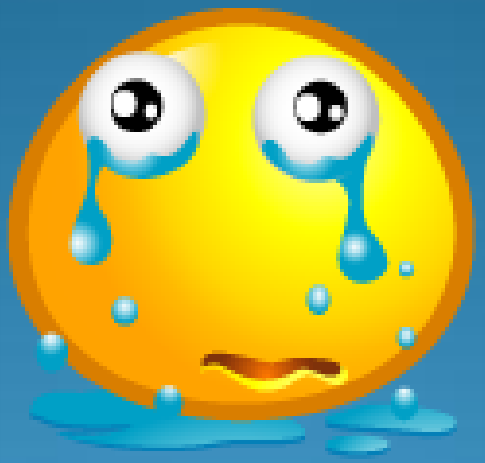
در جست و جوی پرتو محلی ، اطلاعات مفیدی بین k فرایند موازی مبادله میشود

جست و جوی پرتو غیرقطعی

به جای انتخاب بهترین k از جانشینها ، k جانشین تصادفی را بطوریکه احتمال انتخاب یکي تابعی صعودی از مقدارش باشد ، انتخاب میکند

Ahmad Abdali Mohamadi

جستجوی Online حذف شد



الگوریتم ژنتیک در یک نگاه

تا بحال با روش های جستجوی ناآگاهانه ، اگاهانه و متاهیوریستیک برای حل مسائل هوش مصنوعی آشنا شدیم. همچنین دیدیم که در مورد مسادل بهینه سازی اغلب روش های آگاهانه و ناآگاهانه جوابگوی نیاز ما نخواهند بود. چرا که بیشتر مسائل بهینه سازی در رده مسادل NPقرار دارند. بنابراین نیاز به روش جستجوی دیگری داریم که بتواند در فضای حالت مسائل NPبه طرف جواب بهینه سراسری حرکت کند. بدین منظور روش های جستجوی متاهیوریستیک را مطرح کردیم و دیدیم که این روش های جستجو می توانند به سمت بیهنگی های سراسری مسئله حرکت کنند. در کنار روش های جستجوی متاهیوریستیکی ، روش های جستجوی دیگری نیز وجود دارند که به روش های تکاملی معروفند. بر اساس نظریه داروین نسل هایی که از ویژگی های و خصوصیات برتری نسبت به نسل های دیگر برخوردارند شانس بیشتری نیز برای بقا و تکثیر خواهند داشت و ویژگی ها و خصوصیات برتر آنها به نسل های بعدی آنان نیز منتقل خواهد شد. همچنین بخش دوم نظریه داروین بیان می کند که هنگام تکثیر یک ارگان فرزند ، به تصادف رویدادهایی اتفاق می افتد که موجب تغییر خصوصیات ارگان فرزند می شود و در صورتی که این تغییر فایده ای برای ارگان فرزند داشته باشد موجب افزایش احتمال بقای آن ارگان فرزند خواهد شد.

Ahmad Abdali Mohamadi

الگوریتم ژنتیک

در محاسبات کامپیوتری ، بر اساس این نظریه داروین روش هایی برای مسائل بهینه سازی مطرح شدند که همه این روش ها از پردازش تکاملی در طبیعت نشات گرفته اند. ما نیز به این روش های جستجو ، الگوریتم های جستجوی تکاملی می گوییم. انواع مختلف الگوریتم های تکاملی که تا بحال مطرح شده اند ، که یکی از آنها الگوریتم ژنتیک است .

الگوریتم های تکاملی نیز همانند الگوریتم های متاهیوریستیکی ، به شکل تصادفی اما هدایت شده در فضای حالت مسئله حرکت می کنند.

Ahmad Abdali Mohamadi

الگوریتم ژنتیک

یکی از مهمترین کاربردهای الگوریتم ژنتیک در مسائل TSP می باشد. می توان گفت همیشه طبیعت افراد شایسته تر (Fitness) را برای زندگی برمی گزیند نه قویترین ها را.

مثلا" دايناسورها با وجود جثه عظيم و قوي‌تر بودن در طي روندي كاملا طبيعي بازي بقا و ادامه نسل را واگذار كردند در حالي كه موجوداتي بسيار ضعيف‌تر از آنها حيات خويش را ادامه دادند. ظاهرا طبيعت بهترين‌ها را تنها بر اساس هيكل انتخاب نمي‌كند! بدين ترتيب مي‌توان ديد كه طبيعت با بهره‌گيري از يك روش بسيار ساده توانسته است دائما هر نسل را از لحاظ خصوصيات مختلف ارتقا بخشد.

Ahmad Abdali Mohamadi

الگوریتم ژنتیک

این الگوریتم ها مبتنی بر روند تکاملی می باشند که برگرفته از نظام طبیعت است.

مزیت الگوریتم های ژنتیکی در این است که برای مسائلی که جواب قطعی وجود ندارد و یا نمی توان از راه حل های معمول به جواب رسید طی یک روند تکاملی مجموعه ای از نزدیک ترین جواب ها به بهترین جواب را نشان خواهد داد .

Ahmad Abdali Mohamadi

خصوصیات الگوریتم ژنتیک

- یک روش عددی و تصادفی و مبتنی بر تکرار است .
- در هر تکرار همزمان چند نقطه از فضای حالت بررسی می شود پس احتمال اینکه در ماکزیمم محلی گیر کند کم است .
- الگوریتم ژنتیک یک الگوریتم برای مسائل بهینه سازی است .
- الگوریتم ژنتیک به راحتی به مسائل پیوسته و گسسته بکار می رود.
- الگوریتم ژنتیک روندی موازی در حل مسله دارد .

Ahmad Abdali Mohamadi

نقاط قوت الگوريتم ژنتيک

اولين و مهمترين نقطه قوت اين الگوريتم ها اين است که الگوريتم هاي ژنتيک ذاتاً موازي اند اکثر الگوريتم هاي ديگر موازي نيستند و فقط مي توانند فضاي مسئله مورد نظر را در يک جهت در يک لحظه جستجو کنند.از آنجايي که GA چندين نقطه شروع دارد ،در يک لحظه مي تواند فضاي مسئله را از چندجهت مختلف جستجو کند. اگر يکي به نتيجه نرسيد ساير راه ها ادامه مي يابند و منابع بيشتري در اختيار شان قرار مي گيرد.

فاز های اصلی الگوریتم ژنتیک

۱) کدینگ مسله

۲) تعیین جمعیت اولیه

۳) انتخاب کردن

۴) ترکیب یا آمیزش

۵) جهش

۶) پذیرش جمعیت جدید و تست

۷) جایگزینی

تولید جمعیت اولیه

الگوریتم ژنتیک کارخود را با تولید جمعیت اولیه ای از کروموزوم ها آغاز می کند و سپس در یک حلقه به طور مکرر تعدادی از کروموزوم های برتر نسل فعلی را انتخاب کرده و سپس نسل جدیدی را از این کروموزوم ها تولید می کند. همانطور که دیدیم هر کروموزوم نشان دهنده یک حالت از فضای حالت مسئله می باشد و بنابراین منظور از تولید جمعیت اولیه ، تولید تعدادی جواب برای مسئله خواهد بود.

تولید جواب های اولیه نیز به دو صورت تصادفی و هیوریستیکی می تواند انجام پذیرد. به عنوان مثال در مسئله ۸ وزیر می توانیم به شکل تصادفی وزیرها را در خانه های صفحه شطرنج قرار دهیم و یا در مسئله فروشنده دوره گرد می توانیم ترتیب ملاقات شهرها هم به شکل تصادفی و هم به شکل هیوریستیکی انتخاب کنیم.

تعداد جمعیت اولیه چقدر باشد

ممکن است این سوال مطرح شود که تعداد کروموزوم هایی که برای جمعیت اولیه تولید می کنیم چگونه تعیین می گردد ؟ جواب قطعی برای این سوال وجود ندارد و در برخی موارد تعداد افراد (کروموزوم) جمعیت به شکل تجربی انتخاب می شود. اما یک اصل کلی وجود دارد که در تعیین اندازه جمعیت اولیه باید مدنظر قرار دهیم. افزایش اندازه جمعیت با اینکه موجب افزایش قدرت محاسباتی الگوریتم می گردد ، از طرف دیگر نیز به طور چشمگیری مدت زمان اجرای الگوریتم را افزایش می دهد. همچنین در برخی موارد بزرگی جمعیت موجب همگرایی سریع الگوریتم خواهد شد که در نتیجه آن جواب هایی دور از نقاط بهینگی سراسری را تولید خواهد کرد. در مقابل جمعیت با اندازه کوچک نیز توانایی چندانی به خصوص در مسائل پیچیده برای حل مسئله نخواهد داشت.

تعداد جمعیت اولیه چقدر باشد

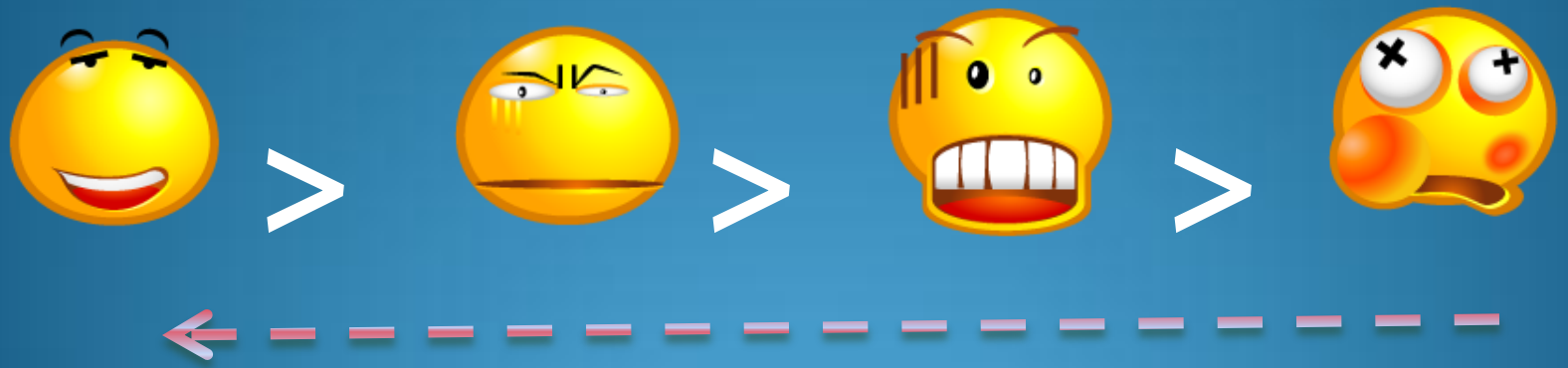
ممکن است این سوال مطرح شود که تعداد کروموزوم هایی که برای جمعیت اولیه تولید می کنیم چگونه تعیین می گردد ؟ جواب قطعی برای این سوال وجود ندارد و در برخی موارد تعداد افراد (کروموزوم) جمعیت به شکل تجربی انتخاب می شود. اما یک اصل کلی وجود دارد که در تعیین اندازه جمعیت اولیه باید مدنظر قرار دهیم. افزایش اندازه جمعیت با اینکه موجب افزایش قدرت محاسباتی الگوریتم می گردد ، از طرف دیگر نیز به طور چشمگیری مدت زمان اجرای الگوریتم را افزایش می دهد. همچنین در برخی موارد بزرگی جمعیت موجب همگرایی سریع الگوریتم خواهد شد که در نتیجه آن جواب هایی دور از نقاط بهینگی سراسری را تولید خواهد کرد. در مقابل جمعیت با اندازه کوچک نیز توانایی چندانی به خصوص در مسائل پیچیده برای حل مسئله نخواهد داشت.

ارزش گذاری

در واقع میزان شایستگی افراد برای ادامه نسل است .

آن را با Fitness نشان می دهند . هرچه کیفیت و شایستگی یک فرد بیشتر

باشد احتمال اینکه در تولید نسل بعد مشارکت کند بیشتر است .



شایستگی بیشتری دارد

کد گذاری هر کروموزوم

الگوريتم ژنتيك بجاي اينكه بر روي پارامترها يا متغيرهاي مساله كاركند ، با شكل كد شده آنها بطور مناسب سروكار دارد. کدینگ استاندارد در ژنتیک کدینگ باینری

است . کدینگ درست مسله در الگوریتم ژنتیک سخت ترین قسمت کار است .

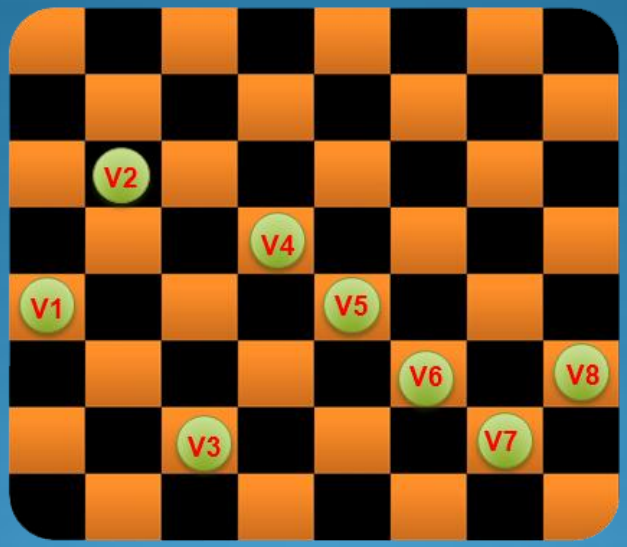
1	0	1	1	0	0	0	1

1	1	0	1	0	1	1	1

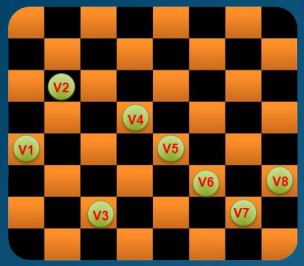
یک مثال از کد گذاری هشت وزیر

در مسئله ۸ وزیر یک روش کدگذاری می تواند استفاده از یک آرایه ۸ عنصری باشد که هر عنصر از آرایه نشان دهنده سطری است که وزیر در آن قرار دارد. به عنوان مثال آرایه زیر را در نظر بگیرید :

5	3	7	4	5	6	7	6



با توجه به مقادیر آرایه در میابیم که وزیر اول در ستون اول و سطر پنجم ، وزیر دوم در ستون دوم و سطر سوم ، وزیر سوم در ستون سوم و سطر هفتم و ... قرار دارد .



نکاتی در مورد کد گذاری هشت وزیر

در الگوریتم های ژنتیک به هرجواب مسئله یک کروموزوم و به هر متغیر از آن یک ژن می گوییم.

در اینجا جواب مسئله را با استفاده از یک آرایه ۸ عنصری نشان دادیم که به آرایه یک کرموزوم و به هر عنصر از آریه یک ژن می گوییم.

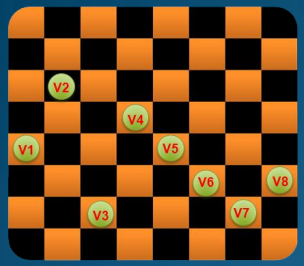
با توجه به مسئله بهینه سازی برای هر ژن می توان سوالات زیر را مطرح کرد :

- آیا مقادیر قابل قبول برای ژن در یک باژه مشخص قرار دارد ؟

  (در مثال ۸ وزیر هر ژن می تواند عددی بین ۱ تا ۸ به خود بگیرد)
- آیا دامنه مقادیر ژن گسسته است یا پیوسته ؟

  (در مثال هشت وزیر این مقادیر گسسته هستند)
- آیا در صورت مسئله محدودیت هایی برای انتخاب مقادیر ژن وجود دارد ؟

  (در مسئله ۸ وزیر محدودیتی نداریم)



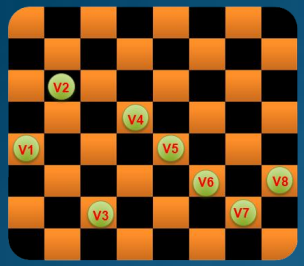
نکات تکمیلی در مورد کد گذاری هشت وزیر

نحوه نمایش جواب مسئله و ژن ها در موفقیت الگوریتم و نحوه پیاده سازی الگوریتم ژنتیک تاثیر بسیار مهمی دارد. در بیشتر مسائل نیز روش های مختلفی برای نشان دادن جواب مساله می توان طراحی کرد. به عنوان مثال در مسئله ۸ وزیر به جای استفاده از بردار یک بعدی

۸ عنصری می توان از یک ماتریس ۸ * ۸ استفاده کرد که در آن هر وزیر می تواند در هریک از ۶۴ خانه صفحه شطرنج قرار گیرد.

اما برای هرچه ساده تر شدن پیاده سازی دیگر مراحل الگوریتم ژنتیک برای این مسئله و افزایش کارآیی زمانی الگوریتم ، می توانیم تریبی اتخاذ کنیم که هیچ دو وزیری در یک سطر

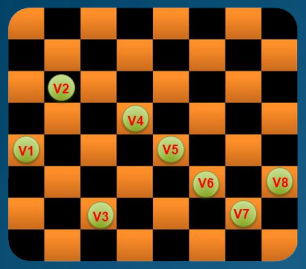
(یا ستون) قرار نگیرد. بنابراین می توانیم از نمایش برداری برای آن استفاده کنیم که در این نحوه نمایش مقدار موجود در هر ژن نشان دهنده شماره سطری (یا ستون) است که وزیر در آن قرار دارد.



نکات تکمیلی در مورد کد گذاری هشت وزیر

در حالت کلی هنگام طراحی یک روش نمایش کروموزوم باید موارد زیر را در نظر بگیریم :

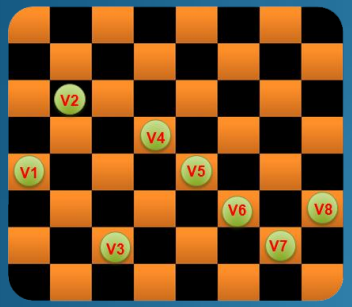
- کروموزوم باید به گونه ای طراحی شود که تا حد ممکن از اطلاعات مسئله برای کاهش تعداد حالات فضای حالت استفاده کند. به عنوان مثال در مسئله ۸ وزیر با تبدیل نمایش ماتریسی به نمایش برداری فضای حالت مسئله به طور چشمگیری کوچک گردید

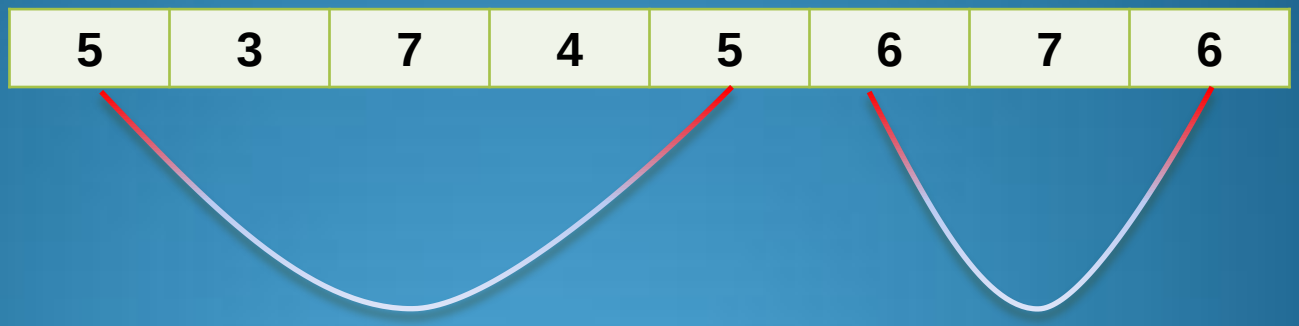


نکات تکمیلی در مورد کد گذاری هشت وزیر

در حالت کلی هنگام طراحی یک روش نمایش کروموزوم باید موارد زیر را در نظر بگیریم :

- در یک کروموزوم باید بتوان براحتی محدودیت های مسئله را اعمال کرد و در اثر اعمال مراحل دیگر الگوریتم ژنتیک محدودیت های مسئله نقض نگردد

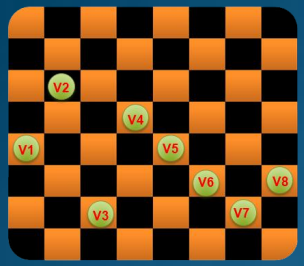




به راحتی می توان دید وزیر ۱ و وزیر ۵ با هم برخورد می کنند یعنی کدینگ ما شفاف است

به راحتی می توان دید وزیر ۶ و وزیر ۸ با هم برخورد می کنند یعنی کدینگ ما شفاف است

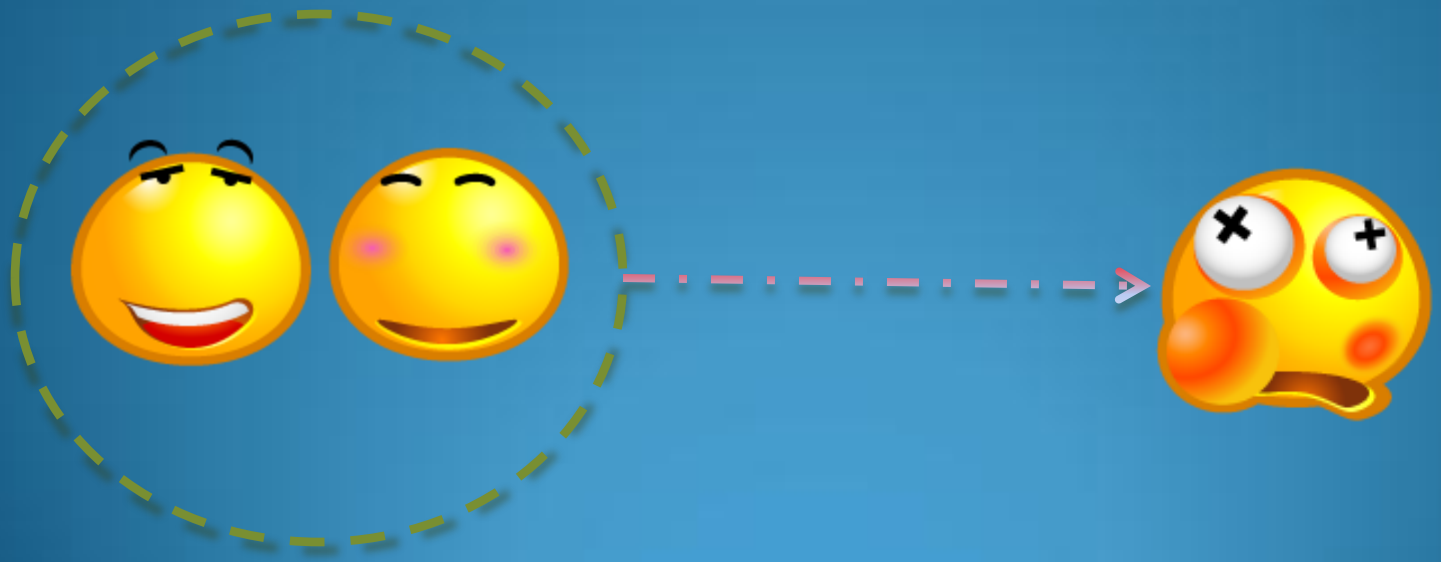
Ahmad Abdali Mohamadi



نکات تکمیلی در مورد کد گذاری هشت وزیر

در حالت کلی هنگام طراحی یک روش نمایش کروموزوم باید موارد زیر را در نظر بگیریم :

- تا حد امکان کروزوم نباید در اثر اجرای مراحله ادغام (تولید نسل جدید) به حالت نامعتبر تبدیل گردد. در برخی مسئله گزیر از حالت های نامعتبر در نحوه نمایش امکان پذیر نخواهد بود و باید در روش ادغام کروموزوم های تغییر ایجاد کنیم.



انواع کد گذاری در ژنتیک

کدینگ باینری :

در این نحوه نمایش هر ژن از کروموزوم مقدار ۱ یا ۰ به خود می گیرد و موجب تشیکل یک رشته باینری در کرموزوم می گردد. به عنوان مثال فرض کنید یک تابع ۳ پارامتری داریم و می خواهیم مقدار این ۳ پارامتر را طوری انتخاب کنیم که مقدار تابع مینیمم گردد.

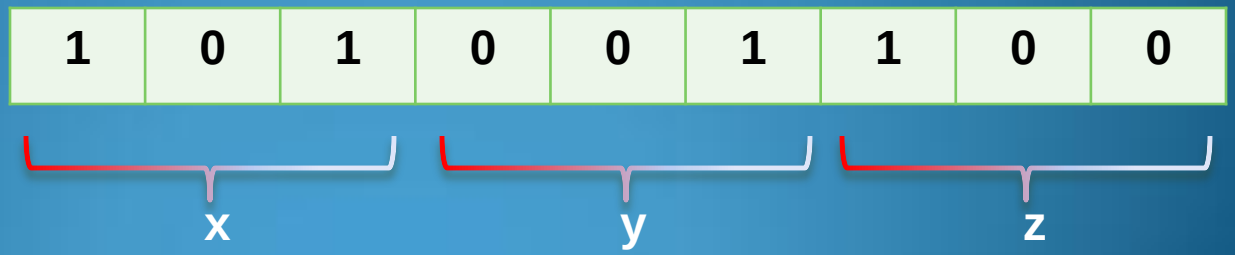
تابع سه متغیره ما :

$$F(x, y, z) = xyz - x^2 + zy$$

فرض کنیم که اعداد ما در بازه ۰ تا ۸ هستند پس هر عدد ۳ بیت نیاز دارد در این

مثال ما کروموزم را ترکیب در نظر میگیریم xyz

A کروموزم



مثال از کد گذاری باینری

$F(x, y, z) = xyz - x^2 + zy$

هدف مینیم تابع سه متغیره است .

A کروموزم

1	0	1	0	0	1	1	0	0

x y z

B کروموزم

1	0	0	1	0	1	0	0	0

x y z

$F(A \text{ کروموزم}) = F(5, 1, 4) = xyz - x^2 + zy = 5\times1\times4 - 5^2 + 1\times4 = -1$

$F(B \text{ کروموزم}) = F(4, 5, 0) = xyz - x^2 + zy = 4\times5\times0 - 4^2 + 0\times5 = -16$

پس کروموزم B شایستگی بیشتری برای بقا دارد

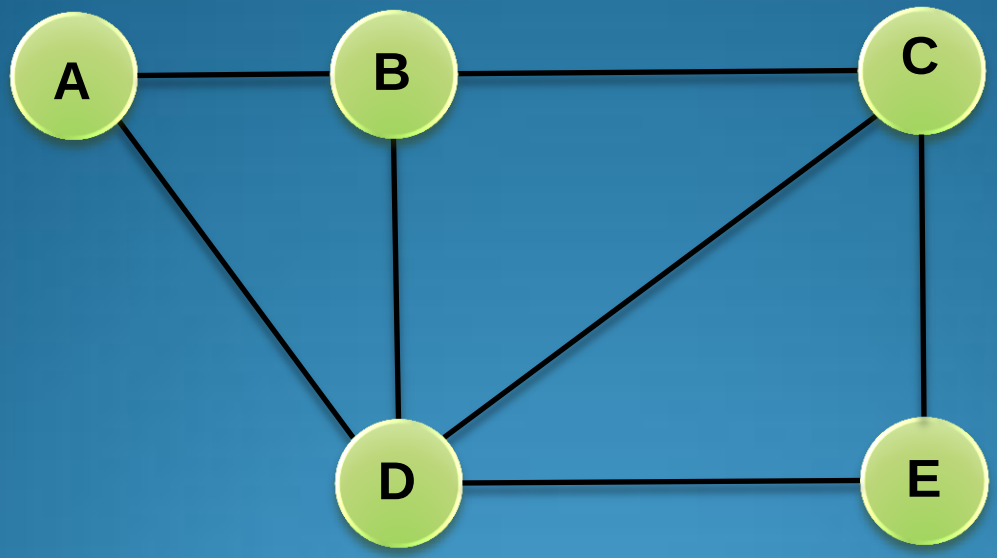
Ahmad Abdali Mohamadi

انواع کد گذاری در ژنتیک

کدینگ جایگشتی :

در بعضی مسائل مانند فروشنده دورگرد با تعدادی از شهر ها روبرو هستیم . که ترکیبی از آنها جواب مسله است . در اینگونه مسائل جواب تکراری نبا ید تولید شود

مثال :



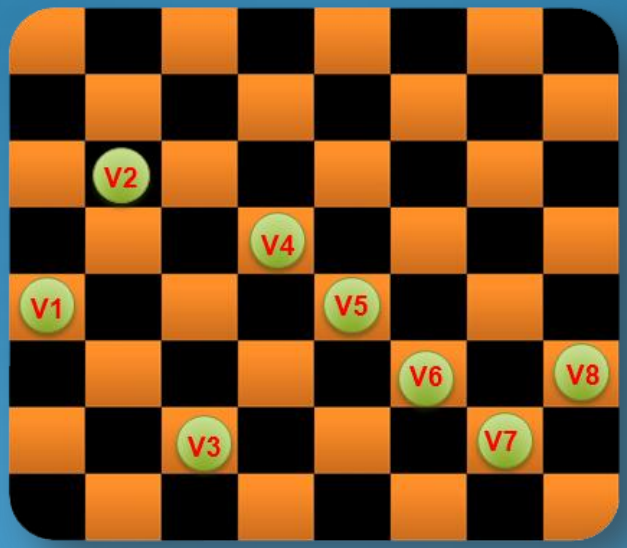
قابل قبول نیست

انواع کد گذاری در ژنتیک

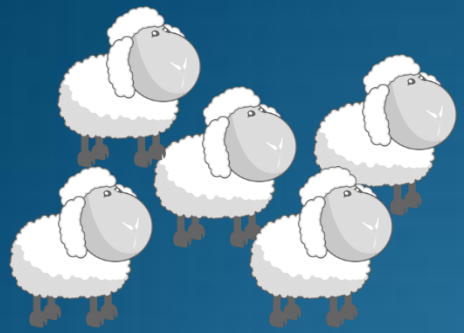
کدینگ مقداری :

در این روش نمایش ، مقادیر واقعی متغیر ها را نشان می دهیم. به عنوان مثال در مسئله ۸ وزیر هر ژن شماره سطری را نشان می دهد که وزیر در آن قرار دارد. با وجود اینکه روش های نمایش دیگری همچون نمایش اکتال ، نمایش هگزادسیمال نیز وجود دارد ، اما با این حال نحوه نمایش حالتی از مسئله وابسته به مسئله بوده و به تجربه و مهارت طراح الگوریتم ژنتیک بستگی دارد.

5	3	7	4	5	6	7	6



ایجاد جمعیت اولیه در ژنتیک



الگوریتم ژنتیک کارخود را با تولید جمعیت اولیه ای از کروموزوم ها آغاز می کند و سپس در یک حلقه به طور مکرر تعدادی از کروموزوم های برتر نسل فعلی را انتخاب کرده و سپس نسل جدیدی را از این کروموزوم ها تولید می کند. همانطور که دیدیم هر کروموزوم نشان دهنده یک حالت از فضای حالت مسئله می باشد و بنابراین منظور از تولید جمعیت اولیه ، تولید تعدادی جواب برای مسئله خواهد بود.

تولید جواب های اولیه نیز به دو صورت تصادفی و هیوریستیکی می تواند انجام پذیرد. به عنوان مثال در مسئله ۸ وزیر می توانیم به شکل تصادفی وزیرها را در خانه های صفحه شطرنج قرار دهیم و یا در مسئله فروشنده دوره گرد می توانیم ترتیب ملاقات شهرها هم به شکل تصادفی و هم به شکل هیوریستیکی انتخاب کنیم.

نکته مهم هنگام تولید جمعیت اولیه آن است که هنگام تولید جواب برای مسئله (تصادفی یا هیوریستیکی) جواب تولید شده باید به شکل کروموزومی که در مرحله قبل تعیین کردیم کدگذاری شده و به مجموعه جمعیت اضافه گردد. بنابراین از تولید جواب هایی که موجب نقض نحوه کدگذاری تعیین شده برای کروموزوم گردد باید اجتناب کنیم.

اندازه جمعیت اولیه

ممکن است این سوال مطرح شود که تعداد کروموزوم هایی که برای جمعیت اولیه تولید

می کنیم چگونه تعیین می گردد؟ جواب قطعی برای این سوال وجود ندارد و در برخی موارد تعداد افراد (کروموزوم) جمعیت به شکل تجربی انتخاب می شود. اما یک اصل کلی وجود دارد که در تعیین اندازه جمعیت اولیه باید مدنظر قرار دهیم. افزایش اندازه جمعیت با اینکه موجب افزایش قدرت محاسباتی الگوریتم می گردد ، از طرف دیگر نیز به طور چشمگیری مدت زمان اجرای الگوریتم را افزایش می دهد. همچنین در برخی موارد بزرگی جمعیت موجب همگرایی سریع الگوریتم خواهد شد که در نتیجه آن جواب هایی دور از نقاط بهینگی سراسری را تولید خواهد کرد. در مقابل جمعیت با اندازه کوچک نیز توانایی چندانی به خصوص در مسائل پیچیده برای حل مسئله نخواهد داشت.

انتخاب کروموزوم

با توجه به نظریه داروین و برای تولید نسل جدید از جمعیت فعلی ، باید کروموزوم هایی از این جمعیت را برای ادغام و تکثیر انتخاب کنیم که هنگام اجرای عمل انتخاب ، کروموزوم هایی که از بهینگی بیشتری برخوردارند (بهینگی هر کروموزوم با استفاده از تابع بهینگی محاسبه می شود) شانس بیشتری برای انتخاب خواهند داشت. کروموزوم های انتخاب شده برای ادغام با دیگر کروموزوم ها به منظور تولید نسل جدید در محل دیگری جمع آوری می شوند. عمل ادغام و تولید کروموزوم های جدید در این محل که آن را حوضچه ژنتیک می نامیم انجام می پذیرد. انتخاب کروموزوم ها از جمعیت فعلی نیز به روش های گوناگونی انجام می پذیرد

Ahmad Abdali Mohamadi

حوضچه ژنتیک

مسئله دیگر در انتخاب کروموزوم ها اندازه حوضچه ژنتیک است. بدین معنی که چه تعداد کروموزوم برای ادغام و تولید نسل جدید انتخاب کنیم. روش معمول آن است که اندازه حوضچه ژنتیک دقیقا به اندازه جمعیت فعلی باشد. ممکن است چنین تصویر شود که انتخاب حوضچه ژنتیک به اندازه جمعیت فعلی بدین معنا است که تمام کروموزوم های جمعیت فعلی را به حوضچه ژنتیک منتقل کنیم. در حالی که چنین نیست و هنگام انتخاب کروموزوم ها ، آنهایی که از بهینگی کمتری برخودارند به این حوضه منتقل نمی شوند و در مقابل کروموزوم های بهینه تر ممکن است چند بار در آن کپی شوند. بر اساس نظریه داروین هنگام تکثیر کروموزوم ها ، خصوصیات برتر هر کروموزوم پس از ادغام با دیگر کروموزوم ها به نسل بعدی منتقل شده و موجب تولید کروموزوم های بهتری می شود. با این تعریف در صورتی که چند کپی از یک کروموزوم بهینه در حوضچه ژنتیک وجود داشته باشد ، موجب می شود که در تولید نسل بعدی از کروموزوم های بهتر نسل فعلی بیشتر استفاده شده و بهینگی نسل بعدی بهتر از نسل فعلی باشد. با وجود اینکه این نظریه در طبیعت صادق است ، اما با این حال در برخی موارد این نظریه هنگام پیاده سازی الگوریتم ژنتیک با شکست مواجه می شود ، اما در بیشتر موارد نتایج بسیار جالبی را با خود به همراه دارد. نتیجه آنکه که کروموزوم های با بهینگی کم نیز باید در حوضچه آمیزش حضور داشته باشند.

روش های انتخاب از جمعیت فعلی

نحوه انتخاب کروموزوم ها از جمعیت فعلی و قرار دادن آن ها در حوضچه ژنتیک می باشد را انتخاب از جمعیت فعلی گویند .

انتخاب کروموزوم ها در حالت کلی به سه روش انجام می پذیرد :

- انتخاب تصادفی
- انتخاب رقابتی
- انتخاب مبتنی بر چرخ رولت

روش های انتخاب از جمعیت فعلی

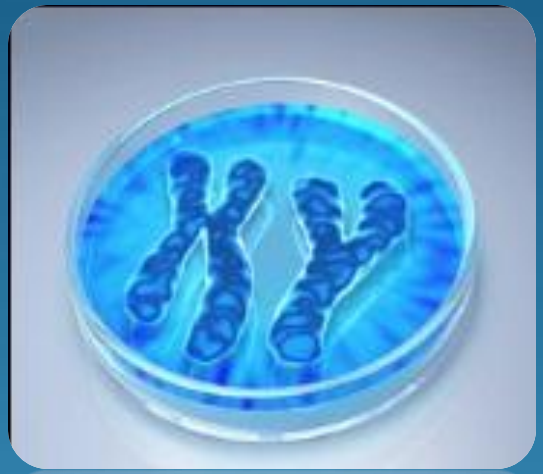
انتخاب تصادفی

ساده ترین روش انتخاب کروموزوم ها این است که بدون در نظر گرفتن میزان بهینگی کروموزوم ها آن ها را به تصادف انتخاب کرده و به حوضچه ژنتیک منتقل کنیم. پیاده سازی این روش بسیار ساده بوده ولی در مقابل از کارآیی بسیار کمی برخوردار است و همانطور که می بینیم با اصول کلی انتخاب کروموزوم ها که در ابتدای این بخش مطرح کردیم ، منافات دارد. اما با این حال می توان در برخی از مسائل بخشی از عمل انتخاب را به شکل تصادفی انجام دهیم. چرا که در این روش کروموزوم های خوب و بد هر دو به یک اندازه شانس انتخاب شدن دارند. باید توجه کنیم که شرکت دادن یک کروموزوم بد نیز در برخی موارد می تواند موجب تولید کرومزوم بهتری شود.

انتخاب رقابتی

در روش انتخاب رقابتی هربار که می خواهیم کروموزومی را از جمعیت فعلی انتخاب کنیم ، چند کروموزوم از جمعیت را به تصادف انتخاب کرده و از میان آن ها کروزوم بهتر را به حوضچه ژنتیک منتقل می کنیم و این عمل تا پرشدن کامل حوضچه ژنتیک ادامه می یابد.

ترکیب در الگوریتم ژنتیک



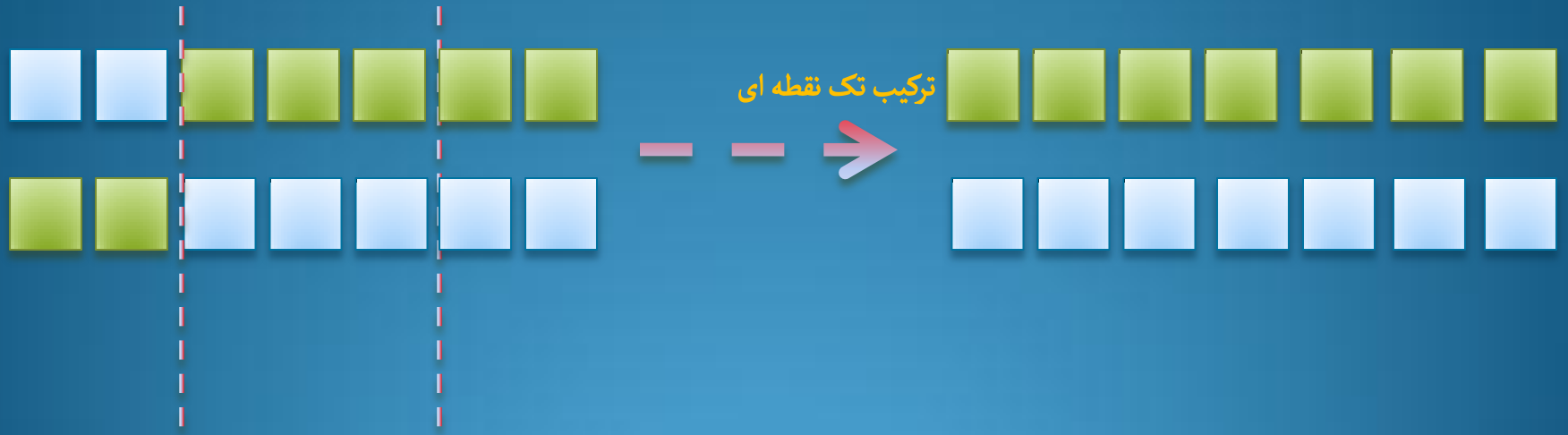
بخشی از فرآیند تکامل در طبیعت بدین صورت است که کروموزوم هایی از والدین انتخاب شده و باهم ترکیب می شوند. ما نیز در مرحله انتخاب ، کروموزوم هایی را برای ترکیب به حوضچه ژنتیک منتقل کردیم. حال وقت آن رسیده است که کروموزوم هایی را از این حوضچه انتخاب کرده و باهم ادغام کنیم. به همین منظور در این بخش روش هایی را برای ادغام کروموزوم ها بررسی خواهیم کرد. توجه کنید که در برخی از مسائل و با توجه به روش کدگذاری کروموزوم ها ، عمل ادغام کروموزوم ها موجب نقض برخی از شرایط مسئله می گردد. بنابراین در چنین مواردی باید با استفاده از روش هایی که طراحی می کنیم به اصلاح کروموزوم ها بپردازیم.

Ahmad Abdali Mohamadi

چند روش ترکیب در الگوریتم ژنتیک

ترکیب تک نقطه ای :

یک نقطه از کروموزوم را به تصادف انتخاب کرده و سپس به روش زیر دو کروموزوم فرزند تولید می کند.



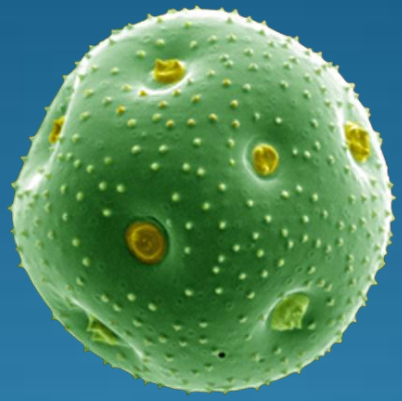
چند روش ترکیب در الگوریتم ژنتیک

ترکیب دو نقطه ای :

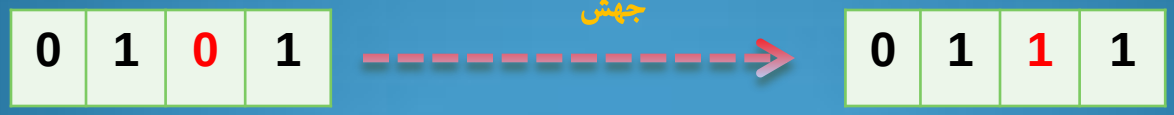
این روش دو نقطه متفاوت از کروموزوم را به تصادف انتخاب کرده و سپس به روش زیر دو کروموزوم فرزند تولید می کند.

ترکیب در ژنتیک باعث از بین رفتن پراگندگی یا تنوع ژنتیکی می شود زیرا باعث می شود ژن های خوب یکدیگر را پیدا کنند .

جهش در الگوریتم ژنتیک



نظریه داروین بیان می کند که پس از تولید کروموزوم های فرزند ممکن است در برخی از این کروموزوم ها جهش هایی به تصادف روی دهند که موجب بهینگی هرچه بیشتر کروموزوم و یا بدتر شدن آن گردند. ما نیز در استفاده از الگوریتم های ژنتیک از این خصوصیت استفاده کرده و جهش هایی را در جواب های تولید شده برای مسئله ایجاد می کنیم. این روش نیز بسته به ابتکار طراح الگوریتم به روش های گوناگونی پیاده سازی می شود. اما روش کلی کار بدین شکل است که مقدار یک یا چند عدد از ژن های یک کرومزوم را تغییر دهیم.



کروموزوم قبل جهش

کروموزوم بعد جهش

Ahmad Abdali Mohamadi

نکات مهم در الگوریتم ژنتیک

سیاست جایگزینی :

جایگزینی جمعیت فعلی با نسل جدید در حالت کلی به دو روش انجام می پذیرد :

- همه کروموزوم های جمعیت فعلی را با کروموزوم های در نسل جدید جایگزین کنیم.
(حالتی که اندازه نسل جدید برابر با جمعیت فعلی باشد ، کل جمعیت فعلی با نسل جدید جایگزین می شود)

- برخی از کروموزوم های جمعیت فعلی را انتخاب کرده و آن ها را با کروزوم دیگری در نسل جدید جایگزین کنیم.

الگوریتم ژنتیکی که از سیاست جایگزینی اول استفاده کند ، GGA گویند .

الگوریتم ژنتیکی که از روش دوم استفاده کند SSGA گوییم .

مثال یک از الگوریتم ژنتیک

مثال : فرض کنید تابع $f(x) = -x^2+6x -3$ را داریم و $0\leq x \leq 15$ و $x \in N$ باشد می خواهیم ماکزیمم تابع را در فاصله داده شده حساب کنیم .

مراحل حل مسله به کمک ژنتیک

کدینگ مسله :

چون اعداد بین ۰ تا ۱۵ هستند برای نمایش آنها ۴ بیت کافی است . در این حالت طول هر کروموزوم برابر ۴ می شود .

0	1	0	1

کروموزومی که معرف عدد ۵ است

توجه داریم که کدینگ مسله از نوع کدینگ باینری است

Ahmad Abdali Mohamadi

مثال یک از الگوریتم ژنتیک

تابع ارزش مسله (Evaluation) :

در این مرحله باید مقدار شایستگی Fitness را برای هر کروموزوم تعیین کنیم در واقع وظیفه تابع ارزش این است که برای هر کروموزوم Fitness آن را حساب کند .

در این مثال اگر دو عدد داشته باشیم (دو کروموزوم) به نام های x ، y و شرط زیر برقرار باشد :

$$f(x) > f(y)$$

می گوییم که عدد x یا کروموزوم x شایستگی بیشتری نسبت به عدد y یا کروموزوم y دارد

کروموزومی که معرف عدد ۵ است	0	1	0	1	$f(5) = -5^2+6\times5 -3 = 2$
کروموزومی که معرف عدد ۷ است	0	1	1	1	$f(7) = -7^2+6\times7 -3 = -10$
کروموزومی که معرف عدد ۲ است	0	0	1	0	$f(2) = -2^2+6\times2 -3 = 5$

کروموزوم ۵ و ۲ شایسته تر از کروموزوم ۷ هستند .

Ahmad Abdali Mohamadi

مثال یک از الگوریتم ژنتیک

تعیین جمعیت اولیه از کروموزوم ها :

معمولاً جمعیت اولیه را به شکل تصادفی انتخاب می کنند در این مورد ما فرض می کنیم
که دو کروموزوم ۵ و ۳ را به عنوان جمعیت اولیه در نظر گرفته ایم .

کروموزومی که معرف عدد ۵ است	0	1	0	1
کروموزومی که معرف عدد ۷ است	0	1	1	1
کروموزومی که معرف عدد ۲ است	0	0	1	0

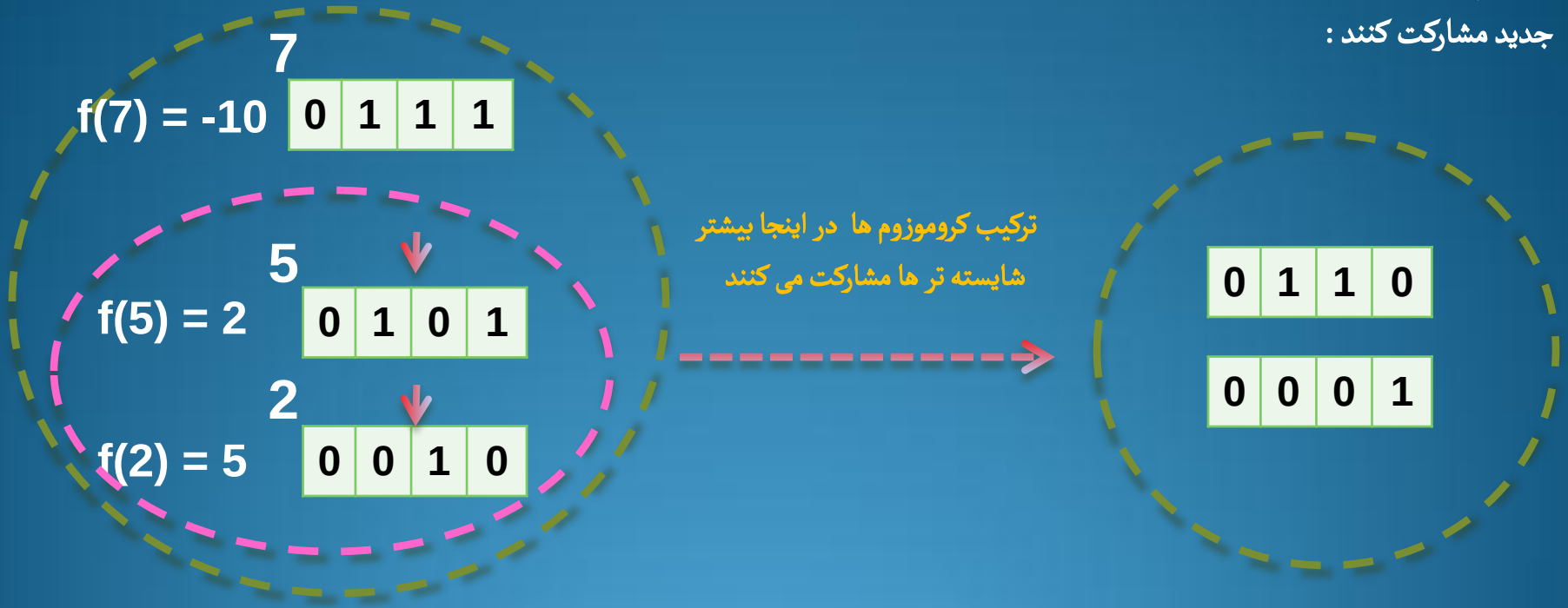
جمعیت اولیه ما شامل سه کروموزوم است

Ahmad Abdali Mohamadi

مثال یک از الگوریتم ژنتیک

ترکیب کروموزوم ها (Crossover):

در این مرحله با اعمال ترکیب جمعیت اولیه را تغییر می دهیم و جمعیت جدید را تولید میکنیم البته توجه کنید آنهای که شایسته تر هستند بیشتر شانس دارند در تولید نسل جدید مشارکت کنند :

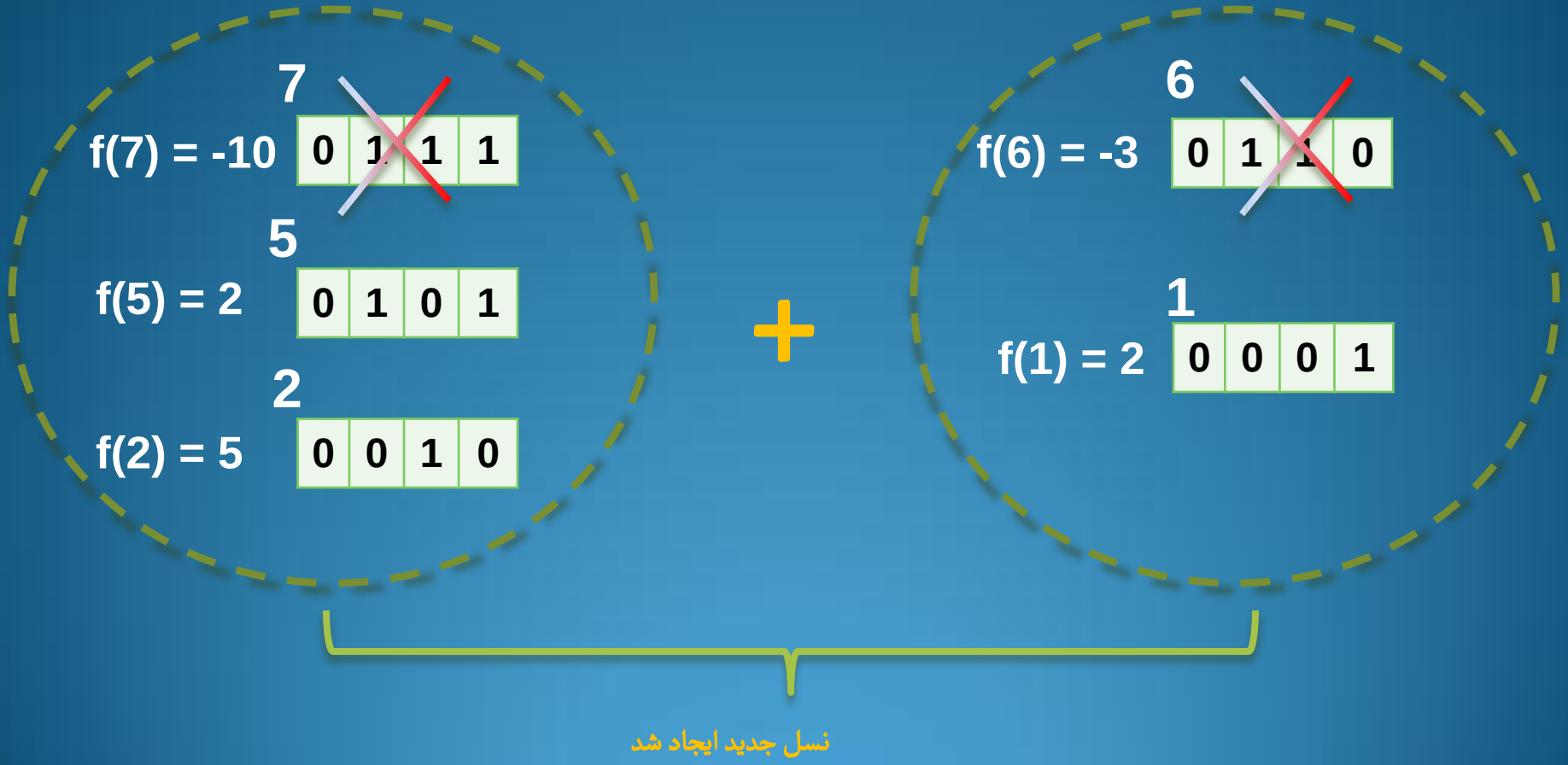


تولد دو کروموزوم جدید که در نسل جدید قرار می گیرند

جمعیت اولیه با دو کروموزوم دچار ترکیب در نقطه ۲ و ۳ می شود تا نسل جدید ایجاد شود

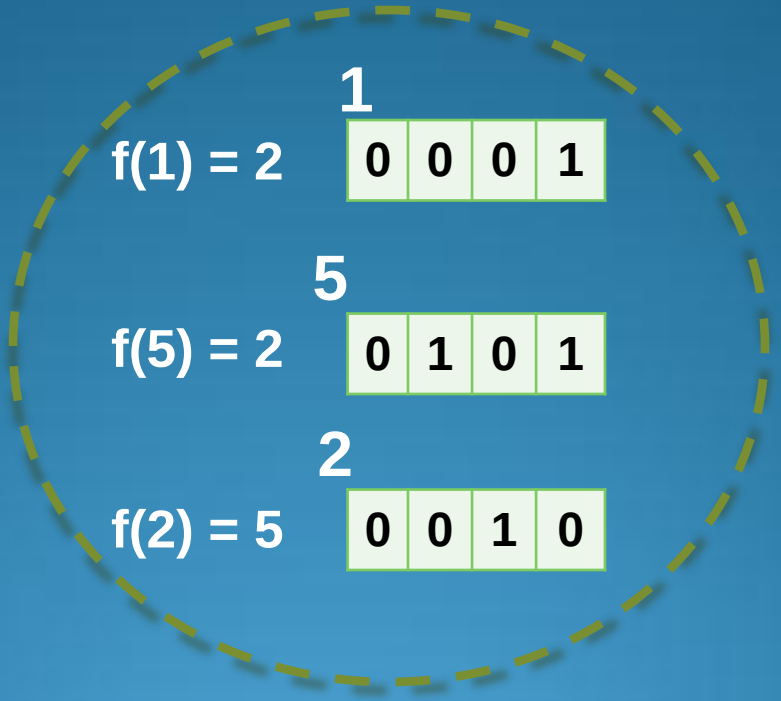
مثال یک از الگوریتم ژنتیک

در این مرحله جمعیت جدید ی به وجود می آید یعنی جمعیت قبلی و جمعیت جدید را با هم جمعیت جدید می گوییم البته بهتر است مواردی که شایسته نیستند ازجمعیت حذف شوند .



مثال یک از الگوریتم ژنتیک

بعد از اصلاح جمعیت مورد نظر نسل جدید ایجاد می شود :

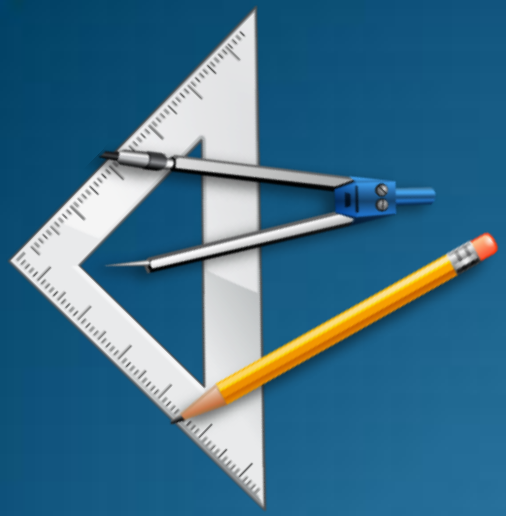


نسل جدیدبعد از پالایش ایجاد شد

مثال یک از الگوریتم ژنتیک

البته کروموزوم ها جمعیت مورد نظر می توانند دچار جهش شوند و اصلاح می کنیم :

البته بهترین جواب در هر مرحله امکان دارد در مرحله بعد از بین رود پس بهتر است در هر مرحله بهترین جواب را در جایی ذخیره کنیم . در مثال بالا در این مرحله $x=3$ را به عنوان بهترین جواب ذخیره می کنیم .



مثال دو از الگوریتم ژنتیک

مسئله معروف مربع جادویی ۳ در ۳ را با ژنتیک حل و مدل سازی کنید ؟

Question ?

Thank You

Ahmad Abdali Mohamadi