الگوریتم‌های تکاملی

چارچوب کلی

فهرست مطالب

- خلاصه مباحث قبل
- حل یک مسأله ساده با استفاده از الگوریتم های تکاملی
- چارچوب کلی یک الگوریتم تکاملی
- ویژگی های کلی الگوریتم های تکاملی
- بازنمایی
- جمعیت
- شروط خاتمه

خلاصه مباحث قبل

- **تکامل در طبیعت باعث ایجاد تولید جواب (موجود) مناسب برای هر محیط شده است.**

- **الگوریتم های تکاملی سعی در استفاده از این الگو برای حل مسائل دارند.**

- **الگوریتم ها ی تکاملی همانند همتای زیستی خود**
  - از دسته الگوریتم های "سعی و خطا" هستند.
  - مبتنی بر جمعیت و تصادفی هستند.
  - از عملگرهای جهش و بازترکیبی برای ایجاد تنوع در جمعیت استفاده می کنند.
  - با رقابت بر سر منابع محدود تنها تعدادی از جواب ها امکان بقا دارند.

فهرست مطالب

- خلاصه مباحث قبل
- **حل یک مسأله ساده با استفاده از الگوریتم های تکاملی**
- چارچوب کلی یک الگوریتم تکاملی
- ویژگی های کلی الگوریتم های تکاملی
- بازنمایی
- جمعیت
- شروط خاتمه

حل مسأله بهینه سازی با استفاده از الگوریتم های تکاملی- تعریف مسأله

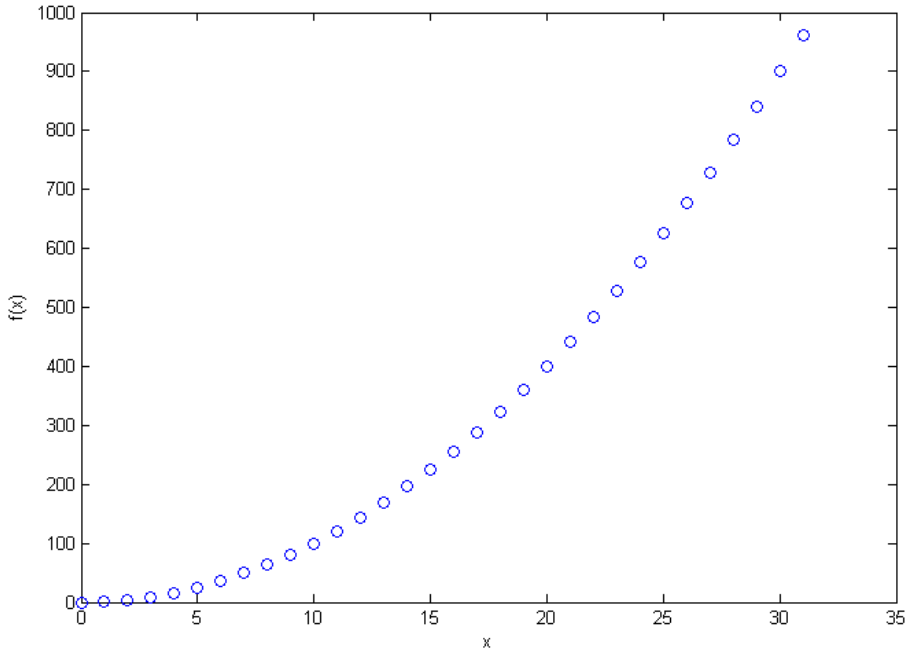
- ماکزیمم تابع

$$f(x) = x^2$$

را در میان مجموعه اعداد

$$\{0,1,2,\ldots,31\}$$

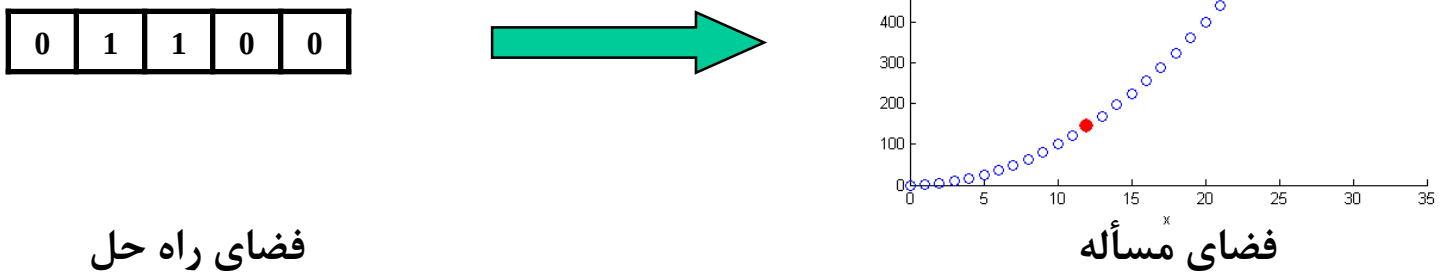
بیابید.



حل مسأله بهینه سازی

- **گام اول: تعیین شکل جواب**
  - نحوه نمایش جوابی که در جستجوی آن هستیم چیست؟
    - یک عدد صحیح
    - یک عدد صحیح در بازه [0,31]
    - یک رشته باینری به طول ۸ بطوریکه جواب مورد نظر به صورت باینری در آن کد شده است
    - یک رشته باینری به طول ۵ بطوریکه جواب مورد نظر به صورت باینری در آن کد شده است
    - روش انتخاب شده را **روش بازنمایی** (representation) گوییم.

- **گام اول: تعیین شکل جواب**
  - یک رشته باینری به طول ۵ بطوریکه جواب مورد نظر به صورت باینری در آن کد شده است
  - هر راه حل در فضای راه حل معادل موجودی در فضای مسأله است.



فضای راه حل **فضای مسأله**

حل مسأله بهینه سازی

- **گام دوم: تولید جمعیت اولیه**
  - تعیین تعداد جواب ها (منابع محدود) (np)
  - تولید np جواب ممکن
    - جمعیت اولیه را به صورت تصادفی تولید می کنیم.

```
for i = 1 to np
        for j = 1 to 5
                if (rand() > 0.5)
                        ind(i,j) = 0;
                else
                        ind(i,j) = 1;
                end
        end
end
```

حل مسأله بهینه سازی

- **گام سوم: ارزیابی جواب های تولید شده**
  - ارائه تابعی به فرم

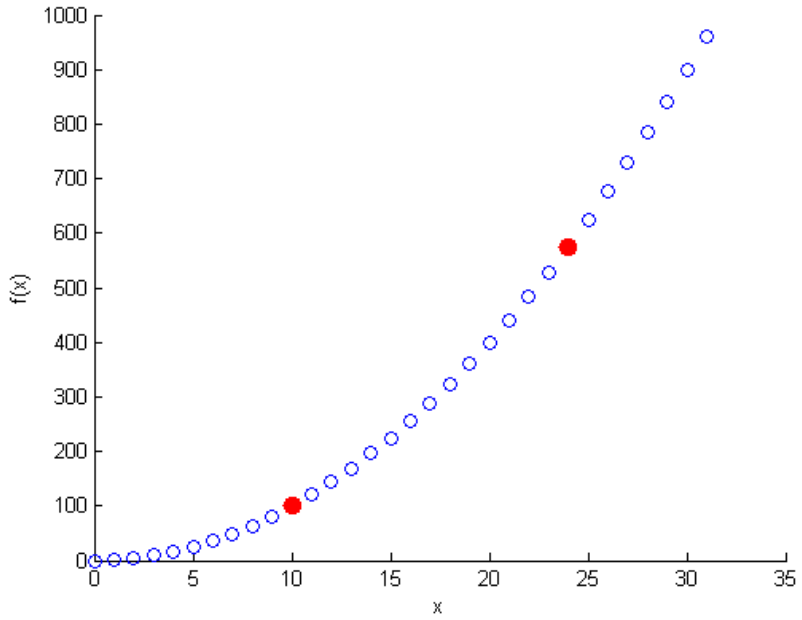
$$eval(x)$$

که ورودی آن یک فرد از جمعیت و خروجی آن یک عدد حقیقی است.

- تابع $eval$ باید به گونه ای باشد که اگر از دید کاربر (فضای مسأله) جواب i از جواب j بهتر باشد مقدار نسبت داده شده به جواب i از مقدار نسبت داده شده به جواب j بیشتر باشد.
- در مسائل ماکزیمم سازی تابع می توان از تابع هدف به عنوان تابع رزیابی استفاده کرد.
- تابع فوق را تابع **ارزیابی شایستگی (fitness evaluation)** و مقدار نسبت داده شده به هر فرد توسط این تابع را **شایستگی (fitness)** فرد نامیده می شود.

حل مسأله بهینه سازی

فرد		شایستگی
0 1 0 1 0	$eval(x)$	100
1 1 0 0 0		576



حل مسأله بهینه سازی

- **گام چهارم: انتخاب والدین**
  - هدف انتخاب تعدادی از افراد جمعیت حاضر برای **تولید فرزندان** جدید است.
  - معمولاً به صورت **تصادفی** صورت می گیرد (همه افراد احتمال انتخاب شدن دارند).
  - در اغلب موارد **متناسب با شایستگی** افراد صورت می گیرد.
  - به عنوان مثال به هر فرد احتمال انتخابی برابر با

$$p_i = \frac{fit_i}{\sum_j fit_j}$$

نسبت داده و به تعداد والدین مورد نیاز فرآیند انتخاب تصادفی را تکرار می کنیم.

  - ممکن است یک فرد بیش از یک بار به عنوان والد انتخاب شود.

حل مسأله بهینه سازی

- نمونه ای از خروجی مرحله انتخاب والدین

String no.	Initial population	x Value	Fitness $f(x) = x^2$	$Prob_i$	Expected count	Actual count
1	0 1 1 0 1	13	169	0.14	0.58	1
2	1 1 0 0 0	24	576	0.49	1.97	2
3	0 1 0 0 0	8	64	0.06	0.22	0
4	1 0 0 1 1	19	361	0.31	1.23	1
Sum			1170	1.00	4.00	4
Average			293	0.25	1.00	1
Max			576	0.49	1.97	2

حل مسأله بهینه سازی

- **گام پنجم: تولید فرزندان از والدین انتخاب شده**

  - **هدف:** ایجاد موجودات جدید به امید یافتن موجودی با شایستگی بیشتر نسبت به والدین خود
  - ویژگی های فرزندان با ایجاد تغییر در ویژگی های والدین یا به ارث به بردن ویژگی ها از والدین حاصل می شود.
  - **دو عملگر عمده در تکامل:**
    - بازترکیبی (Recombination)
    - جهش (Mutation)

حل مسأله بهینه سازی

- **عملگر بازترکیبی**
  - بر روی دو والد انتخاب شده اعمال می شود.
  - با ترکیب ویژگی های والدین فرزندان جدید ایجاد می کند.
  - در مثال مورد بررسی هر بیت بیانگر یک ویژگی فرض می شود.
  - یک نقطه تصادفی در هر دو والد انتخاب و ویژگی های دو والد از محل انتخاب شده جابجا می شوند.

حل مسأله بهینه سازی

- اعمال عملگر بازترکیبی به والدین انتخاب شده در گام چهارم

String no.	Mating pool	Crossover point	Offspring after xover	x Value	Fitness $f(x) = x^2$
1	0 1 1 0 | 1	4	0 1 1 0 0	12	144
2	1 1 0 0 | 0	4	1 1 0 0 1	25	625
2	1 1 | 0 0 0	2	1 1 0 1 1	27	729
4	1 0 | 0 1 1	2	1 0 0 0 0	16	256
Sum					1754
Average					439
Max					729

حل مسأله بهینه سازی

- **عملگر جهش:**
  - بر روی فرزندان تولید شده در مرحله قبل اعمال می شود.
  - برخی ویژگی ها (بیت ها) را به صورت تصادفی تغییر می دهد.
  - برای تغییر هر ویژگی احتمالی در نظر گرفته می شود که از آن به عنوان احتمال جهش (p_m) یاد می شود.

حل مسأله بهینه سازی

- اعمال عملگر جهش به فرزندان تولید شده در بخش قبل

String no.	Offspring after xover	Offspring after mutation	x Value	Fitness $f(x) = x^2$
1	0 1 1 0 0	1 1 1 0 0	26	676
2	1 1 0 0 1	1 1 0 0 1	25	625
2	1 1 0 1 1	1 1 0 1 1	27	729
4	1 0 0 0 0	1 0 1 0 0	18	324
Sum				2354
Average				588.5
Max				729

حل مسأله بهینه سازی

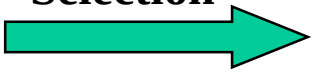
- **گام ششم: انتخاب بازماندگان**
  - جمعیت موجود = ۴ نفر جمعیت اولیه + ۴ فرزند تولید شده
  - به دلیل **محدودیت منابع** (اعضاء جمعیت) ناگزیر به انتخاب هستیم.
  - این مرحله از انتخاب، **انتخاب بازماندگان** (Survivor Selection) نامیده می شود.
  - انتخاب می تواند متناسب با شایستگی یا کاملاً بر اساس شایستگی باشد.

حل مسأله بهینه سازی

- **گام ششم: انتخاب بازماندگان**
  - جمعیت حاصل از جمعیت اولیه و فرزندان را بر اساس شایستگی مرتب و np نفر با شایستگی بیشتر را انتخاب می کنیم.

Previous Population	Fitness
0 1 1 0 1	169
1 1 0 0 0	576
0 1 0 0 0	64
1 0 0 1 1	361
Offspring	**Fitness**
1 1 1 0 0	676
1 1 0 0 1	626
1 1 0 1 1	729
1 0 1 0 0	324

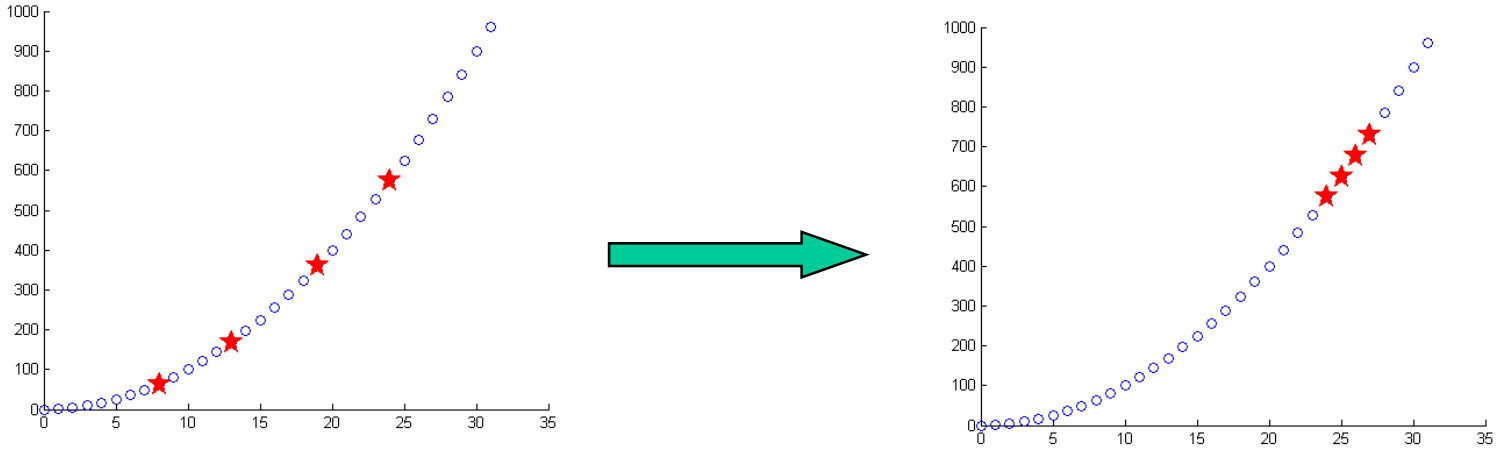
Survivor Selection



Next Population	Fitness
1 1 0 1 1	729
1 1 1 0 0	676
1 1 0 0 1	626
1 1 0 0 0	576

حل مسأله بهینه سازی

- نتیجه یک تولید نسل الگوریتم تکاملی



حل مسأله بهینه سازی

- **گام هفتم: تکرار الگوریتم تا رسیدن به یک شرط خاتمه**
  - اجرای الگوریتم با تکرار گام های سوم تا ششم ادامه می یابد.
    - ارزیابی
    - انتخاب والدین
    - اعمال عملگر ها
    - انتخاب بازماندگان
  - **شرط خاتمه**
    - در اینجا از تعداد تکرار حلقه اصلی (تعداد تولید نسل ها) استفاده می شود.

فهرست مطالب

چارچوب کلی یک الگوریتم تکاملی

۱) تعیین روش بازنمایی

۲) ایجاد جمعیت اولیه

۳) تکرار تا برقراری شرط خاتمه

۳-۱) ارزیابی جمعیت

۳-۲) انتخاب والدین

۳-۳) اعمال عملگرهای ژنتیکی و تولید فرزندان

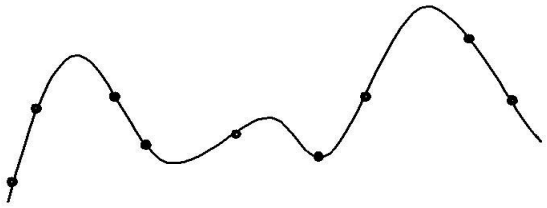
۳-۴) انتخاب بازماندگان

فهرست مطالب

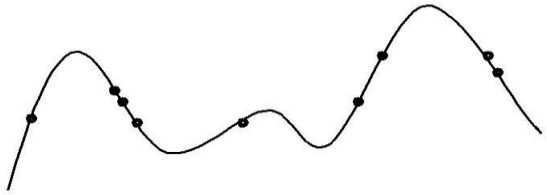
ویژگی های کلی الگوریتم های تکاملی

- مراحل جستجو در بهینه سازی در یک فضای یک بعدی

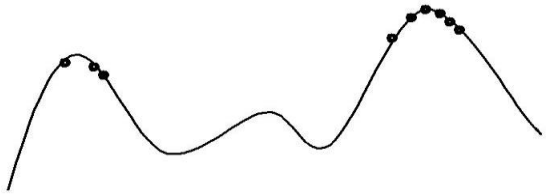
تکرارهای اولیه:
توزیع تقریباً تصادفی جمعیت



تکرارهای میانی:
استقرار جمعیت در اطراف تپه ها

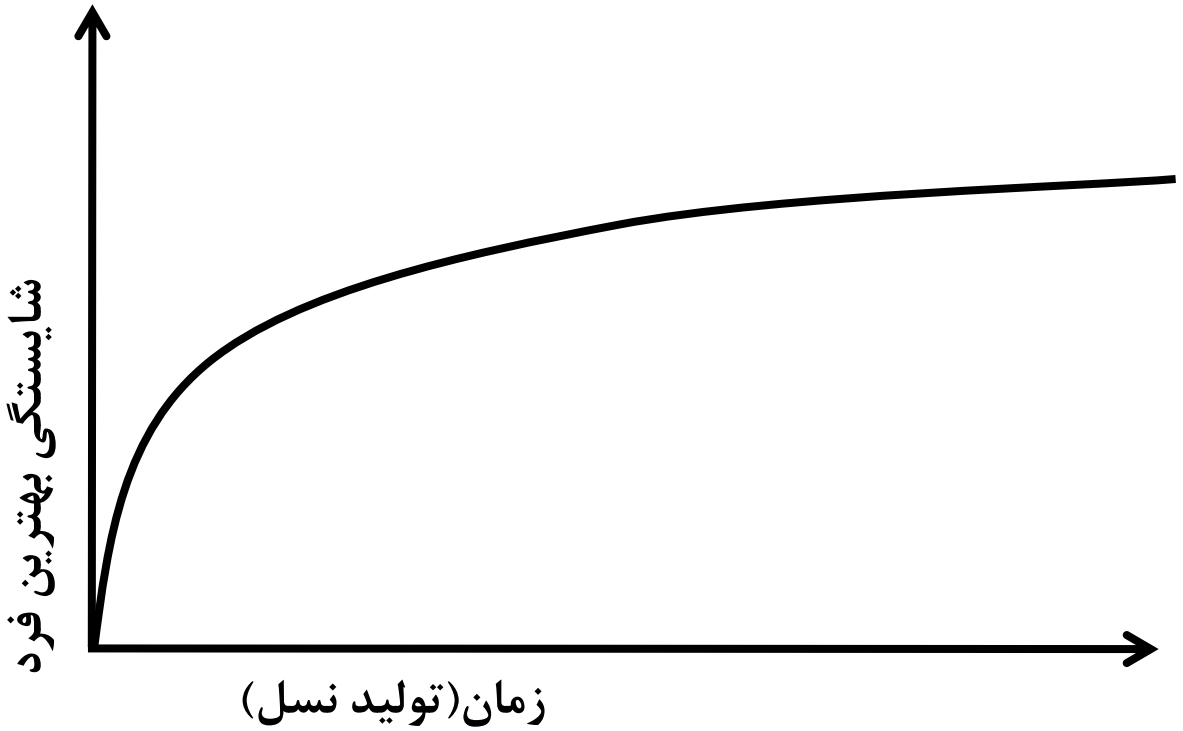


تکرارهای پایانی:
تمرکز جمعیت در اطراف تپه های بلند



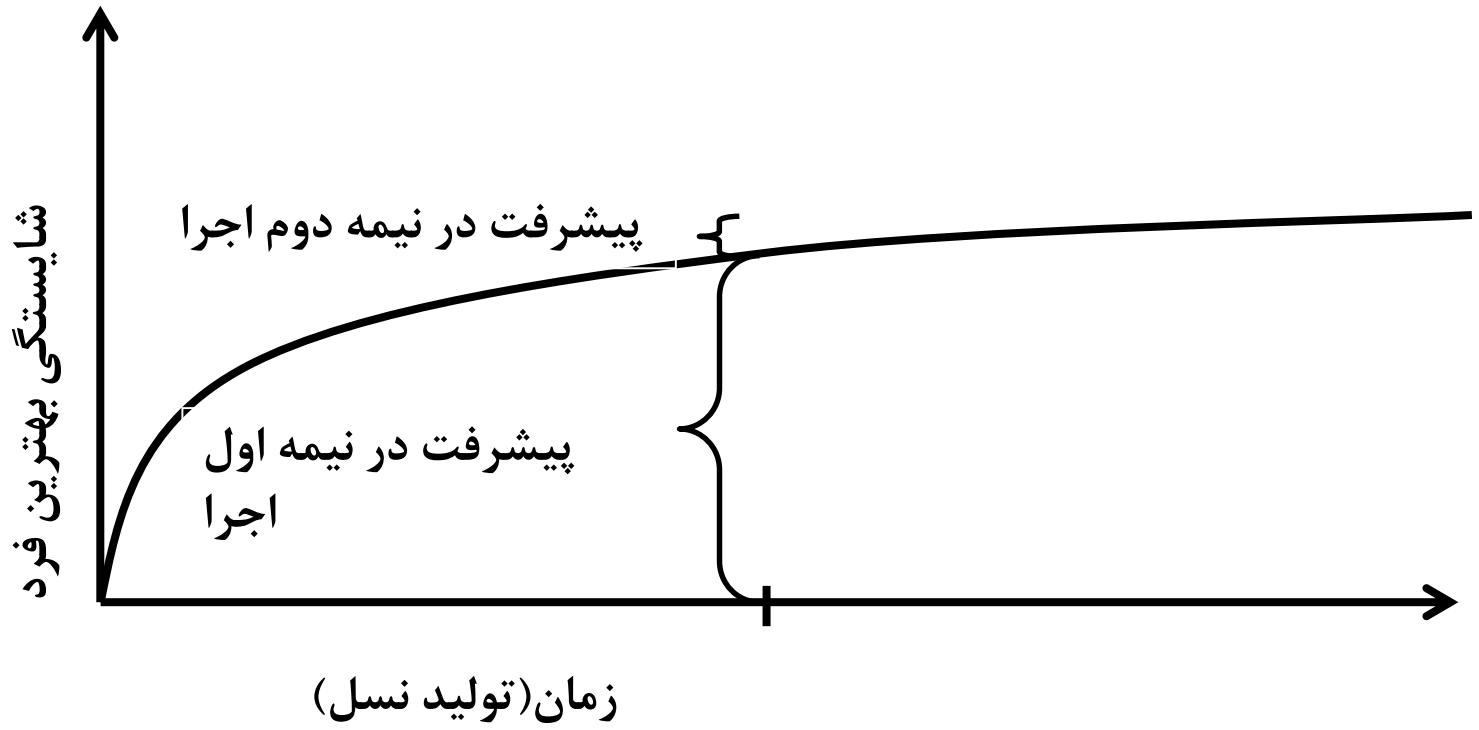
ویژگی های کلی الگوریتم های تکاملی

- نتیجه نوعی اجرای الگوریتم های تکاملی



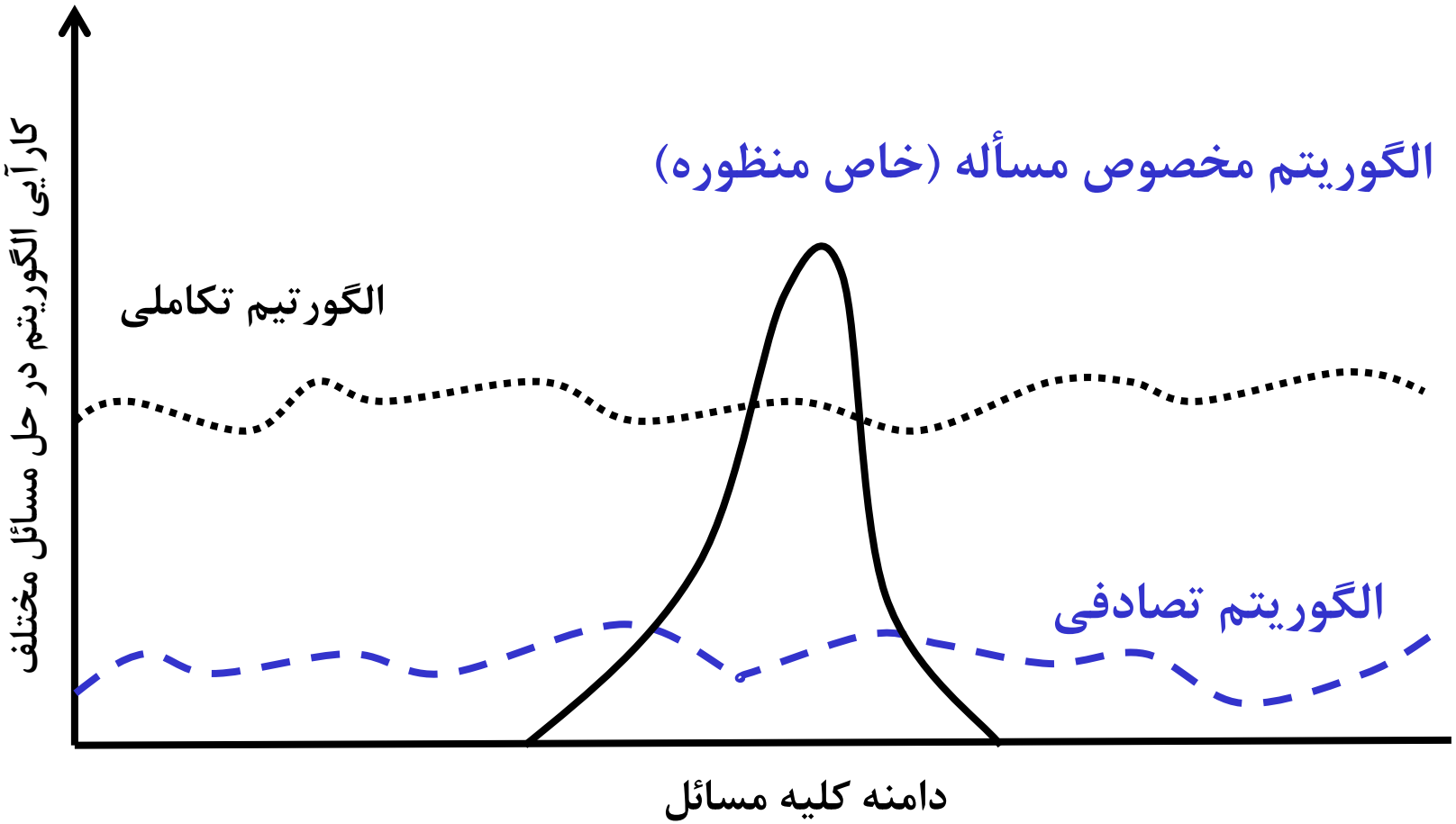
ویژگی های کلی الگوریتم های تکاملی

- اجرای طولانی مدت الگوریتم ژنتیکی یا تکرارهای متوالی آن؟



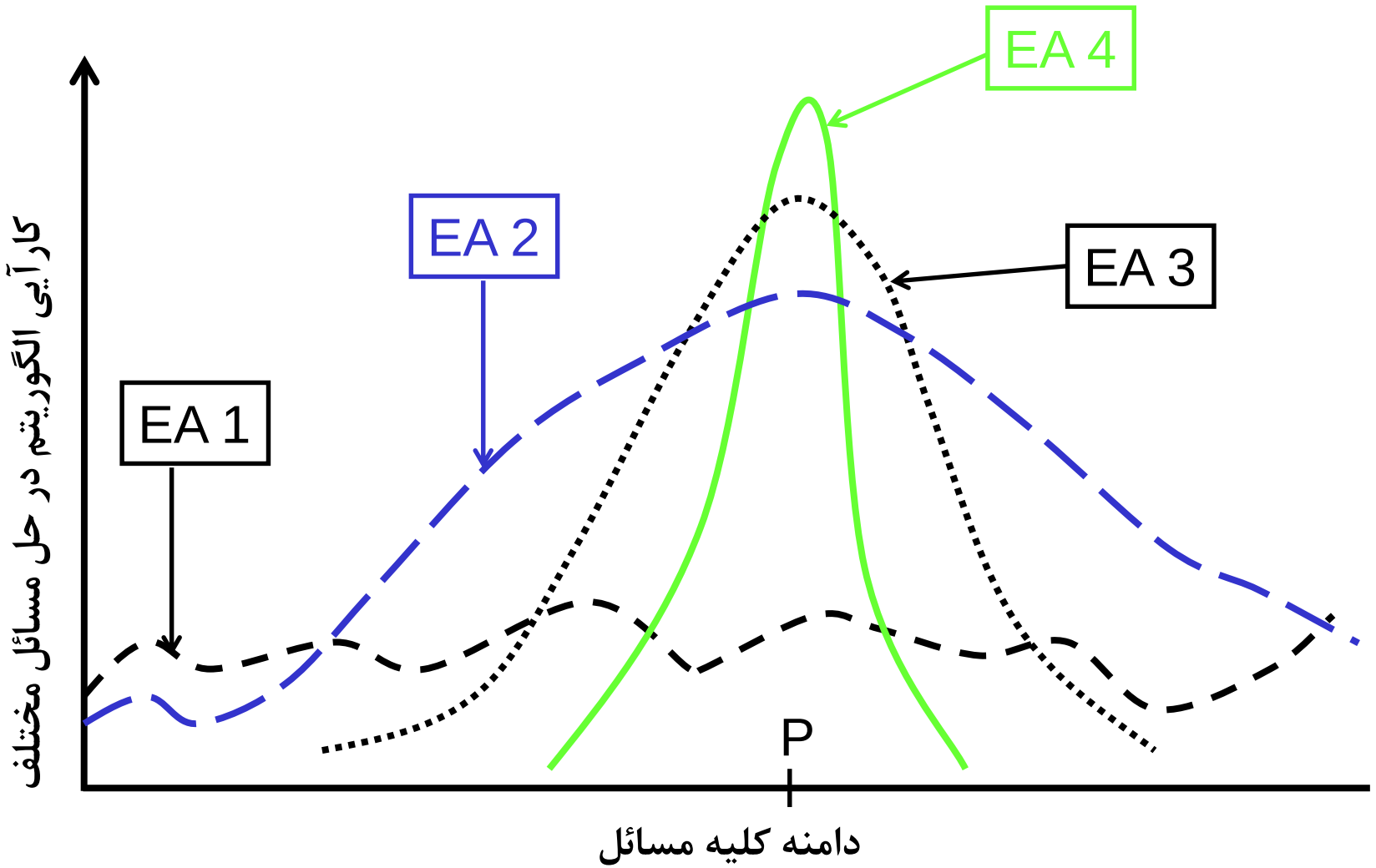
ویژگی های کلی الگوریتم های تکاملی

- الگوریتم تکاملی در مقایسه با سایر الگوریتم ها



ویژگی های کلی الگوریتم های تکاملی

- تأثیر ویژه سازی الگوریتم های تکاملی (استفاده از دانش قلمرو) در کارآیی



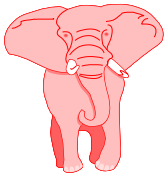
فهرست مطالب

- خلاصه مباحث قبل
- حل یک مسأله ساده با استفاده از الگوریتم های تکاملی
- چارچوب کلی یک الگوریتم تکاملی
- ویژگی های کلی الگوریتم های تکاملی
- **بازنمایی**
- جمعیت
- شروط خاتمه

بازنمایی

- **افراد دارای دو سطح وجودی هستند:**
  - ژنوتیپ: شیئی در فضای مسأله اصلی (جهان خارج)
  - فنوتیپ: کد مربوط به شیئ (که به عنوان کروموزوم یا DNA نیز شناخته می شود)

فنوتیپ



ژنوتیپ

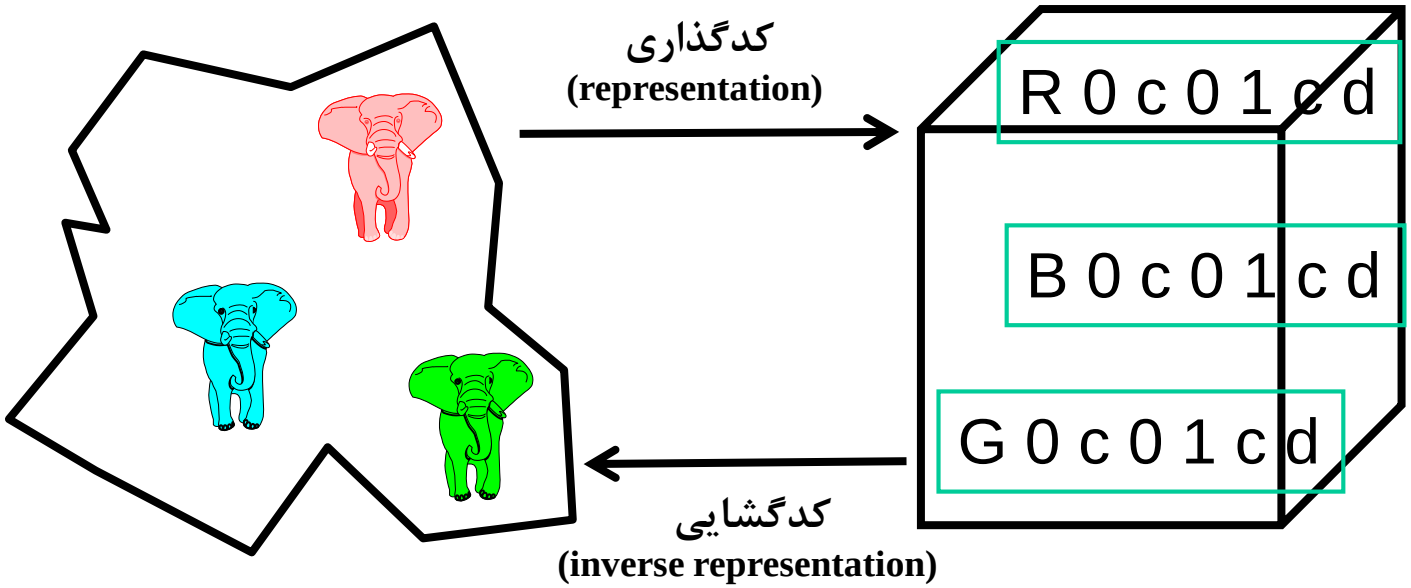
a d c a a c b

- روش بازنمائی بیانگر نحوه ارتباط این دو سطح وجودی فرد (نگاشت فضاهای مسآله و راه حل) است.
- روش بازنمائی بر انتخاب عملگرهای جهش و بازترکیبی تأثیر مستقیم دارد.
- برای آنکه الگوریتم امکان یافتن جواب بهینه را داشته باشد روش بازنمائی باید تمام راه حل های ممکن را پوشش دهد.

بازنمایی

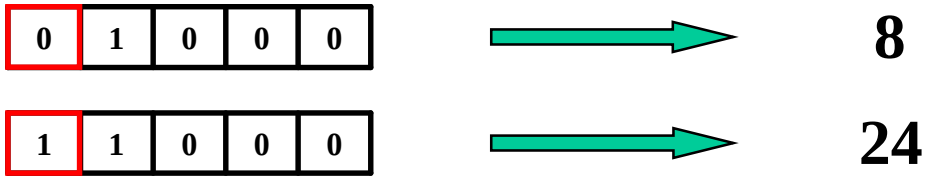
فضای مسأله(فنوتیپ)

فضای راه حل(ژنوتیپ)



بازنمایی

- **روشهای مختلف بازنمائی:**
  - **باینری**
    - ساده
    - اولین روش بازنمائی!
    - ایجاد کروموزوم های طولانی
    - نامناسب در جهش
    - وجود عملگرهای جهش و بازترکیبی فراوان برای آن



بازنمایی

- **روشهای مختلف بازنمائی:**
  - **کد گری**
    - تا حد زیادی مشابه روش ارائه باینری
    - مناسب برای جهش

بازنمایی

- **روشهای مختلف بازنمائی:**
  - **اعداد صحیح**
    - کاربرد در مسائل خاص
  - **اعداد حقیقی**
    - مناسب در مسائل بهینه سازی خصوصاً در فضاهای با ابعاد زیاد
  - **ماشین های متناهی حالت**
  - **ساختارهای درختی**
  - **هر روش دیگری که بتواند راه حل مورد نظر را نشان دهد.**

فهرست مطالب

- خلاصه مباحث قبل
- حل یک مسأله ساده با استفاده از الگوریتم های تکاملی
- چارچوب کلی یک الگوریتم تکاملی
- ویژگی های کلی الگوریتم های تکاملی
- بازنمایی
- **جمعیت**
- شروط خاتمه

جمعیت

- برای نگهداری راه حلهای ممکن مورد استفاده قرار می گیرد.
- معمولاً دارای اندازه ثابت و یک **multiset** از ژنوتیپ ها است.
- در برخی انواع الگوریتم های تکاملی نوعی رابطه مکانی (همانند گرید) نیز میان اعضاء جمعیت وجود دارد.
- تنوع در جمعیت بیانگر تعداد شایستگی، ژنوتیپ و یا فنوتیپ های متفاوت در جمعیت است.
- مقداردهی اولیه جمعیت:
  - بهتر است دارای توزیع یکنواخت و ترکیبات مختلف آلل ها را داشته باشد.
  - میتواند کاملاً تصادفی باشد و یا از راه حل های موجود و هیوریستیک های خاص مسأله استفاده کند.

- **آیا تلاش برای مقداردهی اولیه به شکل هوشمند به صرفه است؟**
  - اگر جواب های خوبی وجود داشته باشد ممکن است.
  - احتمال بایاس الگوریتم و گیر کردن در مینیمم محلی نیز وجود خواهد داشت.

F: شایستگی پس از مقداردهی اولیه هوشمند

T: زمان لازم برای رسیدن به شایستگی F در صورت استفاده از مقداردهی اولیه تصادفی

فهرست مطالب

شروط خاتمه

- **شرط خاتمه در هر تولید نسل بررسی می شود.**
- **می توان از هر یک از شروط زیر و یا هر ترکیبی از آنها استفاده کرد:**
  - رسیدن به یک سطح شایستگی خاص
  - حداکثر تعداد تولید نسل ها
  - رسیدن به یک حداقل تنوع
  - حداکثر تعداد تولید نسل بدون تغییر شایستگی