

# مدیریت محدودیت‌ها در الگوریتم‌های فرا ابتکاری

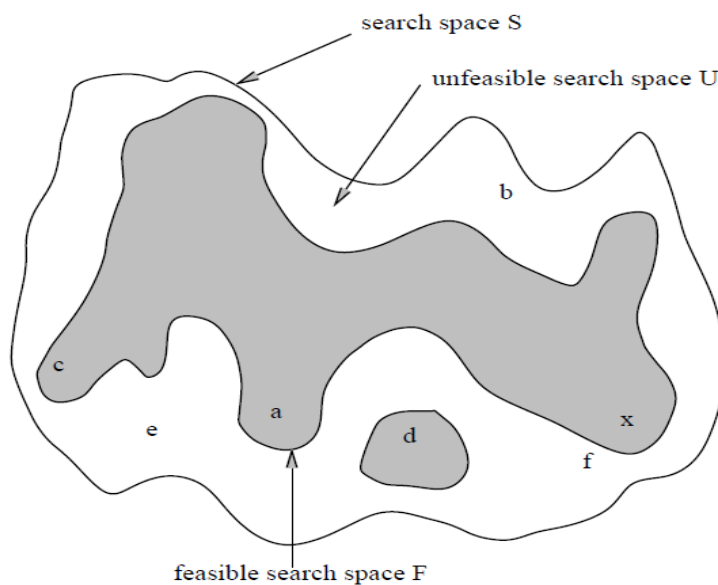
الگوریتم‌های فرا ابتکاری

حسین کریمی

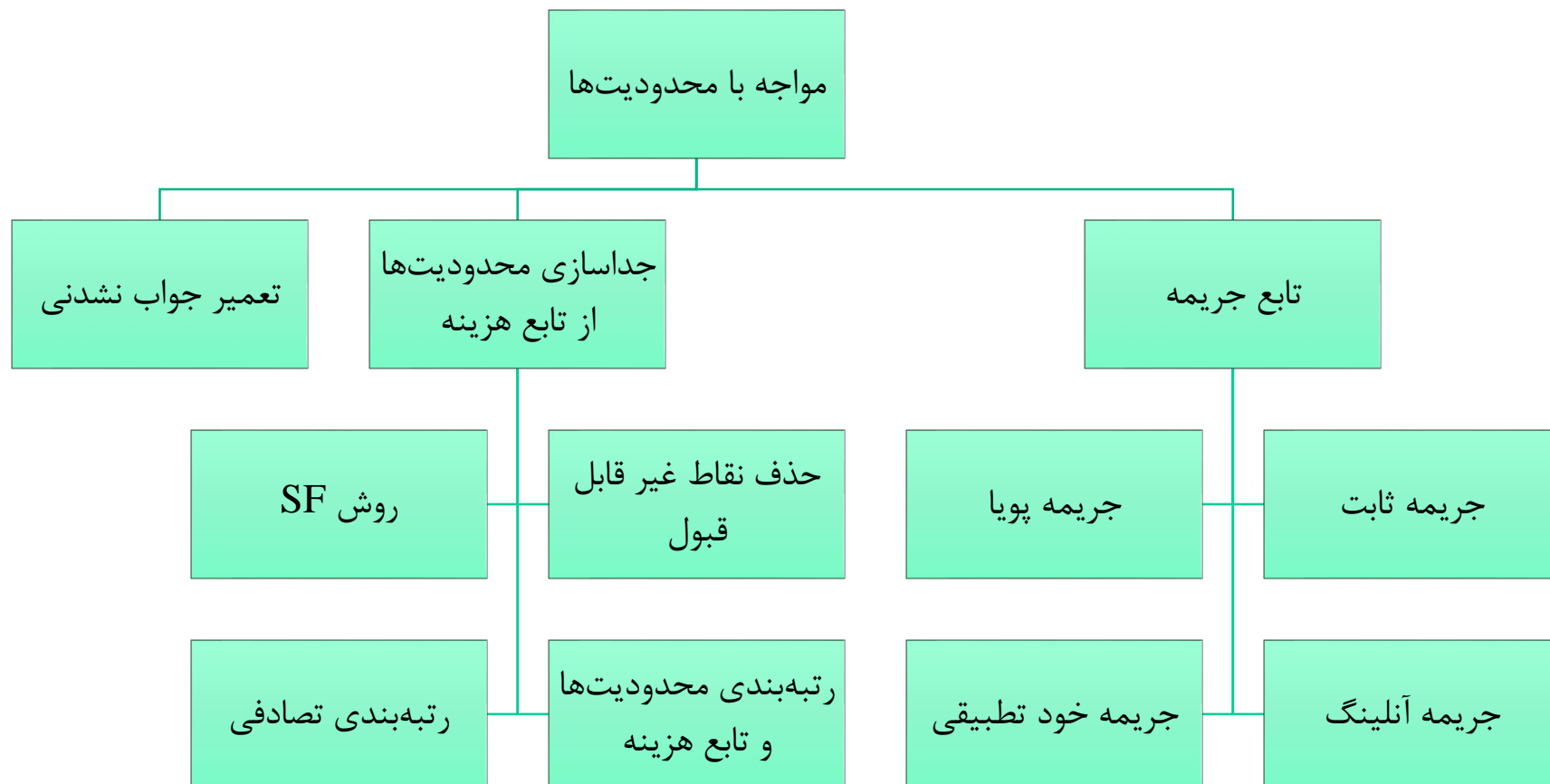
- تعریف مسائل بهینه‌سازی مقید
- روش‌های مدیریت محدودیت‌ها
- روش استفاده از تابع جریمه
- روش تعمیر جواب نشدنی
- جمع‌بندی

# تعریف مسائل بهینه‌سازی مقید

- محدود شدن متغیرهای مسئله به مجموعه‌ای از محدودیت‌ها
- تقسیم فضای جستجو به فضای جستجوی قابل قبول و غیر قابل قبول
- در نظر گرفتن دو تابع برازندگی مربوط به فضای قابل قبول و غیر قابل قبول

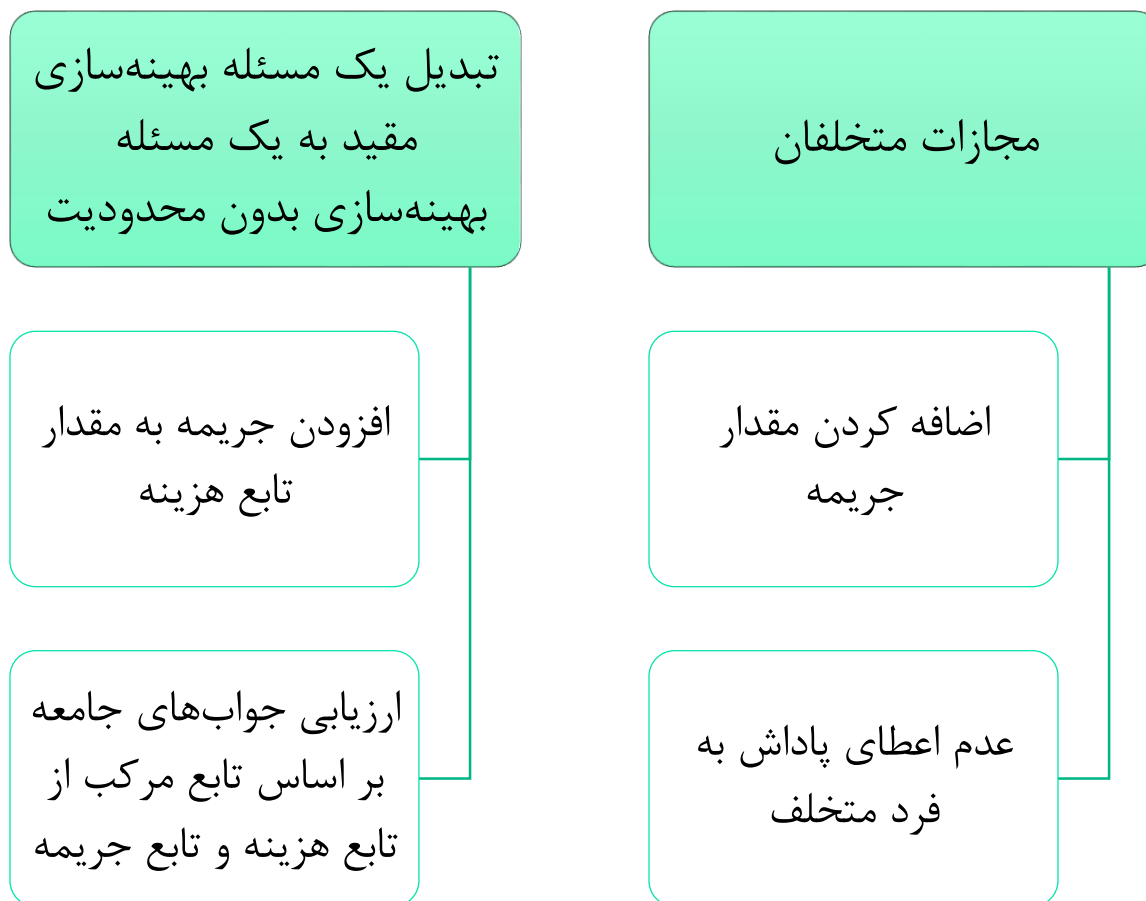


# روش‌های مدیریت محدودیت‌ها



# روش استفاده از تابع جریمه

ایده استفاده از تابع جریمه



# روش استفاده از تابع جریمه – ادامه

## روش جریمه ثابت (پاداش دهی)

■ نزول تابع هزینه متناسب با تعداد محدودیت رعایت شده

$$Eval(x) = f(x) + K - \sum_{i=1}^s \frac{K}{m}$$

$m$  : تعداد محدودیت‌ها

$K$  : میزان پاداش

■ معایب :

■ عدم توجه به میزان تخلفات و در نظر گرفتن مجازات یکسان برای همه ی متخلفان

■ نداشتن ساختاری برای تشویق جواب‌های نشدنی به رعایت محدودیت‌ها

# روش استفاده از تابع جریمه – ادامه

## روش جریمه ثابت (جریمه کردن)

- تعریف  $L$  سطح تخلف به ازای هر محدودیت.
- ایجاد یک ضریب پنالتی، به ازای هر سطح تخلف برای هر محدودیت؛ به طوریکه سطوح بالاتر دارای ضریب پنالتی بیشتر باشند.
- ایجاد جواب‌های تصادفی اولیه شامل پاسخ‌های قابل قبول و غیر قابل قبول.
- مثال: ارزیابی جواب‌های جامعه طبق رابطه زیر می‌تواند باشد.

$$Eval(x) = f(x) + \sum_{i=1}^L \sum_{j=1}^m P_{ij}$$

$P_{ij}$  ضریب پنالتی به ازای  $i$  امین سطح از  $j$  امین محدودیت

# روش استفاده از تابع جریمه – ادامه

## روش جریمه ثابت (جریمه کردن)

### ■ معایب این روش

- وجود تعداد زیادی پارامتر
- وابستگی زیاد به مقادیر پارامترها
- کاهش ضرایب پنالتی موجب امکان همگرایی به پاسخهای غیر قابل قبول می شود.
- افزایش ضرایب پنالتی موجب از دست رفتن تنوع در جمعیت می شود.



# روش استفاده از تابع جریمه – ادامه

## روش جریمه پویا

- ایده اصلی این روش افزایش تدریجی جریمه تخلفات متناسب با گذر زمان و نزدیک شدن به پاسخ بهینه است.
- مراحل این روش:

ایجاد جواب‌های اولیه متشکل از شدنی و نشدنی

تعیین مقادیر ثابت همانند  $\alpha$ ،  $\beta$  و  $C$

ارزیابی جواب‌ها با رابطه‌ای همانند رابطه زیر

$$Eval(x) = f(x) + (C \times t)^\alpha \sum_{j=1}^m (\Delta_j(x))^\beta$$

$t$  زمان حال جستجو و  $\Delta_j$  میزان تخلف محدودیت  $j$ ام است.

# روش استفاده از تابع جریمه-ادامه

## روش جریمه پویا

### ■ مزایا

■ رفع مشکلات روش جریمه ثابت

- کاهش قابل ملاحظه تعداد پارامترهای مورد نیاز
- افزایش تدریجی جریمه‌ها یا کاهش تدریجی جریمه موجب حفظ تنوع می‌شود.
- توجه به میزان تخلفات موجب تشویق جواب‌ها به کاهش جریمه و رعایت محدودیت‌ها می‌شود.

### ■ معایب

■ حساسیت زیاد به مقادیر پارامترها

# روش استفاده از تابع جریمه – ادامه

## روش جریمه آنلینگ

- این ایده به نوعی زیر مجموعه روش جریمه پویا و حالت خاصی از آن است.
- ایده این روش برگرفته شده از روش سرد شدن فلزات در فرآیند فرآوری آنها است.
- نمونه‌ای از روش جریمه کردن در این روش به صورت زیر است.

$$Eval(x, \tau) = f(x) + \frac{1}{\tau} \sum_{j=1}^m (\Delta_j(x))^2$$

$\tau$  دمای حال الگوریتم است که در ابتدا برابر  $T$  که دمای اولیه محیط و در انتها برابر  $T_f$  خواهد بود.

- یکی از روش‌ها جهت رسیدن به دمای انتهایی به صورت  $\tau = r \times \tau$  است که  $r$  نرخ تغییر دما است.

# روش استفاده از تابع جریمه – ادامه

## روش جریمه آنلینگ

### ■ مزایا

■ همانند روش جریمه پویا است.

### ■ معایب:

- حساسیت زیاد به مقادیر پارامترها
- نیاز به تنظیم دمای اولیه و دمای انتهایی
- نیاز به یک سیستم تغییر دمای مناسب
- تاثیر زیاد سیستم تغییر دما بر نتایج حاصل

# روش استفاده از تابع جریمه-ادامه

## روش جریمه خود تطبیقی

- هدف در این روش این است که خود الگوریتم بتواند جریمه‌ای متناسب با شرایط خود اعمال نماید.
- مثال: در یک جستجوی گروهی می‌توان جریمه خودتطبیقی زیر را تعریف کرد.

$$Eval(x, \tau) = f(x) + s(x)$$

$$s(x) = \begin{cases} d(x) & r = 0 \\ \sqrt{n(x)^2 + d(x)^2} & o.w \end{cases}$$

$$n(x) = \frac{f(x) - f_{\min}}{f_{\max} - f_{\min}}$$

$d(x)$ : مقدار تخطی کلی یک جواب از گروه

$n(x)$ : فاصله نرمال شده نقاط تا بهترین نقطه موجود در

جمعیت فعلی از نظر تابع هدف

$r$ : نسبت تعداد نقاط شدنی به کل نقاط

# جداسازی محدودیت‌ها از تابع هزینه

## ■ ایده کلی

■ بررسی مجزای دو معیار تابع هدف و میزان تخلفات

روش حذف جواب‌های نشدنی (مرگ)

## ■ ایده مجازات مرگ

■ اعدام جواب‌های نشدنی بدون در نظر گرفتن نوع و میزان جرم.

ویژگی‌ها

■ ساده

■ کم هزینه

■ تولید مجدد جواب تا زمان ایجاد یک جواب شدنی

# جداسازی محدودیت‌ها از تابع هزینه-ادامه

## روش حذف جواب‌های نشدنی (مرگ)

### ■ مزایا:

■ مناسب برای:

■ محیط‌های محدب

■ جوامعی با تعداد اعضای نشدنی کافی و کم

### ■ معایب:

■ هزینه‌بر به دلیل تولید مجدد جواب‌ها

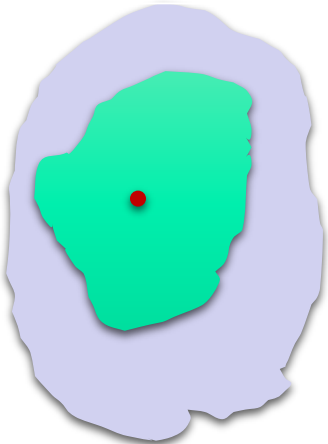
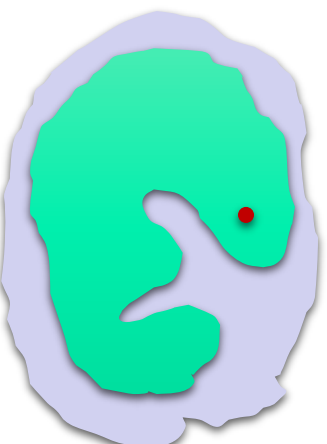
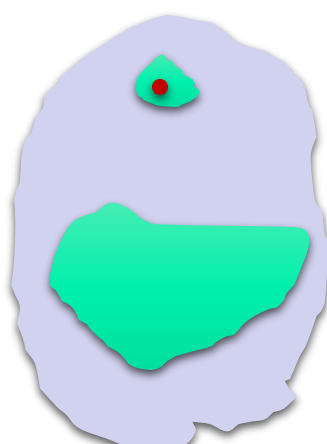
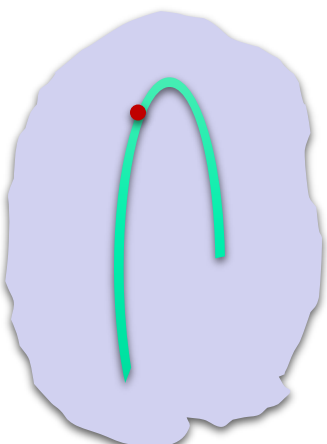
■ امکان نرسیدن به پاسخ بهینه به دلیل:

■ حذف نقاط نشدنی

■ عدم توانایی عبور از نقاط نشدنی و رسیدن به پاسخ بهینه

# جداسازی محدودیت‌ها از تابع هزینه-ادامه

## ■ معرفی برخی از فضاهاى جواب

محدب	غیر محدب	غیر محدب	غیر محدب
تعداد کافی جواب‌های شدنی پیوستگی فضای قابل قبول	تعداد کافی جواب‌های شدنی پیوستگی فضای قابل قبول	تعداد کافی جواب‌های شدنی پراکندگی فضای قابل قبول	تعداد ناکافی جواب‌های شدنی پیوستگی فضای قابل قبول
			
۱	۲	۳	۴



# جداسازی محدودیت‌ها از تابع هزینه-ادامه

## روش SF

■ ایده اصلی :

- استفاده از قانون برتری جواب‌های شدنی نسبت به جواب‌های نشدنی
- فرد  $A$  از فرد  $B$  برتر است اگر:

$$A \in F, B \in S - F \quad \blacksquare$$

$$f(A) < f(b), A, B \in F \quad \blacksquare$$

$$d(A) < d(b), A, B \in S - F \quad \blacksquare$$

# جداسازی محدودیت‌ها از تابع هزینه-ادامه

## روش SF

### ■ ویژگی‌ها

- همگرایی به نقاط نشدنی در محیط‌های نامناسب
  - محیط نامناسب = کوچک بودن نسبت
  - راه حل: ایجاد جمعیت اولیه به روش قبل (تکثیر یک نقطه شدنی)
- تولید جواب‌های بهینه به شرط مناسب بودن محیط
  - محیط مناسب = وجود حداقل یک فرد نشدنی در هر مجموعه جواب

# جداسازی محدودیت‌ها از تابع هزینه-ادامه

## رتبه‌بندی محدودیت‌ها و تابع هزینه

■ ایده اصلی:

- بهینه‌سازی بر اساس تک تک محدودیت‌ها
- در نظر گرفتن تابع هزینه به عنوان هدف نهایی

# جداسازی محدودیت‌ها از تابع هزینه-ادامه

## رتبه‌بندی محدودیت‌ها و تابع هزینه

■ مراحل این روش

■  $j=1$

■ ایجاد جمعیت اولیه به صورت تصادفی

■ رتبه‌بندی محدودیت‌ها

■ تنظیم آستانه تعویض محدودیت برای تک تک محدودیت‌ها

■ رشد و اصلاح جمعیت با هدف رعایت محدودیت  $j$ ام تا رسیدن به حد آستانه

■ افزایش  $j$  (تا زمانی که  $j < m$ ) و تکرار دو گام قبلی

■ رشد و اصلاح جمعیت با معیار کمینه کردن تابع هزینه

# جداسازی محدودیت‌ها از تابع هزینه-ادامه

## رتبه‌بندی محدودیت‌ها و تابع هزینه

### ■ مزایا

- حفظ تنوع حتی در جوامع کوچک و پراکنده
- اطمینان از رسیدن به پاسخ بهینه

### ■ معایب

- بالاسری محاسباتی زیاد برای جوامع پر جمعیت
- تاثیر نحوه رتبه‌بندی محدودیت‌ها در بهبود کارایی

# جداسازی محدودیت‌ها از تابع هزینه-ادامه

## رتبه‌بندی تصادفی

### ■ ایده اصلی

- ارزیابی احتمالاتی بر اساس
- فقط میزان تخلفات و بدون در نظر گرفتن تابع هدف
- فقط مقدار تابع هدف

### ■ روند کلی

- تعیین احتمال  $P$
- بهینه‌سازی بر اساس تابع هدف با احتمال  $P$
- ارزیابی آخرین مرحله تنها با معیار تابع هدف

### ■ مزایا

- سادگی

### ■ معایب

- کاهش کارایی نسبت به روش قبل

# روش تعمیر جواب نشدنی

## ■ ایده اصلی

- هم نشین کردن جواب‌های متخلف با نزدیکترین همسایه شدنی به امید اصلاح تدریجی آنها در طول زمان

## ■ روند الگوریتم

- یافتن نزدیکترین نقطه قابل قبول به نقطه غیر قابل قبول
- نگاشت مقدار تابع هدف برای نقطه شدنی، به عنوان شایستگی نقطه نشدنی
- انتخاب بازماندگان تنها با معیار کمترین شایستگی

## ■ مزایا و معایب

- وابسته به مسئله



- بررسی مهمترین روش‌های موجود برای مواجهه با محدودیت‌ها در مسائل بهینه سازی مقید.
- وابستگی روش‌های مختلف مواجهه با مسائل بهینه سازی مقید
- بنابراین
  - تقسیم جمعیت به تعدادی زیر جمعیت و به کارگیری مجموعه‌ای از روش‌های موجود به صورت موازی
  - ویژگی‌های مهم به هنگام طراحی روش‌های جدید
    - کلی بودن روش
    - کارایی مناسب ( تعادل بین هزینه و کارایی)
    - بیان صریح میزان کارایی در مسائل مختلف

