الگوریتم ژنتیک

Genetic Algorithm

الگوریتم‌های فراابتکار ی – الگوریتم ژنتیک

دکتر مصطفی زندیه – دانشگاه شهید بهشتی

پیشینه GA

- ایده اصلی الگوریتم ژنتیک (GA) مبتنی بر نظریه تکاملی داروین[1] (۱۸۵۹) است.
- نظریه تکاملی داروین:
  - آن دسته از صفات طبیعی که با طبیعت سازگاری بیشتری دارند. شانس بقای بیشتری دارند.
  - نظریه تکاملی داروین بصورت تحلیلی اثبات نشده است اما از نظر تجربی تائید شده است.
- بعبارتی افراد یک جامعه (انسانی/حیوانی/...) از طریق جفتگیری نسل جدیدی ایجاد می‌کنند.
  - در اغلب موارد افراد نسل جدید، سازگاری بیشتری با طبیعت دارند.
  - در موارد استثنایی ممکن جهش‌هایی[2] در خصوصیات یک فرد رخ می‌دهد.
    - اغلب افراد جهش یافته با طبیعت ناسازگارند.
    - در موارد نادر سازگاری بسیار بالایی دارند.

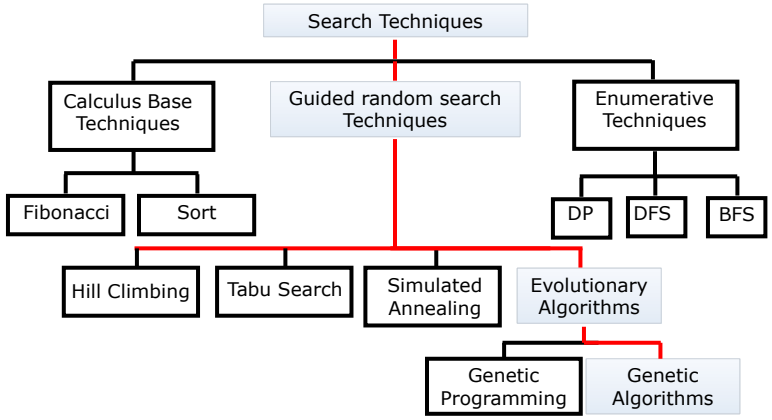
1- Darwin's Theory of Evolution 2- Mutation

پیشینه GA

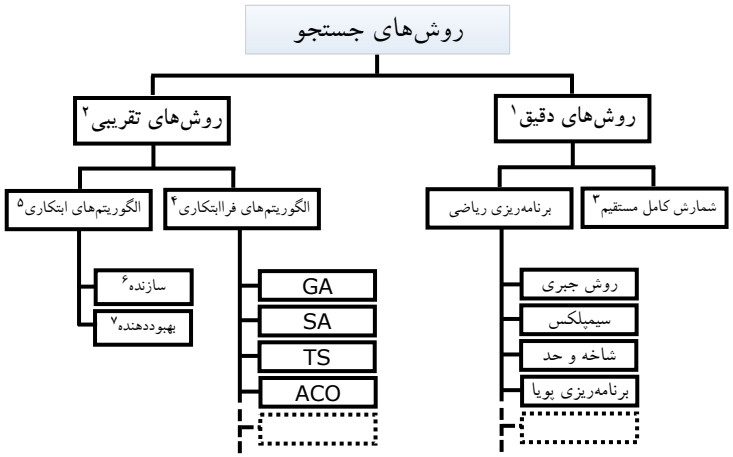
- بطور خلاصه در هر نسل، شانس انتقال خصوصیات گونه‌های بهتر به نسل بعد بیشتر است و گونه‌های با خصوصیات بد بتدریج از بین می‌روند.
  - در نتیجه با گذشت زمان خصوصیات گونه‌ها تکامل می‌یابند.
  - این فرآیند تکاملی، یک فرآیند تصادفی و همگرا است.
- در واقع الگوریتم ژنتیک، شبیه‌سازی این فرآیند تکامل بیولوژیکی[1] برای بهینه‌سازی است (هلند، دیجونگ و گلدبرگ (۱۹۷۰)).
  - الگوریتم ژنتیک در طبقه روش‌های جستجوی تصادفی هدایت شده[2] قرار دارد.
  - این الگوریتم بخصوص برای بهینه‌سازی مسائل پیچیده با فضای جستجوی ناشناخته مناسب است.

1- Biological Evolutionary Process 2- Guided Random Search Technique

طبقه‌بندی روش‌های جستجو



طبقه‌بندی روش‌های جستجو



1- Exact 2- Sub-optimal 3- Enumeration 4- Metaheuristic 5- Heuristic 6- Constructive 7- Improvement

اجزای یک الگوریتم ژنتیک

- روشی برای نمایش جواب[۱]
  - کدگذاری جواب‌ها در قالب یک رشته (بردار/ ماتریس) به عنوان کروموزوم[۲] شناخته می‌شود.
  - مجموعه‌ای از این کروموزوم‌ها در هر نسل الگوریتم را یک جمعیت[۳] می‌نامند.
- تابعی برای ارزیابی جواب‌ها
  - میزان سازگاری کروموزوم‌ها توسط تابع برازندگی[۴] ارزیابی می‌شود.
  - تابع برازندگی را با توجه به تابع هدف مسئله بهینه‌سازی محاسبه می‌کنند.
- استراتژی انتخاب[۵]
  - در هر نسل به کروموزوم‌های برتر، باید شانس بیشتری برای زاد و ولد داده شود.

1- Representation Scheme 2- Chromosome 3- Population 4- Fitness Function 5- Selection Strategy

اجزای یک الگوریتم ژنتیک

- استراتژی انتخاب مناسب می‌تواند تمرکزدهی[1] و تنوع بخشی[2] را در هر نسل از جواب‌ها تامین کند.
- عملگرهای ژنتیکی
  - فرایند زاد و ولد و ایجاد نسل جدید جواب‌ها با استفاده از عملگرهای ژنتیکی سه گانه (تقاطع[3]، جهش[4] و کپی) انجام می‌شود.
- تنظیم پارامتر[5]
  - عملکرد الگوریتم ژنتیک، بستگی به ترکیب مقادیر پارامترهای الگوریتم دارد.
  - ترکیب مقادیر پارامترها روی پایداری، کیفیت جواب و میزان منابع محاسباتی مورنیاز (زمان حل، حافظه و ...) الگوریتم تاثیرگذار است.
  - پارامترهای الگوریتم را با استفاده از روش‌های طراحی آزمایش تنظیم می‌کنند.

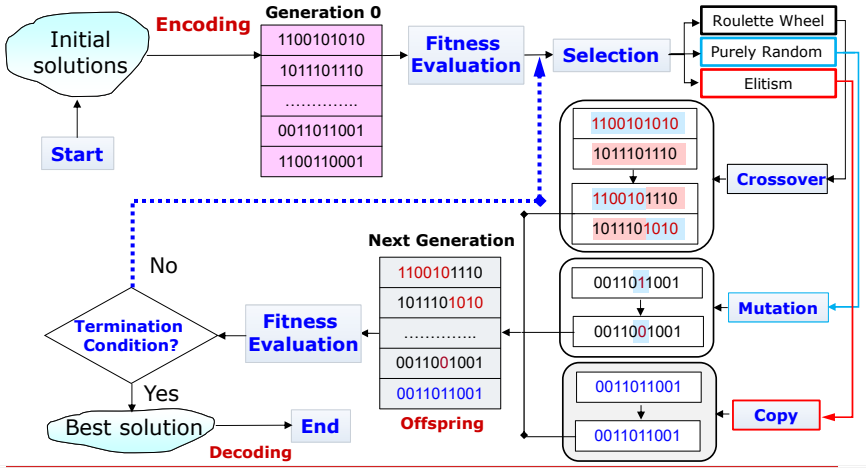
1- Intensification 2- Diversification 3- Crossover 4- Mutation 5- Parameter Tuning

الفاظ معادل (استعاره)[1]

طبیعت	الگوریتم ژنتیک
محیط	مسئله بهینه‌سازی
افرادی[2] که درمحیط زندگی می‌کنند	جواب‌های موجه
درجه سازگاری هر فرد با محیط زندگی	کیفیت/ برازندگی جواب
جمعیتی از افراد	مجموعه‌ای از جواب‌های موجه
انتخاب، ترکیب و جهش در فرآیند تکامل	عملگرهای ژنتیکی
تکامل جمعیت در راستای سازگاری بیشتر با محیط	بکارگیری عملگرهای ژنتیکی بطور مکرر روی مجموعه‌ای از جواب‌های موجه

1- Metaphor 2- Individuals

ساختار عمومی الگوریتم ژنتیک



شبه‌کد الگوریتم ژنتیک

[Initialization]

[Initialize Parameters] (*PopSize, Numgen, Pc, Pm, StopCriteria,..*)

[Initialize Population] Generate *PopSize* chromosomes, randomly.

[Evaluation] Evaluate the fitness of each chromosome.

[New Generation]

Repeat

[Selection] Select Parents based on selection strategy.

[Crossover]Produce (*PopSize* * *Pc*) of offspring with Crossover.

[Mutation] Produce (*PopSize* * *Pm*) of offspring with Mutation.

[Reproduction] Copy remaining chromosomes based on elitism.

[Replacing] Place new offspring in the new population.

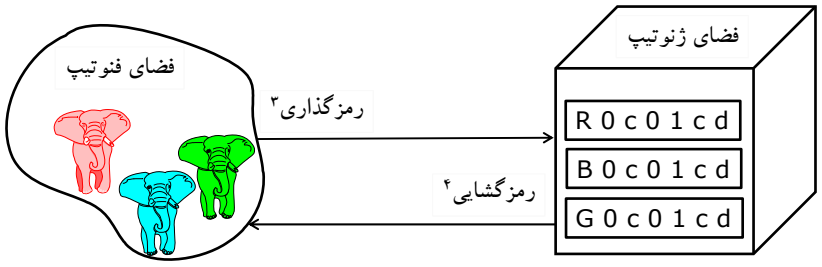
[Evaluation] Evaluate the fitness of each chromosome.

Until *StopCriteria* is met

[End] Return the best solution in final population.

روش نمایش جواب

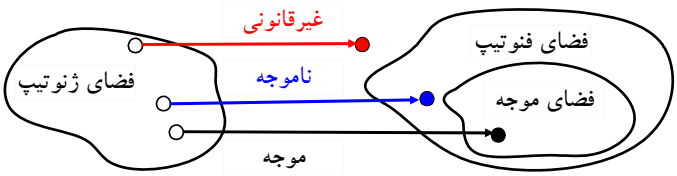
- عملگرهای ژنتیک روی فضای ژنوتیپ[1] (فضای رمز) عمل می‌کنند.
- از طرفی ارزیابی و انتخاب روی فضای فنوتیپ[2] (فضای جواب) انجام می‌شود.



1- Phenotype Space 2- Genotype Space 3- Encoding 4- Decoding

نکاتی درباره نگاشت بین فنوتیپ و ژنوتیپ

- موجه بودن[1] کروموزوم
  - کروموزوم رمزگشایی شده در ناحیه موجه واقع شده است یا نه؟
- قانونی بودن[2] کروموزوم
  - کروموزوم بیانگر یک جواب برای مسئله داده شده است یا نه؟
- منحصر بفرد بودن[3] کروموزوم
  - بین فضای فنوتیپ و فضای ژنوتیپ، رابطه یک–به–یک وجود دارد یا نه؟



1- Feasibility 2- Legality 3- Uniqueness

ناموجه بودن جواب

- ناموجه بودن[1] جواب ناشی از نقض محدودیت‌های مسئله بهینه‌سازی است.
  - روش‌های مختلفی برای لحاظ نمودن محدودیت‌های[2] مسئله بهینه‌سازی وجود دارد.
  - برای مسائل بهینه‌سازی با محدودیت‌های بفرم مساوی یا نامساوی می‌توان روش جریمه‌دهی[3] را برای جواب‌های ناموجه بکار گرفت.
  - معمولا با روش جریمه‌دهی، جواب‌ها به سمت مرز بین فضای موجه و ناموجه هدایت می‌شوند.

1- Infeasibility 2- Constraint Handling 3- Penalty method

غیرقانونی بودن جواب

- غیرقانونی بودن[1] جواب ناشی از طبیعت روش رمزگذاری است.
  - با توجه به اینکه روش‌های رمزگذاری وابسته به مسئله است، از طرفی عملگرهای ژنتیکی عمومی می‌باشند لذا در اکثر مواقع پس از بکارگیری عملگرها جواب از فرم قانونی خارج می‌شود.
  - در این موارد می‌توان با بکارگیری استراتژی تعمیر[2] جوابهای غیرقانونی را اصلاح کرد.

1- Illegality 2- Repairing Strategy

برخی ملاحظات دیگر در مورد روش نمایش جواب

- روش رمزگذاری روی عملکرد[1] الگوریتم تاثیرگذار است.
  - کیفیت جواب‌های[2] الگوریتم (اثربخشی)
    - وسعت ناحیه جستجو.
    - منطقی نمودن روش جستجو (متناسب‌سازی چگالی جستجو با هدف بهینه‌سازی).
  - میزان استفاده از منابع محاسباتی[3] (کارایی).
    - استفاده از روش نمایشی که محدودیت‌های مسئله بهینه‌سازی را لحاظ نماید.
    - استفاده از روش نمایشی که پس از بکارگیری عملگرهای ژنتیکی نیاز به تعمیر نداشته باشد.

1- Performance 2- Solution Quality 3- Computational Resource (Time, Memory, ...)

روشهای مختلف رمزگذاری

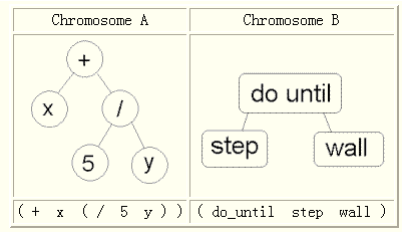
- روش رمزگذاری چند وضعیتی
  - دو وضعیتی[1] $\{0, 1\}$
  - سه وضعیتی[2] $\{0, 1, 2\}$
  - چهار وضعیتی[3] $\{0, 1, 2, 3\}$
  - ....
  - شانزده وضعیتی[4] $\{0, 1, ..., 9, A, B, ..., F\}$
- روش رمزگذاری چند وضعیتی متنی
  - $\{A, B\}$
  - $\{A, B, ..., Z\}$
  - $\{(Back), (Riht), (Forward), (Left)\}$

1- Binary 2- Ternary 3- Quarternary 2- Hexa-Decimal

- روش رمزگذاری با عدد صحیح (ترتیبی[۱])
- روش رمزگذاری با عدد حقیقی[۲]
- روش رمزگذاری درختی[۳]

$\{1,..., 9, 10, 11,12,...\}$

$\{1.12,..., 2.46, 3.01, 2.19,...\}$



1- Permutation 2- Real 3- Tree

مثال: مسئله فروشنده دوره‌گرد[۱]

- مدل ریاضی مسئله فروشنده دوره‌گرد با ۵ شهر.
  - ماتریس فاصله بین شهرها و متغیر تصمیم:

$$D=\begin{bmatrix} - & d_{12} & d_{13} & d_{14} & d_{15} \\ d_{21} & - & d_{23} & d_{24} & d_{25} \\ d_{31} & d_{32} & - & d_{34} & d_{35} \\ d_{41} & d_{42} & d_{43} & - & d_{45} \\ d_{51} & d_{52} & d_{53} & d_{54} & - \end{bmatrix}$$

$$x_{ij}=\begin{cases} 1 & \text{اگر مسیر } i \text{ به } j \text{ در تور باشد.} \\ 0 & \text{در غیراینصورت} \end{cases}$$

  - تابع هدف:

$$\min \; z = d_{12}x_{12}+d_{13}x_{13}+d_{14}x_{14}+d_{15}x_{15}+d_{21}x_{21}+d_{23}x_{23}+d_{24}x_{24}+d_{25}x_{25}+ d_{31}x_{31}+d_{32}x_{32}+d_{34}x_{34}+d_{35}x_{35}+d_{41}x_{41}+d_{42}x_{42}+d_{43}x_{43}+d_{45}x_{45}+ d_{51}x_{51}+d_{52}x_{52}+d_{53}x_{53}+d_{54}x_{54}$$

1- Travelling Salesman Problem (TSP)

مثال: مسئله فروشنده دوره‌گرد

- محدودیت‌های خروج از شهر دقیقا یکبار:

$$\begin{cases} x_{12}+x_{13}+x_{14}+x_{15}=1 \\ x_{21}+x_{23}+x_{24}+x_{25}=1 \\ x_{31}+x_{32}+x_{34}+x_{35}=1 \\ x_{41}+x_{42}+x_{43}+x_{45}=1 \\ x_{51}+x_{52}+x_{53}+x_{54}=1 \end{cases}$$

- محدودیت‌های ورود به هر شهر دقیقا یکبار:

$$\begin{cases} x_{21}+x_{31}+x_{41}+x_{51}=1 \\ x_{12}+x_{32}+x_{42}+x_{52}=1 \\ x_{13}+x_{23}+x_{43}+x_{53}=1 \\ x_{14}+x_{24}+x_{34}+x_{54}=1 \\ x_{15}+x_{25}+x_{35}+x_{45}=1 \end{cases}$$

مثال: مسئله فروشنده دوره‌گرد

- محدودیت‌های جلوگیری از ایجاد تورهای فرعی:

$$\begin{cases} x_{12}+x_{21}\le 1 \\ x_{13}+x_{31}\le 1 \\ x_{14}+x_{41}\le 1 \\ x_{15}+x_{51}\le 1 \\ x_{23}+x_{32}\le 1 \\ x_{24}+x_{42}\le 1 \\ x_{25}+x_{52}\le 1 \\ x_{34}+x_{43}\le 1 \\ x_{35}+x_{53}\le 1 \\ x_{45}+x_{54}\le 1 \end{cases}$$

- محدودیت متغیرهای تصمیم باینری:

$$\forall i,\ j;\ \ x_{ij}=0 \text{ or } 1$$

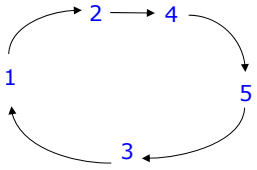
مثال: مسئله فروشنده دوره‌گرد

- روش اول رمزگذاری: باینری[1]
  - هر جواب با یک ماتریس رمزگذاری می‌شود.
  - به راحتی جواب از فرم موجه خارج می‌شود.

ژنوتیپ

	1	2	3	4	5
1	0	1	0	0	0
2	0	0	0	1	0
3	1	0	0	0	0
4	0	0	0	0	1
5	0	0	1	0	0

فنوتیپ



1- Binary

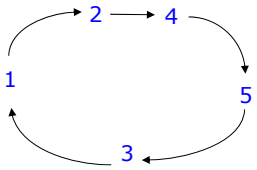
مثال: مسئله فروشنده دوره‌گرد

- روش دوم رمزگذاری: ترتیبی[1]
  - هر جواب با یک جایگشت از شماره شهرها رمزگذاری می‌شود.
  - به ازای هر جواب در فضای فنوتیپ، چند رمز ترتیبی متفاوت در فضای ژنوتیپ وجود دارد.

ژنوتیپ

1	2	4	5	3
2	4	5	3	1
4	5	3	1	2
5	3	1	2	4
3	1	2	4	5

فنوتیپ

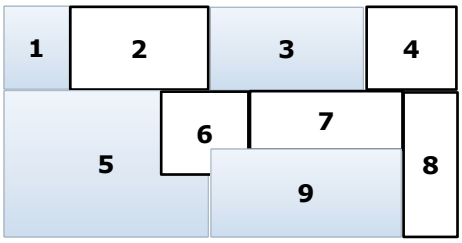


1- Permutation

مثال: مسئله مکانیابی ایستگاه‌های آتش‌نشانی

- مکانیابی ۴ ایستگاه آتش‌نشانی در ۹ منطقه.

فنوتیپ



ژنوتیپ

1	0	1	0	1	0	0	0	1
1	2	3	4	5	6	7	8	9

مثال: مسئله کوله‌پشتی[1]

- انتخاب اقلام از بین ۷ قلم.

فنوتیپ

ژنوتیپ

1	0	1	0	1	0	1
1	2	3	4	5	6	7

1- Knapsack Problem

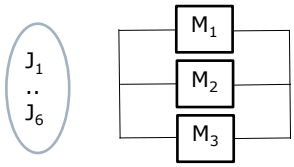
مثال: مسئله زمانبندی ماشین‌های موازی[1]

- روش اول رمزگذاری: ترتیبی
  - زمانبندی ۶ کار روی ۳ ماشین موازی و یکسان.
  - هر جواب با یک بردار از اعداد حقیقی رمزگذاری می‌شود.

ژنوتیپ

1	3	*	4	6	*	5	2

فنوتیپ



1- Parallel Machine Scheduling Problem

مثال: مسئله زمانبندی ماشین‌های موازی

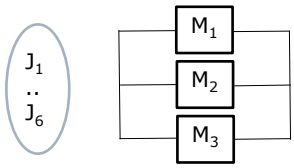
- روش دوم رمزگذاری: کلید تصادفی[1]
  - هر جواب با یک بردار از اعداد حقیقی رمزگذاری می‌شود.

ژنوتیپ

1.34	3.05	2.79	2.83	1.11	3.92
1	2	3	4	5	6

$$R \sim U[1, m+1)$$

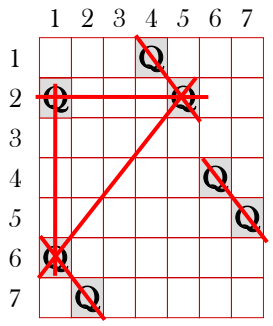
فنوتیپ



1- Random Key

مثال: مسئله n وزیر[۱]

- میخواهیم n وزیر را در یک صفحه شطرنج $n \times n$ بگونه‌ای بچینیم که هیچ دو وزیری یکدیگر را نزنند.
- نمونه: نمایش در فضای فنوتیپ برای ۷ وزیر



۶ = تعداد برخوردها

1- n-Queen

مثال: مسئله n وزیر

- روش اول رمزگذاری: برداری از درآیه‌های دو بعدی (x_i , y_i).
  - نمایش موقعیت هر وزیر در صفحه شطرنج با زوج مرتب (x_i , y_i).
  - عیب: ممکن است که در یک خانه بیش از یک وزیر قرار گیرد.
  - تعداد حالت‌های حاصل از این روش نمایش $(n \times n)^n$.

ژنوتیپ

(x_1 , y_1)	(x_2 , y_2)	(x_i , y_i)	(x_n , y_n)

مثال: مسئله n وزیر

- روش دوم رمزگذاری: باینری
  - انتخاب n خانه از صفحه شطرنج و قرار دادن یک وزیر در هر خانه.
  - مزیت: در هر خانه حداکثر یک وزیر قرار می‌گیرد.
  - تعداد حالتهای حاصل از این روش نمایش $\binom{n\times n}{n}$.

ژنوتیپ

	1	2	3	4	5	6	7
1	0	0	0	1	1	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	1	0
4	0	0	1	0	0	0	0
5	0	0	0	0	0	0	1
6	1	0	0	0	0	0	0
7	0	1	0	0	0	0	0

مثال: مسئله n وزیر

- روش سوم رمزگذاری: ترتیبی.
- مسئله n وزیر را می‌توان بصورت یک مسئله ترتیبی نمایش داد.
- فرض کنید که وزیرها توسط حرف i شماره‌گذاری شده باشند.
- همچنین فرض کنید که وزیر i در ردیف i قرار گرفته باشد.
- فرض کنید شماره ستونی که وزیر ردیف i در آن قرار گرفته را با $\pi(i)$ نمایش دهیم. آنگاه یک جواب بصورت $\Pi = \{\pi(1),\pi(2),...\pi(n)\}$ قابل نمایش است.

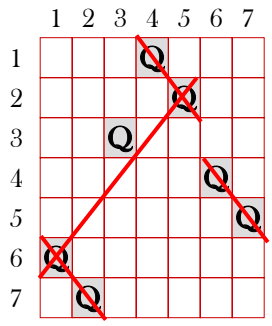
مثال: مسئله n وزیر

ژنوتیپ

4	5	3	6	7	1	2

$n!$

فنوتیپ



۴ = تعداد برخوردها

مثال: مسئله n وزیر

- مقایسه اندازه فضای جستجو با توجه به روش نمایش جواب

$$\frac{(10\times10)^{10}}{\binom{10\times10}{10}} = 5,776,904 \qquad \frac{(11\times11)^{11}}{\binom{11\times11}{11}} = 63,757,785$$

$$\frac{\binom{10\times10}{10}}{10!} = 4,770,257 \qquad \frac{\binom{11\times11}{11}}{11!} = 31,985,279$$

بهتر است روش نمایش جواب به گونه‌ای باشد که:

- حجم حافظه‌ی کمتری را اشغال نماید.
- بررسی شرایط موجه بودن جواب ساده باشد.
- همه فضای موجه به صورت یکنواخت بررسی نشوند.
- فقط زیر مجموعه‌ای از فضای جواب جستجو شوند (مجموعه غالب).
- با انجام عملیات تقاطع و جهش از فضای موجه خارج نشویم.

تابع برازندگی[1]

- گاها به آن تابع ارزیابی[2] نیز گفته می‌شود.
- میزان مطلوبیت یک جواب/کروموزوم را بر اساس این تابع محاسبه می‌کنند.
- در مسائل بیشینه‌سازی، مقدار برازندگی متناسب با مقدار تابع هدف است.

$$fit(i) \propto of(i) \Rightarrow fit(i) = k.of(i)$$

- در مسائل کمینه‌سازی:
  - مقدار برازندگی متناسب با عکس مقدار تابع هدف است.

$$fit(i) \propto \frac{1}{of(i)} \Rightarrow fit(i) = \frac{k}{of(i)}$$

  - مقدار برازندگی متناسب با هزینه‌ی فرصت از دست رفته است.

$$fit(i) \propto \left(M - of(i)\right) \Rightarrow fit(i) = k.\left(M - of(i)\right)$$

1- Fitness Function 2- Evaluation Function

استراتژی‌های انتخاب[۱]

- انتخاب بر اساس چرخ رولت[۲].
- انتخاب کاملا تصادفی[۳].
- انتخاب بر اساس بهترین‌ها[۴].
- انتخاب بر اساس مسابقه[۵].
- انتخاب بر اساس رتبه[۶].
- انتخاب بر اساس حذف درصدی از بدترین اعضاء[۷](انتخاب پایدار[۸]).
- انتخاب بر اساس تابع احتمال بولتزمان[۹].

1- Selection Strategy 2- Roulette Wheel 3- Purely Random 4- Elite 5- Tournament

انتخاب بر اساس چرخ رولت

- ایده‌ی اصلی: تخصیص شانس انتخاب برای هر جواب، متناسب با میزان برازندگی آن جواب

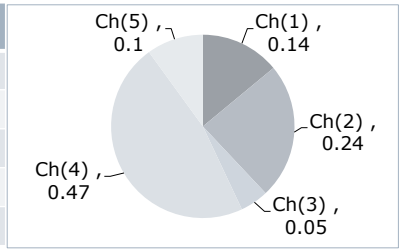
$$p(i)=\frac{fit(i)}{\sum_{i=1}^{popsize} fit(i)}$$

- روش:
  - محاسبه‌ی مقدار برازندگی هر جواب $[fit(1), fit(2),..., fit(popsize)]$
  - محاسبه‌ی احتمال انتخاب هر جواب $[p(1),\ p(2), ...,\ p(popsize)]$
  - محاسبه‌ی احتمال تجمعی برای هر جواب $[cp(1), cp(2),..., cp(popsize)=1]$
  - تولید یک عدد تصادفی از توزیع یکنواخت $R \sim U[0, 1]$
  - انتخاب جواب براساس قاعده‌ی زیر:

$$R \le cp(1) \Rightarrow Chrom(1),\quad ..... ,\ cp(i-1) < R \le cp(i) \Rightarrow Chrom(i)$$

انتخاب بر اساس چرخ رولت

Population	Fitness	Probability	Cumulative Probability
Ch(1)	15	0.14	0.14
Ch(2)	27	0.24	0.38
Ch(3)	6	0.05	0.43
Ch(4)	52	0.47	0.90
Ch(5)	11	0.10	1.00



$$R = 0.13 \le cp(1) = 0.14 \Rightarrow Ch(1)$$
$$cp(3) = 0.43 < R = 0.71 \le cp(4) = 0.90 \Rightarrow Ch(4)$$

ایجاد انعطاف درتابع احتمال

- می‌توان با تغییرات جزئی در تابع احتمال، مکانیزم‌های تمرکزدهی/ تنوع‌بخشی را تقویت یا تضعیف نمود.

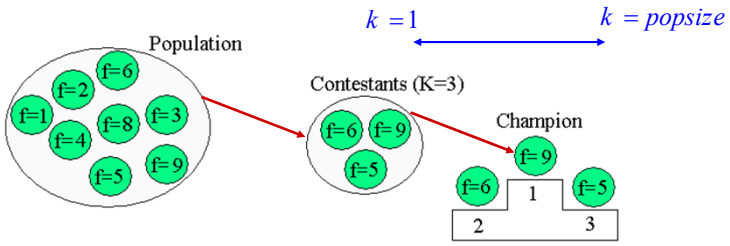
$$p(i) = \frac{(fit(i))^k}{\sum_{i=1}^{popsize} (fit(i))^k}$$

$$k = 0 \longleftrightarrow k \to +\infty$$

Population	Fitness	Probability k=1	Probability k=2	Probability k=0.5
Ch(1)	15	0.14	0.06	0.18
Ch(2)	27	0.24	0.19	0.24
Ch(3)	6	0.05	0.01	0.11
Ch(4)	52	0.47	0.71	0.33
Ch(5)	11	0.10	0.03	0.15

انتخاب بر اساس مسابقه

- خلاصه روش:
  - انتخاب k (اندازه‌ی تورنمنت[1]) $1 \le k \le popsize$
  - انتخاب k کروموزوم بصورت تصادفی و بدون جایگذاری از جمعیت.
  - انتخاب کروموزم نخبه از بین k کروموزوم انتخاب شده در مرحله‌ی قبل.



1- Tournament Size

انتخاب بر اساس رتبه

- خلاصه روش:
  - تخصیص رتبه به هر جواب بر اساس ترتیب نزولی تابع برازندگی.
  - فاصله‌ی بین برازندگی جواب‌ها اهمیت نخواهد داشت.

Population	Fitness	Rank	Probability
Ch(1)	15	3	0.20
Ch(2)	27	2	0.27
Ch(3)	6	5	0.07
Ch(4)	52	1	0.33
Ch(5)	11	4	0.13

$$p(i) = \frac{Max\left(Rank(i)\right) - Rank(i) + 1}{\sum_{i=1}^{Popsize} Rank(i)}$$

عملگرهای ژنتیکی

- عملگر جهش:
  - عملگر جهش برای ایجاد تنوع در فضای جستجو است، لذا درصد این عملگر باید کم باشد (توصیه $P_m < 0.10$).
  - عملگر جهش مبتنی بر یک کروموزوم است.
  - عملگر جهش نباید با هدف بهبود طراحی شود، به همین دلیل ساختاری تصادفی دارد.
  - عملگر جهش وابسته به روش نمایش جواب می‌باشد (روشهای نمایش مختلف، عملگرهای جهش متفاوتی دارند).
  - عملگر جهش باید به یک کروموزوم معتبر (موجه و قانونی) منجر شود.

مثال: سه عملگر جهش برای مسئله TSP

- Inversion

2	1	4	3	6	5	9	8	7
2	1	5	6	3	4	9	8	7

- Swap

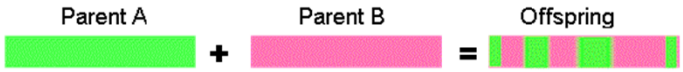
2	1	4	3	6	5	9	8	7
8	1	4	3	6	5	9	2	7

- Displacement

2	1	4	3	6	5	9	8	7
2	1	4	3	6	7	9	5	8

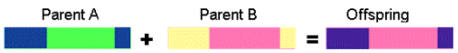
عملگرهای ژنتیکی

- عملگر تقاطع:
  - درصد عملگر تقاطع باید زیاد باشد (توصیه $P_c > 0.85$).
  - عملگر تقاطع مبتنی بر دو کروموزوم است.
  - عملگر تقاطع وابسته به روش نمایش جواب می‌باشد (روشهای نمایش مختلف، عملگرهای تقاطع متفاوتی دارند).
  - عملگر تقاطع باید به یک کروموزوم معتبر (موجه و قانونی) منجر شود.



مثال: عملگر تقاطع PMX برای مسئله TSP

- Partially Mapped (PMX)



1	2	3	4	5	6	7	8	9
4	5	2	1	8	7	6	9	3

x	x	x	4	5	6	7	x	x
x	x	x	1	8	7	6	x	x

x	x	2	4	5	6	7	9	3
x	2	3	1	8	7	6	x	9

1	8	2	4	5	6	7	9	3
4	2	3	1	8	7	6	5	9

مثال: عملگر تقاطع OX برای مسئله TSP

- Order (OX)

1	2	3	4	5	6	7	8	9
4	5	2	1	8	7	6	9	3

2	1	8	4	5	6	7	9	3
2	3	4	1	8	7	6	5	9

مثال: عملگر تقاطع Clever برای مسئله TSP

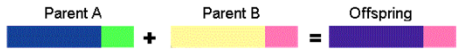
- Clever

1	2	3	4	5	6	7	8	9
4	5	2	1	8	7	6	9	3

4	5	6	7	2	1	8	9	3
1	8	7	6	2	3	4	5	9

مثال: عملگر تقاطع One-Point برای مسئله TSP (نمایش: کلید تصادفی)

- One-Point



1	2	3	4	5	6						
0.91	0.23	0.84	0.67	0.05	0.42	5	2	6	4	3	1
0.31	0.82	0.36	0.07	0.45	0.69	4	1	3	5	6	2
1	2	3	4	5	6						

1	2	3	4	5	6						
0.91	0.23	0.84	0.07	0.45	0.69	4	2	5	6	3	1
0.31	0.82	0.36	0.67	0.05	0.42	5	1	3	6	4	2
1	2	3	4	5	6						

شرط توقف[1] GA

- رسیدن به یک مقدار تلاش محاسباتی خاص.
  - مقایسه تعداد نسل‌های تکامل یافته با حداکثر مقدار از پیش تعیین شده (*Maxgen*).
  - مقایسه زمان محاسباتی با حداکثر مقدار از پیش تعیین شده (*MaxCompt*).
- بهبود ناچیز یا عدم بهبود در تابع هدف[2].
  - مقایسه درصد بهبود میانگین برازندگی جواب‌های دو نسل متوالی با یک مقدار از پیش تعیین شده.

$$\Delta = \frac{\bar{f}_n - \bar{f}_{n-1}}{\bar{f}_n} \le \Delta_{\min}$$

  - مقایسه نسبت برازندگی بهترین جواب به میانگین برازندگی جواب‌های یک نسل با یک مقدار از پیش تعیین شده.

$$\frac{f_{best}}{\bar{f}} \to 1^+$$

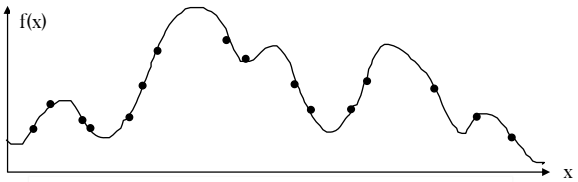
1- Stopping Criteria (Termination Condition) 2- Stagnation

شرط توقف GA

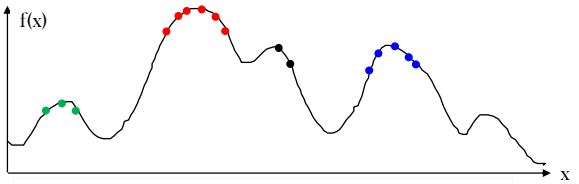
- مقایسه واریانس برازندگی جواب‌های یک نسل با یک مقدار از پیش تعیین شده.
- مقایسه تعداد نسل‌هایی که بهترین جواب ثابت مانده با یک مقدار از پیش تعیین شده.

☐ رسیدن به یک جواب قانع کننده از لحاظ تابع هدف.

توزیع جواب‌ها در فرآیند تکاملی الگوریتم ژنتیک

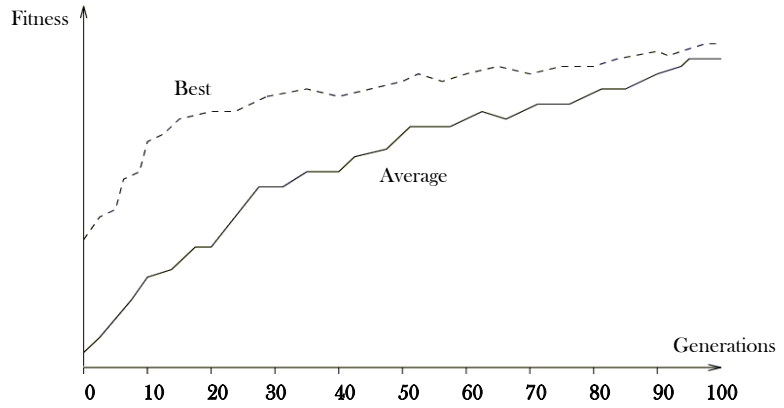


توزیع جوابها در فضای جستجو در نسلهای اولیه



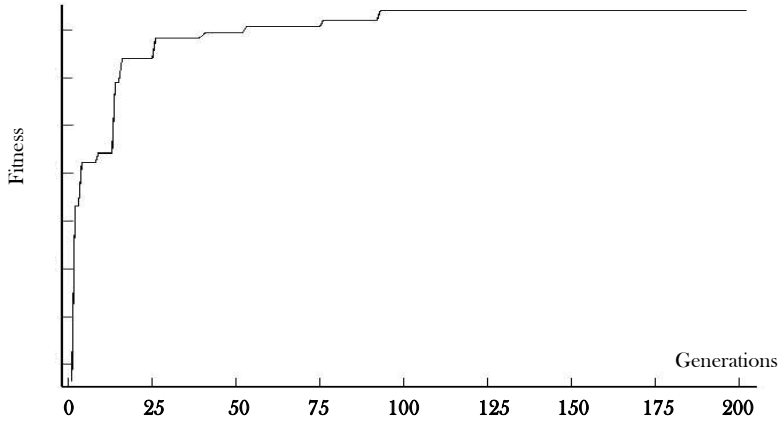
توزیع جوابها در فضای جستجو در نسلهای پایانی

نمودار همگرایی[1]

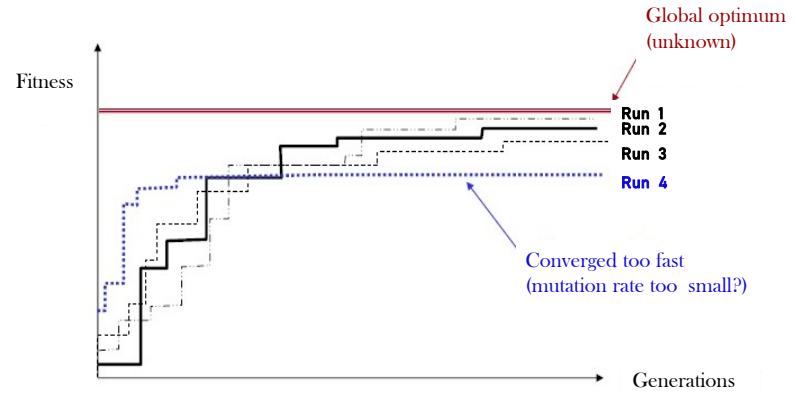


1- Convergence Diagram

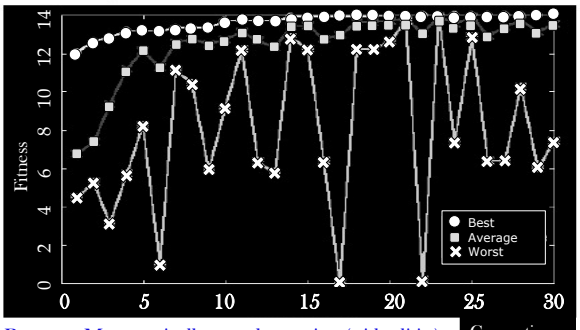
نمودار همگرایی



نمودار همگرایی



نمودار همگرایی



Best: Monotonically non-decreasing (with elitist)
Average: Improves over time
Ratio: Best/Average is a measure of convergence
Worst: Randomly changing

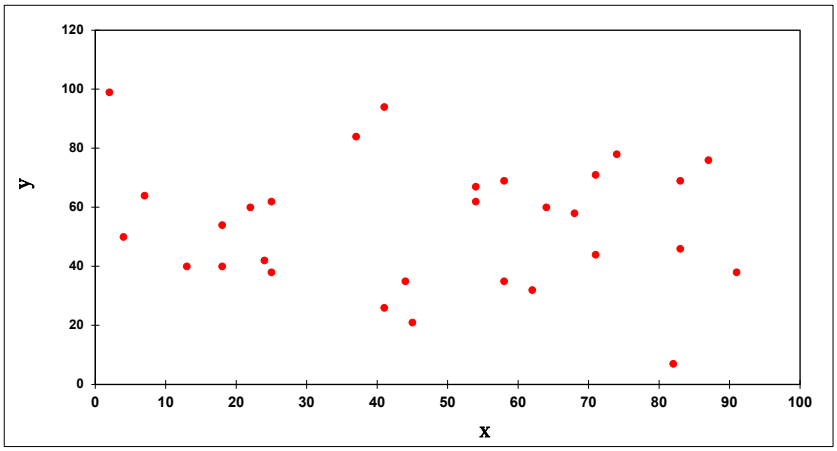
مثال: داده‌های مربوط به مسئله TSP با n=30

No.	x	y
1	82	7
2	91	38
3	83	46
4	71	44
5	64	60
6	68	58
7	83	69
8	87	76
9	74	78
10	71	71

No.	x	y
11	58	69
12	54	62
13	54	67
14	37	84
15	41	94
16	2	99
17	7	64
18	22	60
19	25	62
20	18	54

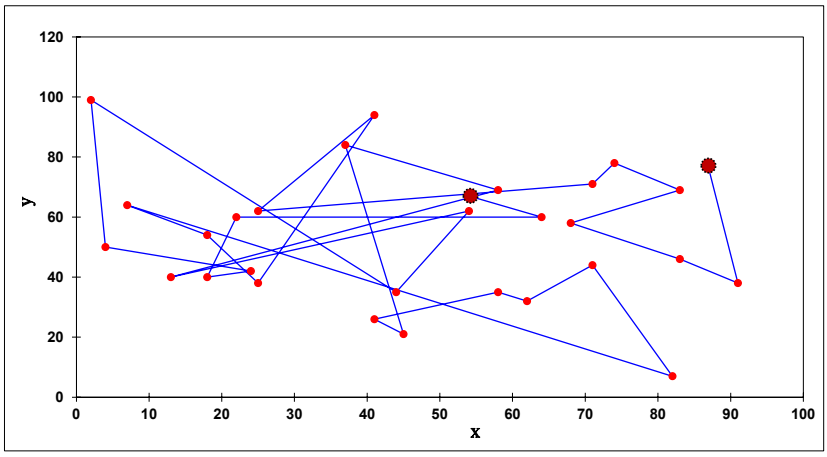
No.	x	y
21	4	50
22	13	40
23	18	40
24	24	42
25	25	38
26	41	26
27	45	21
28	44	35
29	58	35
30	62	32

موقعیت شهرهای مسئله TSP با n=30



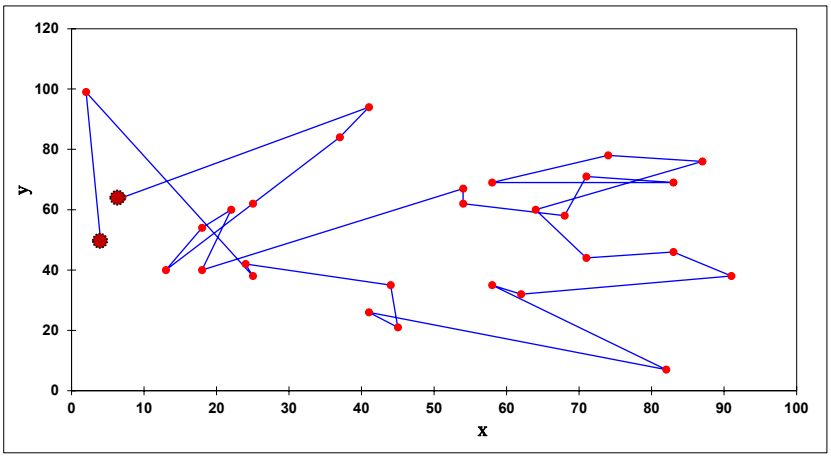
یک جواب موجه مسئله

Distance=941



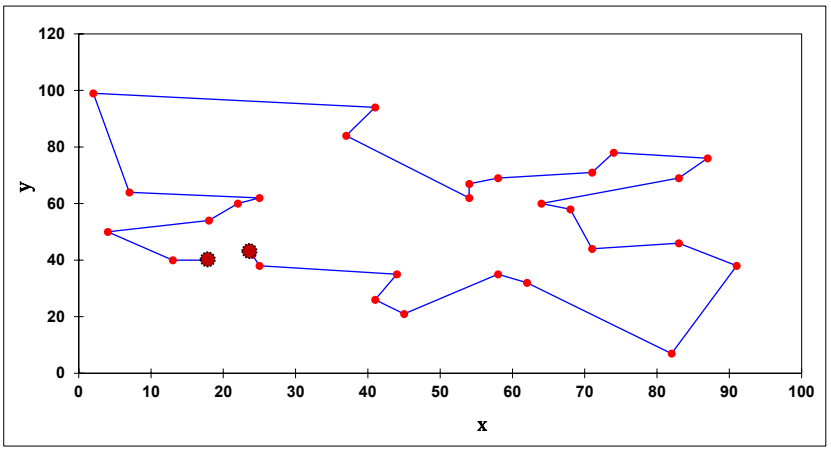
یک جواب موجه مسئله

Distance=652

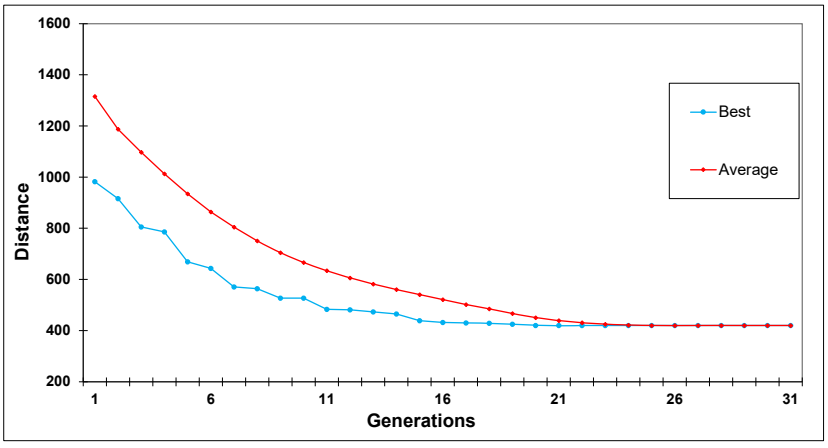


جواب بهینه مسئله

Distance=420



نمودار همگرایی الگوریتم ژنتیک در مسئله TSP

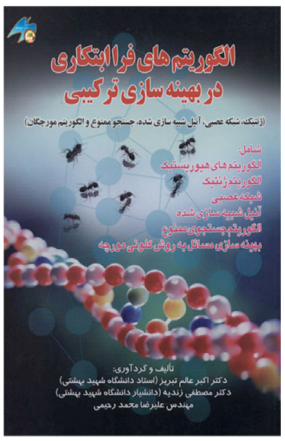


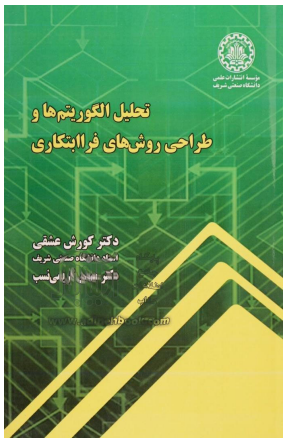
منابع

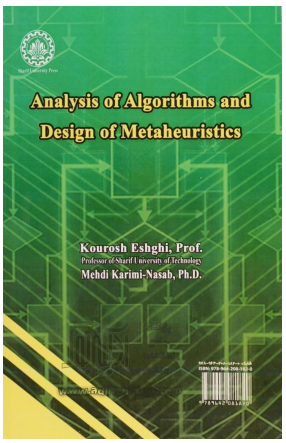
- Talbi, El-Ghazali, Metaheuristics: *From Design to Implementation*, John Wiley & Sons, Inc, 2009.
- Gen, Mitsuo and Cheng, Runwei Cheng, *Genetic Algorithms and Engineering Optimization*, John Wiley & Sons, New York, 2000.
- Yang, Xin-She, *Engineering Optimization: An Introduction with Metaheuristic Applications*, John Wiley & Sons, 2010.
- عالم‌تبریز، اکبر، زندیه، مصطفی و محمدرحیمی، علیرضا، *الگوریتم‌های فراابتکاری در بهینه‌سازی ترکیبی (الگوریتم ژنتیک، شبیه‌سازی تبرید، جستجوی ممنوع، اجتماع مورچگان و شبکه عصبی)*، انتشارات صفار، اشراقی، چاپ ۱۳۸۹.
- عشقی، کوروش و کریمی نسب، مهدی، *تحلیل الگوریتم‌ها و طراحی روش‌های فراابتکاری*، – موسسه انتشارات علمی دانشگاه صنعتی شریف، چاپ ۱۳۹۵.

منابع

منابع


الگوریتم های فرا ابتکاری
در بهینه سازی ترکیبی
(ژنتیک، شبکه عصبی، آنیل شبیه سازی شده، جستجو ممنوع و الگوریتم مورچگان)
شامل
الگوریتم های هیوریستیک
الگوریتم ژنتیک
شبکه عصبی
آنیل شبیه سازی شده
الگوریتم جستجوی ممنوع
بهینه سازی مسائل به روش تکنونی متروپه
تالیف و گردآوری
دکتر اکبر عالم تبریز (استاد دانشگاه شهید بهشتی)
دکتر مصطفی زندیه (دانشیار دانشگاه شهید بهشتی)
مهندس علیرضا محمد رحیمی


تحلیل الگوریتم‌ها و
طراحی روش‌های فراابتکاری
دکتر کورش عشقی


Analysis of Algorithms and
Design of Metaheuristics
Kourosh Eshghi, Prof.
Mehdi Karimi-Nasab, Ph.D.