الگوریتم ژنتیک

Genetic Algorithm

Dr. Mostafa Zandieh
Shahid Beheshti University

PCPARSI.COM

PCPARSI.COM

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 می کنند.
  - در اغلب موارد افراد نسل جدید، سازگاری بیشتری با طبیعت دارند.
  - در موارد استثنایی ممکن جهشهایی[2] در خصوصیات یک فرد رخ می دهد.
    - اغلب افراد جهش یافته با طبیعت ناسازگارند.
    - در موارد نادر سازگاری بسیار بالایی دارند.

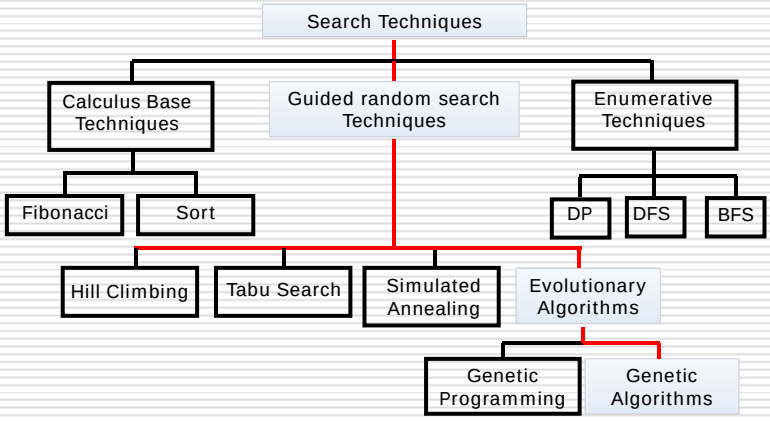
1- Darwin's Evolutionary Theory 2- Mutation

پیشینه GA

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

1- Biological Evolutionary Process 2- Guided Random Search Technique

طبقه بندی روشهای جستجو



اجزای یک الگوریتم ژنتیک

- روشی برای نمایش جواب[1]
  - کدگذاری جوابها در قالب یک رشته (بردار/ ماتریس) به عنوان کروموزوم[2] شناخته می شود.
  - مجموعه ای از این کروموزومها در هر نسل الگوریتم را یک جمعیت می نامند.
- تابعی برای ارزیابی جوابها
  - میزان سازگاری کروموزومها با محیط توسط تابع برازندگی[3] ارزیابی می شود.
  - تابع برازندگی را با توجه به تابع هدف مساله بهینه سازی محاسبه می کنند.
- استراتژی انتخاب[4]
  - در هر نسل به کروموزومهای برتر، باید شانس بیشتری برای زاد و ولد داده شود.

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

اجزای یک الگوریتم ژنتیک

- استراتژی انتخاب مناسب می تواند تمرکزدهی[1] و تنوع بخشی[2] را در هر نسل از جوابها تامین کند.
- عملگرهای ژنتیکی
  - فرآیند زاد و ولد و ایجاد نسل جدید جوابها با استفاده از عملگرهای ژنتیکی سه گانه (تقاطع[3]، جهش[4] و کپی) انجام می شود.
- تنظیم پارامتر[5]
  - عملکرد الگوریتم ژنتیک، بستگی به ترکیب مقادیر پارامترهای الگوریتم دارد.
  - ترکیب مقادیر پارامترها روی پایداری الگوریتم، کیفیت جواب و زمان حل تاثیرگذار است.
  - پارامترهای الگوریتم را با استفاده از روشهای طراحی آزمایشات تنظیم می کنند.

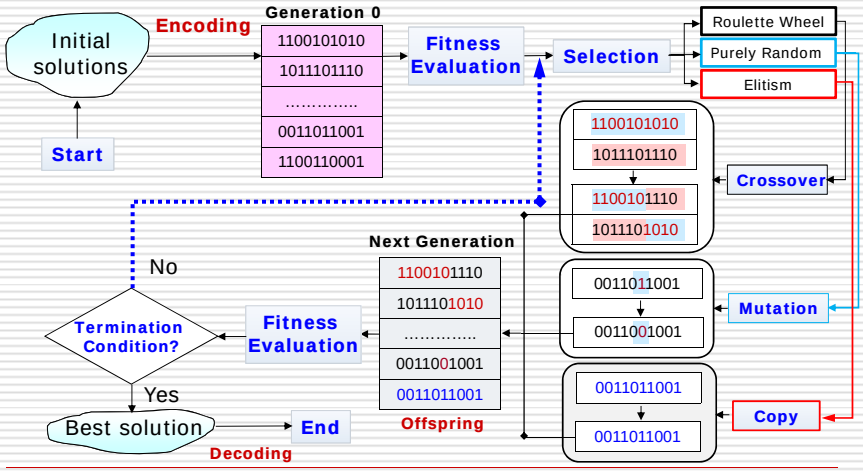
1- Intensification 2- Diversification 3- Crossover 4- Mutation 5- Parameter Tuning

الفاظ معادل[1]

طبیعت	الگوریتم ژنتیک
محیط	مساله بهینه سازی
افرادی که درمحیط زندگی می کنند	جوابهای موجه
درجه سازگاری هر فرد با محیط اطراف	کیفیت/ برازندگی جواب
جمعیتی از افراد	مجموعه ای 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 مجموعه ای از جوابهای موجه

1- Metaphor

ساختار عمومی الگوریتم ژنتیک



شبه کد 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]

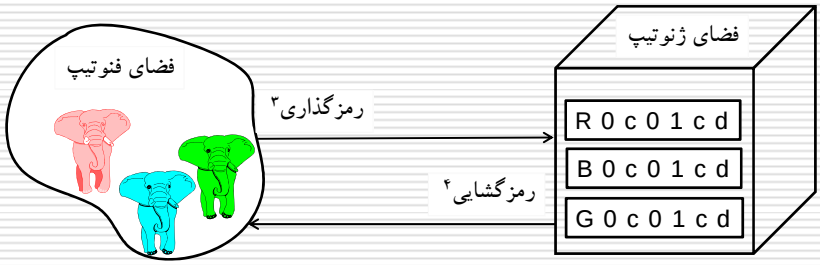
Repeat
- **[Selection]** Select Parents based on selection strategy.
- **[Crossover]** Produce (*PopSize* * *Pc*) of offspring with Crossover.
- **[Mutation]** Produce (*PopSize* * *Pm*) of offspring with Mutation.
- **[Reproduction]** Copy remaining chromosomes based on elitism.
- **[Replacing]** Place 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

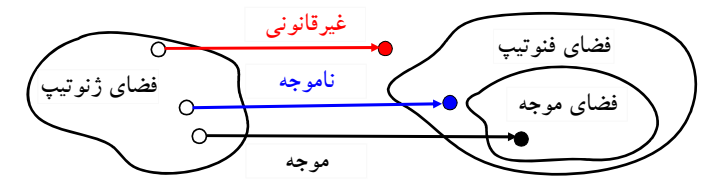
- عملگرهای ژنتیک روی فضای ژنوتیپ[1] (فضای رمز یا کروموزوم) عمل می کنند.
- از طرفی ارزیابی و انتخاب روی فضای فنوتیپ[2] (فضای جواب) انجام می شود.



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 مساله 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 مساله بهینه سازی است.
  - روشهای مختلفی برای لحاظ نمودن محدودیتهای[2] مساله بهینه سازی وجود دارد.
  - برای مسائل بهینه سازی با محدودیتهای بفرم مساوی یا نامساوی می توان روش جریمه دهی[3] را برای جوابهای ناموجه بکار گرفت.
  - معمولا با روش جریمه دهی، جوابها به سمت مرز بین فضای موجه و ناموجه هدایت می شوند.

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 مساله است، از طرفی عملگرهای ژنتیکی عمومی میباشند لذا در اکثر مواقع پس از بکارگیری عملگرها جواب از فرم قانونی خارج می شود.
  - در این موارد می توان 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 الگوریتم (اثربخشی)
    - وسعت ناحیه جستجو.
    - منطقی نمودن روش جستجو (متناسب سازی چگالی جستجو با هدف بهینه سازی).
  - کاهش زمان محاسباتی (کارایی).
    - استفاده از روشی که محدودیتها را لحاظ نماید.
    - استفاده از روشی که پس از بکارگیری عملگرهای ژنتیکی نیاز به تعمیر نداشته باشد.

1- Performance

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] $\{0, 1, \ldots, 9, A, B, \ldots, F\}$
- روش رمزگذاری چند وضعیتی متنی

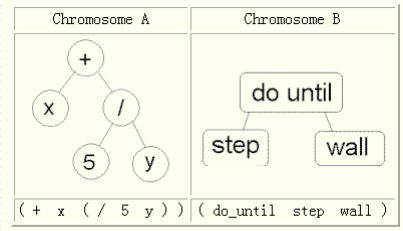
$\{A, B\}$

$\{A, B, \ldots, Z\}$

$\{(Back), (Riht), (Forward), (Left)\}$

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

- روش رمزگذاری با عدد صحیح (ترتیبی[1]) $\{1, \ldots, 9, 10, 11, 12, \ldots\}$
- روش رمزگذاری با عدد حقیقی[2] $\{1.12, \ldots, 2.46, 3.01, 2.19, \ldots\}$
- روش رمزگذاری درختی[3]



1- Permutation 2- Real 3- Tree

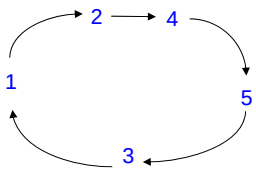
مثال: مساله فروشنده دوره گرد[1]

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

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

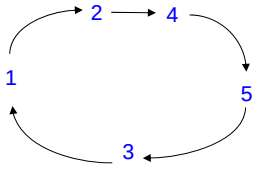
مثال: مساله فروشنده دوره گرد

- روش دوم: ترتیبی[1]
  - هر جواب با یک جایگشت از شماره شهرها رمزگذاری می شود.
  - 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



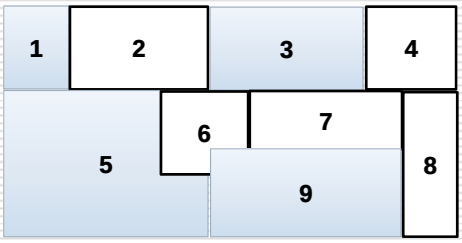
1- Permutation

مثال: مساله مکانیابی ایستگاههای آتش نشانی

ژنوتیپ

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

فنوتیپ



مثال: مساله کوله پشتی[1]

ژنوتیپ

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

فنوتیپ

1- Knapsack Problem

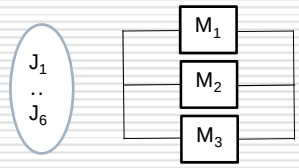
مثال: مساله زمانبندی ماشینهای موازی[1]

- روش اول: ترتیبی
  - هر جواب با یک بردار از اعداد حقیقی رمزگذاری می شود.

ژنوتیپ

1	3	*	4	6	*	5	2

فنوتیپ



1- Parallel Scheduling Problem

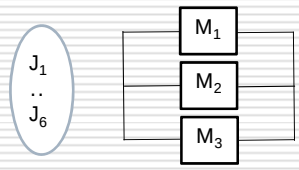
مثال: مساله زمانبندی ماشینهای موازی

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

ژنوتیپ

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

فنوتیپ



1- Random Key

مثال : مساله n وزیر[1]

- میخواهیم n وزیر را در یک صفحه شطرنج $n \times n$ بگونه ای بچینیم که هیچ دو وزیری نتوانند یکدیگر را بزنند.
- مساله n وزیر را می توان بصورت یک مساله جایگشتی (Permutation) نمایش داد.
- فرض کنید که وزیرها توسط حرف i شماره گذاری 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.

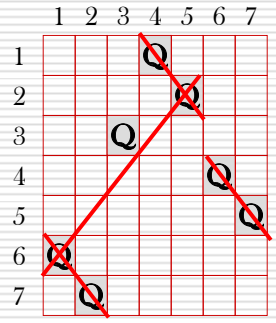
1- n-Queen

مثال : مساله n وزیر

ژنوتیپ

4	5	3	6	7	1	2

فنوتیپ



تعداد برخوردها = 4

تابع برازندگی[1]

- گاها به آن تابع ارزیابی[2] نیز گفته می شود.
- میزان مطلوبیت یک جواب/ کروکوزوم را بر اساس این تابع محاسبه می کنند.
- در مسائل بیشینه سازی، مقدار برازندگی متناسب با مقدار تابع هدف است.

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

- در مسائل کمینه سازی:
  - 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 هزینه ی 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[1]

- انتخاب بر اساس چرخ رولت[2].
- انتخاب کاملا تصادفی[3].
- انتخاب بر اساس بهترینها[4].
- انتخاب بر اساس مسابقه[5].
- انتخاب بر اساس تابع احتمال بولتزمان[6].
- انتخاب بر اساس حذف درصدی از بدترین اعضاء[7] (انتخاب پایدار[8]).
- انتخاب بر اساس رتبه[9].

1- Selection Strategy 2- Roulette Wheel 3- Purely Random 4- Elite 5- Tournament
6- Boltzmann 7- Truncation 8- Steady State 9- Ranked-Based

انتخاب بر اساس چرخ رولت

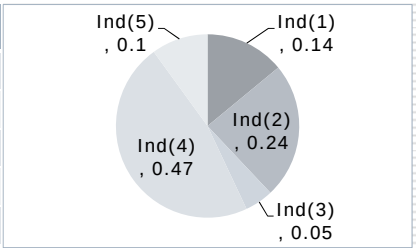
- ایده ی اصلی: تخصیص شانس انتخاب برای هر جواب، متناسب با میزان برازندگی آن جواب

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

- روش:
  - محاسبه ی مقدار برازندگی هر جواب $[fit(1), fit(2), ..., fit(popsize)]$
  - محاسبه ی احتمال انتخاب هر جواب $[p(1), p(2), ..., p(popsize)]$
  - محاسبه ی 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 قاعده ی زیر:

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

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



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

Mutation

- Inversion

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

- Swap

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

- Displacement

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

Crossover

- Partially Mapped (PMX)

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

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

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

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

Crossover

- 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

Crossover

- 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

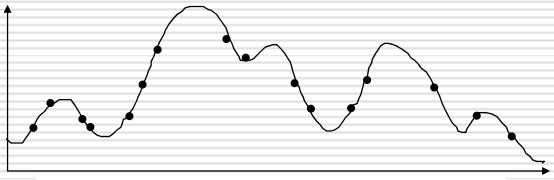
شرط توقف[1] GA

- رسیدن به یک مقدار تلاش محاسباتی خاص.
  - حداکثر مقدار از پیش تعیین شده ای برای تعداد نسلها (*Maxgen*).
  - حداکثر مقدار از پیش تعیین شده ای برای زمان محاسباتی (*MaxCompt*).
- بهبود ناچیز یا عدم بهبود در تابع هدف[2].
  - مقایسه تعداد نسلهایی که بهترین جواب ثابت مانده با یک مقدار از پیش تعیین شده.
  - مقایسه تفاوت میانگین برازندگی جوابهای دو نسل متوالی با یک مقدار از پیش تعیین شده.
  - مقایسه واریانس برازندگی جوابهای یک نسل با یک مقدار از پیش تعیین شده.
  - محاسبه نسبت برازندگی بهترین جواب به میانگین برازندگی جوابهای یک نسل و مقایسه با یک مقدار آستانه از قبل تعیین شده.

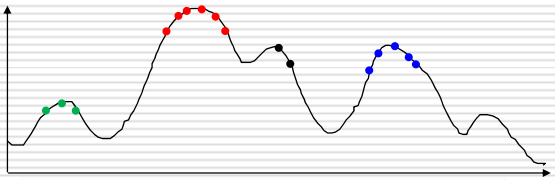
1- Stopping Criteria (Termination Condition) 2- Stagnation

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

توزیع جوابها در فرآیند تکاملی الگوریتم ژنتیک

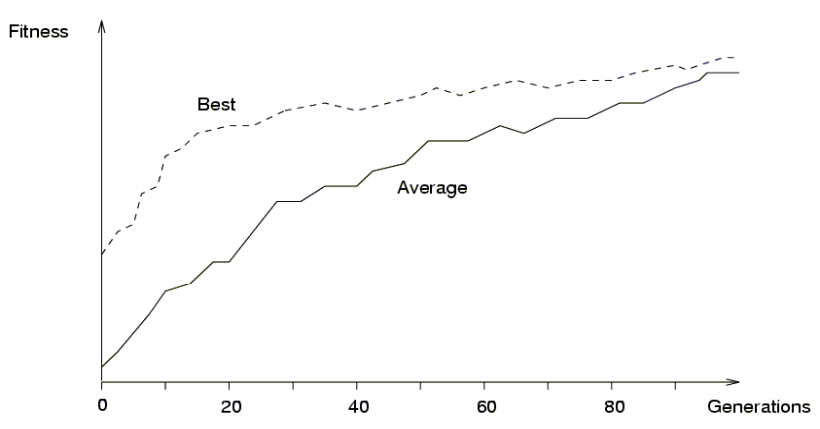


توزیع جوابها در فضای جستجو در نسلهای اولیه

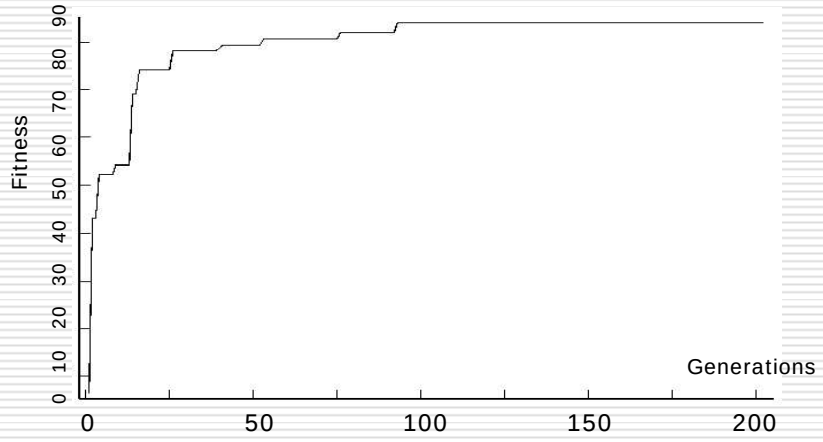


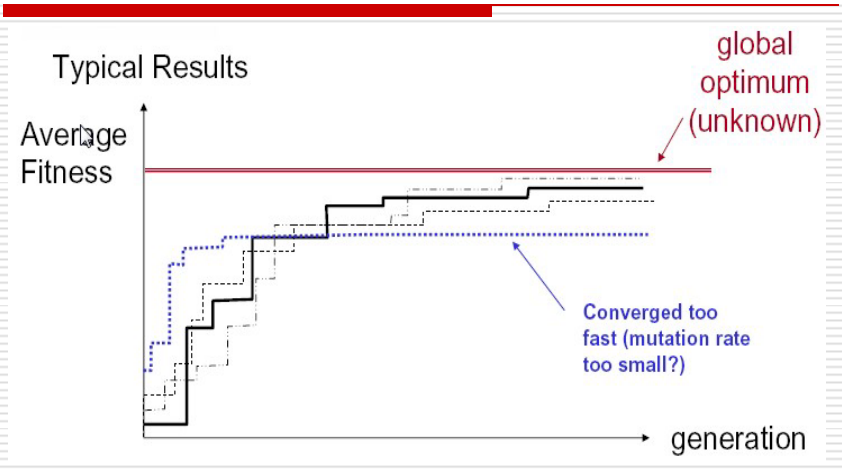
توزیع جوابها در فضای جستجو در نسلهای پایانی

نمودار همگرایی[1]

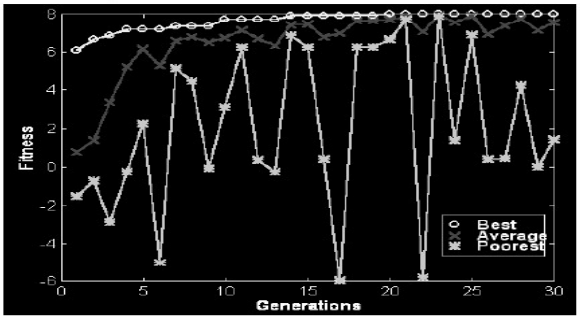


1- Convergence Diagram


Fitness
0
10
20
30
40
50
60
70
80
90
Generations
0
50
100
150
200


Typical Results
Average Fitness
global optimum (unknown)
Converged too fast (mutation rate too small?)
generation

Performance profile

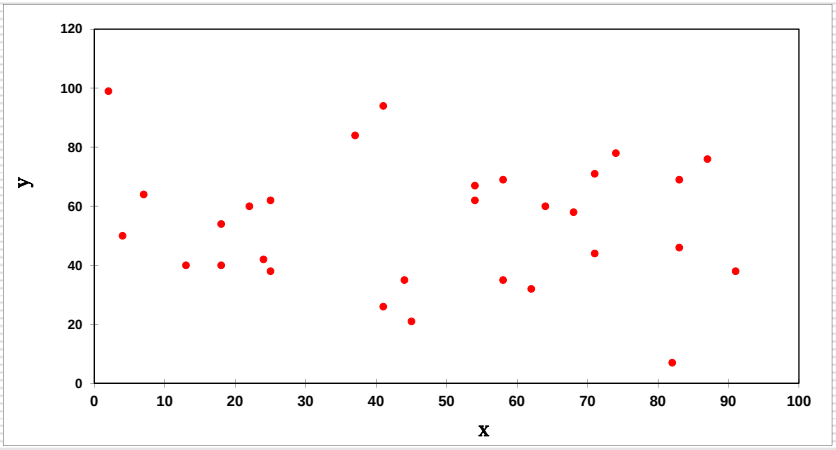


Best: Monotonically non-decreasing (with elitist)
Average: Improves over time
Ratio: Best/Average is a measure of convergence

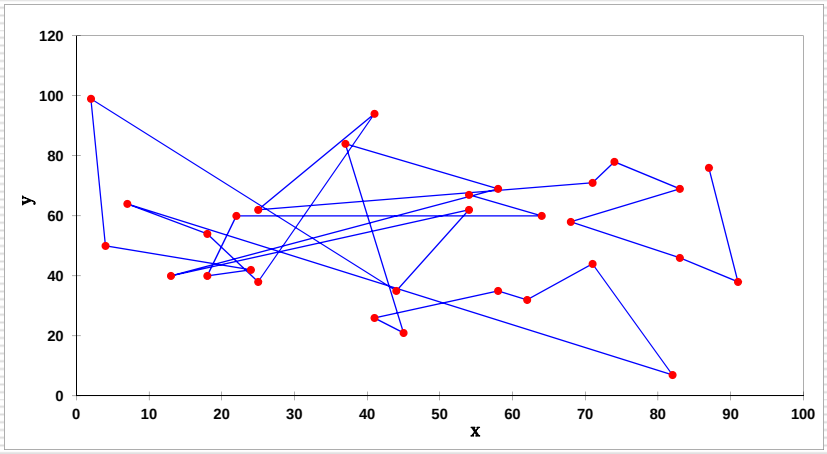
مثال: داده های مربوط به مساله TSP با n=30

No.	x	y	No.	x	y	No.	x	y
1	82	7	11	58	69	21	4	50
2	91	38	12	54	62	22	13	40
3	83	46	13	54	67	23	18	40
4	71	44	14	37	84	24	24	42
5	64	60	15	41	94	25	25	38
6	68	58	16	2	99	26	41	26
7	83	69	17	7	64	27	45	21
8	87	76	18	22	60	28	44	35
9	74	78	19	25	62	29	58	35
10	71	71	20	18	54	30	62	32

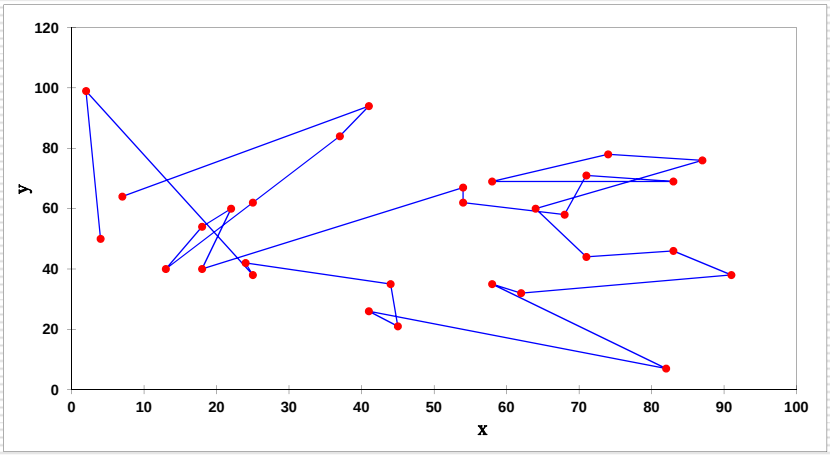
موقعیت شهرهای مساله TSP با n=30



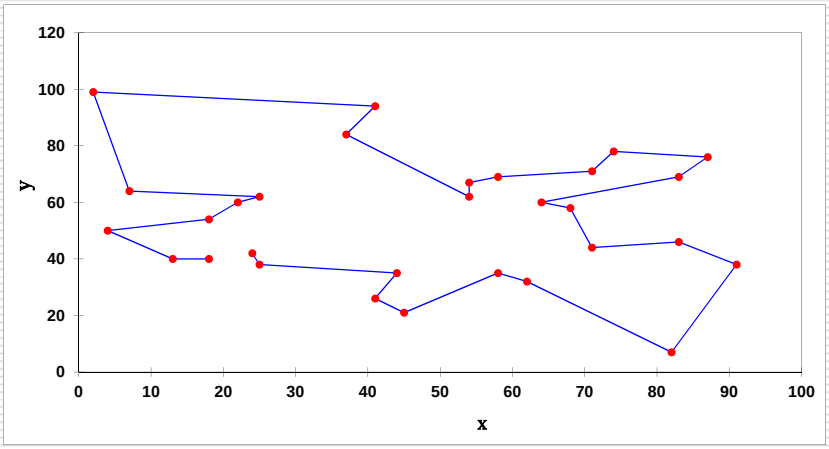
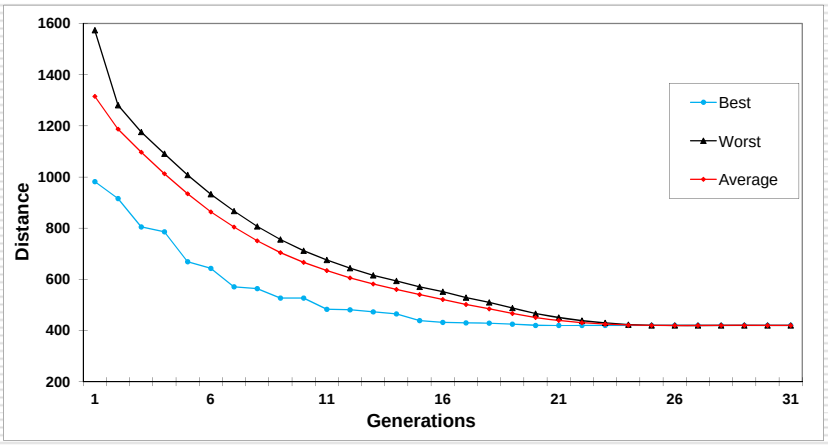
یک جواب موجه مساله TSP Distance=941



یک جواب موجه مساله TSP Distance=652



جواب بهینه مساله TSP Distance=420



نمودار همگرایی الگوریتم ژنتیک در مساله TSP