به نام خدا

و بگو :
پروردگارا مرا [در هر کاری] با درستی و راستی داخل کن
و با درستی و راستی خارج ساز
و از سوی خود تسلّطی یاری بخش قرار ده.

اسراء ۸۰

کاربرد تکنیک های محاسبات تکاملی در برنامه ریزی توسعه تولید

Application of Evolutionary Computation Techniques for Generation Expansion Planning

مروری بر انواع روش‌های بهینه سازی

- هدف از طراحی بهینه

Minimize or Maximize : $f(x)$

در سیستم‌های قدرت سنتی تابع هدف بصورت تابع هزینه معرفی می‌شود که بایستی مینیمم شود در حالیکه در سیستم‌های قدرت تجدید ساختار یافته معمولاً به صورت تابع سود بازیگران بازار برق تعریف می‌شود که بایستی ماکزیمم گردد.

جواب بهینه مسئله : x_0

برای مینیمم‌سازی $f(x_0) \leq f(x)$

برای ماکزیمم‌سازی $f(x) \leq f(x_0)$

فرمول‌بندی مسئله GEP

- **تابع هدف**

$$\text{Min } C_j = \sum_{t=1}^{T} [I_{j,t} - S_{j,t} + M_{j,t} + O_{j,t}]$$

- Capital investment costs (I)
- Salvage value of investment costs (S)
- Maintenance costs (M)
- Cost of the energy not served (O)

- **قیود**

$$[K_t] = [K_{t-1}] + [A_t] - [R_t] + [U_t]$$

- $[K_t]$: Vector of committed additions of units in year t
- $[A_t]$: Vector of committed retirements of units in year t
- $[R_t]$: Vector of candidate generating units added to the system in year t
- $[U_t]$: the system configuration vector

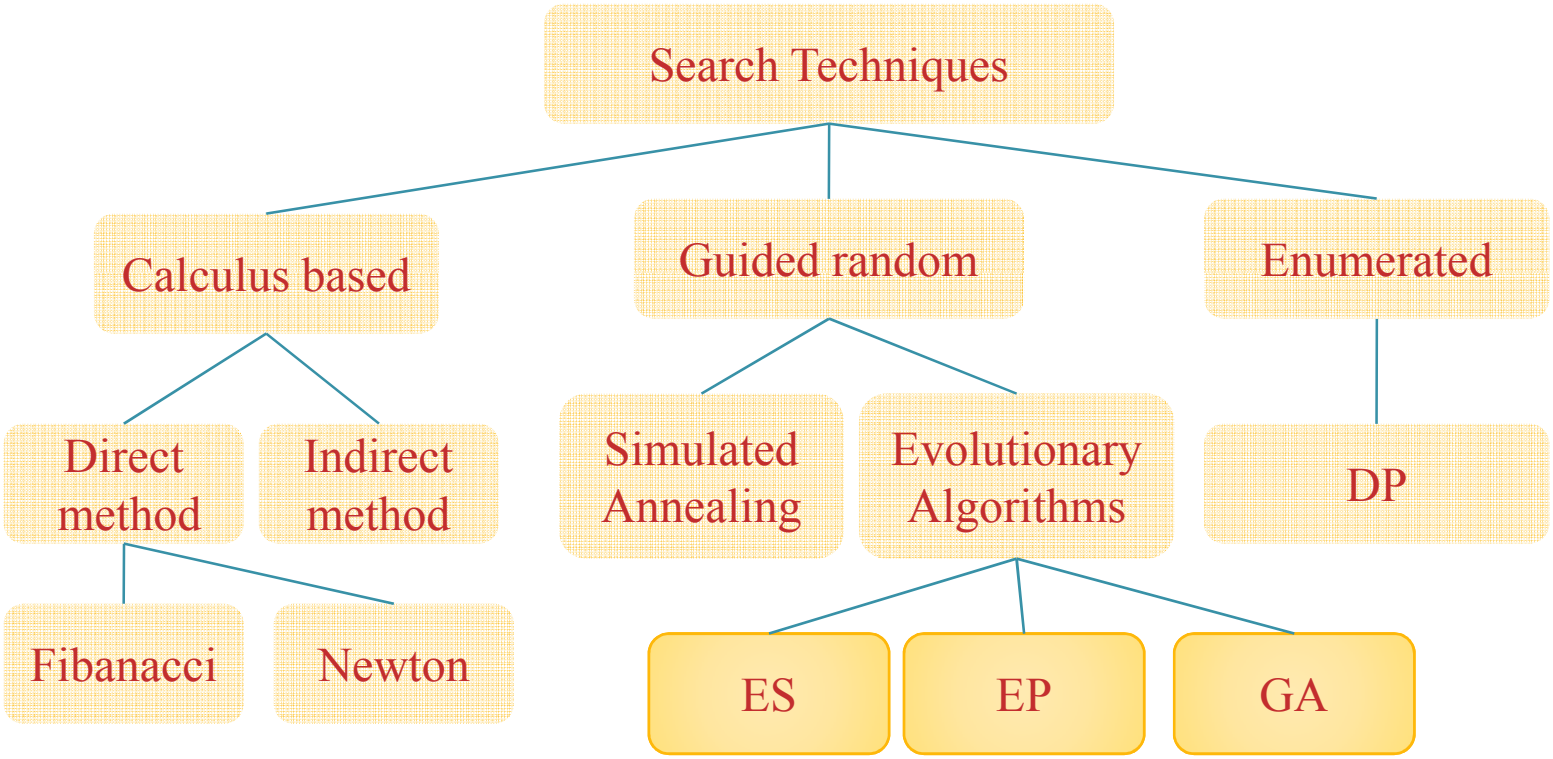
$$0 \leq [U_t] \leq [U_{t,max}]$$

$$R_{min} \leq [K_{t,a}] \leq R_{max}$$

$$FM_{min} \leq [K_{t,a}] \leq FM_{max}$$

$$LOLP(K_{t,a}) \leq \varepsilon$$

مروری بر انواع روش‌های بهینه‌سازی بر اساس جستجو



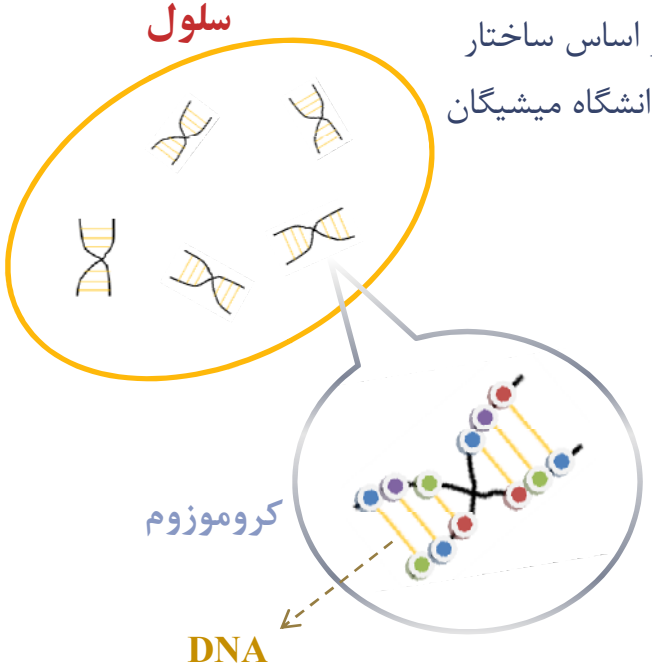
تاریخچه GA

- **پیش زمینه بیولوژیکی ژن‌ها و کروموزوم‌ها :**

✓ ایده اصلی الگوریتم های تکاملی (مانند ES و EP)توسط *Rechenberg* در سال ۱۹۶۰ مطرح گردید.

✓ **GA** منشعب از این نوع الگوریتم ها می باشد.

✓ **GA** روش جستجوی کامپیوتری بر پایه الگوریتم های بهینه‌سازی و بر اساس ساختار ژن ها و کروموزوم ها است که توسط پروفسور *John Holland* در دانشگاه میشیگان مطرح شد.



✓ مهمترین تفاوت‌های **GA** با سایر الگوریتم های بهینه‌سازی:

۱) کد گذاری متغیرهای مسئله (Encoding)

۲) اجرای عملیات بهینه‌سازی بر روی مجموعه‌ای از نقاط (Population)

فضای جستجو (Search Space)

- **تعریف:**

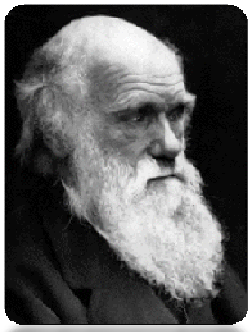
فضای همه حالت‌های ممکن در حل یک مسئله را فضای جستجو می‌نامند.

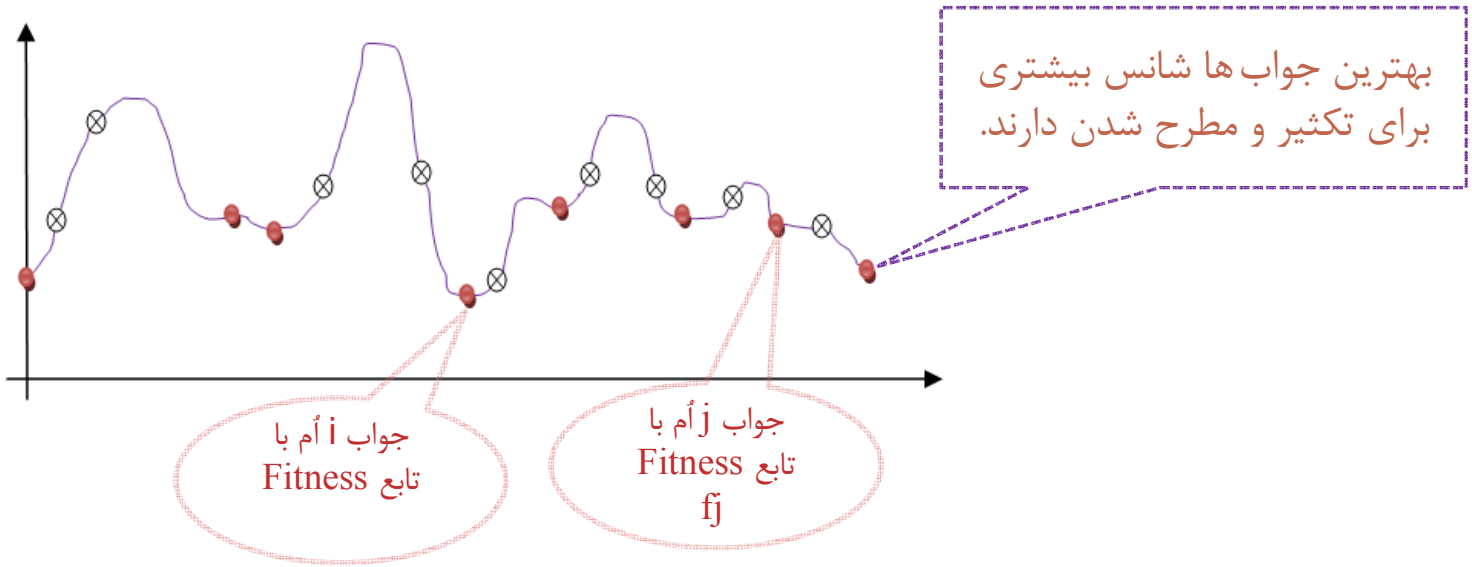
✓ جستجو برای جواب یعنی بررسی برای پیدا نمودن اکسترمم (Minیا Max) در فضای جستجو.

✓ پیچیده و بزرگ بودن فضای جستجو و اینکه باید حل مسئله را از چه نقطه‌ای آغاز نمود ما را به استفاده از **GA** سوق می دهد.

- **GA** با الهام از تئوری ادامه حیات بهترین ها (Survival of the fittest) و تکثیر نوع برتر *Darwin* شکل گرفته است.

Charles Darwin 1809 – 1882





مدل‌سازی مسئله در GA

- ✓ به منظور مدل‌سازی مسئله بهینه‌سازی به صورت الگوریتم ژنتیک بایستی متغیرهای تابع هدف و قیود مسئله بر اساس مفاهیم GA کُدگذاری شوند.

$$\text{Min } f(x)$$

$$s.t.$$

$$X_i^{(L)} \leq X_i \leq X_i^{(U)} \quad i = 1,2,\dots,N$$

$$\text{Max } f(x) \equiv \text{Min}\{-f(x)\}$$

- Value Encoding

انواع روش‌های نمایش ژن‌ها :

۱) String

۲) Array

۳) Tree

۴) List

کُدگذاری متغیرها در GA

(1) String :

1.1 Binary Coding

Chromosome A = 1 0 1 0 0 1 1 1

1.2 Permutation Encoding

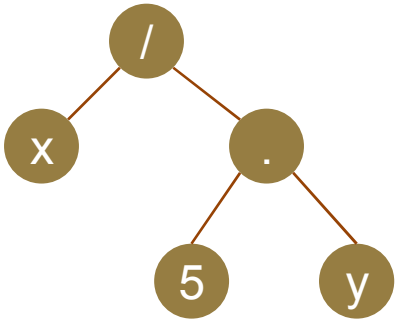
Chromosome B = 5 3 2 … n

1.3 Value Encoding

Chromosome C = 1.23 4.75 0.46

1.4 Tree Encoding

Chromosome D = x / (5.y)



تولید مثل(Reproduction)

- انواع روش های انتخاب (Selection)

۱) Roulette Wheel

۲) Tournament

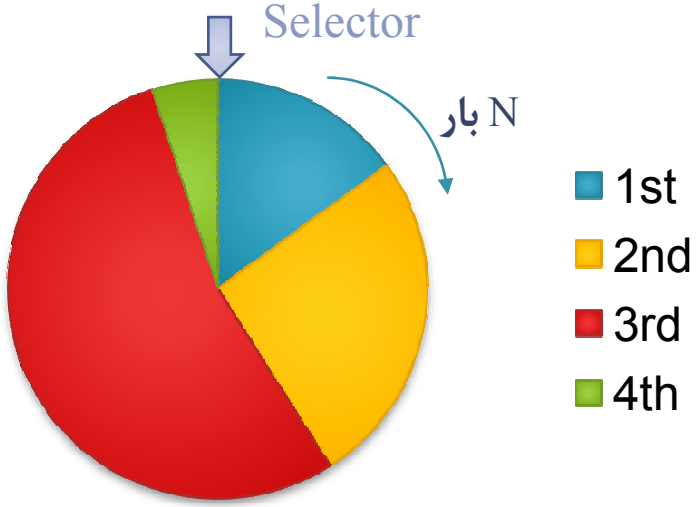
۳) Ranking

Roulette Wheel

No.	Population	F *(fitness)*	Probability	Mating Pool
1	0000 0000	1.5	0.15	0001 0101
2	0010 0001	2.6	0.26	0001 0101
3	0001 0101	5.4	0.54	0010 0001
4	1000 0011	0.5	0.05	0000 0000

$$\sum_{k=1}^{N} F_k = 10$$

$$P_i = \frac{F_i}{\sum_{k=1}^{N} F_k}$$



Tournament

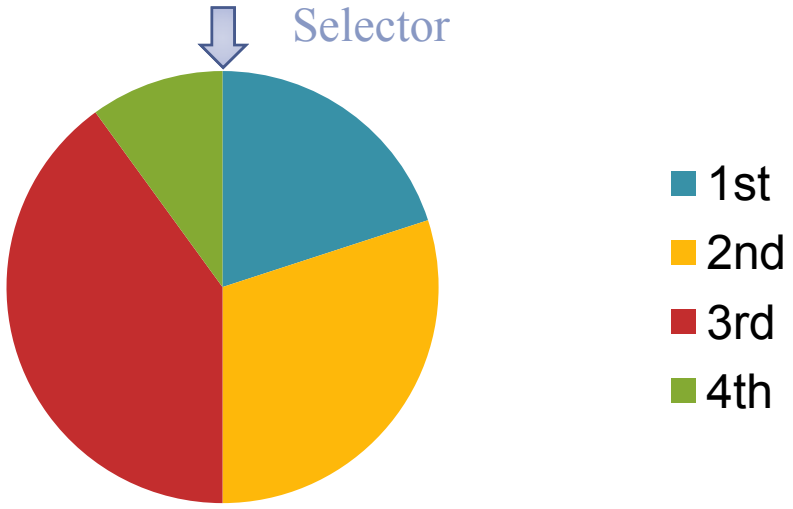
No.	1	2	3	4
Fitness	1.5	2.6	5.4	0.5

انتخاب تصادفی	برنده	Mating Pool
(1,3)	3	0001 0101
(2,3)	3	0001 0101
(3,4)	3	0001 0101
(2,4)	2	0010 0001

Ranking

No.	Population	F *(fitness)*	رتبه بندی	احتمال	Mating Pool
1	0000 0000	1.5	(2)	2/10	0001 0101
2	0010 0001	2.6	(3)	3/10	0001 0101
3	0001 0101	5.4	(4)	4/10	0010 0001
4	1000 0011	0.5	(1)	1/10	0000 0000

$\sum Rank = 10$



دو رگه‌سازی (Crossover)

- عمل تولید مثل(Reproduction) یا انتخاب(Selection) در حقیقت فقط مجموعه‌ای از بهترین رشته‌ها (کروموزوم‌ها یا جواب‌های مسئله) را انتخاب می‌کند و هیچگونه رشته جدیدی را بوجود نمی‌آورد.

- به همین دلیل عملگر Crossover در مرحله بعد با هدف تولید رشته‌های بهتر بر روی Mating Pool اعمال می شود.

- مراحل اجرای Crossover :

۱) انتخاب دو رشته تصادفی از Mating Pool

۲) انتخاب تصادفی بخشی از طول هر رشته

۳) جابجا نمودن مقدار دو رشته در بخش‌های منتخب مرحله قبل برای هر رشته

انواع روش های اعمال Crossover

1. Single Point
2. Two Point
3. Multi Point
4. Uniform (Binary Window)
5. Matrix

انواع روش‌های اعمال Crossover

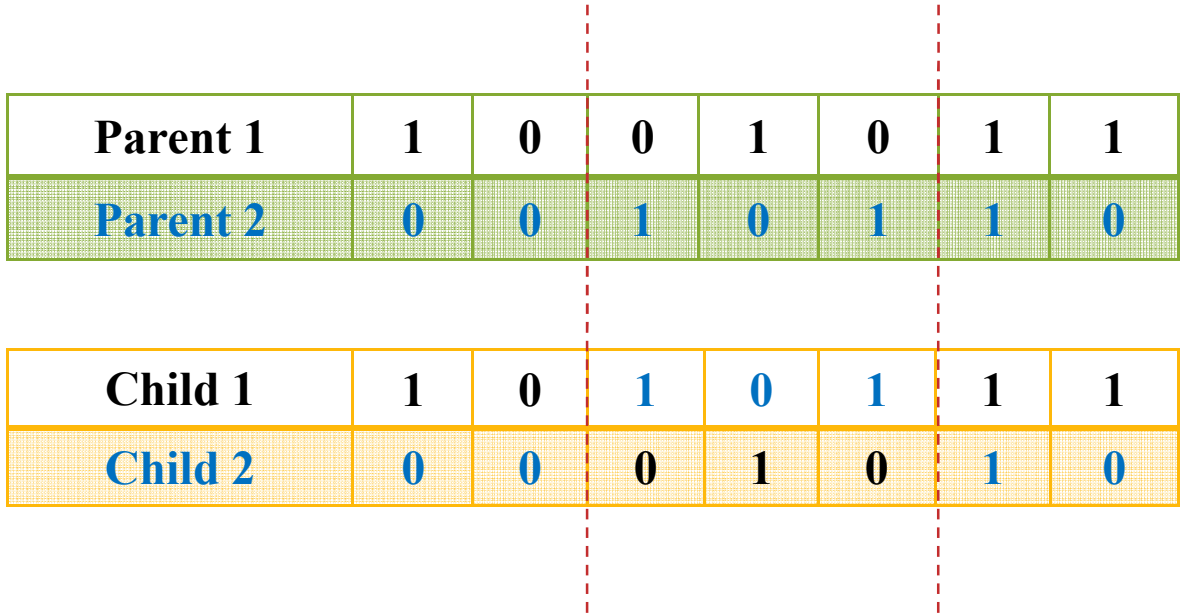
1. Single Point
2. Two Point
3. Multi Point
4. Uniform (Binary Window)
5. Matrix

Parent 1	1	0	0	1	0	1	1
Parent 2	0	0	1	0	1	1	0

Child 1	1	0	0	1	1	1	0
Child 2	0	0	1	0	0	1	1

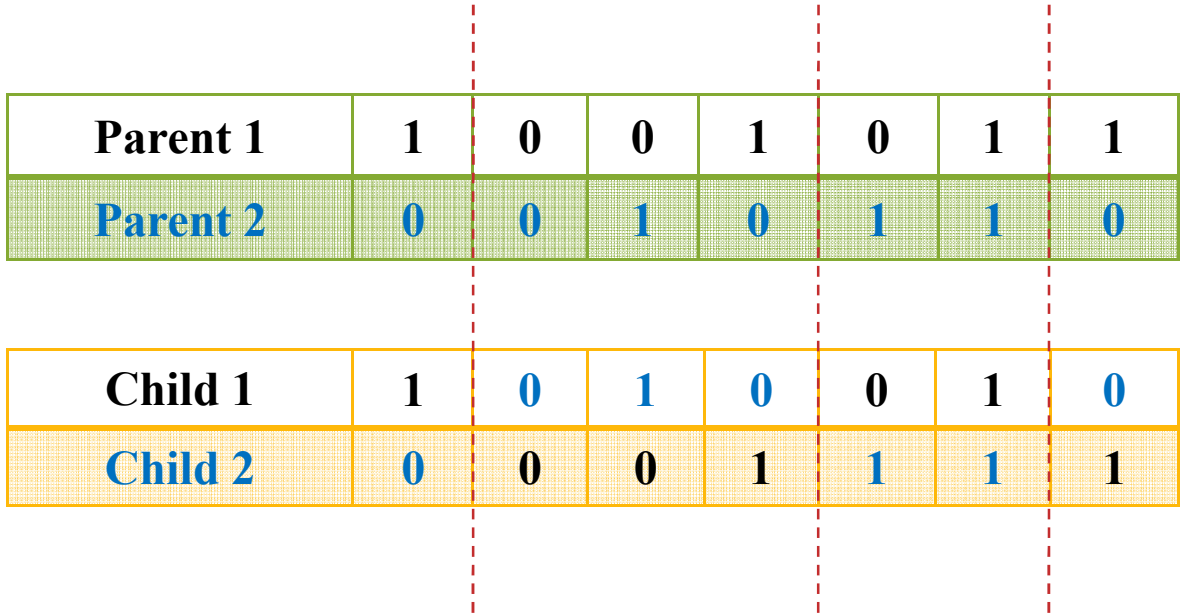
انواع روش های اعمال Crossover

1. Single Point
2. Two Point
3. Multi Point
4. Uniform (Binary Window)
5. Matrix



انواع روش‌های اعمال Crossover

1. Single Point
2. Two Point
3. Multi Point
4. Uniform (Binary Window)
5. Matrix



انواع روش های اعمال Crossover

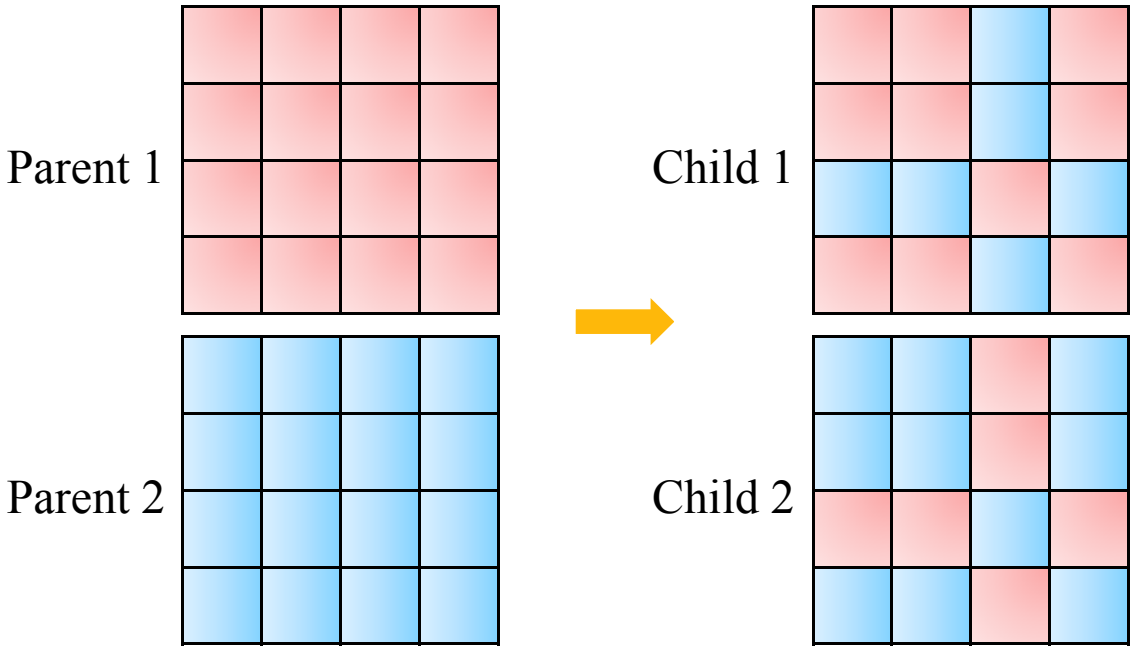
1. Single Point
2. Two Point
3. Multi Point
4. Uniform (Binary Window)
5. Matrix

Mask	1	0	1	1	0	1	0

Parent 1	1	0	0	1	0	1	1
Child 1	1	0	0	1	1	1	0
Parent 2	0	0	1	0	1	1	0
Child 2	0	0	1	0	0	1	1

انواع روش‌های اعمال Crossover

1. Single Point
2. Two Point
3. Multi Point
4. Uniform (Binary Window)
5. Matrix



(Crossover Rate) Pc

- Pc احتمالی است که از نسبت تعداد کروموزومهای جدید (فرزندان) به جمعیت اعضای Mating Pool (Np) بدست می آید($0 \leq Pc \leq 1$).
- به عبارت دیگر 100×Pc درصد از رشته ها (والدین یا کروموزوم های قدیمی) در عملیات Crossover شرکت کرده اند و 100×(1-Pc) درصد از جمعیت (Np) باقی مانده است.
- هر چه مقدار Pc بیشتر باشد بدین معنی است که تعداد کروموزوم های جدید و زیادتری وارد Mating Pool شده اند.
- اگر مقدار Pc خیلی زیاد باشد باعث می شود تا فرصت تطابق در کروموزوم ها از دست برود و در واقع از جواب اوّلیه فاصله زیادی بگیریم و اگر مقدار Pc خیلی کم باشد مسلماً تعداد فرزندان تولید شده برای دسترسی به نسل بهتر کم خواهد بود.
- Pc برای جمعیت هایی با اندازه ۳۰ تا ۲۰۰ مورد در محدوده ۰/۵ تا ۱ خواهد بود.

جهش (Mutation)

- این عملگر به منظور جلوگیری از همگرایی سریع و کمک به الگوریتم جستجو برای فرار از به دام افتادن در Minimumهای محلّی تابع هدف استفاده می‌شود.
- این فرآیند همچنین برای حفظ متفاوت و متمایز بودن کروموزوم‌ها در یک جمعیت به کار می‌رود مثلاً :

0110

0011

0001

0111 Mutation → 1111

- انواع عملگرهای جهش:
  - 1’s Complement
  - Logic Operators (AND , OR , XOR)
  - Shift Operators (Right (>>) , Left (<<))

(Mutation Rate) Pm

- Pm بیانگر احتمال جهش است که بر اساس تعداد بیت‌های جهش یافته محاسبه می‌شود.
- Pm مقدار کوچکی دارد که برای یک جمعیت با اندازه ۳۰ تا ۲۰۰ مورد، مقداری بین ۰/۰۰۱ تا ۰/۵ خواهد داشت.
- مراحل جهش :

۱) عددی تصادفی بین ۰ تا ۱ انتخاب می‌شود ($0 \le Z \le 1$,Z: random)

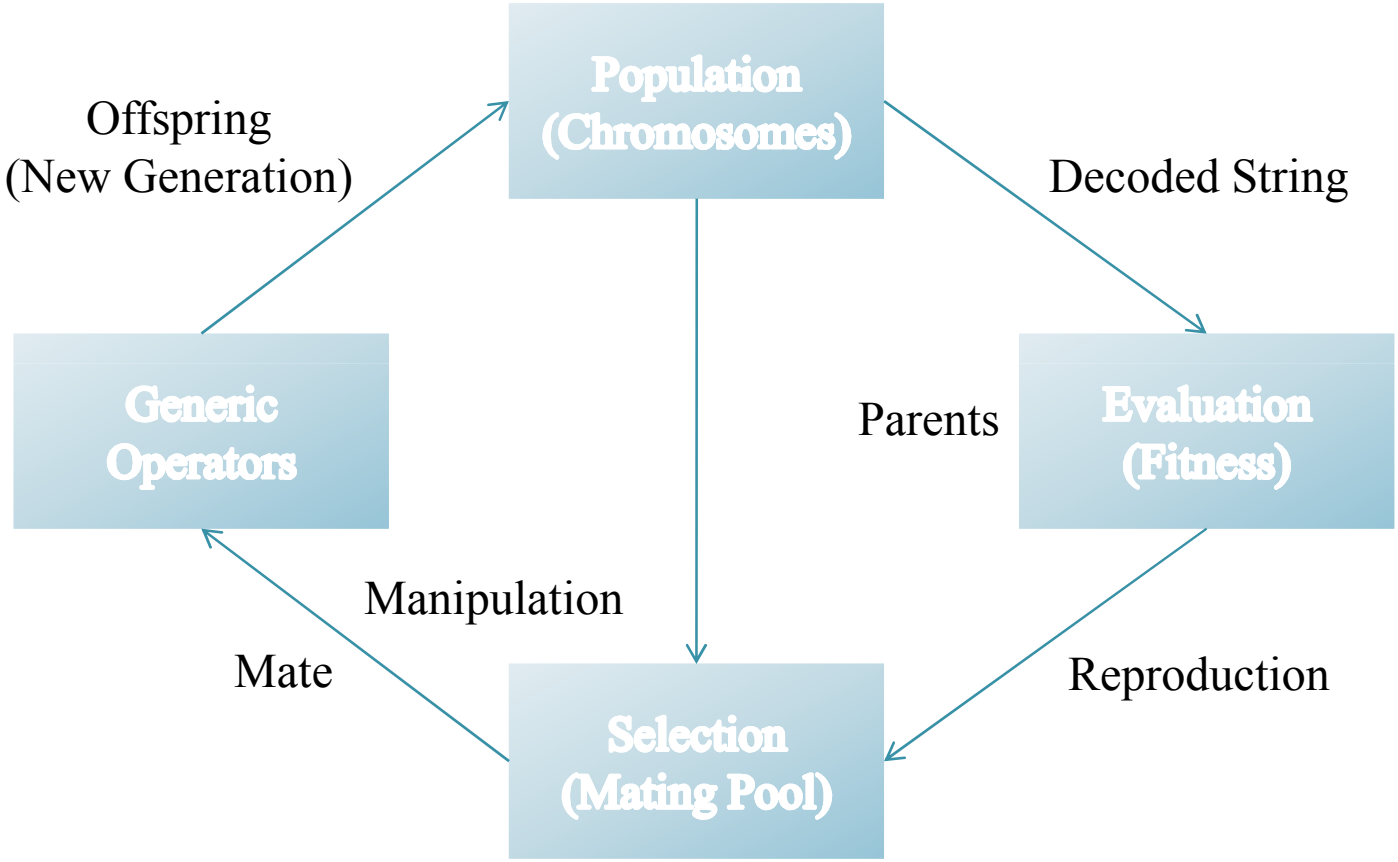
۲) این عدد با مقدار احتمال جهش (Pm) مقایسه می‌شود :

if $Z < Pm$ then ‘true’ => Toggle

if $Z > Pm$ then ‘false’ => Freeze

✓ باید توجه داشت که بیت‌های یک رشته به صورت مستقل جهش می‌یابند و اثری بر یکدیگر ندارند.

چرخه GA


Population
(Chromosomes)
Offspring
(New Generation)
Decoded String
Generic
Operators
Parents
Evaluation
(Fitness)
Manipulation
Mate
Reproduction
Selection
(Mating Pool)

استراتژی تکاملی (Evolutionary Strategy)

1) Initialization

جمعیت اوّلیه متغیرهای تصمیم گیری (Np) بصورت تصادفی از یک مجموعه با توزیع یکنواخت تعیین می شود.

2) Fitness evaluation

تعیین مقادیر f_{sum} ،f_{avg} ،f_{min} ، f_{max} برای جمعیت مورد نظر.

3) Mutation

$$Off_{i,j} = P_{i,j} + N(0, \beta\nabla)$$

4) Competition and Selection

چون جهش روی تمامی خانواده ها اعمال می شود لذا تعداد کل اعضای حوضچه رقابت برابر 2Np خواهد شد لذا در این مرحله این تعداد براساس تابع *Fitness*، خود به صورت نزولی مرتب شده و از میان آنها تعداد Np عضو برای نسل بعد برگزیده می شوند.

5) Convergence

$$\frac{f_{avg}}{f_{max}} \leq \Phi$$

مقدار Φ باید بسیار نزدیک به عدد یک باشد.

برنامه‌ریزی تکاملی (Evolutionary Programming)

1) Initialization

مشابه ES

2) Fitness evaluation

مشابه ES

3) Mutation

$$Off_{i.i} = P_{i.i} + N(0,\sigma)$$

$$\sigma = \beta \times \frac{f_i}{f_{min}} \times \left(X_{max,j} - X_{min,j}\right)$$

4) Competition and Selection

چون جهش روی تمامی خانواده‌ها اعمال می‌شود لذا تعداد کل اعضای حوضچه رقابت برابر 2Np خواهد شد لذا m عضو از بهترین‌های آن برای شرکت در نسل بعد برگزیده شده و تعداد 2Np-m عضو باقی مانده وارد فرآیند انتخاب براساس رقابت می‌شوند تا از میان آنها نیز k عضو برتر (k=Np-m) برای ورود به نسل بعد معرفی شوند.

5) Convergence

مشابه ES

$$W_i = \sum_{t=1}^{N} w_{i,t}$$

$$F_{i,t} = U - \frac{f_r}{f_r + f_i} \qquad w_{i,t} = \begin{cases} 1 & if \quad F_{i,t} < 0 \\ 0 & o.w. \end{cases}$$

بهبود روش‌های تکاملی(ES, EP, GA)

1. Domain Mapping Procedure
2. Penalty Function Approach
3. Random Mutation

بهبود روش‌های تکاملی(ES, EP, GA)

1. Domain Mapping Procedure
2. Penalty Function Approach
3. Random Mutation

- ✓ در این روش چندین متغیر مسئله به یک متغیر ساختگی و منفرد نگاشت می‌شوند تا سرعت پردازش و کارایی الگوریتم افزایش یابد.
- ✓ مثال: فرض کنید سیستم قدرتی دارای 2×1000 MW و 2×700 MW باشد بنابراین

X1 (1000 MW)	X2 (700 MW)	MW کل	Y1(متغیر ساختگی)
0	0	0	0
0	1	700	1
1	0	1000	2
0	2	1400	3
1	1	1700	4
2	0	2000	5
1	2	2400	6
2	1	2700	7
	2	3400	8

بهبود روش‌های تکاملی(ES, EP, GA)

1. Domain Mapping Procedure
2. Penalty Function Approach
3. Random Mutation

- ✓ این روش در واقع به اصلاح Fitness Function می‌پردازد بدین صورت که برای هر کروموزوم موجود در جمعیت یکبار مقدار تابع هدف را محاسبه و تمامی قیود مسئله را بررسی می‌نماید (قیود این مسئله عبارتند از reserve rate ، fuel mix rate و LOLP). چنانچه قیدی نقض شده باشد ضریب جریمه‌ای متناسب به مقدار تابع هدف مسئله افزوده می‌شود.
- ✓ Fitness Function اصلاح شده توسط روش فوق :

$$f_j = [C_j + \alpha_1.\psi_1 + \alpha_2.\psi_2 + \alpha_3.\psi_3]$$

f_j : the fitness function value of j th individual
C_j : the objective function value of j th individual
α_1 : the penalty factor of reserve margin constraint
Ψ_1: the amount of reserve margin constraint violation
α_2 : the penalty factor for fuel mix ratio constraint
Ψ_2 : the amount of fuel mix ratio constraint violation
α_3 : the penalty factor for LOLP constraint
Ψ_3 : the amount of LOLP constraint violation

بهبود روش‌های تکاملی(ES, EP, GA)

1. Domain Mapping Procedure
2. Penalty Function Approach
3. Random Mutation

- ✓ همانطور که گفته شد جهش به منظور تغییر ژن یک کروموزوم یا تمامی کروموزوم‌های موجود در یک جمعیت استفاده می شود.
- ✓ در IES و IEP جهش بر روی تمامی کروموزوم‌ها صورت می گیرد که در روابط تعریف شده برای آنها مقدار β را عددی تصادفی بین ۰/۰۵ تا ۰/۱ انتخاب می نمایند امّا در IGA عمل جهش بر روی تمامی کروموزوم ها اعمال نمی شود بلکه فقط بر روی کروموزوم های خاصی که بصورت تصادفی تعیین شده‌اند (Pm) و آنهم فقط بر روی برخی از ژن های منتخب این کروموزوم‌ها اعمال می گردد.

- IGA=Improved Genetic Algorithm
- IEP=Improved Evolutionary Programming
- IES=Improved Evolutionary Strategy

Test Results

- روش های IES، IEP و IGA بر روی یک شبکه نمونه به منظور بهینه سازی مسئله GEP مورد نظر و برای سه افق برنامه ریزی ۴ ، ۶ و ۱۴ ساله پیاده سازی شد و با توجّه به فرضیات مسئله، نتایج زیر بدست آمد.
- فرضیات مسئله :

۱) بار پیش بینی شده

Stage	0	1	2	3	4	5	6	7
سال	2002	2004	2006	2008	2010	2012	2014	2016
پیک بار (MW)	5000	7000	9000	10000	12000	13000	14000	15000

۲) حدود رزرو: بین ۲۰٪ تا ۴۰٪

۳) هزینه انرژی تأمین نشده(LOEE): ۰/۰۵ $/KWh

۴) هزینه سرمایه گذاری فرض شده است که از ابتدای سال محاسبه می شود.

۵) هزینه Salvage در انتهای دوره برنامه ریزی اعمال می شود.

Test Results

- تنظیم پارامترهای مربوط به انواع تکنیک‌های EC :

پارامتر	IES	IEP	IGA
اندازه جمعیت	50	50	50
بیشترین زاد و ولد	200	200	200
طول رشته	8	8	8
احتمال جهش	β=0.05 to 0.1	β=0.05 to 0.1	Pm=0.01
احتمال دو رگه سازی	-	-	Pc=0.6
تعداد نخبه ها	-	10 ~ 20 %	2
متد انتخاب	-	انتخاب بر مبنای رقابت	Roulette Wheel
متد دو رگه سازی	-	-	Binary Window
قیود	Reserve margin, Fuel Mix Ratio, LOLP		

Test Results

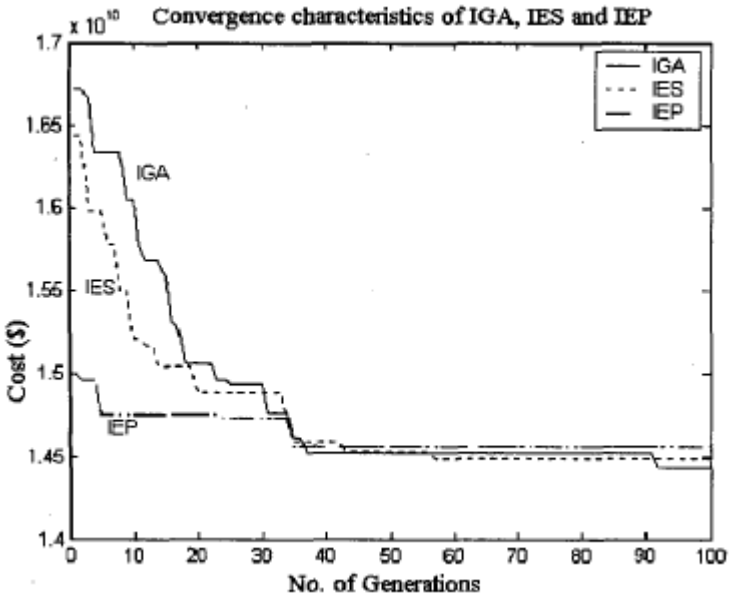
نتایج بدست آمده برای افق های برنامه‌ریزی کوتاه و میان مدّت

روش حل	دوره برنامه‌ریزی ۴ ساله			دوره برنامه‌ریزی ۶ ساله		
	زمان اجرا (دقیقه)	هزینه تجمعی ($\times 10^9$)	خطا (%)	زمان اجرا (دقیقه)	هزینه تجمعی ($\times 10^9$)	خطا (%)
DP	1	5.0058	0	28	7.359	0
IGA	5	5.0058	0	8	7.539	0
IES	4.9	5.0058	0	8	7.539	0
IEP	4.5	5.0058	0	8	7.539	0

DP = Dynamic Programming
IGA = Improved Genetic Algorithm
IEP = Improved Evolutionary Programming
IES = Improved Evolutionary Strategy

Test Results

نتایج بدست آمده برای افق برنامه ریزی بلند مدّت (۱۴ سال)			
متد	IES	IEP	IGA
هزینه ($)	1.4483×10^10	1.4555×10^10	1.4431×10^10
زمان اجرا (دقیقه)	38	41	26



نتیجه گیری :

- متدهای IES، IEP و IGA برای مسئله کمینه نمودن هزینه های GEP مورد بررسی قرار گرفت و نتایج آن با روش برنامه‌ریزی پویا (DP) به ازای افق های برنامه‌ریزی ۴ و ۶ ساله مقایسه شد که در نهایت می توان چنین نتیجه گرفت:

۱. تکنیک های EC یک روش یا چاره دیگری نسبت به روش DP برای مسائل بهینه سازی پیچیده می باشند چرا که پیدا نمودن جواب بهینه برای افق های بلندمدّت برنامه‌ریزی تولید به کمک روش DP به علّت افزایش نمایی زمان اجرای برنامه بسیار مشکل خواهد بود.

۲. نتایج مسئله برای افق های برنامه‌ریزی ۴ و ۶ ساله نشان می دهد که روش های IES ، IEP و IGA می توانند جواب های بهینه عمومی یا جوابی نزدیک به آن را برای این نوع مسائل با کمترین زمان اجرا تولید کنند.

۳. برای این نوع مسائل روش IGA نسبت به دو روش IES و IEP جوابی بهتر و در زمانی کمتر پیدا می کند.

۴. شاید با تنظیم بهتر پارامترهای روش های IES و IEP به صورت سعی و خطا بتوان تقریباً به همان جواب IGA دست یافت که مسلماً موجب افزایش زمان اجرای برنامه خواهد شد.

مراجع

- S. Kannan , et al. "*Application of Evolutionary Computation Techniques for Generation Expansion Planning*" ,IEEE 2003
- *Introduction to the WASP IV model* , User's manual. International Atomic Energy Agency , Vienna , Austria . Nov 2001
- M. Alireza , "*Genetic Algorithms for all Specialists and Researchers*"
- سید هادی حسینی و دیگران، "*ارزیابی نقش تلفات در برنامه ریزی توسعه شبکه انتقال در محیطهای با عدم قطعیت به کمک الگوریتم ژنتیک بهبود یافته*"، بیست و سومین کنفرانس بین المللی برق PSC 2008
- Genetic Algorithm Toolbox of *MATLAB 2008*

” ای انسان، تو با تلاش و رنج به سوی پروردگارت می‌روی و او را ملاقات خواهی کرد. “

انشقاق ۶