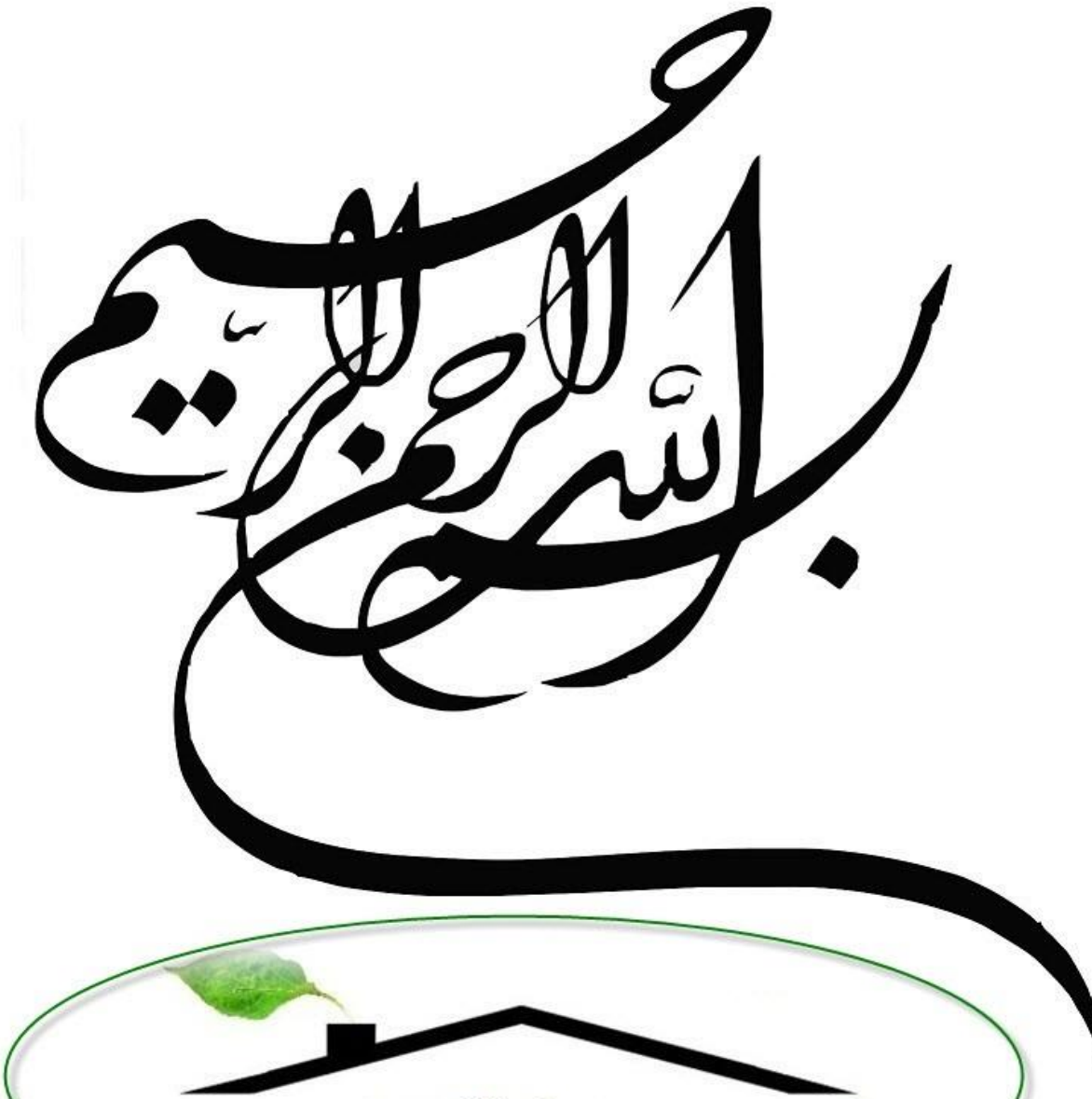
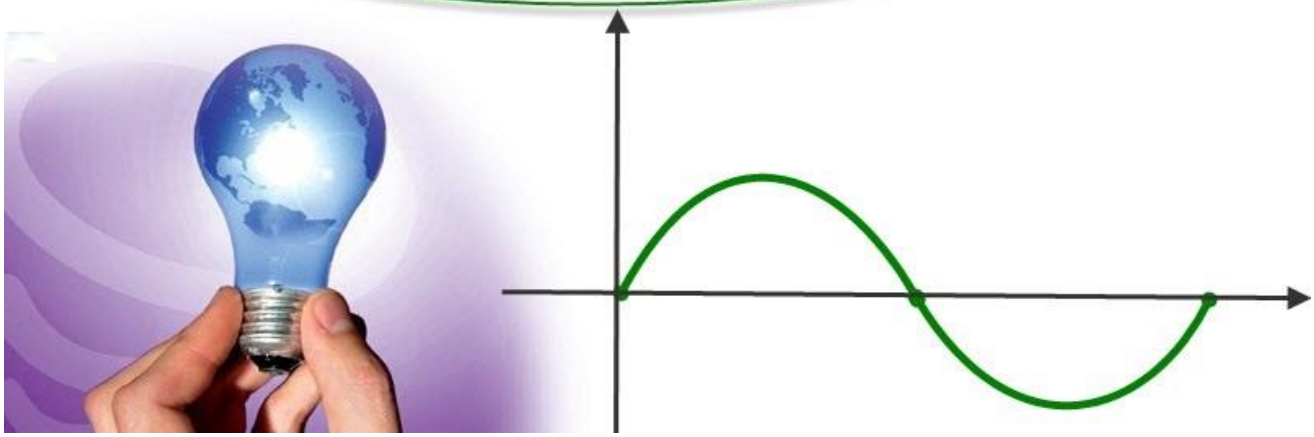
برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه





برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

موضوع پروژه:

الگوریتم های ژنتیک

WikiPower.ir

برای خرید فایل wordاین پروژه اینجا کلیک کنید.

(شماره پروژه = ۳۰۵)

پشتیبانی: ۰۹۳۵۵۴۰۵۹۸۶

بخش اول : الگوریتم های ژنتیک (GA)

فصل اول : آشنایی با مفاهیم اولیه

۱ -۱ مقدمه

به طور کلی انتخاب و طراحی بهینه در بسیاری از مسائل علمی و فنی باعث تولید بهترین محصول یا جواب ممکن در یک شـرایط خاص میشـود. برای مثال تولید محصـولات مناسـب در حوزه های مختلف فنی و مهندسی وابسته به طراحی دقیق و بهینه ی شکل، اندازه و قطعات آن محصول است. مثلاً برای ساخت بال های هواپیما مواد و شکل های مختلفی وجود دارد. اما کدامیک نتیجه ی مطلوب تری خواهد دا شت ؟ آیا از آلومینیوم یا یک آلیاژ خاص استفاده شود. بهتر است یا از مواد کامپوزیت؟ از طرفی شکل، اندازه و وزن آن با توجه به ماده ی به کار رفته چگونه باشد؟

می بینیم که در هر حالت متدهای طراحی و تصـمیم گیری متعددی به وجود می آید اما بهترین روش کدام است و چگونه می توان آن را پیدا کرد ؟

به عنوان مثال دیگر در مهندسی عمران طراحی یک سقف برای پوشش یک مکان وسیع که شامل چندین ستون است با توجه به هزینه و شرایط، نیازمند یک طراحی بهینه است. یا در حوزه ی مهندسی مکاترونیک (روباتیک) می توان به مسـیر بهینه ی حرکت بازوی یک روبات (Robot Trajectory) اشـاره کرد. به طور کلی در همه ی مسـائل به دنبال بهترین جواب ممکن می گردیم، اما از میان این همه راه حل وجواب کدامیک بهینه است ؟

از آنجایی که نتیجه ی کار با توجه به نوع انتخاب این متد ها و روش ها حاصل می شود لذا به

اهمیت موضوع انتخاب بهینه (Optimum) و بهینه سازی در همه ی مسائل پی می بریم پس :

((هدف ما این است که درفضای جواب های ممکن به دنبال بهترین جواب بگردیم.))

همان طور که می دانیم در بیشتر مسائل هدف پیدا کردن یک تابع هدف (Objective Function) و یا تابع هزینه (Cost Function) به صورت حداکثر (Maximum) یا حداقل (Minimum) است. یعنی به طور ریاضی هدف پیدا کردن X0 در A است اگر داشته باشیم :

(۱ - ۱)

برای مینیمم $F(x0) \leq F(x)$

برای ماکزیمم $F(x) \leq F(x0)$

در این میان چند سوال اساسی به وجود می آید :

۱ – اول اینکه آیا اصلاً یک حل یا جواب بهینه وجود دارد ؟

۲ – آیا این جواب یکتا است ؟

۳ – روش حل آن چگونه است ؟

٤ – میزان حساسیت این جواب بهینه چقدر است ؟

۵ – رفتار مسئله به ازای تغییرات کوچکی در پارامترهای آن چگونه خواهد بود ؟

از سال ۱۹۴۰ تاکنون روش های بهینه سازی متعددی مطرح شده است که به عنوان روش های کلاسیک شناخته می شوند. از آن جمله می توان به این روش ها اشاره کرد :

۱ – برنامه ریزی خطی (Linear Programming)

۲ – برنامه ریزی غیرخطی (Non Linear Programming)

۳ – برنامه ریزی پویا (Dynamic Programming)

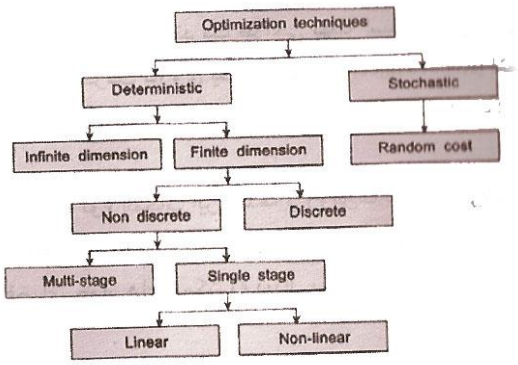
٤ – روش اکتشافی (Inventory)

٥ – روش صف (Queuing)

٦ – روش جایگزینی (Replacement)

٧ – روش زمان بندی (Scheduling) و...

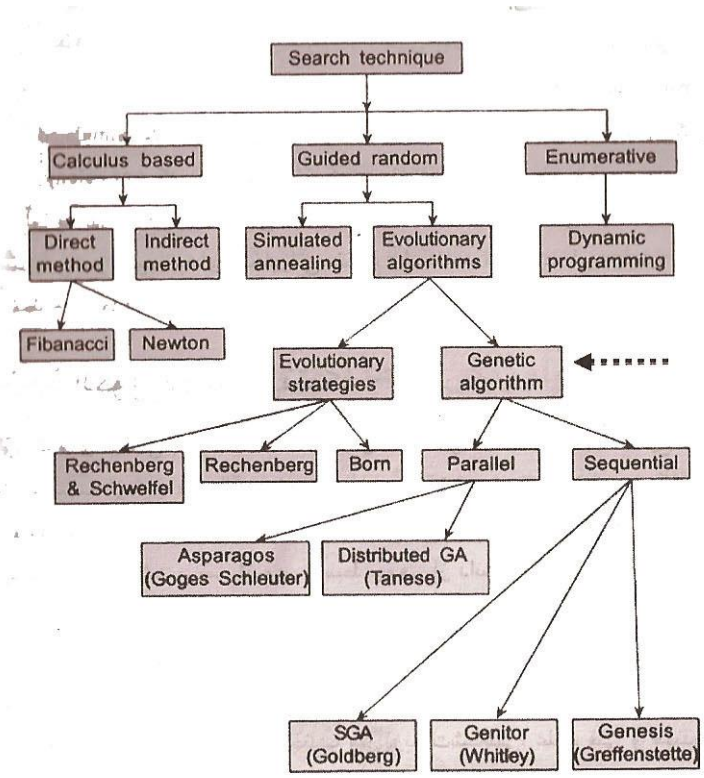
طبقه بندی کلی روش های بهینه سازی (Optimization Techniques) در شکل 1 نشان داده شده است:



شکل ۱

معمولاً از تکنیک جستجوی کلاسیک برای حل معادلات غیر خطی استفاده می کنیم شکل 2 روش های قدیمی کلاسیک و روش های جدید را به صورت طبقه بندی شده نشان می دهد.

هر مسئله ی مهندسی ممکن است دارای چندین جواب مختلف باشد که بعضی از آنها ممکن و بعضی غیر ممکن است. وظیفه ی طراحان پیدا کردن بهترین جواب ممکن از میان جواب های مختلف است. مجموعه ی جواب های ممکن فضای طراحی (Design Space) را شکل می دهند که باید در این فضا به جستجوی بهترین یا بهینه ترین جواب پرداخت.



شکل ۲

WikiPower.ir

جستجوها به دو روش انجام می شود:

۱ – قطعی (Deterministic)

۲ – محتمل یا غیر قطعی (Stochastic)

در روش قطعی (Deterministic) می توان به الگوریتم Steepest gradient و از روش غیر قطعی (Stochastic) می توان به روش تصادفی (Random) اشاره کرد. فارغ از اینکه روش جستجوی ما قطعی یا غیر قطعی باشد. هدف دستیابی به یک نتیجه ی معتبر است، یعنی نتیجه ی به دست آمده بهینه یا نزدیک به جواب بهینه با شد.اما روش های جدید بهینه سازی که امروزه در حل بـ سیاری از م سائل مختلف مورد بررسی قرار می گیرد عبارتند از :

1 - Simulated Annealing

2 - Ant Colony

3 - Random Cost

4 - Evolution Strategy

5 - Genetic Algorithm

6 - Cellular Automata

روش Simulated Annealing که تقلیدی از پدیده ی سرد شدن فلزات مذاب برای ساخت یک روال جستجو است. و یا روش Ant Colony که با الهام از زندگی دسته جمعی حشرات به خصوص مورچه پی ریزی شـــده اســـت و از آن به عنوان هوش هجومی (Swam intelligence) نیز یاد می شـــود در حل مشکل ترافیک شبکه ها و مسیر یابی در سیستم های مخابراتی شلوغ استفاده می شود. این روش اولین بار توسط Deneubourge مطرح و بعدها توسط Dorigo (1999) توسعه یافت.

WikiPower.ir

۲ – ۱ تاریخچه ی الگوریتم های ژنتیک

ایده ی اصلی الگوریتم های تکاملی (Evolutionary) در سال 1960 توسط Rechenberg مطرح گردید. الگوریتم های ژنتیک که منشعب از این نوع الگوریتم ها است، در حقیقت روش جستجوی کامپیوتری بر پایه ی الگوریتم های بهینه ســـازی و بر اســـاس ســـاختار ژن ها و کروموزوم ها اســـت که توســـط پروفسورHolland در دانشگاه میشیگان مطرح شد و پس از وی توسط جمعی از دانشجویانش مثل Goldberg و Ann Arbor توسعه یافت.

تا به امروز کتاب های متعددی توسط افرادی چون Davis - Goldberg – Michalewicz و Deb در این زمینه به رشته ی تحریر در آمده است.

این الگوریتم امروزه در بسیاری از علوم مختلف مثل زیست شناسی، علوم فنی و مهندسی (شبکه های عصبی – پردازش تصویر و تشخیص الگو و ...) علوم پایه، علوم اجتماعی و ... کاربرد دارد.

۳-۱ مفاهیم پایه

الگوریتم های ژنتیک یک روش جستجوی مؤثر در فضاهای بسیار وسیع و بزرگ است که در نهایت منجر به جهت گیری به سمت پیدا کردن یک جواب بهینه می گردد که شاید نتوان در مدت زمان نزدیکی یک فرد به آن جواب بهینه دست یافت.

الگوریتم های ژنتیک تفاوت بسیار زیادی با روش بهینه سازی قدیمی دارند. در این الگوریتم ها باید فضای طراحی (Design Space) به فضای ژنتیک (Genetic Space) تبدیل شود. بنابراین الگوریتم های ژنتیک با یک سری متغیر های کد شده کار می کنند. مزیت کار با متغیرهای کد شده در این است که اصولاً کدها قابلیت تبدیل فضای پیوسته به فضای گسسته را دارند. یکی از تفاوت های اصلی روش GA (Genetic Algorithm) با روش های قدیمی بهینه سازی در این است که در GA باجمعیت (Population) یا مجموعه ای از نقاط در یک لحظه ی خاص کار می کنیم. در حالی که در روش های قدیمی بهینه سازی تنها برای یک نقطه ی خاص عمل می کردیم. این به این معنی است که GA تعداد زیادی از طرح ها را در یک زمان مورد پردازش قرار می دهد. نکته ی جالب دیگر این ا ست که ا صول GA بر پردازش تصادفی (Random) یا به تعبیر صحیح تر پردازش تصادفی هدایت شده (Guided Random) استوار است. بنابراین عملگرهای تصادفی فضای جستجو را به صورت تطبیقی (Adaptive) مورد بررسی قرار می دهند.

اصولاً برای استفاده از GA باید سه مفهوم زیر مشخص شود:

۱ – تعریف تابع هدف (Objective Function) یا تابع هزینه (Cost Function)

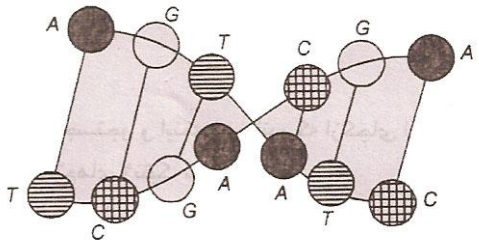
۲ – تعریف و پیاده سازی فضای ژنتیک (Genetic Representation)

۳ – تعریف و پیاده سازی عملگرهای GA

اگر این سه قسمت به طور صحیح تعریف شوند، بدون شک GA به خوبی عمل خواهد کرد و در نهایت می توان با اعمال تغییراتی کارایی سیستم را افزایش داد.

٤ – ١ پیش زمینه ی بیولوژیکی ژن ها و کروموزوم ها

همه ی ارگانیسم های زنده از سلول ها تشکیل شده اند. در هر سلول مجموعه ای از کروموزوم ها به شکل رشته ای از DNA وجود دارند که به صورت یک مدلی از کل ارگانیسم تعبیر می شوند. هر کروموزوم از یک سری ژن در بلوک های DNA تشکیل شده که در شکل زیر تعدادی از ژن های سازنده ی کروموزوم نشان داده شده است.



شکل ۲

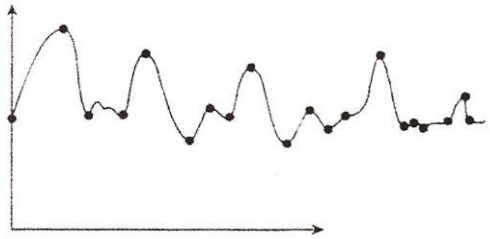
هر ژن یک الگوی خاصی را رمزگشایی (دیکود) می کند. به عبارت دیگر هر ژن یک صفت (Trait) را دیکود می کند. مثلاً رنگ چشم یک فرد به عنوان یک صفت است. مجموعه ای از این صفت ها را Alleles می گوییم. از سوی دیگر هر ژن دارای موقعیت مشخصی در کروموزوم است که به این موقعیت Locus می گویند. مجموعه ی کامل ماده ی ژنتیکی را Genome می گویند و یک مجموعه ی خصوصی از ژن ها را در ژنوم Genotype می نامند که این Genotype اساس Phenotype بوده و ویژگی های فیزیکی و فکری مثل رنگ چشم و هوش و مثل آن را به وجود می اورد.

تولید سلول های جدید

در هنگام تولید سلول های جدید یک تلفیق (Recombination) توسط عمل ادغام (Cross Over) صورت می گیرد. در این فرآیند ژ نهای والد، کروموزوم های جدید را شکل می دهند. این مولودهای جدید جهش یافته (Mutation) یعنی DNA آنها تحول یافته ا ست. این تغییرات ممکن ا ست همراه با خطا در کپی شدن ژن های والد صورت بگیرد. معیار مناسب بودن (Fitness) یک ارگانیسم با توجه به موفقیت این ارگانیسم در ادامه ی حیات آن تعیین می شود.

۵ – ۱ فضای جستجو

وقتی که یک مسئله ای را حل می کنیم هدف ما پیدا کردن بهترین جواب از میان جواب های مختلف است. فضای همه ی حالت های ممکن در حل یک مسئله را فضای جستجو (Search Space) می نامند. هر جواب می تواند با یک مقداری که بیانگر مناسب بودن (Fitness) آن است، نشان داده شود. جستجو برای جواب یعنی جستجو برای پیدا کردن اکسترمم (Maximum یا Minimum) در آن فضای جستجو.فضای جستجو را می توان با زمان حل یک مسئله و تولید نقاط به عنوان فرآیند پیدا کردن جواب بهینه به صورت زیر نشان داد :



شکل ٤

پیچیده و بزرگ بودن فضاهای جستجو و اینکه نمی دانیم که از کجای این فضا باید عمل جستجو را انجام دهیم ما را به استفاده از الگوریتم های ژنتیک سوق می دهد.

الگوریتم ژنتیک با الهام از تئوری داروین درباره ی حیات بهترین ها شـکل گرفته اسـت. بنابراین می توانیم بگوییم که :

"GA بر اسـاس اصـل " ادامه حیات بهترین ها " و " تکثیر نوع برتر " پی ریزی شـده است."

WikiPower.ir

در ابتدا الگوریتم با مجموعه ای از جواب های تصـادفی (کروموزوم ها) که به آنها جمعیت Population گفته می شود آغاز می گردد. از این جواب ها برای ساخت جمعیت جدید بعدی استفاده می شود به این امید که جمعیت های جدید بهتر از جمعیت های قدیم باشند. زیرا روش هایی که برای انتخاب جمعیت های جدید استفاده شده با توجه به مناسب (Fitness) بودن آنها صورت گرفته است. پس بهترین ها شانس بیشتری برای تولید مثل خواهند داشت. این فرآیند آنقدر تکرار می شود تا شرایط خاتمه (برای دستیابی به بهترین راه حل) محقق شود.

٦ – ١ روش کار در GA

در این مباحث بهینه سازی نامقید (Unconstrained) و بهینه سازی مقید (Constrained) با استفاده از GA بررسی می شود.

فرض کنیم که یک مسئله ماکزیمم سازی (Maximization) داشته باشیم :

Maximize F(x) (۱, ۲)

$$xi^{(l)} \le xi \le xi^{(u)} \quad \text{for } i= 1,2,\ldots,N$$

و اگر بخواهیم F(x) مینیمم شود برای F(x) > 0 تابع هدف را به صورت زیر می نویسیم :

Maximize {1 / [1 + f(x)]} (۱, ۳)

و اگر F(x) < 0 باشد به جای مینیمم کردن f(x) می توانیم ماکسیمم f(x) – را به دست آوریم.

Maximize { - f(x) } (۱و۴)

پس با این روش می توانیم مسائل مینیمم و ماکزیمم مختلف را در GA مورد بررسی قرار دهیم. اما نکته ی دیگر اینکه متغیرها در GA باید کد شوند. بنابراین در ادامه روش های کد کردن مقادیر را برر سی می کنیم.

[1] و [2] و [3]

فصل دوم : کد کردن مقادیر

۲ – ۱ کد کردن مقادیر (ENCODING)

براســاس تعریف Holland روش های متعددی برای نمایش ژن های منفرد وجود دارد. مثلاً می توان آنها را به صورت رشته (String) ، آرایه (Array) ، درخت (Tree) یا لیست (List) نشتان داد که در اینجا آنها را به صورت رشته های بیتی (String) مورد بررسی قرار می دهیم.

۲ – ۲ کد مبنای دو (Binary)

یک مثال : مسئله ی کوله پشتی (Knapsack)

در این مسئله فرض کنیم که اشیایی با مقدار و اندازه ی مشخص وجود دارد و بخواهیم آنها را در یک کوله پشتی با ظرفیت مشخص قرار دهیم. نحوه ی انتخاب اشیاء با توجه به حداقل فضایی که اشغال می کنندو استفاده ی بهینه از فضای کوله پشتی صورت می گیرد.

برای حل این مسـئله فرض کنیم هر بیت بیانگر حضـور یا عدم حضـور اشـیاء در کوله پشـتی باشـد. روش Binary Coding از روش های متداول در حل مسائل GA به شمار می آید. شکل زیر کروموزوم هایی را به صورت باینری نشان می دهد :

Chromosome A **10110110011**

اصـولاً روش کـــد مبنـای **Chromosome B** **010011001100**

دو (Binary) امکان تولید کروموزو های بسیاری را باحداقل بیت فراهم می کند. لذا این روش کدگذاری در مسائل واقعی باید همراه با اصطلاحاتی بعد از اعمال عملکرد های ژنتیکی صورت گیرد.

برای حل مسائل ماکزیمم و مینیمم سازی در GA باید متغیرهای مجهول را به صورت یک رشته ی بیتی در بیاوریم. طول این رشته با توجه به دقت مسئله مشخص می شود. برای مثال با رشته ای به طول ۴ بیت می توان ۱۶ عدد را نمایش داد. جدول 1 این مقادیر را نشان می دهد .

جدول – ۱

مقداردهدی	رشته4 بیتی	مقداردهدی	رشته 4 بیتی	مقداردهدی	رشته 4بیتی
12	1100	6	0110	0	0000
13	1101	7	0111	1	0001
14	1110	8	1000	2	0010

0011	3	1001	9	1111	15
0100	4	1010	10		
0101	5	1011	11		

تذکر : کد مبنای دو (باینری) یک عدد صحیح از تقسیم متوالی آن عدد بر دو به دست می آید.

به طور کلی اگر بخواهیم یک تابع دو متغیره را کد کنیم با فرض طول رشـــته ی 4 بیت برای هر متغیر و نمایش متغیر ها با X و X به صورت (0110 1011) و با توجه به معادلات قبلی که برای آنها یک حد بالا و پایین در نظر گرفتیم می توانیم بنویسیم :

$$xi^{(l)} \le xi \le xi^{(u)}$$

مطابق جدول ۱ (که کد مبنای دو اعداد 0 تا ۱۵ را نشـــان داده اســـت) برای نمایش کران این دو متغیر به صورت زیر عمل می کنیم :

$$(X1^{(l)} \; . X2^{(l)}) \; , \; (X1^{(u)} \; . \; X2^{(u)}) \rightarrow (0000 \;\; 0000) \; , \; (1111 \;\; 1111)$$

در واقع (0000) و (1111) دارای مقادیر کد شده ی مینیمم و ماکزیمم خواهند بود.

مقدار معادل برای هر رشته ی 4 بیتی را می توان از فرمول زیر به دست آورد :

$x_i = x_i^{(l)} + [((x_i^{(u)} - x_i^{(l)}) \times$ (مقدار رشته ی کد شده) $) / (2^{n_i} - 1)]$

به عنوان مثالی دیگر اگر برای متغیر x داشته باشیم :

$x_i^{(l)} = 2$ $x_i^{(u)} = 17$

آنگاه مقدار $x_i = (1010)$ برابر است با :

$si = 1010 = 2^3 \times 1 + 2^2 \times 0 + 2^1 \times 1 + 2^0 \times 0 = 10$

$x_i = 2 + [((17 - 2) * 10) / (2^4 - 1)] = 12$

بنابراین دقت به دست آمده برای 4 بیت در حدود 1.16 ام فضای جستجو خواهد بود، اما اگر به طول رشته یک بیت ا ضافه شود دقت به صورت تابع نمایی افزایش یافته و برابر 1.32 ام ف ضای ج ستجو می شود. بنابراین طول یک رشته ی n بیتی که بریا کد کردن یک متغیر به کار می رود بیانگر دقت آن متغیر است. پس برای رشته ای به طول n بیت مقدار دقت تقریباً برابر است با :

$(x_i^{(u)} - x_i^{(l)}) / 2^{n_i}$

و همچنین طول رشته ی s برای متغیرهای پیوسته با فرض ε به عنوان ضریب دقت از این رابطه به دست می آید :

$S = \log [(x_i^{(u)} - x_i^{(l)}) /$]

در بعضی از مواقع نیازی نیست که مقدار x یک تناظر مستقیم داشته باشد و می توان آن را به مقادیر خاصی متناظر کرد. مثلاً در جدول زیر کدهای باینری به زوایای یک فیبر متناظر شده است :

جدول – ۲

شماره	کدمبنای دو	مقدارکدشده	زاویه فیبر
1	0000	0	0
2	0001	1	10
3	0010	2	20
4	0011	3	30
5	0100	4	45
6	0101	5	60
7	0110	6	70
8	0111	7	80
9	1000	8	90
10	1001	9	- 10
11	1010	10	- 20
12	1011	11	- 30
13	1100	12	- 45
14	1101	13	- 60
15	1110	14	- 70

WikiPower.ir

16	1111	15	- 80

کد کردن سایر مبناها

برای کد کردن سایر مبناها نیز مشابه روش قبل عمل می کنیم. مثلاً برای مبنای 8 (Octal) داریم :

(X1 ^(l) . X2 ^(l)) , (X1 ^(u) . X2 ^(u))

WikiPower.ir

مقدار رشته برابر رابطه ی زیر است :

(x i ^ (u) - x i ^ (l)) / 8^n i

توجه : در مقایسه بین ژن های زیستی و الگوریتم ژنتیک می توانیم بگوییم که کروموزوم ها همان متغیر یا رشته ی کد شده و ژن ها به منزله ی بیت های این متغیر یا رشته هستند. به تعدادی از این متغیر های کد شده جمعیت (Population) می گویند و به جمعیت هایی که پس از محاسبه به دست می آید نسل (Generation) گفته می شود.

۳ – ۲ روش کدگذاری جایگشتی (Permutation Encoding)

Chromosome – A	1	5	3	2	4	7	9	8	6
Chromosome – B	8	5	6	7	2	3	1	4	9

این روش در حل مسائلی چون فروشنده ی دوره گرد (TSP) ویا مسائلی که به صورت ترتیبی هستند کاربرد دارد. همان طور که در جدول زیر مشاهده می کنیم، در این روش کروموزوم ها به صورت رشته ای از اعداد نمایش داده می شوند که هر یک از این اعداد بر اساس ترتیبی قرار گرفته اند

٤ – ۲ مسئله فروشنده ی دوره گرد Traveling Salesman Problem

تعریف مسئله : در این مسئله تعدادی شهر (نقطه) با فاصله های مشخصی وجود دارد و قرار است یک فروشنده از همه ی این شهر ها عبور کند. هدف پیدا کردن ترتیب شهرها با حداقل مسافت است. کروموزوم های زیر بیانگر ترتیب شهرهایی است که باید مورد بازدید قرار گیرند. مثلاً در کروموزوم A ابتدا شهر 1 سپس شهر 5 و الی آخر ...

Chromosome – A	1	5	3	2	4	7	9	8	6
Chromosome – B	8	5	6	7	2	3	1	4	9

نکته مهم : این مسئله تا به امروز به روش های مختلفی برای تعداد محدودی شهر حل شده است اما هیچ کدام از روش های بهینه سازی قدیمی نمی توانند این مسئله را برای n شهر حل کنند.

این مسئله با تعداد شهرهای زیاد توسط شبکه های عصبی و الگوریتم ژنتیک نیز پیاده سازی شده است که در مقام مقایسه ی عملکرد GA به مراتب بهتر از شبکه های عصبی (NN) بوده است.

از طرف دیگر کاربرد این مسئله در بسیاری از رشته های مهندسی مثل بهینه سازی در لوله های گاز، طراحی آنتن، نحوه ی قرارگیری ترانزیستورها در مدارات VLSI و ... باعث شده است که نتوان به راحتی از کنار این مسئله گذشت.

۵ –۲ روش کدگذاری مقدار (Value Encoding)

در این روش هر کروموزوم به صورت رشته ای از مقادیر است که این مقادیر می توانند هر چیز مرتبط با مسئله باشند. مثلاً اعداد اعشاری و یا اشیائ کد شده که در جدول زیر مثالی از این روش نشان داده شده است :

Chromosome – A	**1.234**	**5.3243**	**0.4556**	**2.0253**
Chromosome – B	**abdjetijdhj…**			
Chromosome – C	**(Back).**	**(Right).**	**(Forward).**	**(Left)**

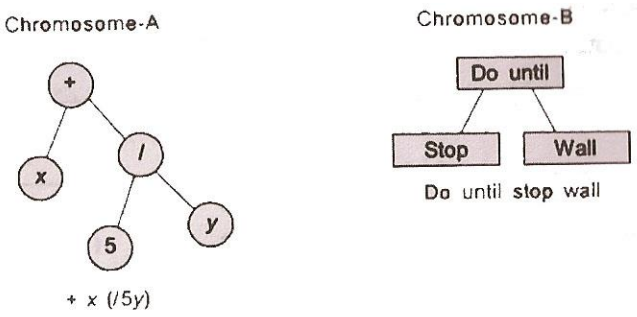
این روش در حل م سائل خا صی کاربرد دارد مثلاً برای پیدا کردن وزن های شبکه های ع صبی یعنی وزن های لیه ورودی – لا یه مخفی (Hidden) ولا یه خروجی می توان از این روش اســتفاده کرد که مقادیر کروموزوم همان وزن های شبکه خواهند بود.

WikiPower.ir

٦ – ٢ روش کدگذاری درختی (Tree Encoding)

از این روش برای برنامه نوی سی ژنتیک و زبان های برنامه نوی سی (هوش م صنوعی) ا ستفاده می شود در روش کدگذاری درختی هر کروموزوم به صورت یک درختی از ا شیاء مثل توابع (Function) یا د ستورات (Command) است (شکل 5).

به عنوان مثال، در زبان برنامه نویسی LISP از این تکنیک استفاده شده است.



شکل ه

مثال دیگری که می توان به ان اشاره کرد، پیدا کردن یک تابع مناسب برای تعدادی داده است.

در این حالت مقادیر ورودی و خروجی مشـــخص بوده و هدف پیدا کردن تابعی با بهترین تقریب بر اســـاس مقادیر به دست امده است.

۲ – ۷ تابع Fitness (Fitness Function)

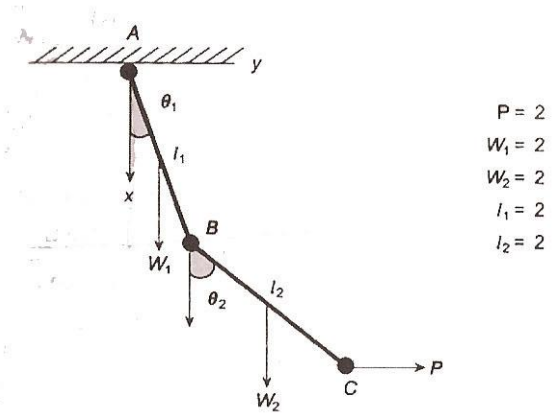
همان طور که اشـاره شـد، اصـولاً GA برای حل مسـائل ماکزیمم (بیشـینه) سـازی (Maximization) منا سب ا ست. بنابراین برای حل م سائل مینیمم (کمینه) سازی (Minimization) باید آن را تو سط یک سری تبدیل ها به فرم ماکزیمم تبدیل کنیم.

تابع Fitness به صورت F(x) یک تابع م شتق شده از تابع هدف در عملیات زنتیکی ا ست که باید مقدار آن مثبت با شد، اما در بع ضی از مواقع این هدف برآورده نمی شود. برای حل این م شکل از تبدیل های زیر استفاده می کنیم:

۲ – ۸ حل یک مسئله با GA

مطابق شکل زیر فرض کنیم دو میله ی یک سان در نقاط A , B به صورت زیر به هم مت صل شده اند که تو سط نقطه ی A پ شتیبانی می شوند و همچنین یک نیروی P در نقطه C به این میله ها وارد می شود. با

فرض چشم پوشی از اصطکاک، مقدار زاویه ها را در حالت تعادل سیستم با توجه به نیرو، وزن و طول میله ها مشخص کنید ؟



شکل ٦

مقدار پتانسیل کل برای این سیستم عبارت است از :

$$\pi = -p[(l1 \sin\theta 1 + l2 \sin\theta 2)] - [w1l1(\cos\theta 1)/2] - w2\{[(l2 \cos\theta 2)/2] + [(l1 \cos\theta 1)]\}$$

که با جایگذاری مقدار $w1$, $w2$, p و طول های آن داریم :

$$\pi(\theta 1, \theta 2) = -4 \sin\theta 1 - 6 \cos\theta 1 - 4 \sin\theta 2 - 6\cos\theta 2$$

$$0 \le \theta 1, \theta 2 \le 90$$

حال برای تعادل سیستم باید مقدار π مینیمم شود.

پس می توان نوشت :

$$\delta \pi = 0, \rightarrow \text{minimum}$$

$$\delta \pi = (\delta \pi / \delta \theta 1)\, \delta \theta 1 + (\delta \pi / \delta \theta 2)\, \delta \theta 2 = 0$$

که با مشتق گیری نسبت به θ2 و θ1 خواهیم داشت :

$(\delta \pi / \delta \theta 1) = 4 \cos \theta 1 - 6 \sin \theta 1 = 0$

$(\delta \pi / \delta \theta 2) = 4 \cos \theta 2 - 2 \sin \theta 2 = 0$

در نتیجه مقدار زاویه ها و پتانسیل بابر خواهد بود با :

$\tan \theta 1 = 0.66 , \theta 1 = 33.7(0.558 \text{ radians})$

$\tan \theta 2 = 2 , \theta 2 = 63.43(1.107 \text{ radians})$

$\pi = - 11.68$

در اینجا هدف پیدا کردن θ1 و θ2 با توجه به شرایط مسئله و حالت تعادل بود.

اما برای حل این مسئله با استفاده از GA به صورت زیر عمل می کنیم :

یک رشته ی 4 بیتی را برای هر مجهول (زاویه) در نظر گرفته و با توجه به فرمول دقت داریم:

WikiPower.ir

$\text{Accuracy} = [(x^{(u)} - x^{(l)})/(2^4 - 1)] = 90 / 15 = 6$

کدهای باینری و زوایای مربوطه را نیز از فرمول زیر محاسبه می کنیم:

$\mathbf{x_i = x + [((x_i^{(u)} - x_i^{(l)})*s) / (2^4 - 1)]}$

توجه شود که s i مقدار کد شده ه i امین کروموزوم است

شماره	کدمبنای 2	زاویه	شماره	کدمبنای2	زاویه
S . no.	Binarycoding	Angle	S . no.	Binarycoding	Angle
1	0000	0	9	1000	48
2	0001	6	10	1001	54
3	0010	12	11	1010	60
4	0011	18	12	1011	66
5	0100	24	13	1100	72
6	0101	30	14	1101	78
7	0110	36	15	1110	84
8	0111	42	16	1111	90

WikiPower.ir

مقدار کد باینری و زاویه های معادل آنها مطابق جدول زیر نشان داده شده است :

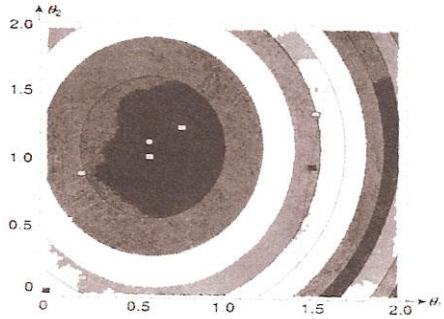
جدول - ۳

همان طور که اشاره شد تابع هدف (یا تابع هزینه) در این مسئله برابر است با :

$$\pi(\theta 1, \theta 2) = -4\sin\theta 1 - 6\cos\theta 1 - 4\sin\theta 2 - 2\cos\theta 2$$

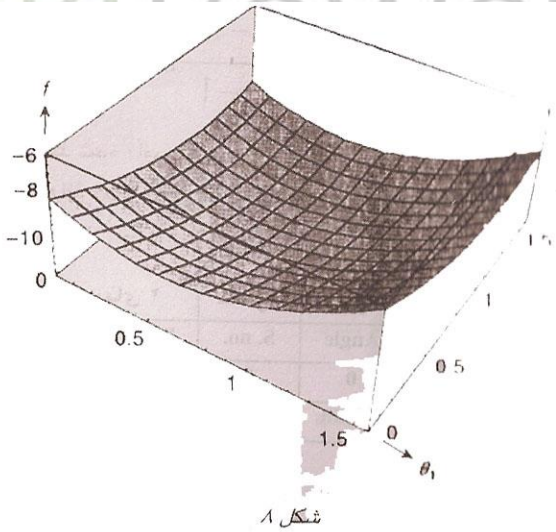
$$0 \le \theta 1, \theta 2 \le 90$$

که در شـکل زیر مسـتطیل تو پر بیانگر جمعیت های حذف شـده و مسـتطیل تو خالی بیانگر جمعیت های انتخاب شده است و دایره ی نشانه نقطه ی مینیمم است.



شکل ۷

همچنین شکل زیر تابع هدف (هزینه) را به صورت سه بعدی نشان می دهد :



شکل ۸

از آنجایی که تابع منفی است :

$$\pi(\theta1, \theta2) = -4\sin\theta1 - 6\cos\theta1 - 4\sin\theta2 - 2\cos\theta2$$

$$0 \le \theta1, \theta2 \le 90$$

به جای مینیمم کردن تابع f مقدار f - را برابر f' قرار داده (f = f'-) و آن را ماکزیمم می کنیم.

با توجه به شکل قبل :

ماکزیمم مقدار f در زمانی که θ1 و θ2 صفر است برابر 8 می شود، از این رو مقدار تابع Fitness برابر خواهد بود با :

$$F = f' - 7 = -f - 7$$

در ادامه ی حل مسئله به صورت کاملاً تصادفی هشت جمعیت (Population) با رشته هایی (کروموزوم) به طول 8 بیت مطابق جدول 4 تولید می کنیم.

لازم به ذکر ا ست که کدهای باینری برای زاویه ها با توجه به جدول 3 نو شته شده ا ست. مثلاً مقدار زاویه ی 6 برابر (0001) یا 12 برابر (0010) است :

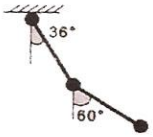
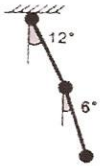
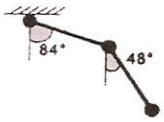
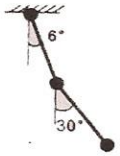
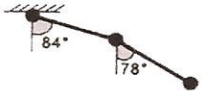
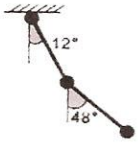
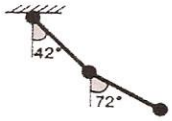
برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

جدول – ٤

Population No.	population	Angles θ۱	θ۲	F = -f-7
1	0000 0000	0	0	1
2	0010 0001	12	6	2.1
3	0001 0101	6	30	3.11
4	0010 1000	12	48	4.01
5	0110 1010	36	60	4.66
6	1110 1000	84	48	1.91
7	1110 1101	84	78	1.93
8	0111 1100	42	72	4.55

با توجه به جدول 4 مقدار تابع F در جدول بیانگر میزان منا سب بودن و نزدیک بودن به جواب ا ست، شکل 9 مقادیر F و زاویه های آن را نشان می دهد.

www.wikipower.ir
برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

جمعیت Population	تابع F	جمعیت Population	تابع F
	1		4.60
	2.10		1.91
	3.11		1.93
	4.01		4.55

شکل ۹

با توجه به شکل 9 و جدول 4 و مقادیر زاویه های به دست امده :

$\theta 1 = 33.7$

$\theta 2 = 63.43$

متوجه می شویم که در اولین نسل از جمعیت تولید شده مقدار زوایای تابع F =4.66 یعنی (60 و 36) با اندکی خطا به جواب هدف نزدیک است و این در حالی است که جمعیت اولیه به صورت کاملاً تصادفی انتخاب شده بود. به همین ترتیب هر رشته (کروموزوم) برای تولید مقدار مناسب مورد ارزیابی مجدد قرار خواهد گرفت. اصولاً برای تولید جمعیت های جدید از سه عمل اصلی تولید مثل (Reproduction)، ادغام (Cross Over) و جهش (Mutation) استفاده می شود که این جمعیت جدید نیز با توجه به شرط خاتمه مورد ارزیابی مجددقرار می گیرد. اگر شرط خاتمه محقق نشود، این عملیات دوباره تکرار شده تا نسل بهینه تر دیگری به وجود آید.

[1] و [2] و [5]

WikiPower.ir

فصل سوم : تولید مثل

۳ – ۱ تولید مثل (Reproduction)

تولید مثل معمولاً اولین عملی است که بر روی جمعیت اعمال می شود.در این روش یک سری کروموزوم از میان جمعیت (Population) به عنوان والد انتخاب شـــده که در نهایت با عمل ادغام (Cross Over) منجر به تولید فرزندان می شوند.

بر ا ساس نظریه ی حیات بهترین ها باید بهترین موارد انتخاب شوند تا ن سل بعدی بهتری را تولید کنند. به همین دلیل گاه به عملگر تولید مثل (Reproduction) عملگر انتخاب (Selection Operator) نیز گفته می شود، روش های انتخاب مختلفی در GA برای انتخاب کروموزوم ها وجود دارد، اما هدف ا صلی در همه ی آنها انتخاب رشته هایی با میانگین بالا از جمعیت فعلی و تولید کپی های چندگانه از آنها و قرار دادن آنها در یک مکان به نام استخرتولد مثل (Mating Pool) بر اساس یک فرم احتمالی است.

۳ – ۲ انواع روش های انتخاب (Selection)

روش های مختلفی برای انتخاب کروموزوم ها و ادغام آنها وجود دارد که مهمترین این روش ها عبارتند از:

۱ – روش چرخ رولت (Roulette Wheel)

۲ – روش بولتزمن (Boltzman)

۳ – روش مسابقه (Tournament)

٤ – روش رتبه بندی (Rank)

٥ – روش حالت پایدار (Steady State)

۱ – ۲ – ۳ روش چرخ رولت (Roulette Wheel)

یکی از روش های متداول در انتخاب یک رشـــته از داخل اســـتخر (Mating Pool – جایی که جمعیت های برتر قرار دارند) اسـتفاده از عملگری اسـت که مبتنی بر یک احتمالی از تابع Fitness باشـد. بنابراین ر شته i ام در جمعیت با یک احتمالی از Fi(که مقدار تابع Fitness برای آن ر شته ا ست) انتخاب می شود که باید حاصـل جمع این احتمال ها نیز برابر یک شـود. برای مثال در مسـئله 1 (مسـئله قبل) و با توجه به جدول 4 مقدار احتمال هر یک از رشته ها برابر مقادیر ستون 4 ام جدول (5) خواهد بود.

WikiPower.ir

برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

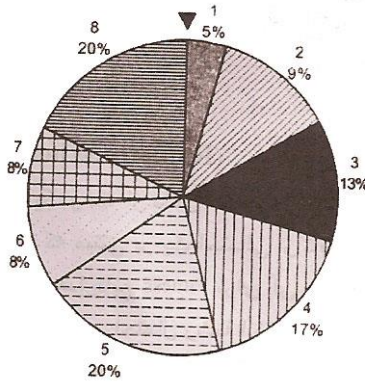
جدول – ۵

F' = 2.908

شماره	جمعیت	تابع	احتمال
Population No.	Population	F = - f - 7	β1
1	0000 0000	1	0.0429
2	0010 0001	2.1	0.0429
3	0001 0101	3.11	0.090
4	0010 1000	4.01	0.1336
5	0110 1010	4.66	0.200
6	1110 1000	1.91	0.082
7	1110 1101	1.93	0.0829
8	0111 1100	4.55	0.1955

WikiPower.ir

حال اگر در صد احتمال این مقادیر را به صورت شکل 10 یک چرخ رولت (Roulette Wheel) نشان دهیم، آن گاه مقدار مناسب بودن جمعیت پس از n بار (که در اینجا 8 بار است) چرخیدن این چرخ به دست خواهد آمد. در هر زمان یک مورد از رشته ها توسط اشاره گر چرخ رولت انتخاب می شود. از آنجایی که محیط چرخ با مقادیر درصد احتمال fitness رشته ها نشانه گذاری شده است، انتظار می رود که مکانیزم چرخ به تعداد F' / Fi کپی از رشته i ام را تولید کند.



شکل ۱۰

همان طور که در شکل 10 نشان داده شده است، 5 مورد از درصد بالاتری برخوردارند. بنابراین انتظار می رود که چرخ رولت این 5 مورد را بیشتر از سایر موارد انتخاب کند.

WikiPower.ir

نتیجه گیری

رشته هایی با مقدار Fitness بالاتر از شانس بیشتری برای کپی شدن در Mating Pool برخوردار خواهند بود که عکس این مطلب نیز صادق است، یعنی رشته هایی با Fitness کمتر احتمال کمتری برای کپی شدن در Mating Pool خواهند داشت.

برای نشان دادن کاربرد این روش به مسئله 1 باز می گردیم. با توجه به محاسبات قبلی (جدول 5) همان طور که در جدول 6 ملاحظه می شود، یک پارامتر به نام مقدار مورد انتظار Expected Count (ستون A جدول) تعریف می کنیم که مقدار آن از رابطه ی زیر به دست می آید:

Expected Count = (n = 8) * pi

سپس برای به دست آوردن احتمال تجمیعی (Cumulative) مقدار احتمال هر رشته را با مقادیر قبل از آن جمع می زنیم.که این مقادیر در ستون B جدول 6 نشان داده شده است، به عنوان مثال برای P5 داریم:

P5 = 0.0429 + 0.090 + 0.1336 + 0.1723 + 0.2 = 0.6388

برای شـــکل دادن Mating Pool نیاز به یک عدد تصـــادفی بین 0 تا 1 داریم. پس از تولید این اعداد تصادفی آنها را در ستون c قرار می دهیم.

اگر عدد تـصادفی تولید شده مثلاً 0.428 با شد، آن را متعلق به ر شته ی4 ام در نظر می گیریم. زیرا این عدد (در ستونB) در بازه ی ر شته های3 و4 (0.266 -0.438) ا ست و شماره ی ر شته (کران بالا) آن (یعنی عدد 4) را در ستون D می نویسیم.

در پایان تعداد کپی های ستون D را شمرده و آن را روبه روی شماره ی رشته ی مورد نظر در ستون E قرار می دهیم. مثلاً رشـته های 4 و 5 هر کدام دو کپی و رشـته های 6 و 7 کپی ای ندارند (صـفر) وبقیه فقط یک کپی دارند. و در ستون آخربا توجه به مقادیر E و جمعیت اولیه جمعیت هایی را (پشت سر هم) حذف یا اضـــافه کرده و جمعیت جدید را تولید می کنیم. همان طور که در جدول ملاحظه می شـــود رشـــته هایی با احتمال بیشـتر در جمعیت جدید (سـتون آخر) بیشـتر تولید شـده اند و این مبین حیات برترین ها برای تولید نسل های بعدی است

WikiPower.ir

جدول – ٦

Population No.	Poplation θ۱	θ۲	βi =Pi	A	B	C	D	E	Population θ۱	θ۲
1	0000	0000	0.0429	0.33	0.0429	0.259	3	1	0000	0000
2	0010	0001	0.090	0.72	0.1329	0.038	1	1	0010	0001
3	0001	0101	0.1336	1.064	0.266	0.486	5	1	0001	0101
4	0010	1000	0.1723	1.368	0.438	0.428	4	2	0010	1000
5	0110	1010	0.200	1.6	0.638	0.095	2	2	0010	1000
6	1110	1000	0.082	0.656	0.720	0.3	4	0	0110	1010
7	1110	1101	0.0829	0.664	0.809	0.616	5	0	0110	1010
8	0111	1100	0.1955	1.56	1.0	0.897	8	1	0111	1100

WikiPower.ir

۳ – ۲ – ۲ روش بولتزمن (Boltzman)

این روش در تکنیک جستجوی Simulated Annealing (با الهام از فرآیند سرد شدن فلزات مذاب برای پیدا کردن یک تابع مینیمم) کاربرد دارد و از مفهوم توزیع احتمال بولتزمن استفاده می کند. از آنجایی که این روش در GA چندان متداول نیست از شرح آن پرهیز می کنیم.

۳ – ۲ – ۳ روش رقابتی (Tournament)

در روش رولت گاهی با مشکلاتی از قبیل کندی و همگرایی ناگهانی جستجو به خاطر کوچک شدن سریع فضای جستجو مواجه می شویم. به همین جهت معمولاً از روش دیگری به نام روش رقابتی استفاده می شود.

بر خلاف رولت در روش رقابتی (Tournament) که شبیه زقابت در طبیعت است، یک زیر مجموعه ی کوچکی از کروموزوم ها (معمولاً دو یا سه) به صورت تصادفی انتخاب شده و به رقابت می پردازند. سرانجام در این رقابت بر اساس میزان مناسب بودن (Fitness) یکی از آنها به پیروزی رسیده و به عنوان والد جدید در Mating Pool کپی می شود. و این فرآیند تا تولید همه ی والدها در جمعیت جدید تکرار می شود.

به عنوان مثال، اگر در مسئله ی 1 بخواهیم با استفاده از روش رقابتی عمل انتخاب را انجام دهیم، خواهیم داشت :

گام 1

در ابتدا موارد 2 و 4 به صورت تصادفی (Random) برای رقابت انتخاب می شوند که در این مرحله مورد 4 برنده می شود (به خاطر Fitness بیشتر) و رشته ی آن یعنی (1000 0010) (از جمعیت اولیه – جدول 5) انتخاب و در Mating Pool کپی می شود.

	1	2	3	4	5	6	7	8
Fitness	1	2.10	3.11	4.01	4.66	1.91	1.93	4.55

www.wikipower.ir

برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

گام ۲

در گام دوم موارد 3 و 8 به صورت تصادفی انتخاب شده و در این رقابت 8 برنده می شود (چرا ؟) و رشته ان یعنی (1100 0111) در استخر کپی می شود.

برای گام های بعدی نیز به همین ترتیب عمل می کنیم که در نهایت Mating Pool آن به صورت زیر می شود :

WikiPower.ir

جدول – ۷

Population No.	Population
1	0010 1000
2	0111 1100
3	0001 0101

برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

4	0110	1010
5	1110	1000
6	0010	0001
7	0010	1000
8	0111	1100

برنده	موارد
4	2 , 4
8	3 , 8
3	1, 3
5	4 , 5
6	1 , 6
2	1 , 2
4	4 , 2
8	8 , 3

WikiPower.ir

در جدول بالا مقادیر 2 و 3 و 5 و 6 هر کدام یک کپی و مقادیر 4 و 8 دو کپی دارند ومقادیر 1 و 7 نیز حذف شـده اند در حالی که در چرخ رولت دیدیم که مقادیر 6 و 7 حذف شـدند و مقادیر 4 و 5 هر کدام دو کپی و بقیه موارد نیز یک کپی داشتند.

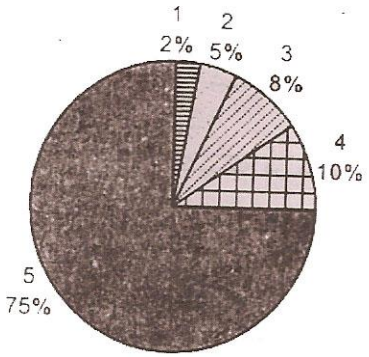
همان طور که ملاحظه می شود، عملکرد روش رقابتی بهتر ا ست. به همین دلیل این روش در جمعیت های بسیار بزرگ به عنوان بهترین روش انتخاب شناخته می شود.

٤ – ٢ – ٣ روش رتبه بندی (Rank)

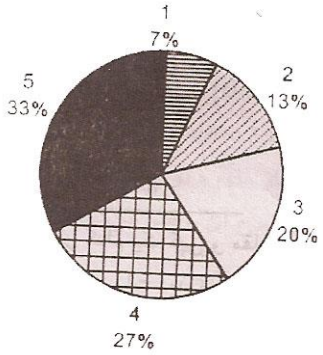
در روش رولت اگر مقادیر Fitness خیلی با هم تفاوت داشته باشند، دچار مشکل خواهیم شد.

مثلاً اگر بهترین Fitness یک کروموزوم 90 در صد با شد،در حدود 90 در صد از شکل را ا شغال خواهد کرد، بنابراین سایر کروموزوم ها شانس کمتری برای انتخاب شدن خواهند داشت.۰

در روش رتبه بندی (Rank) ابتدا جمعیت ها را رتبه بندی کرده و سپس انتخاب کروموزوم هابر اساس Fitness مبتنی بر رتبه صورت می گیرد. مثلاً بدترین حالت دارای Fitness یک و به همین ترتیب مورد بعدی دو والی آخر در نهایت بهترین مورد دارای Fitness ای برابر خواهد بود (که N تعداد کروموزوم های موجود در جمعیت است) شکل 11 تفاوت دو روش را نشان می دهد.



الف) روش رولت مبتنی بر Fitness



ب) روش رولت مبتنی بر رتبه (Rank)

شکل ۱۱

همان طور که در شکل نیز مشخص است. اشکال روش رتبه ای این است که به آهستگی همگرا (Slow Convergence) می شـــود. زیرا بهترین کروموزوم ها تفاوت چندانی با هم ندارند و همین امر باعث همگرایی اهسته می شود.

۵ – ۲ – ۳ روش حالت پایدار (Steady State)

این روش یک متد خا صی برای انتخاب نی ست و ا ساس ان بر این ا صل ا ستوار ا ست که کروموزوم های نامناسب حذف شده و والدهای جدید جایگزین می شوند و بقیه ی موارد نسل بعدی را شکل می دهند.

۳ – ۳ مفهوم برگزیده (Elitism)

بر اسـاس این مفهوم بهترین کروموزوم ها در جمعیت جدید کپی می شـوند. این روش باعث افزایش کارایی GA می گردد، زیرا مانع از گم شدن جواب های خوب به دست آمده می شود.

اگر تابع F (Fitness) مثبت باشد و برای مسئله مینیمم سازی (Minimization) به کار می رود بر اساس نظر گولدبرگ (در سال 1989) پیشنهاد می شود که Fitness هر کروموزوم از یک مقدار ثابت بزرگ کم شـود تا همه ی مقادیر Fitness مثبت باشـند. پس مقدار جدید Fitness را بریا مسـائل مینیمم (کمینه) سـازی (Minimization) می توانیم از رابطه ی زیر به دسـت آوریم :

$$\Phi 1 = (Fmax - Fmin) - Fi(x)$$

همچنین برای مسائل ماکزیمم (بیشینه) سازی (Maximization) با فرض F مثبت داریم :

$$\Phi 1 = Fi(x)$$

حال برای حل مسئله ی 1 بر اساس جدول 8 عمل می کنیم و انتظار داریم که موارد مناسب تر بیشتر کپی شوند.

همان طور که در جدول 8 ملاحظه می شود، ستون سوم (F) همان مقادیر Fitness جدول های قبلی و ستون پنجم جدول (Count) مقدار رند شده ی ستون چهارم است که بر اساس مقادیر به دست آمده در این ستون موارد 1 و 6 حذف شده اند.

جدول – ۸

(F' = 2.908)

Population No	Population	F =Φ	F / F'	Count	Mating Pool
1	0000 0000	1	0.38	0	0010 0001
2	0010 0001	2.1	0.812	1	0001 0101
3	0001 0101	3.11	1.203	1	0010 1010
4	0010 1000	4.01	1.55	1	0110 1010
5	0110 1010	4.66	1.802	2	0110 1010
6	1110 1000	1.91	0.738	0	1110 1101
7	1110 1101	1.93	0.746	1	0111 1100

0111 1100	2	1.76	1.55	0111 1100	8
		0			

---

[1] و [2] و [4] و [5]

WikiPower.ir

فصل چهارم : ادغام

۱ – ٤ عملگرهای GA

در قسمت های قبل نحوه ی کد کردن و به دست آوردن تابع Fitness و همچنین نحوه ی انتخاب کروموزوم ها را مورد بررسی قرار دادیم که بر اساس احتمال و روش های مختلف جمعیت های جدیدی را تولید می کردند.اما هیچ کدام از آنها رشته های جدیدی را در فاز تولید مثل (Reproduction) و یا فاز انتخاب تولید نمی کنند.

در این قسمت به بررسی روش های مختلف برای تولید رشته های جدید و همچنین عملگرهای مهم دیگر مثل GA ادغام (Cross Over) و جهش (Mutation) می پردازیم. عملگرهای وراثتی (Inheritance Operator) متعددی نیز برای تولید رشته های بهتر وجود دارد که هدف این عملگرها جستجوی فضای پارامترها و تا حد امکان حفظ اطلاعات نهفته در رشته ها است. چرا که این والدها بهترین موارد انتخاب شده توسط عملگرهای فاز تولید مثل هستند ونباید از دست بروند.

از مهمترین عملگرهای GA می توان به موارد زیر اشاره کرد :

۱- عمل معکوس کردن (Inversion)

۲- عمل حذف کردن (Deletion)

۳- عمل جدا سازی (Segregation)

٤- عمل نقل مکان (Migration)

٥- عمل بخش بندی (Sharing)

٦- عمل جفت گیری (Mating) یا ادغام (Cross Over)

۷- عمل غالب شدن یا تسلط (Duplication)

۸- عمل کپی کردن (Duplication)

که معمولاًدر یک الگوریتم ژنتیک (GA) ساده تنها از سه عملگر اصلی زیر استفاده می شود :

۱ -تولید مثل (Reproduction)

۲ – ادغام (Cross Over)

۳ – جهش (Mutation)

در قسمت قبل تولید مثل (Reproduction) بیان شد و در این بخش به بررسی ادغام و جهش و چند عمل دیگر می پردازیم.

۲ – ٤ عملگر ادغام (Cross Over)

پس از اینکه مرحله ی تولید مثل تمام شد، جمعیتی از بهترین ها به وجود آمده ا ست. در حقیقت عمل تولید مثل یک مجموعه ای (کولونی) از بهترین رشته ها را انتخاب کرده، اما رشته های جدیدی را به وجود نیاورده ا ست. به همین دلیل عمل ادغام (Cross Over) با هدف تولید ر شته های بهتر بر روی اعمال می شود. هدف از ادغام (Cross Over) جستجوی فضای پارامتر و تا حد امکان حفظ اطلاعات نهفته در ر شته ها است.

عملگر ادغام (Cross Over) یک عملگر ترکیبی است که که شامل سه عمل است. اول عملگر تولید مثل یک جفت ر شته را به صورت تصادفی انتخاب می کند. دوم یک محلی را برای عمل ادغام به طور تصادفی در طول رشته انتخاب کرده و سرانجام در سومین مرحله مقدار دو رشته را با توجه به محل ادغام (که مشخص کردیم) جا به جا می کند.

مثال : فرض کنیم دو رشته ی A = 11111 و B = 00000 انتخاب شده باشند، اگر موقعیت ادغام که به صورت تصادفی به دست آمده است برابر 2 در رشته باشد، مقادیر رشته های جدید پس از عمل ادغام برابر خواهند بود با :

A = 11 111 → A* = 11000

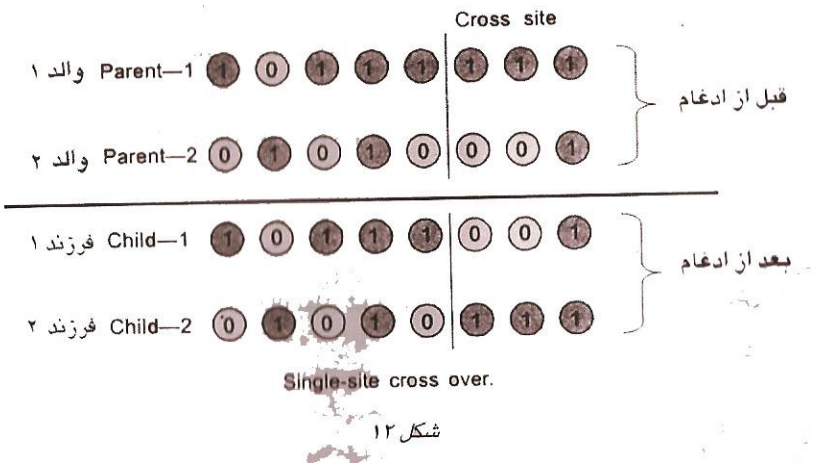
B = 00 000 → B* = 00111

این روش را اصطلاحاً روش ادغام تک مکانی (Single – Sight Cross Over) می نامند. روش های مختلف دیگری برای عمل ادغام وجود دارد که به شرح هر یک از آنها می پردازیم:

۱ – ۲ – ٤ روش ادغام تک نقطه ای یا تک مکانی (Single – Sight Cross Over)

همان طور که در مثال قبل ملاحظه شد، در این روش یک مکان تصادفی در طول رشته انتخاب می شود و بیت های پس از این مکان مطابق با شکل 12 جا به جا می شوند.

برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه



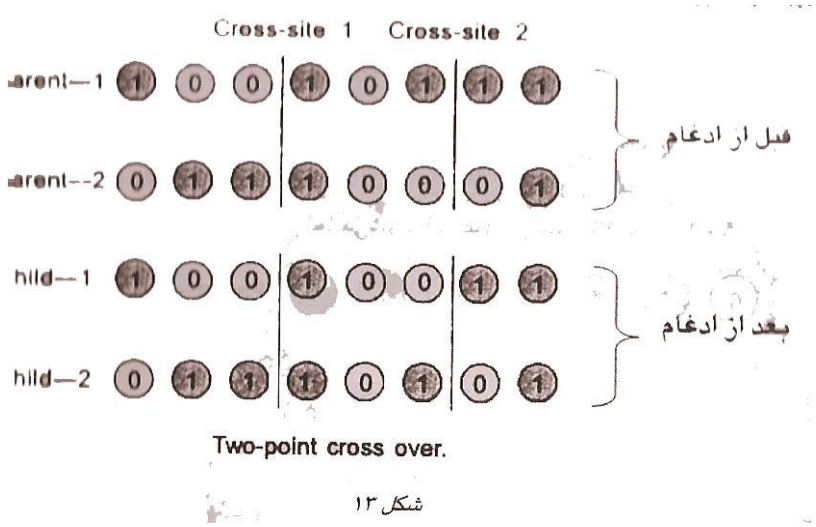
شکل ۱۲

از آنجایی که مکان جا به جایی به صورت تصادفی انتخاب شده است لذا از مناسب بودن این مکان اطلاعی نداریم. اگر این نقطه مکان مناسبی باشد منجر به تولید فرزندان مناسبی می شود در غیر این صورت باعث از بین رفتن کیفیت رشته و حذف شدن رشته در نسل های بعدی می گردد.

WikiPower.ir

۲ – ۲ –٤ روش ادغام دو نقطه ای (Two – Point Cross Over)

عملگر ادغام دو نقطه ای دو جا را به صورت تصادفی انتخاب کرده و مقادیر بین این دو نقطه را جا به جا می کند. مثلاً اگر مکان اول در بیت سوم و مکان دوم در بیت ششم باشد عمل جا به جایی آنها به صورت شکل زیر خواهد بود:



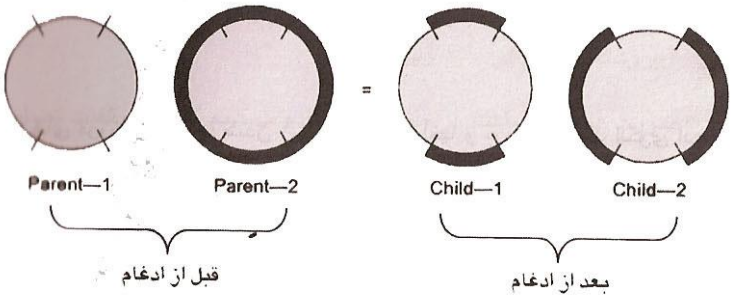
Two-point cross over.

شکل ۱۳

3 - 2 - 4 روش ادغام چند نقطه ای (Multi – Point Cross Over)

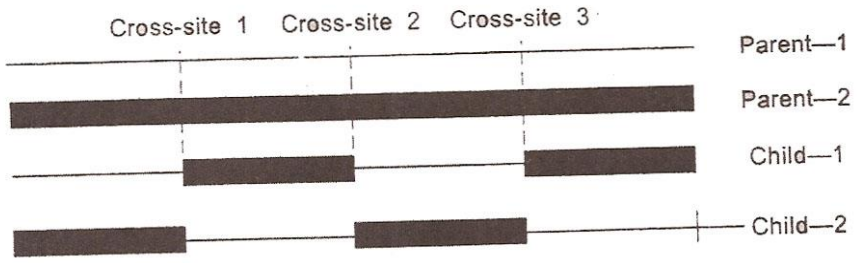
در روش چند نقطه ای ممکن است تعداد مکان ها زوج و یا فرد شود. در حالتی که تعداد زوج باشد، رشته ها به صورت یک حلقه ای که بدون ابتدا و انتها است، خواهند بود. این مکان ها به صورت تصادفی در اطراف دایره انتخاب شده اند.بنابراین اطلاعات بین این مکان ها به صورت شکل زیر جا به جا می شود.

WikiPower.ir



شکل ۱٤

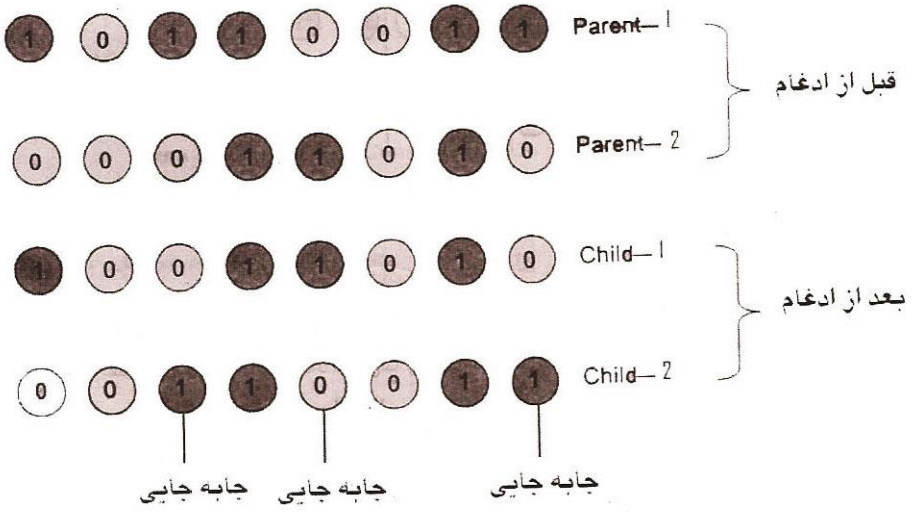
اما اگر تعداد آنها فرد با شد یک مکان متفاوتی در ر شته در نظر گرفته شده و اطلاعات انها به صورت شکل 15 جا به جا می شود:



شکل ۱۵

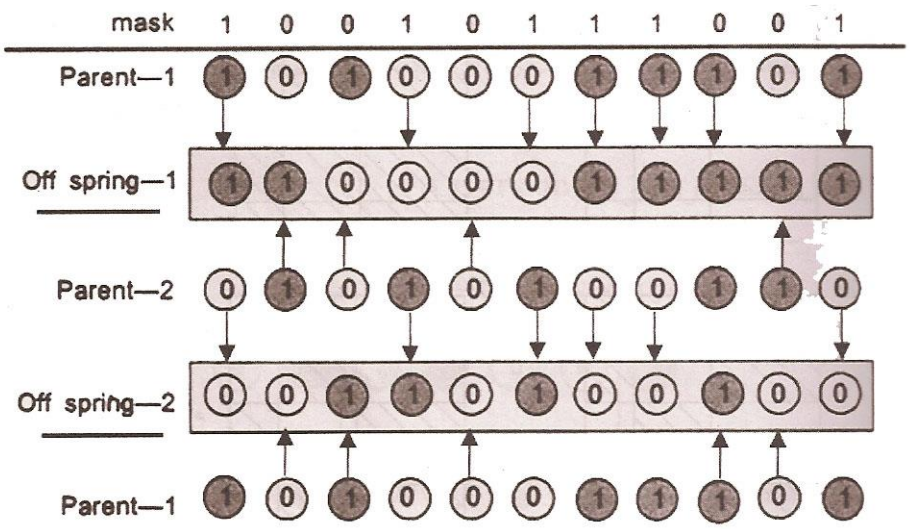
٤-۲-٤ روش ادغام یکنواخت (Uniform Cross Over)

این روش حالت توسعه یافته ی روش چند نقطه ای است. در این روش هر بیت بر اساس یک احتمال پنجاه درصدی از والد هایشان انتخاب شده و جا به جا می شوند این عمل در شکل نشان داده شده است:



شکل ۱۶

گاهی اوقات ژن های فرزندان با کپی شدن ژن های والدین آنها و بر اساس یک الگوی ادغام تصادفی به نام Mask تولید می شوند. هنگامی که در Mask مقدار یک (1) با شد زن از والد اول و اگر مقدار صفر (0) باشد ژن از والد دوم کپی می شود. این عمل در شکل 17 نشان داده شده است.



شکل ۱۷

از آنجایی که مقدار به صورت تصادفی برای هر جفت از والدها تولید می شود، در نتیجه فرزندان یک ترکیبی از ژن های والد خود خواهند بود. تعداد نقاط مؤثر ثابت نبوده و به صورت میانگین در حدود (طول کروموزوم می باشد) است.

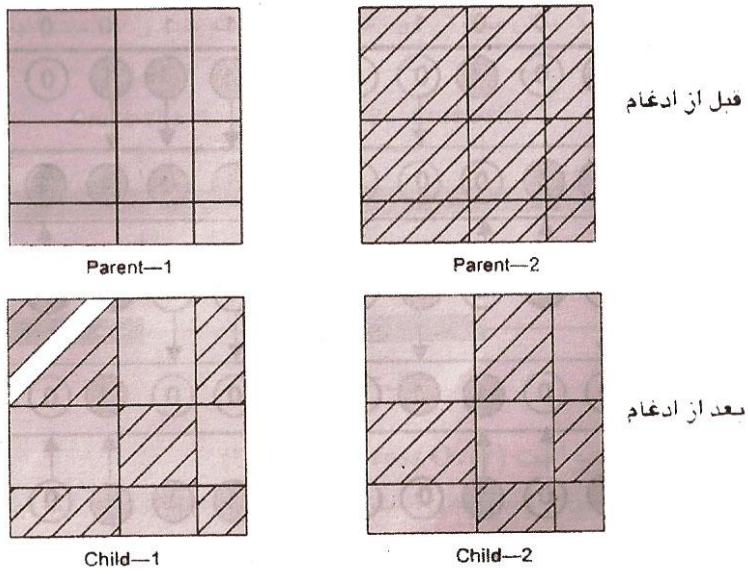
٥ – ٢ – ٤ روش ادغام دو بعدی (Matrix Cross Over)

معمولاً رشته ها را به صورت یک آرایه (یک بعدی) نمایش می دهند. مثلاً در شکل زیر دو زیر رشته به طول 4 بیت به یکدیگر متصل شده اند تا یک رشته را تولید کنند.

String – 1 1 0 1 1 1 0 0 1

1 1 1 0	0 1 0 1	String – 2
substring – 2	Substring – 1	
زیر رشته 2	زیر رشته 1	

اما اگر ر شته ها به صورت دو بعدی با شند از ماتریس ا ستفاده می کنیم که دو موقعیت ت صادفی در سطر و ستون ماتریس انتخاب می شود. عمل ادغام در شکل 18 نشان داده شده است:



شکل ۱۸

این امر باعث تق سیم سطر و ستون به لایه ی افقی و عمودی می شود که در نهایت با انتخاب هر ناحیه از این لایه های عمودی یا افقی و جا به جایی مقادیر آنها با جمعیت های موجود به مقدار مورد نظر دســت می یابیم.

توجه : بر اساس نظرDeb (1995) نمی توان گفت که کدامیک از روش های ادغام بهتر است. بنابراین انتخاب یک روش ادغام مناسب با توجه به سلیقه ی افرار و شرایط مسئله به دلخواه صورت می گیرد.

۳ – ٤ نرخ ادغام (Cross Over - Rate)

در اینجا لازم است به یک مفهوم مهم دیگری به نام نرخ ادغام (Cross Over - Rate) اشاره کنیم. طبق تعریف نرخ ادغام بیانگر احتمال ادغام است که آن را با pc نشان می دهند و مقدار آن بین 0 تا 1 است. این نرخ درGA با پیدا کردن نسبت تعداد جفت های ادغام شده در جمعیت های ثابت به دست می آید و با فرض احتمال ادغام Pc می توان گفت که 100 Pc در صد از ر شته ها در عملیات ادغام به کار رفته اند و 100(1 – Pc) درصد از جمعیت باقی مانده است.

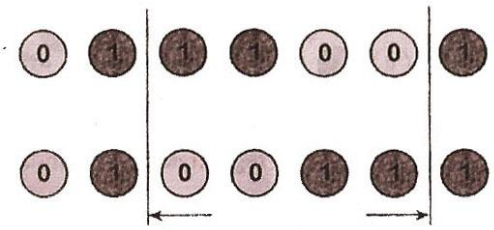
طبق تعریف ساده ی دیگر (Haupt) این نرخ بیانگر تعداد کروموزوم هایی است که وارد Mating Pool شده اند.

هر چقدر این مقدار بیشتر باشد، یعنی کروموزوم های جدید و زیادتری وارد Mating Pool شده اند.

اما اگر این مقدار خیلی زیاد شود باعث می شود تا فرصت تطابق در کروموزوم از دست برود و همچنین اگر این مقدار خیلی کم باشد، تعداد فرزندان تولید شده کافی نخواهد بود. این نرخ برای جمعیت هایی با اندازه ی 30 تا 200 مورد در محدوده ی 0.5 تا 1 خواهد بود.

٤ – ٤ عمل معکوس سازی (Inversion)

در این عمل یک رشته از داخل جمعیت انتخاب شده و به صورت تصادفی دو نقطه از آن انتخاب می شود سپس همه ی بیت های بینابین دو نقطه معکوس می گردد.این عمل در شکل زیر نشان داده شده است :



شکل ۱۹

بیت‌های بین دونقطه معکوس شده است.

روش های مختلفی در معکوس سازی رشته ها وجود دارد که برخی از آنها عبارتند از :

۱ – ٤ – ٤ روش معکوس سازی از انتها و خطی (Linear + End Inversion)

در روش معکوس سازی خطی (Linear inversion) با احتمال مشخص 0.75 عمل معکوس سازی انجام می شود. اما در معکوس سازی از انتها (End inversion) با احتمال مساوی 0.125 از سمت چپ و راست رشته عمل معکوس سازی صورت می گیرد.در این روش یکی از نقاط چپ یا راست به عنوان نقطه ی اول و نقطه ی دوم به صورت تصادفی (که معمولاً بیشتر از نصف طول رشته نیست) انتخاب می شوند. روش معکوس سازی از انتها باعث می شود تا تمایل معکوس سازی خطی برای انتخاب نقاط وسط رشته به نقاط انتهایی نیز معطوف گردد.

۲ –۴ – ۴ روش معکوس سازی پیوسته (Continuous Inversion)

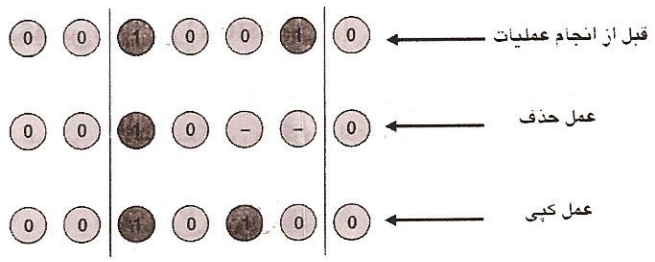
درهنگام به وجود آمدن کروموزوم های جدید عمل معکوس سازی برای هر کدام از آنها با احتمال معکوس سازی مشخصی برابر pr صورت می گیرد.

۳ – ٤ –٤ روش معکوس سازی دسته جمعی (Mass Inversion)

در این روش تا زمانی که یک جمعیت جدیدی به وجود نیاید، عمل معکوس سازی صورت نمی گیرد و پس از به وجود آمدن کامل آن نیمی از این جمعیت تحت عمل معکوس سازی قرار می گیرند.

٤ – ٥ عمل حذف و کپی (Deletion And Duplication)

در این عمل دو یا سه بیت به صورت تصادفی برای حذف شدن انتخاب شده و بیت های قبلی آنها کپی می شود این عمل در شکل زیر نشان داده شده است :

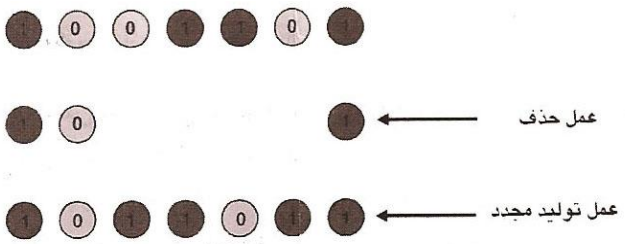


شکل ۲۰ : عمل حذف و کپی

WikiPower.ir

٤ – ٦ عمل حذف و تولید مجدد (Deletion and Regeneration)

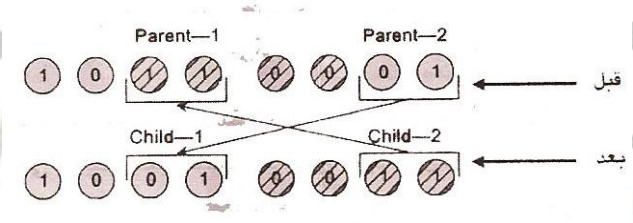
در این عمل ژن های بین دو نقطه حذف شده و به صورت تصادفی مجدداً تولید می شوند. این فرآیند در شکل زیر نشان داده شده است :



شکل ۲۱

۷ – ٤ عمل جداسازی (Segregation)

مطابق شکل، ابتدابیت هایی از والد جدا شده و سپس عمل ادغام برای تولید فرزندان جدید صورت می گیرد:

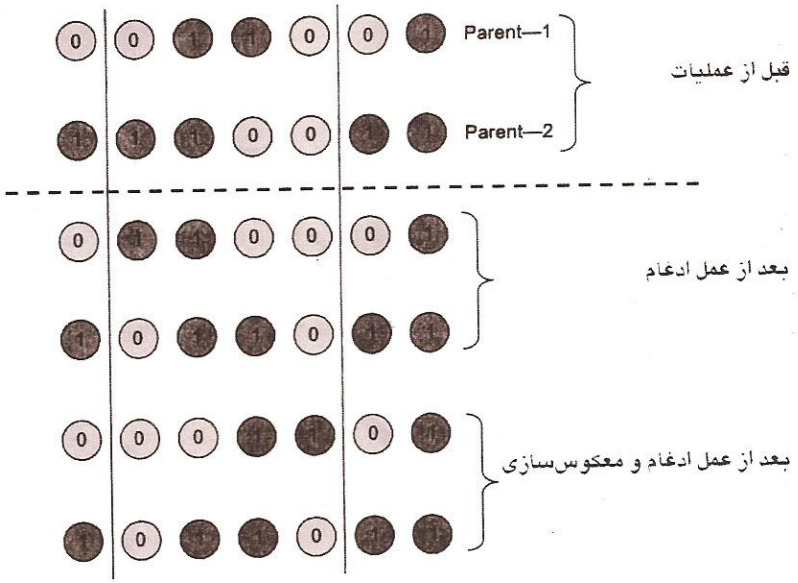


شکل ۲۲

۸ – ٤ عمل ادغام و معکوس سازی (Cross Over And Inversion)

این عمل ترکیبی از عملیات ادغام و معکوس سازی ا ست. در این عمل دو نقطه به صورت تصادفی انتخاب شده و محتوای بین این دو نقطه جا به جا می شود. سپس عمل معکوس سازی بر روی آنها صورت می گیرد. شکل زیر این عملیات را نشان می دهد :

برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه



شکل ۲۳

[1]

WikiPower.ir

فصل پنجم : جهش

۵ – ۱ عمل جهش یا موتاسیون (Mutation)

پس از عمل ادغام رشته ها، نوبت به عمل جهش یا موتاسیون **(Mutation)** می رسد.

عمل جهش یک بیت شامل تبدیل عدد صفر (0) به یک (1) و بالعکس است که بر اساس یک احتمال کوچک مثل pm به صورت بیت به بیت صورت می گیرد. عمل جهش به این ترتیب است که یک عدد تصادفی بین صفر (0) تا یک (1) تولید می شود. اگر عدد تولید شده کوچکتر از pm باشد مقدار خروجی را برابر در ست (True) وگرنه برابر غلط (False) در نظر می گیریم. اگر برای هر بیت مقدار خروجی در ست باشد بیت تغییر می کند و گرنه بیت بدون تغییر باقی خواهد ماند.

بیت های یک رشته به صورت مستقل جهش می یابند، به این معنی که جهش یک بیت بر روی احتمال سایر بیت ها تأثیر نمی گذارد. ایت عمل در یک الگوریتم ژنتیک ساده به منزله ی یک عملگر ثانویه و به منظور حفظ اطلاعاتی که در حال از د ست رفتن ا ستتلقی می گردد. برای مثال، فرض کنید مقادیر بیت های رشته های یک جمعیت در یک محدوده ی خاص برابر صفر شده است و حل بهینه نیاز به یک عدد یک (1) در آن نقطه دارد. این در حالی است که عملگر ادغام نمی تواند عدد یک را در آن موقعیت تولید کند. بنابراین برای تولید عدد یک (1) از عمل جهش یا موتاسیون استفاده می کنیم.

این عمل برای جلوگیری از همگرایی سریع و کمک به الگوریتم جستجو برای فرار از به دام افتادن در مینیمم های مو ضعی مفید ا ست. از سوی دیگر این عمل برای حفظ حالت متفاوت و متمایز بودن کروموزوم ها در یک جمعیت به کار می رود. به عنوان مثال، جمعیت های زیر را در نظر می گیریم :

↓

0110 1011

برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

0011 1101

0001 0110

0111 1100

↑

همان طور که ملاحظه می شود، همه ی بیت های سمت چپ این رشته ها صفر است. اگر راه حل بهینه نیاز به یک بیت در این موقعیت دا شته با شد، در این صورت هیچکدام از عملیات های ادغام و تولید مثل قادر به این کار نخواهند بود اما عمل جهش این کار را برای ما انجام خواهد داد و جمعیت پس از این عمل با توجه به یک احتمال مشخص به صورت زیر در می آید :

0110 1011

0011 1101

0001 0110

→1111 1100

WikiPower.ir

نتیجه : به طور خلاصه عمل جهش باعث جنبش و حرکت در فضای جستجو و ذخیره ی اطلاعات از دست رفته جمعیت می شود.

۵ – ۲ نرخ جهش (Mutation Rate – Pm)

این نرخ بیانگر احتمال جهش است که بر اساس تعداد بیت های جهش یافته به دست می آید.

این احتمال مقدار کوچکی ا ست که معمولاً برای یک جمعیت با اندازه ی 30 تا 200 عضو بین 0.001 تا 0.5 خواهد بود.

۵ – ۳ عملگر های بیتی (Bit – Wise Operator)

کد باینری به عنوان یکی از پرکاربردترین کدهای الگوریتم ژنتیک محسوب می شود.کار کردن بر روی کدهای باینری نیازمند کار با بیت ها ا ست که در زبان هایی مثل C++ و برنامه ی Matlab امکان کار بر روی تک تک بیت ها توسط عملگرهای بیتی (Bit – Wise Operator) وجود دارد.

گاتفرید (Bayron S . Gatfried) این عملگرها را در سه دسته تقسیم بندی می کند :

۱ – عملگرهای مکمل یک (Onse Complement - Operator)

۲ – عملگرهای منطقی (Logical - Operator)

۳ – عملگرهای شیفت (Shift – Operator)

۱ – ۳ – ۵ عملگر مکمل یک (Onse Complement - Operator)

عملگر مکمل یک (~) یک عملگر تکی است که باعث می شود بیت های عملوند معکوس شود. یعنی 1 به 0 و 0 به 1 تبدیل شود.

مثلاً در مسئله (8 - 2) برای دو مقدار دلخواه θ۱, θ۲ عمل مکمل یک به صورت زیر خواهد بود:

	θ۱	θ۲
a = 0100 0001 →	4 1	24 16
~a = 1011 1110 →	11 14	66 84

۲ – ۳ – ۵ عملگرهای منطقی (Logical - Operator)

سه عملگر منطقی عبارتند از :

۱ – عملگر AND (&)

۲ – عملگر OR (|)

۳ – عملگر XOR (^)

هر کدام از این عملگرها نیاز به دو عملوند دارند و به صـــورت بیت به بیت عمل می کنند. گاهی از این عملگرها به جای عمل ادغام (Cross Over) نیز استفاده می شود.

جدول درسـتی (Truth Table) این عملگرها در جدول 9 نشـان داده شـده اسـت. مقادیر a , b همان عملوندهای ما هستند.

جدول – ۹

a	b	AND '&' a & b	ExclusiveOR a ^ b	OR . | . a | b
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	0	1

۱ – ۲ – ۳ – ۵ عملگر AND (&)

در عملگر AND اگر هر دو بیت یک با شند مقدار یک را بر می گرداند و گرنه مقدار صفر برگ شت داده می شود.

برای مثالی از عملگر AND بر روی دو والد (مطابق جدول 9) داریم :

Parent 1a = 1010 1010 → 10 10

Parent 2b = 1100 0011 → 12 3

Parent a&b= 1000 0010 → 8 2

۲ – ۲- ۳- ۵ عملگر OR (|)

عملگر OR اگر یکی از بیت های آن یک باشـــد مقدار یک را بر می گرداند و گرنه مقدار صـــفر را بر می گرداند.

Parent 1a = 1010 1010 → 10 10

Parent 2b = 1100 0011 → 12 3

Parent a | b= 1110 1011 → 13 11

WikiPower.ir

۳- ۲- ۳- ۵ عملگر XOR (^)

عملگر XOR مقدار یک را بر می گرداند اگر یکی از بیت های آن یک و بیت دیگر صفر با شد و گرنه مقدار صفر را بر می گرداند.

مثال :

Parent 1a = 1010 1010 → 10 10

Parent 2b = 1100 0011 → 12 3

Parent a^ b= 0110 1001 → 6 9

۳ – ۳ – ۵ عملگرهای شیفت (Shift - Operator)

دو عملگر شیفت عبارتند از :

۱ – شیفت به چپ (»)

۲ – شیفت به راست («)

هر کدام از این عملگرها بر روی یک متغیر عمل می کنن، اما نیاز به دو عملوند دارند. عملوند اول عدد صحیحی است که عمل شیفت بر روی آن انجام می شود و عملوند دوم بیانگر تعداد شیفت است.

۱ – ۳ – ۳ – ۵ عمل شیفت به چپ (»)

این عمل باعث شیفت همه ی بیت های یک عملوند به تعداد عملوند دوم به سمت چپ می شود. چپ ترین بیت ها از بین رفته و از سمت راست صفر وارد می شود.

مثال :

a = 1010 0110 → 10 6

a «2 = 1001 1000 → 9 8

←

WikiPower.ir

۲ – ۳ – ۳ – ۵ عمل شیفت به راست («)

این عملگر باعث شیفت همه ی بیت های یک عملوند به تعداد عملوند دوم به سمت راست می شود. سمت راست ترین بیت ها از بین رفته و از سمت چپ صفر وارد می شود.

a = 1010 0110 → 10 6

a» 2 = 1001 1000 → 2 9

٤ -٥ استفاده از عملگر های بیتی در GA

عملگر های منطقی به صورت ترکیب های متفاوتی به کار می روند که معمولاً از یک جفت از آنها استفاده می شود.

مثلاً یک جمعیت به صورت تصادفی برای جفت گیری (Mating) یا ادغام انتخاب شده و بر روی هر جفت از بیت های آن عمل AND و عمل OR صورت می گیرد.

به طور مشابه عمل های AND و XOR و یا عمل های OR وXOR نیز می توانند برای تولید فرزندان و تولید نسل بعدی به کار رود.

WikiPower.ir

٥ – ٥ چرخه ی تولید نسل (Generation Cycle)

شکل 24 یک چرخه ی تولید نسل در GA با جمعیت چهاتایی (P1 = 4) و رشته هایی به طول 10 بیت را نشان می دهد.

در این مثال، تابع هدف (هزینه) برابر تعداد یک ها در رشته است که یک عددی بین صفر تا ده است. مقدار تابع Fitness از تقسیم مقدار تابع هدف بر 10 به دست می آید تا یک مقدار نرمال بین 0 تا 1 حاصل شود. بنابراین چهار رشته ی مفروض در این مسأله مقادیرFitness ای برابر با 0.3 و 0.6 و0.6 و 0.9 خواهند داشت.

اگر مقدار میانگین برابر 0.6 باشد

F(Average) = (0.3 + 0.6 + 0.6 + 0.9) / 4 = 0.6

مقدار F / F (av) (انتخاب نسبی) به ترتیب برابر خواهد بود با :

(0.3 / 0.6) 0.5 , (0.6 / 0.6) 1 , (0.6 / 0.6) 1 , (0.9 / 0.6) 1.5

بر اساس نظریه ی حیات بهترین ها بهترین موارد دو کپی، موارد ضعیف حذف (می میرند) شده وبقیه هر هر کدام یک کپی خواهند داشت. از این رو برای رشته هایی با Fitness برابر 0.5 صفر کپی و 1.5 دو کپی و بقیه ی موارد یک کپی در نظر گرفته می شود. که این امر در جمعیت P2 نشان داده شده است.

پس از این مرحله، رشته ها به صورت تصادفی برای انجام عمل ادغام دو به دو جفت می شوند. مثلاً رشته ی 1 و 4 یک جفت و رشته های 2 و 3 نیز جفت بعدی را شکل می دهند. با فرض نرخ ادغامی برابر با 0.5 جفت های 2 و 3 دست نخورده باقی می مانند. نقطه ی ادغام بین بیت های 1 تا 5 انتخاب شده که مقادیر بین آنها در دو رشته جا به جا می شود.

عمل موتاسیون یا جهش جمعیت P3 در جمعیت P4 نشان داده شده است. بیت های جهش یافته عبارتند از بیت ششم از رشته ی 2 و بیت اول از رشته ی اول. نرخ احتمال جهش برابر با 0.05 در نظر گرفته شده است.

جمعیت P4 نسل تولید شده ی بعدی را نشان می دهد. در این چرخه جمعیت های P2 وP3 به عنوان مراحل میانی در چرخه ی تولید نسل محسوب می شوند.

پارامترهایی مثل اندازه ی جمعیت (Population Size) ، نرخ جهش (Mutation Rate) و نرخ ادغام (Cross Over Rate) به عنوان پارامترهای کنترلی (Control Parameter) هستند که باید قبل از اجرای الگوریتم مشخص شوند.

برای خاتمه دادن به الگوریتم نیاز به یک شـــرط خاتمه داریم. این شـــرط می تواند پس از تولید تعداد ثابتی نسل و یا رسیدن به یک حداکثر مقدار Fitness مشخص شود و یا پس از اینکه رشته های جمعیت به یک درجه ی همگنی خاصی رسیدند اتفاق بیفتد. همه ی مراحل فوق در شکل زیر نشان داده شده است :

Population P1

String	f = Fitness	$f/f(av)$	Copy
00000 11100	0.3	0.5	0
10000 11111	0.6	1.0	1
01101 01011	0.6	1.0	1
11111 11011	0.9	1.5	2

جمعیت اول (P1)

Population P2 reproduction نقطه ۱ نقطه ۲

String	f = Fitness	Cross over	CS1	CS2
10000 11111	0.6	4	1	5
01101 01011	0.6	–		
11111 11011	0.9	–		
11111 11011	0.9	1	1	5

جمعیت دوم (P2) پس از تولید مثل

Population P3 cross over

String	f = Fitness
11111 11111	1.0
01101 01011	0.6
11111 11011	0.9
10000 11011	0.5

جمعیت سوم (P3) پس از ادغام

Population P4 mutation

String	f = Fitness
01111 11111	0.9
01101 11011	0.7
11111 11011	0.9
10000 11011	0.5

جمعیت چهارم (P4) پس از جهش

شکل ۲٤

[1]و [4]

فصل ششم : مفاهیم تکمیلی

۱ - ٦ همگرایی الگوریتم ژنتیک (Convergence)

تاکنون تحقیقاتی برای بررسی همگرایی GA بااستفاده از زنجیره ی مارکوف(Markov Chain) صورت گرفته اســت(Nix-Vose-1992).این بررســی ها برروی جمعیت های بزرگ ونرخ موتاســیون پایین به صورت آماری انجام شده است.برای مثال آنها نشان دادند که همگرایی با رسیدن به یک اندازه ی(جمعیت) بحرانی حاصل می شود.

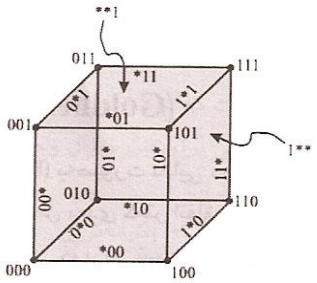
مطالعات دیگری دربارهی پیدا کردن یک حد بالا برای تعداد تکرارها نیز صــورت گرفته اســت(Aytug Greenhalgh 2000وKoehler 1996) اما به طور کلی یک اثبات صریح و جامع ریا ضی درباره ی همگرایی وجود ندارد.

یک معیار دیگر همگرایی را می توان به این صورت بیان کرد که وقتی یک در صد ثابتی از سطر و سطون های ماتریس جمعیت شبیه هم می شوند،میتوان فرض کرد که همگرایی صورت گرفته ا ست این در صد ممکن است ٪ 80 یا 85% باشد.

اما یک روش قدیمی بررسی همگراییGAبا استفاده از قضیه ی الگو(Schema Theorem)است(Holland 1975).

طبق تعریف:یک الگو رشته ای از کاراکترهای صفر(0) و (1) و بی اهمیت (don't care)(*) است.

مثلا الگوی 11**00 که دو بیت وسط آن می تواند صفر یا یک باشد،بیانگر یکی از چهار رشته 110000,111000,110100و111100است.شکل 27 یک الگو را برای یک عدد باینری سه رقمی نشان می دهد.تعداد نقاط ثابت در یک الگو را مرتبه ی آن الگو میگویند (مثلا000**از مرتبه سه است).در این شکل عدد سه رقمی با مرتبه ی سه گوشه های مکعب,مرتبه ی دو ضلع های مکعب و مرتبه ی یک وجوه مکعب را نشان می دهند.



شکل ۲۷

مقدار میانگین Fitness ر شته را به الگو ارتباط می دهیم. که این مقدار میانگین Fitness برای یک الگو با توجه به ترکیب یک جمعیتاز یک نسل به نسل بعدی تغییر می کند.

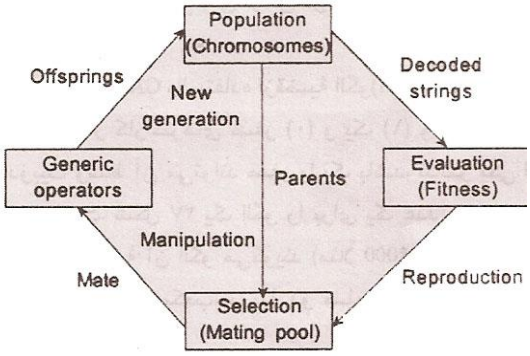
با توجه به ج ستجو برای ر شته های بهینه رقابت الگوها باعث افزایش تعداد موارد در جمعیت می گردد. اگر رشـــته ی بهینه را با توجه به موقعیت الگو با طول کوتاه و میانگین Fitness بالا فرض کنیم، آنگاه نتیجه یک ســـزی رشـــته های بهینه خواهد بود. به چنین الگوهایی که دارای مقدار Fitness بالا و طول کوتاه است، اصطلاحاً بلوک های سازنده Building Blocks گفته می شود. هنگامی که عملگرهای GA را بر روی یک جمعیتی از ر شته ها اعمال کنیم، تعدادی از این بلوک ها در طول ر شته به وجود می آیند. برر سی ها نشان می دهد که الگوهای به وجود آمده ی نامناسب از بین می روند (می میرند).

اما خطری که در اینجا وجود دارد، این ا ست که عملگر ادغام ممکن ا ست الگوهای خوب را هم از بین ببرد. بنابراین انتخاب یک عملگر ادغام مناسب نقش حیاتی را در این زمینه ایفا می کند.

معمولاً در کاربردهای فنی GA از ادغام دو نقطه ای (Two Point Cross Over) یا تک نقطه ای (Single) استفاده می شود.

ا سا ساً معنی ج ستجو در ف ضای ژنتیک تو سعه ی بلوک ها (Building Blocks) ا ست. این بلوک ها با توجه به ترکیب عملگرهای ژنتیک با یکدیگر ترکیب شـــده تا بلوک های بزرگتری و بهتری را بســـازند و سرانجام منجر به همگرایی به یک ناحیه ی بهینه شوند.

به طور خلاصه یک چرخه ی GA را می توان به صورت شکل 28 نشان داد :



شکل ۲۸

۲ – ٦ قضیه ی الگو : (Goldberg - 1989)

الگوهای بهتر (نسبت به میانگین Fitness) به صورت نمایی در نسل بعدی بیشتر اتفاق می افتند و الگوهایی با بدترین مقدار نسبت به میانگین در نسل بعدی کمتر اتفاق خواهد افتاد.

بر این اساس الگوهای مناسب به حیات خود در نسل های بعدی ادامه می دهند. بهترین الگوها در زمان حیات GA یک تخمینی از نرخ همگرایی می باشد.

بنابراین اگر یک الگوی فرضی را با S در جمعیتی از نسل t نشان دهیم ؛ تعداد نمایش این الگو در نسل $t + 1$ برابر است با :

$$S_{t+1} = S_t \; P_t \; (1 + Q_t) \; R_t$$

Pt : احتمال انتخاب یک الگو برای حیات در نسل بعدی.

Qt : احتمال انتخاب یک الگو برای جفت گیری

Rt : احتمال اینکه یک الگو توسط عملگرهای ادغام و موتاسیون از بین نرود.

باید توجه داشت که اگر $P_t (1 + Q_t) < 1$ باشد، سرانجام این الگو از بین می رود.

حیات الگو بر اساس فرمول زیر خواهد بود :

برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

$$St+1 \geq (Pt\ Pt-1\ \ldots.\ P1\ )[(1 + Qt\ )(1 + Qt-1\ )\ldots(1 + Q1\ )](Rt\ Rt-1\ \ldots.R1\ )\ S1$$

این فرمول احتمال انتخاب یک الگو را که از یک نسل با نسل دیگر متفاوت است، نشان می دهد.

نسل اولیه الگوی برابر با $Pt\ (1 + Qt) > 1$ دارد که ممکن است در نسل های بعدی به مقدار $Pt\ (1 + Qt) < 1$ برسد و همان طور که اشاره شد، در این حالت برای مدت زیادی زنده نخواهد ماند.

تئوری الگو برای عملگرهای انتخاب و ادغام تک نقطه ای (Single Point) از فرمول زیر به دست می آید

$$St+1 \geq st\ (f / f)\ \{1 – [Pc\ \delta\ / (Nbits\ - 1)] – \xi\ pm\ \}$$

WikiPower.ir

که :

f : میانگین Fitness کروموزوم های شامل الگو.

f : میانگین Fitness جمعیت.

Pc : احتمال ادغام.

Pm : احتمال جهش (موتاسیون).

δ : فاصله بین اولین و آخرین بیت الگو.

Nbits : تعداد مکان های ادغام.

ξ : تعداد رقم های تعریف شده در الگو.

۳ – ٦ شرط توقف الگوریتم

سؤال مهمی که در اینجا وجود دارد، این است که چه زمانی باید الگوریتم متوقف شود ؟

باید اذعان کرد که هیچ جواب جامع و مناسبی برای این سؤال وجود ندارد.

اما بعضی از شرط های توقف متداول عبارتند از :

۱ – رسیدن به جواب : این ساده ترین و کوته فکرانه ترین روش است. به این معنی که اگر مقدار کروموزوم مناسب ترین باشد، الگوریتم متوقف می شود.

۲ – عدم پیشرفت : یعنی الگوریتم GA پس از x بار تکرار با همان کروموزوم های قبلی ادامه پیدا کند. چه اینکه الگوریتم به جواب بهینه رسیده باشد یا در یک مینیمم موضعی گیر افتاده باشد.

۳ – به روش اماری : اگر مقدار تابع Fitness به یک مقدار مشخصی رسید، الگوریتم متوقف می شود.

٤ – تعداد تکرارها (Iterations) : اگر با هیچ کدام از موارد فوق جواب نداد، شرط توقف را تعداد تکراهای مشخصی قرار می دهیم.

WikiPower.ir

٥ – بهینه ساز موضعی (Local Optimizer) : از یک بهینه ساز موضعی استفاده می کنیم که در صورت عدم پیشرفت متوقف می شود.

نکته ی مهم : اگر الگوریتم به یک حل بهینه همگرا نشود، باید پارامترهای ان را (مثل اندازه ی جمعیت ، نرخ جهش ، نرخ ادغام و امثال آن را) دوباره تغییر دهیم.

٤ – ٦ بهینه سازی چند مرحله ای

بهینه سازی سیستم هاییکه شامل چندین متغیر هستند، در فضاهای جستجوی بزرگ یکی از مشکلات مهم و اساسی به شمار می آیند.

در این مسائل با اضافه شدن متغیرها و فاکتوهای اضافی به مسئله ی فضای جستجو به صورت نمایی رشد می کند. این روش بهینه سازی چند مرحله ای توسط Erbature (2000) مطرح گردید. هدف از این روش کاهش فضای جستجو برای متغیرها در هر مرحله ای از بهینه سازی است. در این روش یک بهنه سازی اولیه به عنوان مرحله ی اول بهینه سازی با استفاده از یک مجموعه ی گسسته از مقادیر صورت می گیرد. در این فازف الگوریتم به صورت اتوماتیک، مجموعه ی گسسته ی اصلی را به چند زیر مجموعه تقسیم می کند.

فرایندی که برای تولید این زیر مجموعه ها به کار می رود عبارتست از :

۱ – مقادیر به دست امده در مرحله ی اول بهینه سازی به عنوان مقادیر اصلی در نظر گرفته می شود.

۲ – مجموعه ی گسسته ی اصلی به چند زیر مجموعه ی مساوی تقسیم می شود.

۳ – متغیرهای مسئله با توجه به مقادیر به دست آمده به زیر مجموعه های مناسب تخصیص می یابند.

٤ – زیر مجموعه ها گسترش می یابند.

این فرآیند مانع از قرار گرفتن در یک بهینه ی موضع خاص می شود. در مرحله ی دوم بهینه سازی متغیرها از زیر مجموعه های کوچکتری استفاده می کنند. از این رو مرحله ی دوم در مناطق بیشتری از فضای جستجو عمل می کند.

ویژگی های بهینه سازی چند مرحله ای عبارتند از :

۱ – **نخست :** اینکه فرآیند بهینه سازی را به سمت راه حل های بهینه و نقاط دلخواه و مناسبی از فضای جستجو هدایت می کند. بنابراین هر مرحله از بهینه سازی را می توان یک گام به سمت جواب تفسیر کرد. همچنین این روش در هر فضای جستجو که مقادیر بهینه نزدیک هم باشند به خوبی عمل می کند.

۲ – **دوم :** از آنجایی که ظرفیت مجموعه ی گسسته ثابت است، هر زیر مجموعه ای که از تقسیم مجموعه های گسسته به دست می آید، باید کمتر از ظرفیت در نظر گرفته شده باشد.

تجربه نشان می دهد که دو یا سه مرحله ی بهینه سازی برای همگرایی به مقدار بهینه در مسائل بهینه سازی گسسته یا پیوسته کافی است.

۵ – ٦ برخی از مفاهیم تکمیلی و پیشرفته در GA

روشهای جدید در عملگرهای ادغام باعث شده است که امکان جستجو در طول متغیر نیز ممکن شود. فرض کنیم xi^(k) , xi^(y) مقدارهای متغیر طراحی شده x دو ر شته والد k و j با شند.عمل ادغام بین این دو مقدار باعث تولید مقادیر جدیدی مثل :

$$Xi^{new} = (1 - \lambda)\, xi^{(j)} + \lambda\, xi^{(k)}$$

خواهد شد که پارامتر λ یک عدد ت صادفی بین صفر تا یک ا ست. این معادله مقادیر جدید دو والد را تولید می کند که برای هر یک از متغیرها محا سبه می شود. این روش ادغام دارای احتمال یک سانی در تولید یک نقطه درون ناحیه محصور بین دو والد است.

٦ – ٦ بهینه سازی چند تابعی (Multi – Objective Optimization)

در این بهینه سازی (Moo) چندین تابع هدف وجود دارد. که این چند تابع را به یک تابع هدف تبدیل می کنیم. این عمل معمولاً با استفاده از فرول زیر صورت می گیرد :

$$Q = w1\, f1 + w2\, f2 + \ldots + wn\, fn$$

در این فرمول wها ، وزن ها و fها توابع ما هســـتند. این معادله یکی از روش های تبدیل چند تابع به یک تابع تکی اســـت. بنابراین حل مســـائل چند تابعی یعنی حل چندین تابع با وزن های مختلف. این جواب ها گاهی با عنوان جواب های بهینه Pareto نیز شناخته می شود.

۶-۷ الگوریتم ژنتیک پیوسته (Continuous – Genetic Algorithm)

تا کنون بیشتر از کدگذاری باینری استفاده شد. اما در مسائل عملی ممکن است مقادیر متغیرهر به صورت پیوسته باشند و تبدیل آنها به باینری مناسب نباشد، زیرا باعث می شود تا اندازه ی کروموزوم ها و جمعیت بسیار بزرگ شود.

در این شـرایط از اعداد ممیز شـناور (Floating - Pint) برای نمایش کدها اسـتفاده می کنیم. مزیت این روش فضای ذخیره سازی کمتر و سرعت بیشتر نسبت به روش باینری است.

۶ – ۸ الگوریتم ژنتیک موازی (Parallel Genetic Algorithm)

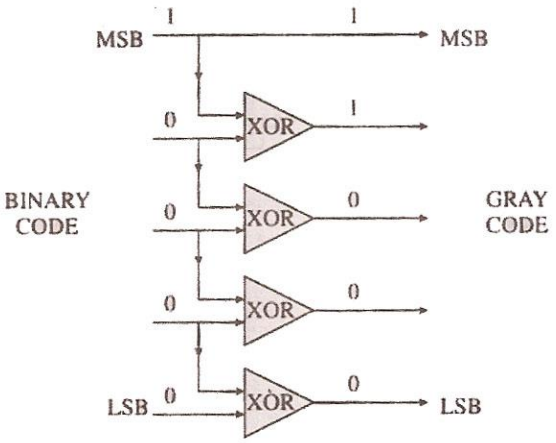
معمولاً در مسائل طراحی مهندسی با توابع هدف یا هزینه ی (Cost – Objective Function) بسیار پیچیده برخورد می کنیم که بار زیادی برای CPU دارند. از ســـوی دیگر امکان پیاده ســـازی الگوریتم های ژنتیک به صورت موازی برای پردازش توابع به صورت همزمان وجود دارد.

این روش باعث افزایش سرعت پردازش های بسیار سنگین می گردد.

۶ – ۹ کد گری (Gray Code)

همان طور که در ابتدا دیدیم ، روش های مختلفی برای کد کردن وجود دارد. یکی از این روش ها روش کد گری (Gray Code) است. که به خاطر طولانی شدن موضوع از شرح کاربردهای این کد و نحوه ی به

دســت آوردن آن خودداری می کنیم. شــکل 29 نحوه تبدیل کد باینری به کد گری را به صــورت کلی با استفاده از عمل XOR نشان می دهد.



شکل ۲۹

۱۰ – ٦ شباهت ها و تفاوت های GA با روشهای قدیمی بهینه سازی

همان طور که دیدیم GA با همه ی روش های بهینه ســازی متفاوت اســت. الگوریتم های ژنتیک با یک سری رشته های کدشده به جای متغیرها کار می کنند. مزیت کار با متغیرهای کد شده در این است که می توان توابع پیوســته را نیز به صــورت گســسته کد کرد. به عبارت دیگر، GA نیاز به مقادیری از تابع در نقاط گسسته دارد که این روش امکان حل بسیاری از مسائل گوناگون را میسر می سازد. از سویی عملگرهای GA از مشابهت های موجود در یک رشته برای جستجوی مؤثر استفاده می کنند. GA با جمعیتی از نقاط به جای یک نقطه ی تکی کار می کند. در GA امکان پیدا کردن راه حل های بهینه ی چندتایی وجود دارد که این امر موجب کاهش زمان و هزینه ی طراحی می گردد.

با وجود تفاوت های GA با روش های قدیمی یک سری شباهت هایی نیز وجود دارد. در روش های جستجوی قدیمی هنگامی که یک جهت جستجو برای پیداکردننقطه ی جدید مشخص می شد، حداقل دو نقطه به صورت صریح برای جهت جستجو تعریف می گردید.

از طرف دیگر دیدیم که در عملگر ادغام از دو نقطه برای تولید نقاط جدید ا ستفاده می شود. بنابراین عملگر ادغام مشابه روش جستجوی جهت دار است. با این تفاوت که جهت جستجو برای همه ی نقاط در جمعیت ثابت نیست. از انجایی که دو نقطه به صورت تصادفی در ادغام انتخاب می شوند جهت های مختلفی برای جستجو امکان پذیر است. عملگر تولیدمثل یک تأثیر غیرمستقیم در جهت جستجو داشته و به هدایت جستجو کمک می کند. هدف از عملگر جهش تولید یک نقطه در نزدیکی نقطه ی فعلی است.

جستجو در عملگر جهش مشابه روش جستجوی محلی و Hooke – jeeves است .

۱۱ – ٦ نکات مهم در هنگام کا با GA

برای کار با GA باید به این نکات توجه کنیم :

۱ – توجه به فاکتورهای اساسی که در پیاده سازی GA مهم است ، مثل

* نحوه ی نمایش.
* اندازه ی جمعیت و نرخ جهش.
* روش های انتخاب و حذف.
* عملگر های ادغام و جهش.

۲ – شرط خاتمه.

۳ – کارایی.

۱۲ – ٦ مزایای GA

بعضی از مزایای GA عبارتند از :

۱ – فهم آسان.

۲ – مجزا بودن و ماجولار بودن آن.

۳ – پشتیبانی از بهینه سازی چند تابعی (هدفی).

٤ – مناسب برای محیط های نویزی.

٥ – همیشه یک جواب داریم که این جواب با گذشت زمان بهتر می شود.

٦ – امکان استفاده به صورت توزیع شده و موازی.

۷ – روش های مختلفی برای افزایش سرعت و پیشرفت الگوریتم وجود دارد.

۸ – بهره برداری ساده از جواب های قبلی.

۹ – انعطاف پذیری در شکل دهی بلوک های سازنده (Building Blocks) برای کاربردهای ترکیبی.

۱۰ – وجود یک حافظه یا تاریخچه ی معتبر.

WikiPower.ir

چه موقعی از GA استفاده می کنیم :

GA در موارد زیر به کار می رود:

۱ – جواب ها بیش از اندازه پیچیده یا بی تحرک باشند.

۲ – نیاز به یک وسیله ی اکتشافی برای پیدا کردن راه های جدید داشته باشیم.

۳ – مسئله شبیه مسائلی که تاکنون در GA حل شده است باشد.

٤ – بخواهیم جواب های موجود را با هم پیوند بزنیم.

۱۳

زمینه	کاربرد
کنترل	خطوط انتقال گاز – پرتاب موشک – سیستم های تعادلی – رهگیری و ..
طراحی	طراحی هواپیمی – طراحی مدارات VLSI – شبکه های ارتباطی و ..
مدیریت و برنامه ریزی	برنامه ریزی تولید – زمان بندی – تخصیص منابع و ..
روباتیک	مسیر حرکت روبات ...
یادگیری ماشین	طراحی شبکه های عصبی – الگوریتم های طبقه بندی و ..
پردازش سیگنال	طراحی فیلتر و ..
سایر موارد	هنر و موسیقی– حل مسئله ی فروشنده ی دوره گرد– مسیر یابی در شبکه ها و..

٦-

کاربردهای GA

برخی از مهمترین کاربردهای GA در علوم مختلف عبارتند از :

WikiPower.ir

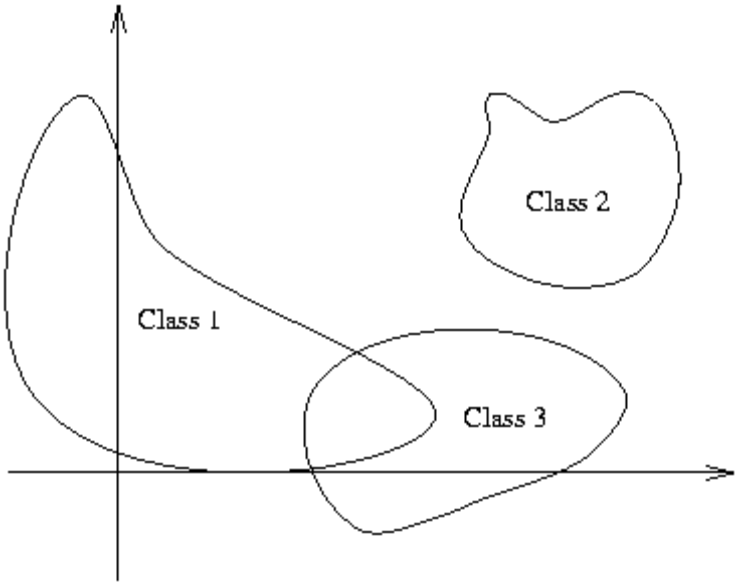
[1]و [3]

فصل هفتم : طبقه بندی کننده ها

۱ – ۷ طبقه بندی کننده ها

شکل تقسیم بندی مربوط است به هماهنگ نمودن یک سری از الگوهای ورودی و قرار دادن آنها در یک طبقه خاص. الگو ها از طریق برخی خصوصیات خاص شناخته شده و دسته بندی می شوند بنابراین کاملاً طبیعی است که آنها را تحت عنوان وکتورهای چند بعدی در نظر بگیریم در اینجا حرف «d» نشانگر تعداد خصوصیات (اشکال) است. این نوع نمایش داده ها مفهوم فضای اشکال را القاء می کند.

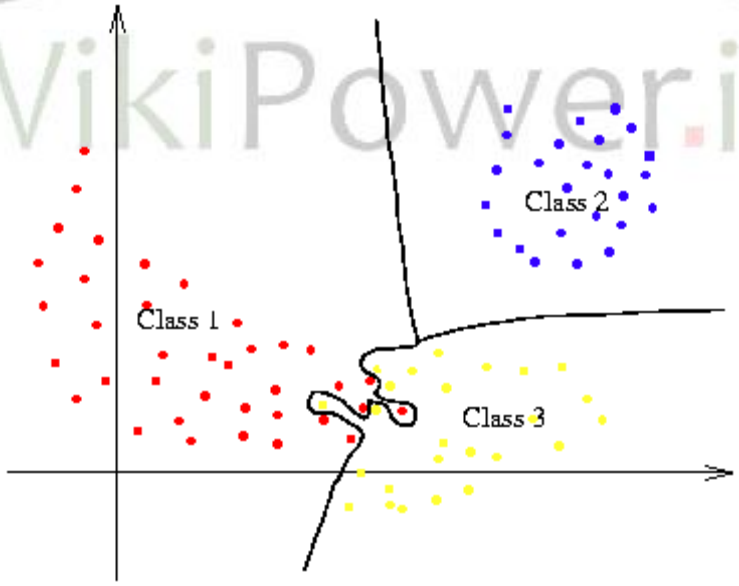
در این ابعاد «d» الگوها بصورت نقطه و طبقات بصورت زیر مجموعه نشان داده می شوند. در اینجا مسئله اصلی این است که تشخیص دهیم هر الگو (Pattern) در کدام طبقه قرار می گیرد. اگر طبقات مختلف با هم تداخل پیدا نکنند، گفته می شود که این طبقات کاملاً جداگانه و مجزاازهمدیگر هستند.درتصویریک شکل دو بعدی را مشاهده می کنیدکه دارای سه طبقه است.



دراینجا هدف ماتعیین یک تصمیم ا ست که محا سبه آن ساده بوده و حتی الامکان احتمال خطای آن پایین با شد. بدین منظور می توان از فر ضیه ی « بای سیان» (Baysian theory) ا ستفاده کنیم، این تئوری از آمار و احتمالات بصورت هماهنگ استفاده می کند.

اطلاعات ما در مورد طبقات (که از آن برای تصمیم گیری ا ستفاده می کنیم) معمولاً از یک د سته محدود از الگوها گرفته می شود که ا صطلاحاً تحت عنوان طبقات پیوندی نامیده می شود. این نمونه معمولاً مجموعه ی آموزشی نامیده می شود.

اگر ما یک مجموعه خاص را انتخاب کنیم هر الگوی موجود در این مجموعه باید بطور کامل تقسیم بندی شود. حتی اگر تقسیم بندی الگوها تداخل پیدا کند. تقسیم بندی کننده ها یا همان (classifiers) به منظور تق سیم بندی بی نقص الگوها بوجود آمده اند، بنابراین می توان آنها را به گروه های دیگر تعمیم داد. ت صویر پایین یک مجموعه ی تقسیم بندی شده را نشان می دهد که با توجه به شکل قبلی تقسیم بندی شده اند. در اینجا تقسیم بندی کامل انجام شده است درحالیکه طبقات با هم تداخل هم دارند.



بنابراین ما باید بدنبال روش های درست برای ساده تر کردن این تقسیم بندی ها باشیم و این کار از طریق آزمودن و خطا و آزمایش نمونه های مختلف امکان پذیر خواهد بود.

ما می توانیم به کمک تابع جدا کننده ها gi(x) تقسیم بندی های گوناگون را عملی کنیم. یکی از اشکال ساده این تابع، الگوی خطی است که می توان تا حد زیادی در این تقسیم بندی ها به ما کمک کند.

همچنین این طبقه بندی کننده ها می توانند با توابعی که منطبق بر روش های دیگر هستند کار کنند برخی از رایج ترین آنها عبارتند از:

* حداقل فاصله، فاصله بین بردار و طبقه ی مربوطه و انتخاب شده
* نزدیک ترین عامل، در اینجا یک الگوی جدید با الگوی مجاور در نظر گرفته می شود.
* شبکه های عصبی

۲ – ۷ الگوریتم ما

در اینجا ما با یک مجموعه از الگوریتم های طبقه بندی شده سر و کار داریم که خودمان آنها را به وجود آورده ایم. قصد ما از انجام اینکار توصیف شیوه های گوناگون و مقایسه ی عملکرد نسبی آنها با هم است.

از آنجائیکه بسیاری از این الگوریتم ها به لحاظ محاسباتی پیچیده هستند و به منظور سرعت بخشیدن به این محاسبات و جلوگیری از بروز خطاهای آشکار ما ابعاد و فضای شکل ها را پایین نگهداشتیم (ارزش d برابر ۲ قرار داده شده است) همچنین به همین دلیل، طبقه بندی کننده یا classifier را محدود نمودیم تا فقط از این طریق بررسی تفاوت های خطی بین آنها بپردازیم.

عملکرد مجزای خطی به دو طبقه ی خاص محدود می گردد. هر الگو دارای اطلاعات خاص خود است که بصورت ۱ یا ۱– کد بندی شده است.

با داشتن مجموعه از الگوهای نمونه، ما از سه تکنیک متفاوت برای یافتن خطی استفاده می کنیم که به بهترین شکل بتواند این مجموعه را به دو طبقه جداگانه و مجزا تقسیم کند. بنابراین عملکرد مجزای خطی در فضای ابعاد «d» را می توان بدین شکل پارامتر بندی نمود:

$$f(x) = \sum_{i=1}^{d} \omega_i x_i + \omega_{d+1}$$

در اینجا xi ابعاد و وکتور شکل را نـ شان می دهد و وزن ها را می باید متنا سب با طبقه بندی کننده، تنظیم نمود.

برای کاهش بیشتر تعداد پارامترها، در صورتیکه موارد ۲ بعدی مدنظر باشد می توان از الگوریتمی مرسوم به Brute force و یا الگوریتم ژنتیکی برای پارامتربندی خط و زاویه استفاده نمود. (البته باید محور x را هم در نظر گرفت) و نزدیک ترین فاصله نسبت به مورد اصلی را در نظر گرفت، این روش اصطلاحاً پارامتر بندی نرمال (طبیعی) نامیده می شود.

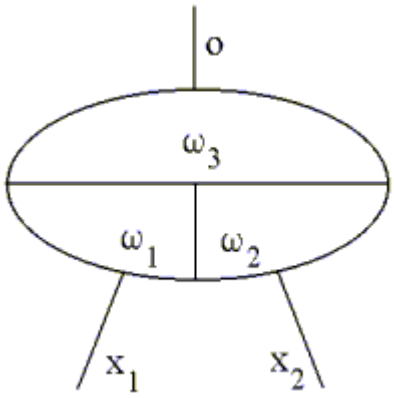
۳ – ۷ روش بوروت فورس (Brute force)

اگر فرض کنیم وکتورهای (بردارها) شکل با ارزش های (۱ و ۱-) تنظیم شده اند به منظور یافتن راه حلی مناسب باید از پارامترهای فاصله ای استفاده نمود زاویه ها بین صفر تا ۳۶۰ درجه محدود شده اند و فواصل صفر تا ۱/۴۱ .

با در نظر گرفتن پارامترهای خاص می توان عملکردهای گوناگون را مورد بررســـی قرار داد که البته مواردی محدود هستند. الگوریتم (در ابعاد d) در زمان $o(n_1 n_2 ..... n_d)$ می باشند و n_k تعداد زیر مجموعه های موجود در امتداد پارامتر k محورها می باشد.

کاملاً واضح است که این شیوه های آزمایش و خطا با افزایش ابعاد و فضای شکل غیر قابل اجرا تر می شود.

٤ – ۷ شبکه خنثی



یک شبکه خنثی به جستجوی راه حلی مطلوب می گردد و این شیوه و فرآیند کاهش خطا استفاده می نماید. هنگامیکه ما بدنبال یک تابع مجزای خطی ه ستیم، ا ستفاده از یک شبکه ی خنثی بدون هیچ گونه لایه ی پنهان، برای ما کفایت می کند.

اگر بردار شکل را داشته باشیم، نتیجه اینگونه محاسبه می شود.

$$o = f\left(\sum_i a_i x_i + b\right) \quad (2)$$

WikiPower.ir

در این فرمول ai اصطلاحاً نیروهای سیناپتیک (synaptic) نامیده می شود، b اصطلاحاً حداقل مقدار (یا آستانه) نامیده می شود و f(x) اصطلاحاً تابع فعال نامیده می شود. اگر برای فعال سازی از تابع هویت (شناخت) استفاده شود در اینصورت شبکه ی خنثی فقط یک تابع خطی خواهد بود.

برای ساده تر کردن محا سبه ی تابع خطا ما از f(x) = c tanh(x) ا ستفاده می کنیم. ارزش تابعی دارای نماد و علامتی م شابه (x) ا ست (که برابر ا ست با ترکیب خطی از معادله ۲)، و طبقه بندی آن بر طبق همان علامت انجام می شود.

(که علت انتخاب ۱- و ۱ برای متغیرهای طبقه را توصیف می کند.)

تمامی الگو ها دارای ارزش (قدر) مثبت برای این تابع فعال هستند که تحت عنوان طبقه ۱ طبقه بندی شده و بقیه ی آنها تحت عنوان طبقه ۱- طبقه بندی می شوند.

قدر مطلق c شیب تابع نسبت به مبنا را محاسبه می کند، تابع خطا عبارتست از مجموعه تفاوت بین طبقه ی الگو و اینگونه محاسبه می گردد.

$$E = \frac{1}{2}\sum_{i}(o - class_i)^2$$

شیوه هایی که بر پایه ی Error – back- propagation استوارند، پارامترها را طوری تنظیم می کنند تا از این طریق تابع خطا را به حداقل بر سانیم. همانطور که شیب تابع به طرف پر شیب ترین نقطه متمایل است جهت دیگر باید بسمت حداقل باشد. بنابراین هر پارامتر P را می توان اینگونه بیان کرد:

$$p = p + \Delta p \Leftrightarrow \Delta p = -e\frac{\partial E}{\partial p}$$

در اینجا «e» عدد کوچکی که اصطلاحاً «سرعت فراگیری» نامیده می شود. «سرعت فراگیر» متغیر است تا بتواند بر سرعت همگرایی تاثیر بگذارد. ارزش ها را می توان پس از م شخص شدن الگوها تنظیم نمود (که به آن فراگیری آنلاین گفته می شـــود) و یا اینکه می توان پس از اتمام کل دوره انجام داد (که به آن فراگیری گروهی گفته می شـــود) در مرحله آموزشـی شـبکه مجموعه ی آموزشـی بطور مکرر از شـبکه عبور می کنند. (هر کدام از این گروه ها اصـــطلاحاً دوره ی زمانی و یا epoch نامیده می شـــوند. تا اینکه بین پارامترها همگرایی بوجود آید و تابع خطا دچار کاهش بیشتر نگردد.)

دقت دا شته با شید که میزان فعال بودن و کارائی همیشه باید با ا ستفاده از بکار روش خاص خارج از آموزش مورد سنجش و بررسی قرار گیرد.

شیوه های کاهش میزان خطا، مانند «شبکه های خنثی» از اطلاعات خاص در مورد شکل و فضای پارامترها استفاده می کنند بنابراین ممکن است منجر به خطر بوجود آمدن حداقل موضعی (local minimum) شود و بنابراین می تواند سبب شکل گیری راه حلی برای مسئله گردد.

۵ – ۷ الگوریتم ژنتیک

الگوریتم های ژنتیک در واقع الگویی برای محاسبه کلی، سفارشی نمودن و تقسیم بندی مسائل هستند و قادرند که حد و مرز بین پارامترها را تعریف کرده و به بررسی و مطالعه آنها بپردازند.

در این زمینه ما نیاز به یک روش مطلوب و بی نقص داریم بطوریکه بتواند از بروز خطاها جلوگیری کند و به نتیجه مطلوب و قابل قبول دست یابد. الگوریتم های ژنتیکی در واقع از ترکیبی از تحقیقات و مطالعات جهانی و اطلاعات محلی استفاده می کند.

یکی از ساختارهای اولیه الگوریتم ژنتیکی «کروموزم» نام دارد که در واقع حامل یک سری اطلاعات در مورد برخی پارامترهاست و نشان دهنده اطلاعاتی پیرامون توابع است. شکل اصلی توابع و پارامتربندی آنها از قبل باید مشخص شود.

هر کروموزم دارای یک تعداد ژن است که مساوی است با تعداد پارامترهای موجود در تابع مجزا یک ژن نمایش دو گانه ی ارزش پارامتر است. این در واقع یک توالی از بیت ها ست (bits)، (صفر و یک) و طول آن (که از قبل تعیین شده است) دقت ارزش پارامتر ها را منعکس می کند. در مطالعات و تحقیقات انجام گرفته توسط ما، تمامی اطلاعات و داده ها دوباره تنظیم ورده بندی شدند بطوریکه تمامی پارامترهای مربوطه در محدوده ی (۱ و ۱–) قرار گیرند و از یک ژن ۱۰ بیت (10bit) برای رسیدن به نتایج با دقت ۰/۱٪ استفاده گردید.

تابع تناسب یک کروموزم در واقع یک نوع سنجش و اندازه گیری کارائی آن است و همچنین کمیت الگوریتمی که برای رسیدن به حداکثر تلاش می کند. در مطالعات ما، این در واقع بخشی از الگو هایی است

که با توابع خاص پارامتربندی شـــده و در یک مجموعه، برای انجام آزمایش قرار می گیرند. این نوع انتخاب به ما اجازه می دهد که الگوریتم ژنتیک را مورد آزمایش قرار داده و آنها را با شیوه های انتخابی قبلی مقایسه کنیم.

یک کروموزم جدید از پیوند دو کروموزم و با اســـتفاده از یک نقطه تلاقی جدید بوجود می آیند، و یک نقطه بصورت راندوم و تصادفی انتخاب می شود تمامی خصوصیات از والد A به نقطه ی تلاقی انتقال می یابد و یا کپی می شود. و سپس تمامی خصوصیالت والد B از نقطه ی تلاقی به انتهای کروموزم کپی می شود. بنابراین کروموزم جدید از قسمت بالا و قسمت سر شبیه یکی از والدین است و از قسمت انتها و دُم شبیه والد دیگر است پیچیدگی موجود بین والدین A و B (که یکی از آنها شبیه سر کروموزم جدید و دیگری شبیه دُم آن است.) توسط یک حرکت راندوم و تصادفی حد و فصل می شود. البته برخی از گوناگونی ها که از بیش از یک نقطه ی تلاقی استفاده می کنند و یا خصوصیات والدین خود از روش های دیگر استفاده می کنند نیز موجود هستند.

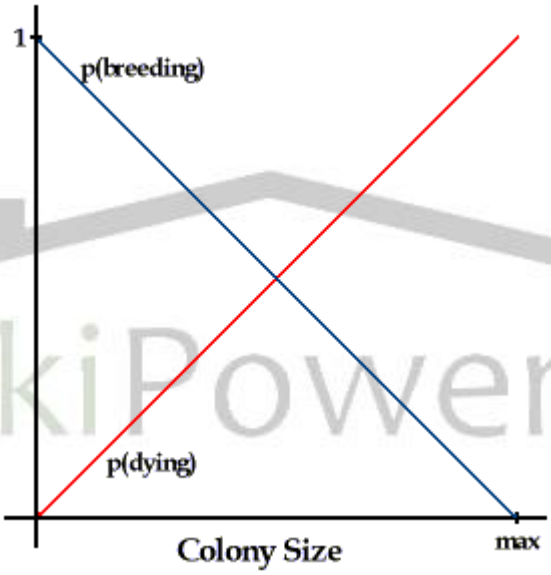
جهش ها یکی از بخش های مهم الگوریتم هستند زیرا آنها به حلق شیوه ها و راه حل های جدیدی منتهی می شوند در تحقیقات ما مشخص شد که هر قسمت از ژنوم کروموزم جدید دارای یک احتمال ثابت برای جهش است، فرآیند جهش در واقع باعث تغییر ذره (bit) از حالت ۰ به ۱ و یا برعکس می شود.

احتمال جهش در واقع یک پارامتر حائز اهمیت بوده و می باید با گذر زمان دچار تحول و تغییر گردد. در ابتدا، سعی بر این است که احتمال جهش بالا رود و این جهش شامل نقاط مختلف فضای پارامتر گردد. همانطور که ما به حالت مورد نظر نزدیک می شویم باید احتمال جهش یابد تا بدین ترتیب کروموزم های جدید ایجاد شده و به اندازه ی کافی به والدین خود شبیه شوند.

یک کُلُونی (Colony) در واقع مجموعه ای از کروموزم هاست که همزمان با پیشرفت الگوریتم، تکامل می یابد و از طریق بوجود آمدن دو کروموزم اتفاقی (راندوم) بوجود می آید و سـپس به رشـد و تکامل خود ادامه می دهد.

اندازه و وسعت این کلونی بستگی به چگونگی و سرعت الگوریتم دارد.

در هر دوره ی زمانی هر ارگانیسمی دارای فرصت برای زاد و ولد است و از این طریق می تواند ژن های خود را به کروموزم های جدید انتقال دهد.



چگونگی و احتمال زاد و لد و تولید مثل از طریق بررسی اندازه ی کلونی ۱ با حداکثر اندازه ی کلونی تعیین می شود. از بین ۱۰ کروموزم تنها یکی از آنها بطور تصادفی انتخاب می گردد. برای حفظ اندازه ی کلونی کروموزم های پیوندی بر اساس معیار غریزی می میرند و از بین می روند بجز قوی ترین و بهترین آنها. بنابراین در هر دوره ی زمانی تمامی کروموزم بجز بهترین و قوی ترین آنها ممکن است بمیرند و علت آن تجمع بیش از حد آن و گنجایش کلونی و حداکثر اندازه ی کلونی است.

برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

علاوه بر این یک بخش از کروموزم ها که دارای عملکرد ثابت پایینی هستند، دارای احتمال مردن (مرگ) در اثر انتخاب طبیعی هستند. این مسئله حائز اهمیت است که آندسته از کروموزم هایی که دچار اختلال در عملکرد هستند کاملاً نابود نمی شوند زیرا آنها هنوز دارای شانس برای تولید اولاد (مولود) می باشند.

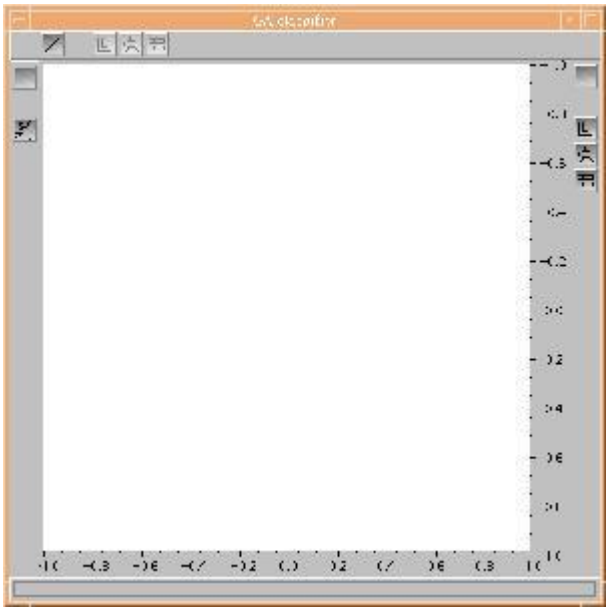
٦ – ٧ آپلت (Applet)

هنگامیکه پنجره ی آپلت (آپلت نام یک برنامه ی کامپیوتری است که گاهی «ریز برنامه» نیز نامیده می شود) باز شد، باید صبر کنید تا کاملاً بارگذاری و یا اصطلاحاً لود (load) شود که این مدت کمی طول می کشد. پس از باز شدن صفحه شما سه میله ابزار یا toolbar و یک محیط اطلاعات را در قاب اصلی آپلت مشاهده خواهید کرد.

چهارچوب اصلی آپلت :

میله ابزار ا صلی در بالای این چهارچوب واقع شده ا ست، که برای طبقه بندی نمودن اطلاعات ا صلی بکار می روند:

WikiPower.ir



- دکمه ی خروج از اپلت
- راه اندازی الگوریتم بروت فورس (Brute Force)
- راه اندازی شبکه خنثی
- راه اندازی الگوریتم ژنتیک
- اطلاعات در مورد اپلت

نوار ابزار داده ها در قسمت سمت چپ قرار گرفته شده است و برای بررسی و جابجایی اطلاعات بکار می رود:

- پاک کردن اطلاعات
- ایجاد و به روز کردن اطلاعات

« نوار ابزار پارامتر» در قسمت سمت راست صفحه قرار گرفته شده است و برای تغییر پارامترهایی که در طبقه بندی کننده ها اثر می گذارد، بکار می رود:

- ذخیره کردن ارزش های پیش فرض همه ی پارامترها
- پارامترهای الگوریتم (Brute Force) بروت فورس
- پارامترهای شبکه عصبی
- پارامترهای الگوریتم ژنتیکی

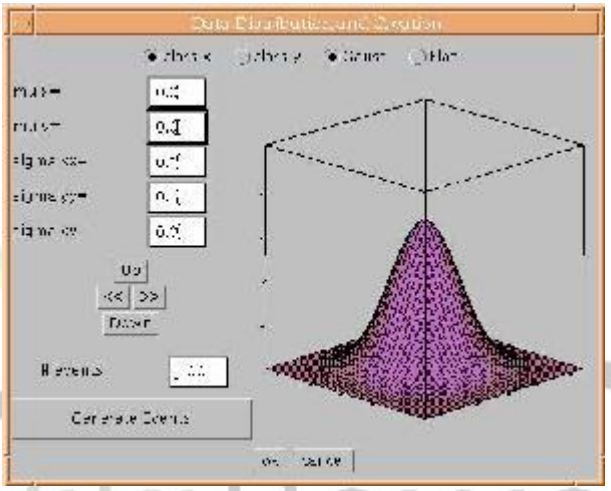
صفحه ی اطلاعات و داده ها، نقاط داده ها و عملکردها(توابع) مختلف را نشان می دهد که بشکل رنگی کُد بندی شده اند.

رنگ سبز نشان دهنده ی بروت فورس (Brute Force) است.

رنگ آبی شبکه عصبی را نشان می دهد (Neural Network)

رنگ قرمز الگوریتم ژنتیکی را نشان می دهد. (Genetic Algorithm)

انتشار داده ها و ایجاد چهارچوب (frame)با استفاده از این چهارچوب شما قادر خواهید بود که اطلاعات مورد نظر را دسته بندی و طبقه بندی کنید.



WikiPower.ir

تعداد خاص مربوط به طبقات مختلف (x یا y) بر طبق تقسیمات خاصی بوجود می آیند. (Flat یا Gaussian). تقسیمات مورد نظر از طریق سه عنصر ناتریس کوواریانسی انجام می شوند.

گاسیان دو ظرفیتی (Gaussian) بشکل طبیعی و معمول تعریف می شود. در حالی که انتشارات مسطح (flat) دارای تراکم بیشتری است و توسط ماتریس کوواریانسی تعریف می شوند.

برای مشاهده ی تصاویر سه بعدی از جهات مختلف می توان از دکمه های چرخشی استفاده کرد.

فریم ها (چهارچوب) انتخاب پارامتر از اینها برای تنظیم پارامترها و طبقات مختلف استفاده می شود که در بخش قبلی توضیح داده شدند. از ابزار اسلایدر (slider) می توانید برای انتخاب یک ارزش خاص استفاده کنید، بدین منظور پیش فرض هایی نیز وجود دارد.

شما باید کار خود را با ایجاد مجموعه داده ها شروع کنید و بعداً می توانید آنها را آپدیت کنید. هر دوی این طبقه ها نیازمند آموزش هستند. (GA و NN) بدین ترتیب نتایج بهتری بدست خواهد آمد. هر عملیات دارای مراحل مختلفی است که باید بترتیب انجام شود. برای ایجاد یک طبقه ی جدید باید از نوار ابزار پارامتر استفاده کنید.

برای کسب اطلاعات بیشتر می توانید مطالب مربوط به javadoc را مطالعه کنید.

برای بالا بردن سرعت عمل خود می توانید ورژن های جدید آپلت را دانلود کنید، سپس باید فایل را از حالت فشرده یا زیپ (zip) خارج کنید و کار با آپلت را با استفاده از نرم افزار java aplet آغاز نمائید.

WikiPower.ir

۷ – ۷ کاربردهای بیشتر

انتخاب شکل

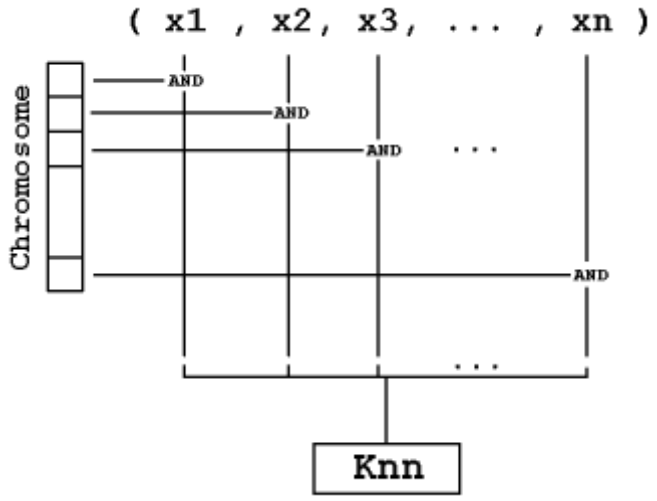
همانطور که قبلاً مشاهده نمودید، هدف اصلی از کاربرد الگوریتم ژنتیکی بهینه سازی است. ما قبلاً شاهد بوده ایم که چگونه می توان از آنها برای مشخص کردن حد و مرز و طبقه بندی ها استفاده کرد.

در هنگام تشخیص این الگوها ما با مشکل طبقه بندی مناسب و موفقیت آمیز روبرو هستیم و این مشکلی است که تحت عنوان انتخاب شکل نامیده می شود. انتخاب شکل مخصوصاً زمانی مفید است که ما با اشکال و الگوهای چند بعدی سر و کار داریم، مثلاً صدها و یا شاید هم هزاران شکل، و این باعث افزایش طبقات و اشکال می گردد.

اما واقعاً ما باید خودمان را به شکل ها و طبقات محدود و خاص، محدود کنیم؟ دلیلش کاملاً آشکار است، انتخاب از بین اشکال و خصوصیات بسیار زیاد، باعث پر هزینه شدن محاسبات می گردد و ما مجبوریم که آنها را کاهش داده و محدود کنیم. از این گذشته برخی از گزینه ها دارای دقت زیادی نیستند بنابراین ما مجبوریم از بسیاری از آنها چشم پوشی کنیم و در عین حال برخی از آنها را مورد توجه قرار دهیم، در این صورت ما می توانیم کنترل بهتری روی آنها داشته باشیم شاید این سوال پیش بیاید که چرا ما اندازه گیری های پر هزینه و غیر مفید را بطور کلی کنار نمی گذاریم؟ در جواب این سوال باید بگوییم که بسیاری از این سنجش ها به طور کلی هم غیر مفید نیستند و می توان از آنها بهمراه دیگر روش ها استفاده کرد و به نتایج قابل قبولی دست یافت. (برای اطلاعات بیشتر به کالمان فیلتر kalman filter) مراجعه کنید.

حالا که ما فهمیدیم که انتخاب اشکال در هنگام سر و کار داشتن از اشکال چند بعدی از اهمیت بالایی برخوردار است، اکنون به مسئله اهمیت و کاربرد الگوریتم ژنتیک در این فضا ها می پردازیم، روشی را که ما در اینجا مطرح می کنیم بطور کلی ژنتیکی است بنابراین جدای از فنون طبقه بندی مطرح می شود.

قصد ما بر این است که از بهترین و دقیق ترین مورد برای طبقه بندی الگوها استفاده کنیم و در این روش از الگوریتم ژنتیک استفاده می کنیم. فرض کنید که ما برای هر الگوی ورودی دارای ۱۰۰ وسیله سنجش و شکل هستیم. ما اقدام به ایجاد یک کروموزم به ۱۰۰ نمودیم که هر کدام از ژن های آن با یک قسمت خاص هماهنگ بود. اگر فرض کنیم که بیت (bit) برابر ۱ باشد ما طبقه بندی knn را در نظر می گیریم و می توانیم از آن چشم پوشی کنیم.



همانطور که قبلاً گفته شد ما به منظور داشتن سنجش و ارزیابی نسبی، این طرح را در نظر می گیریم. اینکار را می توانیم براحتی با جایگزین نمودن هر bit با محدوده عدد صـــحیح ۰ تا ۱۰ انجام دهیم اینکار باعث د ستیابی به نتایج بهتر برای ما می گردد. وزن های بالاتر دارای حـ سا سیت بی شتری هـ ستند در حالیکه وزن های پایین تر از اهمیت کمتری برخوردارند. و وزن های نســـبی نتایج قابل قبول و مطلوبی در اختیار ما قرار می دهند.

تابع تنا سب بر ا ساس چگونگی کروموزم ها در حـ سب طبقه بندی ا ستوار ا ست. ارزیابی آن کاملاً مشخص و وا ضح ا ست، ما برای اینکار از شیوه ی knn برای اطلاعات موجود و یک سری خـ صو صیات تعریف شده استفاده می نماییم.

هماهنگ بسیاری از الگوریتم های ژنتیکی ارزیابی تابع تناسب یکی از با اهمیت ترین و وقت گیرترین بخش ها ست. بنابراین اگر بخواهیم به نتایج مطلوبی و بی نقصی د ست یابیم، بجای اینکه فقط از روش (Brute Force) بروت فورس اســـتفاده کنیم باید از شـــیوه های ارزیابی دیگر نیز بهره بگیریم، (یعنی اینکه هر کروموزم را بصورت جداگانه و دقیق مورد ارزیابی و مطالعه قرار دهیم.) چنین کاری بدون شک هم در وقت و

هم در هزینه مقرون بصرفه خواهد بود. یکی از روش های ساده برای ارزیابی این راندمان استفاده از تابع زیر است:

تناسب = (نتایج درست – مجموع نتایج) ÷ مجموع نتایج

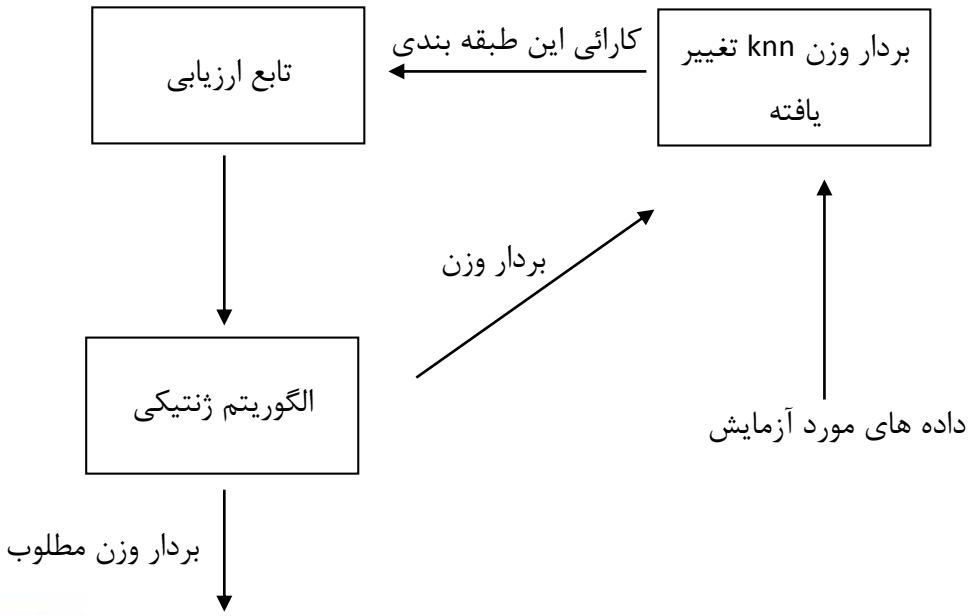
این شیوه بطور موثری الگوهای تقسیم بندی صحیح را به ما نشان می دهد. اگر چه این تابع باعث دست یابی به نتایج دقیق می شود ولی باید گفت که کافی نیست این شیوه بیشتر بر پایه ارزش k استوار است (تعداد مجاوران نزدیک) فرمول زیر می تواند ما را به نتایج بهتری رهنمون سازد:

$$Fitness = \gamma \frac{TotPats - CorrectPats}{TotPats} + \delta \frac{n\min/K}{TotPats}$$

در این فرمول گاما و دلتا قدر مطلق های انتخابی هستند که بر تجربه و آزمایش استوارند و n min تعداد « مجاورها » هستند که در طبقه بندی مورد استفاده قرار نمی گیرند.

شکل زیر به طور خلاصه این روش را بیان می کند:

برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه



اگر استفاده از الگوریتم های ژنتیکی دارای روش های مختلف است، و شیوه قبلی فقط برای مثال و آشنایی شما مطرح شد. باید گفت که الگوریتم ژنتیکی در واقع یک ابزار و وسیله برای حل برخی مشکلات است و به تنهایی برای رسیدن به هدف کافی نیست، اگر الگوریتم های ژنتیکی بصورت کامل و مطلوب مورد استفاده قرار گیرند می توانند تا حد زیادی در تحقیقات به ما کمک کنند، بدون اینکه زیانی را متوجه ما کنند. برای کسب اطلاعات بیشتر پیرامون چگونگی استفاده از الگوریتم های ژنتیک می توانید با مرکز تحقیقات الگوریتم ژنتیکی واقع در دانشگاه میشیگان و همچنین گروه تحقیقاتی Genitor واقع در دانشگاه ایالتی کلورادو تماس حاصل کنید.

۸ – ۷ نتیجه گیری

پس از انجام پاره ای مطالعاتو تحقیقات پیرامون کارائی نسبی شیوه های سه گانه و نرم افزار کامپیوتری آپلت (Applet)، به نتایج زیر دست یافتیم.

* استفاده از شیوه ی نرم افزاری بروت فورس (Brute Force) تا حد زیادی غیر قابل انجام است زیرا پیچیدگی زمان باعث افزایش میزان gd می گردد، در اینجا حرف g نشان دهنده ی میزان و چگونگی محور شکل است و حرف d نشان دهنده ی تعداد اشکال است، اگر بخواهیم در این زمینه به نتایج بهتری دست یابیم، نیاز است که دقت بیشتری بکار ببندیم، بنابراین به زمان بیشتری هم نیازمندیم.

* شبکه های عصبی به مراتب نتایج مطلوب تری را برای ما به جای می گذارند و آموزش آنها نیز نسبتاً سریع تر و آسان تر صورت می گیرد. خصوصاً هنگامیکه ما با شبکه های پیچیده و مبهم سر و کار نداریم. البته متاسفانه باید گفت میزان متمرکز هیچ ارتباطی به سرعت یادگیری و شروع ساختار بندی شبکه ندارد. و همواره احتمال بروز مشکلات بعلت پایین بودن کارائی وجود دارد. در نتیجه حتی ممکن است پس از آموزش طولانی مدت باز هم طبقه بندی ضعیف اتفاق بیافتد.

* طبقه بندی بر اساس الگوریتم ژنتیکی باعث نوعی سازش می شود. و سرعت تمرکز شبکه عصبی پایین می آید ولی احتمال گیر افتادن در یک وضعیت نامطلوب تا حد زیادی کاهش می یابد. البته می توان از این دو شیوه بصورت مشترک استفاده نمود و به کمک الگوریتم ژنتیکی و شیوه ی شبکه عصبی به نتایج بهتری دست یافت. [6]

برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

ضمیمه

بعد از اینکه یک دید کلی نسبت به الگوریتم های ژنتیک پیدا کردیم به مسءله PSO می پردازیم:

"یک استراتژی تحقیقاتی جهانی پیرامون بهینه سازی مجموعه ذراتی که دارای رفتارکوانتومی هستند"

چکیده

در این مقاله،ما با توجه به الگوریتم "بهینه سازی مجموعه ذرات با رفتار و خصوصیات کوانتومی"،(که به طور اختصار QPSO نامیده می شود)اقدام به فرمول بندی و ارزیابی اساسی و فلسفه ی QPSO ارائه می کنیم.سپس اقدام به بررسی و مطالعه الگوریتم اصلاح شده ی QPSO از دیدگاه های مختلف می نمائیم و در پایان نتایج آزمایشاتی را که برتری این الگوریتم نشان می دهند از نظر می گذرانیم

واژه های کلیدی و مهم که در این مقاله بکار رفته اند عبارتند از :

PSO با رفتار کوانتومی (Quantum-behaved pso)،ذره (particle) ، مجموعه (population)،جریان اصلی تفکر (main stream thought) ، یافتن نقطه ی انحراف (learning inclination point).

مقدمه

الگوریتم بهینه سازی مجموعه ذرات (یا همان pso) ، یک نوع تکنیک تکامل یافته ای است که در سال ۱۹۹۵ توسط محققانی به نام های «کندی» و «ابرهارت»(Kennedy & Eberhart) ، معرفی ارائه گردید. سیستم pso در واقع ،مجموعه ذرات را از منظر و دیدگاه علمی شبیه سازی می نماید که در آن،ذرات در ابعاد متنوع و مختلف مورد ارزیابی و بررسی قرار می گیرند تا بدین ترتیب وضعیت مطلوب و یا غیر مطلوب آنها مشخص گردد. در هر مرحله و فرآیند تکرار ،موقعیت ذرات نسبت به وضعیت مطلوب مورد مطالعه و بررسی قرار می گیرد، ذراتی که در مجاورت هم قرار دارند دارای خواصی نسبتاً مشابه می باشند ، آنها از این خواص مشابه (حافظه مشابه) جهت تطبیق سرعت (جنبش) و موقعیت های بعدی خود استفاده می کنند. قبلاً مشخص شده است که الگوریتم pso به لحاظ کارآیی و شیوه ی عملکرد با الگوریتم ژنتیک قابل مقایسه است (الگوریتم ژنتیک اختصاراً GA نامیده می شود).

طبق الگوی اصلی pso ، هر کدام از این ذرات (اجزاء) در فضای چند بعدی مورد مطالعه قرار می گیرد ، بردار مکان و بردار سرعت ذره "i" به صورت زیر بیان می گردد:

$$Xi(t) = (Xi(t) , Xi(t) ,..... xi p((t))$$

$$Vi(t) = (Vi(t) , V3i(t) ,..... Vi(t))$$

این ذرات طبق معادله زیر حرکت می کنند:

$$Vi(t+1) = Vi(t) + Q1(Pi - Xi(t)) + Q2(Pg - Xi(t)) \qquad (1a)$$ (الف)

www.wikipower.ir

برای دریافت فایل wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

$$Xi(t+1) = Xi(t) + Vi(t+1) \qquad (1b)$$ (ب)

در این مطالعه ،Q1 وQ2 اعدادی تصادفی (راندوم) می باشند که محدوده ی بالایی آنها همان پارامتر الگوریتم است.

$$Pi = (Pi1 , Pi2 ,\ldots\ldots, PiD)$$

حالت قبلی (حالتی که به بهترین شکل ارزش مطلوب را نشان می دهد) ذره i را نشان می دهد که اصطلاحاً pbest نامیده می شود و وکتور (vector)

$$Pg =(Pg1 , Pg2 ,\ldots\ldots, PgD)$$

درواقع بهترین ذره را در میان تمامی ذرات نشان می دهد که اصطلاحاً "gbest" نامیده می شود.بسیاری از ورژن های (انواع) اصلاح شده ی الگوریتم pso از زمان اولین پیدایش آن در سال ۱۹۹۵ ، تاکنون به لحاظ عملکردی روندی رو به رشد و تکامل داشته اند.

هر زمان که تحول و تکامل pso مد نظر بوده است از دو شیوه و رویکرد اثر گذار و مفید صحبت به میان آمده است که عبارتند از وزن اینرسی (W) و یا عامل فشار (انقباض) K که در معادله (۱) نشان داده شده است بنابراین معادله "1b " را می توان اینگونه جایگزین نمود.

$$Vi(t+1) =w\ Vi(t)+ Q1(Pi – Xi(t)) + Q2(Pg – Xi(t)) \qquad (2)$$

که در مدل وزن اینرسی بکار می رود و می توان آنرا با فرمول زیر جایگزین نمود.

$$Vi(t+1) = K\,[Vi(t) + Q1(Pi - Xi(t)) + Q2(Pg - Xi(t))] \qquad (3)$$

$$K \quad \frac{2}{\left|2 \quad Q \quad \sqrt{Q^{2} \quad 4Q}\right|} Q \quad Q_{1\max} \quad Q_{2\max}, Q \; 4 \qquad (4)$$

در اینجا Q1maxوQ2max بالای حدود به ترتیب Q3 وQ2 می باشند.

اساساً ، مدل های مختلف سیستم pso که در بالا بیان شده اند. شکل سیستم خطی بوده و در شرایط حضور pbest و gbest پایدارمی باشند و عدد (شماره) راندوم آنها ثابت فرض می گردد. آنالیز و تجزیه و تحلیل آنها نشان می دهد. که از هر الگویی که برای الگوریتم pso استفاده کنیم هر ذره در سیستم pso به محل و نقطه ی اصلی خود متمایل می شود.

$$P = (P1 , P2 , \ldots\ldots , PD)$$

تنها مسیری که هر ذره به سوی آن منتهی می گردد که مختصات آن عبارتند از :

$$Pd = (Q1\ Pid + Q2\ Pgd) / (Q1 + Q2)$$

بنابراین pbest همه ذرات به سوی gbest خاص منتهی می گردد و با i .

ولی باید گفت که یک ارگانیسم اجتماعی در واقع سیستمی به مراتب پیچیده تر بوده که از طریق هر کدام از معادلات (۱)و(۲)و(۳) قابل فرمول بندی است ، باید گفت حتی تصور یک ارگانیسم مجموعه ای (اجتماعی) به حدی پیچیده و دشوار است که معادله خطی به هیچ وجه نمی توان آن را به طور کامل تعریف نماید.

در عمل کامل مشخص است که تکامل تفکر انسان ها مبهم و پیچیده است دقیقاً همانند ذره ای که دارای رفتار و خصوصیات کوانتومی است.

محققی به نام "جان سان" (Jan sun) به کمک همکارانش اقدام به معرفی فرضیه کوانتومی در مورد pso نمودند و الگوریتم pso با رفتاری کوانتومی را پیشنهاد نمودند ، نتایج آزمایشات و تحقیقات صورت گرفته نشان دهنده ی این مطلب است که Qpso از جهات گوناگون بهتر از pso استاندارد عمل می کند، و این عمل در واقع یک الگوریتم مطلوب و امید بخش است.

در این مقاله، همانطور که گفته شد ،روش نوین برای کنترل پارامتر Qpso پیشنهاد می گردد. این روش جدید در برگیرنده ی الگوریتم Qpso به صورت اصلاح شده به همراه یک نقطه ی رجوع و استاندارد بین المللی است که اصطلاحاً "جریان اصلی تفکر" نامیده می شود، این شیوه به منظور سنجش و ارزیابی تحقیقاتی پیرامون این مسئله است. و بخش دیگر این مقاله متشکل از مطالب زیرند:

در بخش دوم و سوم به فرمول ها و تغییر و تحولات Qpso می پردازیم. در بخش چهارم شیوه ی کنترل پارامتر اصلاح شده ی Qpso ارائه می گردد، همچنین نتایج تجربه ها و آزمایشات در بخش چهارم عنوان شده اند و بالاخره در بخش پنجم به نتیجه گیری و ملاحظات می پردازیم.

برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

(۲) دینامیک pso کوانتومی

درالگوی کوانتومی pso چگونگی حالت ذره بر اساس عملکرد موج (تابع موج) $\bar{x},t$ تعریف می گردد نه بر اساس $\vec{x}$ و $\vec{v}$.

رفتار دینامیکی ذره در مقایسه با سیستم های قدیمی pso تا حد زیادی متغیر و متنوع است. یعنی اینکه نمی توان بطور دقیق ارزش های $\vec{x}$ و $\vec{v}$ را بصورت همزمان تعیین نمود. ما فقط قادر هستیم که احتمال پیدایش ذره را در حالت $\vec{x}$ از احتمال تابع تراکم $\vec{x},t^{2}$ تشخیص دهیم و شکل آن بستگی به پتانسیل وضعیتی دارد که ذره در آن موجود است.

محققی به نام "جان سان" (Jan sun) به کمک همکارانش از پتانسیل دلتا (delta) با نقطه ی مرکزی P =(P1 , P2 ,……, PD) برای محدود کردن ذرات کوانتومی از pso استفاده نمودند تا از این طریق ذره را به موضع اصلی $\vec{p}$ برگردانند و از ترکیدن آن جلوگیری کنند.

تابع موج (عملکرد موج) ذره در حالت پتانسیل دلتا به صورت زیر تعریف می گردد.

$$\Psi(x) = 1/(L)^{0.5}[\exp(- ||p-x|| / L]$$

و تابع تراکم احتمال بدین صورت مفروض است:

$$Q(x) = | \Psi (x) |^2 = 1/(L) [\exp(-2||p-x|| / L)]$$

پارامتر L بستگی به میزان انرژی پتانسیل دارد و حیطه ی (منطقه) جستجوی ذره را مشخص می کند که ما در این مقاله اصطلاحاً آن را "خلاقیت" و یا "تصور" ذره می نامیم.

در قلمرو pso با رفتار کوانتومی ، دامنه تحقیقات و ارائه ی راه حل متشکل از دو فضای متفاوت به لحاظ کیفیتی است.

تابع موج و تابع احتمال در واقع حالت ذره را درفضای جستجوتعریف می کند. ولی هیچ اطلاعات خاصی را در مورد موضع قرارگیری ذره که برای سنجش پایداری ذره نیاز است را در اختیار ما قرار نمی دهد.

بنابراین تغییر شکل و انتقال حالت بین دو فضا بدون شک امری لازم و ضروری است.

درقلمرو مکانیک کوانتوم ، تغییر شکل از حالت کوانتومی به حالت کلاسیک اصطلاحاً "ریزش" نامیده می شود، که در شرایط طبیعی به اندازه گیری موقعیت ذرات اطلاق می گردد.

تفاوت موجود بین تغییر حالت Qpso و pso سنتی در شکل های (۱)و (۲) نشان داده شده است.

روش "مونت کارلو" (monte carlo)، با شبیه سازی استوکاسیک (stochastic) ، می تواند این پروسه را با استفاده از کامپیوتر تشخیص ، و حالت آن را می توان اینگونه بیان نمود:

$$X(t) = p \pm L/2\ [\ln(1/U)]$$

پارامتر Lاینگونه ارزیابی می شود:

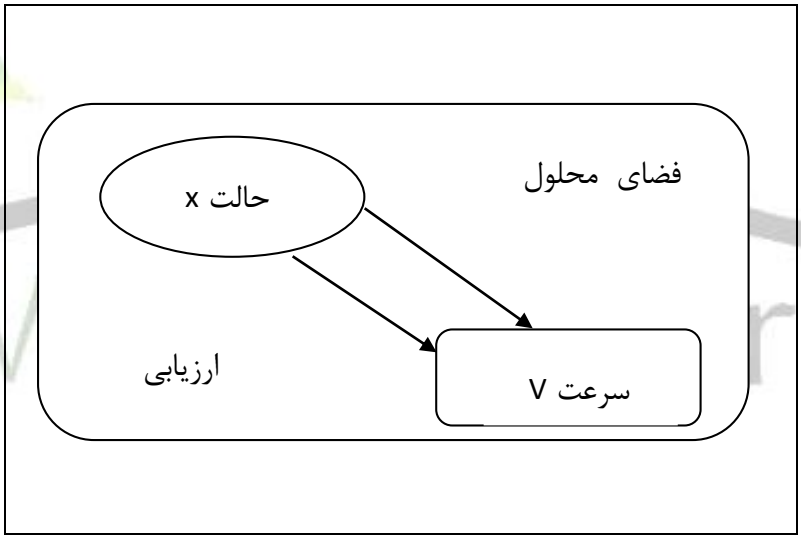
$$L(t+1) = 2*a*|p-x(t)|$$

بنابراین معادله کوانتوم pso بدین صورت بیان می شود:

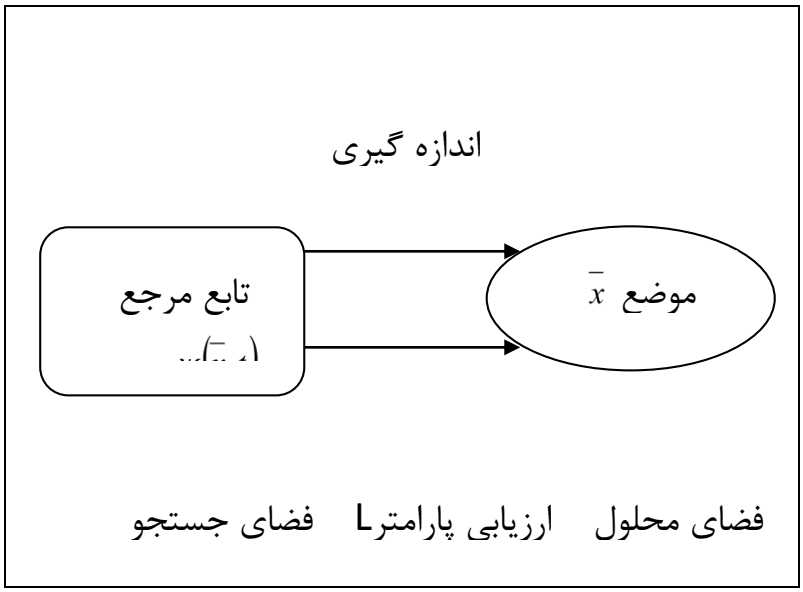
$$X(t+1) = p \pm a*|p-x(t)\ |*\ln(1/U)$$

که جایگزین معادله (۱) در الگوریتم Qpso می گردد.

شکل ۱



شکل ۲



(۳) فلسفه ی کوانتوم pso (pso کوانتومی)

A- مطالعه و بررسی الگوی ذرات

با بررسی علمی مجموعه ارگانیسم ها دو مفهوم اصلی در خصوص pso سنتی وجود دارد که عبارتند از:

pbest و gbest.

همانطور که قبلاً گفته شد pbest بهترین حالت قبلی که توسط هر ذره مفرد به دست می آید و gbest بهترین حالتی است که به همه مجموعه ذرات مربوط می شود. هر کدام از ذرات جستجوی خود را تا نقطه p که بین pbest و gbest است آغاز می کند.نقطه ی p که مختصات آن در معادله (۵) نشان داده شده است نتیجه روابط متقابل مطالعات فردی و آموزش های جمعی است و ما نقطه p را اصطلاحاً نقطه LIP می نامیم. هر بخش بطور جداگانه به دنبال اطلاعاتی جدید در مورد LIP است که از طریق gbest وpbest قابل تعیین است.در pso سنتی، میزان تمایل یک ذره از طریق حرکت و نوسانات آن با LIP در

حالت تعادل نقطه p تعیین می شود. ولی باید گفت که مجموعه ارگانیسم از سیستم خطی دور است این فرآیند به حدی پیچیده است که تشخیص مسیر حرکت آن براحتی قابل تشخیص نیست . از این گذشته مدل pso سنتی نمی تواند به ما بگوید که چرا برخی استعدادهای عجیب مانند "آلبرت انیشتن" در جامعه ظهور می کند. در حقیقت باید گفت که جامعیت ذرات در سیستم pso سنتی باعث محدود شدن دامنه فعالیت ذره شده ودر نتیجه نمی توان گفت از ذره ای با فاصله زیاد از مجموعه ذرات دیگر قرار داشته باشد ، ممکن است در مورد آن راه حلی بهتر از gbest داشته باشد.

B- مزایای مدل کوانتومی (Quantum Model)

مدل کوانتومی pso می تواند بر نقاط ضعف سیستم قدیمی pso چیده شود، و علت آن هم وجود سه خصوصیت زیر است:

WikiPower.ir

* سیستم کوانتومی در واقع یک سیستم غیر خطی پیچیده است که در آن اصل وضعیت فرا موضعی (principle of stak superposition)وجود دارد ،بنابراین یک سیستم کوانتومی دارای حالات و تنوعات بیشتری نسبت به سیستم خطی می باشد.

* سیستم کوانتومی در واقع سیستمی مهم و پیچیده است که از بسیاری از جهات با سیستم استوکاستیک سنتی (stochastic system) متفاوت است. یک ذره در چنین سیستمی ممکن است در هر وضعیتی قرار گیرد که احتمال انتشارهای خاص برای آن وجود دارد.زیرا دارای هیچ تعیین شده ی خاصی نیست، دقیقاً همانند انسانی که قبل از تصمیم گیری در مورد یک موضوع خاص، ایده ها و نظرات زیادی در ذهن خود دارد.

* در سیستم pso هر ذره باید در حالت پیوستگی باشد بطوریکه پیوند آن با مجموعه ذرات دیگر تضمین شود و این همان چیزی که در الگوریتم pso وجود دارد و آن را به وضعیت مطلوب یا نیمه مطلوب در می آورد. در سیستم سنتی pso حالت پیوستگی باعث محدود نمودن ذره در یک منطقه ی محصور و محدود می شود.در حالیکه در سیستم pso کوانتومی یک ذره در حالت پیوستگی از طریق فرمول ۷ محاسبه می شود و شکل آن در شکل شماره ۳ نشان داده شده است و می تواند در محدوده ی مورد نظر به هرحالتی ظاهر گردد و می تواند دارای هر احتمال خاصی حتی بدور از LIP قرار گیرد.چنین حالتی ممکن است در وضعیت مطلوب تر از gbest در مجموعه باشد بنابراین می توان گفت در میان مجموع pso کوانتومی شخصی مانند "انیشتن" می تواند متولد شود.

C – بررسی ذرات بر حسب دانش کوانتومی

در pso کوانتومی ، میزان pbest (بهترین حالت p) ثبت می گرددو میزان pbest آن با دیگر ذرات مجاور مقایسه می گردد تا از این طریق gbest بدست آید. سپس نقطه انحراف آن (p) را می توان با فرمول ۵ محاسبه نمود، بنابراین پتانسیل دلتای آن (delta) تعیین می شود تا میزان تمایل ذره مشخص گردد.

به منظور انجام مرحله ی علمی بعدی باید پارامتر L را مورد ارزیابی قرار دهیم. با پارامترL را اصطلاحاً "خلاقیت" و یا "تصور" می نامیم. زیرا این پارامتر خصوصیات علمی ذره را مشخص کرده و دانشی جدید را در مورد ذرات به روی ما باز می کند.در Qpso ،میزان خلاقیت ذره با ارزیابی فاصله بین جریان ذره و LIPآن بدست می آید و این همان چیزی است که معادله شماره ۹ نشان می دهد. بالاخره اینکه ،با در نظر گرفتن حالت تغییر از فضای جستجو به فضای راه حل می

توان به کمک معادله ۱۰، موضعی جدید را یافت (بدست آورد). اگر این موضع و موقعیت جدید بهتر از pbest باشد ، بنابراین دانش جدید جایگزین pbest خواهد شد.

(٤) pso کوانتومی اصلاح شده

همانطور که در بالا گفته شد ،پارامتر L که "خلاقیت" یا "تصور" نیز نامیده می شود تنها پارامتر موجود در الگوریتم Qpso نامیده می شود.روش کنترل L به لحاظ کارایی برای الگوریتم حائز اهمیت است. ارزیابی پارامتر L از طریق فاصله بین ذرات محاسبه می شود که در این حالت فاصله کنونی آنها با LIP درنظر گرفته می شود.

ولی با این حال باید گفت که این شیوه ی کنترل دارای دو نقطه ضعف است، که عبارتند از :

WikiPower.ir

۱) ارزیاب خلاقیت یک شخص توسط اشخاص دیگر LIP غیر منطقی است.

۲) شیوه ی کنترل پارامتر بر اساس سطح موضعی و فردی است که برای LIP یک قطه فرار و متغیر است .که باعث ایجاد حالتی ناپایدار وناهماهنگ است و هنگامیکه اندازه ی مجموعه کوچک باشد، این وضعیت بوجود می آید.

در این بخش ،ما اقدام به پیشنهاد یک روش جدید و نوین برای پارامتر کنترلی می نامیم که بر اساس استاندارد جهانی است.اگر چه در جامعه انسانی تفکرات گوناگونی وجود دارد ،باید یک تفکر مبنا و اصلی وجود داشته باشد که مورد اتفاق نظر عمومی باشد. می توان از این تفکر اصلی و مبنا برای ارزیابی و سنجش خلاقیت افراد استفاده نمود.

برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

در صورتیکه انحراف تفکر یک فرد از تفکر اصلی و مبنا زیاد باشد ، بطوریکه فرد دارای خلاقیت و تصور بیشتری است و بنابراین قادر است که یک دانش جدید و نوین را کشف کند. و برعکس اگر انحراف آن کم باشد این شخص فاقد قضاوت فردی است و نمی تواند خود را با تغییرات و تحولات تفکر اصلی وقف دهد. چنین فردی دارای دامنه و افق محدودی برای جستجوی دانش است و میزان خلاقیت آن ضعیف است.

ما در مدل Qpso اصلاح شده ی خود ،از یک جریان تفکر اصلی و مبنا برای ارزیابی پارامتر L استفاده می کنیم ،(پارامتر L همان خلاقیت فرد است).

نقطه "جریان اصلی تفکر" یا میانگین بهتر وضعیت (که است ،یعنی :

$$mbest \quad \sum_{i\ 1}^{m} pi \Big/ M \qquad \sum_{i\ 1}^{m} pu \Big/ m_i \sum_{i\ 1}^{m} pm \Big/ m_i ..., \sum_{i\ 1}^{m} pm \Big/ M$$

در این معادله M نشان دهنده ی اندازه ی مجموعه و p_i حالت موضع $gbest$ ذره ی i است ، ارزش L از این طریق نشان داده می شود.

$$L(t+1) = 2*\beta*|mbest - x(t)|$$

در این فرمول β ضریب خلاقیت است ،بنابراین می توان معادله ۱۰ را اینگونه بازنویسی نمود:

$$X(t+1) = p \pm \beta*|mbest - x(t)|* \ln(1/U)$$

این معادله در واقع معادله ی تکراری و برگرفته از الگوریتم Qpso می باشد .

اکنون ما pso اصلاح شده را بصورت زیر بیان می نمائیم :

```
Lnitialize population : random  xi
Do
      For   j=1   to population size   m
                  If   f(xi )  <  f(pi)   then   pi  =
                   xi
         Pg  =  min  (pi)
  Find out mbest using
        equation(11)
         For    d = 1   to dimention   d
             Fi1  = rand  (0,1)  ,  Fi2  = rand  (0,1)
            P = (Fi1 * pid + Fi2 * pgd) / (Fi1 +  Fi2)
           U = rand (0,1)
 If  rand (0,1) .0.5
          Xid = p-beta * abs (mbest  d  –  Xid) *
(ln(1/u))
          else
         Xid = p+beta * abs (mbest d – Xid)*(ln(1/u))
Until termination criterion is met
```

تنها پارامتر در الگوریتم "ضریب خلاقیت " می باشد، بتا( )،که میل به خلاقیت و کار بر روی دیگر ذرات (افراد) به سرعت و کارایی الگوریتم را نشان می دهد.

(٦) نتایج آزمایشات عددی

برای ارزیابی Qpso اصلاح شده، از هفت معیار برای مقایسه با spso و pso استفاده شده است که اینگونه بیان می شود:

$$f(x_1) = \sum_{i=1}^{n} x_i^2$$

دومین تابع ،تابع روزنبروک (Rosen brock) نام دارد که اینگونه تعریف می شود:

$$f(x_2) = \sum_{i=1}^{n} 100\,(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$$

سومین تابع ،" تابع راستریگرین" (rasttrigrin) نام دارد که اینگونه مطرح می شود:

$$f(x) = \sum_{i=1}^{n} x_i^2 - 10\cos(2\pi x_i) + 10$$

چهارمین تابع،"تابع گریوانک" (Grienwank) می باشد که بصورت زیر توصیف می شود:

$$f(x_4) = \frac{1}{4000}\sum_{i=1}^{n} (x_1 - 100)^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i - 100}{\sqrt{i}}\right) + 1$$

پنجمین تابع ،"تابع دوجانگ" (De jong) نامیده می شود که اینگونه بیان می گردد:

$$f(x_5) = \sum_{i=1}^{n} i.x_i^4$$

ششمین تابع "واریانت روزنبروک" نامیده می شود(Rosenbrock variant) که بصورت زیر توصیف می شود:

$$f(x_6) \quad 100\, x_i^2 \quad x_1^{\,2} \quad 1 \quad x_1^{\,2}$$

هفتمین تابع ،تابع شافرز (shaffers) نام دارد که بدین شکل قابل تعریف است

$$f(x_7) \quad 0.5 \quad \frac{\sin\sqrt{x^2 \quad y^2}^{\,2} \quad 0.5}{1.0 \quad 0.001\, x^2 \quad y^2}^{\,2}$$

تمامی این توابع ،کوچک کردن مسائل با قدر مطلق صفر می باشند.

در تمامی این تجربه ها و آزمایشات محدوده ی اولیه مجموعه ای که در جدول ۱ به صورت فهرست وار بیان شده اند، بصورت متقارن بیان می گردند.ارزش ثابت مساوی با ارزش تابع بیان می شود، و مجاور هر ذره ،کل مجموعه ذرات است .ما برای هر مورد ۵۰ آزمون انجام داده ایم و بهترین آنها را ثبت کرده ایم .به منظور بررسی دقیق الگوریتم و انعطاف پذیری آن ، برای هر تابع از اندازه ی مجموعه (M) متنوع در ابعاد مختلف استفاده شده است.

اندازه های مجموعه ها عبارتند از ۲۰،۴۰و۸۰ نتایج بر اساس ۱۰۰۰،۱۵۰۰و ۲۰۰۰ تنظیم شده اند.که متناسب با ابعاد ۱۰،۲۰و۳۰ به ترتیب برای پنج تابع اول هستند و ابعاد دو تابع آخر معادل ۲است.ما برای آزمایش الگوریتم اصلاح شده ی Qpso از دو گروه آزمایشات و تجربیات استفاده کرده ایم. در اولین تجربیات Qpso، اصلاح شده،اصطلاحاً RQPSO1 نامیده شد و ضریب خلاقیت آن beta به صورت خطی از

۱,۰تا ۰,۵ کاسته شد و این هنگامیست که الگوریتم مساوی r باشد ،در دومین سری از آزمایشات ،الگوریتم اصطلاحاً RQPSO2 نامیده شد، که در آن ارزش beta بصورت خطی از ۱,۲ تا ۰,۵ کاسته شد و ارزش Lبا Lmax پیوند داده شد. بدین صورت تمامی ذ رات بصورت LL Lmax بوده و اولین مجموعه تعیین کننده ی Lmax است که بدین صورت تعریف می گردد:

Lmax = Max[| mbest (t=0) – gbest (t=0) |]

بهترین ارزش های تناسب برای هر تابع در جدول ۱ تا جدول ۷ نشان داده شده است .ارزش ستون spso وQDpso از جدول ۱ تا جدول ۸ از (۳و۵) گرفته شده است:

WikiPower.ir

www.wikipower.ir
برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

جدول – ۱

آغاز محدوده ی متقارن	تابع
(۱۰۰ و ۵۰)	f1
(۳۰ و ۱۵)	f2
(۵۱۲ و ۲,۵۶)	f3
(۶۰۰ و ۳۰۰)	f4
(۱۰۰ و ۳۰)	f5
(۱۰۰ و ۳۰)	f6
(۱۰۰ و ۳۰)	f7

www.wikipower.ir

برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

جدول ۲ – میانگین ارزش تناسب برای تابع sphere

M	D	Gmax	SPSO	QDPSO	RQPSO1	RQPSO2
20	10	1000	1e-20	1e-25	1e-31	1e-40
	20	1500	1e-11	1e-15	1e-20	1e-23
	30	2000	1e-06	1e-08	1e-11	1e-16
40	10	1000	1e-23	1e-41	1e-62	1e-64
	20	1500	1e-14	1e-23	1e-32	1e-37
	30	2000	1e-10	1e-14	1e-23	1e-26
80	10	1000	1e-28	1e-61	1e-82	1e-85
	20	1500	1e-17	1e-32	1e-50	1e-55
	30	2000	1e-12	1e-19	1e-38	1e-38

WikiPower.ir

برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

جدول ۳ – میانگین ارزش تناسب برای تابع روزنبروگ (Rosenbrock)

M	D	Gmax	SPSO	QDPSO	RQPSO1	RQPSO2
20	10	1000	96.1715	14.2221	13.8377	10.8643
	20	1500	214.6764	175.3186	116.0543	97.9443
	30	2000	316.4468	242.3770	187.1783	135.8685
40	10	1000	70.2139	15.8623	12.9653	10.2468
	20	1500	180.9671	112.4612	87.9421	80.3442
	30	2000	299.7061	76.4273	75.6933	69.2908
80	10	1000	36.2954	36.3405	11.8327	9.8421
	20	1500	87.2802	23.5443	19.7310	17.6420
	30	2000	205.5596	71.9221	58.5165	53.6345

WikiPower.ir

برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

جدول ۴ – میانگین ارزش تناسب برای تابع راستریگرین (Rostrigrin)

M	D	Gmax	SPSO	QDPSO	RQPSO1	RQPSO2
20	10	1000	5.5572	4.9698	4.5712	4.5823
	20	1500	22.8892	17.0789	16.0244	15.2001
	30	2000	47.2941	48.6199	35.2052	33.5101
40	10	1000	3.5623	2.0328	2.0489	2.1459
	20	1500	16.3504	10.9453	10.2717	9.2517
	30	2000	38.5250	21.3712	23.4756	20.8164
80	10	1000	2.5379	0.9232	0.8871	0.7298
	20	1500	13.426	6.9554	7.2781	6.4174
	30	2000	29.3063	18.130	19.9324	17.3473

WikiPower.ir

جدول ۵ – میانگین ارزش تناسب (تناسب ارزش) برای تابع گریوانگ (Griewank)

M	D	Gmax	SPSO	QDPSO	RQPSO1	RQPSO2
20	10	1000	0.0919	0.1003	0.0078	0.0051
	20	1500	0.0303	0.0086	0.0002	0.0002
	30	2000	0.0182	0.0544	0.0011	0.0009
40	10	1000	0.862	0.0484	0.0009	0.0006
	20	1500	0.0286	0.0004	0.0002	0.0002
	30	2000	0.0127	0.0009	0.0001	0.0000
80	10	1000	0.0760	0.0000	0.0000	0.0000
	20	1500	0.0288	0.0000	0.0000	0.0000
	30	2000	0.128	0.0000	0.0000	0.0000

WikiPower.ir

برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

جدول ۶ – میانگین ارزش تناسب برای تابع دی جانگ (De Jung)

M	D	Gmax	SPSO	QDPSO	RQPSO1	RQPSO2
20	10	1000	0.0000	0.0000	0.0000	0.0000
	20	1500	0.0000	0.0000	0.0000	0.0000
	30	2000	0.0000	0.0000	0.0000	0.0000
40	10	1000	0.0000	0.0000	0.0000	0.0000
	20	1500	0.0000	0.0000	0.0000	0.0000
	30	2000	0.0000	0.0000	0.0000	0.0000
80	10	1000	0.0000	0.0000	0.0000	0.0000
	20	1500	0.0000	0.0000	0.0000	0.0000
	30	2000	0.0000	0.0000	0.0000	0.0000

جدول ۷– میانگین ارزش تناسب برای تابع واریانس روزبروک(Variance Rosenbrock)

M	D	Gmax	SPSO	QDPSO	RQPSO1	RQPSO2
20	2	2000	0.0000	0.0000	0.0000	0.0000
40	2	2000	0.0000	0.0000	0.0000	0.0000
80	2	2000	0.0000	0.0000	0.0000	0.0000

جدول ۸- میانگین ارزش تناسب برای تابع شافرز (Shaffers)

M	D	Gmax	SPSO	QDPSO	RQPSO1	RQPSO2
20	2	2000	0.0012	0.0051	7.9437-e4	2.9951-e5
40	2	2000	0.0006	0.0018	1.5385-e5	2.8818-e7
80	2	2000	0.0002	0.0004	8.5111-e7	7.2959-e8

نتایج عددی نشان داد که الگوریتم Qpso بهتر از QDpso و Spsoعمل می کند. با توجه به حالت $LLL_{\max}$ می توان گفت که تمامی توابع بجز تابع راستریگرین (Rastrigrin) بطور قابل قبول و مطلوب عمل می کنند، به نظر می رسد که علت آن، این باشد که محدوده ی "تابع راستریگرین"خیلی کوچکتر است و در نتیجه محدوده ی L دارای کارایی کمتری است.

نتایج ریاضیاتی نشان می دهد که استفاده از تابع شافرز (shaffers) در جدول ۸ در حال از بین رفتن است. این تابع دارای زیر مجموعه ای ۰۰۰۹۷,۰ می باشد. ثابت شده است که سیستم spso قادر است که نسبت به QDpso بهتر می توان از خطا دوری کند. و علت آن هم این است که ذرات موجود در spso بطور مستمر در جستجوهستند. در حالیکه ذرات QDpso اینگونه نیست و نمی تواند نواحی باریک و ظریف را در نظر گیرد در حالیکه ممکن است. موقعیت مطلوب درآنها نهفته باشد. ولی ذرات RQpso می تواند راحت تر از spso و QDpso از مسیرهای باریک عبور کند، بنابراین با توجه به این مطالب باید گفت که با استفاده از شیوه جدید از کنترل پارامتر ، سیستم Qpso به لحاظ توانایی جستجو دارای موقعیت بهتری است.

برای دریافت فایل Wordپروژه به سایت ویکی پاور مراجعه کنید. فاقد آرم سایت و به همراه فونت های لازمه

نتیجه گیری

در این مقاله ،ما بر اساس ،سیستم pso با رفتار کوانتومی ،اقدام به فرمول بندی فلسفه و مبنای pso کوانتوم نموده ایم ،که البته به صورت مفصل مورد بحث قرار گرفته نشده است.سپس ما اقدام به ارائه یک شیوه ی مطلوب از پارامتر کنترل نمودیم و به کمک آن یک روش نوین و اصلاح شده از الگوریتم pso کوانتوم مطرح کردیم. بر اساس توابعی که در این مقاله مطرح شده است ،نوع اصلاح شده ی pso از نوع کلاسیک و قدیمی آن به مراتب بهتر است و نتایج و مشاهدات هم این مطلب را تصدیق می کنند.در نوع Qpso که اصلاح شده است ارزیابی پارامتر L بستگی به شکل قرارگیری جهانی آن دارد ،بهترین میانگین محل (یا همان mbest)که نسبت به رشد جمعیت پایدار است.

بنابراین باید گفت که در این مدل ذرات مجزا و متفرق با کل مجموعه ی ذرات همخوانی و هماهنگی دارند و این توانایی جستجوی الگوریتم را بالا می برد. بعبارت دیگر در این مدل و الگوی اصلاح شده ،تنظیم پارامتر بتا (beta) ،ضریب خلاقیت آسان و ساده است .

برای آنکه بتوانیم pso کوانتومی را بهتر و مطلوب تر نمائیم،تحقیقات حال حاضر ما بر روی یافتن سیستم باز کوانتوم pso متمرکز شده است.در یک سیستم باز ،هنگامیکه ذرات به نقطه ی تعادل می رسند ، آنها از این نقطه حرکت می کنند تا به جستجوی خود ادامه دهند، و این احتمال وجود دارد که بتوانند با حالتی مطلوب دست یابند.بنابراین می توان گفت که سیستم باز Qpso نسبت به Qpso معمولی ،دارای کارایی بهتری می باشد. [7]

مراجع :

[1] مقدمه ای بر مبانی الگوریتم ژنتیک و کاربردهای آن (ترجمه مهندس مهدی علیرضا)

[2] مبانی الگوریتم ژنتیک (مؤلف پیرسکالی)

[3] مجله علم و کامپیوتر www.ccwmagazine.com

[4] www.talkorigins.org

[5] www.wikipedia.com

[6]پروژه ی پایانی Gabriel Kevorkian و Marko Milek

[7] مقاله ی اقایان Jun Sun و Wenbo Xu و Bin Feng گرفته شده از IEEE

WikiPower.ir