



طرح پژوهشی در قالب گرنت

عنوان طرح:

بهینه سازی حمل و نقل با استفاده از الگوریتم های تکاملی

مجری طرح:

محمد محمدی

ناظر طرح:

محمود حاج علیخانی

خاتمه طرح:

تیرماه ۱۳۹۱

این طرح با بهرگیری از اعتبار ویژه پژوهشی دانشگاه پیام نور در قالب اعتبار گرنت تهیه شده است لذا کلیه حقوق این طرح متعلق به دانشگاه پیام نور می باشد.

۱	فصل اول: حمل و نقل
۸	فصل دوم: تعریف و تاریخچه الگوریتم های تکاملی
۱۶	فصل سوم: بررسی روند کلی الگوریتم های تکاملی
۲۲	فصل چهارم: انتخاب
۴۰	فصل پنجم: باز ترکیبی
۵۳	فصل ششم: جهش
۵۸	فصل هفتم: دوباره گنجاندن دادن
۶۴	فصل هشتم: معیار همگرایی یا توقف الگوریتم
۶۶	فصل نهم: احتمال های وفقی عملگرهای باز ترکیبی و جهش
۷۲	فصل دهم: نتیجه گیری و ارائه مدل
۸۴	منابع

فصل اول

حمل و نقل

ترابری یا حمل و نقل (که در انگلیسی **transport** ، خوانده می شود)، جابجایی و انتقال انسان و کالا است از جایی به جای دیگر . این عبارت از کلمات لاتین **trans** به معنای از یک سو به سوی دیگر و **portare** به معنی حمل کردن ریشه گرفته است. رشته حمل و نقل دارای جنبه های گوناگونی است : در یک طبقه بندی کلی می توان آن را به سه بخش زیر ساخت ، وسایل نقلیه و بهره برداری تقسیم نمود. زیر ساخت شامل شبکه های حمل و نقل (جاده ها ، خطوط راه آهن ، راه های هوایی ، راه های آبی ، خطوط لوله حمل مواد ، غیره) مورد استفاده و همچنین گره ها یا پایانه ها (مانند فرودگاه ها ، ایستگاه های راه آهن ، ایستگاه اتوبوس ، بنادر . وسایل نقلیه عموماً در شبکه ها حرکت می کنند، مانند اتومبیل ها ، قطارها و هواپیماها). بهره برداری ، شامل فعالیتهایی است که کنترل سیستم را انجام می دهند. (مانند : چراغ های راهنمایی و شیب سنج ها ، سوئیچ های راه آهن ، کنترل ترافیک هوایی ، غیره و همچنین سیاستهایی مثل چگونگی اداره امور مالی حمل و نقل) مانند استفاده از عوارض جاده ای یا مالیات سوخت در حمل و نقل بزرگراهی در حالت کلی ، طراحی شبکه در حوزه مهندسی عمران و برنامه ریزی شهری ، طراحی وسایل نقلیه ، در حوزه مهندسی مکانیک و زیر شاخه های تخصصی آن مانند مهندسی دریایی و مهندسی هوا-فضا و بهره برداری ها نیز معمولاً تخصصی شده اند. هرچند که ممکن است متعلق به پژوهش بهره برداری یا مهندسی سیستم ها باشد. حمل و نقل و **ارتباطات** جایگزین و مکمل یکدیگرند. با اینکه ممکن است ارتباطات کاملاً پیشرفته بتواند جایگزین حمل و نقل شود و شخص به جای دیدن از نزدیک کسی به او تلگراف ، تلفن ، فکس یا نامه الکترونیک بزند ، ثابت شده است که این مدهای ارتباطی در واقع در کل اثرات متقابل بیشتری را ایجاد می کنند. پیشرفت در حمل و نقل بدون **ارتباطات** غیر ممکن است. ارتباطات برای سیستمهای حمل و نقلی پیشرفته حیاتی و ضروری اند. از خطوط راه آهن دارای یک خط که می خواهند در هر دو جهت از آنها قطار عبور کند تا کنترل خطوط هوایی که

همواره باید از محل هواپیما در آسمان اطلاع داشته باشد. بنابراین ، مشاهده شده که افزایش و پیشرفت یکی ، منجر به توسعه بیشتر دیگری می گردد.

۲-۱ برنامه ریزی خطی مدل حمل و نقل

برنامه ریزی خطی شاخه ای از برنامه ریزی ریاضی است که موثرترین روشهای اختصاص دادن منابع محدود برای فعالیت های شناخته شده ای برای برآورد هدف خاص مانند ماکزیمم نمودن سود یا مینیمم نمودن هزینه ها سروکار دارد. از ویژگی های خاص برنامه ریزی خطی تابعی است که هدف برنامه ریز را مشخص می کند و به همراه آن قیودی است که تابع هدف در قالب آن سنجیده و مشخص می باشد که همگی خطی می باشند. برنامه ریزی خطی با بهینه سازی یک تابع خطی که از محدودیت های مساوی و یا نامساوی یا ضمنی تشکیل شده است. مسائل برنامه ریزی خطی را ابتدا جرج بی. دانتزیک در حدود سال ۱۹۴۷ ابداع کرد و سپس در سال ۱۹۴۹ ، روش سیمپلکس را برای حل مسائل برنامه های خطی ارائه کرد و از آن زمان به بعد افراد زیادی به روشهای بسیار متعددی از جمله بسط و توسعه نظری، دیدگاه محاسباتی و بکارگیری کاربردهای جدید آن در این حوزه وارد شده اند. مسائل حمل و نقل رده مهمی از مسائل برنامه ریزی جریان های شبکه ای هستند که به آن مدل ترابری یا حمل و نقل گفته می شود. در این مدل هدف کمینه سازی هزینه های حمل و نقل می باشد. در واقع حمل و نقل کالا از تعدادی مبدا به چندین مقصد که در آن مقادیر عرضه در هر مبدا و مقادیر تقاضا در هر مقصد معلومند.

۳-۱- مسئله برنامه ریزی خطی - مسئله حمل و نقل

ابتدا لازم است مقدمه ای از مسائل حمل و نقل که حالتی از مسائل برنامه ریزی خطی می باشد بیان شود، سپس با توجه به ویژگی های خاصی که مسائل حمل و نقل دارند، مدل ویژه ای از الگوریتم های تکاملی را که منطبق با اینگونه مسائل می باشند، ارائه می گردد. مسائل حمل و نقل رده مهمی از مسائل برنامه ریزی جریان های شبکه ای هستند که به آن مدل حمل و نقل گفته می شود. در

این مدل هدف کمینه سازی هزینه های حمل و نقل می باشد. در واقع حمل و نقل کالا از تعدادی مبدا به چندین مقصد که در آن مقادیر عرضه در هر مبدا و مقادیر تقاضا در هر مقصد معلومند. در آغاز لازم است مسائل ترابری را مورد بررسی قرار گیرد، فرض کنید مسئله ترابری دارای m کارخانه به عنوان مبدا و n انبار به عنوان مقصد باشد و همچنین a_i میزان تولید و عرضه کالا در کارخانه i ام ($i=1,2,\dots,m$) و b_j میزان کالا مورد تقاضا در انبار j ام ($j=1,2,\dots,n$) و c_{ij} هزینه ترابری یک واحد کالا در مسیر (i,j) که مبدا i را به مقصد j وصل می کند، باشد [۱]. هدف مسئله عبارت است از پیدا کردن میزان کالائی که باید از مبدا i به مقصد j ارسال شود تا هزینه های ترابری کمینه شود. حال با توجه به این فرضیات مسئله، اگر x_{ij} میزان کالائی باشد که باید از مبدا i به مقصد j ارسال شود در این صورت مدل برنامه ریزی خطی اینگونه مسائل بصورت زیر می باشد.

$$\begin{aligned} \min \quad & z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ \text{st} \quad & \sum_{j=1}^n x_{ij} = a_i \quad (i = 1, 2, \dots, m) \\ \text{st} \quad & \sum_{i=1}^m x_{ij} = b_j \quad (j = 1, 2, \dots, n) \end{aligned}$$

نکته ای که لازم است در حل مسائل حمل و نقل مورد نظر داشت، اینست که مدل حمل و نقل باید متوازن باشد. مدلی را متوازن گویند هرگاه:

$$\sum_{j=1}^n b_j = \sum_{i=1}^m a_i$$

بطور خلاصه مدلی را متوازن گویند هرگاه مجموع عرضه های تمام مبداها باید برابر با مجموع تمام تقاضاهای مقصدها باشد. ولی در عمل اینگونه نیست. مسائلی که متوازن نیستند را مسائل نامتوازن گوئیم. برای تبدیل مسائل نامتوازن به مسائل متوازن دو حالت وجود دارد که می توان با تغییرات مناسب آنها را به حالت متوازن تبدیل کرد [۲].

فصل دوم

تعریف و تاریخچه الگوریتم‌های تکاملی

۲-۱- مقدمه

الگوریتم‌های تکاملی [۳] شاخه‌ای از تکنیک‌های بهینه‌سازی اجتماعی هستند که در آنها از سیستم‌های بیولوژیکی و اصول حاکم بر آنها بهره‌گیری می‌شود. اگر چه این الگوریتم‌ها ارائه دهنده‌ی مدل‌های ساده‌ای از فرایندهای بیولوژیکی در شرایط واقعی هستند اما در عمل توانایی و کارایی زیادی را از خود نشان داده‌اند. ایده اولیه‌ی ارائه الگوریتم‌های تکاملی استفاده از جمعیت محدودی از عناصر است، به طوری که هر یک از آنها دقیقاً نقطه‌ای از فضای جستجو را مشخص می‌کنند. پس از ایجاد، این جمعیت اولیه در یک فرایند خودسامانی توسط فرایندهای آماری توسعه می‌یابد و در جهت یافتن مناطق بهتر و بهینه‌تر از فضای جستجو با یکدیگر ترکیب می‌شوند. فرایند انتخاب، عناصری را از لحاظ اطلاعات ژنتیک جدید تأمین کرده و عملگر بازترکیبی نیز اطلاعات را با هم ترکیب و کنترل می‌کند.

الگوریتم‌های تکاملی یکی از سیستم‌های حل مسئله با استفاده از کامپیوتر که از بعضی از مدل‌های تکاملی الگوبرداری شده است. [۷].

الگوریتم‌های تکاملی دارای انواع مختلف است که مهمترین آنها عبارتند از [۳]:

۱- برنامه تکاملی [Evolutionary programming (Fogel, ۱۹۶۶)]

۲- الگوریتم‌های ژنتیک [Genetic algorithm (Holland, ۱۹۷۵)]

۳- استراتژی‌های تکامل [Evolution strategic (Rechenberg, ۱۹۷۳)]

انواع الگوریتم‌های تکاملی به طور مفهومی تکامل ساختارهای افراد در طبیعت بر طبق روندهای انتخاب، جهش و باز ترکیبی می‌باشد. این عملگرها قابلیت ذخیره‌سازی ساختارهای افراد که به وسیله محیط تعریف شده، را دارا می‌باشند.

هر فرد در جامعه یک اندازه برازندگی در محیط دریافت می‌کند. کانون توجه عملگر بازترکیبی روی افراد با برازندگی بالا می‌باشد. بازترکیبی و جهش باعث آشفستگی می‌شوند.

الگوریتم‌های تکاملی اگرچه در بیشتر سطح‌ها با هم شبیه هستند ولی در روش انجام با هم متفاوت هستند. این اختلاف بیشتر روی ویژگی‌های الگوریتم‌های تکاملی مشاهده می‌شود که شامل انتخاب اطلاعات برای ساختار افراد و انواع مکانیسم انتخاب زیرا اندازه قابلیت عملگرهای مختلف باهم متفاوت هستند.

انواع دیگر از الگوریتم‌های تکاملی که توسعه یافته‌تر از بقیه انواع دیگر است، نیز وجود دارد که عبارتند از [۱]:

- سیستم‌های طبقه‌بندی [Classifier systems (Holland, ۱۹۸۶)]

- برنامه ژنتیک [Genetic programming (Garis, ۱۹۹۰; Koza, ۱۹۹۱)]

- سیستم‌های LS [LS systems (Smith, ۱۹۸۳)]

- سیستم‌های عملگر وفقی [Davis, ۱۹۸۹] "adaptive operator" systems
پایه‌های زیست‌شناسی [۳]

برای درک درست الگوریتم‌های تکاملی کافی است که روندهای زیست‌شناسی که عملگرهای الگوریتم‌های تکاملی براساس آن الگوبرداری شده است را مورد توجه قرار دهیم. ابتدا ما باید در نظر بگیریم که تکامل (در طبیعت یا در هر چیز دیگر) یک روند جهت‌دار و با قصد و هدف می‌باشد. تکامل وسیله انتخاب طبیعی یا رقابت افراد مختلف برای منابع محدود در طبیعت است. افراد بهتر باقی می‌مانند و ژنهای آنها توسعه پیدا می‌کنند. باز ترکیبی جنسی به بعضی از کروموزوم‌ها اجازه تولید فرزند را می‌دهد. از نظر ملکولی باز ترکیبی عبارت است از تغییر یا جابجائی قسمتی از کروموزوم به کروموزوم دیگر است. این عملگر باز ترکیبی از عمل crossover که باعث تغییر ژنها بین دو کروموزوم در طبیعت می‌شود الگوبرداری شده است. عمل باز ترکیبی در محیطی رخ می‌دهد که ابتدا افراد خوب انتخاب می‌شوند و سپس با هم ترکیب می‌شوند تا افراد با برازندگی بالا تولید کند. اغلب الگوریتم‌های تکاملی از یک تابع ساده به عنوان تابع برازندگی افراد برای انتخاب والدین (به طور تصادفی) که تحت تأثیر عملگرهای ژنتیک قرار گیرند مانند crossover استفاده می‌کنند. تکامل نیاز به گوناگونی و تنوع در حین کار دارد. در طبیعت، یک منبع ایجادکننده گوناگونی مهم، جهش است. اهمیت جهش به خاطر ایجاد تنوع زیاد می‌باشد. برنامه کلی الگوریتم‌های تکاملی به صورت زیر می‌باشد:

Algorithm EA is:

```
//start with an initial time  
t:=0;  
// initialize a usually random populayion of individuals  
initpopulation p(t);  
// evaluate fitness of all initial individuals in population  
evaluate p(t);  
//test for termination crierion (time, fitness,etc)  
While not done do  
//increase the time counter  
t:=t+1;  
// select sub-population for offspring production  
p' :=select parents p (t);  
// recombine the "genes" of selected parents  
recombine p' (t);
```

```

// perturb the mated population stochastically
mutate p' (t);
// evaluate its new fitness
evaluate p'(t);
// select the survivors from actual fitness
p: =survive p,p'(t);
do
end EA.

```

۲-۲ برنامه‌ریزی تکاملی

برنامه‌ریزی تکاملی [۴] و [۵]، که به منظور دستیابی به اطلاعات لازم برای ماشین ایجاد شده، عمدتاً الگویی را مورد استفاده قرار می‌دهد که با دامنه‌ی مسئله تناسب داشته باشد. فرض کنید جمعیتی به بزرگی N داشته باشیم که همه اعضای آن به عنوان والدین انتخاب می‌شوند و همچنین الگوی عملگر جهش مشخصی برای تولید N نوزاد مورد استفاده قرار می‌گیرد. سپس همان N نوزاد تکامل خواهند یافت و نسل بعد با استفاده از یک تابع احتمال برازندگی برای تعداد $2N$ عضو انتخاب خواهد شد. عملگر جهش در برنامه‌ریزی تکاملی معمولاً دارای قدرت تطابق با مسئله است.

Algorithm Ep is:

```

// start with an initial time
t: =0;
// initialize a usually random population of individuals
Initpopulation p(t);
// evaluate p(t);
// test for termination criterion (time, fitness, etc.)
While not done do
// perturb the whole population stochastically
p' (t): = mutate p(t);
//evaluate p' (t);
//stochastically select the survivors from actual fitness
p (t+1): =survive p (t), p' (t);
// increase the time counter
t:= t+1
do
end EP.

```

۲-۳- الگوریتم‌های ژنتیک

الگوریتم‌های ژنتیک [۴] یک مدل از الگوریتم‌های تکاملی است که رفتارشان از مکانیسم‌های تکاملی در طبیعت الگوبرداری شده است به طوری که مکانیسم افراد جامعه به وسیله کروموزوم مشخص می‌شوند. جمعیت کروموزوم‌ها (افراد جامعه) بعد از این وارد مرحله شبیه تکامل می‌شوند. الگوریتم‌های ژنتیک در سطح‌های مختلف کاربرد دارند. یک مثال از کاربردهای آنها برای مسائل بهینه‌سازی چندبعدی که کروموزوم‌های آن به وسیله‌ی مقادیر مختلف کدبندی می‌شوند تا بهینه شوند.

در عمل، محاسبات مدل ژنتیک روی آرایه‌های بیتی یا کارکترهایی که کروموزوم را مشخص می‌کند، انجام می‌شود. عملگرهای انجام کار روی یک بیت ساده اجازه انجام کار توسط عملگرهای crossover و جهش و دیگر عملگرها را می‌دهد.

الگوریتم‌های ژنتیکی به طریق و روش دوره‌ای زیر عمل می‌کنند:
ابتدا به طور تصادفی جامعه‌ای از کروموزوم‌ها پدید می‌آوریم و سپس برازندگی تمام کروموزوم‌های (افراد) داخل جامعه محاسبه و تعیین می‌شود. به وسیله عملگرهای crossover و جهش و دیگر عملگرها جامعه‌ای جدید به وجود می‌آوریم. تکرار یک بار حلقه بالا باعث به وجود آمدن یک نسل می‌شود. در هر بار انجام حلقه، از جامعه قبلی صرفه‌نظر می‌شود و به جای آن جامعه جدید قرار می‌گیرد. هیچ دلیل برای این که، این مدل انجام می‌شود، وجود ندارد در حقیقت ما این مدل را مانند یک اصل و قانون نمی‌بینیم اما آن یک مدل و نسخه مناسب است.
نسل اول (نسل صفر) در حقیقت به طور تصادفی انتخاب می‌شوند و سپس با توجه به برازندگی افراد (کروموزوم‌ها) و عملگرهای موجود جامعه به طرف برازندگی بالا سوق پیدا می‌کند.
برنامه کلی الگوریتم‌های ژنتیک:

Algorithm GA is:

```
//start with an initial time  
t:=0;  
//initialize a usually random population of individuals  
Initialize a usually random population of individuals  
Initpopulation p(t);  
//evaluate fitness of all initial individuals of population  
Evaluate p (t);  
//test for termination criterion (time, fitness, etc)  
While not done do  
    //increase the time counter  
    T: =t+1;
```

```

// select a sub-population for offspring production
p' :=selectparents p(t);
//recombine the "genes" of selected paterns
Recombine p'(t);
//perturb the mated population stochastically
mutate p'(t);
//evaluate its new fitness
evaluate p'(t);
// select the survivors from actual fitness
P:=survive p,p'(t);
Do
End GA.

```

۲-۴- استراتژی‌های تکاملی

یکی دیگر از انواع الگوریتم‌های تکاملی استراتژی‌های تکاملی [۳] و [۱] است. استراتژی‌های تکاملی الگوریتم‌هایی هستند که از قواعد تکامل طبیعی به عنوان یک راه حل در مورد مسائل بهینه‌سازی پارامتری استفاده می‌کنند. این استراتژی‌ها در دهه‌ی ۶۰ میلادی در آلمان رواج یافتند که در ادامه مورد بحث قرار می‌گیرد:

در سال ۱۹۶۳ دو دانشجوی دانشگاه فنی برلین به طور مشترک آزمایش‌هایی را انجام می‌دادند. آنها برای انجام این آزمایش‌ها از تونل هوای بخش مهندسی سیالات استفاده می‌کردند. در حین تحقیقات برای ایجاد سازه‌های بهینه در یک جریان هوا، این ایده به ذهن دانشجویان خطور کرد که این مسئله را به عنوان یک استراتژی مطرح کنند. اما تلاش‌هایی که برای استراتژی ساده و مشابه به کار گرفته شدند، همگی ناموفق بودند. سپس یکی از دانشجویان به نام ایگنورچنبرگ که اکنون بیونیک و مهندسی تکامل است، به این فکر افتاد که تغییرات تصادفی پارامترهای معرف سازنده را مورد آزمایش قرار دهد و مثال‌های جهش‌های طبیعی را پیگیری کند بدین ترتیب استراتژی تکاملی شکل گرفت.

استراتژی‌های تکاملی ابتدایی، تقریباً همانند برنامه‌های تکاملی بودند که در آنها از یک شیوه‌ی عددی متغیر استفاده می‌شد و عملگر جهش صرفاً یک عملگر ترکیبی به شمار می‌رفت. آنها در بسیاری از مسائل بهینه‌سازی کاربرد دارند و پارامترهای آنها مدام قابل تغییر می‌باشد. اخیراً این استراتژی‌ها برای مسائل دیگر هم به کار رفته شده‌اند که اجزای اصلی این استراتژی عبارتند از:

برازندگی

برازندگی یک عنصر، میزانارزشی است که آن عنصر تحت تابع هدف موردنظر دارد. بهترین عناصر آنهایی هستند که با استفاده از آنها تابع هدف به سمت بهینه‌ترین مقدار میل می‌کند.

جمعیت اولیه

جمعیت اولیه یا جهش از یک نقطه دیگر حرکت می‌کند که خود این نقطه ممکن است یا به طور تصادفی و یا به طور تقریبی از اطلاعات کلی از منطقه‌ای که جواب در آن واقع شده، انتخاب گردد.

جهش

در استراتژی‌های تکاملی عملگر جهش، عملگر اصلی است و نقش ابداع کننده در جمعیت را دارا می‌باشد. این تغییرات در پارامترهای استراتژی توسط عملگر جهش را «خودسازگاری» نامیده‌است.

استراتژی تکاملی بر پایه جمعیت‌های تک نقطه‌ای

اولین استراتژی تکاملی، بر پایه‌ی جمعیت‌های تک‌عضوی بنا شده است. در آن تنها از یک عملگر ژنتیکی در فرایند تکامل استفاده شده بود و آن جهش می‌باشد. اما نکته جالب توجه آن بود که یک عضو به صورت یک بردار با دو مولفه نشان داده می‌شود، یعنی $V = (x, \delta)$. اولین مؤلفه (X) نمایانگر یک نقطه در فضای جستجو است و دومین مؤلفه، انحراف معیاری برداری است: با جایگزینی X توسط رابطه‌ی $x^{t+1} = x^t + N(0, \delta)$ جهش به وجود می‌آید، که $N(0, \delta)$ یک بردار از اعداد تصادفی و مستقل نرمال با مقدار متوسط صفر و انحراف معیار δ می‌باشد. نوزاد به وجود آمده (عضو جهش‌یافته)، به عنوان یک عضو جدید در جمعیت پذیرفته می‌شود و اگر برازندگی بهتری داشته باشد و همه‌ی قیدها را در صورت وجود ارضاء کند، جایگزین والدین خود می‌گردد. برای مثال، اگر F، یک تابع هدف و بدون قید باشد و بخواهد ماکزیمم گردد، یک نوزاد (x^{t+1}, δ) جایگزین والدین خود (x^t, δ) می‌گردد، اگر $F(X^{t+1}) > F(X^t)$ باشد. در غیر این صورت، نوزاد حذف می‌گردد و جمعیت بدون تغییر باقی می‌ماند. علی‌رغم این واقعیت که جمعیت تنها شامل یک عضو می‌باشد که جهش در مورد آن اعمال می‌شود، استراتژی تکاملی با عنوان «استراتژی تکاملی دو عضوی» نامیده می‌شود. دلیل این مسئله این است که نوزاد با والدین خود رقابت می‌کند.

استراتژی تکاملی چندعضوی

استراتژی تکاملی چندعضوی با استراتژی دو عضوی از لحاظ تعداد جمعیت تفاوت دارد (تعداد جمعیت < ۱). مشخصات دیگر استراتژی تکاملی چندعضوی عبارتند از: همه‌ی اعضای جمعیت، دارای احتمال یکسانی هستند. امکان معرفی یک عملگر ترکیبی با استفاده از دو نفر از والدین که به طور تصادفی انتخاب می‌شوند وجود دارد. برای تولید یک نوزاد، سیستم، مراحل مختلفی را طی می‌کند که بدین صورت بیان می‌گردد: ابتدا دو عضو انتخاب شده و عملگر ترکیب بر روی آنها به کار گرفته شده، نوزاد جدید به دست می‌آید سپس عملگر جهش را در مورد نوزاد به کار می‌بریم. استراتژی‌های تکاملی در مسائل عددی به خوبی کار می‌کنند چرا که آنها به مسائل بهینه‌سازی توابع، اختصاص دارند. آنها نمونه‌هایی از برنامه‌های تکاملی هستند که از ساختارهای اطلاعاتی مناسب و عملگرهای ژنتیکی برای مسائل بهره می‌گیرند.

فصل سوم

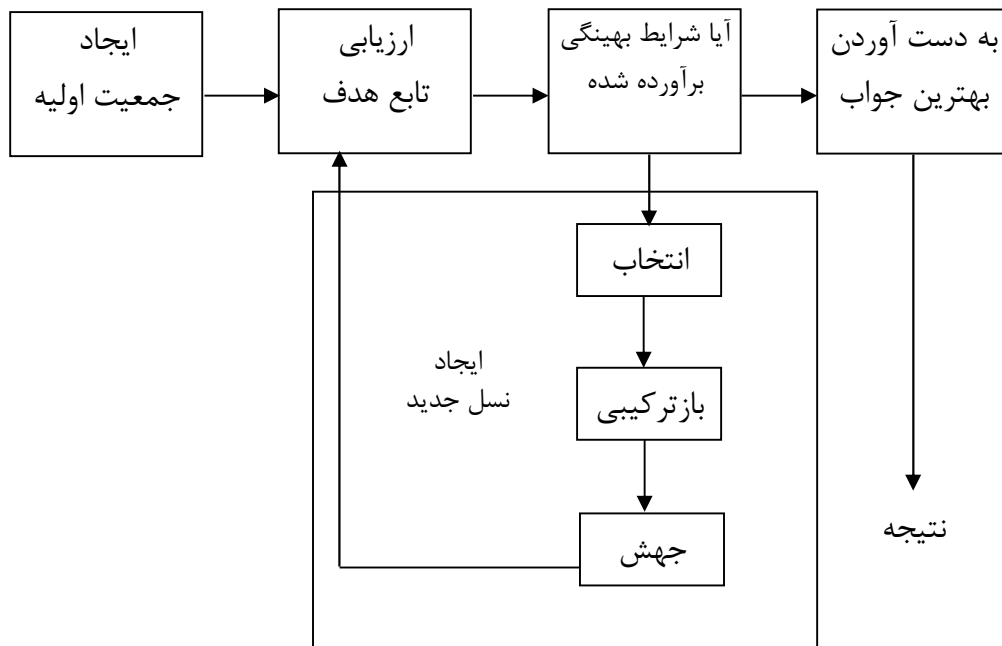
بررسی روند کلی الگوریتم‌های تکاملی

الگوریتم‌های تکاملی [۱]، روش‌های جستجوی تصادفی هستند که از روند تکامل در طبیعت تقلید و الگوبرداری شده است. این الگوریتم‌ها بیشتر برای مسائل بهینه‌سازی کاربرد دارد. و دارای این قابلیت می‌باشند که جوابی بهتر برای مسئله تولید می‌کند. روش کلی الگوریتم‌های تکاملی بدین صورت است که ابتدا برای مسئله به طور تصادفی جواب‌های اولیه که در شرایط مسئله صدق می‌کند در نظر می‌گیرد و سپس با توجه به عملگرهای موجودی الگوریتم‌های تکاملی جواب‌ها را به سوی جواب بهتر سوق می‌دهیم.

در هر نسل یک مجموعه جدید از جواب‌ها به وسیله روندهای انتخاب افراد مطابق سطح برازندگی‌شان در دامنه مسئله، و جفت‌گیری کردن آنها با یکدیگر با استفاده از عملگرهایی که از ژنتیک طبیعی و روند تکامل طبیعی، الگوبرداری شده است، تولید می‌شود. این روندها به تکامل جمعیت‌های افراد با برازندگی بهتر از افراد دیگر در محیطشان منتهی می‌شود.

عملگرهای مدل طبیعی الگوریتم‌های تکاملی عبارت است از: انتخاب، باز ترکیبی و جهش شکل زیر ساختار یک الگوریتم تکاملی ساده را نشان می‌دهد. الگوریتم‌های تکاملی روی یک جمعیت از افراد (راه حل) به جای یک فرد و یک راه حل اعمال می‌شود.

شکل صفحه بعد روند کلی الگوریتم‌های تکاملی را نشان داده است. همانطور که در شکل مشخص است ابتدا یک جمعیت اولیه که شامل جوابهایی برای مسئله می‌شود ایجاد می‌کنیم لازم به ذکر است که این تعداد اعضای این جمعیت اولی در طول مراحل الگوریتم‌های تکاملی با ثابت باشد. بعد از اینکه جامعه اولیه جوابها ارزیابی می‌شود و سپس معیارهای بهینگی بررسی می‌شوند. اگر معیارهای بهینگی برآورد شوند که الگوریتم متوقف می‌شود و بهترین جواب بدست می‌آید در غیر این صورت عملکرد مختلف بر روی جواب‌های داخل جمعیت اعمال می‌شود. (روند کلی عملگرها در فصل‌های آتی به طور کامل مورد بررسی قرار می‌گیرد). سپس جمعیت جدید با همان تعداد عضو ایجاد می‌شود. دوباره این جوابها ارزیابی می‌شود و مانند حالت قبل شرایط بهینگی بررسی می‌شود. این مراحل تا برآورد شدن شرایط بهینگی ادامه پیدا می‌کند. شکل صفحه بعد تمام این روند را نشان می‌دهد.



در شروع روند کلی الگوریتم‌های تکاملی، جمعیت اولیه به طور تصادفی ایجاد می‌شوند و سپس تابع هدف را برای هر یک از افراد جامعه محاسبه و ارزیابی می‌شود. به این طریق است که نسل اول تولید می‌شود. حال اگر معیارهای بهینگی برای نسل کنونی برآورد شود، نتیجه می‌شود این نسل بهینه است و در غیر این صورت یک نسل جدید تولید می‌شود. بدین صورت که افراد طبق برزندگی‌شان انتخاب می‌شوند و سپس با هم ترکیب می‌شوند (والدین با هم ترکیب می‌شوند و فرزند تولید می‌کنند) سپس فرزندان تحت عمل جهش قرار می‌گیرند. فرزندان در جامعه به جای والدین جایگزین می‌شوند و نسل جدید به وجود می‌آید و دوباره مثل قبل برزندگی افراد محاسبه و ارزیابی می‌شود. این چرخه تا برآورد شدن معیارهای بهینگی ادامه پیدا می‌کند.

از بحث بالا اختلاف الگوریتم‌های تکاملی با دیگر روش‌های بهینه سازی و جستجوهای قدیمی مشخص می‌شود.

مهمترین اختلافات عبارتند از:

- الگوریتم‌های تکاملی، جستجو روی یک جمعیت از نقاط به طور موازی انجام می‌دهد و نه تک نقطه‌ای
- الگوریتم‌ها تکاملی نیاز به اطلاعات زیاد و معین ندارد تنها نیاز به تابع هدف دارد.
- الگوریتم تکاملی، برای کاربران بسیار ساده و قابل فهم می‌باشند.
- الگوریتم تکاملی، تعداد زیادی جواب برای یک مسئله بدست می‌دهد و از بین این جوابها، جواب نهایی که شرایط بهینگی را برآورده، مورد استفاده قرار می‌گیرد.

در ادامه این فصل، لیست بعضی از روش‌ها و عملگرها الگوریتم‌های تکاملی آورده شده است و توضیحات بیشتر در فصل‌های آتی آورده شده است.

انتخاب

عملگر انتخاب تعیین می‌کند که کدام یک از افراد برای جفت‌گیری (باز ترکیبی) انتخاب شود و این عمل جفت‌گیری، چه تعداد فرزند بوجود می‌آید.

والدین طبق برازندگی‌شان به وسیله یکی از روش‌های زیر انتخاب می‌شود:

- انتخاب بر اساس رتبه‌بندی

- انتخاب چرخ‌گردان

- انتخاب بر اساس نمونه تصادفی

- انتخاب محلی

- انتخاب هرس

- انتخاب مسابقه‌ای

برای کسب اطلاعات بیشتر به فصل ۴ مراجعه کنید.

باز ترکیبی

باز ترکیبی، افراد جدیدی را از ترکیب اطلاعات افراد انتخاب شده (والدین) به وجود می‌آورد. این عمل توسط ترکیب مقادیر متغیرهای والدین انجام می‌شود و فرزند به وجود می‌آید. انواع روش‌های باز ترکیبی عبارت است از:

- باز ترکیبی مجزا

- باز ترکیبی میانی

- باز ترکیبی خطی

- باز ترکیبی خطی توسعه یافته

- عملگر crossover تک نقطه‌ای / دونقطه‌ای / چند نقطه‌ای

- عملگر crossover یک شکل

- عملگر crossover بر زنده

- عملگر crossover با کاهش جایگزینی

باز ترکیبی‌های میانی و خطی و خطی توسعه یافته روی مقادیر حقیقی عمل می‌کند و باز ترکیبی‌های که روی متغیرهایی با مقادیر باینری عمل می‌کنند به عملگر crossover مشهور است

برای کسب اطلاعات بیشتر به فصل ۵ مراجعه کنید.

جهش

بعد از باز ترکیبی هر فرزند باید جهش را تحمل کند و این مرحله را طی کند. عملگر جهش باعث تغییراتی در هر فرزند می‌شود که این تغییرات خیلی کوچک می‌باشند دو نوع عملگر جهش وجود دارد:

- جهش مقادیر حقیقی

- جهش مقادیر باینری

برای کسب اطلاعات بیشتر به فصل ۶ مراجعه کنید.

دوباره گنجاندن ، دوباره جا دادن

وقتی که فرزندی توسط عملگرهای انتخاب، باز ترکیبی و جهش از والدین‌شان به وجود می‌آید. برازندگی افراد و فرزندان بسیار مهم و تعیین کننده است و باعث حذف شدن تعدادی از فرزندان می‌شود. اگر تعداد فرزندان تولید شده کمتر از تعداد افراد جامعه عمومی باشد در این صورت برای رسیدن به آن تعداد کافی است که افراد جمعیت قبلی را داخل جامعه فعلی دوباره بگنجانیم. دوباره گنجاندن روش‌های مختلف دارد که عبارتند از:

- دوباره گنجاندن عمومی و کلی

- دوباره گنجاندن محلی

برای کسب اطلاعات بیشتر به فصل ۷ مراجعه کنید.

فصل چہارم

انتخاب

۴-۱- انتخاب بر اساس رتبه‌بندی

۴-۲- انتخاب چرخ گردان

۴-۳- انتخاب بر اساس نمونه تصادفی

۴-۴- انتخاب محلی

۴-۵- انتخاب هرسی

۴-۶- انتخاب مسابقه‌ای

۴-۷- مقایسه الگوهای مختلف انتخاب

انتخاب

در قسمت انتخاب، والدین بر اساس نسبت سازگاریشان انتخاب می‌شوند. به هر فرد در قسمت انتخاب یک احتمال تکثیر که وابسته به تابع احتمال است، اختصاص می‌یابد و مقدار تابع هدف همه افراد محاسبه می‌شود و در قسمت انتخاب قرار می‌گیرد. قبل از آنکه وارد بحث شویم اصطلاحات زیر را تعریف می‌کنیم. این اصطلاحات برای مقایسه روش‌های مختلف انتخاب کاربرد فراوان دارند.

۱- فشار انتخابی یا تمرکز انتخابی:

احتمال اینکه بهترین افراد در مقایسه با میانگین احتمال انتخاب همه افراد، انتخاب شوند.

۲- اریب:

اختلاف بین نسبت سازگاری نرمال شده افراد و درجه احتمال تکثیر آنها.

۳- گسترش، انتشار:

تعداد فرزندی که از والدین به وجود می‌آیند.

۴- درصد فقدان وعدم گوناگونی:

تعداد افرادی از جامعه که در طول انتخابمان، انتخاب نمی‌شود.

۵- شدت انتخاب:

میانگین برازندگی افراد جامعه که بعد از کاربرد متد انتخاب در توزیع گوسین نرمال شده انتظار می‌رود.

۶- پراکندگی انتخاب (واریانس انتخاب):

میزان پراکندگی، برازندگی افراد جامعه که بعد از کاربرد متد انتخاب در توزیع گوسین نرمال شده انتظار می‌رود.

۴-۱- انتخاب بر اساس رتبه بندی

در انتخاب بر اساس رتبه بندی، افراد جامعه مطابق تابع هدف دسته‌بندی می‌شوند و برازندگی هر فرد تنها وابسته به موقعیت‌شان در بین افراد جامعه دارد. انتخاب بر اساس رتبه‌بندی باعث همگرایی خیلی سریع می‌شود و عمل جیتجو را کاهش می‌دهد. در انتخاب بر اساس رتبه بندی، میزان تکثیر محدود است و هیچ کدام از والدین فرزند زیادی تولید نمی‌کنند. رتبه دهی به افراد بر اساس میزان سازگاری‌شان، یک راه ساده و موثر برای کنترل تمرکز انتخاب ارائه می‌دهد.

روش انتخاب

فرض کنید که تعداد افراد جامعه برابر N باشد و رتبه یک فرد در این جامعه برابر P باشد. (عدد P از ۱ شروع می‌شود و به N ختم می‌شود فرد با کمترین برازندگی دارای $P=1$ و مناسبترین فرد دارای رتبه $P=N$ می‌باشند) و sp معیار انتخابی است. رتبه گذاری برای هر فرد مانند روش‌های زیر محاسبه می‌شود:
الف: رتبه گذاری خطی:

$$Fitness(P) = 2 - sp + 2 * (sp - 1) * (p - 1) / (n - 1)$$

Sp متغیری است که به طور تصادفی در فاصله $[1, 2]$ انتخاب می‌کنیم $sp \in [1, 2]$
ب) رتبه‌گذاری غیر خطی: استفاده از رتبه‌گذاری غیر خطی اجازه انتخاب sp بزرگتر از روش رتبه‌گذاری خطی را می‌دهد.

$$Fitness(p) = N * x^{p-1} / \sum_{i=1}^N x^{i-1}$$

که در آن x ریشه چند جمله‌ای زیر است.

$$(sp - 1) * x^{n-1} + sp * x^{N-2} + sp * x + sp = 0$$

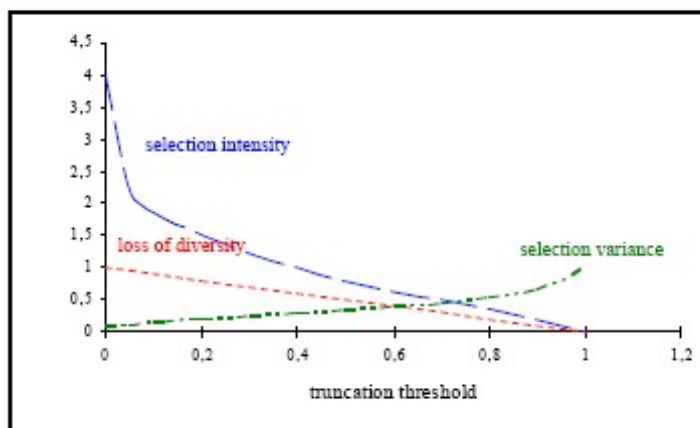
مقدار sp مربوط به رتبه‌گذاری غیر خطی در فاصله $[1, N-2]$ انتخاب می‌شود.

$$sp \in [1, N - 2]$$

حال با توجه به معادله بالا داریم :

$$Fitness(p) = N * x^{p-N}$$

شکل زیر رتبه‌گذاری خطی را از طریق نمودار با هم مقایسه می‌کند و با توجه به پارامترهای مختلف می‌توان این دو روش رتبه‌گذاری مختلف را بررسی کرد.



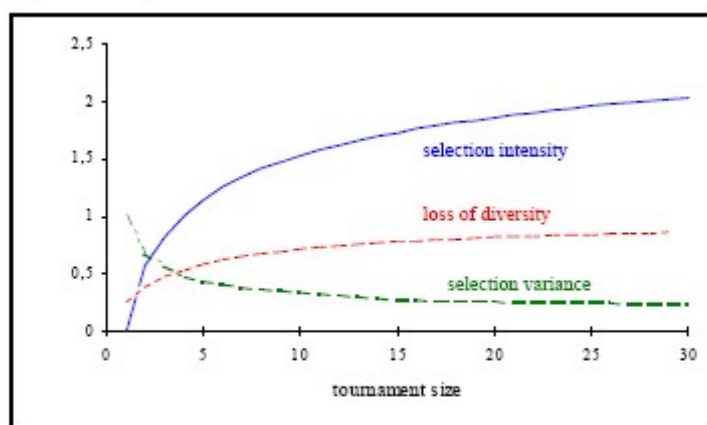
شکل ۴-۱ برآزندگی رتبه‌گذاری خطی و غیر خطی

جدول زیر (جدول ۴-۱) شامل مقادیر مختلف SP با توجه به اینکه جمعیت ما شامل ۱۱ فرد و مسئله ما مینیمم سازی است. جدول صفحه بعد مقادیر رتبه‌دهی خطی و غیر خطی برای جمعیت ۱۱ نفری را بررسی می‌کند و بر اساس برآزندگی هر فرد با توجه به فرمول‌های خطی و غیر خطی به هر فرد در جمعیت یک رتبه اختصاص پیدا می‌کند.

جدول ۴-۱- مقایسه رتبه‌گذاری خطی و غیر خطی

		مقدار برآزندگی هر یک از افراد			
		رتبه‌گذاری خطی	رتبه فرد در جامعه	رتبه‌گذاری غیرخطی	
مقدار تابع هدف	۲	۱/۱	۳/۰	۲/۰	
۱	۲/۰	۱/۱۰	۱۱	۳/۰۰	۲/۰۰
۳	۱/۸	۱/۰۸	۱۰	۲/۲۱	۱/۶۹
۴	۱/۶	۱/۰۶	۹	۱/۶۲	۱/۴۳
۷	۱/۴	۱/۰۴	۸	۱/۹۹	۱/۲۱
۸	۱/۲	۱/۰۲	۷	۰/۸۸	۱/۰۳
۹	۱/۰	۱/۰۰	۶	۰/۶۵	۰/۸۷
۱۰	۰/۸	۰/۹۸	۵	۰/۴۸	۰/۷۴
۱۵	۰/۶	۰/۹۶	۴	۰/۳۵	۰/۶۲
۲۰	۰/۴	۰/۹۴	۳	۰/۲۶	۰/۵۳
۳۰	۰/۲	۰/۹۲	۲	۰/۱۹	۰/۴۵
۹۵	۰/۰	۰/۹۰	۱	۰/۱۴	۰/۳۸

در جدول صفحه قبل در ستون اول مقدار تابع هدف هر فرد نوشته شده است که با توجه به اینکه فرد با برازندگی بیشتر دارای رتبه بهتر است لذا تابع برازندگی ۹۵ دارای رتبه ۱ و به همین ترتیب رتبه تابع برازندگی ۱ برابر ۱۱ می باشد که با توجه به فرمولهای رتبه خطی و غیر خطی و مقادیر مختلف SP مقادیر برازندگی برای افراد جامعه بدست می آید. این روش انتخاب دارای این قابلیت می باشد که توسط انتخابهای دیگر این روش کاملتر می شود که دارای قابلیت های فراوانی می باشد. شکل زیر مشخصات عملگر انتخاب رتبه گزاری خطی را نشان می دهد.



شکل ۲-۴ مشخصات عملگر انتخاب رتبه گزاری خطی

شدت انتخاب

$$SelInt_{LinRank}(sp) = (sp - 1) / \sqrt{\pi} \quad (4-4)$$

فقدان یا عدم گوناگونی

$$LossDiv_{linRank}(sp) = (sp - 1) / 4 \quad (5-4)$$

میزان پراکندگی انتخاب

$$Sel var_{linRank}(sp) = 1 - \frac{(sp - 1)^2}{\pi} = 1 - (selInt_{linRank}(sp))^2 \quad (6-4)$$

جدول زیر سه پارامتر ، شدت انتخاب و فقدان یا عدم گوناگونی و میزان پراکندگی انتخاب را با هم مقایسه کرده است. مقادیر مختلف این سه پارامتر برای ۱۱ فرد جمعیت قبلی بدست آمده است.

جدول ۲-۴ مقایسه پارامترهای انتخاب

Sp	Selection intensity	Loss of diversity	Selection variance
۲	۰/۵۶۴	۰/۲۵۰	۰/۶۸۲
۱/۹	۰/۵۰۸	۰/۲۲۵	۰/۷۴۲
۱/۸	۰/۴۵۱	۰/۲۰۰	۰/۷۹۷
۱/۷	۰/۳۹۵	۰/۱۷۵	۰/۸۴۴
۱/۶	۰/۳۳۹	۰/۱۵۰	۰/۸۸۵
۱/۵	۰/۲۸۲	۰/۱۲۵	۰/۹۲۰
۱/۴	۰/۲۲۶	۰/۱۰۰	۰/۹۴۱
۱/۳	۰/۱۶۹	۰/۰۷۵	۰/۹۷۱
۱/۲	۰/۱۱۳	۰/۰۵۰	۰/۹۸۷
۱/۱	۰/۰۵۶	۰/۰۲۵	۰/۹۹۷
۱	۰	۰	۱

۳-۲-انتخاب چرخ گردان

ساده‌ترین روش انتخاب ، روش انتخاب چرخ گردان است [6] و نیز این روش نمونه گیری تصادفی جایگزینی نامیده می‌شود. این یک الگوریتم تصادفی است و شامل تکنیک‌های زیر است.

۱- افراد به طور پیوسته روی یک پاره خط تصویر می شوند و فاصله‌ای که هر فرد اشغال می‌کند معادل اندازه برازندگی آن فرد است.

۲- عدد تصادفی انتخاب می‌کنیم و فردی که معادل عدد تصادفی است انتخاب می‌شود این روند را تا زمانی که تعداد کل جمعیت تکمیل نشود و تعداد افراد انتخاب شده به تعداد افراد جامعه اصلی نرسد، انجام می‌شود. این مشابه این تکنیک است که یک چرخ گردان را نسبت برازندگی افراد برش دهیم .

جدول زیر ، احتمال انتخاب و برازندگی هر فرد ونیز فشار انتخابی هر فرد را نشان می‌دهد.

فرد اول مناسبترین است لذا بیشترین فاصله را اشغال می‌کند و فرد دهم بعد از فرد یازدهم دارای کمترین برازندگی است و لذا کوچکترین فاصله را روی خط اشغال می‌کند. فرد یازدهم دارای مقدار برازندگی صفر می‌باشد در روند انتخاب ندارد.

جدول ۳-۴ انتخاب چرخ گردان

شماره افراد	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱
مقدار برازندگی	۲/۰	۱/۸	۱/۶	۱/۴	۱/۲	۱/۰	۰/۸	۰/۶	۰/۴	۰/۲	۰
احتمال انتخاب	۰/۱۸	۰/۱۶	۰/۱۵	۰/۱۳	۰/۱۱	۰/۰۹	۰/۰۷	۰/۰۶	۰/۰۳	۰/۰۲	۰

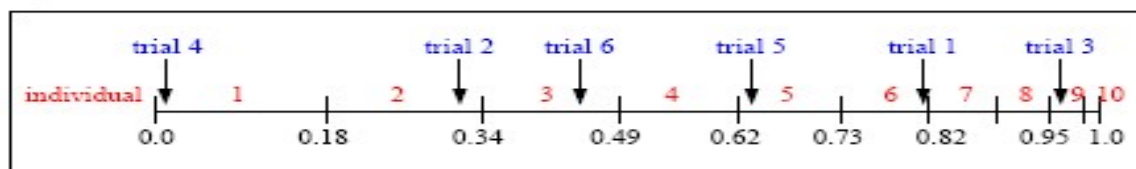
مقدار احتمال انتخاب افراد از تقسیم مقدار برازندگی هر فرد بر مجموع کل برازندگی افراد بدست می‌آید. حال برای هر فرد احتمال تجمعی هر فرد برابر است با احتمال همان فرد بعلاوه احتمال های افراد قبل از آن و سپس این احتمال‌های تجمعی را روی یک پاره خط تصویر می‌کنیم. روند انتخاب بدین صورت است که به همان تعداد که می‌خواهیم جواب مناسب از جمعیت، جواب انتخاب کنیم، عدد تصادفی در فاصله صفر و یک حال انتخاب می‌کنیم. حال هر کدام از اعداد در فاصله‌ای روی پاره خط قرار گیرد فرد نظیر به آن انتخاب می‌شود. مانند مثال زیر:

شش عدد تصادفی در فاصله $[0,1]$ انتخاب می‌کنیم.

۰,۴۲ و ۰,۶۵ و ۰,۰۱ و ۰,۹۶ و ۰,۳۲ و ۰,۸۱

شکل زیر عمل و روند انتخاب افراد را نشان می‌دهد. افراد توسط رابطه $\frac{f_i}{\sum f_i}$ روی پاره خط تصویر

می‌شوند.



شکل ۳-۴ انتخاب چرخ گردان

بعد از انتخاب جامعه شامل افراد زیر می‌باشد

شماره افراد: ۱ و ۲ و ۳ و ۵ و ۶ و ۹

در روش انتخاب چرخ گردان مقدار ارباب برابر صفر بدست می‌آید.

در اغلب روشهای اولیه برای انتخاب چرخ گردان روش بدین صورت بوده است که برازندگی افراد جامعه روی محیط دایره (دایره به شعاع یک) تصویر می‌شود. لذا مانند شکل، قبل هر فردی که دارای برازندگی بیشتری باشد طول قطاع بیشتری را روی دایره اشغال می‌کند و لذا احتمال انتخاب شدن آنها بیشتر است. این روند از روی روند مکانیکی چرخ متحرکی که دارای یک پیکان در مرکز می‌باشد الگو برداری شده است. چرخ را می‌چرخانیم و بعد از مدتی که چرخ متوقف شود پیکان روبروی یک قطاع قرار گرفته است لذا فرد معادل قطاع انتخاب می‌شود.

ما از این روند الگوبرداری کرده‌ایم و افراد را با توجه به فرمول $\frac{2 \times \pi \times f_i}{\sum f_i}$ که در آن f_i میزان برازندگی فرد i است، روی محیط دایره تصویر می‌شوند و با توجه به برازندگی هر فرد آن فرد، کمانی از دایره را اشغال می‌کند. حال توسط رایانه یک عدد به طور تصادفی در فاصله $[0, 2\pi]$ انتخاب می‌شود و سپس فرد معادل آن عدد انتخاب می‌شود و این روند را تا جایی که تمام افراد جامعه، انتخاب شوند ادامه پیدا می‌شود.

به همین دلیل است که عنوان این عملگر انتخاب را چرخ گردان گذاشته شده است.

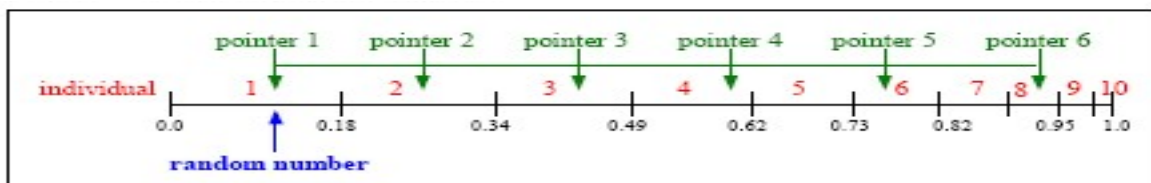
۳-۳- انتخاب بر اساس نمونه گیری کاملاً تصادفی

در این روش [5] انتخاب افراد روی قسمتی از خط تصویر می‌شوند. مانند قسمت چرخ گردان با این تفاوت که در این فاصله اشاره‌گرها به طور مساوی می‌باشند.

- روش انتخاب

حال فرض کنید تعداد افراد که می‌خواهیم انتخاب کنیم برابر NP باشد در این صورت فاصله بین اشاره‌گرها برابر $1/NP$ و موقعیت اولین اشاره‌گر به وسیله انتخاب یک عدد تصادفی که در فاصله $[0, 1/NP]$ تعیین می‌شود. به طور مثال برای انتخاب شش نفر، فاصله اشاره‌گر از هم باید $1/6=0.167$ باشد. و اولین فرد به طور کاملاً تصادفی در فاصله $[0, 0.167]$ انتخاب می‌شود. روند بدین صورت است که اولین اشاره‌گر در فاصله $[0, 0.167]$ به طور کاملاً تصادفی انتخاب می‌شود و اشاره‌گر اول در این فاصله قرار می‌گیرد. مکان اشاره‌گر دوم با اضافه شدن مقدار 0.167 به اشاره‌گر اول بدست می‌آید.

و این روند را ادامه می‌دهیم تا به تعداد مورد نیاز فرد انتخاب شود. شکل زیر روند انتخاب بر اساس نمونه گیری کاملاً تصادفی را نشان می‌دهد.



شکل ۳-۴ روند انتخاب بر اساس نمونه گیری تصادفی

بعد از عمل انتخاب، شماره افراد انتخاب شده عبارت است از:

۱ و ۲ و ۳ و ۴ و ۶ و ۸

در انتخاب بر اساس نمونه گیری کاملاً تصادفی، افرادی که انتخاب می‌شوند نسبت به افرادی که توسط تکنیک انتخاب چرخ گردان انتخاب می‌شوند، مناسب‌تر هستند.

در انتخاب بر اساس نمونه گیری کاملاً تصادفی، مقدار اریب برابر صفر بدست می‌آید.

۳-۴- انتخاب محلی

در انتخاب محلی [7] برای هر فرد، یک محیط محدود شده (همسایگی محلی) در نظر می‌گیریم و عمل انتخاب روی این همسایگی محلی صورت می‌گیرد. (در روش دیگر انتخاب، عمل انتخاب روی کل جمعیت انتخاب می‌شود). وقتی از این روش انتخاب استفاده می‌شود، که در جامعه ما غیر متمرکز و گسترده باشد. این همسایگی را می‌توان طوری انتخاب کرد که افراد داخل این همسایگی از پتانسیل بالایی برای ترکیب شدن و جفت‌گیری داشته باشد.

در روش انتخاب محلی ابتدا جمعیت جوابها محدود می‌شود، و سپس در ناحیه جدید توسط دیگر عملگرها انتخاب، مانند انتخاب چرخ گردان یا انتخاب کاملاً تصادفی و یا دیگر روش‌های انتخاب، افراد جامعه انتخاب می‌شود.

روش انتخاب محلی قابلیت ترکیب شدن با دیگر روش‌های انتخابی را دارا می‌باشد. ساختمان همسایگی‌ها می‌تواند به صورت زیر باشد.

۱- خطی

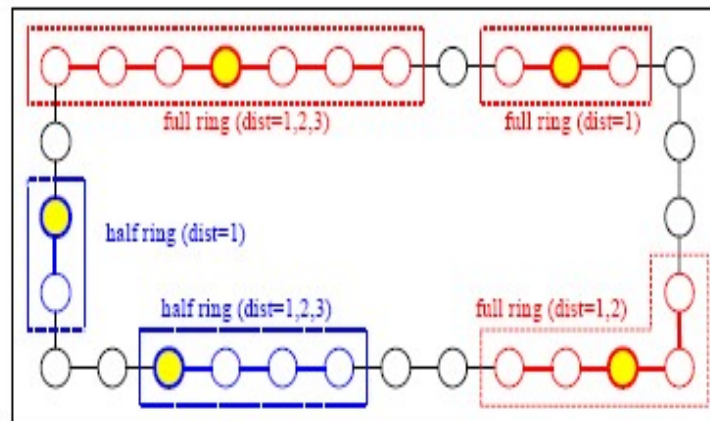
۱-۱ حلقه کامل، حلقه نیمه (شکل ۳-۵)

۲- دو بعدی

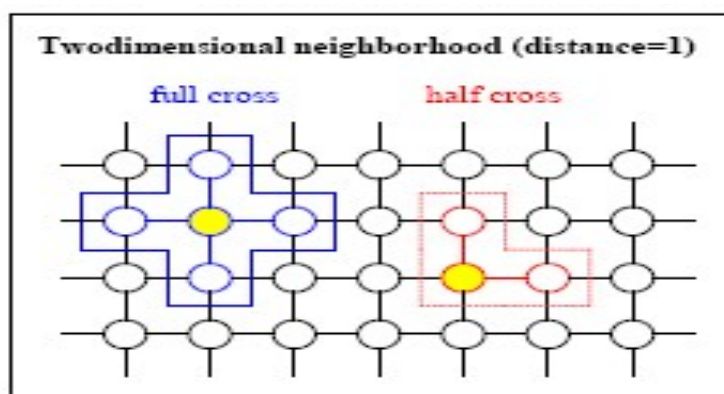
۱-۲ مقاطع کامل، مقاطع نیمه (شکل ۳-۶)

۲-۲ ستاره کامل، ستاره نیمه (شکل ۳-۷)

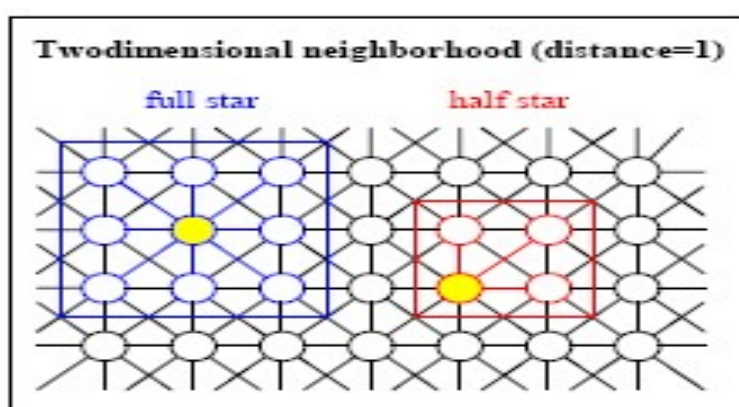
۳- سه بعدی و با پیچیدگی زیاد با هر ترکیبی از ساختارهای بالا



شکل ۳-۵ حلقه کامل و حلقه نیمه



شکل ۳-۶ همسایگی دو بعدی، متداخل نیمه و متداخل کامل



شکل ۳-۷ همسایگی دو بعدی، ستاره کامل و ستاره نیمه

روش انتخاب

اولین گام، انتخاب فردی از جامعه که قابلیت بالایی برای جفت گیری رد، است. که این انتخاب به طور تصادفی و با روش‌های انتخاب به طور تصادفی و با روش‌های انتخاب که قبلاً ذکر شده برای مثال انتخاب بر اساس نمونه گیری تصادفی یا انتخاب چرخ گردان و یا انتخاب کوتاه سازی که بعداً توضیح داده می‌شود.

حال یک همسایگی محلی برای هر فرد انتخاب شده، تعریف می‌کنیم. سپس داخل این همسایگی، افرادی که می‌توانند با هم جفت شوند، تعریف می‌کنیم. سپس داخل این همسایگی، افرادی که می‌توانند با هم جفت شوند (بر اساس معیار بهترین یا مناسبترین و به طور یکنواخت و تصادفی).

در جدول صفحه بعد مثال‌هایی از اندازه همسایگی و نیز فاصله آنها برای ساختارهای مختلف داده شده است.

ساختار	فاصله	
	۱	۲
Structure of selection	۱	۲
Full ring	۲	۴
Half ring	۱	۲
Full cross	۴	۸(۱۲)
Half cross	۲	۴(۵)
Full star	۸	۲۴
Half star	۳	۸

اندازه همسایگی، سرعت تکثیر اطلاعات بین افراد جامعه را تعیین می‌کند. بنابراین بین تکثیر سریع و تنوع زیاد در جامعه تصمیم‌گیری می‌شود و اغلب تمایل به تنوع زیاد می‌شود بنابراین از همگرایی سریع به یک مینیمم محلی جلوگیری می‌شود.

انتخاب محلی در یک همسایگی کوچک قابل‌اجرا تر از انتخاب محلی در یک همسایگی بزرگتر است. با این وجود همبندی کل جامعه باید برقرار بماند. همسایگی دو بعدی با ساختار نیمه ستاره با شعاع همسایگی برابر یک برای انتخاب محلی توصیه می‌شود.

به هر جهت اگر تعداد افراد جامعه ما زیاد باشد در این صورت یا شعاع همسایگی را زیاد در نظر می‌گیریم و یا از همسایگی دو بعدی دیگر باید استفاده کرد.

۳-۵- انتخاب هرسی (کوتاه سازی)

مقایسه مدل‌های انتخابی نشان می‌دهد که روش‌های انتخاب طبیعی الگو برداری شده است اما مدل انتخابی هرسی روش انتخابی مصنوعی است.

در انتخاب کوتاه‌سازی، افراد بر اساس برازندگی‌شان ذخیره می‌شوند. و تنها بهترین افراد به عنوان والدین انتخاب می‌شوند. والدین انتخاب شده به این روش، فرزندان به صورت تصادفی و یکنواخت تولید می‌کنند. این روش انتخاب مانند انتخاب محلی دارای قابلیت ترکیب شدن با دیگر روش‌های انتخاب (انتخاب چرخ گردان یا انتخاب کاملاً تصادفی) را دارا می‌باشد.

پارامتر انتخاب هرسی، *trunce* (پارامتر کوتاه‌سازی مرزی) می‌باشد. (اغلب مقدار پارامتر برابر $trunce = 1/sp$ که در آن $sp \in [1, Np - 2]$ می‌باشد که در آن NP برابر تعداد افراد جامعه است).

پارامتر *trunce* نسبت برازندگی جمعیت را به برازندگی والدین‌های انتخاب شده را تعیین می‌کند و مقدار بدست آمده بین ده درصد و پنجاه درصد می‌باشد. افراد زیر مرز پارامتر کوتاه‌سازی صاحب فرزند نمی‌شوند. جمله شدت انتخاب اغلب در انتخاب هرسی استفاده می‌شود.

جدول زیر رابطه بین پارامتر کوتاه‌سازی و شدت انتخاب را نشان می‌دهد.

جدول ۳-۴ رابطه بین شدت انتخاب و مرز کوتاه سازی.

Truncation threshold	۱٪	۱۰٪	۲۰٪	۴۰٪	۵۰٪	۸۰٪
Selection intensity	۲/۶۶	۱/۷۶	۱/۲	۰/۹۷	۰/۸	۰/۳۴

بررسی پارامترهای مختلف انتخاب هرسی

تراکم انتخاب

$$selInt_{Truncation}(Trunc) = \frac{1}{Trunc} \cdot \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{fc^2}{2}}$$

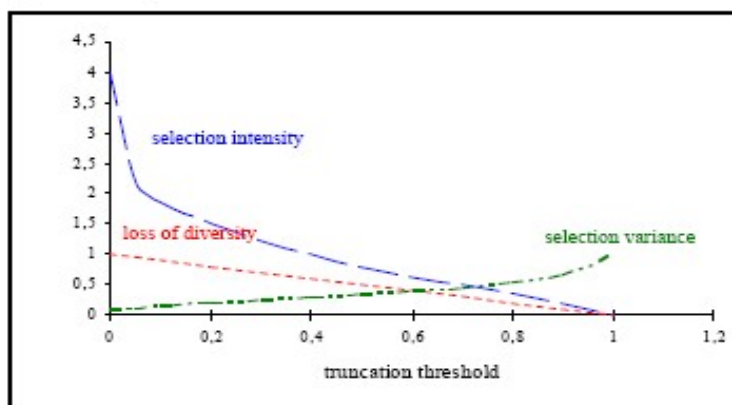
فقدان یا عدم گوناگونی

$$lossDiv_{Truncation}(Trunc) = 1 - Trunc$$

میزان پراکندگی انتخاب

$$selvar_{Truncation}(Trunc) = 1 - selInt_{Truncation}(Trunc) \cdot (selInt_{Truncation}(Trunc) - f_c)$$

شکل زیر ویژگیهای انتخاب هرسی را نشان می دهد



شکل ۳-۷ ویژگیهای انتخاب هرسی

روش انتخاب هرسی

در این قسمت با ارائه یک مثال قصد داریم [3] روش انتخاب هرسی را بررسی کنیم. روند بدین صورت است که ما یک مرز در نظر می گیریم. افرادی که بالای مرز باشند را نگه می داریم و افرادی که زیر مرز می باشند را حذف می کنیم و بعد با استفاده از روشهای انتخاب تعداد افراد مورد نیاز را انتخاب می کنیم.

مثال: فرض کنید مقادیر برازندگی افراد جامعه به صورت زیر باشد

$$Fitnv=[2,0.7,0.7,3.1,4.12.01.60.90.1]$$

حال مقدار پارامتر Tranc را محاسبه می‌کنیم و با توجه به اینکه $sp=2$ انتخاب شده است

$$Trunce = 1/sp = 1/2 = 0.5$$

حال برای تک تک افراد جامعه مقدار $\frac{FitnV}{\max(FitnV)}$ را حساب می‌کنیم.

$$\frac{FitnV}{\max(Fitnv)} = [0/05, 0/45, 0,8/1, 0/2, 0/65, 0/85, 0/35, 0/1]$$

حال افرادی که مقدار $\frac{FitnV}{\max(FitnV)}$ آن کمتر از 0.5 است حذف می‌کنیم و از بین بقیه افراد تعدادی

را انتخاب می‌کنیم. به طور مثال شش نفر را انتخاب کرده که عبارتند از:

$$Nem\ chrom = [3,4,7,4,6,7]$$

انتخاب هرسی، انتخابی است که قابلیت ترکیب شدن با دیگر روشهای انتخابی را دارا می‌باشد.

۳-۶- انتخاب به روش مسابقه‌ای (قرعه کشی)

این روش نیز مانند روش انتخاب هرسی و بر خلاف روشهای قبلی که از روشهای انتخاب طبیعی الگو برداری شده است، یک روش انتخاب مصنوعی است.

در عمل انتخاب به روش مسابقه‌ای [3] هر فرد جامعه دارای یک عدد (شماره) می‌باشند. این روش انتخاب مانند روشهای قرعه کشی می‌باشد و بر اساس قرعه کشی افراد انتخاب می‌شود. البته می‌توان این روش انتخاب را طوری طراحی کرد که احتمال انتخاب افراد با برازندگی بالا بیشتر از افراد دیگر باشد. در این روش انتخاب یک عدد (ToUr) را به طور تصادفی انتخاب می‌کنیم و سپس آن فرد به عنوان والدین انتخاب می‌شود. این روند انتخاب والدین به طور مکرر تکرار می‌شود. این روند انتخاب والدین، به طور تصادفی باعث تولید فرزند می‌شود. پارامتر انتخاب به روش مسابقه‌ای، (tour)، اندازه مسابقه می‌باشد. مقدار (tour) از ۱ تا NP (تعداد افراد جامعه) قابل تغییر است.

بررسی پارامترهای مختلف انتخاب به روش مسابقه‌ای

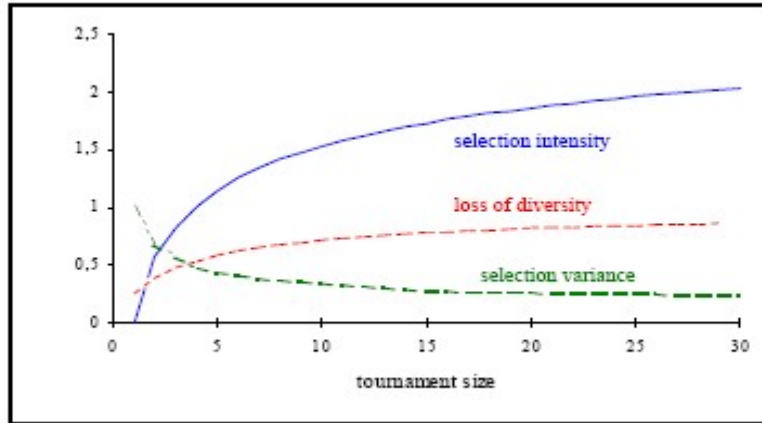
شدت انتخاب

$$sel\ln t_{Turnier}(Tour) \sqrt{2 \cdot (In(Tour) - In(\sqrt{4.14 \cdot In(Tour)}))}$$

درصد عدم گوناگونی

$$lossiv_{urniter}(tour) \frac{0.918}{In(1.186 + 1.328 \cdot Tour)}$$

شکل ۳-۸ رابطه بین اندازه مسابقه و پارامتر تمرکز انتخابی را نشان می‌دهد.



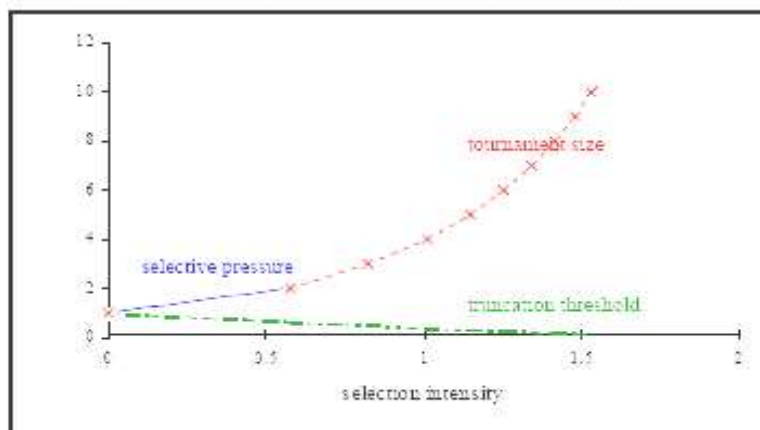
شکل ۸-۳ رابطه بین اندازه مسابقه و پارامتر تمرکز انتخابی

مانند روشهای دیگر انتخابی، انتخاب مسابقه‌ای دارای این ویژگی است که می‌توان با دیگر روشهای انتخابی ترکیب شود و نتایج بهتری برای مسئله ایجاد کند. مثلاً می‌توان افرادی که در مرحله انتخاب مسابقه‌ای انتخاب می‌شوند را دوباره توسط روشهای دیگر انتخاب افراد بهتری برای مسئله انتخاب کرد.

۴-۷- مقایسه الگوهای مختلف انتخاب

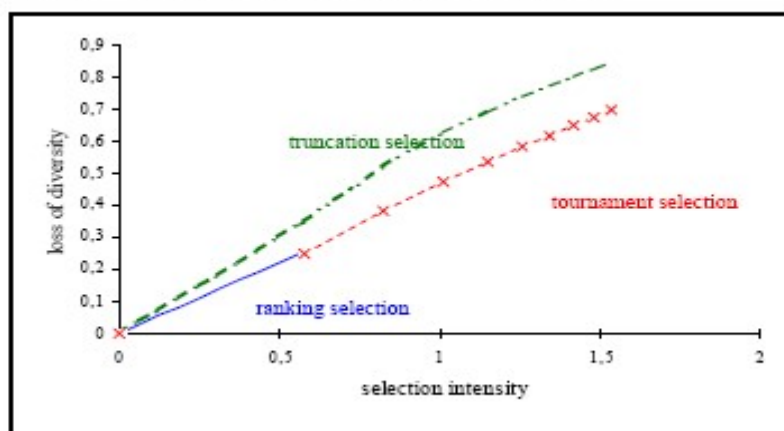
با توجه به مطالب که در قسمتهای قبل نشان داده شد. چنین نتیجه می‌شود که رفتار روشهای مختلف انتخابی شبیه شدت انتخاب می‌باشد.

شکل زیر (شکل ۹-۳) رابطه بین شدت انتخاب و ویژگی‌های پارامترهای روشهای مختلف انتخابی (فشار انتخابی، مرز کوتاه سازی اندازه مسابقه) را نشان می‌دهد. شکل زیر شرح می‌دهد که در انتخاب به روش مسابقه‌ای تنها از مقادیری جدا، اجازه داده می‌شود و در انتخاب به روش رتبه‌گذاری خطی تنها مجاز به استفاده کوچکترین مقدار برای شدت انتخاب می‌باشیم.



شکل ۹-۳ وابستگی بین پارامتر انتخاب و شدت انتخاب

به هر جهت رفتار روشهای مختلف باهم متفاوت است. لذا روشهای انتخاب با تعداد افرادی که در طول انتخاب حذف می‌شوند، با شدت انتخاب مقایسه می‌شوند (شکل ۱۰-۳).

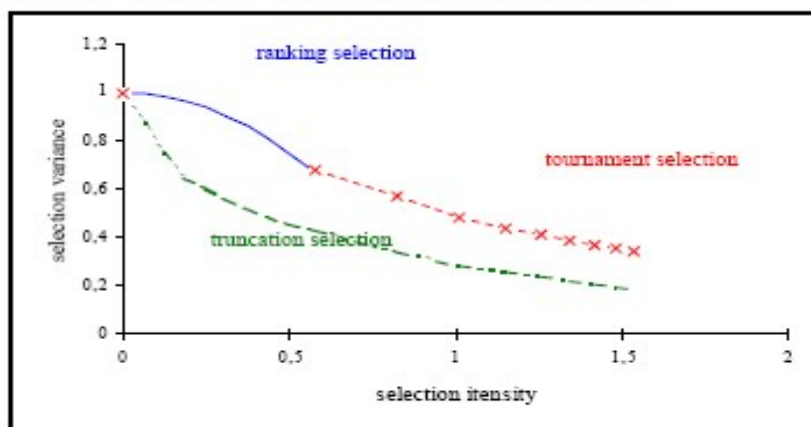


شکل ۱۰-۳ وابستگی تعداد افراد انتخاب نشده به شدت انتخاب

انتخاب به روش کوتاه‌سازی نسبت به روشهای دیگر انتخاب، دارای تعداد زیادی افراد است که انتخاب نشده‌اند.

در مقایسه با دیگر روشهای انتخابی (روش انتخاب رتبه‌گذاری، روش انتخاب مسابقه‌ای). در روش انتخاب هرسی درجه برازندگی افرادی که کمتر از بقیه هستند حذف می‌شوند و به جای آنها فرزندان که دارای درجه برازندگی بیشتر است، جایگزین می‌شوند زیرا افرادی که درجه برازندگی آنها زیر حد انتخاب باشند انتخاب نمی‌شوند.

روش انتخاب رتبه‌گذاری و روش انتخاب مسابقه‌ای دارای رفتار شبیه به روش انتخاب هرسی هستند. به هر جهت انتخاب به روش رتبه‌گذاری در یک سطحی عمل می‌کند که انتخاب به روش کوتاه‌سازی نمی‌تواند عمل کند زیرا دارای مشخصه مجزا از انتخاب به روش مسابقه‌ای است. در شکل (۱۱-۳) میزان پراکندگی انتخاب و شدت با هم مقایسه می‌شوند.



شکل ۱۱-۳ وابستگی بین پراکندگی انتخاب و شدت انتخاب

با توجه به شکل بالا با افزایش شدت انتخاب مقدار پراکندگی انتخاب در سه روش انتخاب، انتخاب رتبه‌گذاری، هرسی و مسابقه‌ای کاهش می‌یابد مانند قبل روش انتخاب رتبه‌گذاری روی قسمت و منطقه ای کار می‌کند که روش انتخاب مسابقه ای کار نمی‌کند و محدوده عمل این دو روش انتخاب از هم جداست زیرا دارای مشخصه مجزا از یکدیگر می‌باشند.

فصل پنجم

باز ترکیبی

۵-۱- باز ترکیبی مجزا

۵-۲- باز ترکیبی مقادیر حقیقی

۵-۲-۱- باز ترکیبی میانی

۵-۲-۲- باز ترکیبی خطی

۵-۲-۳- باز ترکیبی خطی توسعه یافته

۵-۳- باز ترکیبی مقادیر باینری

۵-۳-۱- عملگر Crossover تک نقطه‌ای / دو نقطه‌ای / چند نقطه‌ای

۵-۳-۲- عملگر Crossover یک شکل

۵-۳-۳- عملگر Crossover برزننده

۵- باز ترکیبی

باز ترکیبی از ترکیب اطلاعاتی و یا جابجایی ژنها که والدین دارا باشند افراد جدیدی (فرزند) بوجود می‌آورند. این عمل توسط ترکیب مقادیر متغیرهایی از والدین انجام می‌شود. بخش ۱-۵ عملگر باز ترکیبی مجزا را توصیف می‌کند این عملگر برای هر نوع متغیر، قابل استفاده است بخش ۲-۵- روش‌های مختلف باز ترکیبی برای مقادیر حقیقی را توضیح می‌دهد و روش‌های مختلف باز ترکیبی برای متغیر باینری را نیز در بخش ۳-۵ توضیح می‌دهیم. روش‌های باز ترکیبی برای متغیر با مقادیر باینری مرحله خاصی از باز ترکیبی مجزا می‌باشند. این روش‌ها می‌تواند برای همه متغیرها با مقادیر حقیقی به خوبی قابل استفاده باشد.

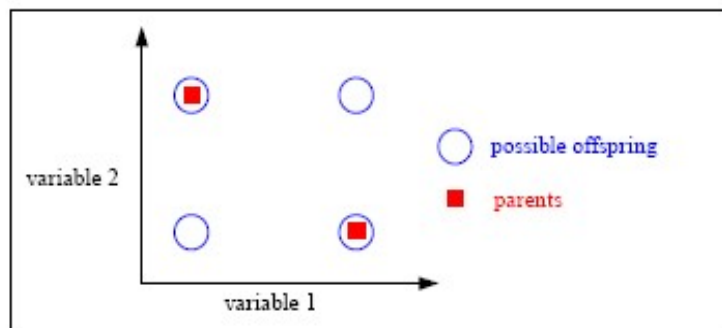
۵-۱- باز ترکیبی مجزا

عملگر باز ترکیبی مجزا روی هر متغیر (حقیقی، باینری و یا نشانه‌ای) قابل استفاده است. باز ترکیبی مجزا، مقادیر یا ژنهای بین افراد جامعه را تغییر می‌دهد. جابجایی مقادیر متغیرهای بین والدین و ایجاد فرزندان دارای احتمال یکسان هستند. مقادیر متغیرها بین والدین طبق فرمول زیر تغییر می‌کند و فرزند به وجود می‌آید.

$$Var_i^o = Var_i^{p1} * a_i + Var_i^{p2} * (1 - a_i)$$

$$a_i \in \{0,1\}, i \in \{1,2,\dots, NVar\}$$

در رابطه بالا a_i به طور یکنواخت و تصادفی برای هر I انتخاب می‌شود. باز ترکیبی مجزا، نقاط گوشه‌ای مربع تعریف شده، توسط والدین را تولید می‌کند. شکل زیر با توجه به رابطه (۵-۱) تاثیر هندسی عملگر باز ترکیبی مجزا را بر روی والدین و همچنین موقعیتهای ممکن فرزندان بعد از انجام عمل باز ترکیبی مجزا روی والدین را نشان می‌دهد.



شکل ۵-۱ موقعیت‌های ممکن فرزندان بعد از باز ترکیبی مجزا

حال با ارائه مثال زیر روند اجرائی عملگر باز ترکیبی مجزا را بررسی می‌کنیم.
 دو فرد زیر را با سه متغیر (ژن) در نظر بگیرید:

فرد یا والدین اول: ۱۲ ۲۵ ۵
 فرد یا والدین دوم: ۱۲۳ ۴ ۳۴

حال برای هر فرد a_i نظیر به خود را به طور تصادفی در مجموعه $[0, 1]$ انتخاب می‌کنیم

$$a_1 = \begin{matrix} 0 & 0 & 1 \\ 1 & 0 & 1 \end{matrix}$$

مقادیر متغیرهای والدین مطابق رابطه (۵-۱) و با توجه به a_i های بالا ترکیب می‌شوند و به طور تصادفی و با احتمال برابر، فرزندان به وجود می‌آید که به صورت زیر می‌باشد.

فرزند اول: ۱۲۳ ۴ ۵
 فرزند دوم: ۱۲ ۴ ۵

عملگر باز ترکیبی مجزا برای هر نوع متغیر (باینری، حقیقی یا نشانه‌ای) قابل استفاده است.

۵-۲-۲-۵- باز ترکیبی مقادیر حقیقی

روشهای باز ترکیبی در این بخش برای باز ترکیبی افراد با مقادیر (ژنها) حقیقی کاربرد دارد.

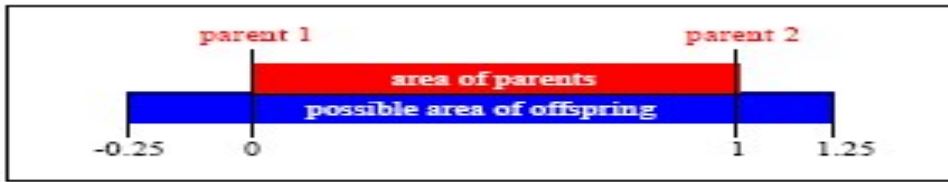
۵-۲-۱-۲-۵- باز ترکیبی میانی

عمل باز ترکیبی میانی [6] روشی است که فقط برای متغیرهای حقیقی قابل استفاده است. از این رو مقادیر متغیرهای (ژنها) فرزندان قدری نزدیک و بین مقادیر متغیرهای (ژنها) والدین بدست می‌آید. عملگر باز ترکیبی میانی طبق رابطه زیر عمل می‌کند.

$$Var_i^o = Var_i^{p_1} * a_i + Var_i^{p_2} * (1 - a_i)$$

$$i \in \{1, 2, \dots, NVar\}$$

که در آن a_i به طور تصادفی در بازه $[-d, 1+d]$ انتخاب می‌شود. هنگامی که $d=0$ انتخاب می‌شود سطح تعریف شده برای فرزندان مشابه اندازه سطح گسترش یافته به وسیله والدین شان می‌باشد. این روش، روش باز ترکیبی میانی استاندارد نامیده می‌شود. در عمل باز ترکیبی میانی توسعه یافته $d > 0$ یک انتخاب خوب و نرمال است. شکل (۲-۵) نشان دهنده محدود تغییرات متغیر فرزند که توسط متغیرهای والدین تعریف شده است.



شکل ۵-۲ عملگر باز ترکیبی میانی

حال با ارائه مثال زیر بیشتر با روند اجرائی عملگر باز ترکیبی میانی که فقط بر روی ژنهای با مقادیر حقیقی اثر می کند آشنا می شویم.

فرد یا والدین اول: ۱۲ ۵ ۵
 فرد یا والدین دوم: ۱۲۳ ۴ ۳۴

حال مقادیر a_i را به طور تصادفی در بازه $[0/۲۵, ۱/۲۵]$ انتخاب می کنیم

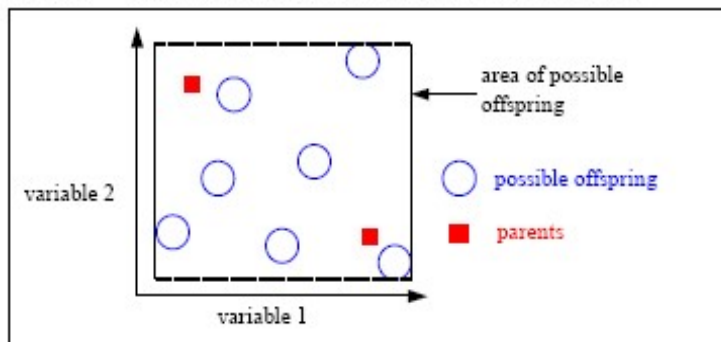
$a_1 = 0/۵ \quad 1/1 \quad -0/1$
 $a_2 = 0/1 \quad 0/۸ \quad 0/۵$

فرزندانی که توسط رابطه (۵-۲) و مقادیر مختلف a_i بدست می آیند

فرزند اول: ۵/۶۷ ۱/۹ ۲/۱
 فرزند دوم: ۲۳/۱ ۸/۲ ۱۹/۵

عمل باز ترکیبی میانی قادر است که هر نقطه مربع فرزند را کمی بزرگتر از قسمتی که توسط والدین تعریف شده است تولید کند.

شکل (۵-۳) سطح که توسط فرزندان بعد از عمل باز ترکیبی میانی را اشغال می کند را نشان می دهد.



شکل ۵-۳ عملگر باز ترکیبی میانی

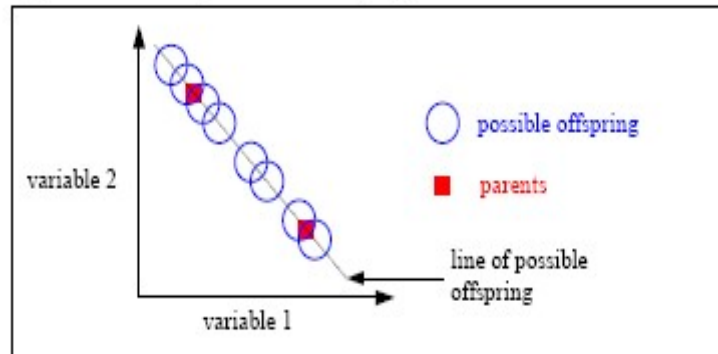
۵-۲-۲-۲-۵ باز ترکیبی خطی

باز ترکیبی خطی شبیه باز ترکیبی میانی [3] است با این تفاوت که تنها یک مقدار برای a همه متغیرها استفاده می شوند.

$$Var_i^o = Var_i^{p1} * a_i + Var_i^{p2} * (1 - a_i)$$

$$i \in \{1, 2, \dots, NVar\}$$

که در آن a به طور تصادفی در بازه $[-d, 1+d]$ انتخاب می‌شود. هنگامی که $d=0$ انتخاب می‌شود سطح تعریف شده برای فرزندان مشابه اندازه سطح گسترش یافته به وسیله والدین‌شان می‌باشد. این روش، روش باز ترکیبی میانی استاندارد نامیده می‌شود. در عمل باز ترکیبی میانی توسعه یافته $d > 0$ می‌باشد. یک انتخاب خوب و نرمال است توصیه می‌شود از این مقدار در محاسبات استفاده شود. شکل (۵-۲) نشان دهنده محدوده تغییرات متغیر فرزند که توسط متغیرهای والدین تعریف شده است.



شکل ۵-۲ عملگر باز ترکیبی خطی

حال با ارائه یک مثال روند اجرائی عملگر باز ترکیبی خطی را بررسی می‌کنیم. در آخر می‌توانیم با اجرای برنامه این عملگر بیشتر با روش آن آشنا شوید.

دو والدین زیر که هر کدام شامل سه متغیر (ژن) است در نظر بگیرید:

فرد یا والدین اول:	۱۲	۲۵	۵
فرد یا والدین دوم:	۱۲۳	۴	۳۴

حال مقادیر a_i را به طور تصادفی در بازه $[1/25, -0/25]$ انتخاب می‌کنیم.

$$a_1 = 0/5$$

$$a_2 = 0/1$$

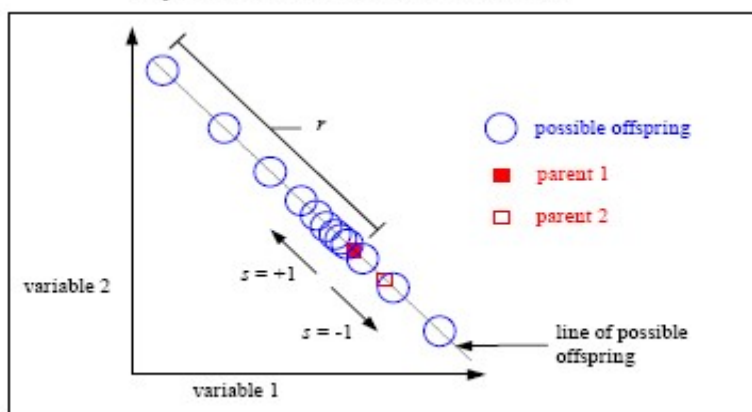
فرزندانی که توسط رابطه (۵-۳) و مقادیر مختلف a_i بدست می‌آیند عبارتند از:

$$\text{فرزند اول: } 67/5 \quad 14/5 \quad 19/5$$

$$\text{فرزند دوم: } 23/1 \quad 22/9 \quad 7/9$$

روش باز ترکیبی خطی قادر است هر نقطه‌ای که روی خطی که توسط والدین تعریف شده است را تولید کند.

شکل (۵-۴) موقعیت‌های ممکن فرزندان را بعد از انجام عمل باز ترکیبی خطی را نشان می‌دهد.



شکل ۴-۵ بازترکیبی خطی

۵-۲-۳- بازترکیبی خطی توسعه یافته

عمل بازترکیبی خطی توسعه یافته [1] فرزندان را در سمت و سوی مقادیر متغیرهای (ژن‌های) والدین به وجود می‌آورد. به هر جهت عمل بازترکیبی خطی توسعه یافته محدود به خط بین والدین و فاصله کوچک بیرون آنها نمی‌باشد و از روی ژن‌های والدین، فرزندان به وجود می‌آیند. میزان سطح ممکن برای ایجاد کردن فرزند به وسیله دامنه‌ی متغیرها تعریف شده است. در منطقه تولید فرزند، فرزندان به طور یکنواخت و تصادفی ایجاد می‌شود. اگر میزان برزندگی والدین در سطح بالایی باشد، تولید فرزند می‌کند. فرزندان طبق قانون و رابطه زیر تولید و تکثیر هستند.

$$VAR_i^O = Var_i^R + s_i \cdot r_i \cdot a_i \cdot \frac{Var_i^{p_2} - Var_i^R}{Var_i^{p_1} - Var_i^{p_2}} \quad i \in (1, 2, \dots, N \text{ var})$$

$a_i = 2^{2^i}$, mutation precision. $u \in [0, 1]$ uniform at random, a_i for all i identical,

$r_i = r$ damain r : ranje of recombination steps,

$s_i \in \{-1, +1\}$, uniform at random: undirected recombination,

+1 With probability > 0.5 : directed recombination

$$r \in [0.110^6 - 6]$$

$$k \in \{4, 5, \dots, 20\}$$

متغیر s_i به طور یکنواخت و تصادفی انتخاب می‌شود.

پارامتر a_i متناسب با اندازه گام است و پارامتر s رابطه مستقیم با اندازه گام بازترکیبی دارد.

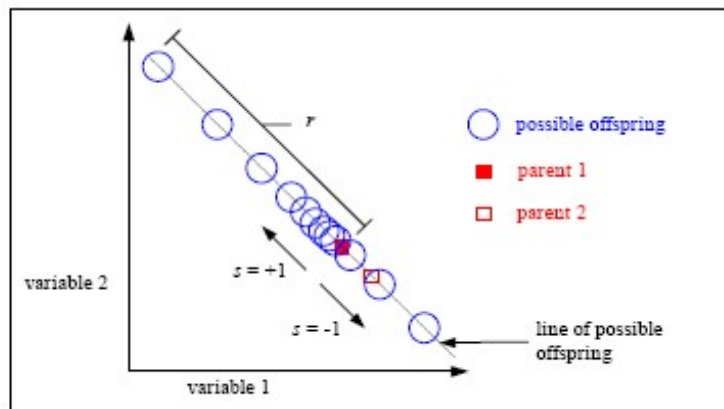
شکل ۵-۵ سعی می‌کند تاثیر روش بازترکیبی توسعه یافته را نشان دهد.

پارامتر k تعیین کننده دقت اندازه گام‌های روش بازترکیبی است. هر چه k بزرگتر باشد اندازه گام

کوچکتر می‌شود. بسته به نوع مسئله کاربر می‌تواند مقدار k را خود از اعداد ۴ تا ۲۰ انتخاب کند.

برای هر مقدار k ، بزرگترین مقدار a برابر ۱ می‌باشد ($u=1$) و کوچکترین مقدار a که وابسته به k دارد برابر $a = 2^k - k$ است ($u=1$) به طور نمونه مقادیر که برای پارامتر k در نظر گرفته می‌شود در سطحی از ۴ تا ۲۰ می‌باشد.

شکل (۵-۵) موقعیت‌های ممکن فرزندان بعد از اعمال عملگر بازترکیبی خطی توسعه یافته روی والدین و تعریف سطح متغیرها را نشان می‌دهد که با توجه به مقدار s که مثبت یک یا منفی یک انتخاب شود فرزندان در دو سوی مختلف تکثیر می‌شوند.



شکل ۵-۵ عملگر بازترکیبی خطی توسعه یافته

بهترین مقدار که می‌توانیم برای r انتخاب کرد دخ درصد از دامنه متغیرها است. به هر جهت مطابق تعریف دامنه متغیرها یا برای مراحل خاص این پارامتر انتخاب شود سطحی از والدین که تولید فرزند می‌کند، کوچک می‌شود. همچنین در این روش پارامتر s از مجموعه $[-1, 1]$ به طور تصادفی انتخاب می‌شود.

۵-۳- بازترکیبی مقادیر باینری

در این قسمت روش‌های بازترکیبی برای افراد با مقادیر باینری توصیف می‌شود. معمولاً این روشها، عملگر crossover نامیده می‌شود. بنابراین عبارت crossover به جای نام روش بازترکیبی استفاده می‌شود.

۵-۳-۱- الف: عملگر crossover تک نقطه‌ای

روش این عملگر بدین صورت است که به طور تصادفی عدد صحیح مثبت k (موقعیت عملگر crossover) را از مجموعه $[1, 2, \dots, Nvar-1]$ انتخاب می‌کنیم که در آن $Nvar$ برابر طول افراد می‌باشد و ژنها یا متغیرهای بعد از عدد تصادفی دو والدین با هم جابجا می‌شوند که این تغییر ژنها باعث به وجود آمدن فرزندان جدید می‌شود. حال با ارائه مثال زیر بیشتر با روند کار این عملگر آشنا می‌شویم.

مثال: دو فرد به طول ۱۱ بیت در نظر بگیرید:

فرد یا والدین اول	۰	۱	۱	۱	۰	۰	۱	۱	۰	۱	۰
فرد یا والدین دوم	۱	۰	۱	۰	۱	۱	۰	۰	۱	۰	۱

به طور تصادفی عدد k را در مجموعه $[۱۰ و ... و ۲ و ۱]$ (چون طول کروموزوم یا والدین برابر ۱۱ می باشد) انتخاب می کنیم.

$K=5$

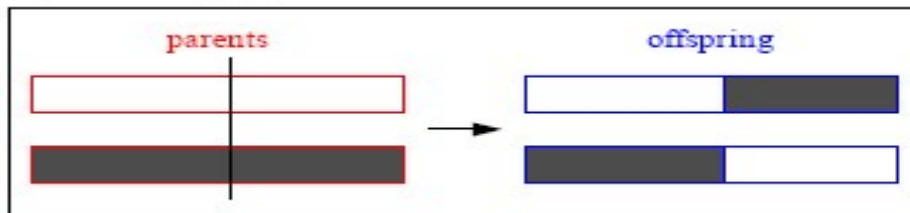
حال محل اثر عملگر را روی دو والدین یا کروموزوم مشخص می کنیم.

فرد یا والدین اول	۰	۱	۱	۱	۰	۰	۱	۱	۰	۱	۰
فرد یا والدین دوم	۱	۰	۱	۰	۱	۱	۰	۰	۱	۰	۱

بعد از اعمال عملگر crossover تک نقطه ای فرزندان جدید به وجود می آیند که عبارتند از:

فرزند اول	۰	۱	۱	۱	۰	۱	۰	۰	۱	۰	۱
فرزند دوم	۱	۰	۱	۰	۱	۰	۱	۱	۰	۱	۰

شکل زیر روش کار عملگر crossover تک نقطه ای را نشان می دهد.



شکل ۱-۳-۵ عملگر crossover تک نقطه ای

۵-۳-۱-ب: عملگر crossover دو نقطه ای

روش عملگر بدین صورت است که به صورت تصادفی دو عدد صحیح مثبت k_1, k_2 را از مجموعه $[۱-nvar و ... و ۲ و ۱]$ انتخاب می کنیم که در آن $Nvar$ طول افراد می باشد و روش این عملگر مانند عملگر قبل است با این تفاوت که ژنهای بین دو عدد جابجا می شود که باعث به وجود آمدن فرزندان جدید می شود. روش این عملگر را با ارائه مثال زیر بیشتر توضیح می دهیم.

مثال: دو نفر به طول ۱۱ بیت در نظر بگیرید

والدین اول	۰	۱	۱	۱	۰	۰	۱	۱	۰	۱	۰
والدین دوم	۱	۰	۱	۰	۱	۱	۰	۰	۱	۰	۱

به طور تصادفی عددهای k_1, k_2 را در مجموعه‌های $[۱۰...۱۲ و ۱]$ انتخاب می‌کنیم.

$$k_1 = 3$$

$$k_2 = 8$$

حال محل اثر عملگر را روی دو والدین یا کروموزوم مشخص می‌کنیم.

والدین اول	۰	۱	۱	۱	۰	۰	۱	۱	۰	۱	۰
والدین دوم	۱	۰	۱	۰	۱	۱	۰	۰	۱	۰	۱

بعد از اعمال عملگر crossover فرزندان جدید زیر به وجود می‌آید.

فرزند اول	۰	۱	۱	۰	۱	۱	۰	۰	۰	۱	۰
فرزند دوم	۱	۰	۱	۱	۰	۰	۱	۱	۱	۰	۱

۵-۳-۱-ج: عملگر crossover چند نقطه‌ای

برای عملگر crossover چند نقطه‌ای [6] تعداد m عدد صحیح مثبت k_i در مجموعه [۱- $Nvar$ و... و ۱] که در آن $m, \dots, i=1$ به طور تصادفی انتخاب می‌کنیم سپس مقادیر بین نقاط crossover به طور پیاپی بین والدین جابجا می‌شوند و فرزندان جدید به وجود می‌آورند. حال با ارائه مثال زیر روند این عملگر را بررسی می‌کنیم.
مثال: دو فرد یا کروموزوم زیر را به طول ۱۱ بیت در نظر بگیرید.

والدین اول	۰	۱	۱	۱	۰	۰	۱	۱	۰	۱	۰
والدین دوم	۱	۰	۱	۰	۱	۱	۰	۰	۱	۰	۱

به طور تصادفی عملگر crossover را انتخاب می‌کنیم ($m=3$)

$$k_1 = 2$$

$$k_2 = 6$$

$$k_3 = 10$$

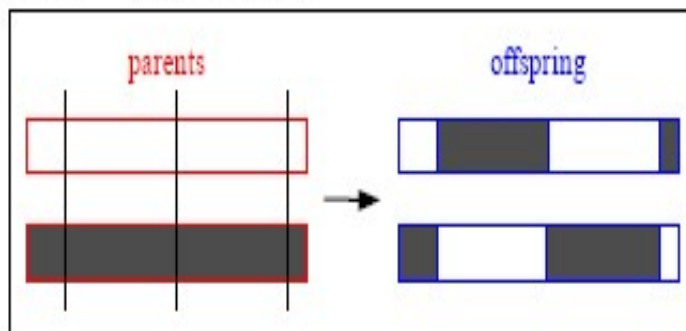
حال محل اثر عملگر را روی دو والدین یا کروموزوم مشخص می‌کنیم.

والدین اول	۰	۱	۱	۱	۰	۰	۱	۱	۰	۱	۰
والدین دوم	۱	۰	۱	۰	۱	۱	۰	۰	۱	۰	۱

بعد از انجام عملگر crossover و تولید فرزند جدید خواهیم داشت.

فرزند اول	۰	۱	۱	۰	۱	۱	۰	۱	۱	۱	۰
فرزند دوم	۱	۰	۱	۱	۰	۰	۰	۰	۱	۰	۱

شکل زیر روش کار عملگر crossover چند نقطه‌ای را نشان می‌دهد.



شکل ۲-۳-۵ عملگر crossover چند نقطه‌ای

۵-۳-۲-عملگر crossover یکنواخت

عملگر crossover تک نقطه‌ای، دو نقطه‌ای و یا چند نقطه‌ای [3] بین نقاطی از مکان هندسی افراد قابل تعریف است، که می‌توان آنها را برش داد. عملگر crossover یکنواخت این عمل را به هر نقطه از مکان هندسی از والدین برای تولید فرزند بهتر تعمیم می‌دهد. روش عملگر crossover یکنواخت به دو صورت است که تفاوت آنها در تعداد الگوئی است که استفاده می‌شود و والدین بر طبق الگوهای ارائه شده تغییر ژن می‌دهند و همین تغییر ژن باعث به وجود آمدن فرزندان جدید می‌شوند. این دو روش را با ارائه مثال‌های مختلف بررسی می‌کنیم. دو فرد زیر با متغیرهای باینری در نظر بگیرید:

والدین اول	۰	۱	۱	۱	۰	۰	۱	۱	۰	۱	۰
والدین دوم	۱	۰	۱	۰	۱	۱	۰	۰	۱	۰	۱

در این روش مقادیر فرزندان به طور تصادفی از روی والدین انتخاب می‌شود از این رو ژن فرزند اول از روی ژن فرد اول یا والدین اول ایجاد می‌شود، هر گاه مقدار بیت نظیر فرد یا والدین اول در الگوی نظیر به آن برابر یک باشد و همچنین بیت نظیر به فرد دوم نظیر صفر باشد و همچنین برای ژن فرزند دوم بر عکس فرزند اول به وجود می‌آید بدین طریق که هر گاه بیت نظیر به فرد دوم برابر یک باشد لذا فرزند دوم از روی فرد یا والدین دوم ایجاد می‌شود.

حال با توجه به توضیحات بالا ما ابتدا دو الگو که رشته‌هایی از مقادیر صفر و یک و با طول مساوی با طول والدین می‌باشد را به طور تصادفی انتخاب می‌کنیم. مانند صفحه بعد:

نمونه اول	۰	۱	۱	۰	۰	۰	۱	۱	۰	۱	۰
نمونه دوم	۱	۰	۰	۱	۱	۱	۰	۰	۱	۰	۱

بعد از اعمال عملگر crossover یکنواخت فرزندان جدیدی به وجود می‌آیند که عبارتند از:

فرزند اول	۱	۱	۱	۰	۱	۱	۱	۱	۱	۱	۱
فرزند دوم	۰	۰	۱	۱	۰	۰	۰	۰	۰	۰	۰

روش دوم به این صورت است که فقط یک الگو برای هر دو والدین انتخاب می‌شود و هر دو والدین طبق این الگو با هم ترکیب می‌شوند (ازدواج می‌کنند) با ارائه مثال زیر بیشتر با این عملگر آشنا می‌شویم.

والدین اول	۰	۱	۱	۱	۰	۰	۱	۱	۰	۱	۰
والدین دوم	۱	۰	۱	۰	۱	۱	۰	۰	۱	۰	۱

حال یک الگو (رشته‌ای با مقادیر صفر و یک به طول مساوی طول والدین) به طور تصادفی انتخاب می‌شود. حال ژنهایی از والدین را جابجا می‌کنیم که بیت‌های به آنها در الگو برابر یک باشند، مانند زیر:

الگو با نمونه	۱	۱	۰	۱	۱	۰	۰	۰	۱	۰	۱
---------------	---	---	---	---	---	---	---	---	---	---	---

حال با توجه به الگو و مطالب گفته شده دو فرزند زیر حاصل می‌شوند.

فرزند اول	۱	۰	۱	۰	۱	۰	۱	۱	۱	۱	۱
فرزند دوم	۰	۱	۱	۱	۰	۱	۰	۰	۰	۰	۰

۵-۳-۳- عملگر crossover بر زدن

عملگر crossover بر زدن مانند عملگر crossover یکنواخت است با این تفاوت که مقدار متغیرهای والدین (والدین اول و والدین دوم) قبل از باز ترکیبی بر زده می‌شود و مقادیر متغیرهای والدین جابجا می‌شود بعد در عمل باز ترکیبی شرکت می‌کنند. روش کار این عملگر کمی شبیه عملگر crossover یکنواخت است با این تفاوت که دیگر نیاز به الگو نداریم.

فصل ششم

جهش

۶-۱- جهش مقادیر حقیقی

۶-۲- جهش مقادیر باینری

به وسیله جهش افراد به طور تصادفی اصلاح و دگرگون می‌شوند. این تغییرات خیلی کوچک است. ایت تغییرات روی متغیرهایی اعمال می‌شود که دارای احتمال جهش پایین باشد. به طور معمول بعد از باز ترکیبی، فرزندان تحت جهش قرار می‌گیرند. و متغیرهای فرزندان به اندازه یک مقدار تصادفی کوچک افزایش پیدا می‌کند که این مقدار را گام جهش می‌گویند.

۶-۱- جهش مقادیر حقیقی

جهش مقادیر حقیقی بدین معنی است که به طور تصادفی مقدار عددی ایجاد می‌شود که به متغیرهای با احتمال پایین جمع می‌شود. بنابراین احتمال جهش یک متغیر و اندازه تغییر برای جهش متغیر (گام جهش) باید تعریف شود.

احتمال جهش یک متغیر با تعداد متغیر (بعد فرد) رابطه عکس دارد هر چه بعد یک فرد بیشتر باشد، احتمال جهش آن کمتر است. مقالات مختلف نتایج مختلف برای میزان جهش بهینه، گزارش داده‌اند. در بیشتر این مقالات میزان جهش را $1/n$ (n تعداد متغیرهای یک فرد) در نظر گرفته‌اند که نتایج خوبی برای یک نوع وسیع از تابع هدف بدست آمده است. این بدین معنی است که تنها یک متغیر هر فرد در هر بار جهش تغییر می‌کند. بنابراین میزان جهش مستقل از اندازه جمعیت است. نتایج مشابهی نیز برای جهش با مقادیر باینری وجود دارد.

اندازه گام جهش معمولاً به طور مختلف انتخاب می‌شود. بهترین اندازه جهش وابسته به مسئله‌ایست که در نظر می‌گیریم و ممکن است در هنگام اجرای الگوریتم و رسیدن به جواب بهینه تغییر کند. مشخص است که جهش با اندازه گام کوچک اغلب کافی است مخصوصاً وقتی که میزان سازگاری افراد جامعه بالا باشد. بنابراین یک عملگر جهش خوب، اغلب توسط اندازه گامهای کوچک تولید می‌شود.

رابطه (۶-۱) عملگر جهش برای مقادیر حقیقی به صورت زیر می‌باشد:

$$Var_i^{mt} = Var_i + s_i \cdot r_i \cdot a_i \quad i \in \{1, 2, \dots, n\} \text{ unificim at random}$$

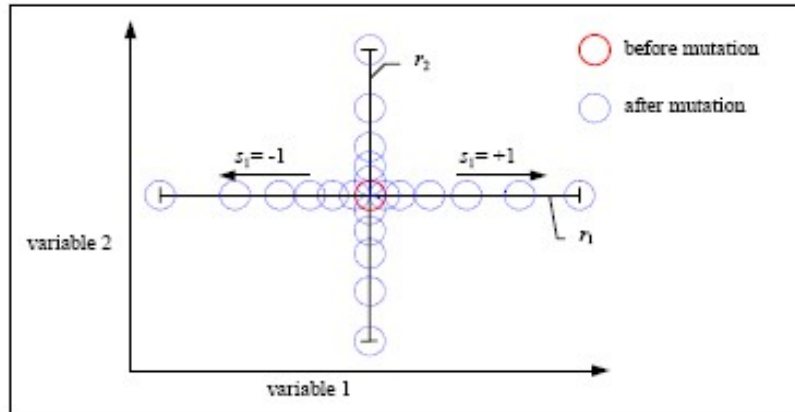
$$s_i \in \{-1, +1\} \text{ uniform at random}$$

$$r_i = r \text{ domain}_i \cdot r.mutationrange(s \text{ tandard} : 10\%)$$

$$a_i = 2^k, u \in [0, 1] \text{ uniform at random, } \lambda \text{ mutaion precision}$$

بیشتر افراد جهش یافته مقادیرشان نزدیک به افراد قبل از جهش است. تنها تعداد کمی از افراد جهش یافته از افراد قبل از جهش دور می‌باشند این بدین معنی است که احتمال جهش با اندازه گام کوچک بیشتر از احتمال جهش با اندازه گام بزرگتر است.

شکل زیر سعی کرده است که تاثیر عملگر جهش را روی افراد نشان دهد.



شکل ۶-۱ جهش مقادیر حقیقی

پارامتر k (دقت جهش) به طور مستقیم به صورت مینیمم اندازه گام ممکن، تعریف می شود و کوچکترین اندازه گام برابر 2^{-k} است و بزرگترین آن برابر $2^0 = 1$ بنابراین گام های جهش داخل بازه $r, r * 2^{-ki}$ که میزان جهش قرار می گیرد. با دقت جهش $k=16$ کوچکترین گام جهش ممکن برابر $r * 2^{-16}$ می باشد. بنابراین وقتی متغیرهای یک فرد خیلی نزدیک به بهینه است یک مقدار جهش زیاد لازم نیست. نوع های مختلف برای پارامترهای عملگر جهش برای معادله (۶-۱) موجود است.

$$\text{mutation precision } k : k \in \{4, 5, \dots, 20\}$$

$$\text{mutation ranger} : r \in [0.1, 10^{-6}]$$

۶-۲- جهش مقادیر باینری

برای افراد با متغیرهای باینری [2]، جهش وسیله برای تغییر سریع مقادیر متغیرها زیرا هر متغیر دارای دو موقعیت است بنابراین اندازه گام همیشه برابر ۱ می باشد. برای هر فرد مقدار متغیر به طور تصادفی انتخاب می شود. نمودار زیر روش جهش یک فرد با ۱۱ متغیر را نشان می دهد به طوری که متغیر شماره ۴ انتخاب شده است.

روش بدین صورت است که ابتدا عددی تصادفی در بازه یک و طول فرد (تعداد متغیرهای فرد) در نظر می گیریم اگر متغیر فرد برابر ۱ است به صفر تبدیل می شود و اگر صفر باشد با ۱ عوض می شود. مانند مثال بالا عدد ۴ را به طور تصادفی در بازه [۱, ۱۱] در نظر می گیریم و محتوای خانه ۴ را عوض می کنیم.

before mutation	0	1	1	1	0	0	1	1	0	1	0
				↓							
after mutation	0	1	1	0	0	0	1	1	0	1	0

فصل هفتم

دوباره گنجاندن دادن

۷- دوباره گنجاندن

۷-۱- دوباره گنجاندن عمومی و کلی

۷-۲- دوباره گنجاندن محلی

زمانی که فرزندی از والدینش توسط عملگرهای انتخاب، باز ترکیبی و جهش به وجود می‌آید سپس برازندگی افراد محاسبه و ارزیابی می‌شود. برازندگی فرزند تعیین کننده است و اگر فرزند دارای برازندگی کم باشد از جامعه حذف می‌شود لذا اگر تعداد فرزندان تولید شده بیشتر از اندازه جمعیت عمومی، کافی است افراد جامعه قبلی را داخل جمعیت فعلی دوباره بگنجانیم. به طور مشابه اگر تعداد فرزندان تولید شده بیشتر از جمعیت عمومی باشد در این صورت یک طرح دوباره گنجاندن استفاده می‌شود تا اندازه جمعیت فعلی به اندازه جمعیت عمومی برسد.

از عملگر انتخاب برای دوباره گنجاندن استفاده می‌شود مانند دوباره گنجاندن محلی که از عملگر انتخاب محلی و دوباره گنجاندن عمومی و کلی که از همه دیگر عملگرهای انتخاب استفاده می‌کند.

۷-۱- دوباره گنجاندن عمومی و کلی

طرح‌های مختلف از دوباره گنجاندن عمومی و کلی [4] موجود است که عبارتند از :

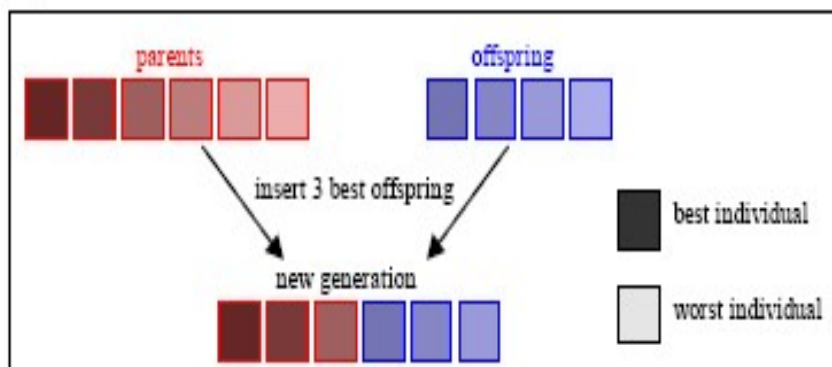
- دوباره گنجاندن خالص: تولید تعداد زیادی فرزند از والدینشان و جایگزین کردن فرزندان به جای والدینشان
- دوباره گنجاندن یکنواخت: تولید فرزند کم از والدین و جایگزین کردن والدین به طور تصادفی به جای فرزندانشان
- دوباره گنجاندن برگزیده: تولید فرزند کم از والدین و جایگزین کردن والدین قوی
- دوباره گنجاندن بر اساس برازندگی: تولید بیش از حد نیاز فرزندان و دوباره گنجاندن فقط فرزندان بهتر.

دوباره گنجاندن خالص، ساده‌ترین روش دوباره گنجاندن است. در این طرح هر فرد فقط به اندازه یک نسل زندگی می‌کند. این طرح بیشتر در الگوریتم‌های ژنتیک ساده مورد استفاده قرار می‌گیرد. لذا خیلی محتمل است که افرادی با برازندگی کم به جای افراد مناسب با برازندگی بالا قرار گیرد که این باعث از دست دادن اطلاعات خوب می‌شود. حال اگر دوباره گنجاندن برگزیده با دوباره گنجاندن بر اساس برازندگی ترکیب شود باعث جلوگیری از بین رفتن اطلاعات می‌شود و این روش توصیه می‌شود.

شکل ۱-۷ نشان می‌دهد که در هر نسل یک تعداد از والدین با برازندگی کم با تعدادی از فرزندان با برازندگی بالا جابجا می‌شود.

الگوی دوباره گنجاندن بر اساس برازندگی، یک انتخاب به روش هرسی بین فرزندان قبل از دوباره جایگزینی آنها در جامعه صورت می‌دهد یعنی قبل از اینکه آنها بتوانند در روند تولید دوباره

شرکت کنند) به عبارت دیگر افراد بهتر می‌توانند تعداد نسل‌های زیادی زندگی کنند. به هر جهت، در هر نسل بعضی افراد جدید دوباره جایگزین می‌شوند و این می‌تواند باعث کاهش میانگین برآزندگی جامعه شود به هر جهت اگر هم فرزندان ناخلف دوباره گنجانده شود در نسل بعدی آنها توسط فرزندان جایگزین می‌شوند.



شکل ۱-۶ الگویی برای دوباره گنجاندن

۶-۲- دوباره گنجاندن محلی

در انتخاب محلی، در یک همسایگی کراندار شده انتخاب می‌شوند و در همان همسایگی، دوباره گنجاندن اتفاق می‌افتد. بنابراین محلی بودن اطلاعات پایدار است. ساختار همسایگی استفاده شده همان ساختار انتخاب محلی است. و همانند آن ابتدا والدین فرزند در آن همسایگی انتخاب می‌شود و سپس عمل دوباره گنجاندن صورت می‌گیرد. برای انتخاب والدین‌ها برای دوباره گنجاندن و همچنین برای انتخاب فرزندان برای دوباره گنجاندن الگوهای زیر استفاده می‌شود.

- انتخاب هر فرزند و جایگزین کردن بجای افراد داخل همسایگی به طور یکنواخت و تصادفی
- انتخاب هر فرزند و جایگزین کردن به جای افراد ضعیفتر داخل همسایگی
- انتخاب فرزند مناسب تر از افراد ضعیفتر در همسایگی، و جایگزین کردن به جای افراد ضعیفتر در همسایگی
- انتخاب فرزند مناسبتر از افراد ضعیفتر در همسایگی، و جایگزین کردن به جای افراد به طور یکنواخت و تصادفی
- انتخاب فرزند مناسبتر از والدین و جایگزین کردن به جای والدین

فصل هشتم

معیار همگرایی یا توقف الگوریتم

همانند دیگر الگوریتم‌های بهینه‌سازی دیگر، در اینجا نیز باید برای توقف الگوریتم باید از معیار همگرایی استفاده کنیم. کلا شرطهای توقف بر دو نوع اند. [۴] و [۵]:

۱. شرطهای غیرفعال که توقف الگوریتم مستقل از نتیجه بدست آمده از الگوریتم است این شرط را می‌توان به صورت گذشت تعداد ثابتی نسل یا گذشت مدت زمان معینی در هنگام اجرای الگوریتم توسط رایانه، تعریف کرد.

۲. شرطهای دنباله‌ای که توقف الگوریتم به جواب بدست آمده از آن وابسته است. شرطی از این نوع عبارت است از اینکه جمعیت از حد معینی یکنواخت‌تر شود. برای مثال اگر ۹۸ درصد ژنهای دارای موقعیت مکانی مشابه و مقداری برابر داشته باشند، آن الگوریتم همگرا شده فرض می‌شود. این شرط می‌تواند اشتباه برانگیز باشد زیرا ممکن است بر اثر انتخاب نامناسب پارامترهای الگوریتم، الگوریتم خیلی زود به بهینه مناسب برسد.

شرط دیگری را می‌توان با در نظر گرفتن تغییرات برازندگی بهترین فرد جمعیتی اجرا نمود، به این معنا که اگر برازندگی در طی چند نسل از حدی بهبود نیابد، اجرای الگوریتم متوقف می‌شود. دنباله‌ای اطمینان‌بخش تکاملی از هر دو نوع شرط توقف استفاده می‌شود ولی اغلب شرط‌های دنباله‌ای اطمینان‌بخش‌تر از شرطهای دیگر (غیرفعال) است. ولی باز هم نمی‌توان فاصله جواب بدست آمده را به جواب بهینه کلی تضمین کنند.

ما در حالت کلی از سه شرط توقف در این الگوریتم‌ها استفاده می‌کنیم که عبارتند از :
الف- اگر درصد اختلاف بین میانگین برازندگی تمام افراد جمعیت و برازندگی بهترین فرد جمعیت به مقدار بسیار کوچک C_{min} برسد، الگوریتم متوقف می‌شود. بنابراین :

$$\left| \frac{f^- - f_i(best)}{f^-} \right| \times 100 \leq C_{min}$$

بطوریکه

$$f^- = (\sum_{i=1}^N f_i) / N$$

در رابطه قبل f^- میانگین برازندگی جمعیت در یک نسل می‌باشد و $f_i(best)$ مقدار تابع هدف برازنده‌ترین کروموزوم، C_{min} نسبت همگرایی و N اندازه جمعیت می‌باشد.

ب- در صورتی که تعداد نسل که همان متغیر کنترل حلقه می‌باشد به میزان حداکثر مقدار خود که کاربر در نظر گرفته است برسد، اجرای الگوریتم متوقف می‌شود.

ج- در صورتیکه اجرای الگوریتم توسط رایانه از یک مدت زمان مشخصی که توسط کاربر تعیین شده است بگذرد، کاربر رایانه را متوقف می‌کند و لذا الگوریتم متوقف می‌شود. [۲]

فصل نهم

احتمال‌های وفقی عملگرهای باز ترکیبی و جهش

برای بهینه‌ساز یا الگوریتم‌های تکاملی ضروریست که خاصیت را داشته باشیم. اولین مشخصه عبارت است از توانایی و ظرفیت همگرا شدن الگوریتم به یک جواب بهینه (مطلق یا نسبی) می‌باشد و دومین مشخصه توانایی و ظرفیت جستجوی فضاها و ناحیه‌های جدید برای یافتن نقطه بهینه می‌باشد. تعادل میان این دو مشخصه از الگوریتم‌های تکاملی توسط مقادیر P_c, P_m و نوع عملگر تقاطع که به کار گرفته می‌شود، دارد. [۳]

افزایش مقادیر P_c, P_m میزان فضاهای جستجو را افزایش می‌دهد اما باعث از دست رفتن قابلیت دوم - یعنی توانایی همگرا شدن الگوریتم به جواب بهینه - می‌شود. به طور معمول مقادیر نسبتاً بزرگ احتمال باز ترکیبی $P_c (5-10)$ و مقادیر کوچک (0.5 و 0.001) برای احتمال جهش P_m در عمل برای الگوریتم‌های تکاملی مورد استفاده قرار می‌گیرد.

در اینجا هدف ما این است که با بکارگیری مقادیر متغیر P_c, P_m به صورت وفقی مرتبط با برازندگی نقاط تعادلی بین جستجوی جواب‌های جدید و همگرایی الگوریتم بیابیم بدین صورت که مقادیر P_c, P_m زمانی که الگوریتم تمایل به سمت نقطه بهینه موضعی دارند، افزایش یافته و زمانی که پراکندگی در جمعیت زیاد شد، کاهش یابند.

محاسبه P_c, P_m وفقی [۷] و [۶]:

برای تغییر دادن P_c, P_m به صورت وفقی برای جلوگیری از همگرایی زودرس الگوریتم به سوی جواب بهینه موضعی، ضروری است که بتوان تشخیص داد که آیا الگوریتم به سوی یک نقطه بهینه همگرا شده است یا نه.

یکی از راه‌های ممکن برای این کار مشاهده فاصله متوسط برازندگی جمعیت با برازندگی ماکزیمم در جمعیت می‌باشد یعنی $f_{\max} - \bar{f}$. به نظر می‌رسد که این مقدار برای جمعیتی که به سوی یک نقطه بهینه همگرا شده است کمتر از زمانی باشد که جمعیت پراکنده است. همان طور که مشاهده می‌شود $f_{\max} - \bar{f}$ زمانی که الگوریتم به سوی یک نقطه بهینه نسبی همگرا می‌شود کاهش یافته و برعکس. ما از مقدار $f_{\max} - \bar{f}$ به عنوان یک معیار برای تشخیص همگرایی الگوریتم تکاملی استفاده می‌نمائیم.

از آنجائی که بایستی P_c, P_m زمانی که الگوریتم تکاملی به سوی یک نقطه بهینه نسبی همگرا شده است افزایش یابند. بنابراین مقادیر آنها بایستی عکس مقدار $f_{\max} - \bar{f}$ تغییر یابد. بنابراین شکلی که ما برای P_c, P_m در نظر می‌گیریم، به صورت زیر است :

$$P_c = \frac{K_1}{f_{\max} - \bar{f}}$$

$$P_m = \frac{K_2}{f_{\max} - \bar{f}}$$

همانطور که مشاهده می‌شود در حالت فوق مقادیر P_c, P_m بستگی به مقدار برازندگی جواب خاصی از مسئله ندارد و برای تمام نقاط جواب مسئله به صورت یکسان تغییر می‌یابد. در نتیجه جواب‌های با برازندگی بالا و جواب‌های با برازندگی پایین به یک نسبت تحت عمل بازترکیبی و جهش قرار می‌گیرند. زمانی که الگوریتم به سوی یک جواب بهینه مطلق یا نسبی همگرا می‌شود، مقادیر P_c, P_m افزایش یافته و باعث شکستن جواب‌های با برازندگی بالا و جواب‌های با برازندگی پایین به یک نسبت تحت عمل بازترکیبی و جهش قرار می‌گیرند. زمانی که الگوریتم به سوی یک جواب بهینه مطلق یا نسبی همگرا می‌شود، مقادیر P_c, P_m افزایش یافته و باعث شکستن جواب‌های نزدیک به بهینه می‌گردند. در این حالت ممکن است جمعیت هیچگاه به سوی جواب بهینه همگرا نگردد. با وجود اینکه ما از گیرافتادن الگوریتم در دام نقاط بهینه نسبی (موضعی) جلوگیری کرده‌ایم اما کارائی الگوریتم تکاملی را از بین خواهد برد.

برای مقابله با مشکل فوق نیاز داریم که جواب‌های خوب را نگه داریم و این با داشتن مقادیر پایین P_c, P_m برای جواب‌های با برازندگی بالا و مقادیر بالای P_c, P_m برای جواب‌های با برازندگی پایین به دست می‌آید. هنگامی که جواب‌ها با برازندگی بالا به همگرایی الگوریتم تکاملی کمک کرده و جواب‌های با برازندگی پایین از به دام افتادن الگوریتم در دام نقاط بهینه موضعی جلوگیری می‌نماید. مقادیر P_m بایستی نه تنها به مقدار $f_{\max} - \bar{f}$ بلکه به مقدار برازندگی هر جواب بستگی داشته باشد. به طور مشابه P_c بایستی به مقادیر برازندگی هر دو والد بستگی داشته باشد. هرچه مقدار f به f_{\max} نزدیک‌تر باشد مقدار P_m کمتری باید داشته باشیم، بنابراین P_m به طور مستقیم با مقدار $f_{\max} - \bar{f}$ تغییر خواهد کرد و به طور مشابه مقدار P_c بایستی مستقیماً با مقدار $f_{\max} - \bar{f}$ تغییر کند. در جایی که f بزرگترین مقدار برازندگی جواب‌هایی است که عمل بازترکیبی روی آنها انجام شده است بنابراین شکل P_c, P_m اکنون به صورت زیر می‌باشد [5]:

$$P_c = \frac{k_1 \left(f_{\max} - f \right)}{\left(f_{\max} - \bar{f} \right)} \quad k_1 \leq 1$$

$$P_m = \frac{k_2 \left(f_{\max} - f \right)}{\left(f_{\max} - \bar{f} \right)} \quad k_2 \leq 1$$

(مقادیر k_1, k_2 بایستی کوچکتر از یک باشند تا مقادیر P_c, P_m در فاصله ۰ و ۱ قرار گیرند). در اینجا بایستی توجه کرد که مقادیر P_c, P_m بایستی برای جواب‌های با برازندگی حداکثر، مقدار صفر به خود گیرند. همچنین $P_c = k_1$ برای جواب‌هایی که $f = \bar{f}$ ، $P_c = k_2$ برای جواب‌هایی که $f = f_{\max}$. برای جواب‌های با مقدار برازندگی زیر حد متوسط، برازندگی جمعیت، ممکن است به نظر رسد که مقادیر P_c, P_m از مقدار ۱ بیشتر می‌شود. برای جلوگیری از این مسئله ما دو محدودیت زیر را اضافه می‌نمائیم.

$$P_c = k_3 \quad f \leq \bar{f}$$

$$P_m = k_4 \quad f \leq \bar{f}$$

در جایی که $k_3, k_4 < 1$ هستند.

ملاحظات عملی و انتخاب مقادیر k_1, k_2, k_3, k_4 .

در قسمت قبل دیدیم که برای یک جواب با مقادیر برازندگی حداکثر، P_c, P_m هر دو معادل صفر بودند. بهترین جواب جمعیت بدون هیچگونه دستکاری به نسل بعد منتقل می‌گردد. همراه با مکانیزم انتخاب، این عمل ممکن است به یک رشد نمایی جواب‌ها در جمعیت منجر شود و ممکن است باعث همگرایی زودرس گردد. برای غلبه بر این مشکل یک نرخ تعریف شده برای جهش معادل (0/005) برای تمام جواب‌ها معرفی می‌نمائیم [7].

حال به انتخاب مقادیر k_1, k_2, k_3, k_4 می‌پردازیم. برای راحتی کار مقادیر P_c, P_m را به صورت زیر در نظر می‌گیریم:

$$\left\{ \begin{array}{l} P_c = \frac{k_1 (f_{\max} - f)}{(f_{\max} - \bar{f})} \quad f \geq \bar{f} \\ P_c = k_3 \quad f \leq \bar{f} \\ P_m = \frac{k_2 (f_{\max} - f)}{(f_{\max} - \bar{f})} \quad f \geq \bar{f} \\ P_m = k_4 \quad f \leq \bar{f} \end{array} \right.$$

که در اینجا $k_1, k_2, k_3, k_4 < 1$ هستند.

همانطور که قبلاً گفتیم مقادیر نسبتاً بالای $p_c (0.5 < p_c < 1)$ و مقادیر کوچک $p_m (0.001 < p_m < 0.5)$ برای عملکرد موفق الگوریتم تکاملی ضروری به نظر می‌رسد. مقادیر نسبتاً بزرگ P_c تکثیر وسیع نمونه‌ها را ترقی می‌دهد، در حالی که مقادیر کوچک P_m برای جلوگیری از تخریب جواب‌ها لازم می‌باشند. این راهنمایی‌ها برای زمانی که متغیر نیستند مفید می‌باشد [3].

یکی از اهداف این روش جلوگیری از به دام افتادن الگوریتم تکاملی در دام نقاط بهینه موضعی می‌باشد. برای رسیدن به این هدف ما نقاط بهینه با برازندگی زیر برازندگی متوسط را به کار می‌گیرند تا فضاهای جستجوی مختلفی را جستجو کنیم تا به ناحیه برسیم که نقطه بهینه در آن قرار دارد. بعضی از این

جوابها کلا بایستی تخریب گردند، بنابراین ما مقدار $0/5$ را برای k_4 انتخاب می کنیم. همچنین جوابهای با مقدار برازندگی \bar{f} بایستی کاملا بهم بریزیم، بنابراین مقدار $0/5$ را به k_2 اختصاص می دهیم. بنا به دلایل قبلی مقدار یک را به k_1, k_2 اختصاص می دهیم، بدین طریق مطمئن هستیم که کلیه جوابها با مقدار برازندگی کمتر یا مساوی \bar{f} اجرا یا تحت عمل بازترکیبی قرار می گیرند. احتمال بازترکیبی هنگامی که مقدار برازندگی به سمت f_{\max} میل کند کاهش یافته و برای جوابهای با برازندگی f_{\max} مقدار این احتمال برابر صفر است.

فصل دهم

نتیجه گیری و ارائه مدل

۱-۱- معرفی الگوریتم های تکاملی مدل حمل و نقل

الگوریتم های تکاملی شاخه ای از تکنیک های بهینه سازی اجتماعی هستند که در آنها از سیستم های بیولوژیکی و اصول حاکم بر آنها بهره گیری شده است. اگر چه این الگوریتم ها ارائه دهنده مدل های ساده ای از فرایند بیولوژیکی در شرایط طبیعی هستند، اما در عمل توانایی و کارایی زیادی از خود نشان داده اند. ایده اولیه ارائه الگوریتم های تکاملی استفاده از جمعیت های محدودی از عناصر که هر یک از آنها دقیقا نقطه ای از فضای جستجو را مشخص می کنند، می باشد. که در آن الگوریتم افراد جامعه به وسیله کروموزوم ها مشخص می شوند [۷]. جمعیت کروموزوم ها (افراد جامعه) بعد از این وارد مرحله شبیه روند تکاملی موجود در طبیعت می شوند. ابتدا بطور تصادفی جامعه ای از کروموزوم ها پدید می آید و سپس عملگرهای نو ترکیبی و جهش و انتخاب و دیگر عملگرهای تکاملی را بسته به نیاز، روی این کروموزوم ها اعمال کرده که نسل جدیدی از کروموزوم ها حاصل می شود و سپس برای این نسل جدید معیارهای بهینگی [۶]، بررسی می شود. اگر برای این نسل جدید معیارهای بهینگی برآورده شود که الگوریتم متوقف می شود و بهترین کروموزوم نسل را به عنوان جواب بهینه مسئله و مدل ترابری بدست می آید. در غیر این صورت عملگرهای مختلف ژنتیکی بر روی کروموزوم های داخل جمعیت اعمال می شود و نسل جدیدی حاصل می شوند و این مراحل را تا برآورده شدن معیارهای بهینگی ادامه پیدا می کند. برتری الگوریتم های تکاملی نسبت به روشهای دیگر بهینه سازی باعث شود که تمایل به این الگوریتم ها بیشتر شده که از این جمله می توان به موارد زیر اشاره کرد [۵].

۱- الگوریتم های تکاملی جستجو روی نقاط یک نسل را بطور موازی انجام می دهند ولی دیگر روشها بصورت نقطه ای عمل می کنند.

۲- الگوریتم های تکاملی نیاز به اطلاعات و داده های زیادی ندارند و تنها نیاز به تابع هدف مسئله دارند.

۳- الگوریتم های تکاملی برای کاربران بسیار ساده و قابل فهم می باشند.

از ابتدای بوجود آمدن الگوریتم های تکاملی صاحب نظران این رشته سعی کردند که این الگوریتم ها را

ساده تر و قابل فهم تر کنند و نیز مدل هایی از الگوریتم های تکاملی ارائه دهند که دارای عملیات کمتر و نیز دارای قابلیت های زیاد در زمینه های مختلف بهینه سازی باشند و خروجی های این الگوریتم ها، بهترین جواب را برای مسئله مورد نظر ارائه دهد.

۲-۱۰- الگوریتم ویژه مسائل حمل و نقل

مدل ارائه شده ، الگوریتمی تکاملی است که برای بهینه سازی مسائل ترابری قابل استفاده است. فرض کنید

مسئله حمل و نقل با مفروضات زیر داده شده است [۳]:

$$a_1=12 \ \& \ a_2=8 \ \& \ a_3=16 \ \& \ a_4=10$$

$$b_1=6 \ \& \ b_2=8 \ \& \ b_3=12 \ \& \ b_4=10 \ \& \ b_5=8$$

همانطور که در بحث الگوریتم های تکاملی اشاره شد، ابتدا در شروع این الگوریتم ها نیاز به تعدادی جواب شدنی است که همگی با توجه به محدودیت های مدل ریاضی مسئله ترابری حاصل می شود. لذا بصورت تصادفی تعدادی جواب شدنی (جواب هایی که در تمام محدودیت های مسئله صدق می کنند) ایجاد می شود. که این مجموعه از جوابها را نسل صفر یا نسل آغازین گویند که همگی بصورت جدول حاصل می شوند و در مراحل مختلف الگوریتم بوسیله اسامی مانند کروموزوم، والدین، فرزند و فردی از جمعیت نامیده می شوند. یک نمونه از جوابهای شدنی مانند جدول زیر داده شده است، که در آن مجموع اعداد داخل هر سطر باید با عدد روبرو و مجموع اعداد داخل هر ستون باید با عدد زیر آن ستون برابر باشد [۲].

6	6	0	0	0	12
0	2	6	0	0	8
0	0	6	10	0	16
0	0	0	0	10	10
6	8	12	10	10	

پس از ایجاد نسل صفر یا نسل آغازین مقدار تابع هدف نظیر به هر کدام از جواب ها را بدست می آید. حال طبق الگوریتمی که برای ساختار کلی الگوریتم تکاملی ارائه شده است، ابتدا مقدار برازندگی جوابها را بدست می آید و سپس معیار بهینگی را برای تک تک جواب ها بررسی می شود اگر برای این نسل از جواب ها معیارهای بهینگی برآورده شد که الگوریتم متوقف می شود و بهترین جواب از بین جواب های این نسل انتخاب می شود که همان جواب بهینه مسئله است، در غیر این صورت سعی می شود که از روی این نسل، نسل جدیدی ایجاد شود که شرایط بهینگی را برآورده کند. برای این کار، کافی است که از عملگرهای تکاملی - ژنتیکی استفاده شود. که در زیر در مورد هر یک بطور کامل توضیح داده شده است و با این عملگرها نسل بهتری نسبت به نسلهای قبلی بدست می آید و این روند را تا برآورده شدن شرایط بهینگی ادامه می یابد.

۳-۱۰- عملگر selection

عملگر جدیدی که در این قسمت معرفی شده است حاصل از ترکیب عملگر انتخاب چرخ گردان و همچنین تکنیک رتبه بندی کروموزوم ها می باشد [۶]. در این عملگر ابتدا به هر یک از کروموزوم های نسل یک رتبه داده می شود که روند رتبه بندی برخلاف رتبه بندی متداول می باشد بدین صورت که بدترین کروموزوم رتبه اول و بهترین کروموزوم رتبه آخر می باشد و براساس فرمول زیر برازندگی مربوط به کروموزوم های یک نسل بدست می آید، که در آن sp یک عدد تصادفی بین اعداد یک و دو می باشد و pos رتبه کروموزوم در نسل می باشد.

$$fitness(pos) = 2 - SP + 2 \times (SP - 1) \times \frac{(POS - 1)}{(N - 1)}$$

حال فرض کنید تعداد کروموزوم های نسل آغازین ما برابر با n باشد در اینصورت رتبه بدترین جواب برابر با یک و به همین ترتیب رتبه بهترین جواب برابر با n می باشد و مطابق فرمول فوق مقدار برازندگی

آنها بدست می آید. مانند f_1, f_2, \dots, f_n حال براساس زیر عمل می شود. ابتدا مقدار

$$F = \sum_{i=1}^n f_i$$

بدست می آید و سپس احتمال انتخاب شدن هر جواب را بصورت زیر بدست می آید:

$$p_i = \frac{f_i}{F}$$

و بر اساس آن احتمال تجمعی هر یک از جوابها بصورت زیر محاسبه می شود.

$$q_1 = p_1, q_2 = p_2 + q_1, \dots, q_n = q_{n-1} + p_n$$

و سپس آنها را روی یک پاره خط بطول واحد از صفر تا یک تصویر می شود.

حال به تعداد انتخابی که باید انجام شود، عدد تصادفی در فاصله $[0, 1]$ انتخاب می شود جوابهایی که نظیر به این اعداد هستند انتخاب می شوند. واضح است که جوابهای قویتر، برازندگی بزرگتری دارند و احتمال انتخاب آنها نسبت به سایر جوابهای یک نسل بیشتر است و این براساس اصل انتخاب طبیعی داروین می باشد که جانداران قویتر باقی می مانند و ضعیف ها محکوم به فنا هستند. حال فرض کنید توسط عملگر انتخاب مدل ترابری تعداد n کروموزوم جدید انتخاب شده است که ممکن است تعداد زیادی از آنها تکراری باشند. حال بعد از اعمال عملگر انتخاب، عملگر نو ترکیبی را بر روی جوابهای این نسل اثر می کند تا کروموزومهای جدیدی حاصل شود.

۴-۱۰ - عملگر recombination

روش کلی این عملگر بدین صورت است که ابتدا بطور تصادفی کروموزوم ها جفت جفت از یک نسل انتخاب و هر دو کروموزوم به عنوان دو والدین در نظر گرفته می شوند و مدل طراحی شده، ژنهای مربوط به والدین تغییر پیدا می کنند و باعث بوجود آمدن جوابهای جدید می شوند که به آنها فرزند گفته

می شود [4]. این عمل را می توان به دفعات مختلف انجام داد که باعث تنوع کروموزوم ها در یک نسل می شود. عملگر نوترکیبی عملگر جدیدی است که بر اساس خواص مدل ترابری طراحی شده است که در مدل ترابری مورد استفاده قرار می گیرد [3].

جدول ۱: جوابی از سیستم به عنوان والدین اول X_1 جدول ۲: جوابی از سیستم به عنوان والدین دوم X_2

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

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

حال با توجه به ماتریسهای X_1 و X_2 ، ماتریس R را بصورت زیر بدست می آید، بطوریکه:

$$R_{ij} = ((x_{ij})_1 + (x_{ij})_2) \bmod 2$$

و طبق فرمول زیر ماتریس D بصورت زیر بدست می آید:

$$d_{ij} = \left[\frac{(x_{ij})_1 + (x_{ij})_2}{2} \right]$$

جدول ۳: ماتریس R که طبق فرمول فوق بدست آمده است

0	1	0	0	1
0	0	1	0	0
1	0	1	0	0
0	1	0	0	1

جدول ۴: ماتریس D

0	2	5	3	1
1	3	1	1	1
1	1	2	6	5
3	0	3	0	2

حال ماتریس R را به دو ماتریس R_1 و R_2 طوری تقسیم می شود، بطوریکه:

$$R = R_1 + R_2$$

جدول ۵: ماتریس R_1

0	0	0	0	1
0	0	1	0	0
0	0	1	0	0
0	1	0	0	0

جدول ۶: ماتریس R_2

0	1	0	0	0
0	0	0	0	0
1	0	0	0	0
0	0	0	0	1

حال فرزندان X_1', X_2' بصورت زیر بدست می آیند بطوریکه داشته باشیم:

$$X_1' = D + R_1 \quad \& \quad X_2' = D + R_2$$

جدول ۷: جدول X_1' به عنوان فرزند اول

0	2	5	3	2
1	3	2	1	1
1	1	3	6	5
3	0	3	0	2

جدول ۸: جدول X_2' به عنوان فرزند دوم

0	3	5	3	1
1	3	1	1	1
2	1	2	6	5
3	0	3	0	3

طبق روند فوق از طریق عملگر نو ترکیبی ، دو جواب به عنوان دو والدین بطور تصادفی انتخاب و براساس آنها دو جواب جدید به عنوان دو فرزند حاصل می شود. این روند را تا آنجا ادامه می یابد که تعداد $2n$ کروموزوم جدید بدست آید. این عمل باعث تکثیر جواب های با برازندگی بالا به جواب های دیگر می شود لذا تنوع جواب ها زیادتیر می شود [5].

۵-۱۰- عملگر mutation

روند اجرائی این عملگر بدین صورت است که ابتدا بصورت تصادفی یک کروموزوم از یک نسل انتخاب می شود و این کروموزوم به عنوان والدین محسوب و توسط روشی که در زیر به آن اشاره می شود به کروموزوم جدیدی به نام فرزند تبدیل می شود [7]. مانند مثال زیر

جدول ۹: نمونه ای از جواب برای مدل ترابری

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

چون ماتریس والدین ما یک ماتریس 4×5 می باشد سعی می شود یک زیر ماتریس Y از والدین X را طوری بدست آید طوری که تعداد سطر و ستون آن کمتر از ماتریس X باشد. روند کلی آن بدین صورت است که ابتدا یک عدد طبیعی در فاصله $[1 \ 4]$ انتخاب می شود، حال به تعداد آن عدد، باز هم عددهای تصادفی در فاصله $[1 \ 4]$ انتخاب می شود. سطرهای نظیر آن اعداد در ماتریس X در نظر گرفته و به طریق مشابه برای ستونها نیز همین روند را انجام می گیرد. بدین صورت است که ابتدا یک عدد طبیعی در فاصله $[1 \ 5]$ انتخاب می شود، حال به تعداد آن عدد باز هم عددهای تصادفی در فاصله $[1 \ 5]$ انتخاب می شود. ستونهای نظیر آن اعداد در ماتریس X در نظر گرفته می شود مانند ماتریس زیر

جدول ۱۰: زیر ماتریس Y

4	0	2
1	0	3

حال براساس زیر ماتریس فوق زیر ماتریس جدیدی تولید می گردد که مجموع درایه های عناصر هر سطر یا هر ستون آن برابر با مجموع درایه سطر یا ستون نظیر آن در زیر ماتریس Y باشد. بعبارت دیگر زیر ماتریسی تولید می شود که درایه های آن در معادلات زیر که از زیر ماتریس Y بدست آمده اند صدق کنند.

$$\begin{cases} y_{11} + y_{12} + y_{13} = 6 \\ y_{21} + y_{22} + y_{23} = 4 \\ y_{11} + y_{21} = 5 \\ y_{12} + y_{22} = 0 \\ y_{13} + y_{23} = 5 \end{cases}$$

در این صورت زیر ماتریس جدیدی مانند Y' طبق دستگاه معادلات فوق بدست می آید.

جدول ۱۱: زیر ماتریس Y'

2	0	4
3	0	1

بعد از این مرحله ماتریس جدید Y' را به جای زیر ماتریس Y در والدین X قرار می‌گیرد و فرزند جدید زیر بدست می‌آید [۴].

جدول ۱۲: نمونه ای از کروموزوم جهش یافته

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

حال عمل جهش را به اندازه n بار تکرار می‌شود تا n جواب جدید جهش یافته بدست آید.

۱۰-۶- مراحل اجرایی الگوریتم

حال ابتدا براساس ناحیه شدنی مسئله نسل صفر یا نسل آغازین را بصورت تصادفی ایجاد می‌کنیم و مقدار برازندگی آنها را بدست می‌آوریم سپس برای این نسل معیارهای بهینگی را بررسی کرده اگر معیارهای بهینگی برآورده شود که مراحل الگوریتم متوقف می‌شود و بهترین کروموزوم آن نسل به عنوان جواب مسئله انتخاب می‌شود در غیر این صورت عملگر انتخاب بر روی این نسل از کروموزومها اجرا می‌شود و تعداد n کروموزوم جدید انتخاب می‌شود. بعد از این مرحله عملگر باز ترکیبی روی کروموزومهای این نسل اعمال می‌شود و n کروموزوم را به $2n$ کروموزوم تکثیر می‌کند که این باعث تنوع و تکثر در

کروموزمها می شود . بعد از این مرحله عملگر جهش اعمال می شود بدین صورت که n کروموزوم را تبدیل به n کروموزوم جدید می کند. طبق روندی که عملگر جهش دارد. لذا بعد از این مراحل تعداد $4n$ کروموزوم جدید حاصل می شود. حال چون تعداد افراد یک نسل باید ثابت باقی بماند لذا از عملگر انتخاب استفاده می شود و تعداد n کروموزوم از $4n$ کروموزوم موجود انتخاب می شود و این n کروموزوم تشکیل نسل جدیدی را می دهند. حال مانند روندی که در بالا اشاره شد معیارهای بهینگی را برای این نسل بررسی می کنیم اگر معیارهای بهینگی برآورده شود، بهترین کروموزوم انتخاب می شود در غیر این صورت عملگرهای تکاملی بر روی این نسل اعمال می شوند و نسل جدیدی حاصل می شود این مراحل را تا جایی ادامه می دهیم که معیارهای بهینگی برآورده شده و الگوریتم فوق متوقف شود و بهترین جواب حاصل شود. این الگوریتم بر اساس ویژگی های خاص مدل تراپی طراحی شده است و توانایی کاربرد برای مسائل دیگر را نیز دارد [۶].

الگوریتم را در حالت کلی می تواند بصورت زیر گام به گام طراحی کرد

گام اول: ایجاد جوابهایی بطور تصادفی به عنوان نسل آغازین

گام دوم: بدست آوردن مقدار تابع هدف برای این نسل و تعیین برازندگی آنها

گام سوم: ارزیابی معیار بهینگی و در صورت بهینه بودن برو به گام هفتم

گام چهارم: اعمال عملگر بازترکیبی

گام پنجم: اعمال عملگر جهش

گام ششم: اعمال عملگر انتخاب و برو به گام دوم

گام هفتم: بدست آوردن جواب بهینه

منابع

- [1]. Bazara, S.M., and Jarvis, J.J., Linear programming and Net work flows, by John wiley and sons Inc, 1977.
- [2]. Taha, H.A., Operations Research and Introduction, Macmillan, NewYork, 1971.
- [3]. Dantzig, G.B., Linear Programming and Extension, Princeton University Press, Princeton, New Jersey, 1964.
- [4]. Gen, M. and R.Chang,Genetic Algorithms and Engineering designs, John Wiley and Sons,INC.,New Yurok, pp.262-289,1997.
- [5]. Pohlheim, H. Genetic and Evolutionary Algorithm Toolbox for use with matlab.www.geatbx.com, 1994-1999.
- [6]. Mohammadi, m.The introduction of Evolutionary Algorithms. Unpublished manuscript, M.S.Thesis, Department of Mathematics Shahid Bahonar University of Kerman, Kerman, 2003.
- [7]. Deb K., Multi-Objective Optimization using Evolutionary Algorithms, First Edition, John Wiley & Sons Pte Ltd, 2002.