

بسمه تعالی

# الگوریتم‌های بهینه‌سازی مبتنی بر هوش محاسباتی

مؤلفان :

دکتر مهدی جعفری شهباززاده

عضو هیئت علمی دانشگاه آزاد اسلامی واحد کرمان

مهندس بهروز صادقی

عضو هیئت علمی دانشگاه پیام نور واحد تایباد

مهندس فرزاد حسین زاده پیرغیبی

عضو هیئت علمی دانشکده سما دانشگاه آزاد اسلامی واحد بندرعباس

انتشارات ارسطو

( چاپ و نشر ایران )

۱۳۹۴

سرشناسه: جعفری شهباززاده، مهدی، ۱۳۴۶ -  
عنوان و نام پدیدآور: الگوریتم‌های بهینه‌سازی مبتنی بر هوش محاسباتی /  
مولفان مهدی جعفری شهباززاده، بهروز صادقی، فرزاد حسین‌زاده پیرغیبی.  
مشخصات نشر: مشهد: ارسطو، ۱۳۹۴.  
مشخصات ظاهری: ۱۷۵ ص.: مصور، جدول(رنگی)، نمودار.  
شابک: ۹۷۸-۶۰۰-۷۹۴۰-۸۹-۱  
وضعیت فهرست نویسی: فیبا  
موضوع: الگوریتم‌ها  
موضوع: الگوریتم‌های ژنتیک  
موضوع: بهینه‌سازی برنامه‌های کامپیوتری  
موضوع: بهینه‌سازی ترکیبی  
موضوع: بهینه‌سازی ریاضی  
شناسه افزوده: صادقی، بهروز، ۱۳۶۱ -  
شناسه افزوده: حسین‌زاده پیرغیبی، فرزاد، ۱۳۶۱ -  
رده بندی کنگره: ۱۳۹۴ الف ۷ ج ۵۸/۵۸ QA۹  
رده بندی دیویی: ۰۰۶/۳۱  
شماره کتابشناسی ملی: ۴۰۹۵۳۷۱

نام کتاب: الگوریتم‌های بهینه‌سازی مبتنی بر هوش محاسباتی  
مولفان: دکتر مهدی جعفری شهباززاده (عضو هیئت علمی دانشگاه  
آزاد اسلامی واحد کرمان) - مهندس بهروز صادقی (عضو هیئت علمی  
دانشگاه پیام نور واحد تایباد) - مهندس فرزاد حسین‌زاده پیرغیبی (عضو  
هیئت علمی دانشکده سما دانشگاه آزاد اسلامی واحد بندرعباس)

ناشر: ارسطو (چاپ و نشر ایران)

صفحه آرای و تنظیم: پروانه مهاجر

تیراژ: ۱۰۰۰

نوبت چاپ: اول - ۱۳۹۴

چاپ: مدیران

قیمت: ۱۲۰۰۰ تومان

شابک: ۹۷۸-۶۰۰-۷۹۴۰-۸۹-۱

تلفن‌های مرکز پخش: ۳۵۰۹۶۱۴۵ - ۳۵۰۹۶۱۴۶ - ۰۵۱

[www.chaponashr.ir](http://www.chaponashr.ir)



انتشارات ارسطو



چاپ و نشر ایران

## فهرست مطالب

صفحه	عنوان
۷	فصل اول: کلیات تحقیق .....
۱۰	۱-۱ مقدمه .....
۱۱	۲-۱ نظریه داروین .....
۱۲	۳-۱ نظریه تکاملی لامارک .....
۱۲	۴-۱ مفاهیم پایه .....
۱۳	۵-۱ الگوریتمهای تکاملی .....
۱۵	۱-۵-۱ الگوریتمهای ابتکاری و فراابتکاری .....
۱۵	الف) یک راه حل گنگ و غیرواضح .....
۱۵	ب) یک راه حل کمترگنگ .....
۱۶	۶-۱ مراحل یک الگوریتم تکاملی .....
۱۸	۷-۱ روشهای نمایش کروموزوم .....
۱۹	۸-۱ جمعیت اولیه .....
۲۱	۹-۱ تابع برازش شایستگی .....
۲۲	۱۰-۱ عملگرانتخاب .....
۲۲	۱-۱۰-۱ انتخاب تصادفی .....
۲۳	۲-۱۰-۱ انتخاب نسبی .....
۲۴	۳-۱۰-۱ انتخاب رتبه ای .....
۲۵	۴-۱۰-۱ انتخاب مسابقه ای .....

۲۵	.....	انتخاب برشی ۵-۱۰-۱
۲۵	.....	نخبه گرایی ۶-۱۰-۱
۲۶	.....	تالارمشاهیر ۷-۱۰-۱
۲۶	.....	عملگرهای تولیدمثل ۱۱-۱
۲۷	.....	باز ترکیب وجهش درنمایش دودویی ۱-۱۱-۱
۲۸	.....	باز ترکیب وجهش درنمایش اعدادحقیقی ۲-۱۱-۱
۳۱	.....	باز ترکیب وجهش درنمایش جایگشت عناصر ۳-۱۱-۱
۳۴	.....	مرحله جایگزینی ۱۲-۱
۳۴	.....	جایگزینی حالت پایدار ۱-۱۲-۱
۳۴	.....	جایگزینی نسلی ۲-۱۲-۱
۳۴	.....	شرایط توقف ۱۳-۱
۳۵	.....	محاسبات تکاملی درمقابل بابینه سازی کلاسیک ۱۴-۱
۳۷	.....	<b>فصل دوم: الگوریتم ژنتیک</b>
۴۰	.....	۱-۲ مقدمه
۴۱	.....	۲-۲ مروری سریع برالگوریتم
۴۳	.....	۳-۲ ساختار کلی الگوریتم ژنتیک
۴۵	.....	۴-۲ روشهای نمایش
۴۶	.....	۵-۲ نقاط قوت الگوریتمهای ژنتیک
۴۸	.....	۶-۲ محدودیتهای الگوریتمهای ژنتیک
۴۹	.....	۷-۲ الگوریتم ژنتیک استاندارد
۴۹	.....	۸-۲ مباحث پیشرفته
۵۰	.....	۱-۸-۲ عملگر باز ترکیب
۵۵	.....	۲-۸-۲ عملگر جهش
۵۶	.....	۹-۲ شبیه سازی مسئله فروشنده دوره گردبا استفاده از الگوریتم ژنتیک ...
۵۶	.....	۱-۹-۲ معرفی مسئله فروشنده دوره گرد

۶۳	..... فصل سوم : محاسبات زیستی
۶۶	..... ۱-۳ هوش جمعی
۶۸	..... ۲-۳ الهام از رفتارهای گروهی موجودات
۷۱	..... ۱-۲-۳ روشهای علامت-محور
۷۲	..... ۲-۲-۳ روشهای تقلید-محور
۷۲	..... ۳-۳ الگوریتمهای علامت-محور
۷۳	..... ۴-۳ الگوریتم مورچگان
۷۳	..... ۱-۴-۳ مقدمه
۷۳	..... ۲-۴-۳ تاریخچه الگوریتم مورچگان
۷۴	..... ۳-۴-۳ الگوریتم بهینه سازی کلونی مورچه ها
۷۸	..... ۴-۴-۳ چگونگی انتخاب کوتاهترین مسیر توسط مورچه ها
۸۱	..... ۵-۴-۳ مزیت‌های الگوریتم بهینه سازی کلونی مورچگان
۸۱	..... ۶-۴-۳ کاربردهای الگوریتم بهینه سازی کلونی مورچگان
۸۲	..... ۷-۴-۳ بهینه‌سازی مسائل به روش الگوریتم بهینه سازی کلونی مورچگان ...
۸۳	..... ۸-۴-۳ شبیه سازی مسئله فروشنده دوره گرد با استفاده از الگوریتم بهینه سازی کلونی مورچه
۸۸	..... ۹-۴-۳ استفاده از الگوی کلونی مورچه‌ها جهت اداره ترافیک بین شهری ...
۸۸	..... ۵-۳ الگوریتم‌های تقلید
۸۹	..... ۶-۳ الگوریتم بهینه سازی ازدحام ذرات
۸۹	..... ۱-۶-۳ مقدمه
۸۹	..... ۲-۶-۳ تشریح الگوریتم بهینه سازی ازدحام ذرات
۹۲	..... ۱-۲-۶-۳ ساختار الگوریتم بهینه سازی ازدحام ذرات
۹۲	..... ۲-۲-۶-۳ پارامترهای الگوریتم بهینه سازی ازدحام ذرات
۹۳	..... ۳-۲-۶-۳ مزایای الگوریتم بهینه سازی ازدحام ذرات
۹۵	..... ۴-۲-۶-۳ کاربردهای الگوریتم بهینه سازی ذرات
۹۶	..... ۵-۲-۶-۳ مقایسه الگوریتم بهینه سازی ازدحام ذرات با دیگر الگوریتمهای تکاملی

۳-۶-۳	شبیه سازی مسئله فروشنده دوره گرد با استفاده از الگوریتم بهینه سازی
۹۶	ازدحام ذرات .....
۳-۷-۱۰۰	الگوریتم رقابت استعماری .....
۳-۷-۱۰۰	مقدمه .....
۳-۷-۱۰۱	تشریح الگوریتم رقابت استعماری .....
۳-۷-۱۰۸	مزایای الگوریتم رقابت استعماری .....
۳-۸-۱۰۸	الگوریتم ترکیبی جهش قورباغه .....
۳-۸-۱۰۸	الگوریتم رفتاری .....
۳-۸-۱۰۹	فرآیند بهینه سازی در الگوریتم جهش ترکیبی قورباغه .....
۳-۸-۱۰۹	جزئیات الگوریتم جهش ترکیبی قورباغه .....
۳-۸-۱۱۱	اکتشاف سراسری .....
۳-۸-۱۱۳	شبه کدمربوط به الگوریتم جهش ترکیبی قورباغه .....
۱۱۵	<b>فصل چهارم: محاسبات غیرزیستی .....</b>
۴-۱-۱۱۸	مقدمه .....
۴-۲-۱۱۸	الگوریتم شبیه سازی تبریدی .....
۴-۳-۱۲۱	بهینه سازی محلی و الگوریتم آنیل .....
۴-۴-۱۲۵	اجرای عملیات تبرید .....
۴-۵-۱۲۵	کاربردهای شبیه سازی تبریدی شده .....
۴-۶-۱۲۶	پارامترهای موثر در کیفیت الگوریتم شبیه سازی .....
۴-۷-۱۲۷	مزایا و معایب روش شبیه سازی تبریدی شده عبارتند از .....
۴-۸-۱۲۷	شبیه سازی مسئله فروشنده دوره گرد با استفاده از الگوریتم بهینه سازی
۴-۸-۱۲۷	شبیه سازی تبریدی .....
۴-۹-۱۳۱	الگوریتم جستجوی گرانشی .....

# فصل اول

## الگوریتم‌های تکاملی





## الگوریتم‌های تکاملی

الگوریتم‌های تکاملی به آن دسته از روش‌های جستجوی فرامکاشفه‌ای گفته می‌شود که مبتنی بر قانون بقاء اصلح نظریه تکاملی داروین هستند. از نقطه نظر طبقه بندی روش‌های جستجوی فرا ابتکاری، این الگوریتم‌ها جمعیتی (البته معمولاً جمعیتی هستند)، زیستی، تکاملی، با حافظه و احتمالی می‌باشند. دلیل اینکه در این کتاب الگوریتم‌های تکاملی را با یک عنوان مستقل و به صورت مجزا از سایر فرامکاشفه‌های زیستی مورد توجه قرار داده ایم آن است که این الگوریتم‌ها به صورتی مستقیم از فرامکاشفه بقاء اصلح که مبتنی بر نظریه داروین است استفاده می‌کنند. این در حالی است که سایر روش‌های جستجوی زیستی به صورتی غیر مستقیم مبتنی بر فرامکاشفه بقاء اصلح داروین هستند. به عبارت دیگر می‌توان نظریه تکاملی داروین را ریشه همه روش‌های فرامکاشفه‌ای زیستی دانست. به دلیل همین اهمیت و جامعیت روش‌های تکاملی، در این بخش به صورت مستقل و ویژه این الگوریتم‌ها را مورد کنکاش قرار خواهیم داد. در این فصل مفاهیم کلیدی و بنیادی درباره تکامل و پردازش تکاملی، که در اکثر فرامکاشفه‌های تکاملی مشترک هستند، عنوان خواهند شد.

## پردازش تکاملی

### ۱-۱ مقدمه

دنیایی که ما در آن زندگی می‌کنیم دائماً در حال تغییر است. برای اینکه موجودات بتوانند در این محیط متغیر و پویا بقاء خود را تضمین کنند، باید توانایی تطبیق پذیری با محیط را داشته باشند. تکامل، فرآیند اصلی این تطبیق پذیری است. البته با هدف بهبود توانایی‌های بقاء از طریق انتخاب طبیعی، بقاء اصلح، تولید مثل، رقابت و همزیستی.

در این فصل پردازش تکاملی را به عنوان شاخه‌ای از محاسبات هوشمند، که فرآیند انتخاب طبیعی را مدل می‌کند، توضیح خواهیم داد. در ادامه، ابتدا نظریه تکاملی داروین را تشریح نموده و سپس مفاهیم پایه‌ای را در حوزه پردازش تکاملی ارائه خواهیم کرد. پس از آن به توصیف مراحل یک الگوریتم تکاملی خواهیم پرداخت. بخش‌های بعد، به تشریح مباحث مربوط به یک الگوریتم تکاملی می‌پردازند. این مباحث عبارتند از: روش‌های نمایش کروموزوم، جمعیت اولیه، تابع برازش، عملگر انتخاب، عملگرهای تولید مثل، نخبه‌گرایی و شرایط توقف. توجه کنید که مطالب این فصل در پیچه‌ای را برای درک شاخه مهم پردازش تکاملی ایجاد می‌کنند. به عبارت دیگر، این مطالب زمینه را برای درک بهتر الگوریتم‌های متنوع مبتنی بر پردازش تکاملی، که در فصل دوم ارائه خواهند شد، مهیا می‌سازند.

طبق تعریف، تکامل، فرآیندی است برای بهینه‌سازی که هدف آن بهبود توانایی یک سیستم برای بقا در محیط‌های متغیر پویا و رقابتی. تکامل مفهومی است که در طول قرن‌ها

به شدت مورد بحث بوده است، و هنوز هم در خصوص آن بحث‌های فراوانی وجود دارد. هنگام صحبت در مورد تکامل، مهم است که نخست حوزه ای را که در آن تکامل می‌تواند تعریف شود شناسایی کنیم، به عنوان مثال، سیستم‌های تکامل کیهانی، شیمیایی، ستاره ای و سیاره ای، آلی یا بشرساخت. برای هر یک از این حوزه‌های متفاوت، تکامل ممکن است به صورتی متفاوت نیز تفسیر شود. در این بخش از کتاب، تمرکز بر تکامل زیست شناختی است. حتی برای این حوزه خاص نیز، تلاش‌ها برای تعریف اصطلاح تکامل زیست شناختی هنوز هم بحث‌های متعددی را ایجاد می‌کنند، که دیدگاه‌های لامارکی و داروینی محبوب‌ترین و مقبول‌ترین آن‌ها هستند. هر چند داروین (۱۸۰۹-۱۸۸۲) عموماً به عنوان بنیانگذار هر دوی نظریه تکامل و اصول نسبی مشترک در نظر گرفته می‌شود، لامارک (۱۷۴۴-۱۸۲۹) احتمالاً اولین کسی بود که در مورد تکامل زیست شناختی نظریه پردازی کرد.

## ۱-۲ نظریه داروین

تکامل یک فرآیند بهینه سازی است، که هدف آن بهبود توانایی یک ارگانیسم (یا سیستم) برای نجات در یک محیط متغیر و پویا می‌باشد. تکامل، مفهومی است که قرن هاست مورد توجه قرار گرفته است. زمانی که درباره تکامل صحبت می‌کنیم، ابتدا باید محیطی را که تکامل در آن تعریف می‌شود، شناسایی کنیم. به عنوان مثال کیهانی، شیمیایی، اختری، نجومی و یا حتی سیستم‌های ساخته شده دست بشر. نظریه تکامل داروین را می‌توان بدین شرح خلاصه کرد: در دنیایی با منابع محدود و جوامع پایدار، هر فرد با دیگران برای بقا رقابت می‌کند. افرادی با "بهترین" ویژگی‌ها (صفات) احتمال بیشتری برای بقا و تولید مثل دارند، و آن ویژگی‌ها به فرزندانشان منتقل خواهند شد. این ویژگی‌های مطلوب توسط نسل‌های بعد به ارث برده می‌شوند، و (در طول زمان) در میان جمعیت، غالب می‌شوند.

بخش دوم نظریه داروین می‌گوید که، در حین تولید ارگانیسم یک کودک، رویدادهای تصادفی باعث ایجاد تغییرات تصادفی در ویژگی‌های ارگانیسم کودک می‌شوند. اگر این

ویژگی های جدید به نفع ارگانسیم باشند، آنگاه شانس های بقا برای آن ارگانسیم افزایش می یابند.

ممکن است که برای این محیط های متفاوت، تکامل به صورت های مختلفی تعبیر شود. در این کتاب تمرکز ما بر روی تکامل زیست شناختی است. در این حوزه خاص نیز در تعریف تکامل زیست شناختی دیدگاه های مختلفی وجود دارد که دیدگاه لامارکی و داروینی نسبت به بقیه محبوب تر هستند.

### ۱-۳ نظریه تکاملی لامارک

این نظریه، نظریه وراثت بود، یعنی توارث صفات اکتسابی. ایده اصلی این است که افراد در طول عمر خود تطبیق می یابند، و صفات خود را به فرزندانشان انتقال می دهند. سپس فرزندان به تطبیق یافتن ادامه می دهند. مطابق با لامارکیسم، روش تطبیق مبتنی بر مفهوم استفاده و عدم استفاده است: در طول زمان، افراد صفاتی را که به آن نیاز ندارند از دست می دهند، و صفاتی را که مفید هستند با انجام آن ها پرورش می دهند.

### ۱-۴ مفاهیم پایه

در ذیل برخی از مفاهیم مهمی که در الگوریتم های تکاملی به وفور از آنها یاد می شود تشریح شده اند:

♦ **کروموزم:** اطلاعات ژنی یک موجود در کروموزوم ذخیره می شود. هر کروموزوم از DNA تشکیل شده است. در یک الگوریتم تکاملی منظور از کروموزوم، یک رشته، گراف، درخت و یا دنباله صفر و یک است که معمولاً، معادل یک پاسخ برای مساله مورد بررسی می باشد.

♦ **ژن:** کروموزوم ها به چندین قسمت تقسیم بندی می شوند که ژن نام دارند. ژن ها خصوصیات گونه ها یا همان افراد را تشکیل می دهند. در یک الگوریتم تکاملی، هر پاسخ از چندین بخش تشکیل شده است که به هر کدام از آن بخش ها، یک

مشخصه یا ویژگی گفته می‌شود. مشخصه یا ویژگی در هر پاسخ، معادل ژن در کروموزوم است.

♦ **آلل:** هر ژن مجموعه ای از مقادیر مجاز را می‌تواند اختیار کند. به هر کدام از این مقادیر یک آلل گفته می‌شود. به عنوان مثال یک ژن مربوط به رنگ چشم است و آلل‌های ممکن برای آن عبارتند از: سیاه، خاکستری، قهوه ای، آبی، سبز و عسلی (و نه سفید). در یک الگوریتم تکاملی، مقادیر مجاز برای مشخصه‌های هر پاسخ معادل مفهوم آلل است.

♦ **ژنوتایپ:** ترکیب کامل تمامی ژن‌ها برای یک فرد مشخص، ژنوتایپ (ژنوتیپ) نامیده می‌شود. دنباله ژنی مربوط به هر پاسخ در یک الگوریتم تکاملی ژنوتایپ آن پاسخ را نشان می‌دهد.

♦ **فنوتایپ:** خصوصیات ظاهری یک شخص که از رمزگشایی یک ژنوتایپ حاصل می‌شود، فنوتایپ (فنوتیپ) نامیده می‌شود، به عبارت دیگر، تجلی معنایی ژنوتایپ مربوط به یک پاسخ در دنیای واقعی، گویای فنوتایپ آن پاسخ است.

♦ **برازش:** شایستگی یک موجود در یک جمعیت، برازش آن موجود نامیده می‌شود. برازش هر پاسخ در جمعیت، با توجه به شایستگی فنوتایپ آن پاسخ تعیین می‌شود. در یک الگوریتم تکاملی، با استفاده از یک یا چند تابع، برازش هر پاسخ محاسبه می‌شود. تابع برازش، ارتباط یک الگوریتم تکاملی با مساله مورد تحلیل می‌باشد.

## ۱-۵ الگوریتم‌های تکاملی

الگوریتم‌های تکاملی زیر مجموعه الگوریتم‌های بهینه سازی میباشند. در این دسته از الگوریتمها، منظور از تکامل یعنی تغییر ویژگی‌های وراثتی افراد در جهت بهبود نسلها. روند کلی الگوریتم‌های تکاملی را میتوان به صورت زیر در نظر گرفت.

**جمعیت:** در تمامی الگوریتم‌های تکاملی، هدف تولید جوابهای بهینه میباشد. این

الگوریتمها، ابتدا مجموعه ای از جوابها را به صورت تصادفی تولید میکنند. تعداد مجموعه جمعیت اولیه از ابتدا مشخص میباشد و تا انتهای اجرای الگوریتم در نسلهای مختلف، تغییری در تعداد جمعیت اولیه رخ نمیدهد. جوابهای اولیه بهتر است از تنوع بالاتری برخوردار باشند.

**والدین:** برای تولید نسل بعدی تعدادی از فرزندان را به عنوان والد انتخاب میکنیم. این انتخاب براساس میزان شایستگی افراد میباشد. هرچند که اگر انتخاب صرفا براساس شایستگی باشد این مشکل وجود دارد که همیشه فرزندان با والدین شایسته تر، شایسته نمیشوند. پس باید درصدی از شانس را نیز در آن دخیل دانست یعنی شایستگی و شانس را همزمان در نظر گرفت بطوریکه براساس میزان شایستگی افراد، شانس را نیز به آنها برای انتخاب شدن به عنوان والد داد. در این حالت شاید کسی که شایستگی بیشتری دارد انتخاب نگردد هرچند که فردی با شایستگی بیشتر، شانس بیشتری نیز برای انتخاب شدن دارد.

اما موضوع بعدی در انتخاب والدین این است که چند درصد از جمعیت اولیه را به عنوان والد باید انتخاب نمود؟ جواب این است که هرچه تعداد والد بیشتری انتخاب گردد پس فرزندان بیشتری هم تولید میگردند و تغییرات بیشتر میشود. اما این درصد انتخاب خود براساس سعی و خطا میباشد و مشخص نیست که چه میزان تغییرات بهتر است چرا که در الگوریتمهای تکاملی اعتقاد براین است که بهبود، به تدریج بدست می آید.

**فرزندان:** برای تولید فرزندان از دو تکنیک "ترکیب مجدد" و "جهش" استفاده میگردد. در حالت اول، والدین ترکیب شده و فرزندان جدید تولید میگردند. در این حالت معمولا تعداد فرزندان به تعداد والدین بستگی دارد مثلا دو تا والد دو فرزند نیز تولید میکنند. در این حالت میزان تغییرات، اساسی میباشد. در حالت دوم که به آن جهش هم میگویند تغییراتی جزئی میباشد بطوری که ابتدا ترکیب مجدد رخ داده سپس روی فرزندان تغییراتی جزئی اعمال میگردد.

## ۱-۵-۱ الگوریتمهای ابتکاری و فراابتکاری

عموما برای حل مسائل بهینه سازی دو رویکرد اصلی زیر وجود دارد:

### الف) یک راه حل گنگ و غیر واضح

این رویکرد مبتنی بر سعی و خطای مطلق میباشد بطوریکه نقش شانس در آن بسیار پررنگ میباشد. چرا که ممکن است اولین راه حل ، بهترین باشد و یا حتی یکصدمین راه حل نیز بهترین راه حل ممکن نباشد. مانند مسئله یافتن یک کلمه عبور چهاررقمی. ساختار کلی این راه حلها که به آنها روشهای حدسی، اکتشافی و یا ابتکاری نیز میگویند به صورت زیر میباشد:

#### *A Dumb Solution*

#### *Repeat*

*Generate a random solution.*

*Evaluate the solution to see how good it is.*

*Until the solution is good enough.*

این روشها در موارد زیر کاربرد ندارند:

در مواردی که دامنه راه حلها ممکن آنقدر زیاد است که با روشهای تصادفی نتوان به نتیجه رسید.

در مواردی که باید وضعیت راه حلها ممکن را در کنار یکدیگر در نظر گرفت.

### ب) یک راه حل کمتر گنگ

این نوع روشها نیز هرچند بهتر از روش قبلی هستند اما چون بازهم جوابهای تصادفی تولید میکنند پس کمی ابهام نیز دارند و به نوعی غیرقطعی نیز میباشند. البته امکان دارد که

به بهترین راه حل نیز دست نیابند. این نوع روشها را فراابتکاری و یا فرااکتشافی گویند. صورت کلی این روشها به صورت زیر میباشند:

### ***A Less Dumb Solution***

*Generate a set of random solution.*

*Repeat*

*Evaluate each solution.*

*Duplicate some good solutions.*

*Make Small changes on Solutions.*

*Remove some bad solutions.*

*Until the best solution is found.*

یکی از دلایل برتری این روش نسبت به روش قبلی انعطاف پذیری بالای آن میباشد چرا که با جوابها خیلی بازی میکند. این روش از هوشمندی بالاتری برخوردار بوده و در کاربردهای بهینه سازی امروزی نیز واقعی تر به نظر میرسد.

## **۶-۱ مراحل یک الگوریتم تکاملی**

فرآیند تکامل، از طریق انتخاب طبیعی موجودات در یک جمعیت، را می توان به صورت یک جستجو در فضای مقادیر ممکن کروموزومها در نظر گرفت. در این صورت یک الگوریتم تکاملی، جستجویی هوشمندانه و فرامکاشفه ایست، برای یافتن یک راه حل به اندازه کافی خوب برای مساله داده شده، مراحل یک الگوریتم تکاملی عبارتند از:

**مرحله ۱ - تولید جمعیت اولیه:** در این مرحله، یک جمعیت از کروموزومها یا همان پاسخهای مساله تولید می شود. بدیهی است که قبل از تولید جمعیت اولیه بایستی نحوه نمایش کروموزومها، که گویای یک پاسخ برای مساله مورد بررسی است، تعیین شود. خروجی مرحله اول، یک جمعیت از پاسخ هاست.



**مرحله ۲- محاسبه برازش جمعیت ورودی:** در این مرحله، برازش تک تک کروموزوم‌های جمعیت تولید شده با توجه به تابع برازش تعیین شده، محاسبه می‌شود. خروجی این مرحله، یک جمعیت از پاسخ‌های ارزیابی شده می‌باشد.

**مرحله ۳- انتخاب برای تولید مثل:** در این مرحله، آن دسته از اعضاء جمعیت ورودی که برازش بالاتری نسبت به سایر اعضاء دارند، برای تبیین قانون بقاء اصلح داروین، انتخاب می‌شوند. خروجی این مرحله، جمعیتی از والدین برازنده است.

**مرحله ۴- باز ترکیب والدین انتخاب شده:** در این مرحله با توجه به جمعیت والدین برازنده و با استفاده از عملگر بازترکیب، جمعیتی از فرزندان تولید می‌شوند. اعمال عملگر بازترکیب، با توجه به یک پارامتر با نام احتمال بازترکیب ( $P_c$ ) انجام می‌شود. خروجی این مرحله جمعیتی از فرزندان تولید شده می‌باشد.

**مرحله ۵- جهش فرزندان تولید شده:** در این مرحله، فرزندان جدید مرحله قبل تحت عملگر جهش قرار می‌گیرند. البته عملگر جهش با یک احتمال بر روی دنباله ژنی فرزندان رخ می‌دهد. این احتمال با نام احتمال جهش ( $P_m$ ) شناخته می‌شود. خروجی این مرحله جمعیتی از فرزندان جهش یافته است.

**مرحله ۶- محاسبه برازش جمعیت فرزندان:** در این مرحله، شایستگی فرزندان جهش یافته، با استفاده از تابع برازش، محاسبه می‌شود. خروجی این مرحله، جمعیت فرزندان ارزیابی شده است.

**مرحله ۷- انتخاب برای جایگزینی:** در این مرحله، با توجه به جمعیت والدین (ورودی مرحله سوم) و جمعیت فرزندان ارزیابی شده (خروجی مرحله ششم) یک جمعیت جدید برای نسل بعد (هر تکرار در این الگوریتم معادل مرحله ششم) یک جمعیت جدید برای نسل بعد (هر تکرار در این الگوریتم معادل یک نسل است) تولید می‌شود. بدیهی است که خروجی این مرحله جمعیتی است که در آن بخشی از والدین نسل قبل و تعدادی از فرزندان جدید تولید شده نسل جاری وجود دارند.

**مرحله ۸- بررسی شرط توقف:** در این مرحله، در مورد ادامه فرآیند تکاملی الگوریتم

تصمیم گیری می شود. در صورت عدم ارضاء شرط توقف، فرآیند تکاملی الگوریتم با رجوع به مرحله ۳ ادامه می یابد. در غیر این صورت الگوریتم متوقف شده و بهترین پاسخ در آخرین نسل به عنوان حاصل جستجوی تکاملی در خروجی ارائه می شود.

### ۷-۱ روش های نمایش کروموزوم

همان گونه که ذکر شد، هر موجود در الگوریتم، بیانگر یک راه حل کاندید برای مساله بهینه سازی است. ویژگی های موجودات، توسط کروموزوم آنها نشان داده می شود. این ویژگی ها مربوط به متغیرها در مسائل بهینه سازی می باشند. هر متغیری که نیاز به بهینه شدن دارد به عنوان یک ژن مشخص می شود. یک ژن، کوچک ترین واحد اطلاعاتی است. یکی از مهم ترین مراحل در طراحی یک الگوریتم تکاملی، یافتن روش نمایش مناسب برای راه حل های کاندید می باشد (کروموزوم). کارآیی و پیچیدگی الگوریتم جستجو به روش نمایش کروموزوم بسیار وابسته است. الگوریتم های تکاملی مختلف از نمایش های گوناگونی استفاده می کنند. در بیشتر الگوریتم های تکاملی، راه حل ها به صورت برداری از یک نوع داده خاص نمایش داده می شوند، (برنامه نویسی ژنتیک یک استثنا می باشد زیرا که موجودات در آن به شکل درخت نمایش داده می شوند).

مدل نمایش سنتی برای الگوریتم های تکاملی، بردار دودویی با طول ثابت است. با داشتن فضای جستجوی  $n_x$ -بعدی، هر موجود دارای  $n_x$  متغیر می باشد که هر متغیر به صورت رشته بیتی کد شده است. برای متغیرهایی با مقادیر اسمی، هر مقدار اسمی را می توان به صورت بردار بیتی  $n$ -بعدی کد نمود به صورتی که  $2^n$  تعداد کل مقادیر اسمی گسسته برای آن متغیر است. (شکل ۱-۱ الف))

برای حل مسائل بهینه سازی با متغیرهای پیوسته، روش نمایش دیگری لازم است که نمایش مبتنی بر اعداد حقیقی نامیده می شود. در این روش هر بیت در کروموزوم های جمعیت تکاملی با یک عدد حقیقی مقدار دهی می شود. (شکل ۱-۱ ب))

۱	۰	۰	۰	۱	۱	۱	۰
---	---	---	---	---	---	---	---

(الف)

۲۵,۶۱	۰,۱۲	۲۳,۹۸	۱۵,۱	۶,۲۷	۲۰,۰۰	۱۰,۸۵	۱۲,۳۵
-------	------	-------	------	------	-------	-------	-------

(ب)

۱	۷	۹	۸	۲	۶	۵	۳
---	---	---	---	---	---	---	---

(ج)

شکل ۱-۱ روش‌های نمایش در الگوریتم‌های تکاملی. (الف) نمایش دودویی، (ب) نمایش اعداد حقیقی؛ (ج) نمایش جایگشت عناصر.

روش دیگر نمایش کروموزوم‌ها، روش جایگشت عناصر است. هدف در این روش، یافتن بهترین دنباله‌ای از عناصر می‌باشد بطوریکه برای هر دنباله، برازشی مشخص قابل محاسبه است. مساله فروشنده دوره گرد بهترین مثال برای استفاده از چنین روش نمایشی می‌باشد (شکل ۱-۱ ج)). همانطور که مشخص است در این روش مقادیر هیچ دو ژنی در هر کدام از والدین نبایستی تکراری باشد.

نمایش درختی، روش نمایش مشهور دیگری است که بایستی به آن اشاره نمایم. در این روش هر کدام از اعضاء جمعیت در الگوریتم تکاملی به شکل یک درخت نمایش داده می‌شوند. از این روش معمولاً برای تکامل برنامه‌ها و یا روابط ریاضی بهره گرفته می‌شود. در برنامه نویسی ژنتیک از این روش نمایش استفاده می‌شود.

## ۸-۱ جمعیت اولیه

الگوریتم‌های تکاملی مبتنی بر جمعیت هستند. هر الگوریتم تکاملی، جمعیتی از راه حل‌های کاندید را نگهداری می‌کند. اولین مرحله در الگوریتم برای حل مسائل بهینه‌سازی، تولید یک جمعیت اولیه است. روش استاندارد برای تولید جمعیت اولیه، نسبت دادن

مقدار تصادفی از دامنه مجاز، به هر یک از ژن های هر کدام از کروموزوم است. این روش با نام مقدار دهی اولیه تصادفی شناخته می شود. هدف از مقداردهی تصادفی این است که جمعیت اولیه را به صورت یکنواخت از کل فضای جستجو بدست آورده باشیم، در صورتی که ناحیه ای از فضای جستجو وجود دارد. روش دیگر مقداردهی جمعیت اولیه در الگوریتم های تکاملی، روش مقداردهی اولیه هوشمندانه است. در این روش، براساس اطلاعات موجود از فضای جستجوی مساله، کروموزوم های اولیه به گونه ای تولید می شوند که برانزنگی آنها در بدو تولد بالا باشد. به عبارت دیگر، در این روش سعی بر آن است که از آغاز فرآیند جستجوی الگوریتم تکاملی بر روی بخش های مهم و ارزشمند فضای جستجو تمرکز نموده و شانس یافتن پاسخ های به اندازه کافی خوب را، افزایش دهیم.

اندازه جمعیت اولیه یکی از پارامترهای مهم الگوریتم های تکاملی است. بایستی توجه شود که این اندازه ممکن است ثابت و یا متغیر باشد. به عنوان مثال در الگوریتم های ژنتیک این اندازه ثابت است در حالی که مثلاً در یک سیستم ایمنی مصنوعی اندازه مزبور پویا است.

جمعیت اولیه، در میزان پیچیدگی محاسباتی و قابلیت های پوشش و ارتفاع الگوریتم های تکاملی اثرگذار است. افزایش این پارامتر سبب تقویت هر دو قابلیت پوشش و ارتفاع می شود. به عبارت دیگر هر چقدر که یک الگوریتم تکاملی توان پوششی بالاتری داشته باشد، با افزایش اندازه جمعیتش، آن توان پوششی تقویت می شود. به همین ترتیب هر چقدر که الگوریتم توان ارتفاعی بیشتری نیز داشته باشد، با افزایش اندازه جمعیتش توان مزبور تقویت خواهد شد. دلیل این مدعا آن است که با افزایش اندازه جمعیت، شانس انجام عملگرهای تولید مثل که معمولاً ترکیبی از دو قابلیت پوشش و ارتفاع را در خود دارند، افزایش می یابد. در هر حال، هر چقدر که تعداد موجودات بیشتر شود، میزان پیچیدگی محاسباتی هر نسل افزایش می یابد. از طرف دیگر، هنگامی که جمعیت تکاملی کوچک باشد، قسمت کوچکی از فضای جستجو توسط الگوریتم کنکاش می شود. در این حالت، با وجود اینکه پیچیدگی زمانی هر نسل پایین است، الگوریتم تکاملی معمولاً نیاز

به نسل‌های بیشتری برای رسیدن به همگرایی دارد.

## ۹-۱ تابع برازش شایستگی

در مدل تکاملی داروینی، موجوداتی با ویژگی‌های برتر شانس بیشتری برای بقا و تولید مثل دارند. به منظور تشخیص توانایی بقا هر کدام از پاسخ‌ها در الگوریتم‌های تکاملی، یک تابع برازش مورد استفاده قرار می‌گیرد. این تابع میزان شایستگی راه حل نمایش داده شده توسط کروموزوم را مشخص می‌نماید. تابع برازش  $f$ ، میزان شایستگی هر کروموزوم را به یک مقدار عددی نگاشت می‌کند.

معمولاً، تابع برازش یک معیار مطلق برای اندازه‌گیری شایستگی است و راه حل نمایش داده شده توسط کروموزوم، مستقیماً توسط تابع هدف مورد ارزیابی قرار می‌گیرد. برای برخی از کاربردها، برای مثال یادگیری قوانین دسته‌بندی در حوزه داده کاوی، امکان یافتن یک تابع برازش قطعی وجود ندارد. در عوض، یک معیار برازش نسبی برای سنجش کارایی موجود در ارتباط با سایر موجودات در جمعیت استفاده می‌شود.

هر کدام از مسائل بهینه‌سازی، تاثیری متفاوت در فرموله کردن تابع برازش دارند. تابع برازش از اهمیت ویژه‌ای برخوردار می‌باشد. مهمترین کاربرد تابع برازش در عملگر انتخاب است که در آن برازنده‌ترین اعضاء در جمعیت الگوریتم تکاملی برای ارسال به مرحله تولید مثل گزینش می‌شوند.

نکته مهمی که هرگز نباید در مورد تابع برازش فراموش کرد، آن است که این تابع، ارزش تجلی ژنوتایپ اعضاء را در دنیای مربوط به مساله مورد بررسی نشان می‌دهد. در بخش ۱-۳، این مفهوم را با نام فنوتایپ اعضاء شناختیم. منظور از برازش فنوتایپی موجود، ارزش پاسخ منتسب به آن موجود در دنیای مربوط به مساله مورد بررسی است. بنابراین می‌توان گفت که تنها رابطه یک الگوریتم تکاملی (و در واقع هر الگوریتم فرامکاشفه‌ای) با مساله بهینه‌سازی مورد بررسی، تابع برازش است. بجز تابع برازش، بقیه بخش‌های الگوریتم برای بسیاری از مسائل ممکن است مشابه باشد!

## ۱-۱۰ عملگر انتخاب

انتخاب یکی از عملگرهای اصلی در الگوریتم‌های تکاملی است و مستقیماً به مفهوم بقا اصلح در نظریه داروین مربوط می‌شود. هدف اصلی در عملگر انتخاب، یافتن راه حل‌های برتر است. در این راستا پاسخ‌های برازنده و برتر (اصلح) در جمعیت نسل جاری برای اعمال عملگرهای تولید مثل انتخاب می‌شوند.

عملگرهای انتخاب زیادی تاکنون در کتب و مقالات مرتبط ارائه شده‌اند. در این بخش برخی از مهمترین عملگرهای انتخاب مورد بحث و بررسی قرار خواهند گرفت. اما قبل از پرداختن به انواع مختلف عملگرهای انتخاب، لازم است که توضیحی درباره مفهوم فشار انتخاب بدهیم.

فشار انتخاب عبارت است از: سرعت پرشدن جمعیت از راه حل‌های خوب توسط بکارگیری عملگر انتخاب. عملگرهایی با فشار انتخاب زیاد، تنوع را در جمعیت به نسبت عملگرهایی با فشار انتخاب پایین به سرعت کاهش می‌دهند؛ که منجر به همگرایی زودرس الگوریتم به راه حل‌هایی با بهینگی محلی می‌شود. فشار انتخاب زیاد (توجه بیش از اندازه به اعضاء برازنده در جمعیت)، قابلیت پویا را در الگوریتم‌های تکاملی محدود نموده و در عوض قابلیت انتفاع را در این الگوریتم‌ها تقویت می‌سازد. انواع مختلف عملگرهای انتخاب که در این بخش مورد بحث قرار می‌گیرند عبارتند از: انتخاب تصادفی، انتخاب نسبی، انتخاب رتبه ای، انتخاب مسابقه ای و انتخاب برشی.

### ۱-۱۰-۱ انتخاب تصادفی

انتخاب تصادفی، یکی از ساده ترین عملگرهای انتخاب است. در این روش انتخاب، هر موجود دارای احتمال یکسان  $\frac{1}{n_s}$  (اندازه جمعیت است) می‌باشد. در انتخاب تصادفی، از هیچ اطلاعات برآزشی استفاده نمی‌شود. به این معنی که بهترین و بدترین موجودات دارای احتمال کاملاً یکسان برای قرار گرفتن در نسل بعدی هستند. انتخاب تصادفی دارای کمترین فشار انتخاب در میان عملگرهای انتخاب که در این بخش توضیح می‌دهیم می‌باشد.

## ۲-۱۰-۱ انتخاب نسبی

انتخاب نسبی توسط هالند پیشنهاد شده است، که در آن شانس انتخاب موجودات برتر، بیشتر است. در این روش انتخاب، یک توزیع احتمالی متناسب با شایستگی هر پاسخ ایجاد می‌شود و موجودات از طریق نمونه برداری از توزیع مزبور انتخاب می‌شوند. به رابطه (۱-۱) توجه کنید.

$$\varphi_s(x_i) = \frac{f_Y(x_i)}{\sum_{l=1}^{n_s} f_Y(x_l)} \quad (1-1)$$

در این رابطه،  $n_s$  تعداد کل موجودات در جمعیت،  $\varphi_i(x_i)$  احتمال انتخاب عضو در جمعیت توسط روش انتخاب نسبی و  $f_Y(x_i)$  برازش عضو  $x_i$  است. برای مسائل کمینه سازی، بایستی برازش عضو  $x_i$  را با استفاده از یک تابع به قسمی تبدیل کنیم که معنای پیشینه سازی در آن حاصل شود (مثلاً  $g_Y(x_i) = \frac{1}{f_Y(x_i)}$ ).

به دلیل نحوه پیاده سازی الگوریتم انتخاب نسبی، از این روش در اغلب موارد با نام انتخاب چرخ رولت نیز یاد می‌شود. در این روش، توزیع احتمال را به عنوان چرخ رولت در نظر می‌گیریم. به صورتی که اندازه هر قطعه متناسب با احتمال نرمال شده انتخاب هر موجود باشد. برای پیاده سازی این روش، ابتدا یک دایره (چرخ دوار) در نظر گرفته می‌شود. این چرخ با توجه به تعداد کروموزوم‌ها به گونه ای تقسیم می‌شود که هر بخش متناظر با مقدار برازندگی کروموزوم مربوطه باشد. سپس به تعداد اعضاء در جمعیت، چرخ دوار را چرخانده و هرکجا که چرخ متوقف شد با توجه به قطاع مشخص شده توسط اشاره گر چرخ دوار را چرخانده شده و هر بار یک عضو انتخاب می‌شود. در این روش به اعضاء دارای برازش بالا توجه زیادی می‌شود. به همین دلیل مشکل بزرگ انتخاب چرخ رولت، فشار انتخاب بالای آن است که ممکن است استفاده از آن در الگوریتم تکاملی سبب همگرایی زودرس الگوریتم شود. چنانچه در روش انتخاب چرخ رولت، بجای یک اشاره گر (تعداد اشاره گرها برابر با اندازه جمعیت یعنی  $n_s$  است) برای انتخاب اعضاء استفاده گردد، احتمال انتخاب اعضاء با برازش زیاد تعدیل شده و شانس

انتخاب کروموزوم‌های ضعیف تر بیشتر می‌شود. در این روش، چرخ رولت تنها یک بار (به جای  $n_s$  بار) چرخانده می‌شود. از آنجا که تعداد اشاره گرها برابر با  $n_s$  است، کل جمعیت انتخابی با یک بار چرخانده شدن چرخ رولت ایجاد می‌شود.

توجه کنید که در روش چرخ رولت با چند اشاره گر، فاصله اشاره گرهای در اطراف چرخ یکسان است. دلیل این امر هم آن است که با این کار احتمال انتخاب پاسخ‌ها به یکدیگر وابسته می‌شود. بدیهی است که در روش چرخ رولت سنتی با یک اشاره گر، احتمال انتخاب پاسخ‌ها در هر دور چرخش چرخ از یک دیگر مستقل بود و همین استقلال باعث می‌شد که بدون توجه به اعضاء انتخاب شده در دوره‌های قبل شانس اعضاء برازنده همیشه در مقایسه با اعضاء ضعیف عددی ثابت و بزرگ باشد. با توجه به توضیحات مطرح شده، کاملاً مشخص است که فشار انتخاب روش چرخ رولت بهبود یافته با چند اشاره گر، نسبت به نوع سنتی با یک اشاره گر، کم تر است.

### ۱-۱۰-۳ انتخاب رتبه ای

در روش انتخاب رتبه ای، بجای استفاده از مقدار مطلق برازندگی، از رتبه برازندگی اعضاء در جمعیت برای تعیین احتمال انتخاب استفاده می‌شود. در این روش، برازش برترین عضو جمعیت مساوی با  $n_s$  در نظر گرفته می‌شود. به دومین عضو برتر جمعیت برازش  $n_s - 1$  نسبت داده می‌شود و این کار تا رسیدن به ضعیف ترین عضو جمعیت ادامه می‌یابد (بدیهی است که برازش ضعیف ترین عضو برابر با ۱ خواهد بود). توجه کنید که در صورت یکسان بودن برازش دو عضو در جمعیت، بایستی به صورت تصادفی به یکی برازش  $k$  و به دیگری برازش  $k - 1$  را نسبت دهیم. در ادامه با استفاده از رابطه (۱-۱)، اعضاء انتخابی را برای معرفی به عملگرهای تولید مثل مشخص می‌کنیم. روش انتخاب رتبه ای در قیاس با روش انتخاب چرخ رولت از فشار انتخاب پایین تری برخوردار است؛ زیرا شانس بالای اعضاء برازنده را در جمعیت تعدیل می‌نماید. به همین دلیل احتمال همگرایی زودرس الگوریتم تکاملی با استفاده از این روش انتخاب کاهش خواهد یافت.



### ۱-۱۰-۴ انتخاب مسابقه ای

الگوریتم انتخاب مسابقه ای، در گام اول یک گروه (به تعداد  $t$  عدد) از موجودات را به طور تصادفی از جمعیت اعضاء انتخاب می‌کند؛ البته با فرض اینکه  $t < n_s$  است ( $n_s$  مجموع کل افراد درون یک جمعیت است). در گام دوم الگوریتم، کارآیی  $t$  عضو انتخابی با یکدیگر مقایسه و بهترین فرد از درون این گروه انتخاب می‌شود.

در صورتی که اندازه  $t$  خیلی بزرگ نباشد، این الگوریتم از انتخاب بهترین افراد جلوگیری می‌کند. بنابراین فشار انتخاب این الگوریتم کم می‌باشد. از طرف دیگر اگر  $t$  خیلی کم باشد شانس انتخاب ضعیف ترین اعضاء افزایش می‌یابد.

با وجود آنکه الگوریتم انتخاب مسابقه ای از اطلاعات برازندگی برای انتخاب بهترین فرد در یک رقابت استفاده می‌کند، انتخاب تصادفی افراد در گام اول الگوریتم، مانع از افزایش فشار انتخاب این الگوریتم می‌شود. به هر حال باید توجه داشت که فشار انتخاب ارتباط مستقیمی با اندازه  $t$  دارد. اگر  $t = n_s$  باشد همیشه بهترین فرد انتخاب می‌شود و در نتیجه فشار انتخاب بسیار زیاد خواهد شد. از طرف دیگر اگر  $t = 1$  باشد، این الگوریتم تبدیل به انتخاب تصادفی می‌شود. در بیشتر مقالات و کتب مقدار  $t$  معمولا برابر با ۲ و یا ۳ تنظیم می‌شود.

### ۱-۱۰-۵ انتخاب برشی

در این روش، که توسط گلدبرگ معرفی شده است، ابتدا اعضاء جمعیت را براساس شایستگی شان مرتب نموده و سپس از میان  $T$  درصد از برترین اعضاء،  $n_s$  عضو را به صورت تصادفی انتخاب می‌کنیم. در این روش هر چقدر مقدار  $T$  بزرگتر باشد، فشار انتخاب کمتر خواهد شد. در حالتی که  $T = 100$  باشد، این روش انتخاب معادل با روش انتخاب تصادفی است.

### ۱-۱۰-۶ نخبه گرایی

نخبه گرایی به فرآیند تضمین اینکه بهترین افراد جمعیت فعلی تا نسل بعدی زنده

می ماند اشاره دارد. بهترین افراد بدون اینکه جهش یابند، به جمعیت جدید کپی می شوند. هر چه افراد بیشتری تا نسل بعدی زنده بمانند، تنوع این جمعیت جدید کمتر است.

### ۱-۱۰-۷ تالار مشاهیر

تالار مشاهیر طرح انتخابی مشابهی با فهرست بهترین بازیکنان بازی آرکید است. برای هر نسل، بهترین فرد برای قرار گرفتن در تالار مشاهیر انتخاب می شود. بنابراین تالار مشاهیر شامل آرشیوی از بهترین افرادی است که از نسل اول یافت شده اند. تالار مشاهیر را می توان به عنوان استخری از والدین برای عملگر همگذاری استفاده کرد، یا، در نسل آخر، بهترین فرد به عنوان بهترین مورد در تالار مشاهیر انتخاب می شود.

### ۱-۱۱ عملگرهای تولید مثل

تولید مثل، فرایند تولید فرزندان از والدین انتخاب شده توسط بکارگیری عملگرهای بازترکیب یا هم برش و جهش می باشد.

بازترکیب، فرآیند تولید یک و یا چند فرزند از طریق ترکیب ژن های تصادفی انتخاب شده از دو و یا بیشتر از دو والد می باشد. این عملگر، برگرفته از مفهوم جفت گیری میان موجودات است. در الگوریتم های تکاملی، عملگر بازترکیب بر روی اعضای برازنده در جمعیت کروموزومها انجام می شود.

حاصل این عملیات فرزندان هستند که خصوصیات مشابه با والدین خود دارند. ماهیت مهم این عملگر، انتقال مقادیر مربوط به ژن های کروموزوم های والدها به فرزندان است. توجه کنید که برای طراحی یک عملگر بازترکیب مناسب، بایستی ماهیت آن همیشه در نظر گرفته شود. در غیر این صورت این عملگر اثربخشی خود را نخواهد داشت. در عملگر باز ترکیب، ایده اصلی آن است که فرزندان تولید شده ژن های والدین خود را به ارث برده و پاسخ هایی با برازش بهتر تولید کنند. البته لزوماً همیشه فرزندان تولید شده برازش بالاتری نسبت به والدین خود ندارند. در صورتی که انتخاب بر روی برترین پاسخها متمرکز باشد، ممکن است فشار انتخاب منجر به همگرایی زودرس الگوریتم

شود. دلیل این رویداد آن است که استفاده از عملگر بازترکیب در طی نسل‌های طولانی منجر به کاهش تنوع در جمعیت خواهد شد. به همین دلیل و با توجه به آنکه عملگر بازترکیب از دستاوردهای جستجو استفاده مستقیم می‌کند، استفاده بیشتر از این عملگر سبب تقویت قابلیت انتفاع در الگوریتم تکاملی شود. عملگر بازترکیب با احتمال  $P_s$  بر روی اعضاء جمعیت انتخاب شده انجام می‌شود.

جهش، فرایند تغییر تصادفی مقدار ژن‌ها در یک کروموزوم می‌باشد. هدف اصلی در عملگر جهش یافتن مقادیر جدید برای ژن‌های فرزندان تولید شده است (مقادیری که در هیچ کدام از والدین وجود نداشته است)، تا تنوع ژنوتایی در جمعیت افزایش یابد. جهش باید به گونه ای انجام شود که ژن‌های خوب در پاسخ‌های برتر خراب نشوند. با توجه به آنکه عملگر جهش سبب بروز یک تغییر تصادفی در کروموزوم شده و استفاده از این عملگر، احتمال یافته شدن مقادیر جدید را برای ژن‌ها افزایش می‌دهد، بهره برداری بیشتر از این عملگر سبب تقویت قابلیت پویش در الگوریتم تکاملی خواهد شد. عملگر جهش با احتمال  $P_m$  بر روی فرزندان تولید شده انجام می‌شود.

انواع مختلف عملگرهای بازترکیب و جهش، بستگی کاملی با انواع گوناگون روش‌های نمایش کروموزوم‌ها در الگوریتم تکاملی دارد. در ادامه این بخش به تشریح برخی از مشهورترین عملگرهای بازترکیب و جهش، با توجه به روش‌های نمایش دودویی، اعداد حقیقی، جایگشت عناصر و درختی خواهیم پرداخت. توجه کنید که کلیه عملگرهای بازترکیب که در این بخش معرفی می‌شوند بر روی دو والد انجام می‌شوند و یک کروموزوم جهش یافته را ایجاد می‌کنند. البته بایستی عنوان نمود که عملگرهای بازترکیبی که بر روی تعداد بیشتری والد انجام می‌شوند نیز وجود دارند. برای مطالعه بیشتر در این زمینه به مراجع انتهایی این فصل مراجعه کنید.

### ۱-۱۱-۱ بازترکیب و جهش در نمایش دودویی

در این بخش، عملگرهای مشهور بازترکیب و جهش در نمایش دودویی معرفی خواهند شد. این عملگرها عبارتند از: عملگرهای هم برش (بازترکیب) تک نقطه ای، دو

نقطه ای، یکنواخت و عملگر جهش معکوس سازی بیت.

در عملگر بازترکیب تک نقطه ای، ابتدا یک نقطه تصادفی در دنباله کروموزوم‌های والدین انتخاب می‌شود و سپس از محل انتخاب شده، کروموزوم هر دو والد برش می‌خورد. بخش اول والد اول و بخش دوم والد دوم برای تولید فرزند نخست استفاده می‌شود. فرزند دوم، شامل بخش اول والد دوم و بخش دوم والد اول است (شکل ۱-۲).

در عملگر بازترکیب دو نقطه ای، دو نقطه تصادفی در دنباله کروموزوم‌های والدین انتخاب می‌شوند و سپس از محل آن نقطه‌ها، کروموزوم والدین برش می‌خورد. بخش‌های اول و سوم والد اول و بخش دوم والد دوم برای تولید فرزند نخست استفاده می‌شود. فرزند دوم، شامل بخش اول و سوم والد دوم و بخش دوم والد اول است (شکل ۱-۲). بدیهی است که با افزایش تعداد نقاط شکست در عملگر بازترکیب چند نقطه ای، شباهت فرزندان به هر کدام از والدین کمتر شده و توانایی پویایی عملگر بازترکیب تقویت خواهد شد.

در عملگر بازترکیب یکنواخت، مقدار ژن فرزند با توجه به مقادیر ژن‌های متناظر هر دو والد انتخاب می‌شود. در این روش، مقادیر ژن‌های هر کدام از والدین، شانسی برابر برای حضور در ژن متناظر فرزند دارند. در عملگر بازترکیب یکنواخت، براساس یک توزیع تصادفی دودویی مشخص می‌شود که مقدار هر ژن فرزند از مقدار ژن متناظر کدام والد انتخاب گردد (شکل ۱-۲).

در عملگر جهش معکوس سازی بیت، یک ژن به طور تصادفی انتخاب شده و مقدارش هر چه که باشد تغییر می‌کند (شکل ۱-۳). مشخص است که این عمل به دلیل عدم بهره برداری از اطلاعات موجود در جمعیت، سنخیت کاملی با تعریف عملگر جهش دارد.

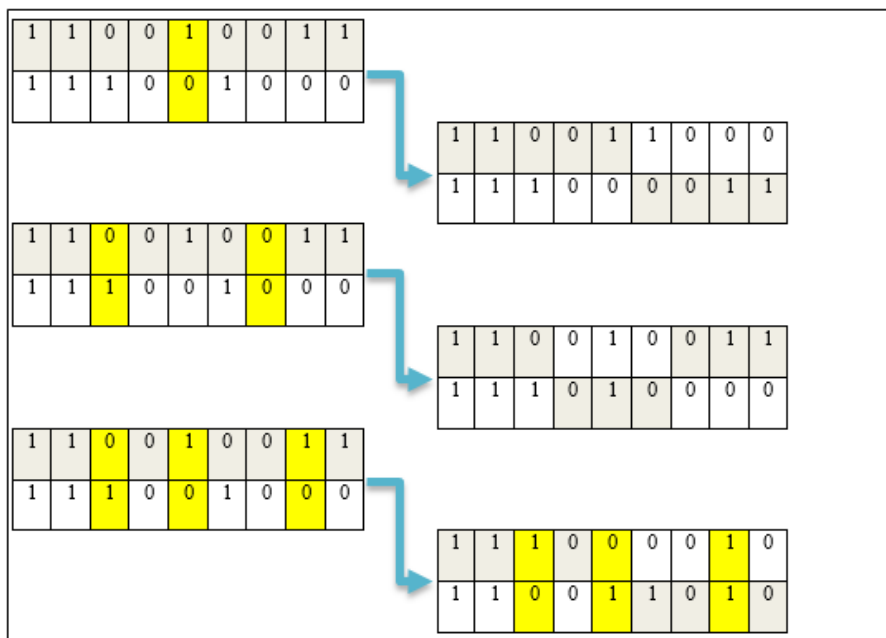
### ۱-۱۱-۲ بازترکیب و جهش در نمایش اعداد حقیقی

در این بخش، عملگرهای شناخته شده بازترکیب در نمایش اعداد حقیقی معرفی خواهند شد. این عملگرها عبارتند از: عملگرهای بازترکیب ساده، حسابی ساده، حسابی

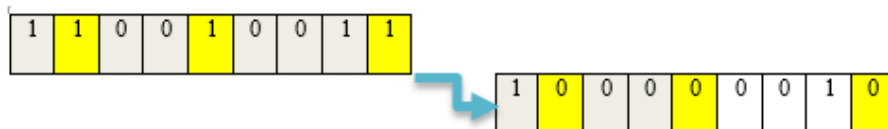
کامل و عملگر جهش مکمل.

در عملگر بازترکیب ساده، ابتدا یک بخش مشابه در کروموزوم‌های والدین انتخاب می‌شود. سپس بخش انتخاب شده در والد اول مستقیماً به فرزند اول و بخش انتخاب شده در والد دوم نیز مستقیماً به فرزند دوم منتقل می‌شود. در ادامه، مقادیر ژن‌های مربوط به بخش‌های انتخاب نشده دو والد با یکدیگر جمع شده و نتیجه در عدد  $\alpha$  که در بازه (۰ و ۱) است ضرب می‌شود. نتیجه هر جفت ژن، به ژن معادل در فرزند اول منتقل می‌شود. همین کار برای فرزند دوم تکرار می‌شود؛ لیکن بجای  $\alpha$ ، حاصل جمع ژن‌های دو والد در  $1-\alpha$  ضرب می‌شود (شکل ۱-۴).

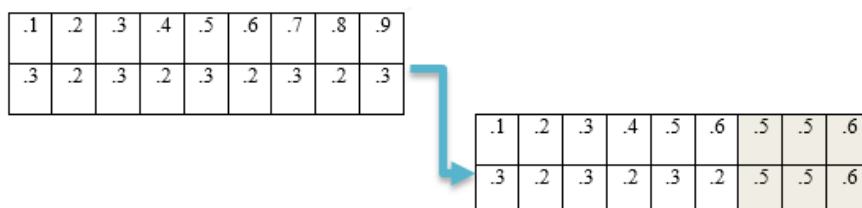
در این عملگر، بجای جمع نمودن مقادیر ژن‌ها در والدین، می‌توان از هر عملگر ریاضی دیگر و یا حتی تابع ریاضی پیچیده‌ای استفاده نمود.



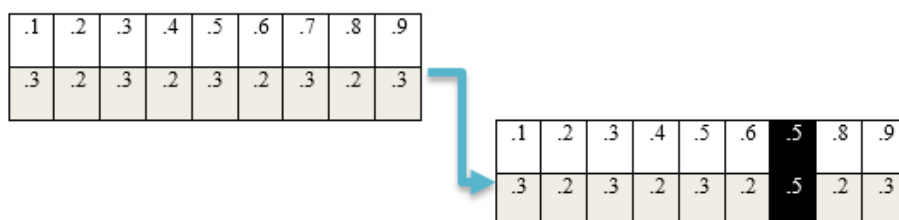
شکل ۱-۲ عملگرهای بازترکیب تک نقطه‌ای، دو نقطه‌ای و یکنواخت در نمایش دودویی



شکل ۱-۳ عملگر جهش معکوس سازی بیت در نمایش دودویی

شکل ۱-۴ عملگر بازترکیب ساده با  $\alpha=0.5$  در نمایش اعداد حقیقی

عملگر بازترکیب حسابی تکین، مشابه با عملگر بازترکیب ساده است؛ با این تفاوت که تنها یک ژن در دو والد برای تولید فرزندان دست خوش تغییر می شود (شکل ۱-۵).

شکل ۱-۵ عملگر بازترکیب حسابی تکین با  $\alpha=0.5$  در نمایش اعداد حقیقی

در عملگر حسابی کامل، همه ژن ها در والدین تغییر کرده و فرزندان در هیچ کدام از ژن هایشان شباهتی به هیچ یک از والدین خود ندارند (شکل ۲-۶) این عملگر بازترکیب در مقایسه با دیگر عملگرهای بازترکیبی در نمایش اعداد حقیقی قابلیت بالاتری دارد.