



وزارت علوم، تحقیقات و فناوری  
مؤسسه آموزش عالی غیردولتی غیرانتفاعی جهاد

پایان نامه دوره کارشناسی ارشد مهندسی برق  
گرایش الکترونیک

ارائه یک ابزار طراحی مدارات مجتمع CMOS با الگوریتم ژنتیک چند هدفه

کامران دلفان همتی

استاد راهنما:  
دکتر عباس گلمکانی

خرداد ۹۰

## تقدیر و تشکر:

باتقدیر و تشکر از استاد کرامی دکتر عباس گل‌مکانی و آرزوی توفیقات روز افزون برای ایشان، که در تمامی مراحل گردآوری و تنظیم این پایان‌نامه صبورانه و با اشتیاق مرا

یاری نمودند.

باتشکر از کلیه اساتیدم در دوره کارشناسی ارشد، که مطالب زیادی از آنها آموختم.

## چکیده

در این پروژه یک CAD به منظور طراحی و بهینه سازی مدارهای مجتمع دیجیتال و RF، پیاده سازی شده است. مسأله مهم در طراحی مدارهای مجتمع (بویره در RF) در تکنولوژی CMOS، توجه به عناصر پارازیتیک ترانزیستور، خازن و سلفهای مجتمع آن است و این مسأله محاسبات دستی طراحی را پیچیده می کند. یک ویژگی مهم این روش استفاده از مدل های دقیق عناصر و در نظر گرفتن کلیه عناصر پارازیتیک در طراحی است لذا جواب بدست آمده به واقعیت بسیار نزدیک است. در این پروژه پس از انتخاب پیکربندی مدار توسط کاربر، مقادیر اجزای مدار شامل ابعاد ترانزیستورها، ولتاژهای بایاس، تعداد دور و قطر سیم پیچ سلف های مدار توسط الگوریتم بهینه سازی پیشنهاد شده و مقادیر بهره، توان مصرفی، S11، S22، عدد نویز و تاخیر توسط این الگوریتم بهینه می شوند. پس در طراحی ما با یک مسأله بهینه سازی چندهدفه روبرو هستیم.

در این پروژه برای بهینه سازی مدارهای مجتمع، الگوریتمی با نام «الگوریتم ژنتیک دسته بندی غلبه نشده مبتنی بر حفظ دسته اول» ارایه شده است. این الگوریتم از ترکیب دو الگوریتم SPEA و NSGA2 حاصل شده است. به این صورت که ابتدا یک جمعیت اولیه تصادفی به اندازه N تشکیل داده و با استفاده از عملگرهای معمول الگوریتم ژنتیک N عضو دیگر ساخته می شود و سپس کل اعضا را در یک مجموعه جدید به اندازه 2N ریخته و مشابه NSGA2 رتبه بندی غلبه نشده صورت می پذیرد. در NSGA2 اگر تعداد افراد رتبه اول بیشتر از N شود افرادی که احتمالاً پراکندگی خوبی ندارند حذف می شوند تا جایی که این تعداد به N برسد. اما در این مواقع این الگوریتم مانند SPEA عمل کرده و این افراد را در یک بایگانی خارجی می گنجاند و از هرس کردنشان خودداری می کند (این کار باعث افزایش سرعت همگرایی الگوریتم می شود چون خیلی از مواقع ممکن است جوابهای خوبی که پس از چندین تکرار حاصل شده است با هرس کردن از بین روند). از بین اعضای بایگانی کسانی که فاصله پراکندگی بیشتری داشته باشند احتمال انتخاب بیشتری برای تشکیل نسل بعد خواهند داشت. برنامه های الگوریتم با Matlab نوشته شده است و شبیه سازی مدار بوسیله Hspice و Hspice-RF با تکنولوژی CMOS 0.18um صورت گرفته است .

در این پروژه چندین ساختار تمام جمع کننده و LNA مجتمع، شامل ترکیب های جمع کننده هیبرید سرعت بالا، سلول جمع کننده هیبرید با سرعت بسیار بالا پیشنهادی، کم کردن توان مصرفی در یک تمام جمع کننده هیبرید سرعت بالا، LNA به صورت تکرار کننده جریان و مدار LNA پیشنهادی با فیلتر حذف تصویر معرفی خواهد شد. این مدارها با استفاده از CAD ارایه شده، طراحی و بهینه شده اند.

## فصل اول:

### ۱ ..... پیش گفتار

۲ ۱-۱- مقدمه

۳ ۲-۱- ساختار پایان نامه

## فصل دوم

### ۵ ..... الگوریتم بهینه‌سازی پیشنهادی

۶ ۱-۲- مروری بر الگوریتم ژنتیک

۶ ۲-۲- مقدمه ای از ژنتیک

۶ ۳-۲- تعریف الگوریتم ژنتیک

۷ ۴-۲- اساس الگوریتم ژنتیک

۷ ۵-۲- بخش های اصلی الگوریتم ژنتیک

۸ ۶-۲- کد گذاری

۸ ۷-۲- عملگرهای الگوریتم ژنتیک

۸ ۱-۷-۲- انتخاب

۱۰ ۲-۷-۲- قطع، تزیج، ترکیب، آمیزش

۱۲ ۳-۷-۲- جهش

۱۲ ۸-۲- احتمال جهش و ترکیب

۱۳ ۹-۲- پارامترهای الگوریتم ژنتیک

۱۳ ۱۰-۲- مزایای استفاده از الگوریتم ژنتیک

۱۴ ۱۱-۲- کاربردهای الگوریتم ژنتیک

۱۴ ۱۲-۲- الگوریتم ژنتیک در یک نگاه

۱۶ ۲-۲- الگوریتم ژنتیک چند هدفه (MOGA)

۱۶ ۳-۲- مسئله بهینه سازی چند هدفه (MOP)

۱۷ ۱-۳-۲- تبدیل مسأله به یک مسأله تک هدفه

۱۷ ۲-۳-۲- بهینه سازی بر اساس جبهه پارتو

۱۹ ۴-۲- الگوریتم تکاملی نقطه‌ی قوت پارتو ۲ (SPEA2)

۲۱	۵-۲- الگوریتم ژنتیک مبتنی بر مرتب سازی نخبه گرای غلبه نشده (NSGA2)
۲۳	۶-۲- الگوریتم بهینه سازی پیشنهادی
۲۳	۱-۶-۲- پیاده سازی الگوریتم پیشنهادی
۲۳	۲-۶-۲- ایجاد جمعیت به اندازه $2N$
۲۳	۳-۶-۲- محاسبه برازندگی و رتبه بندی غلبه نشده
۲۵	۴-۶-۲- به روزرسانی کلیه افراد
۲۵	۵-۶-۲- انتخاب

### فصل سوم

۲۷	طراحی و بهینه سازی مدارات دیجیتال .....
۲۸	۱-۳- مقدمه
۲۹	۲-۳- پارامترهای طراحی تمام جمع کننده
۲۹	۳-۳- طراحی تمام جمع کننده هیبرید سرعت بالا
۳۰	۱-۳-۳- پیاده سازی خازن با Moscap
۳۲	۲-۳-۳- ایجاد یک بهبود در مدار
۳۶	۳-۳-۳- چگونه می توان توان مصرفی معکوس کننده اول را کاهش داد
۳۹	۴-۳-۳- بهینه سازی مدار بهبود یافته برای رسیدن به بهترین جواب ممکن
۴۰	۴-۳- ارائه یک تکنیک مداری برای کم کردن توان مصرفی یک تمام جمع کننده هیبرید سرعت بالا
۴۰	۱-۴-۳- قرار دادن ولتاژ بهینه و پیاده سازی مدار پیشنهادی
۴۱	۲-۴-۳- نتایج شبیه سازی های نهایی
۴۲	۵-۳- سلول جمع کننده هیبرید با سرعت بسیار بالا پیشنهادی

### فصل چهارم

۴۶	طراحی و بهینه سازی تقویت کننده های کم نویز .....
۴۷	۱-۴- طراحی LNA
۵۱	۲-۴- چند ساختار LNA
۵۱	۱-۲-۴- تقویت کننده کم نویز به صورت تکرار کننده جریان

۵۵

۳-۴- نتایج طراحی و بهینه سازی

۵۵

۳-۴-۱- بهینه‌سازی مدار LNA با ساختار تکرار کننده جریان

۵۸

۳-۴-۲- بهینه‌سازی مدار LNA پیشنهادی با فیلتر حذف تصویر

## فصل پنجم

۶۲

نتیجه‌گیری و پیشنهادات

۶۳

۵-۱- نتیجه‌گیری

۶۳

۵-۲- پیشنهاداتی برای ادامه کار

۶۵

مراجع

## فهرست شکل‌ها و جداول

۹	شکل (۱-۲): قاعده کلی انتخاب چرخ رولت [6]
۱۱	شکل (۲-۲) Crossover با نقطه منفرد [7]
۱۱	شکل (۳-۲) Crossover با دو نقطه [7]
۱۲	شکل ۲-۴: اعمال عملگر جهش [8]
۱۵	شکل ۲-۵: نمایش یک الگوریتم نمونه
۱۶	شکل ۲-۶: تصویر نقشه‌ای از فضای متغیرهای تصمیم‌گیری به فضای تابع هدف [9]
۱۸	شکل (۷-۲): مفهوم غالب بودن و جبهه بهینه پارتو [11]
۲۱	شکل ۲-۸: عملکرد NSGA(II) [12]
۲۴	شکل ۲-۹: فلوجارت الگوریتم بهینه‌سازی پیشنهاد شده
۲۵	شکل ۲-۱۰: حلقه اجرای Hspice با Matlab
۲۹	شکل ۳-۱: سلول تمام جمع‌کننده هیبرید سرعت بالا الف) SUM ب) CARRY [13].
۳۰	شکل ۳-۲: نتایج حاصل از بهینه‌سازی تمام جمع‌کننده شکل ۳-۱: a) sum و b) carry
۳۱	شکل ۳-۳: پیاده‌سازی خازن با Moscap
۳۲	شکل ۳-۴: مدار سازنده کری خروجی با ذکر نام گره‌ها
۳۳	شکل ۳-۵: شکل موج ورودی های a و b و c
۳۳	شکل ۳-۶: شکل موج دیده شده در X
۳۳	شکل ۳-۷: شکل موج دیده شده در Y
۳۴	شکل ۳-۸: شکل موج دیده شده در Cout
۳۵	شکل ۳-۹: مدار حاصل از حذف معکوس‌کننده اول در مدار سازنده کری خروجی
۳۶	شکل ۳-۱۰: مدار تشکیل دهنده تابع اکثریت
۳۷	شکل ۳-۱۱: اعمال ولتاژهای جدید به دو سر معکوس‌کننده اول
۳۸	شکل ۳-۱۲: گره a) x، گره b) y، گره c) Cout
۳۹	شکل ۳-۱۳: نتایج حاصل از بهینه‌سازی مدار شکل ۳-۱۰
۴۰	شکل ۳-۱۴: اعمال دو ولتاژ متغییر به سرهای معکوس‌کننده اول

- شکل ۳-۱۵: طرح پیشنهاد شده برای کم کردن توان مصرفی مدار سازندهی کری خروجی
- شکل ۳-۱۶: مقایسه نتایج حاصل از بهینه سازی مدارات مرجع [13] و ساختارهای پیشنهادی این پروژه
- شکل ۳-۱۷: سلول جمع کننده پیشنهاد شده: (a) مجموع (b) کری خروجی
- شکل ۳-۱۸: (a) تمام جمع کننده (24T) (b) سلول تمام جمع کننده CMOS متداول (c) سلول تمام جمع کننده (CPL) (d) سلول تمام جمع کننده (TGA).
- شکل ۴-۱: الف) مدار LNA کسکود ب) مدار معادل [28]ac
- شکل ۴-۲: الف) مدار LNA کسکود با سلفهای تطبیق ب) مدار معادل [28]ac
- شکل ۴-۳: تقویت کننده کم نویز به صورت تکرار کننده جریان [29]
- شکل ۴-۴: فیلتر و تقویت کننده پیشنهادی
- شکل ۴-۵: مدار LNA پیشنهادی با فیلتر حذف تصویر
- شکل ۴-۶: نتایج پارامترهای پراکندگی (S) و NF، اولین جواب منتخب حاصل از بهینه سازی شکل ۴-۳، (a) s11، (b) s22، (c) s21 و (d) NF
- شکل ۴-۷: بهینه سازی S21 مربوط به مدار شکل ۴-۴
- شکل ۴-۸: یک جواب منتخب از شکل ۴-۷
- شکل ۴-۹: نتایج پارامترهای پراکندگی (S) و NF، اولین جواب منتخب حاصل از بهینه سازی شکل ۴-۵، (a) s11، (b) s22، (c) s21 و (d) NF
- جدول ۳-۱: جوابهای برگزیده بهینه سازی
- جدول ۳-۲: مقایسه توان مصرفی در مدار مرجع [13] و ساختار پیشنهادی در یک سرعت خاص
- جدول ۳-۳: نتایج شبیه سازی و مقایسه با مراجع
- جدول ۴-۱: محدوده تغییرات ابعاد المانها
- جدول ۴-۲: نتایج طراحی مدار LNA با ساختار تکرار کننده جریان در فرکانس 5.7GHz



# فصل اول

## پیشگفتار

امروزه بسیاری از پردازش‌های سیگنال به گستره دیجیتال برده شده است همچنین مدارهای آنالوگ به عنوان پایه‌ای برای بسیاری از سیستم‌های پیچیده و کارای امروزی ضروری‌اند. حقیقت مهم در صنعت نیمه رسانای امروزی این است که طراحی مدارهای آنالوگ هم، با استفاده از فناوری‌های IC که بر روی ساخت محصولات دیجیتال بکار می‌روند، صورت می‌گیرد. این فناوری‌ها که بر روی کاربردهای دیجیتال توسعه یافته و مشخص شده‌اند به آسانی قابل بکارگیری در ساخت طرح‌های آنالوگ نیستند و برای دستیابی به کارایی بالا نیاز به مدارها و معماری‌های جدید است.

ایده قرارداد مدارهای الکترونیکی روی یک زیربنا در اوایل دهه ۹۵۰ پدید آمد، در خلال این سالها، فناوری از تئوری تراشه‌های ساده که شامل چندین مولفه‌اند تا ساخت حافظه‌هایی با بیش از یک بلیون ترانزیستور و ریز پردازنده بیش از ده میلیون افزاره، توسعه یافته است. همانگونه که گوردن مور (یکی از بنیان گزاران اینتل) در اوایل دهه ۱۹۷۰، پیش بینی کرد. تعداد ترانزیستورها در هر تراشه به ازای هر ۱/۵ سال، تقریباً دو برابر می‌شود. حداقل ابعاد ترانزیستور از  $25 \mu\text{m}$  در سال ۱۹۶۰ به حدود  $35 \text{ nm}$  در حال حاضر رسیده است که باعث پیشرفت قابل ملاحظه‌ای در سرعت مدارهای مجتمع شده است. فناوری‌های مدار مجتمع که عمدتاً توسط بازار حافظه و ریزپردازنده‌ها تحریک می‌شوند، منجر به تقویت طراحی آنالوگ نیز بطور گسترده شده‌اند و پیچیدگی، سرعت و دقتی را بدست می‌دهند که دستیابی به آنها با پیاده‌سازی‌های گسسته غیرممکن است. مدارهای مجتمع آنالوگ و مرکب (آنالوگ و دیجیتال) که شامل ده‌ها هزار افزاره هستند، اکنون بطور گسترده‌ای در محصولات مصرفی بکار می‌روند. اکنون دیگر نمی‌توانیم برای پیش بینی رفتار و کارایی مدارهای مجتمع آنالوگ و دیجیتال، یک نمونه گسسته از آن را بسازیم [1].

اگر چه، طراحی و آنالیز دو فرایند متفاوت هستند. ساخت مدار، فرایند طراحی و ایجاد شبکه‌ای برای تامین یک پاسخ معین به تحریکی تعیین شده است. که با اندازه و توپولوژی تمام اجزا سروکار دارد. توپولوژی یک مدار شامل نوع هر مولفه و اتصالات آنها است. اندازه‌ی مدار مرکب از بهای اجزای پیوسته شده با هر عضو مدار است. این بر عکس موضوع آنالیز که در آن با یک تحریک معین به کار گرفته شده در یک شبکه داده شده باید پاسخ محاسبه شود است. در مقابل، موضوع ساخت ممکن است یک راه حل منحصر به فرد نداشته باشد و غالباً فضای تحقیق زیاد می‌باشد. این باعث می‌شود طراحی مدار یک مساله کلاسیک (رده‌ای) برای تکنیک تکامل حل باشد.

در طراحی و بهینه‌سازی بلوک‌های دیجیتال و آنالوگ مجتمع، طراح پس از انتخاب ساختار، با توجه به روابط تئوری حاکم بر توپولوژی مربوطه، شروع به محاسبه ابعاد المان‌ها می‌کند. سپس به کمک نتایج این محاسبات، شبیه سازی مدار آغاز می‌شود. به علت تقریبی بودن روابط تئوری و عدم مدل سازی همه عوامل خطی و غیر خطی در آنها، به احتمال زیاد نتایج شبیه‌سازی با جواب‌های مورد انتظار متفاوت خواهند بود. در نتیجه لازم است طراح با توجه به تجربه اندکی که از طراحی‌های قبل و نتیجه تغییر هر یک از پارامترها دارد، اقدام به تغییر آنها کرده و شبیه سازی را تکرار نماید اما متأسفانه این کار بسیار وقتگیر بوده و هیچ تضمینی در رسیدن به ساختار بهینه‌تر ندارد. چراکه

پارامترهای مختلف با یکدیگر مصالحه داشته و بهبود هر یک از آنها می‌تواند باعث خراب شدن دیگری شود. در طراحی و بهینه‌سازی به کمک الگوریتم ژنتیک چند هدفه طراح کفایت پارامترهای مختلف ورودی مدار و محدوده مجاز تغییرات هر یک را مشخص کند، پس از آن الگوریتم ژنتیک شروع به جستجو کرده و مجموعه جواب‌های بهینه (جبهه پارتو) را از نقطه نظرهای لازم از جمله نسبت سیگنال به نویز خروجی، توان مصرفی، تاخیر و غیره معرفی می‌کند. در نهایت طراح با یک دسته جواب قابل قبول مواجه می‌شود که هر کدام از نقطه نظری بهتر هستند. پس بر حسب نیاز یکی را برای پیاده سازی نهایی انتخاب می‌کند [2].

اولین روش الگوریتم ژنتیک چند هدفه، VEGA<sup>1</sup> است که توسط شافر<sup>2</sup> در سال 1985 ابداع شد [3]. سپس روشهای بهینه سازی چند هدفه زیادی بر همین مبنا بسط و گسترش یافتند. روش NPGA<sup>3</sup> توسط هورن، گلدبرگ و نافپایوتیس<sup>4</sup> (1994)، NSGA<sup>5</sup> توسط سرینیواس و دب<sup>6</sup> (1994)، SPEA<sup>7</sup> توسط زیتزلر و تیل<sup>8</sup> (1999)، SPEA2<sup>9</sup> توسط زیتزلر و همکارانش (2001)، PAES<sup>9</sup> توسط نولس و کورن<sup>10</sup> (2000) و سرانجام روش NSGA-II<sup>11</sup> توسط گروه دب، پراتاب، آگاروا و میاریوان<sup>11</sup> (2002) ابداع شدند [4].

در این پروژه ما از ترکیب دو الگوریتم SPEA و NSGA2 استفاده کرده‌ایم که «الگوریتم ژنتیک دسته‌بندی غلبه نشده مبتنی بر عدم حذف دسته اول» نامگذاری شده است. و به این صورت است که ابتدا یک جمعیت اولیه تصادفی به اندازه  $N$  تشکیل داده و با استفاده از عملگرهای معمول الگوریتم ژنتیک  $N$  عضو دیگر ساخته می‌شود و سپس کل اعضا را در یک مجموعه جدید به اندازه  $2N$  ریخته و مشابه NSGA2 رتبه‌بندی غلبه نشده صورت می‌پذیرد. در NSGA2 اگر تعداد افراد رتبه اول بیشتر از  $N$  شود افرادی که احتمالاً پراکندگی خوبی ندارند حذف می‌شوند تا جایی که این تعداد به  $N$  برسد. اما ما در این مواقع مانند SPEA عمل می‌کنیم و این افراد را در یک بایگانی خارجی گنجانده و از هرس کردنشان خودداری می‌کنیم.

## ۱-۲- ساختار پایان نامه

فصل اول پایان نامه، با هدف آگاهی از تلاش‌ها و مطالعات انجام شده در زمینه طراحی مدارات دیجیتال و آنالوگ بیان انگیزه اصلی انجام این پروژه و ارائه اطلاعات کلی به خواننده محترم تنظیم شده است. در فصل دوم به تشریح الگوریتم ژنتیک پرداخته و پس از شرح مراحل انجام آن عملگرهای این الگوریتم را مورد بررسی قرار داده و مزایا، معایب و کاربردهای آن را مورد توجه قرار داده و پس از آن الگوریتم ژنتیک چند هدفه را معرفی کرده چرا که

<sup>1</sup> Vector evaluated Genetic algorithm

<sup>2</sup> Schaffer

<sup>3</sup> Niche Pareto Genetic algorithm

<sup>4</sup> Horn, Goldberg and Nafpliotis

<sup>5</sup> Non-dominated Sorting Genetic Algorithm

<sup>6</sup> Srinivas and Deb

<sup>7</sup> Strength Pareto Evolutionary Algorithm

<sup>8</sup> Zitzler and Thiele

<sup>9</sup> Pareto Envelope-based Selection algorithm

<sup>10</sup> Knowles and Corne

<sup>11</sup> Deb, Pratap, Agarwa and Meyarivan

در بهینه‌سازی مدارهای مجتمع لازم است که چندین هدف بطور همزمان بهینه شوند که این کار با MOGA<sup>1</sup> قابل انجام است و نیز به تشریح الگوریتم تکاملی نقطه‌ی قوت پارتو (SPEA 2) و الگوریتم ژنتیک مبتنی بر مرتب‌سازی نخبه‌گرای غلبه‌نشده (NSGA2) پرداخته و نهایتاً الگوریتم پیشنهادی را شرح داده‌ایم.

در فصل سوم بهینه‌سازی مدارات دیجیتال پرداخته و ساختارهای مختلفی از تمام جمع‌کننده‌ها را معرفی و بهینه‌سازی نموده‌ایم همچنین دو ساختار برای تمام جمع‌کننده معرفی و نتایج حاصل از شبیه‌سازی و بهینه‌سازی را با ساختارهای دیگر مقایسه کرده‌ایم. فصل چهارم، ساختارهای مختلف LNA بررسی شده و مدارهای حذف تصویر برای آنها پیشنهاد شده است. در فصل پنجم به بحث و نتیجه‌گیری نتایج حاصله و ارائه پیشنهاداتی در زمینه ادامه این پژوهش پرداخته شده است. در ادامه، فهرستی از مراجع مورد استفاده در این پروژه ارائه شده است. امید است، که انجام این پروژه گامی در جهت بهبود و پیشبرد روشهای طراحی مدارات مجتمع دیجیتال و آنالوگ باشد.

---

<sup>1</sup> Multiple objective Genetic Algorithm

# فصل دوم

## الگوریتم بهینه‌سازی پیشنهادی

## ۱-۲- مروری بر الگوریتم ژنتیک

### ۲-۲- مقدمه ای از ژنتیک

در جهان اطراف ما همه ارگانیسم‌های حیاتی از ساختارهای قانونمندی تشکیل شده‌اند. ساختارهایی که از سوی آفریدگار هستی در بطن مخلوقات قرار داده شده است. همه این ارگانیسم‌ها از بلوک‌های پایه‌ای به نام سلول تشکیل به وجود آمده‌اند. قوانین مزبور در قالب ژن‌ها به صورت کد شده در هر ارگانیسم وجود دارند. از به هم وصل شدن این ژن‌ها، رشته‌هایی طولانی به نام کروموزوم تولید می‌شود. هر ژن نمایانگر یکی از خصوصیات آن ارگانیسم است، مثلاً رنگ چشم یا رنگ مو و البته هر ژن می‌تواند دارای مقادیر مختلفی باشد. برای نمونه در رابطه با رنگ چشم می‌توانیم دارای مقادیر متناظر با مشکی، قهوه‌ای، آبی و سبز و... باشیم. هنگامی که دو ارگانیسم به تولید مثل می‌پردازند، در حقیقت ژن‌های خود را با هم ترکیب می‌کنند. بدین صورت که ارگانیسم تولید شده دارای نیمی از ژن‌های یک والد ونیم دیگر از والد دیگری است. این عمل را ترکیب می‌نامیم. گاهی اوقات بعضی از ژن‌ها دارای جهش می‌شوند. این جهش تغییری در ساختار کروموزوم ایجاد نمی‌کند، اما با توجه به اینکه مقدار جدیدی به ژن تخصیص می‌یابد، موجب بروز خصوصیت جدید می‌شود که از این اتفاق با نام جهش یاد می‌کنیم.

### ۳-۲- تعریف الگوریتم ژنتیک

الگوریتم ژنتیک (GA) یک تکنیک برنامه نویسی است که از تکامل ژنتیکی به عنوان یک الگوی حل مسئله استفاده می‌کند. نقطه قوت الگوریتم ژنتیک توانایی بهینه‌سازی در حل مسائل گوناگون است. نام الگوریتم ژنتیک به علت تقلید عددی از فرایندهای زیستی در بهینه‌سازی حل است. مسئله‌ای که باید حل شود ورودی است و راه حل -ها طبق یک الگو کد گذاری می‌شوند که تابع برازندگی<sup>۱</sup> نام دارد و هر راه حل کاندید را ارزیابی می‌کند که اکثر آنها به صورت تصادفی انتخاب می‌شوند. الگوریتم ژنتیک (GA) یک تکنیک جستجو در علم رایانه برای یافتن راه حل بهینه و مسائل جستجو است. الگوریتم‌های ژنتیک یکی از انواع الگوریتم‌های تکاملی اند که از علم زیست‌شناسی مثل وراثت، جهش، انتخاب و ترکیب الهام گرفته شده است.

در هر مرحله از اجرای الگوریتم ژنتیکی، یک دسته از نقاط جستجو مورد پردازش‌های تصادفی قرار می‌گیرند. به این صورت که به هر نقطه دنباله‌ای از کاراکترها نسبت داده می‌شود و بر روی این دنباله‌ها، عملگرهای ژنتیکی اعمال می‌شود. سپس دنباله‌های بدست آمده دیکد می‌گردد تا نقاط جدید در فضای جستجو بدست آید. در آخر بر اساس این که تابع هدف در هر یک از نقاط چه مقدار باشد، احتمال شرکت نمودن آنها در مرحله بعد تعیین می‌گردد.

الگوریتم‌های ژنتیکی را می‌توان یک روش بهینه‌سازی تصادفی جهت‌دار دانست که به تدریج به سمت نقطه بهینه حرکت می‌کند. در مورد ویژگی‌های الگوریتم ژنتیک در مقایسه با دیگر روش‌های بهینه‌سازی می‌توان گفت که الگوریتمی است بدون داشتن هیچ اطلاعی از مسئله و هیچ محدودیتی بر نوع متغیرهای آن برای هرگونه مسئله‌ای قابل اعمال است و دارای کارایی اثبات شده‌ای در یافتن بهینه کلی<sup>۲</sup> می‌باشد. توانایی این روش در حل مسائل

<sup>1</sup> fitness

<sup>2</sup> Global Optimum

پیچیده بهینه‌سازی است که روش‌های کلاسیک یا قابل اعمال نیستند و یا در یافتن بهینه‌ی کلی قابل اطمینان نیستند.

## ۲-۴- اساس الگوریتم ژنتیک

الگوریتم ژنتیک با مجموعه جواب اولیه شروع می‌شود که به آن جمعیت اولیه گویند. هر فرد از این جمعیت یک کروموزوم است که به آن یک مقدار صلاحیت یا برازش نسبت داده می‌شود. بر اساس پاسخ‌های یک جمعیت، جمعیت جدید بعدی ایجاد می‌شود. این عمل با این امید انجام می‌شود که صلاحیت افراد از یک جمعیت به جمعیت دیگر بهبود یابد. در یک جمعیت پاسخ‌ها بر اساس صلاحیتشان انتخاب می‌شوند تا افراد جمعیت جدید (فرزندان) را بسازند، افرادی از جمعیت که صلاحیت بیشتری دارند، شانس بیشتری برای انتخاب خواهند داشت. عملیات جستجو در فضای پاسخ‌های قابل قبول (جستجو) ادامه می‌یابد تا به پاسخ‌های بهتر بیانجامد. الگوریتم ژنتیک تا آنجا پیش می‌رود که شرایط پایانی فرا رسد. مراحل فوق را می‌توان به صورت زیر خلاصه کرد:

۱. شروع: جمعیت اولیه شامل  $n$  کروموزوم است که به طور شانسی انتخاب می‌شود. این پاسخ‌ها می‌بایست قابل قبول باشند.

۲. صلاحیت: مقدار صلاحیت هر پاسخ یا کروموزوم تعیین می‌شود.

۳. جمعیت جدید: جمعیت جدید با تکرار مراحل زیر ایجاد می‌شود:

۱.۳ انتخاب: بر اساس صلاحیت کروموزوم، دو کروموزوم به عنوان والدین به طور تصادفی انتخاب می‌شوند.

۲.۳ ترکیب یا ترویج یا قطع: بر اساس احتمال (pc) (که در ادامه معرفی می‌شود) دو کروموزوم انتخابی با هم ترکیب شده تا دو کروموزوم جدید ایجاد شوند.

۳.۳ جهش: بر اساس احتمال جهش (pm) در هر دو کروموزوم جدید تغییر ایجاد می‌شود.

۴.۳ قرار دهی: پاسخ جدید در جمعیت جدید قرار می‌گیرند.

۴. جایگزین: جمعیت جدید جایگزین جمعیت قبلی شده و از آن برای ادامه الگوریتم استفاده می‌شود.

۵. آزمون: اگر شرایط پایانی فرا رسید، متوقف شده و بهترین پاسخ ارائه می‌شود، در غیر این صورت به مرحله دو باز گردانده می‌شود.

## ۲-۵- بخش‌های اصلی الگوریتم ژنتیک

به طور کلی الگوریتم ژنتیک شامل اجزای زیر است:

۱. رشته یا کروموزوم: نمایانگر یک نقطه در فضای جستجو است که ممکن است به صورت کد و یا به شکل اصلی متغیرهای مورد نظر باشد. در صورتی که فضای حل بیش از یک بعد داشته باشد می‌توان با پشت سر هم قرار دادن پارامترها، رشته را ایجاد کرد. هر رشته باید نشان دهنده‌ی یک جواب کامل برای مسئله باشد. گنجاندن اطلاعات اضافی در رشته باعث کاهش سرعت عمل الگوریتم می‌شود.

۲. جمعیت: الگوریتم ژنتیک برخلاف سایر روش‌ها در هر مرحله، از چند نقطه برای انجام عملیات جستجو استفاده می‌کند. رشته‌های مورد استفاده در هر مرحله یک جمعیت تشکیل می‌دهند. این جمعیت می‌تواند با جمعیت مرحله بعد

هم پوشانی داشته باشد. به این معنی که در هر مرحله، بعضی از اعضای جمعیت قبلی بدون تغییر وارد جمعیت جدید می‌شوند که در این صورت به الگوریتم، الگوریتم ژنتیک پایدار گفته می‌شود.

۳. تابع هدف یا برداش: تابعی از متغیرهای مورد نظر که معیار تعیین امتیاز کروموزوم یا رشته‌هاست.

۴. تابع جریمه: در حل مسائل بهینه سازی مقید، جواب‌ها علاوه بر تابع برازش باید در یک یا چند رابطه تساوی یا نامساوی هم صدق کنند، در این حالت به طریقی باید نقاطی که در قیود مسئله صدق نمی‌کنند را حذف کرد. دو روش برای حل این مشکل وجود دارد که عبارتند از اصلاح رشته و اصلاح تابع هدف. در روش اول، رشته به گونه‌ای تعریف می‌شود که نقاط غیرقابل قبول در جمعیت وارد نشوند و در روش دوم تابع هدف را باید طوری تعریف کرد که نقاط غیرقابل قبول امتیاز پایین بیاورند. روش دوم به روش جریمه معروف است. در این روش برای تابع هدف یک ضریب یا یک بخش اضافی در نظر گرفته می‌شود که باعث کاهش امتیاز نقاط غیرقابل قبول می‌شود.

## ۲-۶- کد گذاری

قبل از اینکه یک الگوریتم ژنتیک برای یک مسئله اجرا شود، باید یک روش برای کد کردن جواب‌ها به زبان کامپیوتر به کار رود. یکی از روش‌های معمول کد کردن به صورت رشته‌های باینری است. دو کروموزوم نشان داده شده در زیر به روش باینری کد شده‌اند.

Chromosome A 101100101100101011100101

Chromosome B 111111100000110000011111

یک روش مشابه دیگر، کد کردن راه حل‌ها در آرایه‌ای از اعداد صحیح یا اعشاری است که دوباره هر جایگاه یک جنبه از ویژگی‌ها را نمایش می‌دهد. این راه حل در مقایسه با قبلی پیچیده‌تر و مشکل‌تر است. سومین روش برای نمایش راه حل‌ها بکارگیری یک رشته از حروف است، که هر حرف دوباره نمایش دهنده یک خصوصیت از راه حل است. کروموزم‌های زیر با اعداد و حروف کد شده‌اند:

Chromosome A 1.2324 5.3243 0.4556 2.3293 2.4545

Chromosome B ABDJEIFJDHDIERJFDLDFLFEGT

خاصیت هر ۳ تایی این روش‌ها این است که آنها تعریف سازنده‌ای که تغییرات تصادفی در آنها ایجاد می‌کنند را آسان می‌سازند (مثلاً ۰ را به ۱ و برعکس یا اضافه یا کم کردن ارزش یک عدد یا تبدیل یک حرف به حرف دیگر) [5].

## ۲-۷- عملگرهای الگوریتم ژنتیک

۱. عملگر انتخاب ۲. عملگر قطع یا تزویج یا ترکیب ۳. عملگر جهش

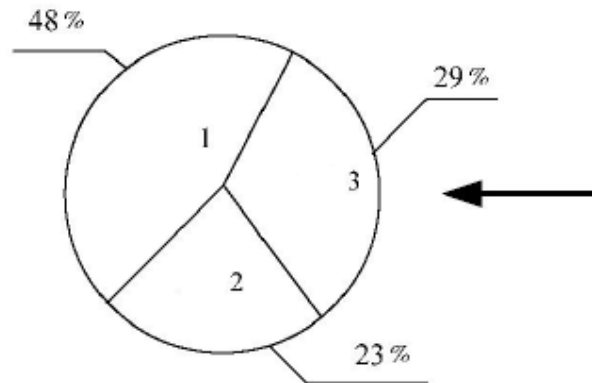
### ۲-۷-۱- انتخاب

همان طور که در طرح کلی الگوریتم ژنتیک مطرح گشت ابتدا کروموزوم‌ها انتخاب می‌شود تا والدین نسل بعد باشند فرآیندهای برش و جهش روی آن‌ها انجام گیرد حال ببینیم که این انتخاب والدین چگونه و بر چه معیاری استوار است. روش‌های مختلفی برای انتخاب وجود دارد که چند روش مهم در اینجا توضیح داده می‌شود:



## الف: انتخاب چرخ رولت

در این روش والدین با توجه به بالا بودن برازندگی انتخاب می‌شوند و در نتیجه بهترین کروموزوم‌ها شانس انتخاب شدن بیشتری دارند. در تصویر زیر همه کروموزوم‌های یک نسل نشان داده شده است. قسمت‌های مختلف این نمودار برای هر کروموزوم در نظر گرفته شده است همانطور که دیده می‌شود اندازه‌ی این قسمت‌ها با هم برابر نیست بلکه این اندازه‌ها به ارزشی که تابع برازندگی به هر کروموزوم می‌دهد وابسته است. (شکل ۲-۱)



شکل ۲-۱: قاعده کلی انتخاب چرخ رولت [6]

انتخاب در اینجا از یک نوع بازی الهام گرفته شده روش به این صورت است که گردونه را می‌چرخانیم و بعد از مدتی گردونه می‌ایستد و ما از قبل یک نشان در بیرون گردونه گذاشته‌ایم کروموزومی را که نشان می‌کند را ما انتخاب می‌کنیم آشکار است که کروموزومی که ارزش برازندگی بالایی دارد قسمت بزرگی از گردونه را به خود اختصاص داده است در نتیجه با توجه به علم آمار احتمال انتخاب شدن بیشتری دارد و دفعات بیشتری از انتخاب را به خود اختصاص می‌دهد.

## ب: انتخاب ترتیبی<sup>۱</sup>

روش انتخاب قبلی که توضیح داده شد روش خوبی است، ولی در حالتی که اختلاف برازندگی در کروموزوم‌ها زیاد باشد دچار مشکل می‌شود، مثلاً اگر ارزش بهترین کروموزوم 90% باشد مجموع کروموزوم‌های دیگر شانس بسیار کمتری برای انتخاب شدن دارند. در شیوه‌ی انتخاب ترتیبی به این صورت عمل می‌کنیم که ابتدا جمعیت را مرتب می‌کنیم سپس به هر کروموزوم با توجه به برازندگی آن عددی اختصاص می‌دهیم مثلاً بدترین کروموزوم 1 کروموزوم ما قبل بدترین ۲ و الی آخر تا اینکه به بهترین کروموزوم N را می‌دهیم (N تعداد کروموزوم‌های نسل) با این نوع انتخاب همه کروموزوم‌ها احتمال انتخاب شدن دارند البته در این روش همگرایی بسیار آهسته اتفاق می‌افتد به خاطر اینکه اختلاف کروموزوم‌ها کاهش پیدا کرده است.

<sup>۱</sup> Rank Selection

### ج: انتخاب حالت یکنواخت<sup>۱</sup>

این روش شیوه ی مخصوصی برای انتخاب والدین نیست و ایده آل اصلی این روش برای انتخاب جمعیت یا نسل جدید این است که قسمت بزرگی از کروموزومها بتوانند برای نسل جدید حفظ شوند. روش انتخاب حالت یکنواخت در GA به این صورت عمل می کند که در همه ی تولیدها تعداد محدودی کروموزوم خوب (با Fitness بالا) برای ساختن فرزندان جدید انتخاب می شوند، و سپس تعدادی از کروموزوم های بعدی (با Fitness پایین) حذف می شوند و به جای آنها فرزندان جدید جایگذاری می شود. و نسل جدید به وجود آمده برای تولید جدید حفظ می شود.

### د: انتخاب مقیاس گذاری<sup>۲</sup>

به موازات افزایش متوسط برازش جامعه، سنگینی انتخاب بیشتر و جرئی تر می شود. این روش وقتی کاربرد دارد که مجموعه دارای عناصری باشد که عدد برازش بزرگی دارند و فقط تفاوت های کوچکی آنها را از هم تفکیک می کند.

### ه: انتخاب مسابقه ای<sup>۳</sup>

در این روش،  $n$  کروموزوم از بین جمعیت با روش گردونه شانس و یا به طور تصادفی انتخاب می شوند و برازنده ترین آنها برای تولید نسل بعد انتخاب می شود.  $n$  می تواند حداقل دو باشد. عدد موثر دیگر  $p$  است.  $p$  تعداد حوضچه های آمیزش است.  $n$  کروموزوم در  $p$  اتاق مختلف قرار می گیرند، سپس از بین آنها برازنده ترین ها (بسته به نوع روش تعریف برزندگی متفاوت خواهد بود) انتخاب می شوند. پیشنهاد می شود که  $n=2$  باشد و با زیاد کردن تعداد اتاقها (حوضچهها) فشار انتخاب بیشتر شود. یعنی بخت انتخاب افراد دارای برزندگی کمتر کاهش می یابد و بخت افراد برازنده تر زیادتر می شود.

### ۲-۷-۲- قطع، تزویج، ترکیب، آمیزش<sup>۴</sup>

بعد از تشکیل جمعیت جدید اعضای این مجموعه به صورت تصادفی، دو به دو با هم جفت می شود. این عملگر برای هر جفت از رشته ها به صورت تصادفی یک محل قطع در طول رشته انتخاب می کند و سپس قسمتی از این دو رشته را با هم مبادله می کند که حاصل این عمل دو رشته ی جدید با طول مساوی رشته های اولیه است. دو رشته ی اول را والدین و رشته های جدید را فرزندان می نامند.

---

<sup>1</sup> Steady State Selection

<sup>2</sup> Scaling Selection

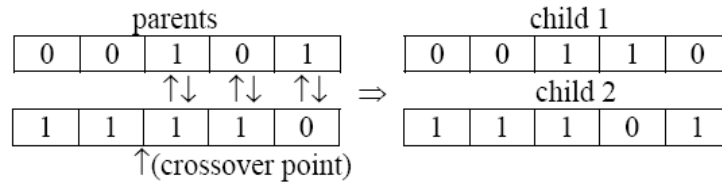
<sup>3</sup> Tournament Selection

<sup>4</sup> Crossover

انواعی از این عملگر در زیر آمده است:

### Crossover با نقطه منفرد:

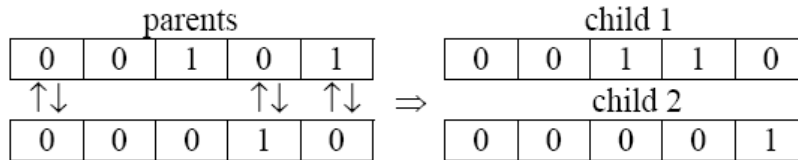
ابتدا نقطه‌ای که قطع در آنجا صورت می‌گیرد انتخاب می‌شود بعد در رشته‌ی باینری کروموزوم اول از آغاز رشته تا نقطه‌ی Crossover کپی می‌شود از رشته دوم از نقطه‌ی Crossover تا پایان کپی می‌شود و فرزند اول به وجود می‌آید و به همین ترتیب فرزند دوم، در شکل (۲-۲) مراحل ذکر شده قابل مشاهده است.



شکل (۲-۲) Crossover با نقطه منفرد [7]

### Crossover با دو نقطه:

ابتدا دو نقطه Crossover انتخاب می‌کنیم. برای تولید فرزند اول در رشته‌ی باینری از ابتدای کروموزوم والد اول تا اولین نقطه‌ی Crossover کپی می‌کنیم، بعد از رشته‌ی والد دوم تا نقطه دوم Crossover کپی می‌شود و ادامه را نیز از نقطه دوم Crossover در رشته‌ی اول کپی می‌کنیم. بنابراین فرزندان به صورت شکل (۳-۲) به وجود می‌آیند.



شکل (۳-۲) Crossover با دو نقطه [7]

**Uniform crossover:** بیت‌ها به صورت رندوم از والدین کپی می‌شوند. نمونه‌ای از این نوع ترکیب در زیر آمده است.

$$11001011 + 11011101 = 11011111$$

**Arithmetic crossover:** با توجه به قواعد و دستورات ریاضی خاصی فرزندها تولید می‌شوند، مثل and که در زیر نشان

داده شده است.

$$11001011 + 11011111 = 11001011$$

لازم به ذکر است که آمیزش معمولا بر روی همه زوج کروموزوم‌های انتخاب شده برای جفت‌گیری به کار برده نمی‌شود. معمولا احتمال آمیزش برای هر جفت کروموزوم بین ۰/۶ تا ۰/۹۵ در نظر گرفته می‌شود که به این عدد نرخ آمیزش گفته می‌شود که با PC نمایش داده می‌شود. در صورتی که بر روی یک زوج کروموزوم عمل آمیزش صورت نگیرد، فرزندان با تکرار نمودن والدین تولید می‌شوند.

