



الگوریتم ژنتیک در فضای تک و چندهدفه (مفاهیم و ابزارها)

نویسندگان

عضو هیئت علمی دانشگاه شاهد

دکتر مهدی بشیری

مهندس مجید جلیلی

انتشارات دانشگاه شاهد (۱۴۲)

سرشناسه	شیری، مهدی، ۱۳۵۶ -
عنوان و نام پدیدآور	الگوریتم ژنتیک در فضای تک و چندهدفه (مفاهیم و ابزارها) مهدی شیری و مجید جلیلی
مشخصات نشر	تهران: دانشگاه شاهد، مرکز چاپ و انتشارات، ۱۳۹۳.
مشخصات ظاهری	۴۲۲ ص: مصور، جدول و نمودار
شابک	۹۷۸ ۶۰۰ ۶۱۲۱-۳۵-۲
وضعیت فهرست نویسی	فیا.
یادداشت	کتابنامه: ۳۶۱ ۳۵۸.
موضوع	الگوریتم های ژنتیک
شناسه افزوده	جلیلی، مجید، ۱۳۶۴ -
شناسه افزوده	دانشگاه شاهد، مرکز چاپ و انتشارات
رده بندی کنگره	۱۳۹۳ ۷۲۵:۴۸۴ Q
رده بندی دیویی	۰۰۵۱
شماره کتابشناسی ملی	۳۶۰۸۳۵۱



الگوریتم ژنتیک در فضای تک و چندهدفه (مفاهیم و ابزارها)

نویسندگان	: مهدی شیری و مجید جلیلی
تعداد	: ۱۰۰۰ نسخه
نوبت چاپ	: اول ۱۳۹۳
قیمت	: ۱۳۹۰۰۰ ریال
نشانی ناشر	: تهران، ابتدای آزادراه خلیج فارس، روبه روی حرم مطهر امام خمینی (ره)، معاونت پژوهش و فناوری، مرکز چاپ و انتشارات دانشگاه شاهد

فهرست مطالب

صفحه

عنوان

مقدمه ۱

فصل اول : الگوریتم ژنتیک تک هدفه

۱-۱ مقدمه ۱

۱-۲ پیچیدگی مسائل ۱۱

۱-۳ چه وقت از الگوریتم ژنتیک استفاده کنیم؟ ۱۳

۱-۴ تفاوت های الگوریتم ژنتیک با سایر الگوریتم های متعارف ۱۳

۱-۵ اصطلاحات الگوریتم ژنتیک ۱۴

۱-۶ تشریح کلی از الگوریتم ژنتیک ۱۵

۱-۷ شبه کد الگوریتم ژنتیک ۱۷

۱-۸ مثالی ساده از الگوریتم ژنتیک ۱۸

۱-۹ نکات مهم در هنگام کار با الگوریتم ژنتیک ۲۱

۱-۱۰ کد کردن مقادیر (Coding - Representation) ۲۱

۱-۱۰-۱ کدگذاری باینری (Binary Encoding) ۲۳

۱-۱۰-۲ کدگذاری گری (Gray Encoding) ۲۷

۱-۱۰-۳ کدگذاری جایگشتی (Permutation Encoding) ۳۰

۱-۱۰-۴ کدگذاری ارزشی (Value Encoding) ۳۰

۱-۱۰-۵ کدینگ درختی (Tree Encoding) ۳۱

- ۱-۱۱ مسائل مربوط به کدینگ ۳۲
- ۱-۱۲ عملگر انتخاب (تکثیر یا تولید مثل) (Reproduction or Selection) ۳۵
- ۱-۱۲-۱ روش چرخه رولت (Roulette Wheel) ۳۷
- ۱-۱۲-۱-۱ مقیاس‌بندی خطی ۳۹
- ۱-۱۲-۱-۲ مقیاس‌بندی خطی پویا ۴۰
- ۱-۱۲-۱-۳ مقیاس‌بندی جداسازی زیگی ۴۰
- ۱-۱۲-۱-۴ مقیاس‌بندی توان ۴۱
- ۱-۱۲-۱-۵ مقیاس‌بندی لگاریتمی ۴۱
- ۱-۱۲-۱-۶ نرمال‌سازی ۴۱
- ۱-۱۲-۲ روش مسابقه (Tournament) ۴۱
- ۱-۱۲-۳ روش رتبه‌بندی (Rank) ۴۲
- ۱-۱۲-۴ روش حالت پایدار (Steady State) ۴۴
- ۱-۱۳ عملگرهای الگوریتم ژنتیک ۴۵
- ۱-۱۳-۱ عملگر تقاطع (ادغام یا ترکیب) (Cross Over) ۴۵
- ۱-۱۳-۱-۱ انواع روش‌های تقاطع ۴۸
- ۱-۱۳-۱-۲ نرخ تقاطع (Crossover Rate) ۵۰
- ۱-۱۳-۲ عملگر جهش (Mutation) ۵۰
- ۱-۱۳-۲-۱ نرخ جهش (Mutation Rate) ۵۳
- ۱-۱۴ مفهوم شماتا ۵۴
- ۱-۱۵ شرط توقف الگوریتم ژنتیک ۵۷
- ۱-۱۶ نقاط قوت الگوریتم ژنتیک ۵۸
- ۱-۱۷ نقاط ضعف الگوریتم ژنتیک ۶۱
- ۱-۱۸ همگرایی الگوریتم ژنتیک ۶۲

۱۹-۱ الگوریتم ژنتیک پیوندی.....	۶۳
۲۰-۱ مسئله رستوران همبرگر.....	۶۴
۲۱-۱ مسئله مربع وقفی.....	۷۱
۲۲-۱ کاربردهای الگوریتم ژنتیک.....	۷۴
۲۳-۱ توابع آزمون.....	۸۳
۲۴-۱ الگوریتم ژنتیک باینری و پیوسته.....	۹۸
۱-۲۴-۱ شرح الگوریتم ژنتیک باینری.....	۱۰۰
۲-۲۴-۱ شرح الگوریتم ژنتیک پیوسته.....	۱۰۹
تمرین.....	۱۱۵
مراجع.....	۱۱۶

فصل دوم: حل مسئله QAP با استفاده از الگوریتم ژنتیک

۱-۲ معرفی مسئله QAP.....	۱۱۹
۲-۲ حل مسئله QAP با استفاده از الگوریتم ژنتیک.....	۱۲۷
۱-۲-۲ شرح الگوریتم اول.....	۱۲۹
۲-۲-۲ شرح الگوریتم دوم.....	۱۴۱
۳-۲-۲ مقایسه نتایج حاصل از الگوریتم‌ها.....	۱۵۶
۳-۲ حل مسئله QAP با روش 2opt.....	۱۶۴
۴-۲ حل مسئله QAP با روش Greedy 2opt.....	۱۶۸
۵-۲ حل مسئله QAP با روش VNZ.....	۱۷۱
تمرین.....	۱۷۷
مراجع.....	۱۷۸

فصل سوم: تعیین سطوح پارامترهای الگوریتم ژنتیک با استفاده از طراحی آزمایش‌ها

۱۷۹	۳-۱ مقدمه
۱۸۰	۳-۲ تعیین سطوح پارامترهای الگوریتم ژنتیک
۱۸۶	۳-۳ تحلیل نتایج متغیر پاسخ هزینه (Cost)
۱۸۹	۳-۴ تحلیل نتایج متغیر پاسخ زمان (Time)
۱۹۵	تمرین
۱۹۶	مراجع

فصل چهارم: حل مسئله پرموتاسیون با استفاده از الگوریتم

۱۹۷	۴-۱ معرفی مسئله پرموتاسیون
۲۰۰	۴-۲ شرح الگوریتم پرموتاسیون
۲۰۵	۴-۳ اجرای الگوریتم پرموتاسیون
۲۰۹	۴-۴ رویکرد ژنتیک در حل مسئله
۲۲۰	۴-۵ تحلیل نتایج حاصل از اجرای الگوریتم
۲۲۲	تمرین
۲۲۲	مراجع

فصل پنجم: جعبه ابزار الگوریتم ژنتیک در متلب

۲۲۳	۵-۱ مقدمه
۲۲۶	۵-۲ شرح جعبه ابزار الگوریتم ژنتیک در متلب
۲۲۶	۵-۲-۱ Population
۲۲۹	۵-۲-۲ Fitness scaling options
۲۳۱	۵-۲-۳ Selection options

۲۳۳	 Reproduction options ۵-۲-۴
۲۳۴	 Mutation options ۵-۲-۵
۲۳۷	 Crossover options ۵-۲-۶
۲۴۱	 Migration options ۵-۲-۷
۲۴۲	 Algorithm Settings ۵-۲-۸
۲۴۳	 Multiobjective options ۵-۲-۹
۲۴۳	 Hybrid Function options ۵-۲-۱۰
۲۴۵	 Stopping Criteria options ۵-۲-۱۱
۲۴۶	 Plot options ۵-۲-۱۲
۲۴۸	 Output Function options ۵-۲-۱۳
۲۴۹	 Display to Command Window options ۵-۲-۱۴
۲۵۰	 ۵-۳ چگونگی تعریف تابع در جعبه ابزار متلب به همراه مثال
۲۶۲	 تمرین

فصل ششم: الگوریتم ژنتیک چندهدفه

۲۶۳	 ۶-۱ مقدمه
۲۶۹	 ۶-۲ پاسخ‌های غیرمسلط
۲۷۸	 ۶-۳ بهینگی محلی پارتو
۲۷۹	 ۶-۴ بهینگی سراسری پارتو
۲۷۹	 ۶-۵ الگوریتم ژنتیک در مسائل چندهدفه
۲۸۷	 ۶-۶ رویکردهای به کاررفته برای الگوریتم ژنتیک در مسائل چندهدفه
۲۸۸	 ۶-۶-۱ دسته اول: روش‌های غیر پارتو (شناسایی یک نقطه غیرمسلط)
۲۸۹	 ۶-۶-۱-۱ روش کلاسیک
۲۹۱	 ۶-۶-۱-۲ روش RWGA
۲۹۴	 ۶-۶-۱-۳ روش VEGA

- ۶-۶-۱-۴ روش لکسیکوگراف ۲۹۵
- ۶-۶-۱-۵ روش ϵ -constraint ۲۹۶
- ۶-۶-۱-۶ روش‌های مبتنی بر حرکت به سمت بردار هدف ۲۹۷
- ۶-۶-۲ دسته دوم: روش‌های مبتنی بر مرز پارتو (شناسایی مجموعه نقاط غیر مسلط) ۳۰۱
- ۶-۶-۲-۱ روش‌های ایجاد همگرایی به مرز برتر (روش‌های رتبه‌بندی) ۳۰۱
- ۶-۶-۲-۱-۱ روش رتبه‌بندی گلدبرگ ۳۰۲
- ۶-۶-۲-۱-۲ روش رتبه‌بندی فونسیسا و فلمینگ ۳۰۴
- ۶-۶-۲-۱-۳ روش رتبه‌بندی SPEA ۳۰۵
- ۶-۶-۲-۱-۴ روش رتبه‌بندی ARDS ۳۰۷
- ۶-۶-۲-۲ روش‌های ایجاد پراکندگی در مرز برتر ۳۰۸
- ۶-۶-۲-۲-۱ اشتراک کارایی ۳۰۹
- ۶-۶-۲-۲-۲ روش چگالی سلول محور ۳۱۲
- ۶-۶-۲-۲-۳ فاصله ازدحام ۳۱۳
- ۶-۶-۲-۳ الگوریتم‌های دسته دوم (روش‌های مبتنی بر مرز پارتو) ۳۱۶
- ۶-۶-۲-۳-۱ روش MOGA ۳۱۶
- ۶-۶-۲-۳-۲ روش PESA ۳۱۸
- ۶-۶-۲-۳-۳ روش RDGA ۳۲۰
- ۶-۶-۲-۳-۴ روش NSGA II ۳۲۱
- ۶-۶-۲-۳-۵ روش SPEA-II ۳۲۶
- ۶-۷ الیتیسیم ۳۲۷
- ۶-۷-۱ راهکارهای دسته اول (الیتیسیم درون جمعیت داخلی) ۳۲۸
- ۶-۷-۲ راهکارهای دسته دوم (الیتیسیم درون جمعیت خارجی) ۳۲۸
- ۶-۷-۲-۱ رویه الیتیسیم در روش SPEA ۳۳۰

۳۳۱	۸-۶ حل یک مسئله چندهدفه با روش VEGA
۳۳۵	۹-۶ کد الگوریتم VEGA در متلب
۳۵۵	تمرین
۳۵۸	مراجع
۳۶۳	پیوست ۱ کد متلب مربوط به مثال مربع وقفی
	پیوست ۲ نتایج حاصل از اجرای الگوریتم‌های 2opt و Greedy 2opt برای مسئله QAP
۳۷۶	
	پیوست ۳ کد متلب الگوریتم اول برای مسئله پرموتاسیون (مطابق با کد متلب نوشته شده برای مسئله QAP)
۳۸۸	

مقدمه

سیستم‌های پیچیده درون طبیعت، مسائل بسیاری را پیش روی ما قرار می‌دهند. مسیر کامیون‌های حمل‌ونقل باید تعیین شود، ابزارها یا نقاط فروش محصول‌ها باید جایابی شوند، شبکه‌های ارتباطی باید طراحی شوند، کانفیگ‌ها باید بازگیری شوند، رابط‌های رادیویی می‌بایست دارای فرکانس مناسب باشند، مواد اولیه از قبیل «چوب، فلز، شیشه و چرم باید به اندازه‌های لازم بریده شوند و ...» از این دست مسائل بی‌شمارند. نظریه پیچیدگی به ما می‌گوید که مسائل، ترکیب‌هایی از چند جمله‌ای¹ نیستند، بدین معنی که این مسائل در اندازه‌های کاربردی و عملی خود به قدری بزرگ هستند که نمی‌توان پاسخ بهینه آنها را در مدت زمان قابل پذیرش به دست آورد؛ باوجوداین، این مسائل باید حل شوند، بنابراین چاره‌ای نیست که به پاسخ‌های زیر بهینه بسنده کرد، به گونه‌ای که دارای کیفیت قابل قبول بوده، در مدت زمان قابل پذیرش به دست آیند.

رویکردهای بسیاری برای طراحی پاسخ‌های باکیفیت، در مدت زمان قابل پذیرش پیشنهاد شده‌است. الگوریتم‌هایی هستند که می‌توانند یافتن پاسخ‌های خوب در فاصله‌ای مشخص از پاسخ بهینه را تضمین کنند که به آنها الگوریتم‌های تقریبی می‌گویند؛ الگوریتم‌هایی دیگر هستند که تضمین می‌دهند با احتمال بالا پاسخ نزدیک بهینه تولیدکنند که به آنها الگوریتم‌های احتمالی گفته می‌شود.

جدای از این دو دسته، می‌توان الگوریتم‌های حل را به سه دسته کلی تقسیم‌بندی کرد:

¹ Polynomial

- ۱- روش‌های دقیق^۱: این روش‌ها برگرفته از اصول اثبات‌شده ریاضی هستند و دستورالعملی مشخص را به منظور رسیدن به پاسخ مسئله طی می‌کنند.
- ۲- روش‌های ابتکاری^۲: در این روش‌ها برای حل مسائل از ابتکار استفاده شده است و هیچ تضمینی در ارائه پاسخ ندارند اما براساس شواهد و سوابق نتایج آنها، به‌طور متوسط بهترین تقابل کیفیت و زمان حل را برای مسئله مورد بررسی به همراه داشته‌اند؛ مانند روش نیوتن در یافتن ریشه‌های یک معادله یا روش HAP^۳ به منظور یافتن نقطه بهینه در مسئله تعیین محل^۴.
- ۳- روش‌های فراابتکاری^۵: برای بهبود الگوریتم‌های ابتکاری از اواسط دهه ۱۹۷۰، موجی تازه از رویکردها آغاز شد؛ این رویکردها شامل الگوریتم‌هایی است که به‌صراحت یا به‌صورت ضمنی، تقابل میان ایجاد تنوع جستجو (وقتی علائمی وجود دارند که جستجو به سمت مناطق بد فضای جستجو می‌رود) و تشدید جستجو (با این هدف که بهترین پاسخ در منطقه مورد بررسی را پیدا کنند) را مدیریت می‌کنند؛ این الگوریتم‌ها بر هدایت هیوریستیک یک الگوریتم سازنده یا جستجوی محلی متمرکز می‌شوند به گونه‌ای که آن الگوریتم بتواند بر شرایط حساس (مانند فرار از بهینه محلی) غلبه کند؛ به این الگوریتم‌ها، فراابتکاری گفته می‌شود و از جمله ویژگی‌های این الگوریتم‌ها، می‌توان به موارد زیر اشاره کرد:
- ۱- وجود خرق عادت: به این معنی که ممکن است در زمان اجرای الگوریتم، عملی را خلاف عادت انجام دهیم برای نمونه به جای استفاده از بهترین پاسخ‌های موجود برای یافتن پاسخ‌های بهتر، از پاسخ‌های نه چندان خوب یا حتی بد استفاده کنیم؛ یکی از علل انجام این کار می‌تواند افزایش تصادفی‌سازی در زمان جستجو باشد.

¹ Exact

² Heuristic

³ Hyperboloid Approximation Procedure

⁴ Location

⁵ Metaheuristic

۲- تصادفی سازی: دیگر ویژگی این روش ها، وجود عنصر تصادفی سازی است؛ این امر در عدم به دام افتادن در نقطه بهینه محلی، بسیار تأثیر دارد؛ در واقع، یکی از علل عدم کارایی روش های دقیق در یافتن نقاط بهینه سراسری، عدم داشتن عنصر تصادفی سازی است.

۳- منطق کلان: هرچند که این روش ها از عامل تصادفی سازی بهره می برند، این امر به این معنی نیست که فقط روش هایی برای جستجوی تصادفی فضای حل هستند؛ در واقع در نهان هریک از این روش ها منطقی کلان، مستور است و هریک از این روش ها از یک واقعیت طبیعی الهام گرفته شده اند.

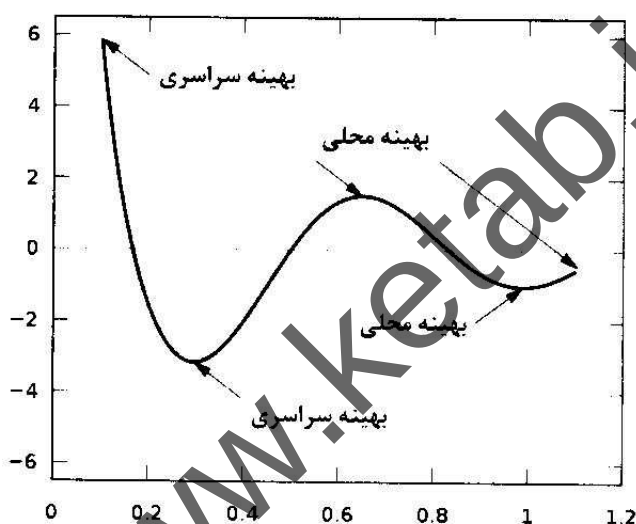
۴- الهام گرفته از طبیعت: ایده اصلی تمامی روش های فراابتکاری برگرفته از طبیعت است؛ این طبیعت است که به پدید آورندگان این روش ها، شیوه مناسب رسیدن به یک پاسخ بهینه یا نزدیک به بهینه را آموخته است.

در جدول ۱، ویژگی های این روش ها مقایسه شده است؛ در این جدول، چهار عامل دقت، سرعت، منطبق جزئی و احتمال به دام افتادن در نقطه بهینه محلی برای روش های دقیق، ابتکاری و فراابتکاری مقایسه شده است؛ برای نمونه، همان طور که ملاحظه می فرمایید، روش های دقیق، دقتی بیشتر نسبت به روش های ابتکاری و روش های ابتکاری، دقتی بیشتر نسبت به روش های فراابتکاری دارند.

جدول ۱. مقایسه روش های دقیق، ابتکاری و فراابتکاری

روش های حل	دقت	سرعت	منطق جزئی	احتمال به دام افتادن در نقطه بهینه محلی
دقیق	↑	↓	↑	↑
ابتکاری				
فراابتکاری				

درواقع، یکی از ایرادهایی که می‌توان علاوه بر زمان‌بر بودن به روش‌های دقیق گرفت، به‌دام‌افتادن در بهینه‌های محلی^۱ است، بدون اینکه هیچ شانس برای فرار از آنها داشته باشند. نقطه بهینه محلی، نقطه است که تنها در بخشی از فضای پاسخ مسئله بهینه است. به‌دام‌افتادن در نقاط بهینه محلی سبب می‌شود که الگوریتم از رسیدن به پاسخ‌های سراسری^۲ که پاسخ بهینه مسئله در سراسر فضای پاسخ است بازماند. در شکل ۱، پاسخ‌های محلی و سراسری روی یک مسئله به‌نمایش درآمده است.



شکل ۱. نقاط بهینه سراسری و محلی

هیوریستیک‌ها عبارت‌ند از: معیارها، روش‌ها یا اصولی برای تصمیم‌گیری میان چندین خط‌مشی و انتخاب اثربخش‌ترین آنها برای دستیابی به اهداف مورد نظر. هیوریستیک‌ها نتیجه برقراری اعتدال میان دو نیازند: نیاز به ساخت معیارهای ساده و هم‌زمان توانایی تمایز درست میان انتخاب‌های خوب و بد. یک هیوریستیک می‌تواند حسابی سرانگشتی باشد که برای هدایت یک

^۱ Local Optimum

^۲ Global Optimum

دسته از اقدام‌ها به کار می‌رود؛ برای نمونه، روشی مشهور برای انتخاب طالبی رسیده عبارت‌است از فشار دادن محل اتصال به ریشه از یک طالبی نامزد و سپس بوکردن آن محل. اگر بوی آن محل، مانند بوی داخل طالبی باشد، آن طالبی به احتمال زیاد رسیده‌است؛ این قاعده سرانگشتی نه تضمین می‌کند که طالبی‌های رسیده به عنوان نامزد انتخاب شوند و نه تضمین می‌کند که طالبی‌های رسیده آزمایش شده، رسیده تشخیص داده شوند؛ اما به هر حال، این روش، اثربخش‌ترین روش شناخته شده‌است؛ به عنوان مثالی دیگر از استفاده هیوریستیک‌ها، یک استاد بزرگ شطرنج را در نظر بگیرید که با انتخاب میان چندین حرکت ممکن، مواجه شده‌است؛ وی ممکن است تصمیم بگیرد که حرکتی خاص، اثربخش‌ترین حرکت خواهد بود زیرا موقعیتی فراهم می‌آورد که به نظر می‌رسد بهتر از موقعیت‌های حاصل از حرکت‌های دیگر باشد؛ به کارگیری این معیار چنین می‌نماید که خیلی ساده‌تر از تعیین دقیق حرکت یا حرکاتی خواهد بود که حریف را به مات شدن مجبور کند؛ این واقعیت که استادان بزرگ شطرنج، همواره پیروز بازی نخواهند بود، نشان‌دهنده این است که هیوریستیک‌های آنها انتخاب اثربخش‌ترین حرکت را تضمین نمی‌کند؛ در نهایت، وقتی از آنها خواسته می‌شود که هیوریستیک خود را تشریح کنند آنها فقط توصیفی ناقص از مجموعه‌ای قواعد ارائه می‌دهند و به نظر خود آنها، انجام آن قواعد از توصیف آن ساده‌تر است.

خاصیت هیوریستیک‌های خوب این است که ابزاری ساده برای تشخیص خط‌مشی‌های بهتر ارائه دهند و در حالی که به صورت شرط لازم، تشخیص خط‌مشی‌های اثربخش را تضمین نمی‌کنند، اغلب به صورت شرط کافی، این تضمین را فراهم می‌آورند. بیشتر مسائل پیچیده به ارزیابی تعداد انبوهی از حالت‌های ممکن برای تعیین پاسخی دقیق نیازمندند. گاهی زمان لازم برای یافتن یک پاسخ دقیق، اغلب بیشتر از طول عمر یک انسان است. هیوریستیک‌ها با استفاده از روش‌هایی که به ارزیابی‌های کمتر نیازمندند، پاسخ‌هایی در محدودیت‌های زمانی قابل قبول ارائه می‌کنند که دارای نقشی اثربخش در حل چنین مسائلی است.

از میان الگوریتم‌های فراابتکاری می‌توان به موارد زیر اشاره کرد:

- شبیه‌سازی تبرید (Simulated Annealing (SA)

- جستجوی ممنوع (Tabu Search (TS)
- الگوریتم‌های ژنتیک (Genetic Algorithms (GA)
- شبکه‌های عصبی مصنوعی (Artificial Neural Networks (ANN)
- الگوریتم مورچگان (Ant Colony Optimization (ACO)

مراجع ۹ و ۱۰ از نویسندگان کتاب نیز کاربردی از الگوریتم ژنتیک در مسئله QAP و تصمیم‌گیری چندمعیاره^۱ را نشان می‌دهند؛ این دو مورد در این کتاب با تفصیل بررسی می‌شوند. در مقاله ۷ با استفاده از GFD^۲، پارامترهای مناسب برای مسئله QAP در روش جستجوی ممنوع تعیین شده است؛ سپس الگوریتم به‌کاررفته طی یک مثال شرح داده شده است. در مقاله ۱۱ مروری بر روش‌های هیوریستیک و متاهیوریستیک مطرح شده برای مسئله QAP، صورت گرفته است؛ در این مقاله نشان داده شده است که از میان روش‌های هیوریستیک 2opt و Greedy 2opt و Greedy 3opt و 3opt بررسی شده، روش 3opt، نتایجی بهتر را دربردارد؛ همچنین از میان روش‌های TS، SA و PSO بررسی شده در این مقاله به‌عنوان روش‌های متاهیوریستیک، روش TS در مدت زمانی کمتر، نتیجه‌ای بهتر دارد.

مرجع ۱ نیز می‌تواند در یادگیری روش ACO استفاده شود؛ همچنین مرجع ۲ می‌تواند ابزاری مفید برای آشنایی با سایر الگوریتم‌های متاهیوریستیک و کاربرد آنها در سیستم‌های صنعتی و مسائل مکان‌یابی باشد.

در کتاب پیش رو به معرفی مبانی و قدم‌های الگوریتم ژنتیک پرداخته شده، سپس تعدادی از کاربردهای این الگوریتم در حل مسائل مختلف بیان شده است. تمامی الگوریتم‌ها در نرم‌افزار متلب کدنویسی شده‌اند و به دلیل جنبه آموزشی کتاب، تمامی کدهای نوشته شده برای الگوریتم‌ها شرح داده شده‌اند به‌طوری‌که امید است خواننده پس از مطالعه آنها توانایی لازم برای استفاده از این الگوریتم را در مسئله مورد نظر خود و کدنویسی آن را به‌خوبی فراگیرد. لازم به اشاره است که

^۱ Multi Criteria Decision Making (MCDM)

^۲ General Factorial Design

به منظور فهم کدهای نوشته شده و همچنین کدنویسی در متلب، آشنایی مقدماتی با نرم افزار متلب، ضروری است. توصیه می شود در شروع کار به جای خواندن کتاب های حجیم «آشنایی با متلب» که پس از مدتی، سبب دلزدگی خواننده می شوند، از جزوه های کاربردی و کم حجمی که به وفور در اینترنت یافت می شوند، استفاده کنید؛ با این حال باید توجه داشت که بهترین مرجع برای یادگیری نرم افزار متلب، استفاده از Help آن است.

برای استفاده از کتاب توصیه می شود که ابتدا، مبانی الگوریتم ژنتیک که در فصل اول آمده است به خوبی فراگیری شده، سپس بسته به مسئله مورد نظر خود، یک شبه کد مناسب برای آن بنویسید. به کاربردن راهکار مناسب در هنگام کار با الگوریتم ژنتیک، اهمیتی بسزا دارد؛ منظور از راهکاری، فرایند جستجوی الگوریتم در فضای حل است؛ این فرایند خود به چندین بخش کوچک تر تقسیم می شود که لازم است در هر مورد، سیاستی درست اتخاذ شود.

اهمیت این امر در مسائل به کار رفته در این کتاب نشان داده شده است. به طور کلی، مهم ترین بخش الگوریتم ژنتیک، معرفی راهکاری مناسب برای جستجو در فضای حل است. یک راهکار مناسب، تأثیری فزاینده در نزدیکی بیشتر به پاسخ بهینه، همگرایی و کاهش زمان حل دارد. مهم ترین بخش در یک راهکار، به کارگیری و تعریف مناسب عملگرهای الگوریتم ژنتیک است چراکه وظیفه جستجوی فضای حل بر دوش این عملگرهاست. همان طور که در فصل اول فراخواهید گرفت، الگوریتم ژنتیک، عملگرهایی فراوان برای جستجوی فضای حل دارد؛ این هنر طراح است که با استفاده مناسب از این عملگرها یا ترکیب آنها یا حتی معرفی عملگری جدید، بسته به مسئله مورد نظر خود، توانایی الگوریتم را در حل مسئله افزایش دهد.

توصیه می شود پس از نگارش شبه کد، آن را به صورت دستی، چند بار حل کنید. بدین معنی که مراحل الگوریتم را چند بار انجام دهید؛ این امر به عیب یابی الگوریتم و بهینه سازی آن، بسیار کمک می کند؛ برای نمونه، ممکن است الگوریتم در هر تکرار خود، کارهایی تکراری را انجام دهد که موجب صرف زمان شود و بتوان با انجام تغییرهایی در الگوریتم، این معضل را حذف کرد. در فصل

دوم کتاب، پس از معرفی مسئله تخصیص درجه دوم^۱ که از دسته مسائل مکان‌یابی است، الگوریتم ژنتیک پیشنهادی برای حل آن ارائه و کدهای نوشته‌شده برای آن تشریح شده‌است؛ با مطالعه این فصل، آشنایی عملی درباره چگونگی حل یک مسئله با الگوریتم ژنتیک را کسب خواهید کرد.

بسته به نوع مسئله و راهکار به‌کاررفته به‌یقین، الگوریتم پیشنهادی دارای تعدادی پارامتر کنترلی خواهد بود که با تغییر آنها الگوریتم به پاسخ‌هایی متفاوت خواهد رسید؛ بنابراین لازم است تحلیل دقیق روی این پارامترها صورت گیرد؛ این تحلیل می‌تواند به‌صورت سعی و خطا باشد بدین معنی که تأثیر تغییر این پارامترها بر پاسخ مسئله بررسی شده، درنهایت، طراح، دستورالعملی برای اندازه پارامترها ارائه دهد؛ شاید این روش، ساده به‌نظریاید اما روشی قابل‌اتکا نیست و نتایج به‌دست‌آمده از طریق این روش، قابل‌تجزیه و تحلیل آماری نیستند؛ روش دیگر، این است که با استفاده از مبانی علم آمار، به انجام آزمایشی آماری دست‌زده، دستورالعملی معتبر با توجه به نتایج به‌دست‌آمده ارائه دهیم. در فصل سوم کتاب با استفاده از مبانی طراحی آزمایش‌ها (DOE)^۲، تحلیلی جامع روی یکی از مسائل حل‌شده صورت گرفته‌است.

در فصل چهارم نیز از الگوریتم ژنتیک برای حل یکی از مسائل تصمیم‌گیری چندمعیاره^۳ به نام پرموتاسیون^۴ استفاده شده‌است.

البته گاهی می‌توان یک مسئله را بدون نوشتن یک کد در نرم‌افزار متلب حل کرد. نرم‌افزار متلب در جعبه ابزار^۵ خود، ابزاری به نام الگوریتم ژنتیک دارد که می‌توان از همین ابزار نیز برای حل مسئله مورد نظر استفاده کرد؛ در فصل پنجم کتاب، این ابزار معرفی شده‌است و در انتهای فصل، مثالی با استفاده از این جعبه ابزار حل شده‌است.

^۱ Quadratic Assignment Problem (QAP)

^۲ Design of Exprimnt

^۳ Multi Criteria Decision Making (MCDM)

^۴ Permutation

^۵ Toolbox

همان‌طور که مطرح شد، در فصل چهارم یکی از مسائل تصمیم‌گیری چندمعیاره با استفاده از الگوریتم ژنتیک، مورد بررسی قرار گرفته است. گاهی ممکن است در برابر مسائل چندمعیاره (مسائل چندپاسخه)¹ مجبور شویم از روش‌های حل چندپاسخه استفاده کنیم؛ در این حالت، نیاز به استفاده از الگوریتم ژنتیک در حالت چندپاسخه احساس می‌شود. در فصل ششم، این دسته از مسائل و روش‌هایی که برای حل آنها در الگوریتم ژنتیک معرفی شده‌اند، مورد بررسی قرار گرفته‌اند؛ در پایان این فصل، یکی از روش‌های معرفی شده برای مسائل چندپاسخه در نرم‌افزار متلب کد شده است. در پایان لازم به اشاره است که این کتاب نمی‌تواند از نقص و کمبود، خالی باشد؛ بنابراین نویسندگان آن از هر گونه نظر، پیشنهاد و انتقادی، صمیمانه استقبال می‌کنند.

Bashiri@shahed.ac.ir

¹ Multi Response Problem