



دانشگاه تهران

پردیس دانشکده‌های فنی
دانشکده مهندسی برق و کامپیوتر

عنوان:

توسعه الگوریتم بهینه‌سازی اجتماعی و بررسی کارایی آن

نگارش:

اسماعیل آتش‌پز گرگری

اساتید راهنما: دکتر بابک نجار اعرابی و دکتر کارو لوکس

استاد مشاور: دکتر مجید نیلی احمدآبادی

پایان‌نامه برای دریافت درجه کارشناسی ارشد

در

رشته مهندسی برق-گرایش کنترل

مهر ماه ۱۳۸۷

چکیده

الگوریتم های بهینه سازی الهام گرفته از طبیعت به عنوان روشهای هوشمند بهینه سازی در کنار روش های کلاسیک موفقیت خوبی از خود نشان داده اند. از جمله این روش ها می توان به الگوریتم های ژنتیک^۱ (الهام گرفته از تکامل بیولوژیکی انسان و سایر موجودات)، بهینه سازی کلونی مورچه ها^۲ (بر مبنای حرکت بهینه مورچه ها) و روش بازپخت شبیه سازی شده^۳ (با الهام گیری از فرایند تبرید فلزات) اشاره نمود. این روش ها در حل بسیاری از مسائل بهینه سازی در حوزه های مختلفی چون تعیین مسیر بهینه عامل های خودکار، طراحی بهینه کنترل کننده برای پروسه های صنعتی، حل مسائل عمده مهندسی صنایع همانند طراحی چیدمان بهینه برای واحدهای صنعتی، حل مسائل صف و نیز در طراحی عامل های هوشمند استفاده شده اند.

الگوریتم های بهینه سازی معرفی شده، به طور عمده الهام گرفته از فرایندهای طبیعی می باشند و در ارائه این الگوریتم ها به سایر نمودهای تکامل انسانی توجهی نشده است. در این نوشتار الگوریتم جدیدی برای بهینه سازی مطرح می شود که نه از یک پدیده طبیعی، بلکه از یک پدیده اجتماعی – انسانی الهام گرفته است. بطور ویژه این الگوریتم به فرایند استعمار، به عنوان مرحله ای از تکامل اجتماعی – سیاسی بشر نگریسته و با مدل سازی ریاضی این پدیده تاریخی، از آن به عنوان منشأ الهام یک الگوریتم قدرتمند در زمینه بهینه سازی بهره می گیرد. در مدت کوتاهی که از معرفی این الگوریتم می گذرد، از آن برای حل مسائل بسیاری در حوزه بهینه سازی استفاده شده است. طراحی چیدمان بهینه برای واحدهای صنعتی، آنتن های مخابراتی هوشمند، سیستم های پیشنهاددهنده هوشمند و نیز طراحی کنترل کننده بهینه برای سیستم های صنعتی شیمیایی تعدادی محدود از کاربردهای گسترده این الگوریتم در حل مسائل بهینه سازی می باشد.

¹ Genetic Algorithms (GA)

² Ant Colony Optimization (ACO)

³ Simulated Annealing (SA)

فهرست مطالب

عنوان صفحه

۱ مقدمه ۲-۱

| | |
|-----|---|
| ۱-۱ | هدف و اهمیت مسئله: ۲-۱ |
| ۲-۱ | الگوریتم توسعه داده شده: ۳-۱ |
| ۳-۱ | مزایای الگوریتم توسعه داده شده: ۵-۱ |
| ۴-۱ | ساختار پایان نامه ۶-۱ |

۲ بهینه سازی و روشهای موجود ۲-۲

| | |
|-------|---|
| ۱-۲ | انواع مسائل بهینه سازی ۴-۲ |
| ۲-۲ | روش های بهینه سازی کمینه جو ۷-۲ |
| ۱-۲-۲ | بهینه سازی تحلیلی ۷-۲ |
| ۲-۲-۲ | جستجوی خط ۸-۲ |
| ۳-۲-۲ | روش های نیوتونی ۹-۲ |
| ۴-۲-۲ | روش کاهشی نلدر - مید با اشکال غیر مرکب ۱۲-۲ |
| ۳-۲ | الگوریتم ژنتیک ۱۵-۲ |
| ۴-۲ | الگوریتم بازپخت شبیه سازی شده ۲۲-۲ |
| ۵-۲ | بهینه سازی گروه ذرات ۲۷-۲ |
| ۶-۲ | کلونی مورچه ها ۲۸-۲ |
| ۷-۲ | برنامه ریزی ژنتیک ۳۲-۲ |

۳ استراتژی بهینه سازی مبتنی بر تکامل اجتماعی-سیاسی ۲-۳

| | |
|-------|---|
| ۱-۳ | مقدمه ۲-۳ |
| ۲-۳ | مروری تاریخی بر پدیده استعمار ۳-۳ |
| ۱-۲-۳ | هند ۵-۳ |
| ۲-۲-۳ | مالزی ۶-۳ |

| | | |
|--|-------|------|
| هندوچین فرانسه..... | ۳-۲-۳ | ۶-۳ |
| هند شرقی (اندونزی)..... | ۴-۲-۳ | ۷-۳ |
| الگوریتم پیشنهادی..... | ۳-۳ | ۸-۳ |
| شکل دهی امپراطوری‌های اولیه..... | ۱-۳-۳ | ۱۰-۳ |
| مدل‌سازی سیاست جذب: حرکت مستعمره‌ها به سمت امپریالیست..... | ۲-۳-۳ | ۱۴-۳ |
| جابجایی موقعیت مستعمره و امپریالیست..... | ۳-۳-۳ | ۱۷-۳ |
| قدرت کل یک امپراطوری..... | ۴-۳-۳ | ۱۸-۳ |
| رقابت استعماری..... | ۵-۳-۳ | ۱۸-۳ |
| سقوط امپراطوری‌های ضعیف..... | ۶-۳-۳ | ۲۱-۳ |
| همگرایی..... | ۷-۳-۳ | ۲۲-۳ |
| مثال کاربردی..... | ۴-۳ | ۲۳-۳ |
| نتیجه‌گیری..... | ۵-۳ | ۲۸-۳ |
| توابع هزینه مورد استفاده..... | ۶-۳ | ۲۹-۳ |

۴ پیاده‌سازی‌های انجام شده..... ۲-۴

| | | |
|---|-------|------|
| استفاده از الگوریتم معرفی شده برای طراحی یک کنترل‌کننده PID بهینه..... | ۱-۴ | ۲-۴ |
| کنترل‌کننده PID..... | ۱-۱-۴ | ۳-۴ |
| طراحی کنترل‌کننده PID بهینه توسط الگوریتم رقابت استعماری..... | ۲-۱-۴ | ۵-۴ |
| نتیجه‌گیری..... | ۳-۱-۴ | ۹-۴ |
| استفاده از الگوریتم رقابت امپریالیستی برای طراحی کنترل‌کننده PID چند متغیره | ۲-۴ | |
| برای سیستم صنعتی ستون تقطیر..... | ۹-۴ | |
| مقدمه..... | ۱-۲-۴ | ۹-۴ |
| کنترل‌کننده PID برای فرایند چند متغیره..... | ۲-۲-۴ | ۱۰-۴ |
| نتایج شبیه‌سازی..... | ۳-۲-۴ | ۱۲-۴ |
| نتیجه‌گیری..... | ۴-۲-۴ | ۲۰-۴ |
| الگوریتم رقابت استعماری؛ ابزاری برای یافتن نقطه تعادل نش..... | ۳-۴ | ۲۰-۴ |
| یک بازی غیر خطی استاتیک ساده..... | ۱-۳-۴ | ۲۰-۴ |

| | | |
|------|--|-------|
| ۲۲-۴ | یک بازی با پیچیدگی بیشتر | ۲-۳-۴ |
| ۲۴-۴ | طراحی بهینه آنتهای آرایه‌ای | ۴-۴ |
| | استفاده از الگوریتم رقابت استعماری برای شناسایی ویژگی مواد از آزمون فرورفتگی | ۵-۴ |
| ۲۶-۴ | | |
| ۲۶-۴ | مقدمه | ۱-۵-۴ |
| ۲۷-۴ | توصیف مسئله معکوس | ۲-۵-۴ |
| ۲۸-۴ | حل مسئله معکوس توسط الگوریتم رقابت استعماری | ۳-۵-۴ |
| ۳۱-۴ | نتیجه‌گیری | ۴-۵-۴ |
| ۳۱-۴ | کنترل فازی اتومبیل | ۶-۴ |
| ۳۱-۴ | مدل اتومبیل | ۱-۶-۴ |
| ۳۳-۴ | نتایج | ۲-۶-۴ |

۶ خلاصه، نتیجه‌گیری و پیشنهادات ۲-۵

۶ مراجع ۲

پیوست: جعبه‌ابزار بهینه‌سازی تهیه‌شده ۲

| | |
|---|---------------------------|
| ۴ | بیان مسئله |
| ۴ | تنظیم پارامترهای الگوریتم |
| ۶ | نمایش نتایج |
| ۶ | انتخاب شیوه نمایش |

پیوست: مقالات مستخرج ۲

| | |
|---|----------------|
| ۲ | مقالات ژورنال |
| ۳ | مقالات کنفرانس |

فهرست شکل‌ها

| عنوان | صفحه |
|--|------|
| شکل ۱-۱: شمای کلی الگوریتم رقابت استعماری | ۴-۱ |
| شکل ۲-۱: حرکت مستعمرات به سمت امپریالیست (سیاست جذب) | ۴-۱ |
| شکل ۳-۱: شمای کلی رقابت استعماری | ۵-۱ |
| شکل ۱-۲: فرایند یا تابعی که بهینه‌سازی می‌شود. در بهینه‌سازی ورودی‌ها یا متغیرها به نحوی تغییر داده می‌شوند که خروجی مطلوب به دست بیاید. | ۳-۲ |
| شکل ۲-۲: مسأله‌ی حمل و نقل | ۵-۲ |
| شکل ۳-۲: نمودار کانتوری یک تابع درجه دو در فضای دو بعدی و مسیری که در بهینه‌سازی خطی برای یافته شدن کمینه‌ی این تابع طی شده است. حرکت در هر مسیر، تا زمانی ادامه دارد که منجر به کاهش مقدار تابع شود. در غیر این صورت، مسیر حرکت عوض خواهد شد. | ۹-۲ |
| شکل ۴-۲: الگوریتم بهینه کردن یک سیمپلکس در فضای دو بعدی. به شکل ۵-۲ توجه کنید. | ۱۳-۲ |
| شکل ۵-۲: یک سیمپلکس دو بعدی (مثلث) که مراحل مختلف الگوریتم شکل ۴-۲ بر روی آن انجام شده است، تا سیمپلکس بهتری ساخته شود. | ۱۴-۲ |
| شکل ۶-۲: نمونه‌ای از نحوه‌ی عملکرد الگوریتم کاهشی سیمپلکس بر روی یک تابع دو متغیره. | ۱۴-۲ |
| شکل ۷-۲: مراحل مختلف الگوریتم ژنتیک باینری | ۱۹-۲ |
| شکل ۸-۲: نحوه‌ی عملکرد الگوریتم بازپخت شبیه‌سازی شده | ۲۵-۲ |
| شکل ۹-۲: یک مسأله‌ی فروشنده‌ی دوره‌گرد با چهار شهر. | ۲۶-۲ |
| شکل ۱۰-۲: رفتار مورچه‌های آرژانتینی در آزمایش گاس و همکارانش | ۳۰-۲ |
| شکل ۱۱-۲: جمعیت کوچک توابع چندجمله‌ای در برنامه‌ریزی ژنتیک | ۳۲-۲ |
| شکل ۱-۳: اعمال سیاست جذب از طرف استعمارگران بر مستعمرات | ۵-۳ |
| شکل ۲-۳: فلوچارت الگوریتم پیشنهادی | ۹-۳ |
| شکل ۳-۳: اجزای اجتماعی سیاسی تشکیل دهنده یک کشور | ۱۱-۳ |
| شکل ۴-۳: چگونگی شکل‌گیری امپراطوری‌های اولیه. امپریالیست شماره ۱ قوی‌ترین امپراطوری را ایجاد کرده است و بیش‌ترین تعداد مستعمرات را دارد. | ۱۴-۳ |
| شکل ۵-۳: شمای کلی حرکت مستعمرات به سمت امپریالیست | ۱۵-۳ |

- شکل ۳-۶: حرکت واقعی مستعمرات به سمت امپریالیست ۳-۱۶
- شکل ۳-۷: کل امپراطوری، پس از تغییر موقعیت‌ها ۳-۱۷
- شکل ۳-۸: تغییر جای استعمارگر و مستعمره ۳-۱۷
- شکل ۳-۹: شمای کلی رقابت استعماری: امپراطوری‌های بزرگ‌تر، با احتمال بیشتری، مستعمرات امپراطوری‌های دیگر را تصاحب می‌کنند. ۳-۱۹
- شکل ۳-۱۰: سقوط امپراطوری ضعیف؛ امپراطوری شماره ۴، به علت از دست دادن کلیه مستعمراتش، دیگر قدرتی برای رقابت ندارد و باید از میان بقیه امپراطوری‌ها حذف شود. ۳-۲۲
- شکل ۳-۱۱: شبه کد مربوط به الگوریتم رقابت استعماری ۳-۲۳
- شکل ۳-۱۲: شمای کلی الگوریتم توسعه داده شده ۳-۲۴
- شکل ۳-۱۳: نمایش سه بعدی از تابع مسئله G_1 ۳-۲۵
- شکل ۳-۱۴: امپراطوری‌های اولیه؛ هر رنگ نمایش‌دهنده‌ی یک امپراطوری می‌باشد. ۳-۲۶
- شکل ۳-۱۵: امپراطوری‌ها در نسل ۱۰ام؛ ۴ تا از امپراطوری‌ها باقی مانده‌اند. ۳-۲۶
- شکل ۳-۱۶: امپراطوری‌ها در نسل ۳۰ام؛ تنها دو امپراطوری باقی مانده‌اند. ۳-۲۶
- شکل ۳-۱۷: امپراطوری‌ها در نسل ۳۳ام (همگرایی)؛ تنها یک امپراطوری واحد داریم. ۳-۲۷
- شکل ۳-۱۸: هزینه‌ی مینیمم و میانگین همه‌ی امپراطوری‌های مسئله G_1 بر حسب تکرار الگوریتم. ۳-۲۷
- شکل ۳-۱۹: هزینه‌ی مینیمم و میانگین همه‌ی امپراطوری‌های مسئله G_2 ، G_3 و G_4 ، بر حسب تکرار الگوریتم. ۳-۲۸
- شکل ۳-۲۰: هزینه‌ی مینیمم و میانگین همه‌ی امپراطوری‌های مسائل G_5 و G_6 بر حسب تکرار الگوریتم ۳-۲۸
- شکل ۴-۱: بلوک دیاگرام یک سیستم کنترل فیدبک ۴-۲
- شکل ۴-۲: زمان صعود (t_r)، زمان نشست (t_s)، بیشترین فراجهش (M_p) و انتگرال قدرمطلق خطا (مساحت ناحیه‌ی تیره رنگ). ۴-۴
- شکل ۴-۳: امپراطوری‌های اولیه در مسئله طراحی کنترل‌کننده PID ۴-۶
- شکل ۴-۴: امپراطوری‌ها در نسل ۳۰ام؛ دو امپراطوری ضعیف سقوط کرده و تنها ۴ امپراطوری باقی مانده‌اند. ۴-۶
- شکل ۴-۵: امپراطوری‌ها در نسل ۵۰ام؛ تنها ۲ تا از امپراطوری‌ها زنده‌اند و بصورت شدیدی برای حذف یکدیگر رقابت می‌کنند. ۴-۷

- شکل ۶-۶: هزینه‌ی مینیمم و میانگین امپریالیست‌ها بر حسب تکرار الگوریتم، در مسئله طراحی کنترل‌کننده PID. ۷-۶
- شکل ۷-۶: هزینه‌ی مینیمم جمعیت الگوریتم ژنتیک، بر حسب تکرار، در مسئله طراحی کنترل‌کننده PID. ۸-۶
- شکل ۸-۶: پاسخ پله‌ی سیستم به ورودی پله با استفاده از هر چهار کنترلر موجود در جدول ۶-۱. ۱۰-۶
- شکل ۹-۶: بلوک دیاگرام یک سیستم چند متغیره را به همراه کنترل‌کننده. ۱۰-۶
- شکل ۱۰-۶: شمای ساده‌ای از فرایند تقطیر شیمیایی. ۱۳-۶
- شکل ۱۱-۶: فلوچارت الگوریتم رقابت استعماری مورد استفاده برای طراحی یک کنترل‌کننده PID برای سیستم ستون تقطیر. ۱۵-۶
- شکل ۱۲-۶: هزینه مینیمم ICA و GA را بر حسب تکرار نسل. ۱۶-۶
- شکل ۱۳-۶: پاسخ سیستم ستون تقطیر به تاخیرهای متفاوت در ورودی پله (الف: ورودی اول، ب: ورودی دوم). ۱۸-۶
- شکل ۱۴-۶: قدر مطلق خطای خروجی فرایند ستون تقطیر به تأخیرهای مختلف در ورودی پله. (الف: قدر مطلق خطای ورودی اول، ب: قدر مطلق خطای ورودی دوم). ۱۹-۶
- شکل ۱۵-۶: شمای کلی الگوریتم رقابت استعماری، اعمال شده به مسئله تعیین نقطه تعادل نش. ناحیه داخل خط چین مربوط به الگوریتم و بقیه مربوط به مسئله می‌باشد. ۲۱-۶
- شکل ۱۶-۶: همگرایی هزینه برای الگوریتم ژنتیک. ۲۲-۶
- شکل ۱۷-۶: همگرایی هزینه برای الگوریتم رقابتی استعماری. ۲۲-۶
- شکل ۱۸-۶: نحوه پخش $(f1, f2)$ برای ۵۰۰۰۰ نقطه تصادفی. ۲۳-۶
- شکل ۱۹-۶: همگرایی الگوریتم ژنتیک. ۲۴-۶
- شکل ۲۰-۶: همگرایی الگوریتم معرفی شده. ۲۴-۶
- شکل ۲۱-۶: شمای کلی یک آنتن آرایه ای. ۲۴-۶
- شکل ۲۲-۶: نمودار همگرایی الگوریتم رقابت استعماری و الگوریتم ژنتیک. ۲۵-۶
- شکل ۲۳-۶: مقایسه LMS با الگوریتم ژنتیک. ۲۶-۶
- شکل ۲۴-۶: مقایسه الگوی الگوریتم ژنتیک و الگوریتم معرفی شده. ۲۶-۶
- شکل ۲۵-۶: فلوچارت ICA، اعمال شده به آنالیز معکوس آزمون فرو رفتگی. ۲۸-۶
- شکل ۲۶-۶: امپراطوری‌های اولیه. ۲۹-۶

| | |
|--|------|
| شکل ۴-۲۷: امپراطوری‌ها در نسل ۳۵..... | ۲۹-۴ |
| شکل ۴-۲۸: امپراطوری‌ها در نسل ۷۷ (همگرایی). جواب نهایی مسئله..... | ۲۹-۴ |
| شکل ۴-۲۹: هزینه میانگین و مینیمم همه‌ی امپریالیست‌ها بر حسب تکرار الگوریتم..... | ۲۹-۴ |
| شکل ۴-۳۰: مقایسه جواب‌های دقیق و نتایج الگوریتم معکوس برای یک دسته از نتایج آزمون فرورفتگی به دست آمده از FEA..... | ۳۰-۴ |
| شکل ۴-۳۱: مقایسه جواب‌های دقیق و نتایج الگوریتم معکوس برای یک دسته از نتایج آزمون فرورفتگی به دست آمده از FEA، در صفحه‌ی n, σ_y | ۳۰-۴ |
| شکل ۴-۳۲: مقایسه جواب‌های دقیق و نتایج الگوریتم معکوس برای یک دسته از نتایج آزمون فرورفتگی به دست آمده از FEA، در صفحه‌ی n, E | ۳۰-۴ |
| شکل ۴-۳۳: مقایسه جواب‌های دقیق و نتایج الگوریتم معکوس برای یک دسته از نتایج آزمون فرورفتگی به دست آمده از FEA، در صفحه‌ی σ_y, E | ۳۰-۴ |
| شکل ۴-۳۴: اتومبیل در جاده شیب‌دار..... | ۳۲-۴ |
| شکل ۴-۳۵: توابع عضویت مربوط به متغیر سرعت..... | ۳۳-۴ |
| شکل ۴-۳۶: توابع عضویت متغیر θ | ۳۳-۴ |
| شکل ۴-۳۷: توابع عضویت متغیر T..... | ۳۴-۴ |
| شکل ۴-۳۸: سرعت خروجی برای $V(0)=0$ و $\theta=1$ | ۳۴-۴ |
| شکل ۴-۳۹: سرعت خروجی برای $V(0)=0$ و $\theta=5$ | ۳۴-۴ |
| شکل ۴-۴۰: سرعت خروجی برای $V(0)=100$ و $\theta=7$ | ۳۴-۴ |
| شکل ۴-۴۱: سرعت خروجی برای $V(0)=100$ و $\theta=-6$ | ۳۴-۴ |
| شکل ۱-۰: شمای کلی جعبه ابزار گرافیکی تهیه شده بر مبنای الگوریتم رقابت استعماری..... | ۲ |
| شکل ۲-۰: صفحه انتخاب الگوریتم مورد نظر برای بهینه‌سازی..... | ۳ |
| شکل ۳-۰: صفحه راهنمای مربوط به الگوریتم ژنتیک..... | ۳ |
| شکل ۴-۰: بخش مربوط به بیان مسئله..... | ۴ |
| شکل ۵-۰: بخش مربوط به تنظیم پارامترهای الگوریتم رقابت استعماری..... | ۵ |
| شکل ۶-۰: الگوریتم رقابت استعماری در حال بهینه‌سازی تابع GUI_Testfunction..... | ۵ |
| شکل ۷-۰: انتخاب شیوه نمایش نتایج..... | ۶ |

فهرست جدول‌ها

عنوان صفحه

| | |
|--|--|
| جدول ۴-۱: پارامترهای کنترل‌کننده‌های مختلف طراحی شده، به همراه میزان هزینه‌های مختلف آن‌ها ۴-۸ | |
| جدول ۴-۲: مقادیر پارامترهای کنترل‌کننده‌های به دست آمده از اعمال الگوریتم رقابت استعماری، | |
| الگوریتم ژنتیک و روش DRF ۴-۱۷ | |
| جدول ۴-۳: بخش‌های مختلف تابع هزینه به دست آمده از روش‌های ICA، GA و DRF ۴-۱۷ | |
| جدول ۴-۴: پارامترهای مورد استفاده در الگوریتم رقابت استعماری ۴-۲۵ | |
| جدول ۴-۵: پارامترهای مورد استفاده در الگوریتم ژنتیک ۴-۲۵ | |
| جدول ۴-۶: قوانین فازی مورد استفاده در کنترل سرعت اتومبیل ۴-۳۳ | |

فصل اول

مقدمه

۱-۱ هدف و اهمیت مسئله:

بهینه‌سازی اهمیت زیادی در بسیاری از شاخه‌های علوم دارد. به عنوان مثال فیزیک‌دانها، شیمی‌دانها، و مهندسان علاقه دارند تا یک طرح بهینه برای طراحی یک پروسه شیمیایی به کار برند و محصول تولید شده را با داشتن شروطی مثل هزینه و آلودگی کم، بیشینه کنند. همچنین در برازش غیر خطی مدل و منحنی نیز، به نوعی به بهینه‌سازی، نیاز داریم. اقتصاددانان و تحقیق‌کنندگان در عملیات نیز باید جایابی بهینه منابع در جامعه و صنعت را پیدا کنند. روشهای مطرح شده برای بهینه‌سازی می‌توانند در دو دسته عمده طبقه‌بندی شوند؛ بهینه‌سازی محلی و بهینه‌سازی فراگیر یا عام.

برای بهینه‌سازی عام، اغلب از روش‌های تکاملی استفاده می‌شود. این الگوریتم‌ها شامل الگوریتم‌های ژنتیک، بهینه‌سازی گروه ذرات، بازپخت شبیه‌سازی شده و... می‌باشند. الگوریتم‌های ژنتیک شناخته‌شده‌ترین الگوریتم‌های تکاملی هستند. قواعد اساسی الگوریتم ژنتیک برای اولین بار در سال ۱۹۶۲ توسط هالند معرفی گردید و تا به امروز کاربردهای فراوانی در بهینه‌سازی توابع و شناسایی سیستم پیدا کرده‌اند. آنچه که واضح است این است که تکامل فکری و فرهنگی بشر بسیار سریع‌تر از تکامل جسمی و ژنتیکی او صورت می‌پذیرد. بنابراین تکامل فرهنگی و دیدگاهی بشر نیز نادیده گرفته نشده و دسته‌ای از الگوریتم‌ها، موسوم به الگوریتم‌های فرهنگی معرفی شده‌اند. الگوریتم‌های فرهنگی در حقیقت یک دسته کاملاً جدید از الگوریتم‌ها نیستند. بلکه ایده‌ی اصلی این است که این الگوریتم‌ها با افزودن قابلیت تکامل فرهنگی (با افزودن امکان تبادل اطلاعات میان اعضای جمعیت) به الگوریتم‌های موجود، سرعت همگرایی آنها را مطابق انتظار افزایش می‌دهند.

با توجه به این که اغلب روشهای عمده و شناخته شده محاسبات تکاملی، شبیه‌سازی کامپیوتری فرایندهای طبیعی و زیستی هستند، در این نوشتار یک الگوریتم جدید در زمینه محاسبات تکاملی معرفی می‌شود که بر مبنای تکامل اجتماعی و سیاسی انسان پایه‌گذاری شده است.

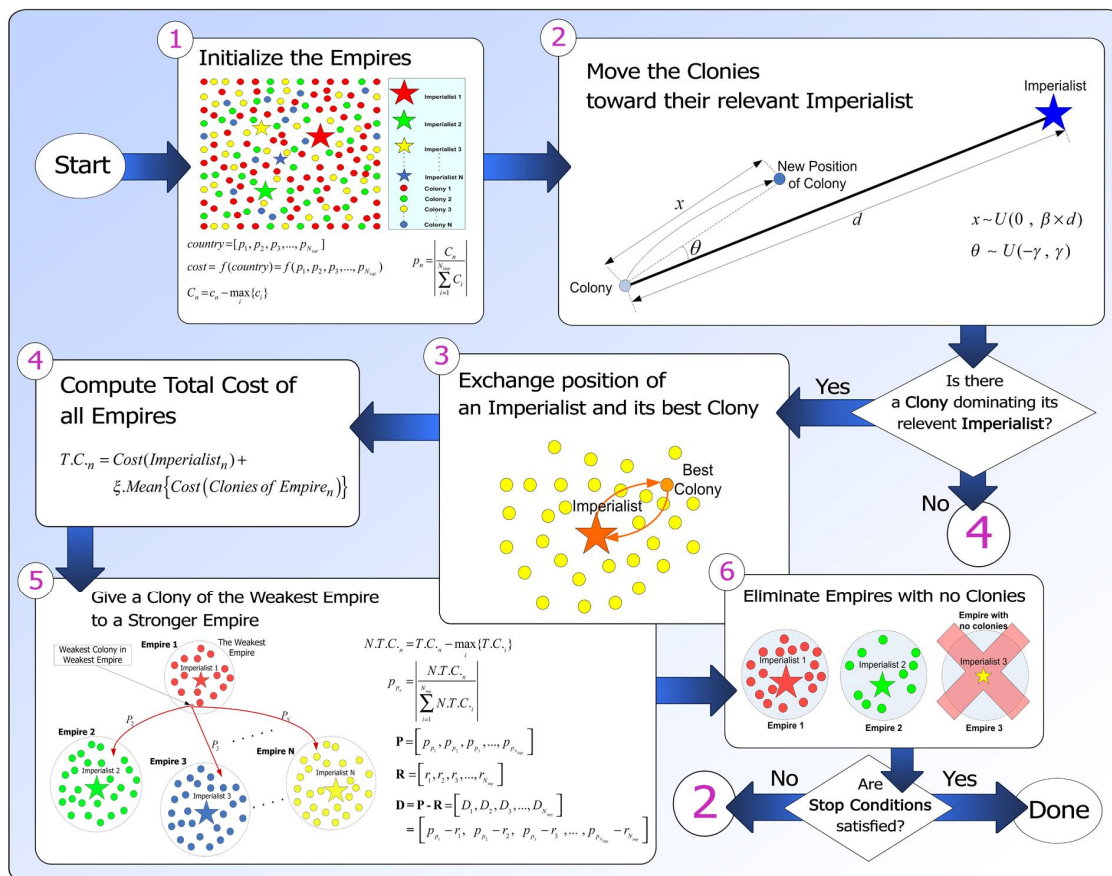
۲-۱ الگوریتم توسعه داده شده:

با در نظر گرفتن الگوریتم‌های بهینه‌سازی مطرح شده، آنچه که قابل توجه است این است که اغلب روش‌های بهینه‌سازی عام مطرح شده، شبیه‌سازی کامپیوتری فرایندهای طبیعی هستند. شاید یک دلیل برای این کار، ملموس بودن و سادگی فرموله کردن و درک تکامل این فرایندها است. در نقطه مقابل، در ارائه‌ی الگوریتم‌های بهینه‌سازی، علی‌رغم توجه به تکامل زیستی انسان و سایر موجودات (الگوریتم‌های ژنتیک و ...)، به تکامل اجتماعی و تاریخی او به عنوان پیچیده‌ترین و موفق‌ترین حالت تکامل، توجه چندانی نشده است. در این طرح، یک الگوریتم الهام گرفته از تکامل اجتماعی انسان، برای بهینه‌سازی، توسعه داده شده است. الگوریتم جدید معرفی شده با الهام‌گیری از یک فرایند اجتماعی سیاسی، نسبت به روش‌های مطرح شده دارای توانایی بالایی بوده و تا حد بسیار زیادی نیز، سریع می‌باشد. شکل ۱-۱ شمای کلی الگوریتم توسعه داده شده موسوم به الگوریتم رقابت استعماری^۱ (ICA) را نشان می‌دهد.

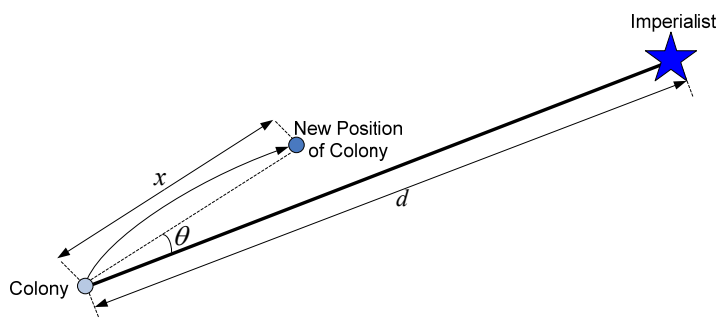
الگوریتم توسعه داده شده، همانند سایر روش‌های بهینه‌سازی تکاملی، با تعدادی جمعیت اولیه شروع می‌شود. در این الگوریتم، هر عنصر جمعیت، یک کشور نامیده می‌شود. کشورها به دو دسته مستعمره و استعمارگر تقسیم می‌شوند. هر استعمارگر، بسته به قدرت خود، تعدادی از کشورهای مستعمره را به سلطه خود درآورده و آن‌ها را کنترل می‌کند. سیاست جذب و رقابت استعماری، هسته اصلی این الگوریتم را تشکیل می‌دهند. مطابق سیاست جذب که به صورت تاریخی، توسط کشورهای استعمارگری همچون فرانسه و انگلیس، در مستعمراتشان اعمال می‌شد، کشورهای استعمارگر با استفاده از روش‌هایی همچون احداث مدارس به زبان خود، سعی در از خود بی خود کردن کشور مستعمره، با از میان بردن زبان کشور مستعمره و فرهنگ و رسوم آن داشتند. در ارائه این الگوریتم، این سیاست با حرکت دادن مستعمرات یک امپراطوری، مطابق یک رابطه خاص صورت می‌پذیرد. شکل ۲-۱ این

^۱ Imperialist Competitive Algorithm

حرکت را نشان می‌دهد.



شکل ۱-۱: شمای کلی الگوریتم رقابت استعماری



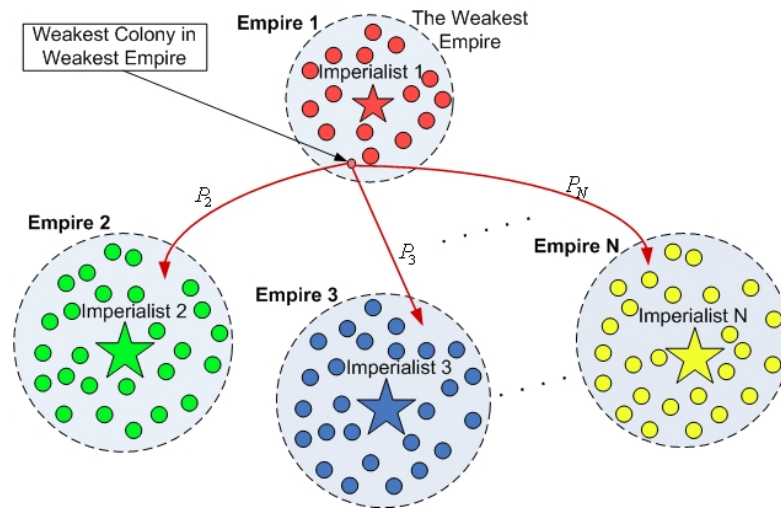
شکل ۱-۲: حرکت مستعمرات به سمت امپریالیست (سیاست جذب)

اگر در حین حرکت، یک مستعمره، نسبت به استمارگر، به موقعیت بهتری برسد، جای آن دو با

هم عوض می‌شوند. در ضمن، قدرت کل یک امپراطوری به صورت مجموع قدرت کشور استعمارگر به اضافه درصدی از قدرت میانگین مستعمرات آن تعریف می‌شود. یعنی

$$T.C_n = Cost(imperialist_n) + \xi \text{ mean}\{Cost(colonies of empire_n)\} \quad (1-1)$$

همانگونه که قبلاً نیز بدان اشاره شد، رقابت استعماری، بخش مهم دیگری از این الگوریتم را تشکیل می‌دهد. در طی رقابت استعماری، امپراطوری‌های ضعیف، به تدریج قدرت خود را از دست داده و به مرور زمان با تضعیف شدن از بین می‌روند. رقابت استعماری باعث می‌شود که به مرور زمان، به حالتی برسیم که در آن تنها یک امپراطوری در دنیا وجود دارد که آن را اداره می‌کند. این حالت زمانی است که الگوریتم رقابت استعماری با رسیدن به نقطه بهینه تابع هدف، متوقف می‌شود. شکل ۳-۱ زیر شمای کلی رقابت استعماری را نشان می‌دهد.



شکل ۳-۱: شمای کلی رقابت استعماری

۳-۱ مزایای الگوریتم توسعه داده شده:

الگوریتم توسعه داده شده، در وحله اول با داشتن یک دیدگاه کاملاً نو به مبحث بهینه‌سازی، پیوندی جدید میان علوم انسانی و اجتماعی از یک سو و علوم فنی و ریاضی از سوی دیگر، برقرار می‌کند. ارتباط میان این دو شاخه از علم به گونه‌ای می‌باشد که غالباً ریاضیات به عنوان ابزاری قوی و دقیق در خدمت علوم انسانی کلی نگر قرار گرفته و به درک و تحلیل نتایج آن کمک می‌کند. اما الگوریتم

توسعه داده شده بر خلاف معمول، نقطه‌ی قوت علوم انسانی و اجتماعی، یعنی کلی‌نگری و وسعت دید آن را به خدمت ریاضیات درآورده و از آن به عنوان ابزاری برای درک بهتر ریاضیات و حل بهتر مسائل ریاضی استفاده می‌کند. بنابراین حتی بدون در نظر گرفتن قابلیت‌های ریاضی و عملی روش توسعه داده شده، پیوند ایجاد شده میان این دو شاخه به ظاهر جدا از هم، به عنوان یک پژوهش میان رشته‌ای، در نوع خود دارای ارزش بسیاری می‌باشد.

مزایای الگوریتم اجتماعی پیشنهادی را می‌توان به صورت زیر خلاصه کرد.

- نو بودن ایده‌ی پایه‌ای الگوریتم: به عنوان اولین الگوریتم بهینه‌سازی مبتنی بر یک فرایند اجتماعی-سیاسی
- توانایی بهینه‌سازی هم‌تراز و حتی بالاتر در مقایسه با الگوریتم‌های مختلف بهینه‌سازی، در مواجهه با انواع مسائل بهینه‌سازی
- سرعت مناسب یافتن جواب بهینه

۱-۴ ساختار پایان‌نامه

در این نوشتار ابتدا در فصل دوم مروری بر بهینه‌سازی و روشهای مطرح شده برای حل مسائل بهینه‌سازی خواهیم داشت. سپس در فصل سوم الگوریتم معرفی شده بیان شده و جزئیات و نحوه پیاده‌سازی آن مورد بررسی قرار می‌گیرد. در فصل چهارم نیز نمونه‌ای از مسائل حل شده توسط الگوریتم معرفی شده را خواهیم دید و در نهایت در فصل پنجم، نتیجه‌گیری و بررسی مزایا و معایب الگوریتم و پیشنهادات برای ادامه کار را خواهیم داشت. در پیوست های ۱ و ۲ نیز به ترتیب در مورد جعبه ابزار تهیه شده و خلاصه مقالات استخراج شده مطالبی را خواهیم داشت.

فصل دوم

بهینه‌سازی و روشهای

موجود

فصل سوم

استراتژی بهینه‌سازی مبتنی
بر تکامل اجتماعی-سیاسی

۳-۱ مقدمه

در این فصل، الگوریتم مطرح شده برای بهینه‌سازی، که از مدل‌سازی ریاضی رقابت‌های امپریالیستی الهام گرفته شده است، معرفی شده و اجزای مختلف آن توضیح داده می‌شود. با داشتن تابع $f(x)$ ، در بهینه‌سازی می‌خواهیم آرگومان x را به گونه‌ای بیابیم که هزینه متناظر آن، بهینه باشد (معمولاً کمینه).

در این فصل، الگوریتم جدیدی برای جستجوی عام معرفی می‌شود که از رقابت‌های استعماری الهام گرفته شده است. بطور خلاصه، این الگوریتم، از چندین کشور در حالت اولیه شروع می‌شود. کشورها در حقیقت جوابهای ممکن مساله هستند و معادل کروموزوم در الگوریتم ژنتیک و ذره در بهینه‌سازی گروه ذرات هستند. همه‌ی کشورها، به دو دسته تقسیم می‌شوند: امپریالیست و مستعمره. کشورهای استعمارگر با اعمال سیاست جذب (همگون‌سازی) در راستای محورهای مختلف بهینه‌سازی، کشورهای مستعمره را به سمت خود می‌شکند. رقابت امپریالیستی در کنار سیاست همگون‌سازی، هسته‌ی اصلی این الگوریتم را تشکیل می‌دهد و باعث می‌شود که کشورها به سمت مینیمم مطلق تابع حرکت کنند. در این فصل به استعمار به عنوان جزئی لاینفک از سیر تکامل تاریخی انسان نگریسته شده و از چگونگی اثرگذاری آن بر کشورهای استعمارگر و مستعمره و نیز کل تاریخ، به عنوان منبع الهام یک الگوریتم کارا و نو در زمینه محاسبات تکاملی استفاده شده است. این فصل، چگونگی مدل‌سازی رقابت امپریالیستی، و نیز چگونگی پیاده‌سازی الگوریتم را توضیح می‌دهد. ابتدا در بخش دوم، یک مرور خلاصه بر جوانب مختلف تاریخی و بعضی از پدیده‌های تاریخی مربوط به استعمار و تاثیر آن بر تکامل اجتماعی سیاسی انسان ارائه می‌شود. در بخش سوم این فصل، الگوریتم معرفی شده، ارائه شده و بخش‌های مختلف آن مورد بررسی قرار می‌گیرند. در بخش چهارم نیز کارایی الگوریتم بر روی چند تابع هزینه استاندارد آزموده می‌شود. در نهایت نیز بخش پنجم، نتیجه‌گیری بحث را ارائه می‌کند.

۲-۳ مروری تاریخی بر پدیده استعمار

امپریالیزم^۱، در لغت به سیاست توسعه قدرت و نفوذ یک کشور در حوزه خارج از قلمرو شناخته شده برای آن، اطلاق می‌شود. یک کشور می‌تواند کشور دیگر را به طور قانونگزاری مستقیم و یا از طریق روش‌های غیر مستقیم، مثل کنترل کالاها و مواد خام، کنترل کند. مورد اخیر اغلب / استعمار نو^۲ خوانده می‌شود [۳۱]. استعمار یک پدیده ذاتی در تاریخ بوده است. استعمار در مراحل ابتدایی، به صورت نفوذ سیاسی-نظامی در کشورها و به صورت صرف استفاده از منابع زمینی، انسانی و سیاسی بوده است. بعضی مواقع نیز استعمار، به صرف جلوگیری از نفوذ کشور استعمارگر رقیب انجام می‌شد. به هر حال کشورهای استعمارگر رقابت شدیدی را برای به استعمار کشیدن مستعمرات همدیگر نشان می‌دادند. این رقابت به نوبه خود باعث رشد و توسعه کشورهای استعمارگر از لحاظ سیاسی، نظامی و اقتصادی گردید. زیرا کشورها برای داشتن امکان رقابت، مجبور به توسعه بودند.

در حالت‌های قدیمی‌تر، استعمارگران با بهره‌گیری از منابع زمینی، انسانی و غیره کشور مستعمره، فقط در صدد افزایش قدرت خود بودند و اینکه آیا مستعمرات پیشرفت می‌کنند یا نه مهم نبود. اما بعدها با فزایش ارتباط میان ملل و رشد انسانی، استعمارگران برای ادامه نفوذ خود، به نوعی از اقبال عمومی (حمایت مردمی) نیز احتیاج پیدا کردند. بدین ترتیب کشورهای استعمارگر شروع به ایجاد عمران و آبادی (هر چند ظاهری) در مستعمراتشان نمودند. بدین ترتیب، مستعمرات، شاهد پیشرفت در زمینه‌های اقتصادی، اجتماعی و انسانی شدند که عامل این پیشرفت به اجبار، کشور استعمارگر بود. دلیل نامگذاری این فرایند با نام "استعمار" که ریشه در کلمه عمران و آبادی دارد، نیز، همین مساله می‌باشد. البته دریافت اقبال عمومی تنها دلیل ایجاد عمران توسط استعمارگران در مستعمرات نبود. یک دلیل دیگر ایجاد سلطه فرهنگی بر مسعمرات در راستای اجرای سیاست همگون‌سازی بود. به عنوان مثال کشورهای نظیر فرانسه و انگلیس به ایجاد مدارس انگلیسی زبان در مستعمرات خود پرداختند. این اقدام به دلایل مختلفی

^۱ Imperialism

^۲ neocolonialism

صورت می‌گرفت که در راس این دلایل افزایش نفوذ فرهنگی در مستعمرات بوده است. نا گفته نماند که فرایند استعمار (حد اقل بعد فرهنگی آن) با همه تبعات منفی آن در بعضی از کشورهای امپریالیست به چشم یک جهاد فکری برای نجات بشر نیز نگریسته می‌شد. اشعاری وجود دارند که به مدح و ستایش جوانان انگلیسی می‌پردازند که با هدف آموزش راهی کشورهای مستعمره شده‌اند و در آنها از این جوانان به عنوان قهرمانان ملی در جبهه نجات بشری یاد می‌شود [۳۳].

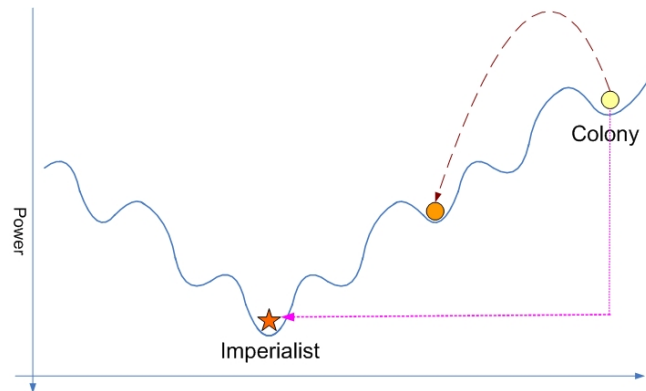
امپریالیزم، نگرش عمومی نسبت به تمدن غرب را تغییر داد. داروینیست‌های اجتماعی، امپریالیزم را تفسیر کرده و این ایده که فرهنگ غرب، نسبت به فرهنگ شرق، برتر است؛ را تقویت کردند [۳۲]. در آفریقا تنها آنهایی که بعضی از استانداردهای فرهنگی غرب را داشتند، دارای بخشی از حقوق اجتماعی خود بودند. پرتغالی‌ها این مردم را جذب شده^۱ و فراسوی‌ها بطور توهین‌آمیزی آن‌ها را تکامل‌یافته^۲ می‌نامیدند.

به هر حال مستقل از اثرات و تبعات مثبت و منفی آن، استعمار به عنوان یک فرایند ذاتی در تاریخ بشر ایجاد شد، و در عین وارد کردن خسارتهای جبران ناپذیر به زیربنایهای اساسی یک کشور (خصوصاً زیربنایهای فرهنگی) در بعضی موارد اثرات مثبتی را نیز برای کشورها مستعمره داشت. از دید بهینه‌سازی، استعمار بعضی از کشورها را که در یک دره معمولی تمدن قرار داشتند، خارج کرده و آنها را به یک حوزه مینیمم دیگر برد که در بعضی موارد وضعیت این حوزه مینیمم بهتر از موقعیت قبلی کشور مستعمره بود. اما به هر حال این حرکت مستلزم پیشروی مستعمره در راستای محورهای مختلف اقتصادی و فرهنگی به سمت یک امپریالیست قویتر بود، یعنی از میان رفتن بعضی از ساختارهای فرهنگی و اجتماعی. شکل ۱-۳ این وضعیت را به خوبی نشان می‌دهد. در این شکل، مستعمره در نتیجه سیاست همگون‌سازی از یک ناحیه مینیمم خارج شده و وارد یک ناحیه مینیمم دیگر می‌شود که در آن وضعیت بهتری را دارا می‌باشد. به هر حال هزینه‌ای که بابت این حرکت پرداخت شده است، نزدیکی به کشور

^۱ assimilados

^۲ evolues

استعمارگر در راستای محورهای مختلف اقتصادی، سیاسی و اجتماعی است. ادامه این حرکت می‌تواند به جذب کامل کشور مستعمره در کشور استعمارگر بیانجامد.



شکل ۳-۱: اعمال سیاست جذب از طرف استعمارگران بر مستعمرات

در این بخش به بررسی چند مورد از مستعمرات کشورهای استعماری و رفتار متقابل آنها نسبت به هم می‌پردازیم و سعی بر آن است تا با دسته‌بندی این رفتارها و با کشف نظم درونی آنها و در نهایت مدل‌سازی ریاضی واکنشهای متقابل مستعمرات و استعمارگران، این پدیده‌ی پیچیده‌ی اجتماعی را در قالب یک الگوریتم بهینه‌سازی ریاضی بریزیم.

۳-۲-۱ هند

اروپاییها از طریق دریا به هند آمدند و در نهایت به تهدیدی علیه این سرزمین تبدیل شدند. نخستین باری که اروپاییها در قرن شانزدهم به هند آمدند، امپراطوری مغول قدرت را در این کشور در دست داشت. معه‌ذا، در قرن هیجدهم خاندان مغول در حال اضمحلال بود و با جنگهای داخلی و دخالت خارجی، قدرت سیاسی آنها تجزیه گردید. در نیمه قرن هیجدهم، هند، طعمه امپریالیست‌های رقیب، بریتانیا و فرانسه شده بود. بریتانیا و فرانسه بر سر توفیق استعماری بر جهان در حال نبرد با یکدیگر بودند و بریتانیای کبیر در هند به پیروزی رسید. بخش‌هایی از هند مستقیماً تحت حاکمیت بریتانیا قرار گرفت و در حالی که بر بخش‌های دیگر، شاهزادگان هند با نظارت بریتانیا، حکومت می‌کردند [۳۴].

بعد از آرام کردن این کشور، بریتانیاییها به تاسیس مدارس انگلیسی‌زبان و احداث جاده، راه آهن و خط تلگراف پرداختند. حکومت برتانیا همچنین برای اصلاح رسوم اجتماعی که در مقایسه با معیارهای

غربی نادرست تلقی می‌شدند، تدابیری اتخاذ کرد. این تدابیر مشتمل بود [۳۴] بر منسوخ کردن رسوم و عاداتی چون

- خودسوزی بیوه‌زنان کاست بالای جامعه که برای نشان دادن وفاداری به شوهر انجام می‌شد.
- سرکوب مجرمانی که به نام مذهب، دزدی و جنایت می‌کردند.
- افزایش حداقل سن ازدواج برای دختران.

بسیاری از هندیها، این اصلاحات را سودمند تلقی کرده و به تردید و چالش غربی‌ها، در قبال ارزش‌های سنتی خود، با بازبینی و ارزشیابی مجدد مذهب و جامعه خویش، پاسخ گفتند. در نتیجه، بسیاری از هندیها، فعالانه از این اصلاحات نظیر قانون ۱۸۹۱ که ازدواج دختران خردسال را منع می‌نمود، حمایت کردند. معهدا سنت‌گرایان هندو خشمگین شده و به عنوان مصداق دخالت بریتانیا در جامعه هند، به قانون مذکور اعتراض کردند [۳۴].

۳-۲-۲ مالزی

مالزی یکی از مستعمرات بریتانیا شد. حاکمیت بریتانیا بر مالزی، غیر مستقیم، و از طریق حکام بومی موسوم به سلطان، اعمال می‌شد که در آن زمان تبدیل به نیمه‌دست‌نشانده شده بودند. بخش سودمند اعمال کنترل بریتانیا بر مالزی، شامل الغای برده‌داری و مالیات‌های خودسرانه، احداث راهها، خطوط آهن، مدارس و برقراری نظام جدید بهداشتی بود. معهدا مسلمانان مالزی به میزان ناچیزی از توسعه سریع اقتصادی منتفع شدند و اغلب آنان همچنان با کشاورزی و ماهیگیری، امرار معاش می‌کردند [۳۳، ۳۴].

۳-۲-۳ هندوچین فرانسه

امپریالیزم فرانسه، در هندوچین مستقر شد. دلایل علاقه‌مندی فرانسه به این منطقه، متعدد بود:

- منابع طبیعی
- قلمروی برای فعالیت مبلغان کاتولیک
- دروازه فرعی ورود به چین

- اهرم مقابله با امپریالیسم بریتانیا

از لحاظ فرهنگی و سیاسی، فرانسه سیاست دو محوری "جذب" و "همراهی" را تعقیب می‌کرد. هدف سیاست جذب ایجاد یک فرانسه جدید در ویتنام، از طریق شیوه‌هایی نظیر تأسیس مدارس فرانسوی و توسعه زبان و رسوم فرانسوی بود. فرانسه امیدوار بود، سرانجام در میان ویتنامی‌ها طبقه ممتاز جدیدی به وجود آید که با حاکمیت فرانسه موافق باشد. اما جریان امور به این صورت پیش نرفت. اقلیت کوچک ویتنامی که به فرهنگ فرانسوی دست یافته بودند، مانند هندی‌های صاحب تحصیلات بریتانیایی، دانش جدید خود را در راه مخالفت با سلطه فرانسه و حمایت از استقلال ویتنام، به کار گرفتند.

از لحاظ سیاسی، خط مشی فرانسه توسعه تدریجی کنترل خود از طریق سیاست "همراهی" بود. به این معنی که مقامات فرانسوی مقیم، از نزدیک دستگاه اداری حکومت‌های محلی را سرپرستی می‌کردند.

فرانسوی‌ها برای تسهیل در امر توسعه اقتصادی، به احداث جاده و خطوط آهن پرداختند و سیستم آبیاری را توسعه دادند و تسهیلات آموزشی و بهداشت عمومی مدرن، ایجاد کردند. ویتنامی‌ها عموماً بهای سنگینی برای این عمران و آبادی پرداختند، اما بسیار کم از آن منتفع شدند [۳۴].

۳-۲-۴ هند شرقی (اندونزی)

هلندیها از اوایل قرن هفدهم در هند شرقی، دخالت داشتند. هلندیها از آغاز، مستعمرات خویش را به عنوان منابع ارزشمند مواد خام و بعدها به عنوان بازاری برای فروش محصولات صنعتی، قلمداد می‌کردند. در اوایل قرن بیستم، با استخراج معادن و حفاری چاههای نفت، بهره‌برداری از این ثروت‌های جدید آغاز شد. دوره‌ای از توسعه اقتصادی به هند شرقی آمده بود. معه‌ذا این رفاه به جای اندونزیاییها، نصیب هلندی‌ها شد. این روند علی‌رغم "سیاست اخلاقی" که هلندیها در آستانه چرخش قرن مطرح می‌کردند به وقوع پیوست. سیاست پدر سالارانه هلندیها، بر تعهد اخلاقی در قبال مردم بومی و بهبود وضعیت رفاهی آن تاکید می‌کرد. این سیاست منجر به تأسیس مدارس ابتدایی دولتی و اتخاذ تدابیری،

برای حمایت از مردم عادی اندونزی در مقابل اشکال آشکارتر استثمار اقتصادی شد. هلند بر خلاف فرانسه به سیاست "جذب"، علاقه‌ای نداشت و برای اشاعه نظریات و شیوه‌های غربی در میان مردم بومی کوششی نکرد [۳۴].

با در نظر گرفتن رفتار چند کشور استعمارگر در قبال کشورهای مستعمره، به نظر می‌رسد که اگرچه سیاست‌های مذکور نتوانستند قدرت و نفوذ کشورهای امپریالیست را در میان مستعمراتشان، افزایش دهند، و مستعمرات بعد از مدتی خواستار خودمختاری سیاسی شدند؛ اما به همراه همه معایشان، این سیاست‌ها تغییرات سیاسی-اجتماعی سریعی را در میان مستعمرات، ایجاد کردند. حوادث قرن بیستم به گونه‌ای رقم خورد که اکثر کشورهای مستعمره، در نتیجه انقلاب داخلی و یا ضعف کشور استعمارگر، توانستند به استقلال (حداقل سیاسی) دست پیدا کنند. اما نوع جدیدی از استعمار در حال شکل‌گیری بوده و جایگزین شیوه قدیمی آن شد و در حال حاضر نیز این روند ادامه دارد و چنین به نظر می‌رسد که حد توقف این روند (رقابت‌های امپریالیستی) زمانی خواهد بود که یک دنیای تک قطبی داشته باشیم، با یک امپریالیست قدرتمند. بر پایه‌ی چنین روندی است که الگوریتم معرفی شده (ICA) در این فصل پایه‌گذاری شده و از آن در طی فصول بعدی برای اهداف مختلف بهینه‌سازی، استفاده می‌شود.

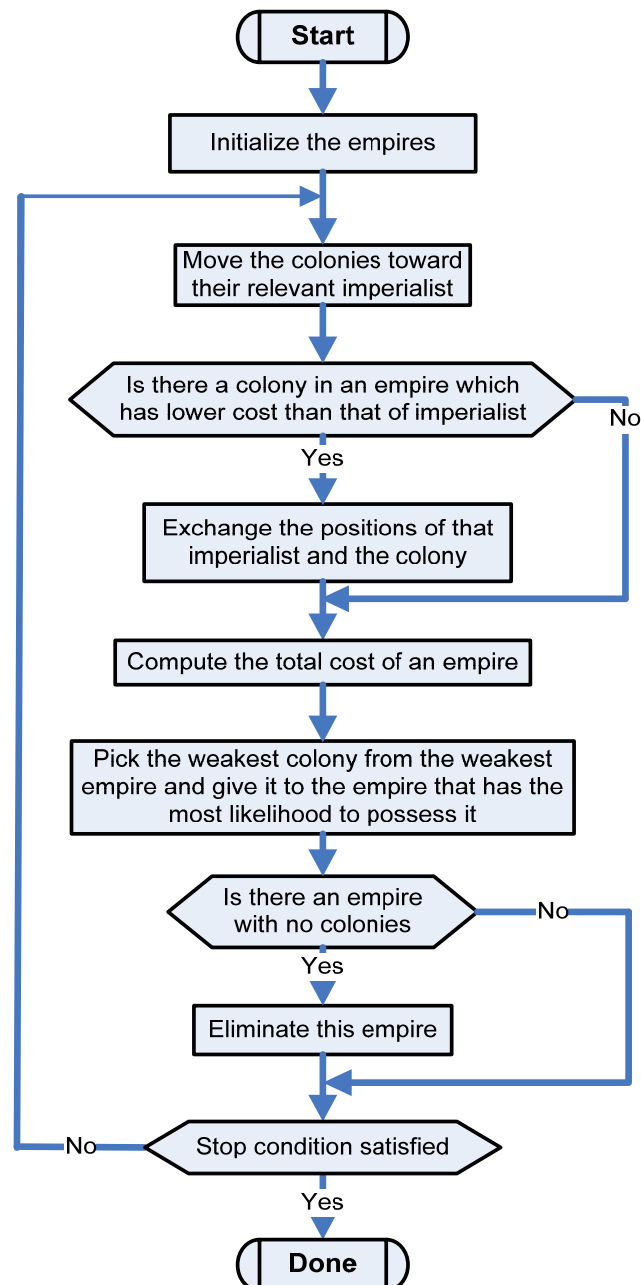
۳-۳ الگوریتم پیشنهادی

شکل ۳-۲ فلوچارت الگوریتم پیشنهادی را نشان می‌دهد. همانند دیگر الگوریتم‌های تکاملی، این الگوریتم، نیز با تعدادی جمعیت اولیه تصادفی که هر کدام از آنها یک "کشور" نامیده می‌شوند؛ شروع می‌شود. تعدادی از بهترین عناصر جمعیت (معادل نخبه‌ها در الگوریتم ژنتیک) به عنوان امپریالیست^۱ انتخاب می‌شوند. باقیمانده جمعیت نیز به عنوان مستعمره^۲، در نظر گرفته می‌شوند. استعمارگران بسته به قدرتشان، این مستعمرات را با یک روند خاص که در ادامه می‌آید؛ به سمت خود می‌کشند. قدرت کل

^۱ Imperialist

^۲ Colony

هر امپراطوری، به هر دو بخش تشکیل دهنده آن یعنی کشور امپریالیست (به عنوان هسته مرکزی) و مستعمرات آن، بستگی دارد. در حالت ریاضی، این وابستگی با تعریف قدرت امپراطوری به صورت مجموع قدرت کشور امپریالیست، به اضافه در صدی از میانگین قدرت مستعمرات آن، مدل شده است.



شکل ۳-۲: فلوچارت الگوریتم پیشنهادی

با شکل‌گیری امپراطوری‌های اولیه، رقابت امپریالیستی میان آن‌ها شروع می‌شود. هر امپراطوری‌ای

که نتواند در رقابت استعماری، موفق عمل کرده و بر قدرت خود بیفزاید (و یا حداقل از کاهش نفوذش جلوگیری کند)، از صحنه رقابت استعماری، حذف خواهد شد. بنابراین بقای یک امپراطوری، وابسته به قدرت آن در جذب مستعمرات امپراطوری‌های رقیب، و به سیطره در آوردن آنها خواهد بود. در نتیجه، در جریان رقابت‌های امپریالیستی، به تدریج بر قدرت امپراطوری‌های بزرگتر افزوده شده و امپراطوری‌های ضعیف‌تر، حذف خواهند شد. امپراطوری‌ها برای افزایش قدرت خود، مجبور خواهند شد تا مستعمرات خود را نیز پیشرفت دهند.

با گذشت زمان، مستعمرات، از لحاظ قدرت به امپراطوری‌ها نزدیک‌تر خواهند شد و شاهد یک نوع همگرایی خواهیم بود. حد نهایی رقابت استعماری، زمانی است که یک امپراطوری واحد در دنیا داشته باشیم، با مستمراتی که از لحاظ موقعیت، به خود کشور امپریالیست، خیلی نزدیک هستند. در ادامه مباحث این فصل، بخش‌های مختلف الگوریتم، مورد بررسی قرار می‌گیرند.

۳-۳-۱ شکل دهی امپراطوری‌های اولیه

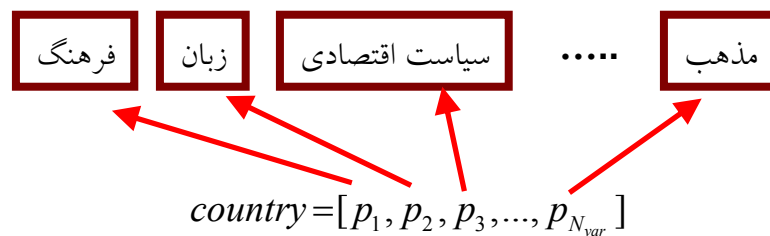
در بهینه‌سازی، هدف یافتن یک جواب بهینه بر حسب متغیرهای مسئله، است. ما یک آرایه از متغیرهای مسئله را که باید بهینه شوند، ایجاد می‌کنیم. در الگوریتم ژنتیک این آرایه، کروکوزوم^۱ نامیده می‌شود. در اینجا نیز آن را یک کشور می‌نامیم. در یک مسئله بهینه‌سازی N_{var} بعدی، یک کشور، یک آرایه $1 \times N_{var}$ است. این آرایه به صورت زیر تعریف می‌شود.

$$country = [p_1, p_2, p_3, \dots, p_{N_{var}}]$$

مقادیر متغیره‌ها در یک کشور، به صورت اعداد اعشاری نمایش داده می‌شوند. از دیدگاه تاریخی-فرهنگی، اجزای تشکیل دهنده یک کشور را می‌توان ویژگی‌های اجتماعی-سیاسی آن کشور، همچون فرهنگ، زبان، ساختار اقتصادی و سایر ویژگی‌ها در نظر گرفت. شکل ۳-۳ این مسئله را به

¹ chromosome

خوبی نشان می‌دهد. مطابق این شکل متغیرهای مجهول تابع هزینه که ما در طی فرایند بهینه‌سازی به دنبال آنها می‌گردیم، در نگاه اجتماعی-سیاسی ویژگی‌های تاریخی و فرهنگی‌ای هستند که یک کشور را به نقطه مینیمم تابع هزینه رهنمون می‌سازند. در حقیقت در حل یک مسئله بهینه‌سازی توسط الگوریتم معرفی شده، ما به دنبال بهترین کشور (کشوری با بهترین ویژگی‌های اجتماعی-سیاسی) هستیم. یافتن این کشور در حقیقت معادل یافتن بهترین پارامترهای مسئله است که کمترین مقدار تابع هزینه را تولید می‌کنند.



شکل ۳-۳: اجزای اجتماعی سیاسی تشکیل دهنده یک کشور

به عنوان یک مثال فرض کنیم که می‌خواهیم یک کنترل کننده PID برای یک سیستم کنترلی طراحی کنیم که مثلاً دارای کمترین میزان مجموع فراجش و انتگرال قدر مطلق خطا باشد. در یک حالت نوعی، جوابهای ممکنه می‌توانند به صورت جوابهایی که به یک خروجی پایدار منجر می‌شوند، تعریف شوند. برای این مسئله دسته‌ای از جوابهای ممکنه به صورت اولیه ایجاد می‌کنیم. در این مساله کشور iam به صورت زیر تعریف می‌شود.

$$country_i = [KP_i, KI_i, KD_i] \quad (۳-۱)$$

برای شروع الگوریتم باید تعدادی از این کشورها (به تعداد کشورهای اولیه الگوریتم) ایجاد شوند. بنابراین ماتریس کل کشورها به صورت تصادفی اولیه تشکیل می‌شود.

$$COUNTRY = \begin{bmatrix} country_1 \\ country_2 \\ country_3 \\ . \\ . \\ . \\ country_{N_{country}} \end{bmatrix} = \begin{bmatrix} KP_1 & KI_1 & KD_1 \\ KP_2 & KI_2 & KD_2 \\ KP_3 & KI_3 & KD_3 \\ . & . & . \\ . & . & . \\ . & . & . \\ KP_{N_{country}} & KI_{N_{country}} & KD_{N_{country}} \end{bmatrix} \quad (3-2)$$

هزینه‌ی یک کشور با ارزیابی تابع f در متغیرهای $(p_1, p_2, p_3, \dots, p_{N_{var}})$ یافته می‌شود. بنابراین

$$cost_i = f(country_i) = f(p_1, p_2, p_3, \dots, p_{N_{var}}) \quad (3-3)$$

در مسئله طراحی کنترل کننده، با هدف در نظر گرفته شده، این تابع به صورت زیر خواهد بود.

$$F = w_1 \times MaxOvershoot + w_2 \times IAE \quad (3-4)$$

که در آن $MaxOvershoot$ ماکزیمم فراجش و IAE انتگرال قدر مطلق خطا است. w_1 و w_2 نیز وزنهایی هستند که میزان اهمیت هر یک از هدف‌ها را نشان می‌دهند. بنابراین کاری که برای بدست آوردن هزینه یک کشور (دسته پارامتهای کنترل کننده PID) باید انجام شود، این است که هر دسته از این ضرایب به عنوان کنترل کننده در نظر گرفته شده و پاسخ پله سیستم برای این کنترلر بدست می‌آید. در نهایت با محاسبه ماکزیمم فراجش و انتگرال قدر مطلق خطا، مجموع آنها را به عنوان هزینه این کشور (ضرایب کنترل کننده) محاسبه می‌شود. ما به دنبال بهترین کشور (بهترین دسته ضرایب کنترل کننده) می‌گردیم. الگوریتم معرفی شده در این نوشتار، با تولید یک دسته اولیه از این ضرایب و دسته بندی آنها در قالب امپراطوری‌ها و اعمال سیاست جذب از طرف استعمارگران به روی مستعمرات و همچنین با ایجاد رقابت استعماری میان امپراطوریه‌ها به جستجوی بهترین کشور می‌پردازد.

برای شروع الگوریتم، تعداد $N_{country}$ کشور اولیه را ایجاد می‌کنیم. N_{imp} تا از بهترین اعضای این جمعیت (کشورهای دارای کمترین مقدار تابع هزینه) را به عنوان امپریالیست انتخاب می‌کنیم. باقیمانده N_{col} تا از کشورها، مستعمراتی را تشکیل می‌دهند که هرکدام به یک امپراطوری تعلق دارند. برای تقسیم مستعمرات اولیه بین امپریالیست‌ها، به هر امپریالیست، تعدادی از مستعمرات را که این تعداد، متناسب با

قدرت آن است، می‌دهیم. برای انجام این کار، با داشتن هزینه همه امپریالیست‌ها، هزینه نرمالیزه آن‌ها را به صورت زیر در نظر می‌گیریم.

$$C_n = \max_i \{c_i\} - c_n \quad (3-5)$$

که در آن c_n ، هزینه امپریالیست n ام، $\max_i \{c_i\}$ بیشترین هزینه میان امپریالیست‌ها و C_n ، هزینه نرمالیزه شده این امپریالیست، می‌باشد. هر امپریالیستی که دارای هزینه بیشتری باشد (امپریالیست ضعیفتری باشد)، دارای هزینه نرمالیزه کمتری خواهد بود. با داشتن هزینه نرمالیزه، قدرت نسبی نرمالیزه‌ی هر امپریالیست، به صورت زیر محاسبه شده و بر مبنای آن، کشورهای مستعمره، بین امپریالیست‌ها تقسیم می‌شوند.

$$p_n = \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \quad (3-6)$$

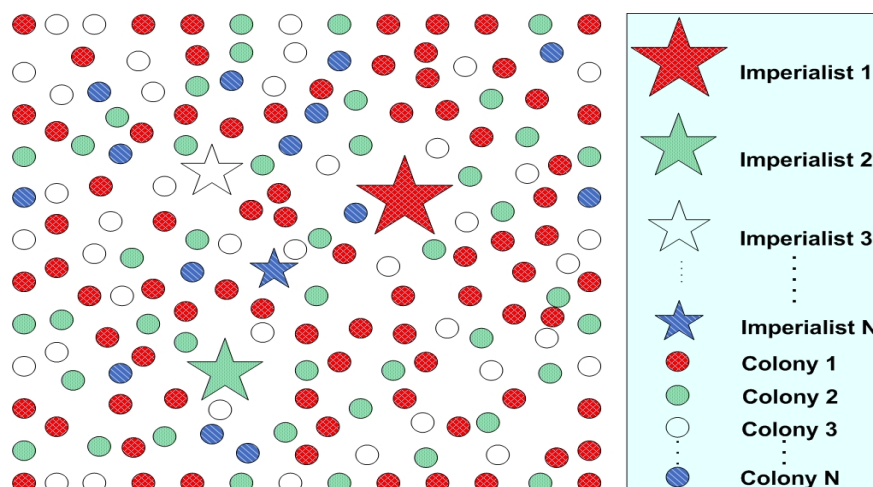
از یک دید دیگر، قدرت نرمالیزه شده یک امپریالیست، نسبت مستعمراتی است که توسط آن امپریالیست اداره می‌شود. بنابراین تعداد اولیه‌ی مستعمرات یک امپریالیست برابر خواهد بود با

$$N.C_n = \text{round}\{p_n \cdot (N_{col})\} \quad (3-7)$$

که در آن $N.C_n$ ، تعداد اولیه مستعمرات یک امپراطوری و N_{col} نیز تعداد کل کشورهای مستعمره موجود در جمعیت کشورهای اولیه است. round نیز تابعی است که نزدیک‌ترین عدد صحیح به یک عدد اعشاری را می‌دهد. با در نظر گرفتن $N.C.$ برای هر امپراطوری، به این تعداد از کشورهای مستعمره اولیه را به صورت تصادفی انتخاب کرده و به امپریالیست n ام می‌دهیم. با داشتن حالت اولیه تمام امپراطوری‌ها، الگوریتم رقابت استعماری شروع می‌شود. روند تکامل در یک حلقه قرار دارد که تا برآورده شدن یک شرط توقف، ادامه می‌یابد.

شکل ۳-۴ چگونگی شکل‌گیری امپراطوری‌های اولیه را نشان می‌دهد. همانگونه که در این شکل نشان داده شده است. امپراطوری‌های بزرگتر، تعداد بیشتری مستعمره دارند. در این شکل، امپریالیست

شماره ۱ قوی‌ترین امپراطوری را ایجاد کرده است و بیش‌ترین تعداد مستعمرات را دارد.



شکل ۳-۴: چگونگی شکل‌گیری امپراطوری‌های اولیه. امپریالیست شماره ۱ قوی‌ترین امپراطوری را ایجاد کرده است و بیش‌ترین تعداد مستعمرات را دارد.

۳-۳-۲ مدل‌سازی سیاست جذب: حرکت مستعمره‌ها به سمت امپریالیست

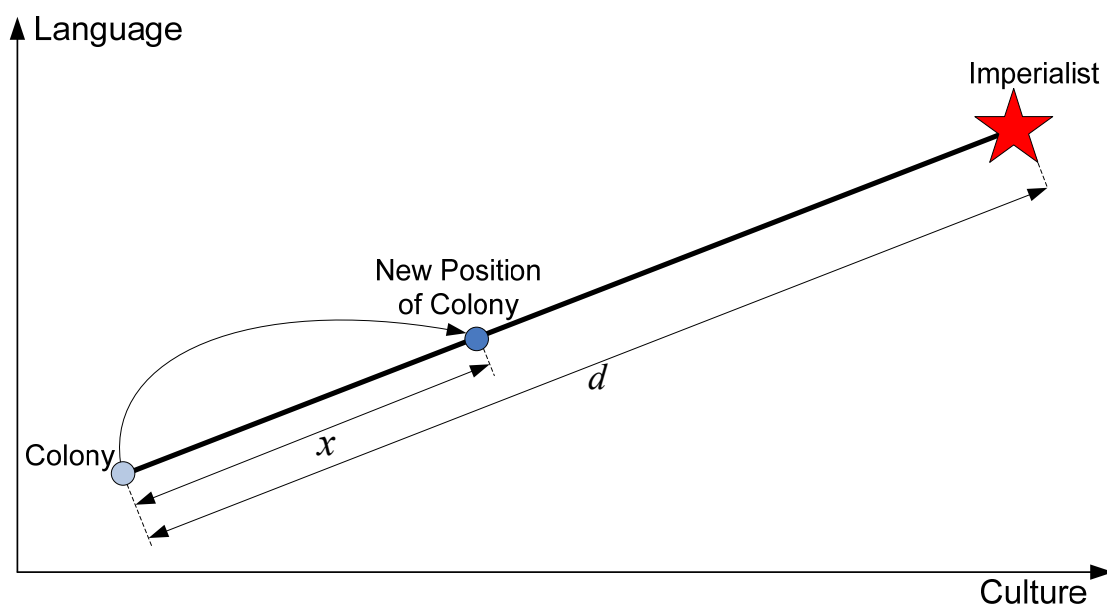
سیاست همگون‌سازی^۱ (جذب) با هدف تحلیل فرهنگ و ساختار اجتماعی مستعمرات در فرهنگ حکومت مرکزی انجام می‌گرفت. همانگونه که قبلاً نیز بیان شد، کشورهای استعمارگر، برای افزایش نفوذ خود، شروع به ایجاد عمران (ایجاد زیرساخت‌های حمل و نقل، تاسیس دانشگاه و ...) کردند. به عنوان مثال کشورهایی نظیر انگلیس و فرانسه با تعقیب سیاست همگون‌سازی در مستعمرات خود در فکر ایجاد انگلیس نو^۲ و فرانسه نو^۳ در مستعمرات خویش بودند. با در نظر گرفتن شیوه نمایش یک کشور در حل مسئله بهینه‌سازی، در حقیقت این حکومت مرکزی با اعمال سیاست جذب سعی داشت تا کشور مستعمره را در راستای ابعاد مختلف اجتماعی سیاسی به خود نزدیک کند. این بخش از فرایند استعمار در الگوریتم بهینه‌سازی، به صورت حرکت مستعمرات به سمت کشور امپریالیست، مدل شده است. شکل

^۱ Assimilation

^۲ New England

^۳ New France

۵-۳، شمای کلی این حرکت را نشان می‌دهد.

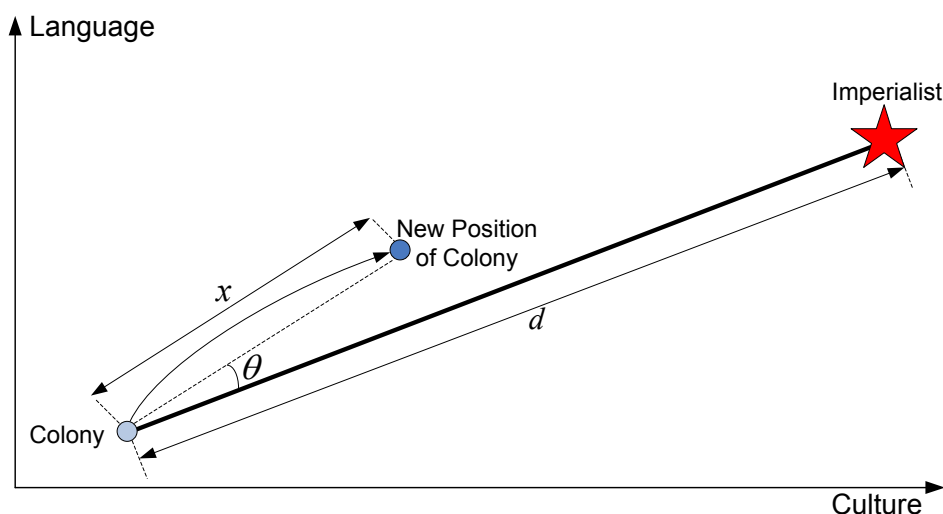


شکل ۵-۳: شمای کلی حرکت مستعمرات به سمت امپریالیست.

مطابق این شکل کشور امپریالیست کشور مستعمره را در راستای محورهای فرهنگ و زبان به سمت خود جذب می‌کند. همانگونه که در این شکل نشان داده شده است، کشور مستعمره (Colony)، به اندازه x واحد در جهت خط واصل مستعمره به استعمارگر (Imperialist)، حرکت کرده و به موقعیت جدید (New Position of Colony)، کشانده می‌شود. در این شکل، فاصله میان استعمارگر و مستعمره با d نشان داده شده است. x نیز عددی تصادفی با توزیع یکنواخت (و یا هر توزیع مناسب دیگر) می‌باشد. یعنی برای x داریم.

$$x \sim U(0, \beta \times d) \quad (3-8)$$

که در آن β عددی بزرگتر از یک و نزدیک به ۲ می‌باشد. یک انتخاب مناسب می‌تواند $\beta=2$ باشد. وجود ضریب $\beta > 1$ باعث می‌شود تا کشور مستعمره در حین حرکت به سمت کشور استعمارگر، از جهت‌های مختلف به آن نزدیک شود.



شکل ۶-۳: حرکت واقعی مستعمرات به سمت امپریالیست

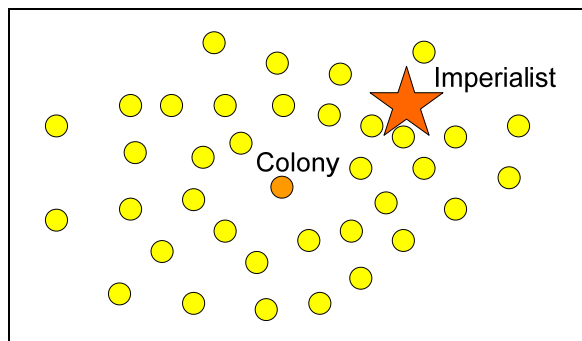
با بررسی تاریخی پدیده همگون‌سازی، یک حقیقت آشکار در این زمینه این است که علی‌رغم اینکه کشورهای استعمارگر بطور جدی پیگیر سیاست جذب بودند، اما وقایع بطور کامل مطابق سیاست اعمال شده آنها پیش نمی‌رفت و انحرافات در نتیجه کار وجود داشت. در الگوریتم معرفی شده، این انحراف احتمالی با افزودن یک زاویه تصادفی به مسیر جذب مستعمرات، انجام می‌گیرد. بدین منظور، در حرکت مستعمرات به سمت استعمارگر، کمی زاویه تصادفی نیز به جهت حرکت مستعمره، اضافه می‌کنیم. شکل ۶-۳ این حالت را نشان می‌دهد. بدین منظور این بار به جای حرکت به اندازه x ، به سمت کشور استعمارگر و در جهت بردار واصل مستعمره به استعمارگر، به همان میزان، ولی با انحراف θ در مسیر، به حرکت خود ادامه می‌دهیم. θ را به صورت تصادفی و با توزیع یکنواخت در نظر می‌گیریم (اما هر توزیع دلخواه و مناسب دیگر نیز می‌تواند استفاده شود). پس

$$\theta \sim U(-\gamma, \gamma) \quad (3-9)$$

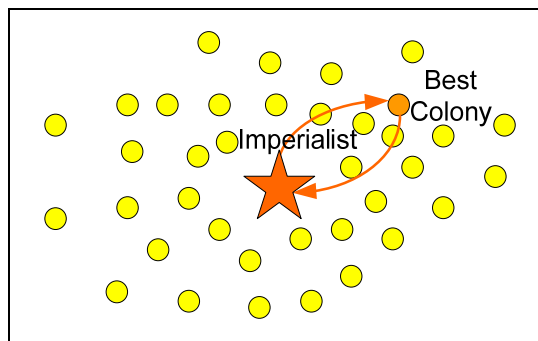
در این رابطه، γ پارامتری دلخواه می‌باشد که افزایش آن باعث افزایش جستجوی اطراف امپریالیست شده و کاهش آن نیز باعث می‌شود تا مستعمرات تا حد ممکن، به بردار واصل مستعمره به استعمارگر، نزدیک حرکت کنند. با در نظر گرفتن واحد رادیان برای θ ، عددی نزدیک به $\pi/4$ ، در اکثر پیاده‌سازی‌ها، انتخاب مناسبی بوده است.

۳-۳-۳ جابجایی موقعیت مستعمره و امپریالیست

سیاست جذب در عین نابودی ساختارهای اجتماعی سیاسی کشور مستعمره در بعضی موارد نتایج مثبتی را نیز برای آنها در پی داشت. بعضی از کشور در نتیجه اعمال این سیاست به نوعی از خودباوری عمومی دست یافتند و پس از مدتی همان تحصیلکرده‌گان (به عبارت دیگر جذب شدگان فرهنگ استعماری) بودند که به رهبری ملت خود برای رهایی از چنگال استعمار پرداختند. نمونه‌های فراوانی از این موارد را می‌توان در مستعمرات انگلیس و فرانسه یافت. از سوی دیگر نگاهی به فراز و نشیب چرخش قدرت در کشورها به خوبی نشان می‌دهد که کشور هایی که زمانی در اوج قدرت سیاسی - نظامی بودند، پس از مدتی سقوط کردند و در مقابل کشورهایی سکان قدرت را در دست گرفتند که زمانی هیچ قدرتی در دست نداشتند. در مدلسازی این واقعه تاریخی در الگوریتم معرفی شده به این صورت عمل شده است که در حین حرکت مستعمرات به سمت کشور استعمارگر، ممکن بعضی از این مستعمرات به موقعیتی بهتر از امپریالیست برسند (به نقاطی در تابع هزینه برسند که هزینه کمتری را نسبت به مقدار تابع هزینه در موقعیت امپریالیست، تولید می‌کنند). در این حالت، کشور استعمارگر و کشور مستعمره، جای خود را با همدیگر عوض کرده و الگوریتم با کشور استعمارگر در موقعیت جدید ادامه یافته و این بار این کشور امپریالیست جدید است که شروع به اعمال سیاست همگون‌سازی بر مستعمرات خود می‌کند. تغییر جای استعمارگر و مستعمره، در شکل ۳-۷ نشان داده شده است. در این شکل، بهترین مستعمره‌ی امپراطوری، که هزینه‌ای کمتر از خود امپریالیست دارد، به رنگ تیره‌تر، نشان داده شده است. شکل ۳-۸، کل امپراطوری را پس از تغییر موقعیت‌ها، نشان می‌دهد.



شکل ۳-۷: کل امپراطوری، پس از تغییر موقعیت‌ها



شکل ۳-۸: تغییر جای استعمارگر و مستعمره

۳-۳-۴ قدرت کل یک امپراطوری

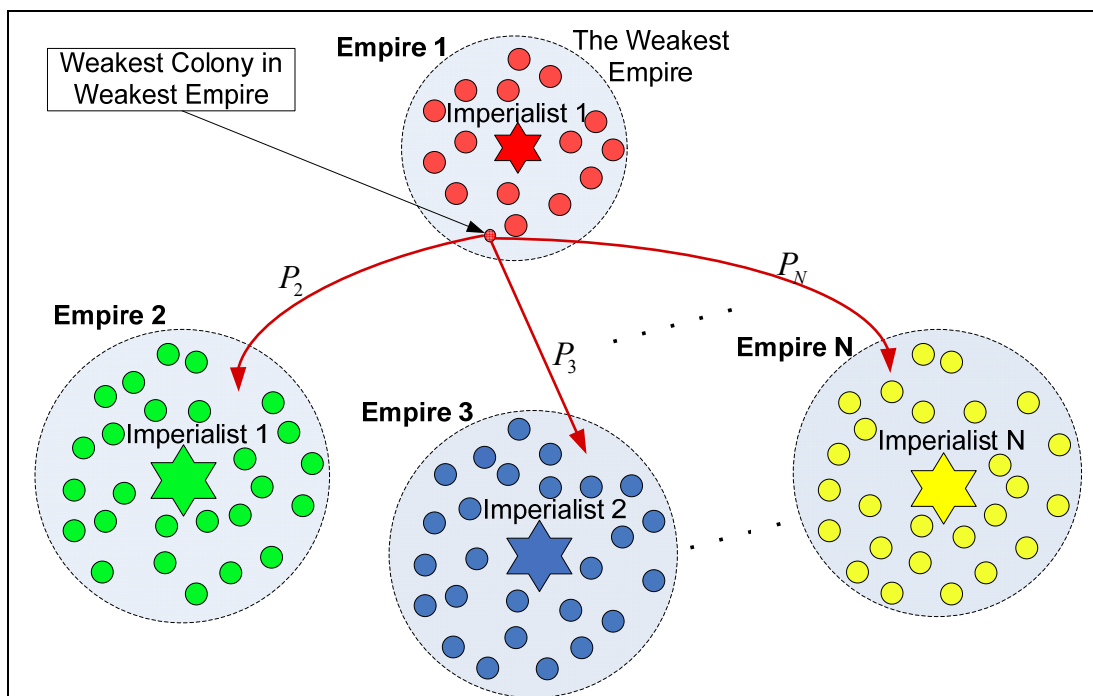
قدرت یک امپراطوری برابر است با قدرت کشور استعمارگر، به اضافه درصدی از قدرت کل مستعمرات آن. بدین ترتیب برای هزینه کل یک امپراطوری داریم.

$$T.C._n = Cost(imperialist_n) + \xi \text{ mean}\{Cost(colonies of empire_n)\} \quad (3-10)$$

که در آن $T.C._n$ هزینه کل امپراطوری n ام و ξ عددی مثبت است که معمولاً بین صفر و یک و نزدیک به صفر در نظر گرفته می‌شود. کوچک در نظر گرفتن ξ ، باعث می‌شود که هزینه کل یک امپراطوری، تقریباً برابر با هزینه حکومت مرکزی آن (کشور امپریالیست)، شود و افزایش ξ نیز باعث افزایش تاثیر میزان هزینه مستعمرات یک امپراطوری در تعیین هزینه کل آن می‌شود. در حالت نوعی $\xi = 0.05$ در اکثر پیاده‌سازی به جوابهای مطلوبی منجر شده است.

۳-۳-۵ رقابت استعماری

همانگونه که قبلاً نیز بیان شد، هر امپراطوری‌ای که نتواند بر قدرت خود بیفزاید و قدرت رقابت خود را از دست بدهد، در جریان رقابت‌های امپریالیستی، حذف خواهد شد. این حذف شدن، به صورت تدریجی صورت می‌پذیرد. بدین معنی که به مرور زمان، امپراطوری‌های ضعیف، مستعمرات خود را از دست داده و امپراطوری‌های قویتر، این مستعمرات را تصاحب کرده و بر قدرت خویش می‌افزایند. برای مدل کردن این واقعیت، فرض می‌کنیم که امپراطوری در حال حذف، ضعیف‌ترین امپراطوری موجود است. بدین ترتیب، در تکرار الگوریتم، یکی یا چند تا از ضعیف‌ترین مستعمرات ضعیف‌ترین امپراطوری را برداشته و برای تصاحب این مستعمرات، رقابتی را میان کلیه امپراطوری‌ها ایجاد می‌کنیم. مستعمرات مذکور، لزوماً توسط قویترین امپراطوری، تصاحب نخواهند شد، بلکه امپراطوری‌های قویتر، احتمال تصاحب بیشتری دارند. شکل ۳-۹ شمای کلی این بخش از الگوریتم را نشان می‌دهد.



شکل ۳-۹: شمای کلی رقابت استعماری: امپراطوری‌های بزرگ‌تر، با احتمال بیشتری، مستعمرات امپراطوری‌های دیگر را تصاحب می‌کنند.

در این شکل امپراطوری شماره ۱ به عنوان ضعیف‌ترین امپراطوری در نظر گرفته شده و یکی از مستعمرات آن در معرض رقابت امپریالیستی قرار گرفته است و امپراطوری‌های ۲ تا N برای تصاحب آن با هم رقابت می‌کنند. برای مدل‌سازی رقابت میان امپراطوری‌ها برای تصاحب این مستعمرات، ابتدا احتمال تصاحب هر امپراطوری (که متناسب با قدرت آن امپراطوری می‌باشد)، را با در نظر گرفتن هزینه کل امپراطوری، به ترتیب زیر محاسبه می‌کنیم. ابتدا از روی هزینه کل امپراطوری، هزینه کل نرمالیزه شده آن را تعیین می‌کنیم.

$$N.T.C_n = \max_i \{T.C_i\} - T.C_n \quad (3-11)$$

در این رابطه $T.C_n$ ، هزینه کل امپراطوری n ام و $N.T.C_n$ نیز، هزینه کل نرمالیزه شده آن امپراطوری می‌باشد. هر امپراطوری که $T.C_n$ کمتری داشته باشد $N.T.C_n$ بیشتری خواهد داشت. در حقیقت $T.C_n$ معادل هزینه کل یک امپراطوری و $N.T.C_n$ معادل قدرت کل آن می‌باشد. امپراطوری با کمترین هزینه، دارای بیشترین قدرت است. با داشتن هزینه کل نرمالیزه شده، احتمال (قدرت) تصاحب

مستعمره رقابت، توسط هر امپراطوری، به صورت زیر محاسبه می‌شود.

$$p_{p_n} = \frac{N.T.C._n}{\sum_{i=1}^{N_{imp}} N.T.C._i} \quad (3-12)$$

با داشتن احتمال تصاحب هر امپراطوری، مکانیزمی همانند چرخه رولت^۱ در الگوریتم ژنتیک مورد نیاز است تا مستعمره مورد رقابت را با احتمال متناسب با قدرت امپراطوریه‌ها در اختیار یکی از آنها قرار دهد. در کنار امکان استفاده از چرخ رولت موجود، در این نوشتار مکانیزم جدیدی برای پیاده‌سازی این فرایند معرفی شده است که نسبت به چرخه رولت دارای هزینه محاسباتی بسیار کمتری می‌باشد. زیرا عملیات نسبتاً زیاد مربوط به محاسبه تابع توزیع جمعی احتمال^۲ را که در چرخه رولت مورد نیاز است را حذف می‌کند و فقط به داشتن تابع چگالی احتمال^۳ نیاز دارد. در ادامه مکانیزم مطرح شده برای اختصاص متناسب با احتمال مستعمره مورد رقابت به امپراطوری‌های رقیب توضیح داده می‌شود.

با داشتن احتمال تصاحب هر امپراطوری، برای اینکه مستعمرات مذکور را به صورت تصادفی، ولی با احتمال وابسته به احتمال تصاحب هر امپراطوری، بین امپراطوری‌ها تقسیم کنیم؛ بردار \mathbf{P} را از روی مقادیر احتمال فوق، به صورت زیر تشکیل می‌دهیم.

$$\mathbf{P} = [p_{p_1}, p_{p_2}, p_{p_3}, \dots, p_{p_{N_{imp}}}] \quad (3-13)$$

بردار \mathbf{P} دارای سایز $1 * N_{imp}$ می‌باشد و از مقادیر احتمال تصاحب امپراطوری‌ها تشکیل شده است. سپس بردار تصادفی \mathbf{R} ، همسایز با بردار \mathbf{P} را تشکیل می‌دهیم. آرایه‌های این بردار، اعدادی تصادفی با توزیع یکنواخت در بازه $[0,1]$ می‌باشند.

¹ Roulette Wheel

² Cumulative Distribution Function (CDF)

³ Probability Density Function (PDF)

$$\mathbf{R} = [r_1, r_2, r_3, \dots, r_{N_{imp}}] \quad (3-14)$$

$$r_1, r_2, r_3, \dots, r_{N_{imp}} \sim U(0,1)$$

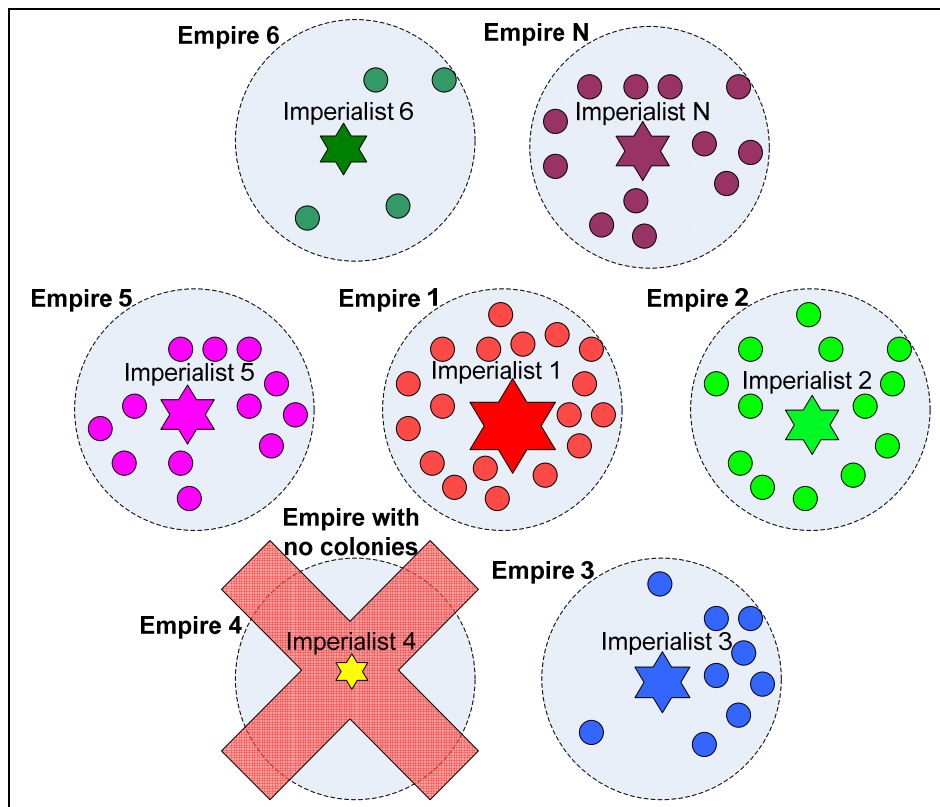
سپس بردار \mathbf{D} را به صورت زیر تشکیل می‌دهیم.

$$\begin{aligned} \mathbf{D} = \mathbf{P} - \mathbf{R} &= [D_1, D_2, D_3, \dots, D_{N_{imp}}] \\ &= [p_{p_1} - r_1, p_{p_2} - r_2, p_{p_3} - r_3, \dots, p_{p_{N_{imp}}} - r_{N_{imp}}] \end{aligned} \quad (3-15)$$

با داشتن بردار \mathbf{D} ، مستعمرات مذکور را به امپراطوری‌ای می‌دهیم که اندیس مربوط به آن در بردار \mathbf{D} بزرگتر از بقیه می‌باشد. امپراطوری‌ای که بیشترین احتمال تصاحب را داشته باشد، با احتمال بیشتری اندیس مربوط به آن در بردار \mathbf{D} ، بیشترین مقدار را خواهد داشت. عدم نیاز به محاسبه CDF باعث می‌شود که این مکانیزم نسبت به چرخه رولت با سرعت به مراتب بیشتری عمل کند. مکانیزم جدید مطرح شده نه تنها می‌تواند در اختصاص مستعمره به امپراطوری بر حسب احتمال تصاحب آنها مفید باشد، بلکه به عنوان یک مکانیزم انتخاب بر حسب احتمال می‌تواند جایگزین چرخه رولت در الگوریتم ژنتیک برای انتخاب والدین شود و سرعت اجرای عملیات در آن را تا حد زیادی افزایش دهد. با تصاحب مستعمره توسط یکی از امپراطوری‌ها، عملیات این مرحله از الگوریتم نیز به پایان می‌رسد.

۳-۳-۶ سقوط امپراطوری‌های ضعیف

همانگونه که بیان شد، در جریان رقابت‌های امپریالیستی، خواه ناخواه، امپراطوری‌های ضعیف به تدریج سقوط کرده و مستعمراتشان به دست امپراطوری‌های قوی‌تر می‌افتد. شروط متفاوتی را می‌توان برای سقوط یک امپراطوری در نظر گرفت. در الگوریتم پیشنهاد شده، یک امپراطوری زمانی حذف شده تلقی می‌شود که مستعمرات خود را از دست داده باشد. شکل ۳-۱۰ این مسئله را به خوبی نشان می‌دهد. در این شکل، امپراطوری شماره ۴ به علت از دست دادن کلیه مستعمراتش، دیگر قدرتی برای رقابت ندارد و باید از میان بقیه امپراطوری‌ها حذف شود.



شکل ۳-۱۰: سقوط امپراطوری ضعیف؛ امپراطوری شماره ۴، به علت از دست دادن کلیه مستعمراتش، دیگر قدرتی برای رقابت ندارد و باید از میان بقیه امپراطوری‌ها حذف شود.

۳-۳-۷ همگرایی

الگوریتم مورد نظر تا برآورده شدن یک شرط همگرایی، و یا تا اتمام تعداد کل تکرارها، ادامه می‌یابد. پس از مدتی، همه امپراطوری‌ها، سقوط کرده و تنها یک امپراطوری خواهیم داشت و بقیه کشورها تحت کنترل این امپراطوری واحد، قرار می‌گیرند. در این دنیای ایده‌آل جدید، همه‌ی مستعمرات، توسط یک امپراطوری واحد اداره می‌شوند و موقعیت‌ها و هزینه‌های مستعمرات، برابر با موقعیت و هزینه کشور امپریالیست است. در این دنیای جدید، تفاوتی، نه تنها، میان مستعمرات، بلکه میان مستعمرات و کشور امپریالیست، وجود ندارد. به عبارت دیگر، همه‌ی کشورها، در عین حال، هم مستعمره و هم استعمارگرند. در چنین موقعیتی رقابت امپریالیستی به پایان رسیده و به عنوان یکی از شروط توقف الگوریتم متوقف می‌شود. شبه کد مربوط به الگوریتم پیشنهادی در شکل ۳-۱۱، نشان داده شده است.

۱. چند نقطه تصادفی روی تابع انتخاب کرده و امپراطوری‌های اولیه را تشکیل بده.
۲. مستعمرات را به سمت کشور امپریالیست حرکت بده (سیاست همسان‌سازی).
۳. اگر مستعمره‌ای در یک امپراطوری، وجود داشته باشد که هزینه‌ای کمتر از امپریالیست داشته باشد؛ جای مستعمره و امپریالیست را با هم عوض کن.
۴. هزینه‌ی کل یک امپراطوری را حساب کن (با در نظر گرفتن هزینه‌ی امپریالیست و مستعمراتشان).
۵. یک مستعمره از ضعیف‌ترین امپراطوری انتخاب کرده و آن را به امپراطوری‌ای که بیشترین احتمال تصاحب را دارد، بده.
۶. امپراطوری‌های ضعیف را حذف کن.
۷. اگر تنها یک امپراطوری باقی مانده باشد، توقف کن و گرنه به ۲ برو.

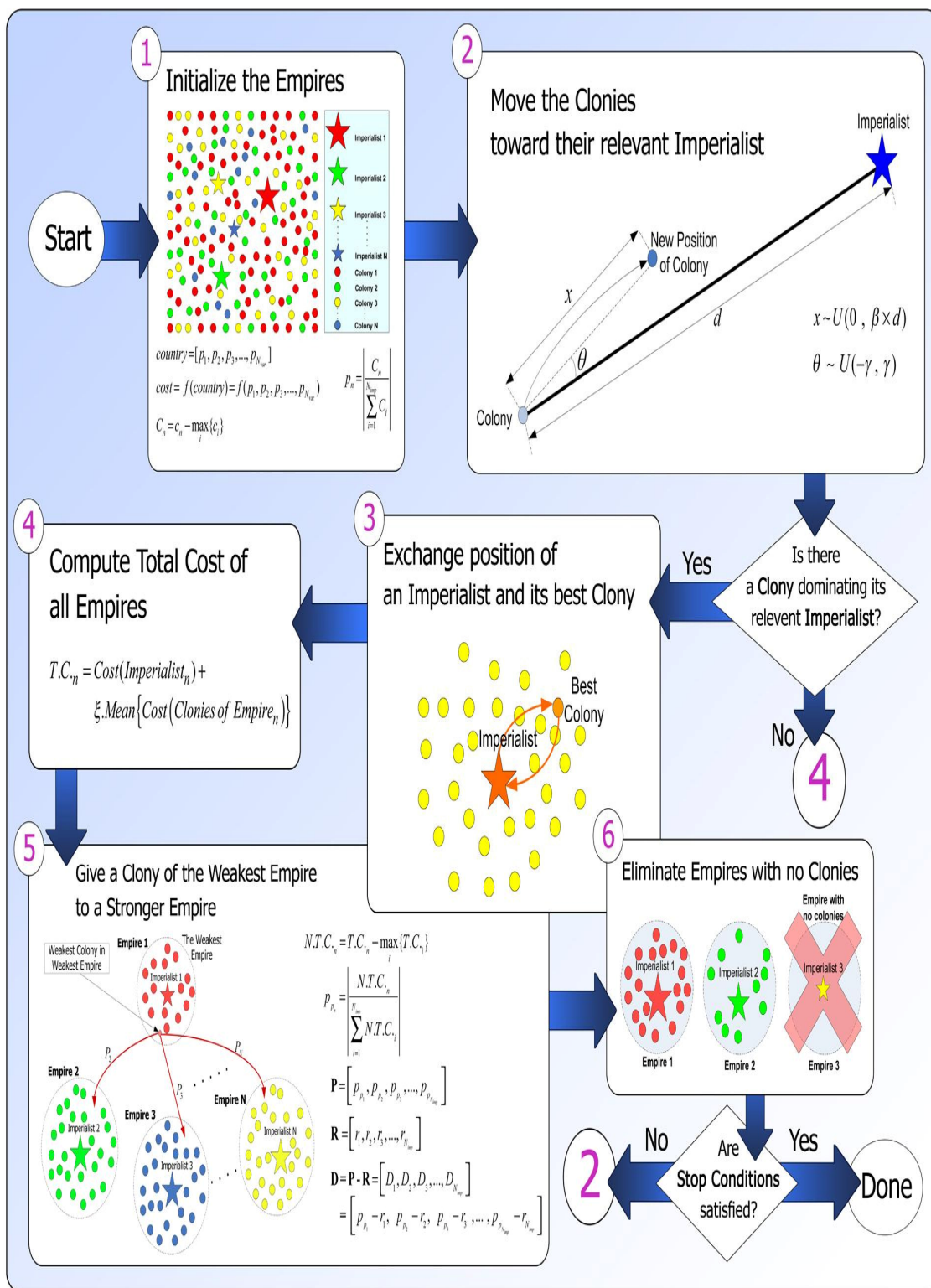
شکل ۳-۱۱: شبه کد مربوط به الگوریتم رقابت استعماری

شمای کلی الگوریتم به صورت گرافیکی در شکل ۳-۱۲ نیز نشان داده شده است. مطابق این شکل، الگوریتم با جمعیت اولیه تصادفی و تشکیل امپراطوری‌های اولیه آغاز شده و در یک چرخه سیاست جذب و رقابت امپریالیستی تکرار می‌شوند.

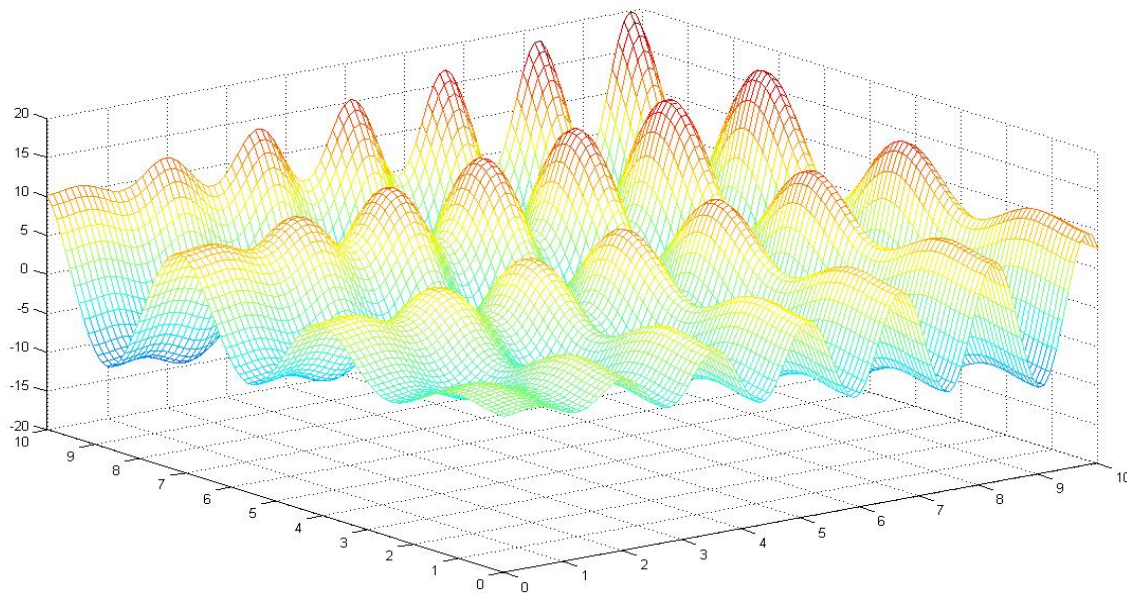
۳-۴ مثال کاربردی

در این بخش، الگوریتم معرفی شده را بر روی بعضی از توابع رایج که در مقایسه روش‌های مختلف بهینه‌سازی استفاده می‌شوند؛ پیاده‌سازی می‌کنیم. این توابع ۲ بعدی بهینه‌سازی، از مرجع [۳] استفاده شده‌اند و همه آن‌ها، مسائل مینیمم‌سازی هستند. جزئیات این توابع و نقطه بهینه و نیز یک نمای سه‌بعدی از آن‌ها در پیوست این فصل آمده است. از میان این توابع، روند بهینه‌سازی تابع مسئله‌ی G_1 بطور مشروح مورد بررسی قرار می‌گیرد اما در مورد بقیه، تنها به ذکر نتایج اکتفا می‌شود.

شکل ۳-۱۳ یک نمای سه‌بعدی از تابع مسئله G_1 را نشان می‌دهد. نقطه مینیمم مطلق این تابع در بازه‌ی $0 < x, y < 10$ در نقطه‌ی $(9.039, 8.668)$ قرار گرفته است و دارای مقداری برابر با -18.5547

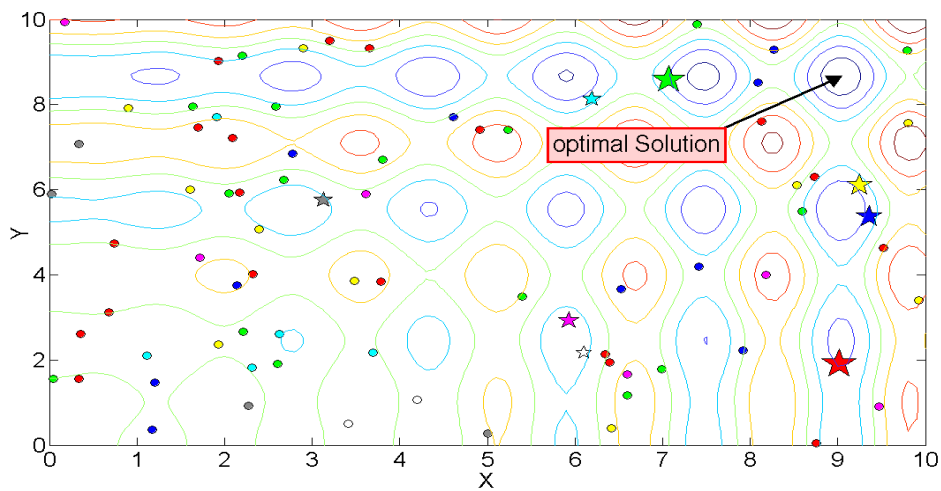


شکل ۳-۱: شمای کلی الگوریتم توسعه داده شده

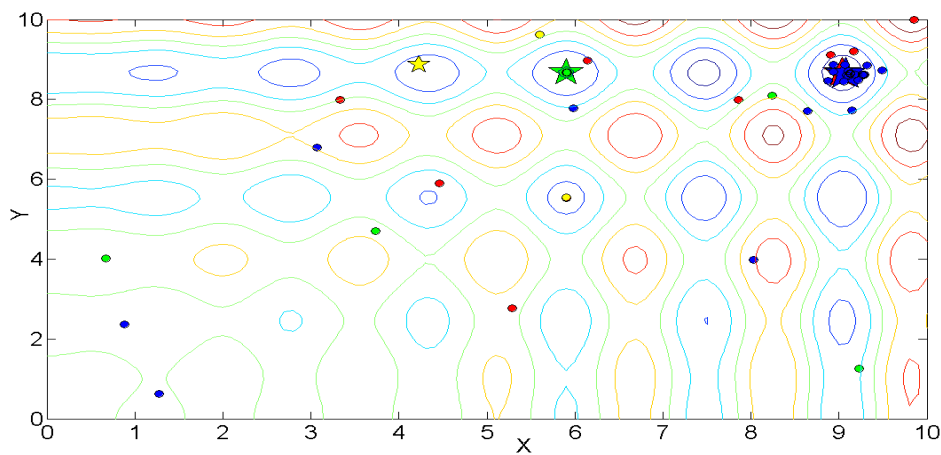


شکل ۳-۱۳: نمایش سه بعدی از تابع مسئله G_1

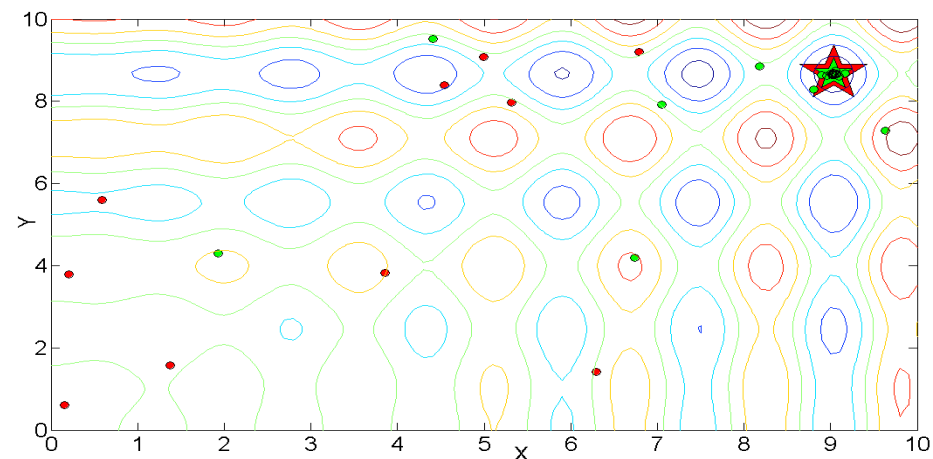
تعداد ۸۰ کشور اولیه‌ی مورد استفاده برای بهینه‌سازی، در شکل ۳-۱۴ به همراه یک نمای کانتور از تابع، نشان داده شده‌اند. ۸ تا از این کشورها به عنوان امپریالیست انتخاب شده و با کنترل بقیه‌ی ۷۲ کشور، امپراطوری‌های اولیه را تشکیل می‌دهند. امپریالیست‌ها با علامت ★ در رنگ‌های مختلف، نشان داده شده‌اند. مستعمرات هر امپراطوری نیز به همان رنگ امپریالیست و با علامت • مشخص شده‌اند. هر چه تعداد مستعمرات یک امپراطوری بیشتر باشد، کشور امپریالیست مربوط به آن با علامت ستاره بزرگتری، نمایش داده می‌شود. شکل‌های ۳-۱۵، ۳-۱۶ و ۳-۱۷، امپراطوری‌ها را به ترتیب، در نسل‌های ۱۰، ۳۰ و ۳۳ (همگرایی) نشان می‌دهند. همانگونه که در شکل ۳-۱۵ نیز نمایش داده شده است، در نسل ۱۰، چهار تا از امپراطوری‌ها سقوط کرده‌اند و چهار تا امپریالیست باقی‌مانده، به حواشی مینیمم‌های محلی تابع رسیده‌اند و حتی دو تا از آن‌ها، در ناحیه‌ی مقعر اطراف نقطه‌ی مینیمم مطلق قرار گرفته‌اند. در نسل ۳۰، تنها ۲ امپراطوری زنده‌اند و بقیه‌ی آن‌ها سقوط کرده‌اند. در نسل ۳۳، نیز همه‌ی امپراطوری‌ها به جز یکی، سقوط کرده و به یک دنیای تک قطبی رسیده‌ایم؛ دنیایی که کل آن، یک امپراطوری واحد را تشکیل می‌دهد و همه‌ی مستعمرات و حتی خود امپریالیست از موقعیت یکسانی برخوردار می‌باشند.



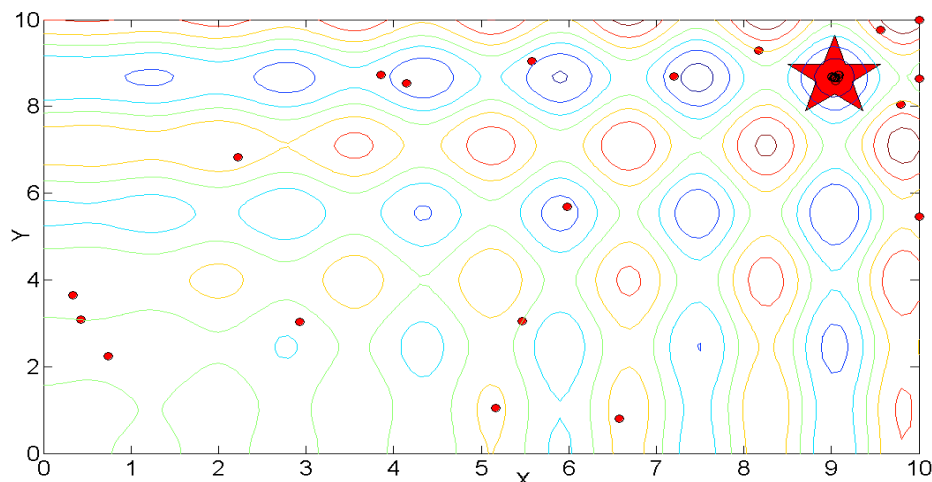
شکل ۳-۱۴: امپراطوری‌های اولیه؛ هر رنگ نمایش‌دهنده‌ی یک امپراطوری می‌باشد.



شکل ۳-۱۵: امپراطوری‌ها در نسل ۱۰ام؛ ۴ تا از امپراطوری‌ها باقی مانده‌اند.

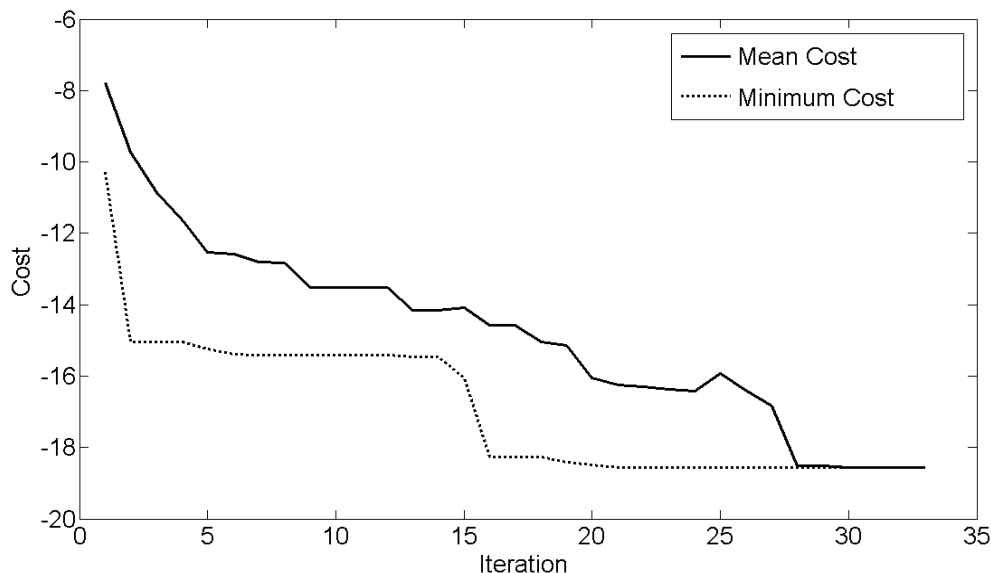


شکل ۳-۱۶: امپراطوری‌ها در نسل ۳۰ام؛ تنها دو امپراطوری باقی مانده‌اند.



شکل ۳-۱۷: امپراطوری‌ها در نسل ۳۳ام (همگرایی): تنها یک امپراطوری واحد داریم.

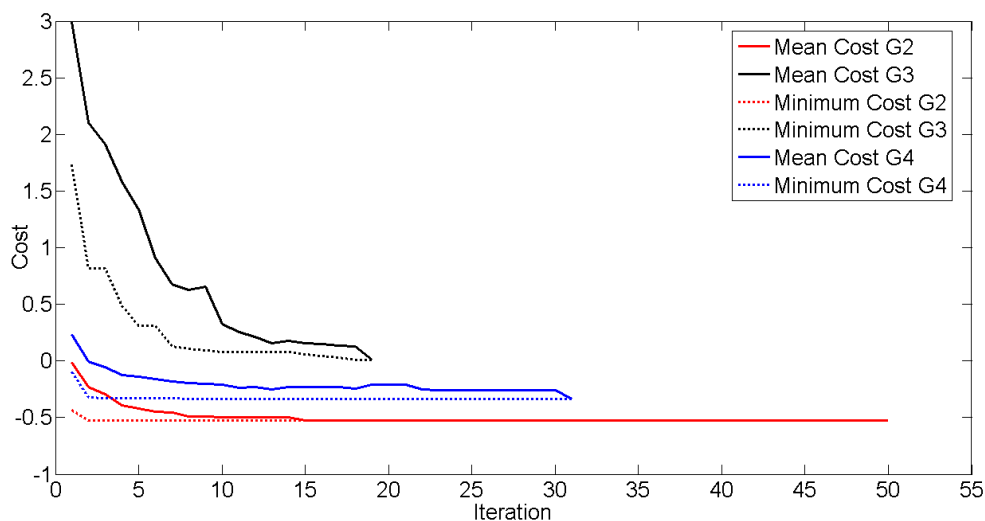
برای داشتن یک درک کلی از فرایند رقابت امپریالیستی، هزینه‌ی میانگین و مینیمم همه‌ی امپریالیست‌ها بر حسب نسل، در شکل ۳-۱۸ نشان داده شده است. مطابق این شکل، در طی رقابت امپریالیستی، مینیمم مطلق تابع در نسل حدود ۲۰ یافته شده است. اما تا نسل ۳۳، بقیه‌ی امپریالیست‌ها نیز در موقعیت خوبی بوده و هنوز قادر به رقابت هستند. اما با سقوط تک تک آن‌ها، در نسل ۳۳ام، تنها یک امپراطوری پابرجا است؛ امپراطوری‌ای که به نقطه‌ی مینیمم مطلق، زودتر نزدیک شد.



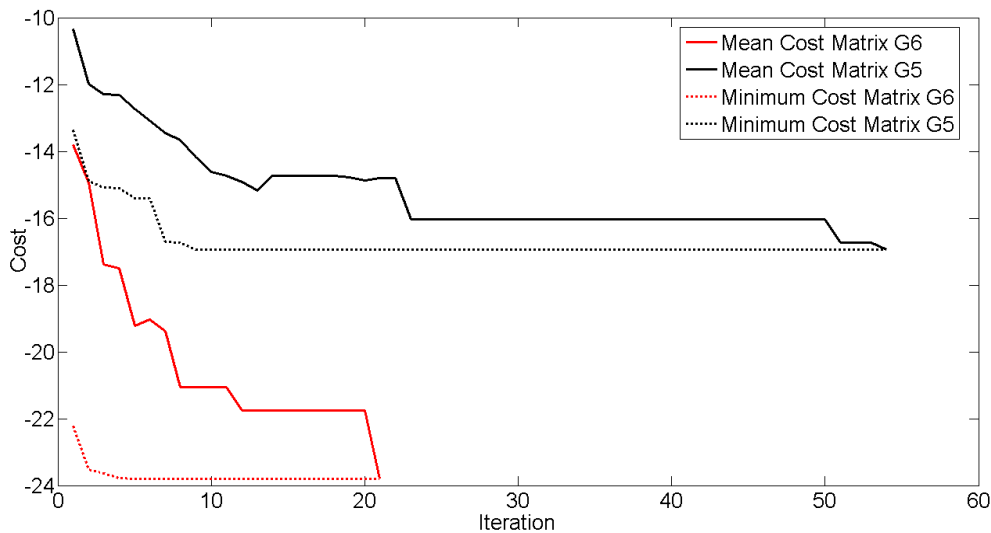
شکل ۳-۱۸: هزینه‌ی مینیمم و میانگین همه‌ی امپراطوری‌های مسئله G_1 بر حسب تکرار الگوریتم

الگوریتم معرفی شده برای یافتن مینیمم مطلق توابع مسائل $G_2 - G_6$ نیز به کار رفته است. تعداد

کل کشورها و نیز تعداد امپریالیست‌ها در مسئله G_3 به ترتیب، برابر ۱۵۰ و ۱۵، در مسئله G_3 به ترتیب برابر ۵۰ و ۵، و در مسائل G_4-G_6 به ترتیب برابر ۸۰ و ۸ است. شکل‌های ۳-۱۹ و ۳-۲۰، هزینه میانگین و مینیمم امپریالیست‌ها را در به ترتیب در مسائل G_2, G_3 و G_4 و نیز در مسائل G_5 و G_6 نشان می‌دهند.



شکل ۳-۱۹: هزینه‌ی مینیمم و میانگین همه‌ی امپراطوری‌های مسئله G_2, G_3 و G_4 ، بر حسب تکرار الگوریتم



شکل ۳-۲۰: هزینه‌ی مینیمم و میانگین همه‌ی امپراطوری‌های مسائل G_5 و G_6 بر حسب تکرار الگوریتم

۳-۵ نتیجه‌گیری

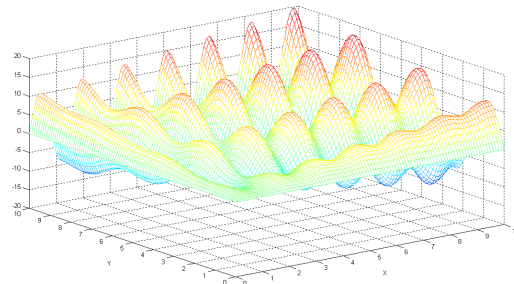
در این فصل، الگوریتمی برای بهینه‌سازی معرفی شد که بر پایه مدل‌سازی رقابت‌های امپریالیستی

می‌باشد. این الگوریتم، با تعدادی کشور اولیه شروع می‌شود. کشورها به دو دسته تقسیم می‌شوند؛ مستعمرات و کشور امپریالیست، که با هم تعدادی امپراطوری اولیه را تشکیل می‌دهند. رقابت میان امپراطوری‌ها، برای تصاحب مستعمرات همدیگر، هسته‌ی این الگوریتم را تشکیل می‌دهد و منجر به همگرایی کشورها، به مینیمم مطلق تابع هزینه می‌شود. در طی این رقابت، امپراطوری‌های ضعیف، به تدریج، سقوط کرده و در نهایت یک امپراطوری باقی می‌ماند که کل دنیا را کنترل می‌کند. آزمودن الگوریتم، توسط چندین تابع هزینه استاندارد، کارایی آن را در حل مسائل مختلف بهینه‌سازی، نشان داد.

۳-۶ توابع هزینه مورد استفاده

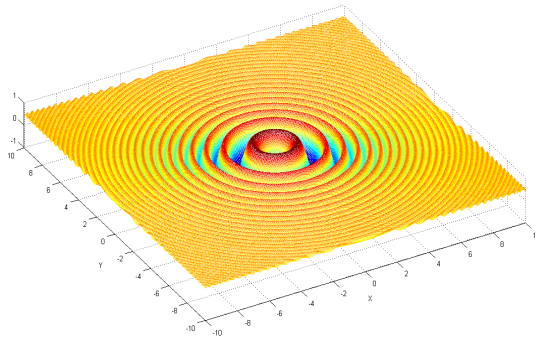
$$\text{Problem } G_1: f = x \cdot \sin(4x) + 1.1y \cdot \sin(2y) \quad (3-16)$$

$$0 < x, y < 10, \text{ minimum: } f(9.039, 8.668) = -18.5547$$



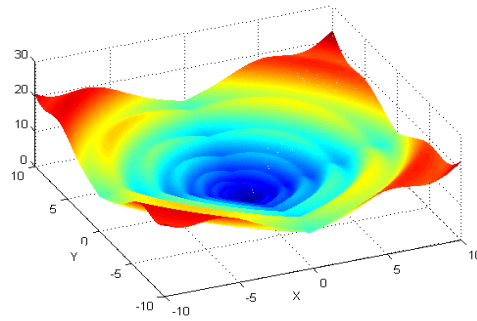
$$\text{Problem } G_2: f = 0.5 + \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{1 + 0.1(x^2 + y^2)} \quad (3-17)$$

$$-\infty < x, y < +\infty, \text{ minimum: } f(1.897, 1.006) = -0.5231$$



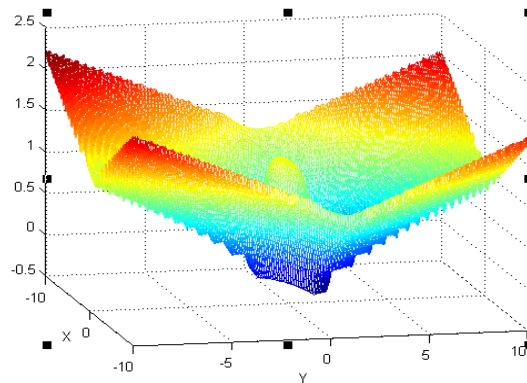
Problem G_3 : $f = (x^2 + y^2)^{0.25} \times \sin \left\{ 30 \left[(x + 0.5)^2 + y^2 \right]^{0.1} \right\} + |x| + |y|$ (٣-١٨)

$-\infty < x, y < +\infty$, minimum: $f(0, 0) = 0$



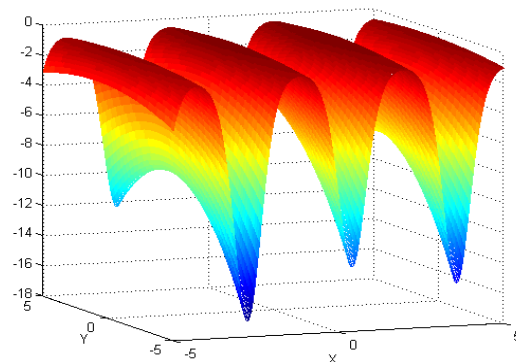
Problem G_4 : $f = J_0(x^2 + y^2) + 0.1|1 - x| + 0.1|1 - y|$ (٣-١٩)

$-\infty < x, y < +\infty$, minimum: $f(1, 1.6606) = -0.3356$



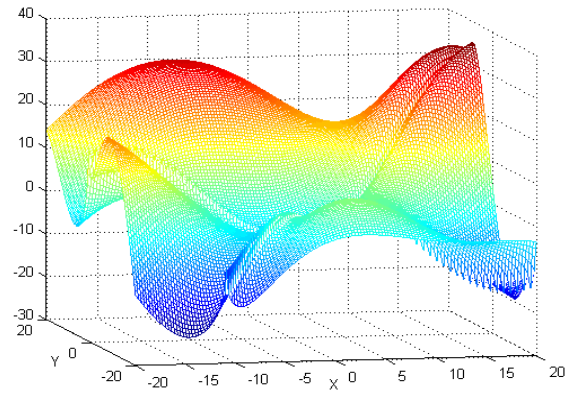
Problem G_5 : $f = -\exp(-0.2\sqrt{x^2 + y^2}) + 3(\cos 2x + \sin 2y)$ (٣-٢٠)

$-5 < x, y < 5$, minimum: $f(-2.7730, -5) = -16.947$



Problem G_6 $f = -x \sin(\sqrt{|x - (y + 9)|}) - (y + 9) \sin(\sqrt{|y + 0.5x + 9|})$ (3-21)

$-20 < x, y < 20$, *minimum* : $f(-14.58, -20) = -23.806$



فصل چهارم

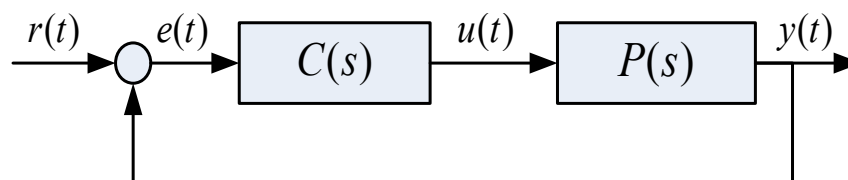
پیاده‌سازی های انجام شده

در این فصل به برخی از موارد کاربرد الگوریتم توسعه داده شده در انواع مختلف مسائل مهندسی اشاره می‌شود. در این راستا چند مورد از این کاربرد به تفصیل نسبی بیان می‌شوند و توضیحات در مورد سایر کاربردها به اشاره‌ای کوتاه و گذرا محدود می‌شود.

۴-۱ استفاده از الگوریتم معرفی شده برای طراحی یک کنترل‌کننده PID بهینه

به عنوان ساده‌ترین پیاده‌سازی انجام شده توسط الگوریتم معرفی شده، در این بخش، می‌خواهیم با استفاده از این الگوریتم، یک کنترل‌کننده PID برای سیستم کنترلی تک‌ورودی-تک‌خروجی، طراحی کنیم. هدف از بیان این پیاده‌سازی ساده آشنایی با چگونگی اعمال الگوریتم رقابت استعماری به مسائل عام در حوزه مهندسی می‌باشد.

کنترل‌کننده تناسبی-انتگرالی-مشتقی^۱ (PID Controller)، از دهه‌ی ۱۹۵۰ تا کنون، شناخته شده‌ترین و پرکاربردترین مکانیزم فیدبک بوده است و بطور وسیعی در کنترل پروسه‌های صنعتی مختلف، مورد استفاده قرار گرفته است. در کنار قابلیت‌های فراوان آن، این کنترل‌کننده، به سادگی می‌تواند برای اغلب پروسه‌های صنعتی، پیاده‌سازی شود. سیستم نشان داده شده در شکل ۴-۱ را در نظر می‌گیریم. در این شکل، $C(s)$ کنترل‌کننده و $P(s)$ سیستمی است که باید کنترل شود. هدف، طراحی کنترل‌کننده‌ای است که در آن، خروجی کل، دارای یک سری خصوصیات مطلوب تعریف شده، باشد.



شکل ۴-۱: بلوک دیاگرام یک سیستم کنترل فیدبک

^۱ Proportional-integral-derivative controller

۴-۱-۱ کنترل کننده PID

در حالت کنترل کننده PID، تابع تبدیل $C(s)$ به صورت زیر تعریف می شود.

$$C(s) = K_p + \frac{K_i}{s} + K_d s \quad (۴-۱)$$

که در آن K_p ، K_i و K_d به ترتیب ضرایب تناسبی، انتگرالی و مشتقی هستند. بدین ترتیب، خروجی کنترل کننده به صورت زیر خواهد بود.

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \dot{e}(t) \quad (۴-۲)$$

منظور از طراحی کنترل کننده PID برای یک سیستم، تعیین ضرایب K_p ، K_i و K_d است. بسته به کاربرد، عملکرد مطلوب خروجی سیستم، می تواند به شیوه های متفاوتی، بیان شود. در این بخش از نوشتار، چهار ویژگی مهم زمانی پاسخ یک سیستم، استفاده شده و معیار ما از مطلوبیت خروجی سیستم، به واسطه ی آن ها، تعریف خواهد شد. این ویژگی ها عبارتند از: زمان صعود^۱، زمان نشست^۲، بیشترین فراجهش^۳ و انتگرال قدر مطلق خطا^۴. در ادامه تعریف مختصری از هر یک از این ویژگی ها بیان شده و از روی آنها معیار بهینگی پاسخ خروجی، تعریف می شود.

۴-۱-۱-۱ زمان صعود

زمان صعود، زمانی است که در طی آن، پاسخ سیستم، از ۱۰ درصد مقدار نهایی خود، به ۹۰ درصد آن، می رسد. زمان صعود در شکل ۴-۲ نشان داده شده است.

^۱ Rise Time

^۲ Settling time

^۳ Max overshoot

^۴ Integral Absolute error (IAE)

۴-۱-۱-۲ زمان نشست

زمان نشست، به زمانی اطلاق می‌شود که بعد از آن زمان پاسخ سیستم، در فاصله ۲ درصدی از پاسخ نهایی‌اش، باقی می‌ماند. زمان نشست، در شکل ۴-۲ نشان داده شده است.

۴-۱-۱-۳ بیشترین فراجهش

ماکزیمم فراجهش، به صورت تفاضل مقدار دو پاسخ y_{ss} و y_{max} تعریف می‌شود. y_{ss} و y_{max} به ترتیب، مقدار بیشینه پاسخ، و حد نهایی آن را نشان می‌دهند. بدین ترتیب، برای ماکزیمم فراجهش، داریم:

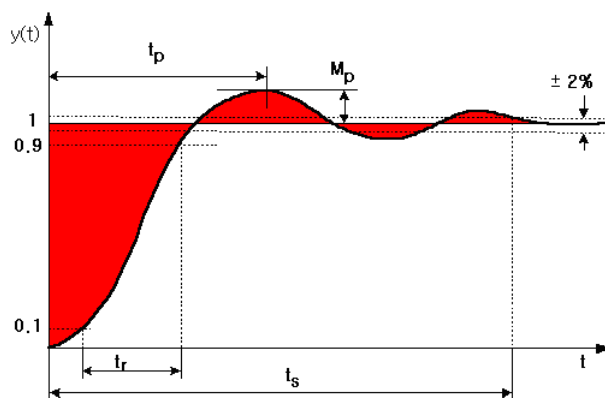
$$MaxOvershoot = y_{max} - y_{ss} \quad (۴-۳)$$

۴-۱-۱-۴ انتگرال قدر مطلق خطا

انتگرال قدر مطلق خطا به صورت زیر تعریف می‌شود.

$$IAE = \int_0^{\infty} |e(t)| dt \quad (۴-۴)$$

به خاطر پیاده‌سازی زمان گسسته، در محاسبه این انتگرال، حد بالای آن تا یک حد معین (معمولاً تا سه برابر زمان نشست)، در نظر گرفته می‌شود که جواب قابل قبولی برای این انتگرال به دست می‌دهد. انتگرال قدر مطلق خطا در شکل ۴-۲ نشان داده شده است.



شکل ۴-۲: زمان صعود (t_r)، زمان نشست (t_s)، بیشترین فراجهش (M_p) و انتگرال قدر مطلق خطا (مساحت ناحیه‌ی تیره رنگ)

۴-۱-۲ طراحی کنترل کننده PID بهینه توسط الگوریتم رقابت استعماری

در این بخش، الگوریتم رقابت استعماری، به مسئله طراحی یک کنترل کننده بهینه PID، برای یک سیستم معین، اعمال می شود. برای ایجاد امکان مقایسه نتایج با نتایج [۳۵]، برای طراحی این کنترل کننده، تابع هزینه ای به صورت زیر در نظر می گیریم.

$$f_{Total} = f_{MO} + f_{RT} + f_{ST} + f_{IAE} \quad (۴-۵)$$

در این تابع هزینه، f_{MO} ، f_{RT} ، f_{ST} و f_{IAE} ، به ترتیب، ماکزیمم فراجش، زمان صعود، زمان نشست و انتگرال قدر مطلق خطا هستند. کمینه کردن تابع هزینه فوق، به طور کلی باعث می شود که خروجی سیستم، هم از لحاظ پاسخ گذرا و هم از لحاظ پاسخ ماندگار، خروجی مطلوبی باشد. کم بودن ماکزیمم فراجش، زمان صعود و زمان نشست باعث می شوند که پاسخی داشته باشیم که به اندازه کافی سریع بوده و نوسانات قابل قبولی داشته باشد. کم بودن انتگرال قدر مطلق خطا نیز باعث می شود که خروجی سیستم، تفاوت کمی با خروجی مطلوب داشته باشد. همانگونه که در [۳۵] نیز استفاده شده است، تابع تبدیل سیستم به صورت زیر در نظر گرفته شده است.

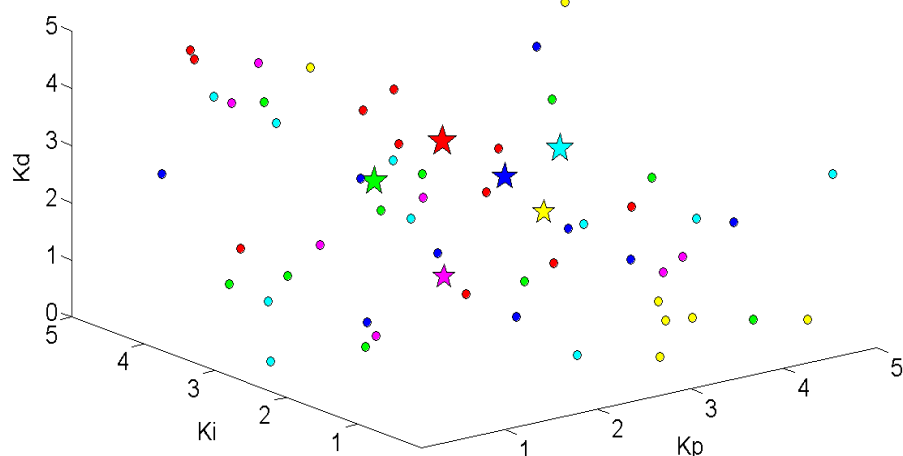
$$P(s) = \frac{4.228}{(s + 0.5)(s^2 + 1.64s + 8.456)} \quad (۴-۶)$$

تعداد کشورهای اولیه در نظر گرفته شده برای این مسئله ۶۰ می باشد. از این میان، ۶ کشور به عنوان امپریالیست، انتخاب شده اند تا امپراطوری های اولیه را شکل دهند. کنترل کننده PID نتیجه شده به صورت زیر می باشد.

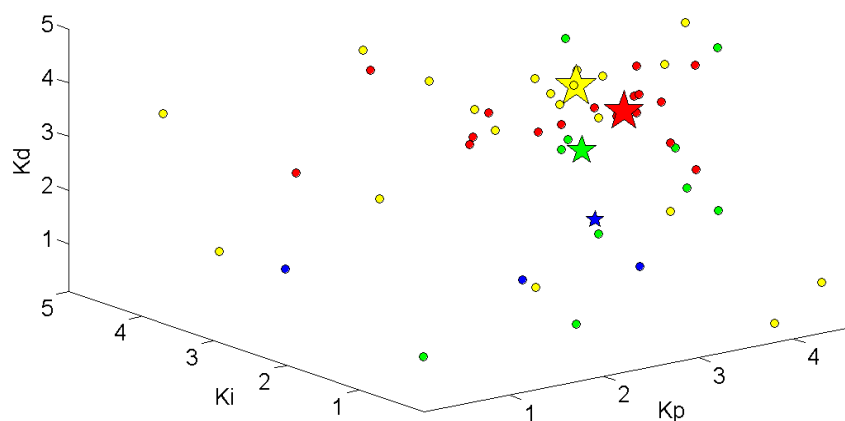
$$C(s) = 3.5273 + \frac{2.1671}{s} + 4.1560s \quad (۴-۷)$$

برای داشتن دید بهتری نسبت به فرایند رقابت امپریالیستی و چگونگی رسیدن به جواب مساله، هر ۶ امپراطوری اولیه، در شکل ۴-۳، نشان داده شده اند. کشورهای امپریالیست، با علائم ستاره به رنگ های مختلف که نماد امپراطوری های متفاوت هستند، نشان داده شده اند. مستعمرات هر امپریالیست نیز با علامت • و به همان رنگ امپریالیست، نشان داده شده اند. هر چه تعداد مستعمرات یک امپریالیست،

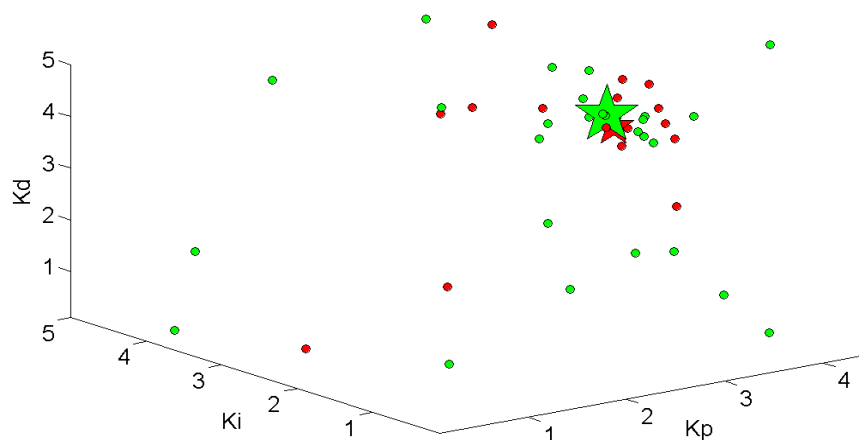
بیشتر باشد (هر چه امپراطوری اش بزرگتر باشد)، علامت ستاره مربوط به آن بزرگتر است. شکل های ۴-۴ و ۴-۵، امپراطوری ها را در نسل های ۳۰ و ۵۰ نشان می دهند. همانگونه که در این شکل نشان داده شده است، در نسل ۳۰م، دو امپراطوری ضعیف سقوط کرده و تنها ۴ امپراطوری باقی مانده و قادر به رقابت هستند. در نسل ۵۰م تنها ۲ تا از امپراطوری ها زنده اند و بصورت شدیدی برای حذف یکدیگر رقابت می کنند و این رقابت تا نسل ۷۳ (همگرایی الگوریتم) ادامه می یابد. در این نسل یکی از دو امپراطوری، دیگری را شکست داده و کل دنیا را کنترل می کند. فرایند رقابت استعماری می تواند در شکل ۴-۶ بازبینی شود. در این شکل، هزینه میانگین و مینیمم کل امپراطوری ها بر حسب نسل، نشان داده شده است.



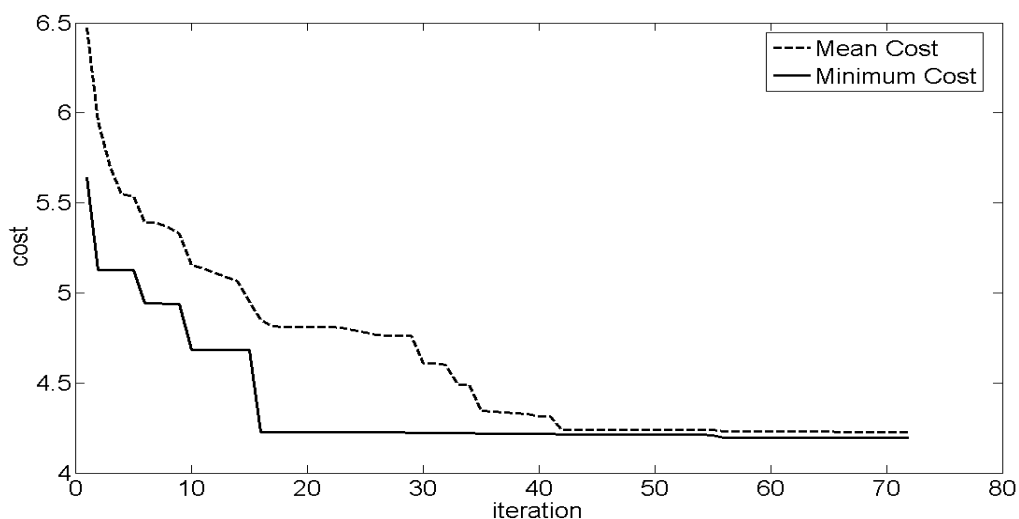
شکل ۴-۳: امپراطوری های اولیه در مسئله طراحی کنترل کننده PID



شکل ۴-۴: امپراطوری ها در نسل ۳۰م؛ دو امپراطوری ضعیف سقوط کرده و تنها ۴ امپراطوری باقی مانده اند.



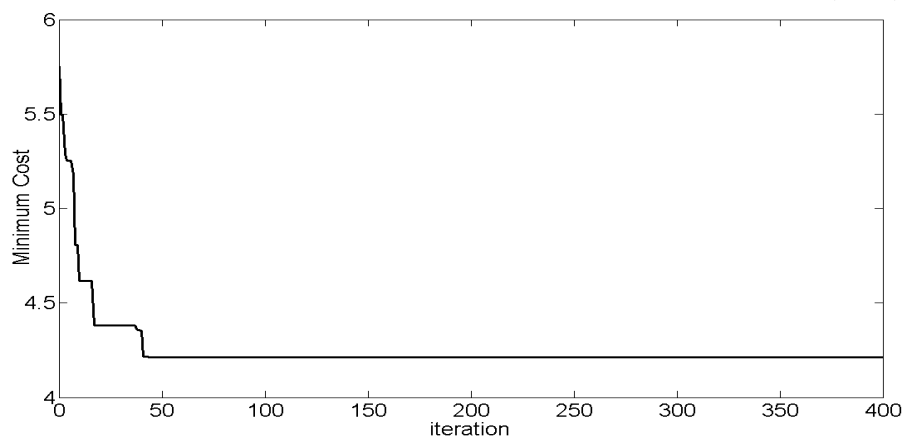
شکل ۴-۵: امپراطوری‌ها در نسل ۱۵۰؛ تنها ۲ تا از امپراطوری‌ها زنده‌اند و بصورت شدیدی برای حذف یکدیگر رقابت می‌کنند.



شکل ۴-۶: هزینه‌ی مینیمم و میانگین امپریالیست‌ها بر حسب تکرار الگوریتم، در مسئله طراحی کنترل‌کننده PID.

برای مقایسه نتایج، مسئله، بار دیگر توسط الگوریتم ژنتیک نیز حل شده است. تعداد جمعیت اولیه و نرخ جهش به ترتیب برابر ۶۰ و ۰/۲ در نظر گرفته شده‌اند. شکل ۴-۷ هزینه مینیمم جمعیت GA را بر حسب نسل، نشان می‌دهد. در جدول ۴-۱، هر چهار کنترلر حاصل از چهار متد مختلف، با هم مقایسه شده‌اند. این متدها عبارتند از: روش زیگلر نیکلز، کلونی مورچه‌ها، ژنتیک الگوریتم پیوسته و الگوریتم رقابت استعماری. نتایج دو روش اول از مرجع [۳۵] استخراج شده‌اند. نتایج، نشان می‌دهند که الگوریتم رقابت استعماری بهترین کنترل‌کننده را نتیجه داده است. همچنین به غیر از f_{MO} ، نتایج الگوریتم رقابت

استعماری، بهترین است. در مورد f_{MO} ، کنترل‌کننده حاصل از الگوریتم کلونی مورچه‌ها به نتیجه بهتری رسیده است. مقایسه نشان می‌دهد که کنترل‌کننده به دست آمده از الگوریتم ژنتیک، به نتیجه‌ای نزدیک به کنترلر حاصل از الگوریتم رقابت استعماری رسیده است. این شباهت در شکل ۴-۸ که پاسخ پله همه کنترلرها با هم رسم شده است، به راحتی قابل مشاهده است.



شکل ۴-۷: هزینه‌ی مینیمم جمعیت الگوریتم ژنتیک، بر حسب تکرار، در مسئله طراحی کنترل‌کننده PID.

جدول ۴-۱: پارامترهای کنترل‌کننده‌های مختلف طراحی شده، به همراه میزان هزینه‌های مختلف آن‌ها

| Method and Controller | f_{MO} | f_{RT} | f_{ST} | f_{LAE} | f_{Total} |
|---|----------|----------|----------|-----------|-------------|
| الگوریتم معرفی شده | | | | | |
| $K_P = 3.445$ $K_I = 2.166$ $K_D = 4.233$ | 0.1818 | 0.3 | 2.85 | 0.8614 | 4.1932 |
| الگوریتم ژنتیک پیوسته | | | | | |
| $K_P = 3.369$ $K_I = 2.125$ $K_D = 4.204$ | 0.1762 | 0.31 | 2.86 | 0.8669 | 4.2131 |
| بهینه‌سازی کلونی مورچه‌ها | | | | | |
| $K_P = 2.517$ $K_I = 2.219$ $K_D = 1.151$ | 0.156 | 0.62 | 4.99 | 0.8861 | 6.6525 |
| Zeigler Nichols | | | | | |
| $K_P = 2.19$ $K_I = 2.126$ $K_D = 0.565$ | 0.165 | 0.73 | 5.37 | 0.9595 | 7.2242 |

۴-۱-۳ نتیجه گیری

در این بخش، الگوریتم رقابت استعماری برای طراحی یک کنترل کننده بهینه PID، مورد استفاده قرار گرفت. کنترل کننده PID به گونه ای طراحی شد که مجموع زمان صعود، زمان نشست، بیشترین فراجهش و انتگرال قدرمطلق خطا را کمینه می کند. مقایسه میان نتایج حاصل از الگوریتم رقابت استعماری، الگوریتم ژنتیک، الگوریتم کلونی مورچه ها و روش زیگلر و نیکولز نشان می دهد که الگوریتم رقابت استعماری روش مناسبی برای بهینه سازی و در حالت خاص برای طراحی کنترل کننده PID می باشد.

۴-۲ استفاده از الگوریتم رقابت امپریالیستی برای طراحی کنترل کننده PID چند متغیره برای سیستم صنعتی ستون تقطیر

۴-۲-۱ مقدمه

طراحی کنترل کننده برای پروسه های چند متغیره^۱ در مقایسه با معادل تک متغیره شان، بسیار پیچیده تر می باشد. مسئله اصلی در سیستم های چندورودی-چندخروجی^۲ (MIMO)، توزیع بین ورودی و خروجی است [۳۶]. در چند دهه اخیر، طراحی کنترل کننده برای سیستم های چند متغیره، علایق تحقیقاتی زیادی را به خود جلب کرده است و روش های کنترل چند متغیره فراوانی ارائه شده اند [۳۷، ۳۸]. در میان روش های پیشنهادی برای کنترل سیستم های MIMO، کنترل کننده های PID، بیشترین استفاده را داشته اند. دلیل این امر به پیچیدگی کمتر، کارایی بالا و پیاده سازی آسان این کنترل کننده ها برمی گردد [۳۹، ۴۰، ۴۱]. همچنین، بعضی روش های جستجوی عددی نیز برای طراحی کنترل کننده های MIMO پیشنهاد شده اند که

^۱ Multi Variable Process

^۲ Multi-Input Multi Output

که در آن $g_{ij}(s)$ تابع تبدیل ورودی u_j به خروجی y_i است. همچنین بردارهای \mathbf{Y} ، \mathbf{Y}_d و \mathbf{U} و \mathbf{E} به صورت زیر تعریف می‌شوند.

$$\mathbf{Y}_d = [y_{d1} \quad y_{d2} \quad \cdots \quad y_{dn}]^T \quad (۴-۹)$$

$$\mathbf{Y} = [y_1 \quad y_2 \quad \cdots \quad y_n]^T \quad (۴-۱۰)$$

$$\mathbf{U} = [u_1 \quad u_2 \quad \cdots \quad u_n]^T \quad (۴-۱۱)$$

$$\mathbf{E} = \mathbf{Y}_d - \mathbf{Y} = [e_{11} \quad e_{22} \quad \cdots \quad e_{nn}]^T \quad (۴-۱۲)$$

کنترل‌کننده PID چندمتغیره در شکل ۱ نیز به صورت زیر می‌باشد.

$$\mathbf{C}(s) = \begin{bmatrix} c_{11}(s) & \cdots & c_{1n}(s) \\ \vdots & \ddots & \vdots \\ c_{n1}(s) & \cdots & c_{nn}(s) \end{bmatrix} \quad (۴-۱۳)$$

که در آن $c_{ij}(s)$ که $i, j \in \{1, 2, \dots, n\}$ به صورت زیر می‌باشد.

$$c_{ij}(s) = K_{Pij} + K_{Iij} \frac{1}{s} + K_{Dij} s \quad (۴-۱۴)$$

در این رابطه، K_{Pij} ، K_{Iij} و K_{Dij} به ترتیب گین‌های تناسبی، انتگرالی و مشتقی کنترل‌کننده $c_{ij}(s)$ هستند.

در طراحی کنترل‌کننده‌های PID، هدف تنظیم ضرایب K_{Pij} ، K_{Iij} و K_{Dij} به گونه‌ای است که خروجی دارای یک سری مشخصات مطلوب باشد. در حوزه‌ی زمان، معمولاً این ویژگی‌ها بر حسب فراجش، زمان صعود، زمان نشست و خطای حالت دائم، داده می‌شوند. دو نوع از معیار کارایی که معمولاً با هدف ردیابی خروجی، تعریف می‌شوند، انتگرال مجذور خطا^۱ (ISE) و انتگرال قدر مطلق

^۱ Integral Squared Error (ISE)

خطا^۱ (IAE) می‌باشند.

در طراحی کنترل‌کننده چند متغیره، یکی از اهداف عمده، حذف تزویج در ماتریس تابع انتقال $P(s)$ می‌باشد. یعنی کنترل‌کننده به گونه‌ای طراحی می‌شود که $y_i(t)$ ، خروجی مطلوب $y_{di}(t)$ را تعقیب کرده و پاسخ آن به $y_{dj}(t)$ را برای هر $i, j \in \{1, 2, \dots, n \mid i \neq j\}$ حذف می‌کند. با در نظر گرفتن هدف حذف تزویج، IAE به صورت زیر تعریف می‌شود.

$$IAE = \int_0^{\infty} \left(\sum_{i=1}^n |e_{ii}(t)| + \sum_{\substack{i=1 \\ i \neq j}}^n |e_{ij}(t)| \right) dt \quad (۴-۱۵)$$

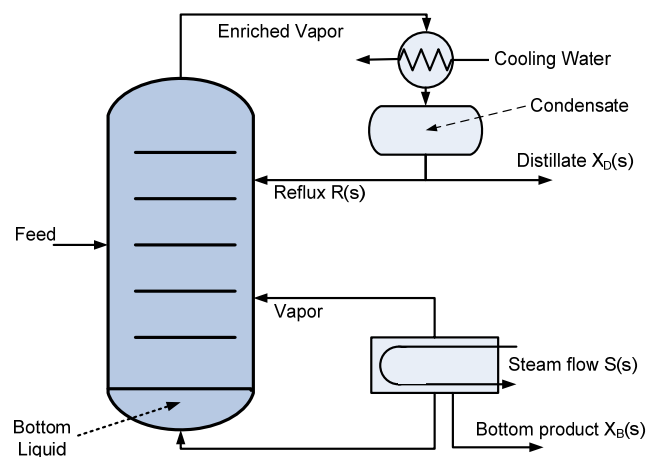
که در آن $|e_{ii}(t)|$ خطای مطلق خروجی $y_i(t)$ در تعقیب ورودی $y_{di}(t)$ بوده و $|e_{ij}(t)|$ قدر مطلق خطای ناشی از تاثیر ورودی $y_{dj}(t)$ روی خروجی $y_i(t)$ برای $(i \neq j)$ است. منبع $|e_{ij}(t)|$ وجود مشکل تزویج است. هدف، طراحی کنترل‌کننده‌ای است که تا حد امکان، خروجی‌های مطلوب را دنبال کرده و تزویج پروسه کنترل شده را از بین می‌برد. بدین منظور، با استفاده از IAE به عنوان معیاری برای شایستگی، در این فصل، پارامترهای بهینه یک کنترل‌کننده PID را برای سیستم ستون تقطیر، تنظیم کرده و نتایج به دست آمده از الگوریتم رقابت استعماری و الگوریتم ژنتیک را با نتایج بدست آمده در [۴۵] مقایسه می‌کنیم.

۴-۲-۳ نتایج شبیه‌سازی

در این بخش یک کنترل‌کننده چندمتغیره PID، برای یک سیستم چند ورودی-چند خروجی، طراحی می‌کنیم. سیستم مورد بررسی، یک مدل نوعی از فرایند ستون تقطیر [۴۵] است. یک شماتیک ساده از سیستم ستون تقطیر^۲ (DCS) در شکل ۴-۱۰ نشان داده شده است.

^۱ Integral Absolute Error (IAE)

^۲ Distillation Column System (DCS)



شکل ۴-۱۰: شمای ساده‌ای از فرایند تقطیر شیمیایی

تابع تبدیل ماتریسی DCS به صورت زیر در نظر گرفته می‌شود [۴۶].

$$\begin{bmatrix} X_D(s) \\ X_B(s) \end{bmatrix} = \begin{bmatrix} \frac{12.8e^{-s}}{1+16.7s} & \frac{-18.9e^{-3s}}{1+21s} \\ \frac{6.6e^{-7s}}{1+10.9s} & \frac{-19.4e^{-3s}}{1+14.4s} \end{bmatrix} \begin{bmatrix} R(s) \\ S(s) \end{bmatrix} \quad (۴-۱۶)$$

که در آن $X_D(s)$ و $X_B(s)$ درصد متانول، به ترتیب، در محصول تقطیر^۱ و در محصولات خروجی هستند. همچنین $R(s)$ و $S(s)$ به ترتیب، نرخ جریان شار^۲ و نرخ جریان بخار^۳ در بازجوشاننده^۴ هستند.

DCS یک سیستم MIMO نوعی 2×2 است که برهم‌کنش شدیدی بین دو جفت ورودی و خروجی آن وجود دارد. چهار تابع تبدیل در نظر گرفته شده برای این سیستم، دارای دینامیک مرتبه یک بوده و تاخیر زیادی در هرکدام از آن‌ها، وجود دارد. هدف کنترلی، تعقیب خروجی‌های y_1 و y_2 توسط

¹ distillate

² reflux flow rate

³ steam flow rate

⁴ reboiler

ورودی‌های کنترلی $y_{1d}=y_{2d}=1$ و همچنین حذف کاپلینگ در پروسه کنترل شده، تا حد ممکن، می‌باشد. در [۴۵] یک کنترل‌کننده PID برای این سیستم، با استفاده از روش decentralized relay feedback (DRF) طراحی شده است. در این مرجع، عناصر قطری و غیر قطری این کنترلر به ترتیب به فرم PI و PID طراحی شده‌اند. این کنترلر به صورت زیر می‌باشد.

$$C(s) = \begin{bmatrix} 0.184 + 0.0469 \frac{1}{s} & 0.0102 - 0.0229 \frac{1}{s} + 0.0082s \\ 0.0674 + 0.0159 \frac{1}{s} - 0.0536s & 0.066 - 0.0155 \frac{1}{s} \end{bmatrix} \quad (۴-۱۷)$$

برای مقایسه نتایج الگوریتم رقابت استعماری و روش DRF، کنترل‌کننده PID برای سیستم (۱۶-۴) به صورت زیر در نظر گرفته شده است.

$$C(s) = \begin{bmatrix} K_{P11} + K_{I11} \frac{1}{s} & K_{P12} + K_{I12} \frac{1}{s} + K_{D12}s \\ K_{P21} + K_{I21} \frac{1}{s} + K_{D21}s & K_{P22} + K_{I22} \frac{1}{s} \end{bmatrix}. \quad (۴-۱۸)$$

یعنی، عناصر قطری و غیر قطری کنترل‌کننده طراحی شده به ترتیب، به فرم PI و PID خواهند بود. بنابراین مسئله طراحی، یک مسئله بهینه‌سازی ۱۰ بعدی، برای تعیین ضرایب بهینه

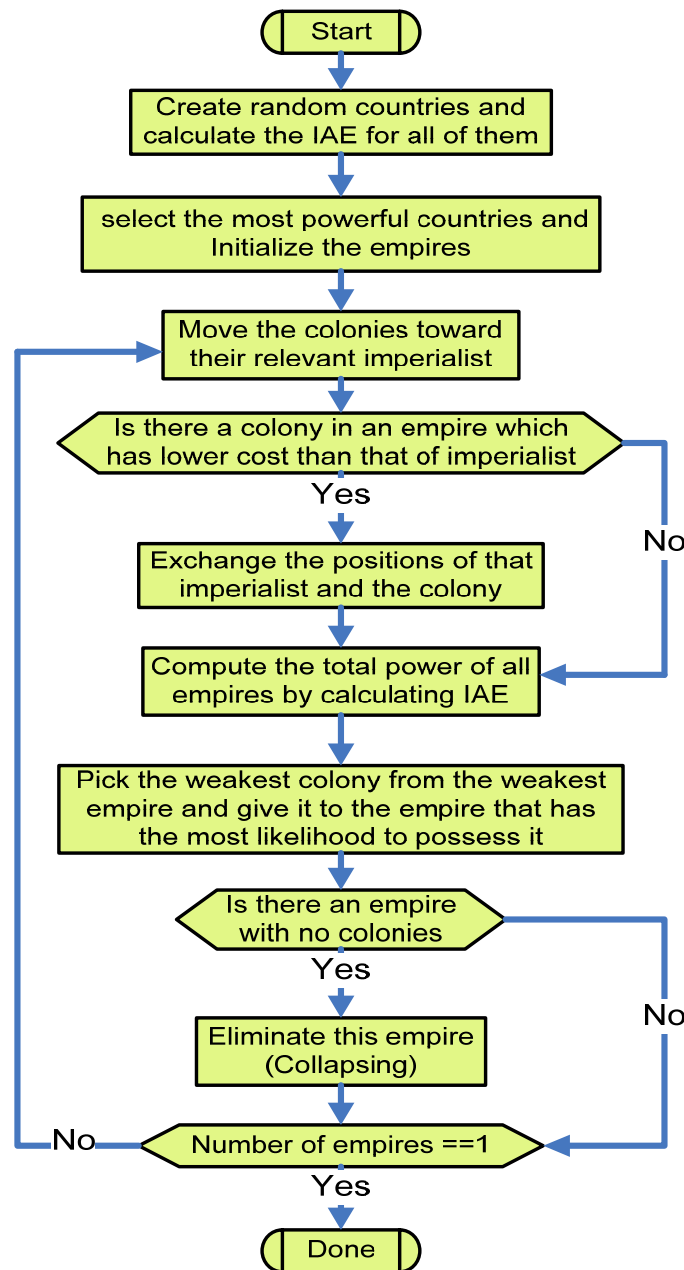
$$[K_{P11} \ K_{I11} \ K_{P12} \ K_{I12} \ K_{D12} \ K_{P21} \ K_{I21} \ K_{D21} \ K_{P22} \ K_{I22}]$$

برای مینیمم کردن تابع هزینه زیر می‌باشد.

$$IAE = \int_0^\infty |e_{11}(t)| dt + \int_0^\infty |e_{12}(t)| dt + \int_0^\infty |e_{21}(t)| dt + \int_0^\infty |e_{22}(t)| dt \quad (۴-۱۹)$$

$$= IAE_{11} + IAE_{12} + IAE_{21} + IAE_{22}$$

شکل ۴-۱۱ فلوچارت الگوریتم رقابت استعماری که به مسئله طراحی یک کنترل‌کننده PID برای سیستم ستون تقطیر، اعمال شده است؛ را نشان می‌دهد.



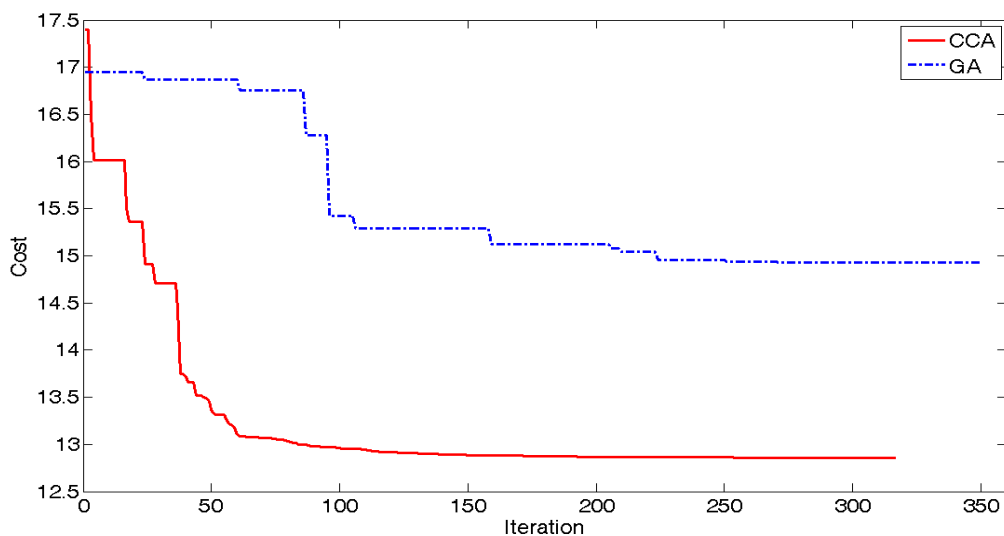
شکل ۴-۱۱: فلوجارت الگوریتم رقابت استعماری مورد استفاده برای طراحی یک کنترل کننده PID برای سیستم ستون تقطیر

در ICA، تعداد کشورهای اولیه و تعداد امپراطوری‌های اولیه، به ترتیب برابر ۲۰۰ و ۱۵ در نظر گرفته شده‌اند. همچنین β و γ به ترتیب روی ۲ و ۰/۵ تنظیم شده‌اند. تعداد تکرارهای توقف الگوریتم برابر ۳۵۰ در نظر گرفته شده است اما در تکرار ۳۱۷، الگوریتم با رسیدن به مقدار هزینه IAE برابر با

۱۲/۸۵۴۹ متوقف شده است. این به این خاطر است که در تکرار ۳۱۷، رقابت استعماری به مرحله‌ای رسیده است که تنها یک امپراطوری پابرجاست و رقابت امپریالیستی با این تعداد تکرار و با از بین رفتن کلیه امپراطوری‌های رقیب، خاتمه یافته است.

همچنین الگوریتم ژنتیک نیز برای تنظیم ضرایب کنترل‌کننده PID برای این سیستم، به کار رفته است. تعداد جمعیت اولیه برابر ۲۰۰ و نرخ‌های انتخاب و جهش برابر ۵۰٪ و ۳۰٪ در نظر گرفته شده‌اند. هزینه کمینه جمعیت‌های الگوریتم ژنتیک در تکرار حدود ۳۵۰ به مقدار نهایی خود، یعنی ۱۴/۹۳۳۴ رسید. قابل ذکر است که برای رسیدن به نتایج بهتر، جمعیت اولیه در الگوریتم ژنتیک و الگوریتم رقابت استعماری در یک ابرمکعب به مرکز ضرایب کنترل‌کننده به دست آمده از روش DRF، پخش شده‌اند.

شکل ۴-۱۲ هزینه مینیمم ICA و GA را بر حسب تکرار نسل نشان می‌دهد. همانگونه که در این شکل نیز نشان داده شده است، حد نهایی همگرایی ICA برابر ۱۲/۸۵۴۹ است که کمتر از مقدار معادل آن در GA یعنی ۱۴/۹۳۳۴ است. همچنین نرخ همگرایی الگوریتم رقابت استعماری (ICA) بسیار بیشتر از الگوریتم ژنتیک (GA) می‌باشد.



شکل ۴-۱۲: هزینه مینیمم ICA و GA را بر حسب تکرار نسل

پارامترهای کنترل‌کننده PID و هزینه IAE بدست آمده از هر یک از روش‌های ICA، GA و DRF در جداول ۴-۲ و ۴-۳ نشان داده شده‌اند. مطابق جدول ۴-۳، کنترلر به دست آمده از GA به

کمترین هزینه‌های IAE_{11} و IAE_{12} رسیده است. یعنی در مقایسه با ICA و DRF، با استفاده از کنترلر بدست آمده از GA، خروجی اول، ورودی اول را به بهترین نحو ممکن دنبال می‌کند و تا حد زیادی از ورودی دوم، جدا شده است. همچنین با در نظر گرفتن IAE_{12} ، کنترلر بدست آمده از DRF، کارایی بهتری نسبت به کنترلر به دست آمده از الگوریتم رقابت استعماری دارد. یعنی DRF در جدا کردن خروجی اول از ورودی دوم نسبت به ICA، بهتر عمل می‌کند. اما در مورد ورودی دوم، کنترل‌کننده بدست آمده از ICA، بهترین کنترل‌کننده می‌باشد. زیرا هزینه‌های متناظر IAE_{21} و IAE_{22} کمترین بوده و در نتیجه بهترین ردیابی را داشته و کمترین تزویج را نتیجه می‌دهد. همچنین با در نظر گرفتن هزینه کل IAE، کنترلر حاصل از ICA، به طور کلی، بهترین کنترل‌کننده است.

جدول ۴-۲: مقادیر پارامترهای کنترل‌کننده‌های به دست آمده از اعمال الگوریتم رقابت استعماری، الگوریتم ژنتیک و

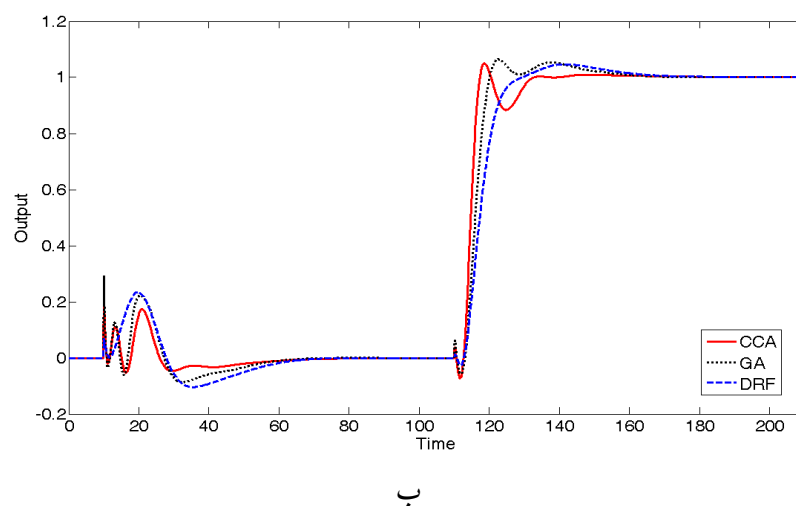
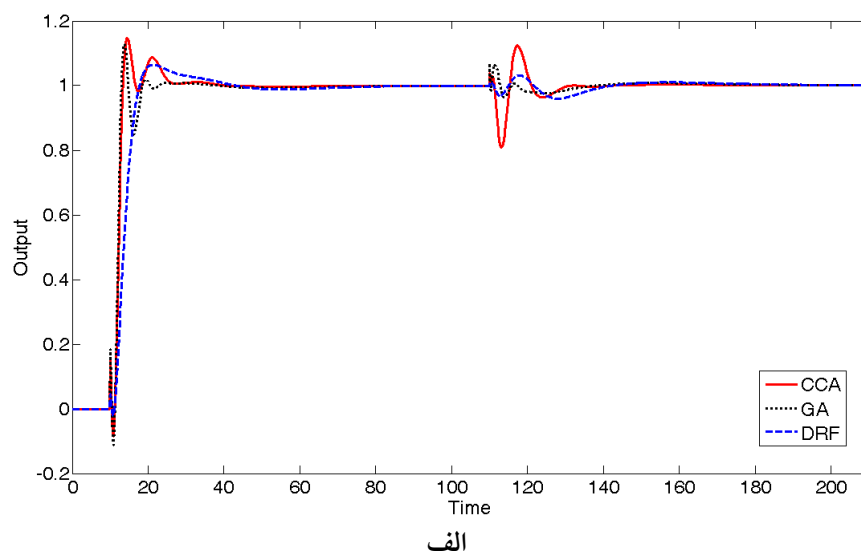
روش DRF

| روش | CCA | GA | DRF |
|-----------|---------|---------|---------|
| K_{P11} | 0.4133 | 0.2023 | 0.184 |
| K_{I11} | 0.0967 | 0.0738 | 0.0469 |
| K_{P12} | -0.1408 | 0.0002 | -0.0102 |
| K_{I12} | -0.0317 | -0.0464 | 0.0229 |
| K_{D12} | 0.0605 | 0.0044 | 0.0082 |
| K_{P21} | -0.0347 | -0.0654 | -0.0674 |
| K_{I21} | 0.0328 | 0.0252 | 0.0159 |
| K_{D21} | -0.2215 | -0.0372 | -0.0537 |
| K_{P22} | -0.1708 | -0.0960 | -0.0660 |
| K_{I22} | -0.0206 | -0.0322 | -0.0155 |

جدول ۴-۳: بخش‌های مختلف تابع هزینه به دست آمده از روش‌های ICA، GA و DRF

| روش | IAE_{11} | IAE_{12} | IAE_{21} | IAE_{22} | IAE |
|-----|------------|------------|------------|------------|---------|
| ICA | 3.2152 | 1.2750 | 2.2864 | 6.0784 | 12.8549 |
| GA | 3.0229 | 0.7830 | 3.4988 | 7.6288 | 14.9334 |
| DRF | 4.9551 | 1.0614 | 4.4166 | 9.0709 | 19.5040 |

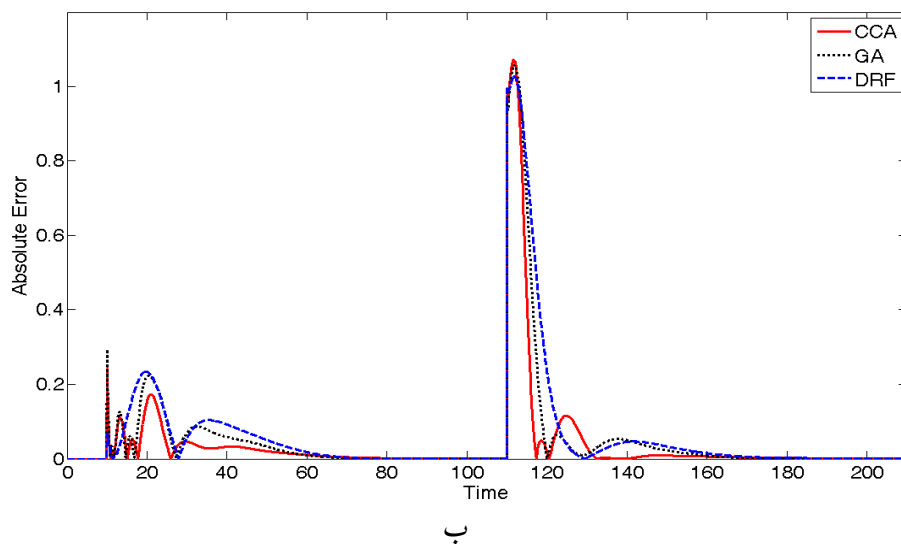
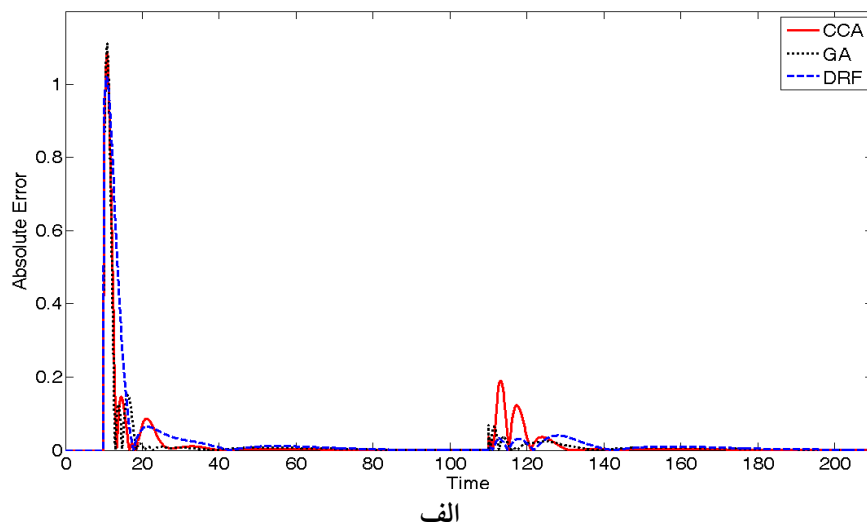
شکل ۴-۱۳، خروجی سیستم کنترل شده ستون تقطیر، توسط کنترلرهای مختلف بدست آمده از



شکل ۴-۱۳: پاسخ سیستم ستون تقطیر به تاخیرهای متفاوت در ورودی پله (الف: ورودی اول، ب: ورودی دوم).

روشهای ICA، GA و DRF را برای ورودی پله، نشان می‌دهد. برای داشتن دید بهتر به تزویج ایجاد شده توسط کنترلرهای مختلف؛ ورودی‌های پله، با تأخیرهای ۱۰ و ۱۱۰ ثانیه‌ای اعمال شده و خروجی‌ها باهم رسم شده‌اند.

همچنین برای داشتن درک بهتری از مقادیر جدول ۴-۳، قدر مطلق خطاهای ردیابی و تزویج برای هر دو خروجی اول و دوم در شکل زیر نشان داده شده‌اند.



شکل ۴-۱۴: قدر مطلق خطای خروجی فرایند ستون تقطیر به تأخیرهای مختلف در ورودی پله. (الف: قدر مطلق خطای ورودی اول. ب: قدر مطلق خطای ورودی دوم)

با در نظر گرفتن شکل ۴-۱۴، همانگونه که قبلاً نیز بیان شد، مشاهده می‌شود که خطای جداسازی خروجی اول از ورودی دوم با استفاده از کنترلر بدست آمده از ICA بیشترین است. اما در حالت کلی، این کنترلر، عملکرد ضعیف خود در مورد خطای IAE_{12} را با داشتن کمترین هزینه‌های IAE_{21} و IAE_{22} و هزینه‌ی IAE_{11} در حدود هزینه GA، جبران می‌کند.

۴-۲-۴ نتیجه گیری

در این بخش الگوریتم رقابت استعماری و الگوریتم ژنتیک به مسئله طراحی یک کنترل کننده PID چند متغیره، برای فرایند ستون تقطیر، اعمال شدند. هدف طراحی، تعیین ضرایب کنترل کننده به گونه ای بود که انتگرال قدر مطلق خطا، کمینه شود. نتایج، حاکی از آن بودند که الگوریتم رقابت استعماری، بیشترین نرخ همگرایی را داشته و همچنین، خروجی های کنترل شده در این روش، کارایی بهتری از خود نشان داده و ویژگی های مطلوب تری نسبت به GA و DRF داشتند.

۴-۳ الگوریتم رقابت استعماری؛ ابزاری برای یافتن نقطه تعادل نش

در این بخش کاربرد الگوریتم رقابت استعماری در نظریه بازیها و یافتن نقطه تعادل نش ارائه می شود. نتایج بدست آمده از اعمال الگوریتم معرفی شده (ICA)، با نتایج حاصل از الگوریتم ژنتیک مقایسه می گردد.

۴-۳-۱ یک بازی غیر خطی استاتیک ساده

در این بخش روش مورد استفاده بر روی یک مثال ساده بازی مورد بررسی قرار می گیرد. یک بازی با دو بازیکن با توابع هزینه معرفی شده زیر را در نظر بگیرید.

$$f_1(x_1, x_2) = (x_1 - 1)^2 + (x_1 - x_2)^2 \quad (۴-۲۰)$$

$$f_2(x_1, x_2) = (x_2 - 3)^2 + (x_1 - x_2)^2 \quad (۴-۲۱)$$

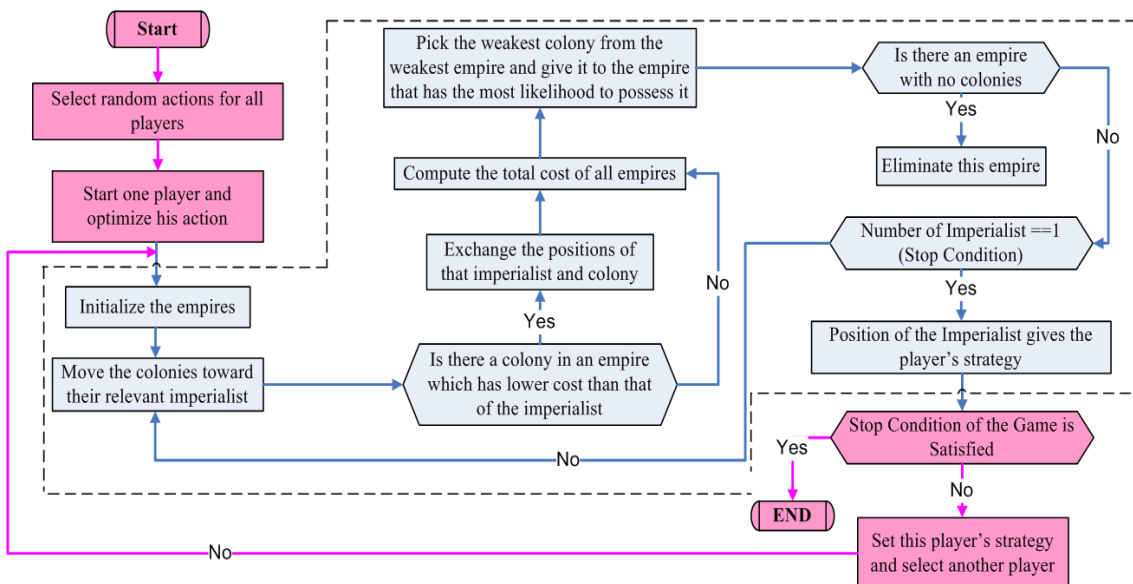
با استفاده از روش تحلیلی، نقطه تعادل نش این بازی به صورت زیر بدست می آید.

$$\begin{cases} \frac{\partial f_1(x_1, x_2)}{\partial x_1} = 0 \\ \frac{\partial f_2(x_1, x_2)}{\partial x_2} = 0 \end{cases} \Rightarrow \begin{cases} x_2 = 2x_1 - 1 \\ x_2 = \frac{x_1 + 3}{2} \end{cases} \Rightarrow \begin{cases} x_1^* = \frac{5}{3} = 1.6667 \\ x_2^* = \frac{7}{3} = 2.3333 \end{cases} \quad (۴-۲۲)$$

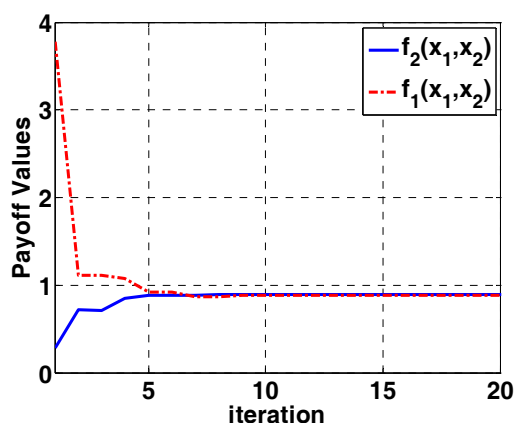
بنابراین نقطه تعادل نش این گیم ساده $(x_1^*, x_2^*) = (1.6667, 2.3333)$ با مقادیر ارزش تبادل

(0.88889, 0.88889) می‌باشد. این مقادیر برای ارزیابی نتایج حاصل از اعمال الگوریتم ژنتیک و الگوریتم رقابت استعماری مورد استفاده قرار خواهند گرفت. برای الگوریتم ژنتیک در این مساله از جمعیت اولیه برابر با ۲۰ و نرخ جهش ۰/۲ و نرخ تولید ۰/۵ استفاده شده است و الگوریتم معرفی شده نیز با ۲۰ کشور اولیه که ۵ تا از آنها به عنوان امپریالیست انتخاب شده اند، بهینه‌سازی مساله را انجام داده است.

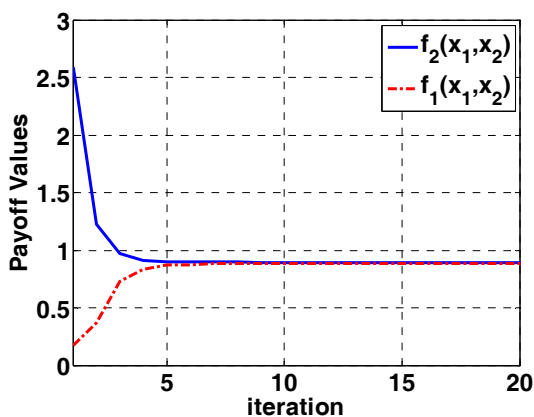
شکل ۴-۱۵ شمای کلی اعمال الگوریتم معرفی شده به مسئله تعیین نقطه تعادل نش را نشان می‌دهد. بخش آبی رنگ داخل خط چین، مربوط به الگوریتم می‌باشد. شکل‌های ۴-۱۶ و ۴-۱۷ نیز همگرایی هزینه برای الگوریتم ژنتیک و الگوریتم استعماری را نشان می‌دهند. همانگونه که از دیاگرام‌های همگرایی نشان داده شده بر می‌آید، الگوریتم رقابت استعماری به نزدیک و حتی نسبتاً بهتری نسبت به الگوریتم ژنتیک رسیده است. هر دو الگوریتم توانسته اند به همان جواب تحلیلی مساله برسند. البته مساله در نظر گرفته شده فقط برای نشان دادن شیوه اعمال الگوریتم معرفی شده در زمینه نظریه بازی‌ها بود و مساله بسیار ساده‌ای بوده و برای به چالش کشیدن دو الگوریتم موفق بهینه‌سازی مناسب نمی‌باشد.



شکل ۴-۱۵: شمای کلی الگوریتم رقابت استعماری، اعمال شده به مسئله تعیین نقطه تعادل نش. ناحیه داخل خط چین مربوط به الگوریتم و بقیه مربوط به مسئله می‌باشد.



شکل ۴-۱۶: همگرایی هزینه برای الگوریتم ژنتیک



شکل ۴-۱۷: همگرایی هزینه برای الگوریتم رقابتی استعماری

۴-۳-۲ یک بازی با پیچیدگی بیشتر

دو تابع معیار زیر را در نظر می‌گیریم.

$$f_1(x_1) = 4x_1 \quad (۴-۲۳)$$

$$f_2(x_1, x_2) = g(x_2).h(f_1(x_1), g(x_2)) \quad (۴-۲۴)$$

که در آن

$$g(x_2) = \begin{cases} 4 - 3 \exp\left(-\left(\frac{x_2 - 0.2}{0.02}\right)^2\right) & \text{if } 0 \leq x_2 \leq 0.4 \\ 4 - 2 \exp\left(-\left(\frac{x_2 - 0.7}{0.2}\right)^2\right) & \text{if } 0.4 \leq x_2 \leq 1 \end{cases} \quad (۴-۲۵)$$

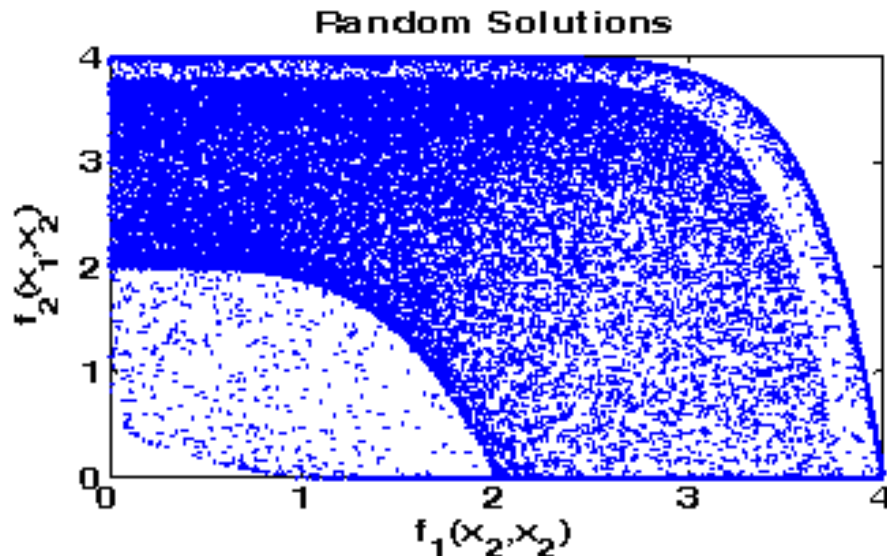
$$h(f_1, g) = \begin{cases} 1 - \left(\frac{f_1}{g}\right)^\alpha & \text{if } f_1 \leq g \\ 0 & \text{otherwise} \end{cases} \quad (۴-۲۶)$$

$$\alpha = 0.25 + 3.75(g(x_2) - 1) \quad (۴-۲۷)$$

x_1 و x_2 استراتژی بازی هستند. در این بخش می‌خواهیم با استفاده از الگوریتم معرفی شده، نقطه

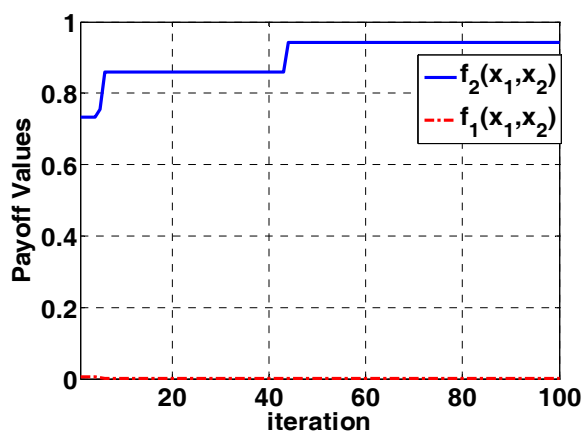
تعادل نش این توابع را بیابیم. بررسی دقیقتر این مسئله بهینه‌سازی ۲ هدفه نشان می‌دهد که این مسئله

دارای یک مجموعه پرتو عام محدب و یک مجموعه پرتو محلی مقعر است. شکل ۱۸-۴ نحوه پخش (f_1, f_2) را برای ۵۰۰۰۰ جفت نقاط تصادفی (x_1, x_2) در بازه $[0, 1]$ نشان می‌دهد.

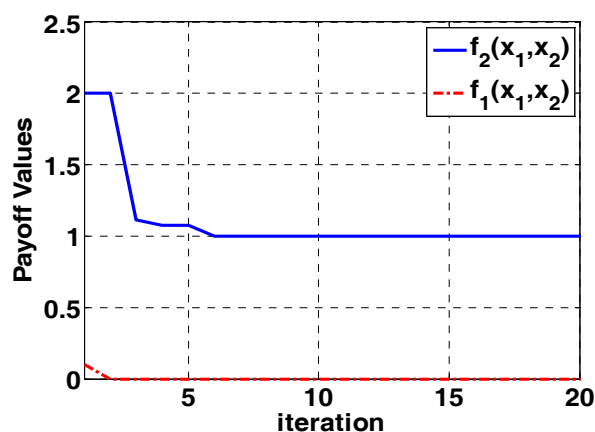


شکل ۱۸-۴: نحوه پخش (f_1, f_2) برای ۵۰۰۰۰ نقطه تصادفی

حال هر دو روش الگوریتم ژنتیک و الگوریتم معرفی شده (ICA) را به مساله فوق اعمال می‌کنیم. با همان تنظیمات قبلی برای الگوریتم‌های فوق، هیچ کدام نتوانستند به نقطه تعادل نش برسند. بنابراین تعداد جمعیت اولیه الگوریتم ژنتیک و تعداد کشورهای اولیه الگوریتم استعماری افزایش داده شدند. اما همانگونه که در شکل‌های زیر نشان داده شده است، با افزایش جمعیت تا ۲۰۰ به همراه افزایش تعداد تکرارها، الگوریتم ژنتیک نتوانست به نقطه تعادل نش سیستم دست پیدا کند. این در حالی است که الگوریتم رقابت استعماری در تعداد کشور اولیه ۳۵ توانست به نقطه تعادل $(x_1^*, x_2^*) = (0, 0.20002)$ با توابع معیار $(0,1)$ دست پیدا کند که این مقدار بر روی منحنی پرتو قرار دارد. شکل‌های زیر همگرایی الگوریتم ژنتیک و الگوریتم رقابت استعماری را نشان می‌دهد.



شکل ۴-۱۹: همگرایی الگوریتم ژنتیک

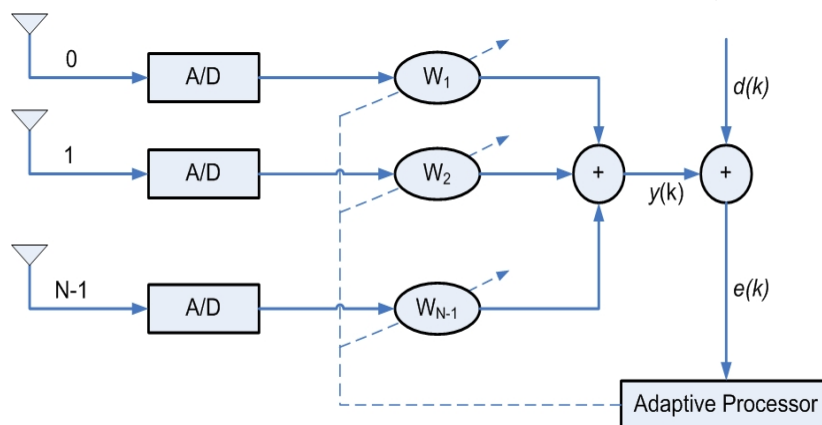


شکل ۴-۲۰: همگرایی الگوریتم معرفی شده

۴-۴ طراحی بهینه آنتهای آرایه‌ای

شکل زیر سیستم آنتن آرایه‌ای را نشان می‌دهد. در این آنتنها هدف، تنظیم وزنهای w با هدف

داشتن بیشترین $SINR^1$ می‌باشد.



شکل ۴-۲۱: شمای کلی یک آنتن آرایه‌ای

در این پیاده‌سازی از الگوریتم معرفی شده و نیز الگوریتم ژنتیک، جهت تعیین بهینه وزنهای آنتن

¹ Signal-to-Interference-Plus-Noise-Ratio

استفاده شده است. پارامترهای الگوریتم ژنتیک و الگوریتم رقابت استعماری مطابق جداول ۴-۴ و ۴-۵ زیر می‌باشند.

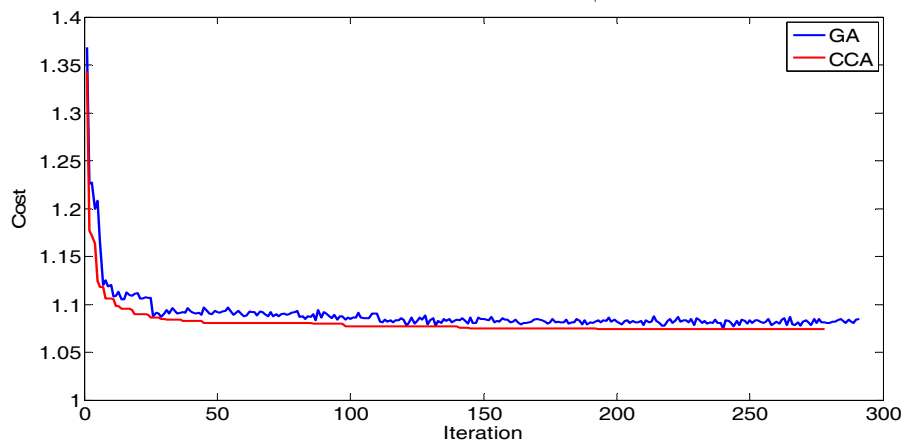
جدول ۴-۴: پارامترهای مورد استفاده در الگوریتم رقابت استعماری

| | |
|-------------------------|-----------|
| تعداد کشورهای اولیه | 60 |
| تعداد استعمارگران اولیه | 6 |
| تعداد مستعمرات | $60-6=54$ |
| β | 2 |
| γ | $\pi/4$ |
| ζ | 0.01 |

جدول ۴-۵: پارامترهای مورد استفاده در الگوریتم ژنتیک

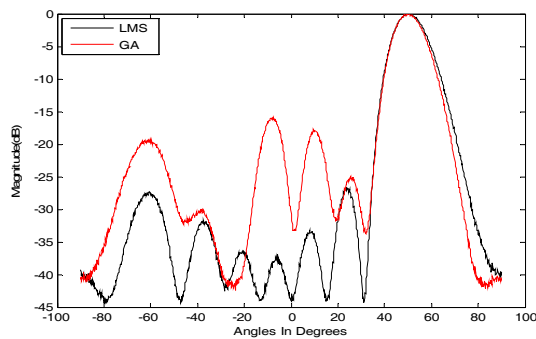
| | |
|-------------------------|----------------|
| تعداد جمعیت اولیه | 60 |
| درصد تولید نسل | 60 |
| درصد جهش | 10 |
| درصد انتقال به نسل بعدی | 30 |
| روش انتخاب والدین | Cost weighting |

نمودار همگرایی این دو الگوریتم نیز در زیر نشان داده شده اند.

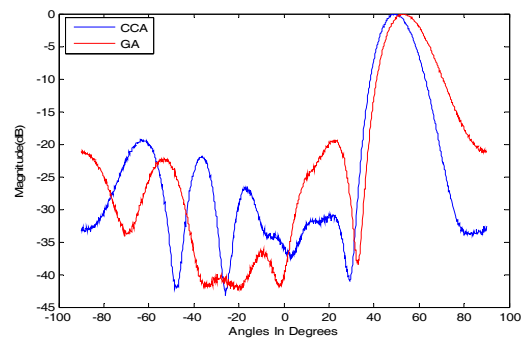


شکل ۴-۲۲: نمودار همگرایی الگوریتم رقابت استعماری و الگوریتم ژنتیک

شکلهای زیر نیز نتایج مقایسه الگوهای آنتن شکل یافته را توسط الگوریتم رقابت استعماری، الگوریتم ژنتیک و یک الگوریتم کلاسیک موسوم به LMS را نشان می‌دهند.



شکل ۴-۲۳: مقایسه LMS با الگوریتم ژنتیک



شکل ۴-۲۴: مقایسه الگوی الگوریتم ژنتیک و الگوریتم معرفی شده

بررسی مسئله از دیدگاه مسائل انتهای آرایه‌ای نشان می‌دهد که الگوریتم رقابت استعماری به خوبی توانسته است به آرایه آنتنی را به گونه‌ای شکل دهد که دارای بیشترین $SINR$ در زاویه مطلوب و کمترین $SINR$ در زاویه نویز می‌باشد. کارایی آنتن طراحی شده توسط این الگوریتم، نسبت به مورد مشابه الگوریتم ژنتیک آن بهتر می‌باشد.

۴-۵ استفاده از الگوریتم رقابت استعماری برای شناسایی ویژگی مواد از آزمون فرورفتگی

۴-۵-۱ مقدمه

پاسخ آزمون فرورفتگی^۱، که عمق نفوذ فرورونده، h را بر حسب بار اعمالی به فرورونده، P ، حساب می‌کند، مورد علاقه‌ی بسیاری از محققان زمینه مواد و تحلیل تنش در سالهای اخیر بوده است. استفاده از پاسخ آزمون فرورفتگی، برای پیش‌بینی خواص الاستوپلاستیک، کاربرد دیگری است، که توجه بسیاری از محققان را در حوزه علم مواد به خود جلب کرده است. روش‌های استخراج خواص

¹ Indentation test response

الاستوپلاستیک، در پژوهش‌های فراوانی، مورد مطالعه قرار گرفته‌اند. مطالعه آزمون فرورفتگی تیز، برای تخمین خواص مکانیکی مواد، از دو گام عمده تشکیل یافته است:

- در گام اول، پاسخ آزمون فرورفتگی مواد، شامل ویژگی‌یابی منحنی جابجایی بار، با استفاده از خواص مکانیکی مواد و خواص مکانیکی و ویژگی هندسی فرورونده، پیش‌بینی می‌شود (الگوریتم مستقیم).

- در گام بعد، خواص الاستوپلاستیکی مواد با تفسیر پاسخ آزمون فرورفتگی مواد، پیش‌بینی می‌شود (الگوریتم معکوس).

در این بخش، یک شبکه عصبی مصنوعی چند لایه مستقیم، برای پیش‌بینی پاسخ آزمون فرورفتگی مواد، مورد استفاده قرار می‌گیرد. مقادیر محاسبه شده برای توابع بدون بعد، برای یادگیری، این شبکه عصبی، مورد استفاده قرار می‌گیرند. در نهایت نیز الگوریتم رقابت استعماری، به مسئله ارزیابی خواص مواد از پاسخ آزمون فرورفتگی، اعمال می‌شود.

۴-۵-۲ توصیف مسئله معکوس

برای یک ماده الاستوپلاستیک معین، پاسخ آزمون فرورفتگی، شامل C ، dP_u/dh و h_r/h_m می‌تواند از طریق آزمون فرورفتگی ابزاری، تعیین شود. هدف اصلی این بخش، پیش‌بینی خواص الاستوپلاستیک مواد از پاسخ آزمون فرورفتگی (الگوریتم معکوس) است. پس از آموزش یک شبکه عصبی برای روش مستقیم، از الگوریتم معرفی شده برای آنالیز معکوس آن استفاده می‌کنیم. در حقیقت با این کار می‌خواهیم از خروجی شبکه به ورودی آن برسیم. برای بیان دوباره مسئله‌ای که باید حل شود، تابعی به صورت زیر در نظر می‌گیریم.

$$F(X) = ANN(X) - Y \quad (4-28)$$

که در آن X ، یک آرایه 1×3 از ویژگی‌های الاستوپلاستیکی E, σ_y, n و Y نیز ماتریسی 1×3 است که مربوط به تست فرورفتگی، $\left[C, d_p/d_h \Big|_{h=h_m}, h_r/h_m \right]$ یک ماده معین است. ANN نیز، شبکه

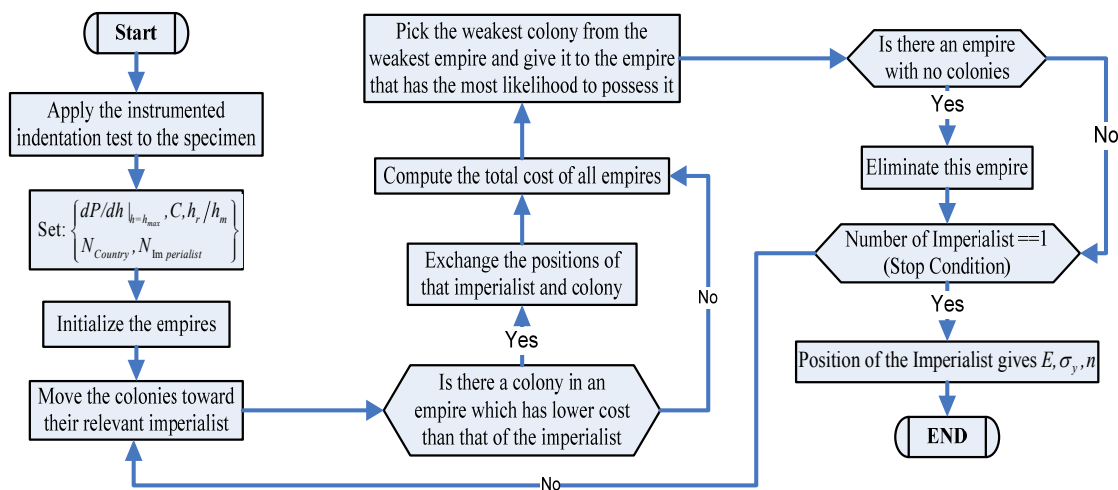
عصبی مصنوعی ای است که برای پیش‌بینی پاسخ آزمون فرورفتگی آموزش دیده است. ایده‌ی اصلی الگوریتم معکوس، یافتن $X(E, \sigma_y, n)$ مناسب به گونه‌ای است که معادله (۲۸-۴) را وقتی برابر صفر قرار می‌دهیم، راضی می‌کند. تابع مذکور، یک بهینه‌سازی چند هدفه با پیچیدگی بالا را ایجاد می‌کند. برای استفاده از الگوریتم تک هدفه، تابع هزینه را به صورت زیر تعریف می‌کنیم:

$$\begin{cases} ANN(X) = (ann_1, ann_2, ann_3) \\ Y = (C, d_p/d_h|_{h=h_m}, h_r/h_m) \end{cases} \Rightarrow \begin{cases} f_1 = (ann_1 - C) \\ f_2 = (ann_2 - dp_u/dh|_{h=h_m}) \\ f_3 = (ann_1 - h_r/h_m) \end{cases} \Rightarrow Cost = \sqrt{\frac{1}{3} \sum_{i=1}^3 (f_i)^2} \quad (۲۹-۴)$$

الگوریتم رقابت استعماری (ICA)، برای بهینه‌سازی تابع هدف فوق، مورد استفاده قرار گرفته است.

۴-۵-۳ حل مسئله معکوس توسط الگوریتم رقابت استعماری

در این بخش، الگوریتم رقابت استعماری به مسئله معکوس، اعمال می‌شود. شکل زیر فلوچارت الگوریتم اعمال شده به آنالیز معکوس آزمون فرورفتگی را نشان می‌دهد. الگوریتم، خواص الاستوپلاستیک را با کمینه کردن تابع هزینه داده شده در معادله (۲۹-۴)، پیش‌بینی می‌کند.



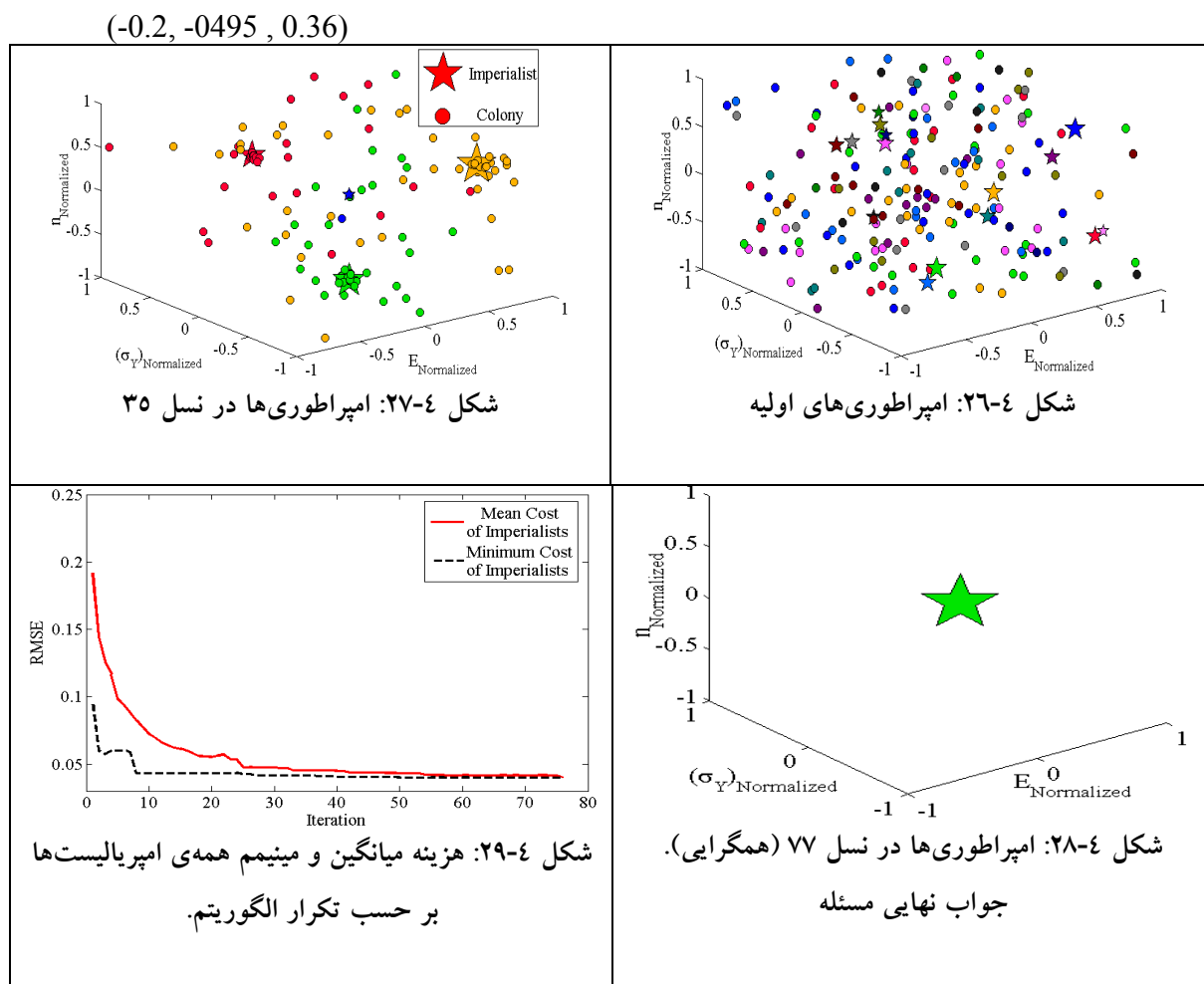
شکل ۴-۲۵: فلوچارت ICA، اعمال شده به آنالیز معکوس آزمون فرورفتگی.

تنظیمات اولیه‌ی ICA عبارتند از $N_{country}$ و $N_{imperialist}$. اجرای الگوریتم برای یک دسته از

پاسخ‌های آزمون فرورفتگی و با تنظیمات اولیه متفاوت، نشان داد که وقتی که $N_{country} \geq 300$ و $N_{imperialist} \geq 20$ ، الگوریتم، بهترین همگرایی را داشته و به جواب یکتایی برای همه دسته ورودی‌ها بدست آمده از FE می‌رسد. شکل ۴-۲۶، امپراطوری‌های ICA اعمال شده به یک نمونه از آزمون فرورفتگی حاصل از FE را نشان می‌دهد. خواص الاستوپلاستیکی این نمونه، برابر است با

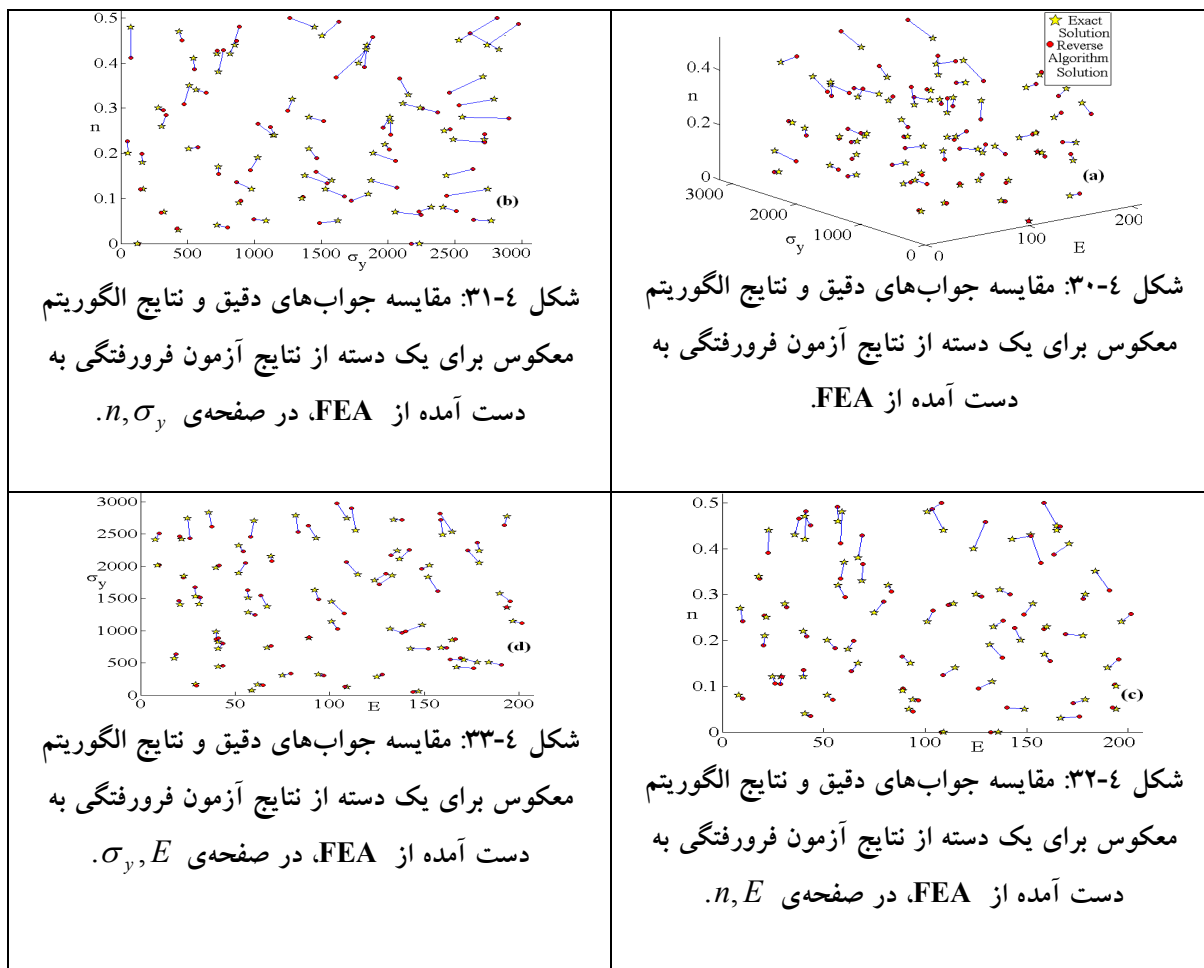
$$(E = 90 \text{ Gpa}, n = 0.34, \sigma_y = 780 \text{ Mpa})$$

که در حالت نرمالیزه، به صورت زیر در می‌آید.



تعداد اولیه‌ی کشورها برابر با ۲۰۰ بوده که ۲۰ امپراطوری اولیه را تشکیل می‌دهند. شکل‌های ۴-۲۶، ۴-۲۷ و ۴-۲۸ به ترتیب، امپراطوری اولیه و امپراطوری‌ها در نسل‌های ۳۵ و ۷۷ (همگرایی) را نشان می‌دهند. هزینه مینیمم و میانگین همه امپریالیست‌ها نیز، در شکل ۴-۲۹ نشان داده شده‌اند.

اگرچه جواب الگوریتم به شرایط اولیه بستگی دارد، اما انتخاب $N_{country}$ و $N_{imperialist}$ به حد کافی زیاد، که فضای متغیرها را پوشش می‌دهد، می‌تواند، یکتایی جواب‌ها را تضمین کند. شکل ۴-۳۰، خواص الاستوپلاستیک (جواب دقیق) و نتایج الگوریتم معکوس را برای یک دسته از پاسخ‌های آزمون فرو رفتگی، نشان می‌دهد.



در شکل ۴-۳۰، در هر دیاگرام، هر محور، نشان دهنده یک خاصیت الاستوپلاستیک است. برای داشتن یک دید مجزا، نتایج در صفحات تصویر شده، در شکل‌های ۴-۳۱، ۴-۳۲ و ۴-۳۳، نشان داده شده‌اند. تفاوت بسیار کوچک میان خروجی‌های الگوریتم معرفی شده و نتایج دقیق، نه تنها قابلیت الگوریتم معکوس ارائه شده را نشان می‌دهد، بلکه این نتایج تأکید می‌کنند که ICA استراتژی قدرتمندی، برای بهینه‌سازی توابع با پیچیدگی از درجه بالا می‌باشد.

۴-۵-۴ نتیجه گیری

نتایج این بخش نشان می‌دهند که آزمون فرورفتگی، می‌تواند به عنوان یک تکنیک ساده و سریع برای پیش‌بینی خواص الاستوپلاستیکی مواد، مورد استفاده قرار گیرد. در این بخش شبکه عصبی مصنوعی و روش المان محدود برای مدلسازی آزمون فرورفتگی (الگوریتم مستقیم) اعمال شده است. نتایج الگوریتم معکوس، که به حوزه وسیعی از خواص الاستوپلاستیکی اعمال شده است، تاکید می‌کنند که الگوریتم ارائه شده، جواب یکتا و دقیقی برای همه نمونه‌ها می‌دهد. همچنین، در نظر گرفتن پیچیدگی بالای تابع هزینه‌ای که باید کمینه شود و نیز با توجه به یکتایی و دقت جواب‌های بدست آمده، تاکید دوباره بر توانایی الگوریتم رقابت استعماری، در برخورد با مسائل بهینه‌سازی پیچیده است.

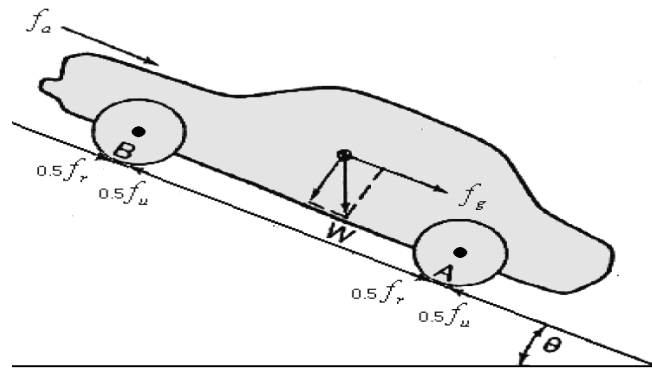
۴-۶ کنترل فازی اتومبیل

در این بخش، الگوریتم رقابت استعماری به عنوان یک استراتژی جدید بهینه‌سازی، به مسئله طراحی کنترل‌کننده فازی اعمال شده است. هدف طراحی کنترل‌کننده ای است که رفتار گذرا و ماندگار سیستم را تا حد زیادی بهبود دهد. با داشتن قوانین فازی، توابع عضویت متغیرهای ورودی و خروجی سیستم فازی به صورت بهینه طراحی شده اند و روش طراحی به یک مدل از خودرو اعمال شده است. ورودی‌های سیستم فازی سرعت وسیله و شیب جاده هستند. کنترل‌کننده فازی سرعت وسیله را با تنظیم مقدار سوخت ورودی به موتور تنظیم می‌کند. نتایج مقایسه میان کنترل‌کننده طراحی شده توسط الگوریتم رقابت استعماری و کنترل‌کننده طراحی شده توسط کارشناس نشان می‌دهد که کنترلر بدست آمده توسط الگوریتم رقابت استعماری کارایی بهتری نسبت به کنترلر کارشناس دارد.

۴-۶-۱ مدل اتومبیل

شکل زیر یک اتومبیل را در یک جاده شیب‌دار نشان می‌دهد. روابط حاکم بر اتومبیل به صورت زیر می‌باشند [۴۷].

$$m\left(\frac{d^2x}{dt^2}\right) = f_u - f_g - f_r - f_a \quad (۴-۳۰)$$



شکل ۴-۳۴: اتومبیل در جاده شیب‌دار

که در آن m جرم وسیله نقلیه، f_u نیروی وارده به اتومبیل از طرف موتور و f_g نیروی ناشی از وزن اتومبیل، f_r مقاومت اصطکاک تایرها و f_a مقاومت آیرودینامیکی است. خواهیم داشت:

$$f_r = k_r \left(\frac{dx}{dt} \right), f_a = k_a \left(\frac{dx}{dt} \right)^2, f_g = mg \sin \theta, f_u = k_u T \quad (۴-۳۱)$$

که در θ آن شیب جاده و T نیز میزان سوخت ورودی به موتور است. در فضای حالت خواهیم داشت:

$$\begin{aligned} \dot{X}_1 &= X_2 \\ \dot{X}_2 &= \frac{k_u}{m} T - g \sin \theta - \frac{k_r}{m} X_2 - \frac{k_a}{m} X_2^2 \end{aligned} \quad (۴-۳۲)$$

حالت‌های X_1 و X_2 به ترتیب موقعیت و سرعت اتومبیل هستند. در این بخش هدف طراحی کنترل کننده‌ای است که بتواند برای زوایای مختلف جاده و سرعت‌ها اولیه متفاوت، سرعت اتومبیل را به ۵۰ مایل بر ساعت برساند. کنترل کننده فازی با در نظر گرفتن زوایا و سرعت‌های اولیه متفاوت، این کار را با تنظیم مقدار سوخت ورودی به موتور تنظیم می‌کند. بدین منظور سرعت اتومبیل با توابع عضویت Low, Medium, High در نظر گرفته می‌شود. شیب نیز دارای توابع عضویت Up, Level, Down می‌باشد. سوخت ورودی نیز دارای ۵ تابع عضویت Very Low, Low, Medium, High, Very High می‌باشد. بنابراین در کل ۹ قانون خواهیم داشت.

$$\text{IF } V = MF^V_i \text{ \& } \theta = MF^\theta_j \text{ THEN } T = MF^T_k, 1 \leq i \leq 3, 1 \leq j \leq 3, 1 \leq k \leq 5 \quad (۴-۳۳)$$

با تعیین قوانین فازی توسط کارشناس [۴۷]، توابع عضویت متغیرها برای داشتن کمترین مجموع وزن‌دار زمان نشست و زمان صعود و فراجش و انتگرال قدر مطلق خطا توسط الگوریتم معرفی شده

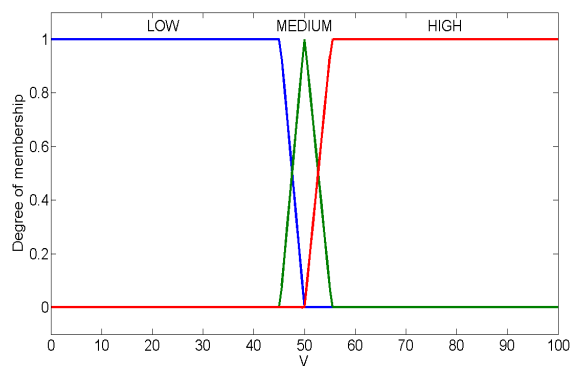
تعیین می‌گردند. قوانین فازی مورد استفاده در جدول زیر نشان داده شده‌اند.

جدول ۶-۴: قوانین فازی مورد استفاده در کنترل سرعت اتومبیل

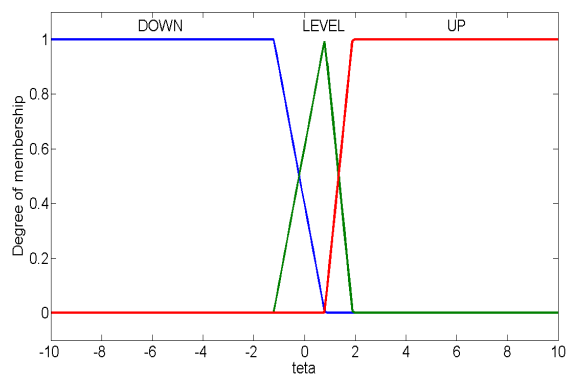
| $\theta \backslash V$ | DOWN | LEVEL | UP |
|-----------------------|------|--------|------|
| LOW | HM | HM | HIGH |
| OK | LM | Medium | HM |
| HIGH | LOW | LM | LM |

۴-۶-۲ نتایج

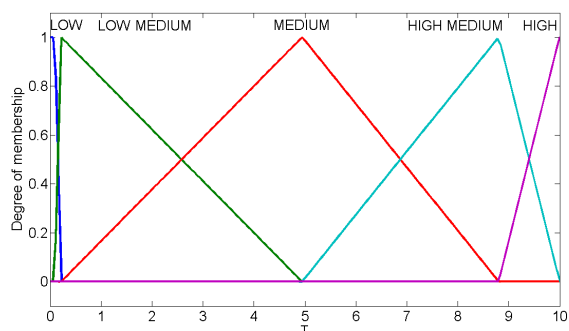
با مثلی گرفتن توابع عضویت، توابع عضویت شکل یافته در شکلهای زیر نشان داده شده‌اند.



شکل ۴-۳۵: توابع عضویت مربوط به متغیر سرعت

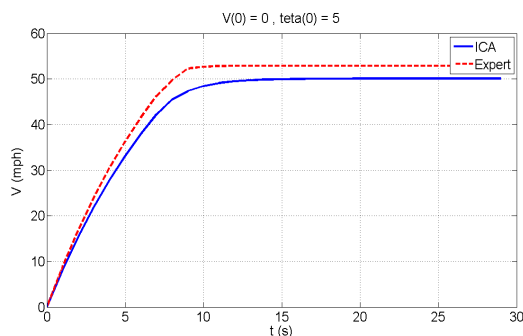


شکل ۴-۳۶: توابع عضویت متغیر θ

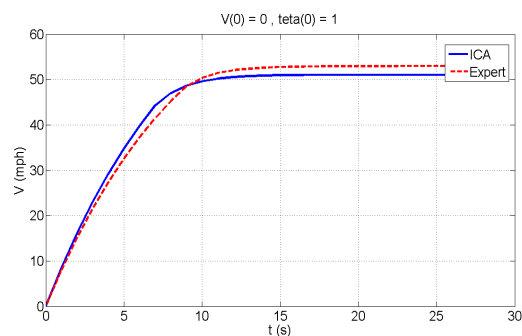


شکل ۴-۳۷: توابع عضویت متغیر T

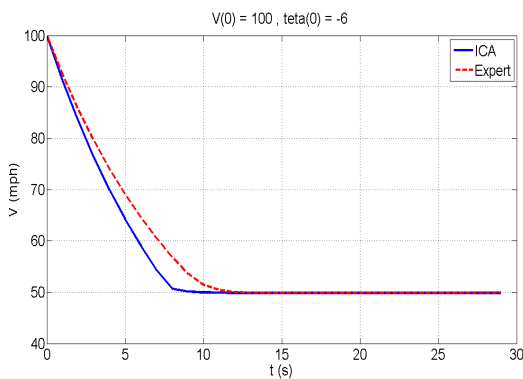
در شکل‌های زیر نیز خروجی حاصل از کنترل کننده کارشناس و کنترل کننده طراحی شده توسط الگوریتم معرفی شده با هم مقایسه شده‌اند. نتایج حاکی از موفقیت نسبی کنترل کننده اخیر نسبت به کارشناس دارد.



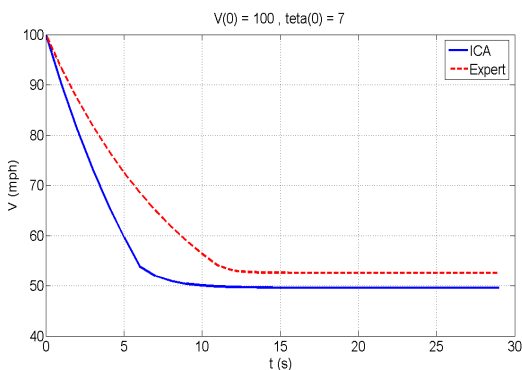
شکل ۴-۳۹: سرعت خروجی برای $V(0)=0$ و $\theta=5$



شکل ۴-۳۸: سرعت خروجی برای $V(0)=0$ و $\theta=1$



شکل ۴-۴۱: سرعت خروجی برای $V(0)=100$ و $\theta=-6$



شکل ۴-۴۰: سرعت خروجی برای $V(0)=100$ و $\theta=7$

فصل پنجم

خلاصه، نتیجه گیری و
پیشنهادهات

آنچه در این نوشتار مورد بررسی قرار گرفت ارائه یک الگوریتم بهینه‌سازی جدید بر مبنای مدلسازی ریاضی فرایند اجتماعی-سیاسی پدیده استعمار بود. روش‌های مختلفی برای حل مسائل بهینه‌سازی معرفی شده‌اند. بعضی از این روش‌ها به صورت تکراری و بر مبنای گرادیان، نقطه بهینه تابع هزینه را پیدا می‌کنند. این روش‌ها معمولاً سرعت بالایی دارند ولی در درعوض مشکل افتادن در دام بهینه محلی را با خود حمل می‌کنند. در نقطه مقابل روش‌هایی وجود دارند که به جستجوی نقطه بهینه مطلق تابع می‌پردازند. الگوریتم‌های ژنتیک و بهینه‌سازی گروه ذرات نمونه‌هایی از این روش‌ها هستند. نکته قابل توجه در مورد اکثر روش‌های بهینه‌سازی تکاملی مطرح شده، این است که این روش‌ها معمولاً برگرفته از تکامل زیستی و مدلسازی پدیده‌های طبیعی هستند و معمولاً جنبه‌هایی از تکامل که مدل شناخته‌شده‌ای از آن وجود ندارد، در حاشیه تحقیقاتی قرار گرفته است. در حقیقت انگیزش اصلی نگارش این پایان‌نامه پر کردن این خلا و بررسی جوانب پاسخ منفی‌ای بود که به سوال زیر داده می‌شد:

"آیا تکامل موجودات و به ویژه انسان، تنها به تکامل زیستی او محدود می‌شود؟!؟"

و آنچه در ادامه مسیر مطرح شد، یافتن پاسخ به این سوال بود که "آیا جوانب دیگر تکامل انسانی می‌توانند به عنوان منبع الهام یک الگوریتم بهینه‌سازی مورد استفاده قرار بگیرند؟"

الگوریتم معرفی شده در این نوشتار، "الگوریتم رقابت استعماری"، یکی از پاسخ‌های مثبتی بود که می‌شد به این سوال داد. بطور ویژه در معرفی این الگوریتم، یک فرایند خاص مورد بررسی ویژه‌ای قرار گرفت. فرایند اجتماعی-سیاسی-تاریخی استعمار، پدیده‌ای بود که در این نوشتار برای ارائه الگوریتم مورد استفاده قرار گرفت.

بررسی تاریخی رفتار متقابل مستعمرات و استعمارگران نشان داد که فرایند همگون‌سازی، از سوی استعمارگران برای جذب مستعمرات در فرهنگ و رسوم آنها اعمال می‌شد. همانگونه که موارد تاریخی نشان می‌دهند، اعمال سیاست جذب در بعضی موارد موجب ایجاد تغییرات سریع اجتماعی، سیاسی و اقتصادی در مستعمرات شد. سیاست جذب در کنار رقابت استعماری، هسته‌های الگوریتم معرفی شده را تشکیل می‌دهند.

بطور خلاصه الگوریتم معرفی شده، با تعدادی کشور اولیه شروع می‌شود. این کشورها به دسته‌هایی به نام امپراطوری تقسیم می‌شوند. هر امپراطوری از تعدادی مستعمره و یک امپریالیست تشکیل شده است. در داخل امپراطوری، سیاست جذب از سوی استعمارگران به مستعمرات اعمال شده و آنها را در راستای محورهای مختلف اجتماعی-سیاسی به سوی خود می‌کشند. به همراه سیاست جذب، رقابتی نیز میان امپراطوری‌ها برقرار است و همه آنها برای در دست گرفتن مستعمرات همدیگر تلاش می‌کنند. حاصل این چرخه جذب و رقابت، همگرایی کشورها (جواب‌های ممکن مسئله) به سمت نقطه بهینه مطلق است.

نتایج آزمایش روش پیشنهادی بر روی توابع هزینه مختلف نشان می‌دهد که الگوریتم معرفی شده در یافتن نقطه بهینه این توابع کاملاً موفق عمل می‌کند. همچنین مسائل مختلف کاربردی حل شده با این الگوریتم نشان می‌دهند که استراتژی بهینه‌سازی مطرح شده می‌تواند با موفقیت کامل در کنار سایر روش‌های مطرح بهینه‌سازی همچون الگوریتم ژنتیک و گروه ذرات، به حل مسائل کاربردی و مهندسی کمک کند. مقایسه نتایج حاصله توسط الگوریتم مطرح شده با روش‌های رایج بهینه‌سازی نیز از برتری نسبی این الگوریتم حکایت دارد.

الگوریتم معرفی شده به عنوان نسخه اولیه یک الگوریتم مبتنی بر یک فرایند اجتماعی-سیاسی و بطور اخص پدیده پیچیده استعمار می‌باشد. بنابراین مطمئناً می‌توان اصلاحاتی در آن نیز ایجاد نمود. الگوریتم معرفی شده در حال حاضر برای حل مسائل پیوسته بهینه‌سازی مناسب می‌باشد. برای حل مسائل گسسته بهینه‌سازی باید تغییراتی در الگوریتم اعمال شود. ارائه نسخه گسسته الگوریتم می‌تواند برای حل مسائلی همچون انتخاب ورودی در شناسایی سیستمها و انتخاب ویژگی برای اهداف بازشناسی الگو مفید باشد. الگوریتم‌های رایجی همچون بهینه‌سازی گروه ذرات نیز در نسخه اولیه خود برای حل مسائل پیوسته مطرح شده بودند و بعدها نسخه‌های گسسته آنها معرفی گردیده است. در کاربردهای اعمال شده نیز، همه مسائل بهینه‌سازی دارای تنها یک تابع هدف بودند. الگوریتم مطرح شده کنونی،

می‌تواند برای حل مسائل بهینه‌سازی چندبعدی^۱ و برای یافتن منحنی پرتو^۲ نیز استفاده شود ولی نتایج بدست آمده از آن به خوبی نتایج الگوریتم‌های بهینه‌سازی مخصوص مسائل چند هدفه (همانند *NSGA-II*، نسخه چند هدفه الگوریتم ژنتیک) نخواهد بود. بنابراین در ادامه کار می‌توان با اعمال تغییراتی در ساختار الگوریتم آن را برای حل مسائل بهینه‌سازی چند هدفه مناسب نمود.

همانگونه که بیان شد، روشهای تکاملی ویژگی گریز از نقطه مینیم محلی را دارند. در مقابل روشهای کلاسیک بهینه‌سازی دارای سرعت همگرایی بیشتری می‌باشند. برای داشتن هم سرعت همگرایی بالا و هم گیر نکردن در نقاط بهینه محلی، یک روش رایج ترکیب الگوریتم‌های تکاملی با روشهای کلاسیک بهینه‌سازی همچون روش نیوتون است. در ادامه کار می‌توان ترکیبی از الگوریتم مطرح شده را نیز با الگوریتم‌های کلاسیک بهینه‌سازی ترکیب نمود. با این کار امید آن است که نتایج به مراتب بهتری (از دید سرعت همگرایی) بدست آید.

بنابراین گام‌های عمده پیش‌روی ادامه کار عبارتند از:

- ارائه نسخه گسسته الگوریتم برای حل مسائلی همچون انتخاب ورودی در شناسایی سیستم‌ها
- ایجاد تغییرات در الگوریتم برای حل مسائل بهینه‌سازی با چند تابع هدف
- ترکیب الگوریتم معرفی شده با الگوریتم‌های کلاسیک بهینه‌سازی و آزمایش آن برای حل مسائل مختلف بهینه‌سازی

در راستای گام‌های اساسی فوق، می‌توان می‌توان پیشنهادات زیر را نیز برای ادامه کار مطرح نمود.

- مطالعه دقیقتر پیرامون فرایند اجتماعی-سیاسی تکامل انسانی و سعی در مدلسازی فرایندهای مدل نشده در الگوریتم.
- در الگوریتم معرفی شده ظهور یک امپراطوری مدل نشده است و الگوریتم با امپراطوری‌های اولیه شروع شده و با سقوط آنها ادامه می‌یابد. به عنوان یک تغییر در الگوریتم می‌توان تولد

¹ Multi-Objective Optimization

² Pareto Front

یک امپراطوری نیز وارد مدل کرد.

- اعمال الگوریتم معرفی شده به مسائل بیشتر در حوزه مهندسی برای یافتن نقاط ضعف و قوت آن

- [1] Randy. L. Haupt and Sue Ellen Haupt, *Practical Genetic Algorithms*, 2nd Edition, John Wiley & Sons Inc., 2004.
- [2] Pablo Pedregal, *Introduction to Optimization*, Springer-Verlag New York Inc., 2004.
- [3] Stephen Boyd and Lieven Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- [4] Edwin K. P. Chong and Stanislaw H. Żak, *An Introduction to Optimization*, John Wiley & Sons Inc., 2001.
- [5] Wenyu Sun and Ya-Xiang Yuan, *Optimization Theory and Methods: Nonlinear Programming*, Springer Science + Business Media, LLC, 2006.
- [6] Jorge Nocedal and Stephen J. Wright, *Numerical Optimization*, 2nd Edition, Springer Science + Business Media, LLC, 2006.
- [7] Mitchell Melanie, *An Introduction to Genetic Algorithms*, MIT Press, 1999.
- [8] Stephen Olariu and Albert Y. Zomaya, *Handbook of Bioinspired Algorithms and Applications*, Taylor & FrancisGroup, LLC, 2006.
- [9] Jens Gottlieb and Güntel R. Raidl, *Evolutionary Computation in Combinatorial Optimization*, Springer-Verlag Berlin Heidelberg, 2006.
- [10] Fred Glover and Gray A. Kochenberger, *Hanbook of Metaheuristics*, Kluwer Academic Publishers, 2003.
- [11] Patrick Siarry and others, *Metaheuristics for Hard Optimizations*, Springer-Verlag Berlin Heidelberg, 2006.
- [12] Anthony Brabazon and Michael O'Neill, *Biologically Inspired Algorithms for Financial Modelling*, Springer-Verlag Berlin Heidelberg, 2006.
- [13] Amit Konar, *Artificial Intelligence and Soft Computing*, CRC Press LLC, 2000.
- [14] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization" in *Proceedings of the 4th IEEE International Conference on Neural Networks*, pp. 1942-1948, 1995.

- [15] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, pp. 39-43, 1995.
- [16] Marco Dorigo, Luca Maria Gambardella and others, *Ant Colony Optimization and Swarm Intelligence*, Springer-Verlag Berlin Heidelberg, 2006.
- [17] Marco Dorigo and Gianni Di Caro, *The Ant Colony Optimization Metaheuristic*, Iridia University, 1999.
- [18] V. Chellaboina and M. K. Ranga, "Reduced Order Optimal Control Using Genetic Algorithms", *2005 American Control Conference*, June 8-10, 2005. Portland, OR, USA
- [19] B. Bontoux. and D. Feillet, "Ant colony optimization for the traveling purchaser problem", *Computers & Operations Research*, In Press, Corrected Proof, Available online 26 May 2006
- [20] Y. T. Hsiao, C. L. Chuang and C. C. Chien, "Ant Colony Optimization for Designing of PID Controllers", *2004 IEEE International Symposium on Computer Aided Control Systems Design*, September 24, 2004, Taipei, Taiwan
- [21] H. H. Balaci and J. F. Valenzuela, "Scheduling Electric Power Generators Using Particle Swarm Optimization Combined with Lagrangian Relaxation Method", *International Journal of Applied Mathematics and Computer Science*, 2004, Vol. 14, No. 3, 411–421
- [22] Pack, D., G. Toussaint, and R. Haupt. 1996. *Robot trajectory planning using a GA*. Int. Symp. on Optical Science, Engineering, and Instrumentation. SPIE's Annual Meeting, Denver, CO.
- [23] Obayashi, S., D. Sasaki, Y. Takeguchi, and N. Hirose. 2000. *Multiobjective evolutionary computation for supersonic wing-shape optimization*. IEEE Trans. Evol. Comput. 4:182–187.
- [24] Loughlin, D. H., S. R. Ranjithan, J.W. Baugh, Jr., and E. D. Brill Jr. 2000. *Application of GAs for the design of ozone control strategies*. J. Air Waste Manage. Assoc. 50:1050–1063.
- [25] Chambers, L. (ed.). 1995. *GAs, Applications*, Vol. 1. New York: CRC Press.

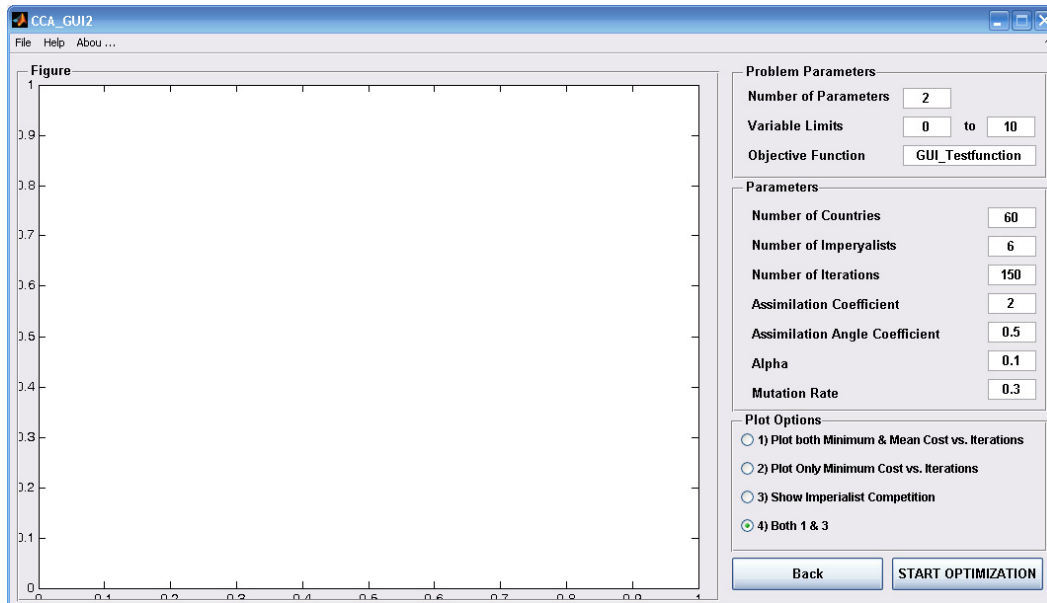
- [26] Angeline, P. J. 1996. *Evolving Fractal Movies*. Proc. 1st An. Conf. on Genetic Programming, MIT Press, Cambridge, MA, pp. 503–511.
- [27] Biles, J. A. 1994. GenJam: *A GA for generating jazz solos*. Proc. Int. Computer MusicvT Conf, San Francisco: 131–137.
- [28] Horner, A., and D. Goldberg. 1991. *GAs and computer-assisted music composition*. Proc. 4th Int. Conf. on GAs. Urbana-Champaign, IL.
- [29] Johnson, C.G., and J. J. Romero Cardalda. 2002. *GAs in visual art and music*. Leonardo 175–184.
- [30] Sims, K. 1991. *Artificial evolution for computer graphics*. Siggraph '91 Proc. 25:319–328.
- [31] *The Hutchinson Dictionary of World History*, Oxford: Helicon Publishing, 1999
- [32] R. R. Palmer, “*A History of the Modern World*”, New York: Alfred A. Knopf, 1964, ©1956.
- [33] C. V. Findley & J. A. M. Rothney, *Twentieth century world*, 3rd edition, Boston: Houghton Mifflin Company, 1986
- [34] R. Goff, J. Terry, W. Moss, J. H. Upshur, *The Twentieth Century: A Brief Global History*, 6th edition, Boston: McGraw-Hill, 2001, part1.
- [35] Y. T. Hsiao, C. L. Chuang and C. C. Chien, “*Ant Colony Optimization for Designing of PID Controllers*”, 2004 IEEE International Symposium on Computer Aided Control Systems Design, September 24, 2004, Taipei, Taiwan
- [36] Xiong Q., Cai W. J., He M.J., (2007), “*Equivalent transfer function method for PI/PID controller design of MIMO processes*”, Journal of Process Control 17, pp 665–673
- [37] Christen U., Musch H. E. and Steiner M., (1997) “*Robust control of distillation columns: μ - vs. H_∞ - synthesis*” J Proc C, mt Vol 7, No 1, pp. 19–30, 1997

- [38] Roffel B. et al (2000), "*First principles dynamic modeling and multivariable control of a cryogenic distillation process*" Computers and Chemical Engineering 24, pp111–123
- [39] Halevi Y., Palmor Z.J. and Efrati T., (1997), "*Automatic tuning of decentralized PID controllers for MIMO processes*" J. Proc. Cont. Vol. 7, No. 2, pp. I19-I28
- [40] Ruiz-López I.I., Rodríguez-Jimenes G.C., García-Alvarado M.A., (2006) "*Robust MIMO PID controllers tuning based on complex/real ratio of the characteristic matrix eigenvalues*", Chemical Engineering Science 61, pp 4332 – 4340
- [41] García-Alvarado M.A., Ruiz-López I.I., Torres-Ramos T., (2005), "*Tuning of multivariate PID controllers based on characteristic matrix eigenvalues, Lyapunov functions and robustness criteria*", Chemical Engineering Science 60, pp 897 – 905
- [42] Chang W. D. , (2007), "*A multi-crossover genetic approach to multivariable PID controllers tuning*", Expert Systems with Applications 33, pp 620–626
- [43] Hsin-Chieh Chen, Jen-Fuh Chang, Jun-Juh Yan, Teh-Lu Liao, (2007), "*EP-based PID control design for chaotic synchronization with application in secure communication*" Expert Systems with Applications, Article in press
- [44] Su C. T. , Wong J. T., (2007), "*Designing MIMO controller by neuro-traveling particle swarm optimizer approach*", Expert Systems with Applications 32, pp 848–855.
- [45] Wang, Q. G., Zou, B., Lee, T. H., & Qiang, B. (1997). *Auto-tuning of multivariable PID controllers from decentralized relay feedback*. Automatica, 33(3), pp 319–330.
- [46] Luyben, W. L. (1986). "*A simple method for tuning SISO controllers in a multivariable system*". Industrial & Engineering Chemistry Product Research and Development", 25, pp 654–660.
- [47] T.J.Ross, *Fuzzy Logic with Engineering Applications*, John Wiley & Sons, New York, NY, USA, 2nd edition, 2004

پیوست

جعبه ابزار بهینه سازی
تهیه شده بر مبنای الگوریتم
رقابت استعماری

برای استفاده ساده‌تر از الگوریتم معرفی شده، یک جعبه ابزار گرافیکی تهیه شده است که با استفاده از آن می‌توان الگوریتم معرفی شده را به سادگی به مسائل مختلف بهینه‌سازی بدون درگیری با پیچیدگی‌های کد نویسی، اعمال کرد. شکل زیر شمای کلی محیط تهیه شده را نشان می‌دهد. در این محیط کافی است که نام تابع هزینه مورد نظر که به صورت یک تابع متلب^۱ نوشته شده است، را وارد کنیم.

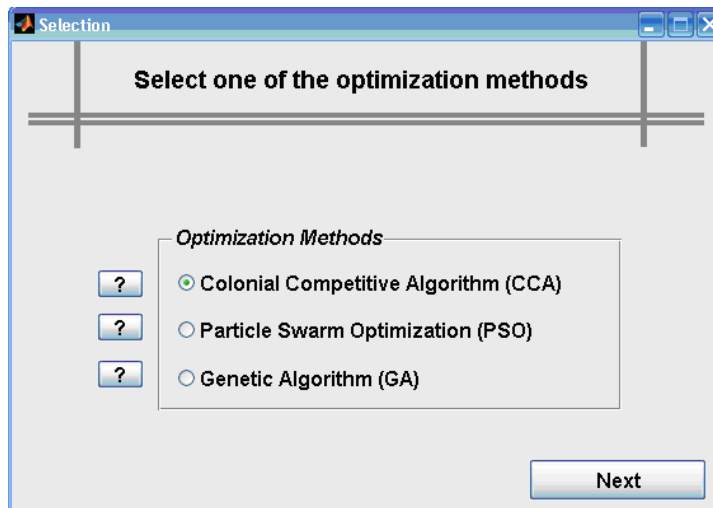


شکل ۱-۰: شمای کلی جعبه ابزار گرافیکی تهیه شده بر مبنای الگوریتم رقابت استعماری

محیط تهیه شده علاوه بر پیاده‌سازی الگوریتم معرفی شده، بهینه‌سازی‌های الگوریتم ژنتیک و بهینه‌سازی گروه ذرات را نیز شامل می‌شود. هدف از این امر امکان مقایسه نتایج الگوریتم معرفی شده با هر یک از این دو روش رایج بهینه‌سازی می‌باشد. در حقیقت با وارد شدن به محیط جعبه ابزار، قبل از صفحه فوق صفحه نمایش داده شده در شکل ۷-۲ نشان داده می‌شود که در آن از کاربر می‌خواهد که الگوریتم مورد نظر خود را از میان ۳ گزینه موجود، انتخاب کند. گزینه‌های موجود عبارتند از:

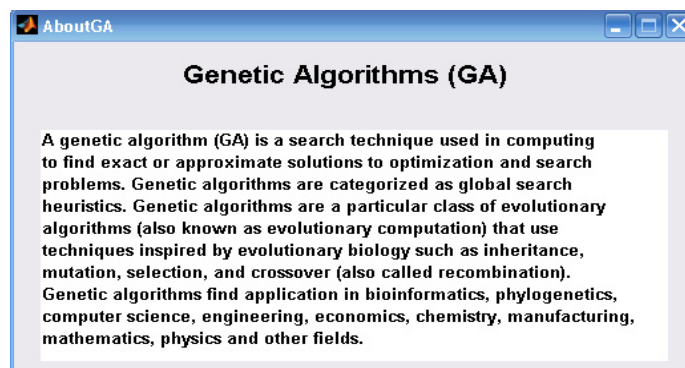
^۱ MATLAB Function

- الگوریتم رقابت استعماری
- الگوریتم بهینه‌سازی گروه ذرات
- الگوریتم ژنتیک



شکل ۲-۰: صفحه انتخاب الگوریتم مورد نظر برای بهینه‌سازی

با انتخاب هر یک از این گزینه‌ها و کلیک روی دکمه Next صفحه مربوط به آن الگوریتم بهینه‌سازی باز می‌شود. در کنار هر یک از گزینه‌ها یک علامت راهنما وجود دارد که با کلیک بر روی آن توضیحات مختصری در مورد الگوریتم مربوطه داده می‌شود. به عنوان مثال با کلیک بر روی علامت راهنمای الگوریتم ژنتیک صفحه زیر نشان داده می‌شود.



شکل ۳-۰: صفحه راهنمای مربوط به الگوریتم ژنتیک

با انتخاب هر یک از روشها، صفحه مربوط به آن باز می‌شود. به طور کلی هر کدام از صفحات،

دارای ۴ بخش (پنل) می‌باشند.

۱. بیان مسئله

۲. تعیین پارامترهای الگوریتم

۳. نمایش نتایج

۴. تنظیم نوع نمایش

بیان مسئله

بیان مسئله در هر سه الگوریتم یکسان می‌باشد. شکل زیر این بخش را نشان می‌دهد.

| Problem Parameters | | |
|----------------------|---|------------------------------------|
| Number of Parameters | <input type="text" value="2"/> | |
| Variable Limits | <input type="text" value="0"/> | to <input type="text" value="10"/> |
| Objective Function | <input type="text" value="GUI_Testfunction"/> | |

شکل ۴-۰: بخش مربوط به بیان مسئله

یک مسئله بهینه‌سازی با تعریف یک تابع هزینه که به صورت یک تابع متلب است، پیاده‌سازی می‌شود. در بخش بیان مسئله کافی است نام تابع هزینه به همراه تعداد پارامترهای مورد جستجو و نیز بازه بررسی متغیرها را وارد کنیم. در شکل فوق، تابع هزینه GUI_Testfunction با تعداد ۲ پارامتر و در بازه جستجوی [0 10] بهینه‌سازی می‌شود.

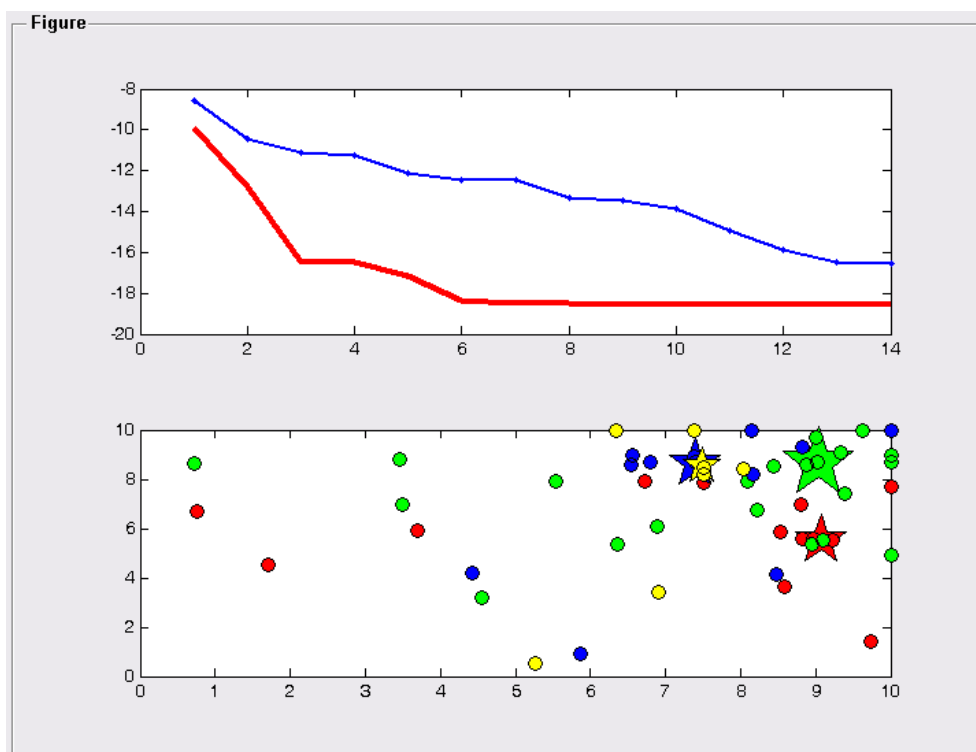
تنظیم پارامترهای الگوریتم

بخش تنظیم پارامترهای الگوریتم برای هر یک از سه الگوریتم پیاده‌سازی شده، متفاوت می‌باشد. شکل زیر این بخش را برای الگوریتم رقابتی معرفی شده نشان می‌دهد. در این بخش امکان تنظیم تعداد کشورهای اولیه و تعداد امپراطوری‌های اولیه و سایر پارامترها وجود دارد. در تنظیم پارامترهای الگوریتم ژنتیک نیز مواردی همچون نرخ جهش و بازترکیب و غیره تنظیم می‌شوند. در تنظیم ضرایب بهینه‌سازی

گروه ذرات نیز مواردی همچون ضرایب فردی و اجتماعی و تعداد تکرارها و غیره تنظیم می‌شوند.

| Parameters | |
|--------------------------------|-----|
| Number of Countries | 60 |
| Number of Imperialists | 6 |
| Number of Iterations | 150 |
| Assimilation Coefficient | 2 |
| Assimilation Angle Coefficient | 0.5 |
| Alpha | 0.1 |

شکل ۵-۰: بخش مربوط به تنظیم پارامترهای الگوریتم رقابت استعماری



شکل ۶-۰: الگوریتم رقابت استعماری در حال بهینه‌سازی تابع GUI_Testfunction

نمایش نتایج

با کلیک بر روی گزینه START OPTIMIZATION، نتیجه حاصل از بهینه‌سازی به صورت گرافیکی نمایش داده می‌شود. امکان نمایش هم تکامل تابع هزینه و هم موقعیت کشورها و ذرات و کروموزومها وجود دارد. نحوه نمایش بستگی به انتخاب یکی از ۴ گزینه موجود در بخش انتخاب شیوه نمایش دارد. شکل زیر الگوریتم رقابت استعماری را در حال بهینه‌سازی تابع GUI_Testfunction نشان می‌دهد.

انتخاب شیوه نمایش

انتخاب شیوه نمایش می‌تواند از میان هر یک از چهار گزینه موجود صورت پذیرد.



Plot Options

- ☐ 1) Plot both Minimum & Mean Cost vs. Iterations
- ☐ 2) Plot Only Minimum Cost vs. Iterations
- ☐ 3) Show Imperialist Competition
- ☒ 4) Both 1 & 3

شکل ۷-۰: انتخاب شیوه نمایش نتایج

در گوشه سمت چپ و بالا نیز گزینه‌هایی برای آشنایی و راهنمایی در مورد جعبه ابزار و پدید آورندگان و نیز چاپ نتایج وجود دارد.



در این بخش تعدادی از مقالات مربوط به الگوریتم معرفی شده به همراه چکیده آنها لیست شده است. مقالات لیست شده تنها شامل مقالاتی است که ارائه شده و یا توسط مجله و یا کنفرانس پذیرفته شده و یا به چاپ رسیده‌اند. از لیست کردن مقالات ارسالی، خودداری شده است.

مقالات ژورنال

Colonial Competitive Algorithm: A Novel Approach for PID Controller Design in MIMO Distillation Column Process

Esmail Atashpaz Gargari, Farzad Hashemzadeh, Ramin Rajabioun, Caro Lucas

Purpose - This paper describes Colonial Competitive Algorithm (CCA), a novel socio-politically inspired optimization strategy, and how it is used to solve real world engineering problems by applying it to the problem of designing a multivariable PID controller. Unlike other evolutionary optimization algorithms, CCA is inspired from a socio political process, the competition among imperialists and colonies. In this paper CCA is used to tune the parameters of a multivariable PID controller for a typical distillation column process.

Design/Methodology/Approach – The controller design objective was to tune the PID controller parameters so that the integral of absolute errors, overshoots and undershoots be minimized. This multi-objective optimization problem is converted to a mono-objective one by adding up all the objective functions in which the absolute integral of errors is emphasized to be reduced as long as the overshoots and undershoots remain acceptable.

Findings – Simulation results show that the controller tuning approach, proposed in this paper, can be easily and successfully applied to the problem of designing MIMO controller for control processes. As a result not only was the controlled process able to significantly reduce the coupling effect, but also the response speed was significantly increased. Also a Genetic Algorithm (GA) and an analytical method are used to design the controller parameters and are compared with CCA. The results showed that CCA had a higher convergence rate than GA, reaching to a better solution.

Originality/value – The proposed PID controller tuning approach is interesting for the design of controllers for industrial and chemical processes, e.g. MIMO evaporator plant. Also the proposed evolutionary algorithm, CCA, can be used in diverse areas of optimization problems including, industrial planning, resource allocation, scheduling, decision making, pattern recognition and machine learning.

International Journal of Intelligent Computing and Cybernetics, Vol. 1 No. 3, 2008, pp. 337-355

Application of Imperialist Competitive Algorithm for Material Properties Characterization from Sharp Indentation Test

A. Biabangard-Oskouyi, E. Atashpaz-Gargari, N. Soltani and C. Lucas

Abstract: In this paper a novel technique is proposed for characterizing the elasto-plastic properties of materials from sharp indentation test. Indentation test response is obtained for a wide range of engineering materials from finite element modeling. Finite element results are utilized for training a multi-layer artificial neural network which predicts indentation test response from elasto-plastic properties. Finally a new optimization algorithm inspired from historical imperialist competition called “Imperialist Competitive Algorithm” is developed and is employed for material properties evaluation from indentation test curve. Results obtained from applying the proposed method to a variety of sharp indentation test responses, indicate the good ability of proposed method for interpreting the indentation test responses for material properties determination.

Accepted to be published in International Journal of Engineering Simulation (IJES)

مقالات کنفرانس

Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition

Esmail Atashpaz Gargari & Caro Lucas

Abstract: This paper proposes an algorithm for optimization inspired by the imperialistic competition. Like other evolutionary ones, the proposed algorithm starts with an initial population. Population individuals called country are in two types: colonies and imperialists that all together form some empires. Imperialistic competition among these empires forms the basis of the proposed evolutionary algorithm. During this competition, weak empires collapse and powerful ones take possession of their colonies. Imperialistic competition hopefully converges to a state in which there exist only one empire and its colonies are in the same position and have the same cost as the imperialist. Applying the proposed algorithm to some of benchmark cost functions, shows its ability in dealing with different types of optimization problems..

2007 IEEE Congress on Evolutionary Computation (CEC 2007) . pp 4661 - 4667

Designing an optimal PID controller using Colonial Competitive Algorithm

Esmaeil Atashpaz Gargari & Caro Lucas

Abstract: in this paper we use a new search heuristic called “Colonial Competitive Algorithm” to design a optimal PID controller for a sample system. The PID controller is designed in such a way that it minimizes the sum of settling time, rise time, maximum overshoot and integral absolute error. A comparison among Colonial Competitive Algorithm and Genetic Algorithm is made through designing the controller.

First Iranian Joint Congress on Intelligent and Fuzzy Systems, September 2007, Mashhad, Iran.

Designing MIMO PID Controller using Colonial Competitive Algorithm: Applied to Distillation Column Proces

Esmaeil Atashpaz Gargari, Farzad Hashemzadeh, Caro Lucas

Abstract: In this paper, a colonial competitive algorithm is applied to the problem of designing a multivariable PID controller. The goal is to design a controller to decouple the controlled process, and to track the desired inputs by outputs of the process as much as possible. The method is used to design a multi variable controller for a typical distillation column process. Also a GA and an analytical method are used to design the controller parameters. Comparison results among these methods show that the controller obtained by colonial competitive algorithm has better performance than the others.

2008 IEEE Congress on Evolutionary Computation (CEC 2008) . pp 1929 - 1934

Colonial Competitive Algorithm as a Tool for Nash Equilibrium Point Achievement

Ramin Rajabioun, Esmaeil Atashpaz-Gargari, Caro Lucas

Abstract: This paper presents an application of Colonial Competitive Algorithm (CCA) in game theory and multi-objective optimization problems. The recently introduced CCA has proven its excellent capabilities, such as faster convergence and better global optimum achievement. In this paper CCA is used to find Nash Equilibrium points of nonlinear non-cooperative games. The proposed method can also be used as an alternative approach to solve multi-objective optimization problems. The effectiveness of the proposed method, in comparison to Genetic Algorithm, is proven through several static and dynamic example games and also multi-objective problems.

Springer LNCS, ICCSA 2008, Italy

Adaptive Beamforming Using Colonial Competitive Algorithm

Mahnaz Roshanaei and Esmail Atashpaz-Gargari, Caro Lucas

Abstract: Recently introduced Colonial Competitive Algorithm (CCA) has shown its excellent capability on diverse optimization tasks. In this study a Uniform Linear Array (ULA) adaptive antenna that uses this global search heuristic is developed. The obtained results are compared with those of a Genetic Algorithm (GA) and Least Mean Square (LMS). The evolutionary algorithms are applied to the problem of beamforming based on minimizing Signal-to-Interference-plus-Noise-Ratio (SINR). The results show not only GA and CCA perform better than LMS in both parts, but also CCA outperforms GA and LMS in these parts.

2nd International Joint Conference on Computational Engineering, KMCK 2008, Vancouver, Canada

Vehicle Fuzzy Controller Design Using Imperialist Competitive Algorithm

Ashkan MohammadZadeh Jasour, Esmail Atashpaz Gargari, Caro Lucas

Abstract: In this paper a novel socio-politically inspired optimization strategy, an Imperialist Competitive Algorithm, is applied to the problem of designing a fuzzy controller. The goal is to design a controller to enhance the transient and steady state behaviour of the system output. Having fixed the rule base of the fuzzy system, the controller is designed through determining the membership functions of the input and output variables. The design method is applied to a model of vehicle. The inputs of the fuzzy system are velocity of the vehicle and the slope of the road. The fuzzy controller controls the speed of the vehicle by adjusting the amount of gas into vehicle engine. Comparison results among the designed controller and the controller designed by the expert show that the controller obtained by Imperialist competitive algorithm has better performance than the expert controller.

Second Iranian Joint Congress on Intelligent and Fuzzy Systems, September 2007, Mashhad, Iran.