بهینه سازی اقتصادی واحد بازیابی گاز طبیعی مایع با استفاده از ژنتیک الگوریتم با جست‌وجوی مجذوری

ترانه جعفری بهبهانی [*۱،۲]، زهرا جعفری بهبهانی [۳]

[۱]عضو هیئت علمی پژوهشگاه صنعت نفت، پژوهشکدهٔ توسعهٔ فناوری‌های پالایش و فراورش نفت، تهران، استان تهران، ایران، صندوق پستی ۱۳۷- ۱۴۶۶۵

[۲]دانشگاه صنعتی خواجه نصیرالدین طوسی، تهران، ایران

[۳]دانشگاه آزاد اسلامی، واحد مرودشت ،گروه ریاضیات، مرودشت، ایران

دریافت: ۹۲/۱۱/۲۳ پذیرش: ۹۳/۱۱/۱

چکیده

ژنتیک الگوریتم با جستجوی مجذوری یک روش ترکیبی برای بهینه سازی اقتصادی یک واحد فرآیندی مدل شده با نرم افزارهای شبیه سازی فرآیند می باشد. از طریق ترکیب الگوریتم ژنتیک عادی با الگوریتمی بر اساس جستجوی مجذوری می‌توان تعداد محاسبات مربوط به توابع هدف را کاهش داد. روش پیشنهادی شامل مزایای ژنتیک الگوریتم عادی و تکنیک‌های جستجوی مجذوری می باشد. از جمله این مزایا تعیین بهینه مطلق توابعی با احتمال ناپیوستگی بالا و با همگرایی بیش‌تری نسبت به ژنتیک الگوریتم عادی می‌باشد. در این پژوهش بهینه سازی اقتصادی واحد بازیابی گاز طبیعی مایع با استفاده از ژنتیک الگوریتم با جستجوی مجذوری و ژنتیک الگوریتم عادی با یکدیگر مقایسه شده‌اند. نتایج نشان می‌دهد در صورت استفاده از پارامترهای ژنتیکی یکسان، همگرایی ژنتیک الگوریتم با جستجوی مجذوری و همچنین دقت روش پیشنهادی بهتر از ژنتیک الگوریتم عادی می باشد.

کلمات کلیدی: ژنتیک الگوریتم، جستجوی مجذوری، فرآیند بازیابی گاز طبیعی مایع، بهینه سازی

مقدمه

نرم افزارهای تجاری شبیه سازی فرآیند مانند ASPEN، HYSYS و PRO II کاربرد گسترده و توسعه پرشتابی در شبیه سازی های صنعتی پیدا کرده اند که مهم‌ترین دلیل آن توانایی این شبیه سازها در ایجاد مدل‌های دقیق بر پایه ترمودینامیک و نیز مدل‌هایی قابل انعطاف برای تجهیزات فرآیندی می باشد که باعث ساده تر شدن رویه شبیه سازی فرآیند، برای مهندسین فرآیند شده است[۱]. هنگامی که از یک شبیه ساز فرآیند، برای بهینه سازی فرآیند استفاده می شود، لازمه همگرایی کل فرآیند، محاسبه مقادیر مربوط به تابع

*jafarit@ripi.ir

هدف می باشد و با توجه به این نکته که کل فرآیند شامل چندین واحد مستقل می باشد، معمولا تنها در صورت همگرایی تک تک واحدهای مستقل و به کار بردن تکنیک های مناسب برای همگرایی جریان‌ها، همگرایی کل فرآیند حاصل می شود. این ویژگی شبیه سازهای فرآیند سبب شده که استفاده از روش‌های بهینه سازی که پایه آن‌ها براساس گرادیان است با مشکلات فراوانی مواجه شود؛ زیرا در این حالت ممکن است که داده های لازم برای محاسبه گرادیان، همواره وجود نداشته باشد و از طرف دیگر این احتمال وجود دارد که قسمت‌های مختلفی از فرآیند از خود یک رفتار غیر خطی نمایش دهند که در این حالت حل مساله بهینه سازی می تواند دارای چندین بهینه محلی باشد. روش‌هایی که پایه آن‌ها بر اساس گرادیان نیست نظیر ژنتیک الگوریتم و الگوریتم های تکامل تدریجی عموما برای حل مسائل بهینه سازی که دارای چندین بهینه محلی بوده و یا فاقد داده های لازم برای محاسبه گرادیان هستند، مورد استفاده قرار می گیرند. ژنتیک الگوریتم از جمله پرکاربرد ترین الگوریتم هایی می باشد که از الگوریتم جستجوی اتفاقی استفاده می‌کنند و در کلاس خاصی از مسائل بهینه سازی نظیر مسائل بهینه سازی در فضاهای غیر محدب بر سایر روش‌هایی که دارای پایه گرادیانی هستند ترجیح داده می شود، با این حال ژنتیک الگوریتم دارای این عیب می‌باشد که به علت طبیعت اتفاقی این روش نمی توان تعداد مراحل لازم برای رسیدن به یک دقت مشخص را تعیین نمود که این عامل ممکن است سبب به وجود آمدن حجم زیادی از محاسبات ریاضی شود. در صورتی که برای بهینه سازی یک فرآیند از ژنتیک الگوریتم به همراه یک شبیه ساز فرآیند استفاده شود، تعداد محاسبات مربوط به تابع هدف عامل مهمی در تعیین حجم محاسبات انجام شده است، به این دلیل که تعیین هر کدام از اعضای جمعیت ژنتیک الگوریتم نیاز به همگرایی کل فرآیند دارد و از آن‌جا که زمان محاسبات لازم برای همگرایی کل فرآیند در شبیه سازی خیلی بیش‌تر از زمان محاسبات انجام شده توسط ژنتیک الگوریتم می باشد انتخاب و به کار بردن یک الگوریتم جستجوی کارآمد که کم‌ترین نیاز را به محاسبه تابع هدف داشته باشد از اهمیت زیادی برخوردار است[۱،۲،۳]. هلند[1] نخستین فردی بود که ایده مربوط به ژنتیک الگوریتم را مطرح کرده و آن را گسترش داد و پس از آن این روش کاربرد های متنوعی در حل مسائل بهینه سازی مختلف پیدا کرد و در این مدت پارامتر های گوناگونی برای ایجاد جمعیت[2] های متنوع و اجتناب از همگرایی نارس[3] (قرار گرفتن جواب در یک بهینه محلی و عدم امکان بهبود) به آن اضافه گردید [۴].

هیبریداسیون[4] به معنای ترکیب ژنتیک الگوریتم با الگوریتمی با پایه گرادیانی برای افزایش کارایی عملیات جست‌وجو می باشد. یکی از پرکاربرد ترین روش ها در این مورد الگوریتمی می باشد که از ژنتیک الگوریتم به منظور جستجوی پاسخ در تمام فضا استفاده می کند[۵]. زمانی که از جستجو با پایه گرادیان برای تعیین کردن بهینه محلی با شروع از اعضای جمعیت استفاده می شود، ترکیب جدیدی از ژنتیک الگوریتم بر اساس

[1]Holland
[2]Population
[3]Premature Convergence
[4]Hybridization

تکامل تدریجی[۱] داروین[۲] و لامارکین[۳] ارایه شده است که در آن برازندگی[۴] برای نسل های جدید[۵] تولید شده با اپراتورهای عمومی ژنتیک با به کار بردن روش سنتی hill-climbing اصلاح می شود و سپس نسل های جدید برای انتخاب به عنوان اعضای نسل بعدی با مولد[۶] های خود به رقابت می پردازند[۶]. ترکیب شناخته شده دیگر، استفاده هم زمان از ژنتیک الگوریتم و برنامه نویسی خطی می باشد جایی که در آن از ژنتیک الگوریتم برای یافتن پاسخ های محتمل در فضای سیستم استفاده شده و سپس به وسیله برنامه نویسی خطی[۷] و با استفاده از پاسخ های ژنتیک الگوریتم به عنوان نقاط ابتدایی، بهینه سازی انجام می شود. ترکیب دیگر رایج و مورد استفاده، به کارگیری همزمان ژنتیک الگوریتم و روش پاول[۸] می باشد، که در این ترکیب از ژنتیک الگوریتم و روش پاول اصلاح شده به طور متناوب و با توجه به یک سری معیارهای مشخص استفاده می شود[۳،۷]. کاربرد های ژنتیک الگوریتم ترکیبی، تمام رشته های مهندسی از مهندسی شیمی گرفته تا مهندسی هوانوردی را شامل می‌شود. ژنتیک الگوریتم با جستجوی مجذوری ترکیبی از ژنتیک الگوریتم عادی با یک زیر مجموعه بهینه ساز بوده که بر اساس الگوریتم جستجوی مجذوری از اعضای نزدیک به پاسخ بهینه تقریبی استفاده می کند، این تکنیک ممتاز در ژنتیک الگوریتم به صورت ذاتی سبب می شود که نتایج به سمت یک خوشه، اطراف بهینه مطلق یا یک بهینه محلی مورد نظر هدایت شوند [۸]. این خاصیت اغلب سبب ایجاد همگرایی نارس در ژنتیک الگوریتم عادی می شود. ژنتیک الگوریتم با جستجوی مجذوری از اعضای جمعیت خوشه[۹] اطراف بهینه مطلق برای ایجاد یک مدل مجذوری استفاده کرده که با یک روش صریح ریاضی قابل حل است. فرض اساسی در این تکنیک، خوشه ای است که احتمالا در ناحیه محدب، اطراف بهینه مطلق حضور دارد و کوچک‌ترین شعاع این خوشه که نزدیک‌ترین پاسخ به بهینه مطلق است توسط مدل مجذوری تعیین می شود. در نتیجه این خواص، ژنتیک الگوریتم با جستجوی مجذوری تعداد محاسبات مربوط به مقادیر تابع هدف را به منظور تعیین بهینه ای با دقت قابل قبول، به طور قابل توجهی کاهش می دهد. در این پژوهش با استفاده از چهار تابع آزمایشی بهینه سازی اقتصادی واحد بازیابی گاز طبیعی مایع با استفاده از ژنتیک الگوریتم با جستجوی مجذوری و ژنتیک الگوریتم عادی با یکدیگر مقایسه شده اند.

ژنتیک الگوریتم

از ژنتیک الگوریتم در مسائل بهینه سازی متنوعی استفاده می شود که دلیل آن قابلیت انعطاف زیاد و پیشرفت های اخیر حاصله در تئوری پشتیبان این روش می باشد. موارد استفاده از ژنتیک الگوریتم در علوم

[1]Evolution
[2]Darwin
[3]Lamarckian
[4]Fitness
[5]Offspring
[6]Parent
[7]Simplex
[8]Powell
[9]Cluster

بیولوژی، پزشکی، کامپیوتری، اجتماعی و مهندسی می باشد و همچنین ژنتیک الگوریتم کاربردهای فراوانی درعرصه های مختلف مهندسی شیمی نظیر کنترل فرآیند، طراحی خطوط لوله گاز، مدل‌سازی فرآیندهای چند متغیره، بهینه سازی پارامترهای مربوط به سرعت واکنش های شیمیایی، طراحی واحدهای شیمیایی ناپیوسته چند منظوره و زمان بندی دارد[۹]. ژنتیک الگوریتم برای حل مسایل بهینه‌سازی از روش‌های انتخاب طبیعی و ژنتیک طبیعی استفاده می کند و آغاز به کار آن از طریق یک جمعیت ابتدایی تولید شده براساس اتفاق می باشد و تا زمان حاصل شدن پاسخ، جمعیت‌های جدید به طور مکرر از جمعیت‌های مادر ساخته می شوند. بر اساس تئوری تنازع و بقا داروین تنها اشخاصی که با محیط طبیعی تطابق پیدا می کنند توانایی لازم برای حضور در نسل های بعدی را دارند. ژنتیک الگوریتمی که در مقاله حاضر مورد استفاده قرار می گیرد مطابق الگوریتم عمومی ارایه شده توسط Gen and Cheng (1997) می باشد، با این حال چند تغییر در ژنتیک الگوریتم عادی برای برقرای تنوع جمعیت در هر نسل ایجاد شده است[۱۰]. جمعیت ابتدایی به صورت اتفاقی انتخاب می شود و مسائل بهینه سازی مقید نیاز به یک سری تکنیک‌های کنترل قید نظیر اپراتورهای ژنتیکی دارند، نسل‌های جدید توسط اپراتورهای ژنتیکی، اپراتور قطع و جهش و یک ضریب اتفاقی برای بهبود تنوع جمعیت و روش‌های متنوع برای انتخاب اتفاقی اشخاص حاضر در نسل جدید تولید می شوند. اکثریت نسل جدید تولید شده حاصل اپراتور قطع می باشد و مابقی آن نیز از اپراتور جهش و اپراتور اتفاقی نتیجه می شوند. نسل های جدید تولید شده در محلی به عنوان استخر جمعیت ۱ به همراه مولد های خود ذخیره می شوند و تعداد زیادی از اشخاص نسل آینده از استخر جمعیت و با استفاده از روش چرخ رولت ۲ انتخاب می شوند[۱۱]. شرح جزئی مطالب اشاره شده به همراه کدهای برنامه نویسی آن‌ها در مراجع وجود دارد. استفاده از این روش سبب می شود اشخاصی که دارای شانس بیش‌تری برای بقا بوده به همراه عناصر اتفاقی وارد نسل جدید شوند. این الگوریتم بهترین اشخاص را برای حضور در نسل جدید به کار می برد همچنین در این الگوریتم تعداد محدودی از اشخاص بدون در نظر گرفتن برازندگی به‌طور اتفاقی انتخاب شده و برای حفظ و ایجاد تنوع وارد نسل جدید می شود[۱۲]. نسل تولید شده توسط عناصر اتفاقی عمدتا دارای برازندگی کم‌تری بوده و از شانس کم‌تری برای حضور در نسل آینده برخوردار هستند اما زمانی که از روش انتخاب اتفاقی در مقایسه با روش چرخ رولت استفاده می شود اشخاصی که دارای برازندگی کم‌تری هستند ممکن است دارای یک شانس بهبود یافته برای حضور در نسل آینده باشند. بعد از کامل شدن تعداد مشخصی از نسل‌ها، بهینه آخرین نسل تولید شده به عنوان پاسخ مساله بهینه سازی در نظر گرفته می شود.

معرفی اشخاص

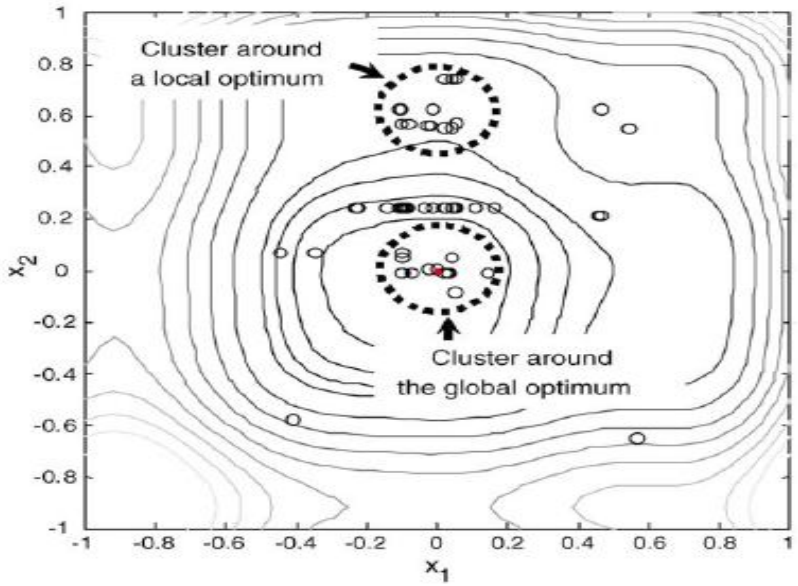
یک رشته ۳ عددی متشکل از ارقام صفر و یک، کروموزم ۴ نامیده می شود که در واقع معرف یک شخص است هر کروموزوم شامل چندین ژن بوده که قسمت‌های اصلی تشکیل دهنده هر رشته می باشند هر کدام

[1]Population Pool
[2]Roulette Wheel
[3]String
[4]Chromosome

از ژن‌ها متعلق به یکی از متغیرهای سیستم می باشد و کروموزم نیز می‌تواند مشخص کننده بردار سیستم باشد و از آن‌جا که یک دسته از ارقام ناپیوسته متغیرهای سیستم را مشخص می‌کنند لذا عامل تعیین کننده دقت، تعداد ژن های موجود در هر رشته است.

اپراتورهای ژنتیک

اپراتورهای ژنتیک رایج و پرکاربرد اپراتورهای متقاطع[1] و جهش یافته[2] می باشند که از اپراتور متقاطع به منظور بهبود کیفیت متوسط یک جمعیت و از اپراتور جهش یافته به منظور ایجاد تنوع و پوشش دادن کل فضای متغیرها استفاده می شود. علاوه بر اپراتورهای ژنتیک رایج، استفاده از یک اپراتور اتفاقی می‌تواند تنوع را در هر نسل برقرار کند. انتخاب اپراتورهای ژنتیک برای تولید نسل جدید به صورت اتفاقی و بر اساس نوع تولید نسل در هر حلقه صورت می‌گیرد. اپراتورهای متقاطع و جهش یافته نیاز به انتخاب جفت مولد/مولد برای تولید نسل جدید را دارند در حالی که اپراتورهای ژنتیک اتفاقی می‌توانند بدون حضور مولد اقدام به تولید نسل جدید کنند. انتخاب جفت مولد/مولد برای اپراتورهای ژنکیکی متقاطع و جهش یافته با استفاده از روش چرخ رولت بر اساس برازندگی متناظر با برازندگی مقدار تابع هدف صورت می‌گیرد و در این حالت اغلب اشخاصی که دارای برازندگی بیش‌تری نسبت به تابع هدف هستند انتخاب می شوند.



شکل۱. نمونه از خوشه های تشکیل شده حول بهینه مطلق و محلی[۴]

در اپراتور ژنتیکی متقاطع برای تولید اعضای نسل جدید ابتدا دو رشته مولد انتخاب می شوند و سپس این دو رشته از یک نقطه که به صورت اتفاقی انتخاب می شود به دو قسمت تقسیم می شوند سپس قسمت‌های جدا شده از رشته‌های مختلف به یکدیگر متصل شده و دو عضو نسل جدید حاصل می شوند. در

[1]Crossover
[2]Mutation

اپراتور ژنتیکی جهش یافته نیز برای تولید اعضای نسل جدید ابتدا یک رشته مولد توسط روش چرخ رولت به صورتی اتفاقی انتخاب شده سپس یک تغییر در رقم دودویی موجود در قسمتی از رشته که به‌صورت اتفاقی انتخاب می شود ایجاد می‌گردد که نتیجه آن تولید یکی از اعضای نسل جدید است. اگر که یکی از اعضای ایجاد شده توسط اپراتورهای ژنتیک در نسل جدید، شبیه یکی از اشخاص جمعیت مادر خود باشد از بین می‌رود و عملیات تا زمانی که عضو نسل جدید هیچ گونه شباهتی با اشخاص حاضر در جمعیت مادر خود نداشته باشد ادامه پیدا می کند، که این خاصیت سبب ایجاد تنوع بیش‌تر در نسل حاضر می شود و از آنجا که این الگوریتم به کار رفته اجازه انتقال یک شخص از یک نسل به نسل دیگر را می دهد لذا به تولید چندین عضو با ژن‌های یکسان نیازی نمی باشد[۱۳]. در شکل ۱ دو خوشه یکی حول بهینه مطلق و دیگری حول بهینه محلی به نمایش در آمده است، در حالتی که خوشه حول بهینه مطلق تشکیل شده باشد جستجوی مجذوری منجر به نزدیک شدن پاسخ به بهینه مطلق می گردد. با توجه به فرض اساسی موجود در روش جستجوی مجذوری، زمانی که اندازه خوشه به حد کافی کوچک می شود فضای متعلق به خوشه می‌تواند توسط یک مدل مجذوری تقریب زده شود و پاسخ حاصل شده در این حالت بسیار نزدیک به بهینه واقعی در توابع هدف پیوسته است[۱۴].

محاسبه برازندگی

روش‌های متنوعی برای محاسبه برازندگی اعضای یک جمعیت وجود دارد و انتخاب هرکدام از این روش‌ها بر نحوه گزینش مولدها برای اپراتورهای ژنتیکی تاثیر گذار است. تا کنون روش‌های گوناگونی برای محاسبه برازندگی اعضای یک جمعیت بررسی شده است و در این مقاله ترکیبی از روش‌های normalization و Ranking مورد استفاده قرار گرفته است. برازندگی حاصل از ترکیب این دو روش همان‌طور که در ادامه شرح داده می‌شود از یک ثابت تناسب استفاده می‌کند. برازندگی یک عضو حاصل از روش normalization مطابق زیر است:

$$\bar{f}_k^{\,n} = \frac{f_{\max} - f_k}{f_{\max} - f_{\min}}$$

که در آن $\bar{f}_k^{\,n} \in [0,1]$ و :

$f_{\min}$: کم‌ترین مقدار تابع

$f_{\max}$: بیش‌ترین مقدار تابع

برازندگی یک عضو حاصل از روش ranking مطابق زیر است :

$$\bar{f}_k^{\,r} = \frac{r^k}{n_p}$$

که در آن $\bar{f}_k^{\,r} \in (0,1]$ و :

r_k: رتبه به ترتیب کاهش اعضا

n_p: اندازه جمعیت

و با توجه به روابط بالا برازندگی نهایی هر کدام از اعضای جمعیت از رابطه زیر حاصل می شود:

$$\bar{f}_k = \alpha \bar{f}_k^{\,n} + (1-\alpha)\bar{f}_k^{\,r}$$

که در آن $\alpha \in (0,1)$

کنترل و مدیریت قید

مسائل بهینه سازی که دارای قیدهای ساده‌تر هستند نسبت به مسائلی که قیدهای پیچیده‌تری دارند توسط ژنتیک الگوریتم راحت‌تر حل می شوند که این ناشی از عناصر اتفاقی حاضر در اپراتورهای ژنتیکی قطع و جهش می باشد که ممکن است سبب قرار گرفتن نسل های جدید در مناطق غیر قابل قبول در مسائلی با قیدهای سخت باشد. یک راه مناسب برای برخورد با این نوع مسائل فرمول‌بندی مجدد این نوع مسائل به‌صورتی است که قیدهای موجود به صورت توابع جریمه[1] در تابع هدف ظاهر شوند؛ این فرمول بندی جدید سبب می‌شود که اعضای نسل جدید که باعث نقض شدن قیود می‌شوند دارای برازندگی کم‌تری شده و نتوانند ژن های خود را به نسل بعدی انتقال دهند.

جست‌وجوی مجذوری

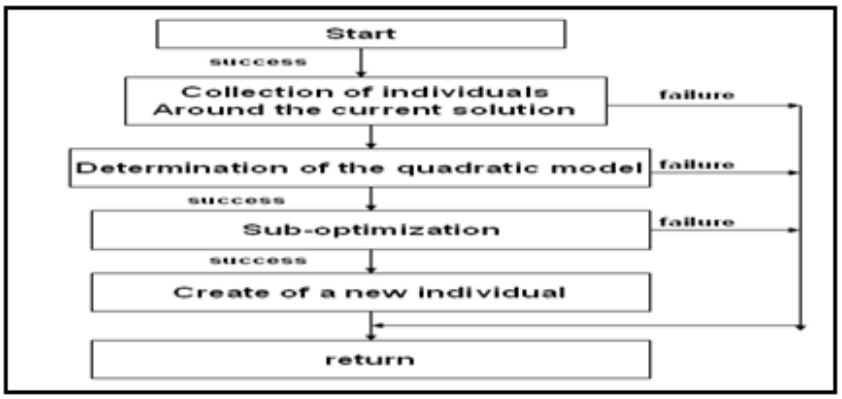
اگر چه تعداد زیادی از کاربردهای موفق ژنتیک الگوریتم نشان دهنده این موضوع است که ژنتیک الگوریتم روش مناسبی برای برخورد با مسائل بهینه سازی ناپیوسته و بهینه سازی‌های ترکیبی می باشد با این حال هزینه‌های زیاد مربوط به محاسبات کامپیوتری این روش همواره به عنوان یکی از مشکلات اساسی این روش مطرح بوده است. ژنتیک الگوریتم با جستجوی مجذوری یک روش ترکیبی از بهینه سازی جبری و اتفاقی می باشد که می‌تواند هزینه مربوط به محاسبات پر حجم این روش را تا مقدار قابل ملاحظه‌ای کاهش دهد، جست‌وجوی مجذوری تعداد محاسبات مربوط به میزان تابع هدف را تا رسیدن به مقدار بهینه کاهش می‌دهد. جست‌وجوی مجذوری که در پایه یک جست‌وجوی جبری است می‌تواند سرعت همگرایی، ژنتیک الگوریتم عادی را افزایش دهد که دلیل آن جمعیت خوشه‌های ایجاد شده اطراف یک یا چند بهینه محلی بعد از چندین مرتبه تکرار است، هر چند که، همگرا شدن به مقدار دقیق بهینه محلی در روش ژنتیک الگوریتم عادی نیاز به تولید تعداد قابل توجهی نسل های جدید دارد.

شکل ۲ نشان دهنده یک برنامه ساده برای ژنتیک الگوریتم با جست‌وجوی مجذوری است، یک مزیت عمده جست‌وجوی مجذوری یافتن بهینه مطلوب برای یک تابع هدف مجذوری با یک بار تکرار است که نتیجه مستقیم آن کاهش قابل توجه تعداد محاسبات مربوط به تابع هدف است. علاوه بر این ژنتیک الگوریتم می‌تواند به راحتی با جست‌وجوی مجذوری تعامل داشته باشد بدون این‌که هیچ کدام از خواص منحصر به فرد خود را از دست بدهد[۱۵].

[1] Penalty Function

```
START
Generate initial population;
k=0;
Repeat
   k=k+1;
   i=0;
   Repeat
      i=i+1;
      Choose genetic operator at random;
      if (mutation selected);perfume mutation;
      if (crossover selected);perfume crossover;
      i=i+1;
      if (random selected);perfume random;
      if (i=population size-1);perfume mutation
   Until (i<population size)
   Update the optimal point;
   Select individuals for the next population
Until (k < desired number of generation)
Print the optimal point
END
```

شکل۲. برنامه ژنتیک الگوریتم با جستجوی مجذوری



شکل۳. فلوچارت جست‌وجوی مجذوری

فلوچارتی که شرح دهنده مراحل مختلف جست‌وجوی مجذوری می باشد در شکل ۳ مشخص شده است. ژنتیک الگوریتم با جستجوی مجذوری از این خاصیت استفاده کرده و اقدام به تعیین یک مدل مجذوری با استفاده از اعضای متعلق به خوشه های اطراف نقاط بهینه می کند، که بهینه این مدل‌های مجذوری از طریق تحلیلی قابل تعیین شدن است.

جدول ۱. توابع امتحانی

نام تابع	تابع	محدوده	مینیمم
F1		$-5.12<x_i<5.12$	0 at $\forall xi$ =0
F2	$100(x_1^2-x_2)^2+(1-x_1)^2$	$-2.048<x_i<2.048$	0 at $\forall xi$ =1
F3	$25+\sum_1^3 integer\ (x_i)$	$-5.12<x_i<5.12$	0 at $\forall xi$
F3		$-1.28<x_i<1.28$	0 at $\forall xi$ =0

جدول۲. پارامترهای استفاده شده در ژنتیک الگوریتم عادی و ژنتیک الگوریتم با جستجوی مجذوری

	F1	F2	F3	F4	PE
الگوریتم ژنتیک					
Population size	۳۰	۲۰	۶۰	۶۰	۳۰
Number of offspring	۱۵	۱۰	۳۰	۳۰	۱۵
Number of generations	۱۰۰	۳۰۰	۱۵۰	۵۰۰	۱۰۰
Number of binary digits in each gene	۱۲	۱۲	۱۲	۱۲	۱۲
Probability of mutation	۰/۰۵	۰/۰۵	۰/۰۵	۰/۰۵	۰/۰۵
Probability of random	۰/۰۵	۰/۰۵	۰/۰۵	۰/۰۵	۰/۰۵
Probability of crossover	۰/۰۹	۰/۰۹	۰/۰۹	۰/۰۹	۰/۰۹
Probability of the random Selection of parents in crossover	۰/۰۵	۰/۰۵	۰/۰۵	۰/۰۵	۰/۰۵
Rate of fitness based Individual selection for the next generation	۰/۷	۰/۷	۰/۷	۰/۷	۰/۷
Rate of random individual selection for the next generation	۰/۳	۰/۳	۰/۳	۰/۳	۰/۳
الگوریتم مجذوری					
Factor for the number of data (w)	۱/۲	۱/۲	۱/۲	۱/۲	۱/۵
Number of past generations for quadratic modeling	۱۰	۱۰	۱۰	۱۰	۱۰

PE: بهینه سازی اقتصادی فرآیند

توابع مورد استفاده در جداول ۱ و ۲ ظاهر شده‌اند و تمامی پارامترهای مربوط به هر دو روش و در هر تابع امتحانی نیز در این جدول آمده است که این پارامترها شامل تمامی نسل‌های جدید، منجر شده از اپراتورهای ژنتیکی قطع، جهش و انتخاب اتفاقی، بر اساس احتمال استفاده از هر کدام از این اپراتورها می‌باشد. هر کدام از نقطه‌های نشان داده شده در واقع متوسطی از صد شبیه سازی انجام شده است و تمامی پارامترهای مورد استفاده در این الگوریتم بهینه‌سازی برابر مقادیر رایجی هستند که در این گونه بهینه‌سازی‌ها مورد استفاده قرار می‌گیرند. علاوه بر نسل‌های جدیدی که توسط ژنتیک الگوریتم عادی تولید می‌شود خود جست‌وجوی مجذوری منجر به تولید یک نسل جدید می‌شود که تا زمان موفق بودن این روش جست‌وجو، در استخر جمعیت حضور دارند. جست وجوی مجذوری دارای چهار الگوریتم زیر مجموعه، مهم می‌باشد که شامل جمع آوری اطلاعات لازم، تعیین مدل مجذوری، انجام یک بهینه سازی جزئی بر اساس مدل مجذوری و ایجاد یک سری اعضای جدید بر اساس پاسخ بهینه سازی مجذوری می باشد[۱۶]. با توجه به اطلاعات مشخص شده در شکل، تولید هر عضو جدید، که نیاز به یک بار محاسبه میزان تابع هدف دارد، قبل از تکمیل موفقیت‌آمیز تمام الگوریتم‌های زیر مجموعه دیگر صورت نمی‌گیرد. تمامی داده‌های لازم برای ایجاد یک مدل مجذوری از اعضای حاضر در نسل قبلی و بر اساس فاصله این اعضا با بهینه جاری، فراهم می‌شود. اعضایی که کم‌ترین فاصله را با بهینه جاری داشته باشند برای ایجاد مدل مجذوری مورد استفاده قرار می‌گیرند و هرچند که به علت ذخیره شدن اعضای نسل قبلی در حافظه، نیازی به نگه داری این اعضا در نسل جاری نمی‌باشد ولی به سبب نیاز به یک جمعیت بزرگ برای انجام بهینه سازی چند بعدی و اطمینان از این موضوع که تعداد اعضای تشکیل دهنده خوشه برای تعیین مدل مجذوری کافی باشد، این عمل صورت می‌گیرد. برای ایجاد یک مدل مجذوری حتما باید تعداد اعضای مورد استفاده در تعیین مدل، از تعداد پارامترهای لازم برای شناخت مدل مجذوری، بیش‌تر باشد، تعداد نقاط مورد نیاز از طریق رابطه زیر تعیین می شود:

$$m = \min\left\{ g \in N \;\middle|\; g \geq \omega \frac{(n+1)(n+2)}{2} \right\}$$

جایی که:

m: کم‌ترین تعداد نقاط مورد نیاز

n: بعد سیستم

ω: عددی حقیقی که بزرگتر یا مساوی واحد است.

بعد از محاسبه تعداد داده های لازم، می توان مدل مجذوری را به صورت زیر نمایش داد:

$$F = XB$$

جایی که:

F: بردار مربوط به مقادیر تابع هدف با در نظر گرفتن اعضای به کار رفته در مدل مجذوری؛

X: شامل قسمت هایی از مدل مجذوری می باشد که وابسته به اعضا هستند؛

B: بردار مربوط به پارامتر های مدل مجذوری می باشد.

$$F=\begin{bmatrix} f_1 \\ \vdots \\ f_m \end{bmatrix} \quad , \qquad B=\begin{bmatrix} b_1 \\ \vdots \\ b_{\frac{(n+1)(n+2)}{2}} \end{bmatrix}$$

در صورتی که X^TX یک مقدار مثبت تعریف شده باشد مدل مجذوری می‌تواند توسط روش حداقل مربعات تخمین زده شود که در واقع مبین این مطلب است که رتبه X برابر n است. بعد از تعیین پارامترهای بردار B نتایج بهینه سازی جزئی به صورت زیر است:

Minimize:

$$q=\frac{1}{2}x^THx+f^Tx+c$$

Subject to:

$$g_i(x)\le 0 \quad i=1,...,k$$

$$h_j(x)=0 \quad j=1,...,l$$

جایی که:

H : ماتریس متقارن هسین

f : بردار گرادیان

k : تعداد قید های نامساوی

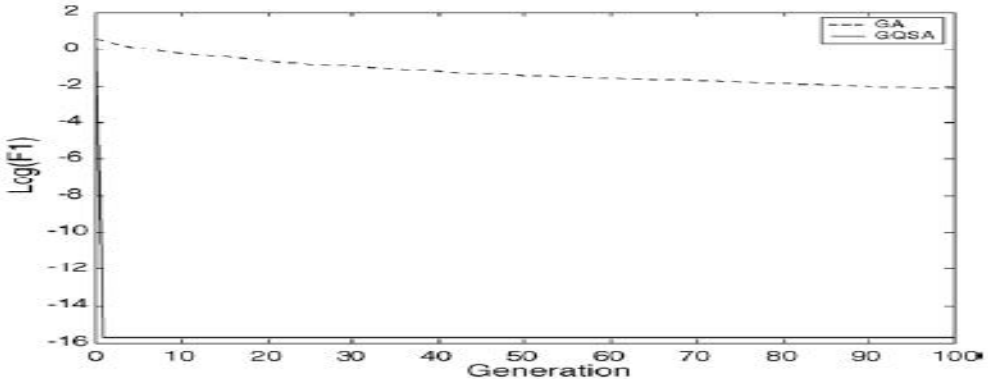
l : تعداد قید های مساوی

H و f و c در معادله قبلی با تنظیم عناصر B حاصل می شوند، برای حل بهینه سازی جزئی، هر دو روش برنامه نویسی مجذوری و برنامه نویسی مجذوری متوالی مناسب هستند ولی برای انتخاب یکی از این روش‌ها باید به عواملی نظیر نوع قیود موجود (خطی، غیر خطی و ناپیوسته و ...) توجه نمود. جست‌وجوی مجذوری می‌تواند روش برنامه نویسی مجذوری را برای بهینه سازی مسائلی که تنها دارای قیود خطی هستند به کار ببرد در غیر این صورت باید از برنامه نویسی خطی متوالی و یا سایر روش های بهینه یابی استفاده کرد.

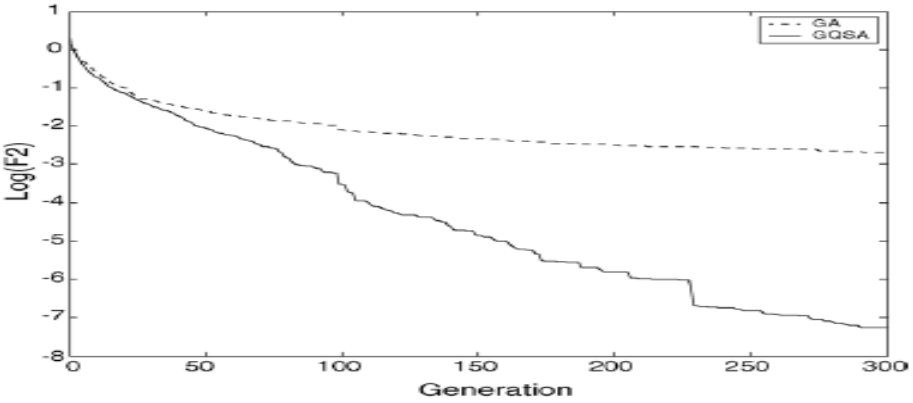
مقایسه کارایی ژنتیک الگوریتم عادی و ژنتیک الگوریتم با جستجوی مجذوری

برای مقایسه کارایی دو روش مذکور از چهار تابع امتحانی مختلف استفاده شده است که هر کدام از متغیرهای آن‌ها با یک کروموزم دوازده ژنی در ژنتیک الگوریتم مشخص شده‌اند و تمامی متغیرها دارای حدود بالا و پایین هستند. مساله به گونه‌ای فرمول بندی شده است که تمامی دوازده ژن موجود به طور دقیق کل پهنای متغیر را پوشش بدهند و در این صورت این اطمینان حاصل می شود که کل اعضای جمعیت تولیدی توسط ژنتیک الگوریتم امکان پذیر هستند. مقایسه بین این دو روش بر اساس بهترین اعضای موجود در هر نسل صورت می‌گیرد. الگوریتم مورد استفاده در این مقاله با نرم افزار MATLAB Version 6.0 برنامه نویسی شده و محاسبات لازم توسط یک دستگاه کامپیوتر شخصی انجام شده است.

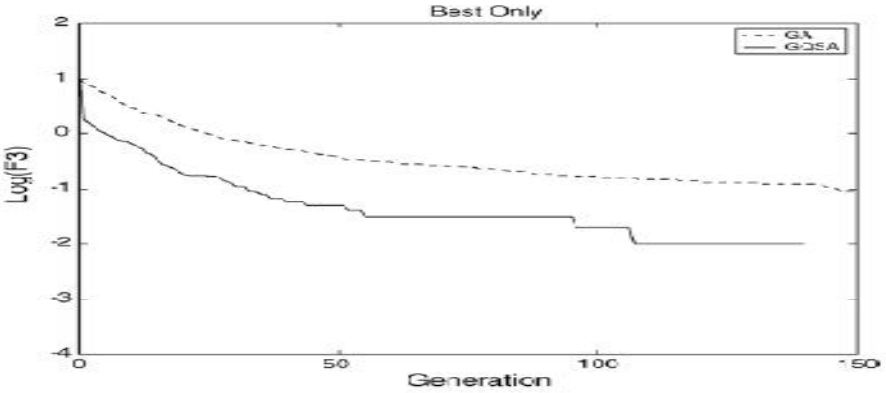
نتایج حاصل از شبیه سازی با هر دو روش در شکل‌های ۷-۴ نمایش داده شده‌اند و از آن‌جایی که برای تعیین مدل مجذوری F1 جمعیت ابتدایی به اندازه کافی بزرگ انتخاب شده است، ژنتیک الگوریتم با جست‌وجوی کوادراتیک می‌تواند مدل مجذوری مربوط به تابع F1 را محاسبه کرده و به سرعت اقدام به تعیین بهینه تابع بنماید و این در حالی است که ژنتیک الگوریتم عادی برای یافتن بهینه تابع F1 نیاز به تولید نسل‌های بیش‌تری دارد.



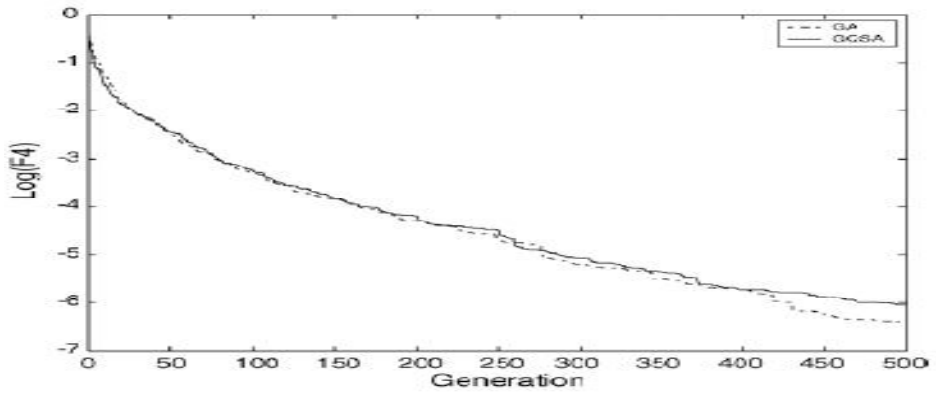
شکل ۴. مقایسه کارایی هر دو روش برای تابع F_1



شکل۵. مقایسه کارایی هر دو روش برای تابع F_2



شکل۶. مقایسه کارایی هر دو روش برای تابع F_3



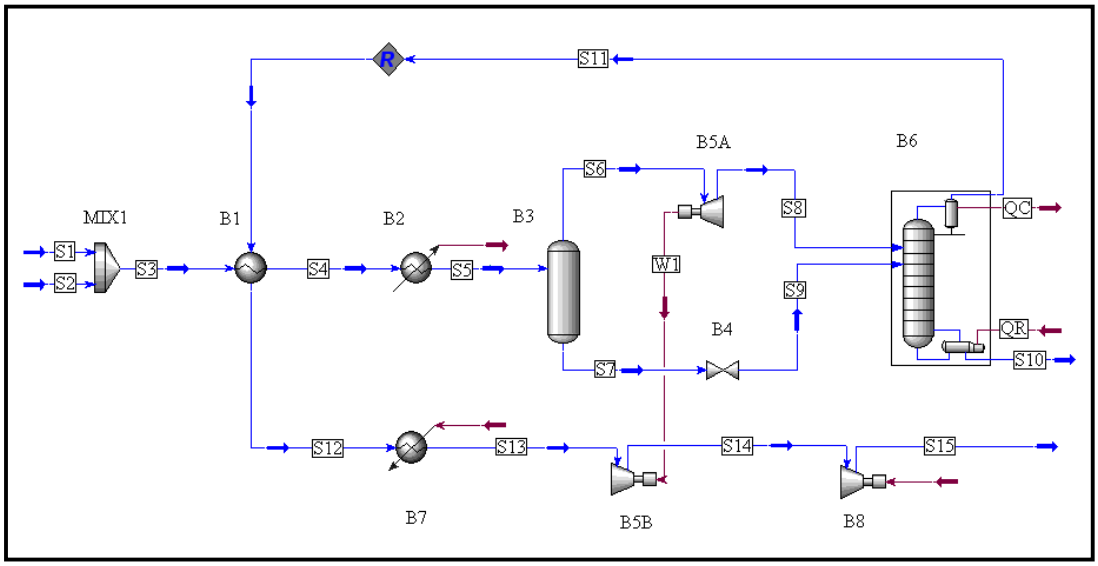
شکل۷. مقایسه کارایی هر دو روش برای تابع F_4

در مورد تابع F2 هر دو روش همگرایی یکسانی را پس از سی نسل نشان می دهند، در حالی که در نقطه branch-off بعد از سی نسل، همگرایی ژنتیک الگوریتم عادی به طور قابل توجهی کاهش پیدا کرده و این در حالی است که همگرایی ژنتیک الگوریتم با جست‌وجوی مجذوری سرعت بیش‌تری پیدا می‌کند. همان طور که در قسمت قبلی توضیح داده شد، کارایی جستجوی مجذوری، زمانی افزایش پیدا می‌کند که خوشه‌ای از اعضای موجود، حول بهینه مطلق تشکیل شود، بر خلاف تابع F1 بهینه سازی تابع F2 نیاز به تولید نسل های بیش‌تر برای ایجاد خوشه ای دارد که حاوی بهینه مطلق باشد و دلیل آن این است که F2 یک تابع مجذوری نبوده بنابراین در حالی که خوشه حول بهینه مطلق تشکیل شده است ولی جست‌وجوی مجذوری قادر به یافتن سریع بهینه مطلق نیست؛ زیرا تنها در یک همسایگی کوچک اطراف بهینه رفتار مجذوری مشاهده می شود. F3 تابعی بوده که علاوه بر متغیرهای پیوسته شامل متغیرهای ناپیوسته نیز می‌باشد، در این مورد نیز ژنتیک الگوریتم با جست‌وجوی مجذوری بسیار بهتر از ژنتیک الگوریتم عادی عمل کرده و همگرایی آن نیز، حتی در نسل‌های ابتدایی بسیار سریع‌تر است که دلیل آن مجذوری بودن تابع F3 می باشد. نمودار موجود در شکل ۶ نشان می‌دهد که ژنتیک الگوریتم با جستجوی مجذوری بعد از صدوچهل نسل به مقدار بهینه رسیده است در حالی که ژنتیک الگوریتم عادی هنوز همگرا نشده است. تابع F4 رفتاری متفاوت از سه تابع قبلی از خود نشان می‌دهد که دلیل آن این نکته است که تابع F4 اطراف نقطه بهینه دارای شیب کمی بوده که باعث ایجاد تداخل در جستجوی مجذوری می شود؛ زیرا در این حالت امکان ایجاد یک مدل مجذوری مناسب وجود ندارد با این حال باز هم کارایی این روش قابل مقایسه با ژنتیک الگوریتم عادی است.

بهینه سازی اقتصادی واحد فرآیندی

در این قسمت از الگوریتم توسعه یافته ژنتیک با جستجوی مجذوری برای بهینه سازی اقتصادی واحد بازیابی گاز طبیعی مایع به منظور جداسازی گاز طبیعی مایع از جریان گاز طبیعی خام در دماهای پایین استفاده می شود. این روش یک فرآیند پرکاربرد در بازیابی گاز طبیعی مایع بوده به این دلیل که جداسازی اتان و پروپان را از گاز طبیعی با راندمان بالایی انجام می دهد. نمای کلی یک فرآیند بازیابی گاز طبیعی

مایع در حال تحقیق در شکل ۸ نشان داده شده است. دو جریان خوراک S1 وS2 از منابع مختلف تهیه می‌شوند. جریان خوراک S1 رقیق بوده و شامل مقادیر زیادی متان می باشد (۹۱/۵ درصد مولی متان، ۷ درصد مولی اتان و پروپان، ۱/۵درصد مولی از سایر هیدروکربن ها). جریان خوراک S2 غلیظ بوده و شامل مقادیر زیادی از هیدروکربن‌های سنگین می باشد، (۳۲ درصد مولی متان، ۶۴ درصد مولی اتان و پروپان و ۴ درصد مولی سایر هیدروکربن‌ها). هر دو جریان خوراک ابتدا وارد مخلوط کننده (MIX1) شده و سپس جریان مخلوط شده (S3) در مبدل حرارتی (B1) با استفاده از جریان گاز (S11) خروجی از دی متانایزر (B6) سرد می شود. جریان سرد شده (S4) پس از عبور از چیلر (B2) با کاهش دما مواجه شده و جریان حاصل شده (S5) با کاهش فشار در درام (B3) به دو فاز مایع و گاز تبدیل می شود. جریان گاز (S6) خروجی از درام وارد expander (B5) شده و با کاهش دما و فشار مواجه می شود و از انرژی ایجاد شده توسط expander برای متراکم کردن مجدد جریان گاز خروجی از مبدل حرارتی استفاده می شود.



شکل۸. فرآیند بازیابی گاز طبیعی

دی متانایزر (B6) شامل ۱۹ سینی و جوش‌آور بوده که برای جداسازی جریان‌های محصول گاز و مایع استفاده می‌شود. برج دی متانایزر ۹۹ درصد از متان موجود در خوراک ورودی را بازیابی می‌کند و جریان خروجی از پایین دی‌متانایزر بدون هیچ فرآیند اضافی به عنوان گاز طبیعی مایع فروخته می‌شود و جریان بالای برج برای تبدیل به یک محصول قابل فروش نیاز به انجام چندین فرآیند اضافی در مبدل‌های سری (B1) و (B7) و کمپرسورهای (B5) و (B8) می‌باشد. زمانی که قیمت محصولات و هزینه خدمات تغییر کند واحد فرآیندی می‌تواند با تغییر یک سری شرایط عملیاتی موجود به شرایط بهینه اقتصادی جدیدی دست پیدا کند که مهم‌ترین این شرایط عملیاتی برای واحد تولید گاز طبیعی مایع عبارتند از:

- x1: دمای خروجی از چیلر B2 (-20 to 0 °F)

- x2: فشار عملیاتی درام B3 (600 to 1100 psi)
- x3: فشار خروجی از B5A expander (400-600 psi)
- x4: فشار بالای برج تقطیر B6 (260 to 300 psi)

در این حالت خواسته ما تعیین بیش‌ترین مقدار سود حاصل با تنظیم پارامترهای طراحی می‌باشد و از آن‌جا که ژنتیک الگوریتم با جست‌وجوی مجذوری تنها قادر به تعیین مینیمم مقدار یک تابع هدف است؛ لذا باید تغییراتی در مساله بهینه سازی اقتصادی واحد ایجاد شود:

Minimize:

$J(x)=-R_{Gas}(x)-R_{Liq}(x)+C_{Cap}+C_{Opr}(x)$

Subject to:

$-20 \le x_1 \le 0 \qquad 600 \le x_2 \le 1100$

$400 \le x_3 \le 600 \qquad 260 \le x_4 \le 300$

که در آن:

RGas: قیمت سالانه محصول گاز

RLiq: قیمت سالانه محصول مایع

CCap: هزینه سرمایه گذاری کلی به صورت سالانه

COpx: هزینه عملیاتی به صورت سالانه

جدول ۳ شامل تمامی اطلاعات لازم برای محاسبه تابع هدف می‌باشد و تمامی پارامترهای ژنتیک الگوریتم با جست‌وجوی مجذوری و ژنتیک الگوریتم در جدول ۲ نشان داده شده است و در این بهینه سازی باید یک بهینه سازی جزیی با روش مجذوری حل شود:

Minimize:

$$q = \frac{1}{2} x^T H x + f^T x + c$$

Subject to:

$-20 \le x_1 \le 0 \qquad 600 \le x_2 \le 1100$

$400 \le x_3 \le 600 \qquad 260 \le x_4 \le 300$

از آنجا که قیود مساله بهینه سازی جزیی غیر خطی بوده لذا از روش برنامه نویسی مجذوری می‌توان برای حل این مساله استفاده کرد.

ژنتیک الگوریتم با جست وجوی مجذوری و ژنتیک الگوریتم عادی از طریق نرم افزار MATLAB 6.0 کدنویسی شده و به نرم افزار شبیه ساز ASPEN plus جهت بهینه سازی اقتصادی فرآیند بازیابی گاز طبیعی مایع متصل شده است. کارایی ژنتیک الگوریتم با جست‌وجوی مجذوری و ژنتیک الگوریتم عادی برای بهینه‌سازی اقتصادی واحد فرآیندی بازیابی گاز طبیعی مایع در شکل ۹ نمایش داده شده است و هرکدام از نقاط موجود روی این نمودار حاصل متوسط‌گیری از ده تکرار پشت سر هم می باشد. نتایج این شبیه‌سازی نشان می‌دهد که همگرایی ژنتیک الگوریتم با جست‌وجوی مجذوری به طور قابل ملاحظه‌ای سریع‌تر از ژنتیک الگوریتم عادی به‌خصوص بعد از چند نسل ابتدایی می باشد. متوسط بیش‌ترین سود

حاصل در ژنتیک الگوریتم با جست‌وجوی مجذوری ۶,۰۷۷,۳۰۰ دلار و در ژنتیک الگوریتم عادی ۶,۰۵۷,۶۰۰ دلار می‌باشد. اما سرانجام در بین ده شبیه سازی انجام شده بهترین پاسخی که توسط ژنتیک الگوریتم با جست‌وجوی مجذوری یافته شده است سود ۶,۰۸۷,۵۰۰ دلار با شرایط عملیاتی: $x^* = (0 , 916.61 , 400.88 , 299.28)$ بوده و بهترین پاسخ ژنتیک الگوریتم عادی، سود ۶,۰۶۷,۷۰۰ دلار با شرایط عملیاتی: $x^* = (0.16606 , 890.35 , 400.29 , 299.36)$ می باشد. ژنتیک الگوریتم با جست‌وجوی مجذوری در کل ۲۰,۰۰۰ دلار سود بیش‌تری را نسبت به ژنتیک الگوریتم عادی پیش بینی می‌کند. تلاش‌هایی نیز در جهت اتصال شبیه ساز Aspen plus با نرم افزار MATLAB 6.0 به‌منظور بهینه‌سازی اقتصادی براساس روش‌هایی که بر پایه گرادیان هستند صورت گرفت که مشخص شد برنامه نوشته شده قادر به همگرا کردن مساله بهینه سازی نیست که علت آن ناتوان بودن این روش در تعیین جهت‌های کم شونده به علت وجود اختلالات در نتایج عددی حاصل از اتصال شبیه ساز با یک بهینه ساز خارجی می باشد. جدول شماره ۴ دقت توابع هدف را نشان می‌دهد.

جدول۳. اطلاعات مورد نیاز برای بهینه سازی اقتصادی واحد فرآیندی

تعداد ساعت های عملیاتی در سال (h/yr)	۸۵۰۰
کل هزینه سرمایه گذاری ($)	۷۱۷۵۰۰۶
کل هزینه سرمایه گذاری به طور سالانه ($/yr)	۹۹۲۷۱۷
قیمت محصول گازی ($/MMBTU)	۱.۲
قیمت محصول مایع ($/gal)	۰.۰۵
قیمت منبع حرارتی ($/MMBTU)	۱.۰
قیمت منابع خنک کننده ($/MMBTU)	۷.۰
قیمت الکتریسیته ($/kwh)	۰.۰۲۸

جدول۴. دقت الگوریتم های مورد مطالعه

الگوریتم	دقت
ژنتیک	٪۹۵
مجذوری	٪۹۸

نتایج

در این مقاله یک هیبرید جدید از الگوریتم ژنتیک، تحت عنوان ژنتیک الگوریتم با جست‌وجوی مجذوری از ترکیب ژنتیک الگوریتم عادی با الگوریتم جست‌وجوی مجذوری بررسی شده است. از این الگوریتم جدید برای بهینه سازی چهار تابع امتحانی و بهینه سازی اقتصادی یک واحد فرآیندی از طریق اتصال به یک شبیه‌ساز استفاده شد. از آن‌جا که محاسبه میزان تابع هدف در مسایل بهینه‌سازی که در آن یک شبیه‌ساز با یک بهینه‌ساز خارجی در ارتباط است شامل حجم محاسبات خیلی بالایی می‌باشد؛ لذا انتخاب و به کار بردن یک الگوریتم جست‌وجوی کارآمد به منظور کاهش تعداد دفعات محاسبه تابع هدف از اهمیت بسیار بالایی برخوردار است و این در حالی است که ژنتیک الگوریتم با جست‌وجوی مجذوری تمامی این شرایط را به علت ترکیب کردن مزایای ژنتیک الگوریتم و الگوریتم جست‌وجوی مجذوری دارا می‌باشد. در ضمن این روش می‌تواند بهینه مطلق با احتمال ناپیوستگی بالا را به خوبی بهینه سازی غیر محدب و با همگرایی بهتر از ژنتیک الگوریتم عادی تعیین کند. کارایی ژنتیک الگوریتم با جست‌وجوی مجذوری و ژنتیک الگوریتم عادی برای حل مسایل بهینه سازی مربوط به چهار تابع امتحانی نشان داده شده در جدول ۱ مبین این موضوع می‌باشد که همگرایی ژنتیک الگوریتم با جست‌وجوی مجذوری در تمامی موارد بهتر و حداقل قابل مقایسه با ژنتیک الگوریتم عادی می‌باشد و در مورد بهینه‌سازی اقتصادی واحد فرآیندی بازیابی گاز طبیعی مایع نیز ژنتیک الگوریتم با جست‌وجوی مجذوری کارایی بیش‌تری نسبت به ژنتیک الگوریتم عادی از خود نشان می دهد. قسمت جست‌وجوی مجذوری در ژنتیک الگوریتم با جست‌وجوی مجذوری در بسیاری از مواقع به عنوان یک اپراتور ژنتیک نظیر جهش و قطع عمل کرده که سبب کاهش تعداد نسل‌های مورد نیاز برای تعیین بهینه با سطح دقت مشخص می‌شود که در همان زمان مشخص با توجه به هزینه موجود برای محاسبات ریاضی در هر نسل توسط ژنتیک الگوریتم عادی قابل محاسبه نیست.

منابع

1. Leboreiro, J., & Acevedo, J. (2004). Processes synthesis and design of distillation sequences using modular simulators: a genetic algorithm framework. Computers and Chemical Engineering, 28, 1223–1236.
2. Tayal, M. C., Fu, Y., & Diwekar, U. M. (1999). Optimal design of heat exchangers: a genetic algorithm framework. Industrial and Engineering Chemistry Research, 38, 456–467.
3. Wang, K., L¨ohl, T., Stobbe, M., & Engell, S. (2000). A genetic algorithm for online-scheduling of a multiproduct polymer batch plant. Computers and Chemical Engineering, 24, 393–400.
4. Gross, B., & Roosen, P. (1998). Total process optimization in chemical engineering with evolutionary algorithms. Computers and Chemical Engineering, 22, s229–s236.
5. Li, P., L¨owe, K., Arellano-Garcia, H., & Wozny, G. (2000). Integration of simulated annealing to a simulation tool for dynamic optimization of chemical processes. Chemical Engineering and Processing, 39, 357– 363.
6. Rao, S. S. (1996). Engineering optimization: theory and practice. New York: John Wiley & Sons.

7. Wang, K., Salhi, A., & Fraga, E. S. (2004). Process design optimization using embedded hybrid visualisation and data analysis techniques within a genetic algorithm optimisation framework. Chemical Engineering and Processing, 43(5), 657–669.
8. Holland, J. H. (1975). Adaptation in natural and artificial systems. Ann Arbor, MI: University of Michigan Press.
9. Kasprzyk, G. P., & Jasku, M. (2004). Application of the hybrid geneticsimplex algorithm for deconvolution of electrochemical responses in SDLSV method. Journal of Electroanalytical Chemistry, 567(1), 39– 66.
10. Hanagandi, V., & Nikolaou, M. (1998). A hybrid approach to global optimization using a clustering algorithm in a genetic search framework. Computers and Chemical Engineering, 22, 1913–1925.
11. Grefensetette, J. (1994). Lamarckian learning in multi-agent environment. In Proceedings of the Fourth International Conference on Genetic Algorithms. San Mateo, CA: Morgan Kaufmann Publishers.
12. Goldberg, D. E. (1989). Genetic algorithms in search, optimization and machine learning. Reading, MA: Addison-Wesley.
13. Gen, M., & Cheng, R. (1997). Genetic algorithms and engineering design. New York: John Wiley & Sons.
14. Edgar, T. F., Himmelblau, D. M., & Lasdon, L. S. (2001). Optimization ofchemical processes. New York: McGraw-Hill.
15. De Jong, K. A. (1975). An analysis of the behavior of a class of genetic adaptive systems. Ph.D. Dissertation. University of Michigan, Ann Arbor, MI.
16. Ahuja, R. K., Orlin, J. B., & Tiwari, A. (2000). A greedy genetic algorithm for the quadratic assignment problem. Computer and Operational Research, 27, 917–934.