

طراحی یک فیلتر پایین گذر به کمک الگوریتم ژنتیک و هوش جمعی و مقایسه این دو روش بهینه سازی نوین .

گلنار خمامی _ عباس گلمکانی

موسسه آموزش عالی سجاد مشهد

Golnar.khomami@gmail.com

Abbas_golmakani@yahoo.com

چکیده - در این مقاله الگوریتم ژنتیک و هوش جمعی را بعنوان روشهای نوین بهینه سازی معرفی و از آنها در طراحی یک فیلتر پایین گذر استفاده کرده ایم. در انتها ضمن مقایسه این دو روش با یکدیگر، نحوه عملکرد عملگرهای الگوریتم ژنتیک در نتیجه طراحی بررسی شده است. برنامه های الگوریتم ژنتیک و *pso* با نرم افزار مطلب و شبیه سازی مدار با نرم افزار *hspice* صورت گرفته است.

کلید واژه - الگوریتم ژنتیک، فیلتر پایین گذر، کروموزوم، هوش جمعی، *particles*

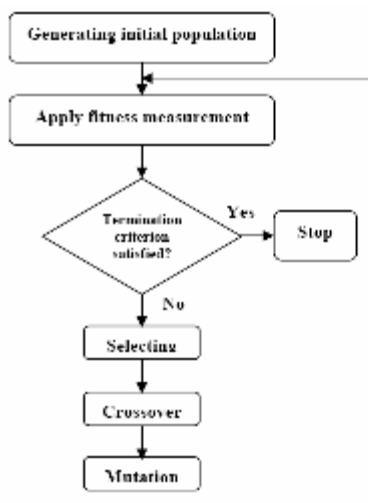
و بخش جدید اتفاقی برای رسیدن به یک جواب مناسب به وجود می آید.

طراحی مدارات الکترونیکی از جمله زمینه هایی است که در آن الگوریتم ژنتیک به خوبی پاسخ داده است.

1- مقدمه

1-1- الگوریتم ژنتیک

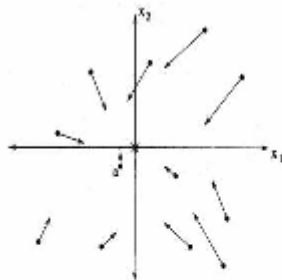
الگوریتم ژنتیک یک روش بهینه سازی عددی است که بر اساس اصول داروین و با الهام از انتخاب طبیعی و ژنتیک طبیعی کار می کند. رقابت میان موجودات زنده برای تصاحب منابع مشترک محدود، باعث می شود که کارآمدترین افراد در این رقابت پیروز شده و در انتقال ژن ها به نسل های بعدی، نقش موثرتری داشته باشند. نتیجه این امر، برازندگی بیشتر افراد هر نسل و نیز سازگاری بیشتر افراد در نسل های بعدی با شرایط حاکم بر محیط اطراف خواهد بود. الگوریتم ژنتیک یکی از مجموعه روش های تکاملی است. این روش مناسب ترین رشته ها را از میان اطلاعات تصادفی سازمان دهی شده با روش جستجوی انسانی انتخاب می کند. در هر نسل یک گروه جدید رشته ها با استفاده از بهترین قسمت های دنباله های قبلی



شکل 1- چرخه عملکرد الگوریتم ژنتیک

1-2- هوش جمعی¹

در فرمولهای فوق، $V[i]$ سرعت افراد (particle)، $present[i]$ مقدار کنونی افراد، $Rand()$ ، عددی رندوم در بازه (0 و 1) و $c1$ ، $c2$ فاکتورهای یادگیری هستند. معمولاً $c1=c2=2$ در نظر گرفته می شود.



شکل 2- نحوه تغییر مکان افراد با توجه به جهت حرکت نزدیکترین فرد به هدف (a)

1-3- طراحی مدارات الکترونیکی به کمک

روشهای نوین [2]

اغلب در هنگام طراحی مدارهای الکترونیکی، به خاطر وجود محدودیت هایی در روال طراحی، مجبوریم از روابط و مدل های تقریبی برای المانهای مختلف استفاده کنیم و این سبب می شود که نتایج پس از شبیه سازی، با مقادیر مورد انتظار متفاوت باشد. این تفاوت ممکن است در مدارهای ساده و کوچک قابل چشمپوشی باشد، اما در مدارهای پیچیده، بسیار بزرگ و چند طبقه ایجاد مشکل می کند. تا جایی که طراح ممکن است پس از طراحی به خواسته های مورد نظر نرسد. در این حالت دست یافتن به ابزاری که به کمک آن بدون نیاز به اطلاعات خیلی خاص و پیچیده بتوان به طراحی مناسبی برای یک مدار رسید ایده آل است. روشهای نوین الهام گرفته شده از طبیعت به دو صورت می توانند عمل کنند.

در روش اول ابتدا طراح خواسته های مسئله را به صورت تابعی از متغیرهای مدار در می آورد سپس تابع مورد نظر را با یکی از این روشها بهینه می کنند. این روش در

الگوریتم PSO یک الگوریتم جستجوی اجتماعی است که از روی رفتار اجتماعی دسته های پرندگان مدل شده است. در ابتدا این الگوریتم به منظور کشف الگوهای حاکم بر پرواز همزمان پرندگان و تغییر ناگهانی مسیر آنها بکار گرفته شد. در این روش افراد² برای رسیدن به یک هدف نهایی همکاری می کنند. این روش مؤثرتر از زمانی است که particle ها جداگانه عمل کنند. Swarm را می توان به صورت مجموعه ای سازمان یافته از موجوداتی تعریف کرد که با یکدیگر همکاری می کنند. در PSO، particle ها در فضای جستجو جاری می شوند. تغییر مکان particle ها در فضای جستجو تحت تأثیر تجربه و دانش خودشان و همسایگانشان است. بنابراین موقعیت دیگر particle های Swarm روی چگونگی جستجوی یک particle اثر می گذارد.

نتیجه ی مدل سازی این رفتار اجتماعی فرایند جستجویی است که particle ها به سمت نواحی موفق میل می کنند. در این روش، رفتار جمعی فقط وابسته به رفتار فردی افراد اجتماع نیست بلکه به چگونگی تعامل میان افراد نیز وابسته است. تعامل بین افراد، تجربه ی افراد درباره ی محیط را افزایش می دهد و موجب پیشرفت اجتماع می شود. ساختار اجتماعی Swarm بین افراد مجموعه کانالهای ارتباطی ایجاد می کند که طی آن افراد می توانند به تبادل تجربه های شخصی بپردازند، مدل سازی محاسباتی Swarm ها کاربردهای موفق و بسیاری را در پی داشته است.

1-2-1- الگوریتم هوش جمعی [7]

اساس کار PSO بر این اصل استوار است که در هر لحظه هر فرد، مکان خود را در فضای جستجو با توجه به بهترین مکانی که تاکنون در آن قرار گرفته است (Pbest) و بهترین مکانی که در کل همسایگی اش وجود دارد (Gbest)، تنظیم می کند. این تغییر مکان پس از بدست آمدن این دو مکان توسط دو فرمول زیر محاسبه می شود.

$$V[i+1] = V[i] + c1 * rand(i) * (Pbest[i] - \dots present[i]) + c2 * rand(i) * Gbest[i] - present[i]$$

$$Present[i+1] = present[i] + v[i]$$

¹- Particle swarm optimization

²-particles

اجرا می شود. 3- نرم افزار شبیه ساز.

3-1- طراحی مدار با الگوریتم ژنتیک

برای نوشتن برنامه الگوریتم ژنتیک باید کروموزومها که معرف اندازه المانها در هر مرحله هستند را تشکیل دهیم. در این مساله یک کروموزوم از 9 ژن تشکیل شده که به علت شباهت المانها تعداد بیتهایی که برای هر کدام از ژنها در نظر گرفته ایم یکسان و برابر 10 بیت است. اندازه جمعیت اولیه و تعداد نسلها را در هر مرحله متغیر گرفته ایم. برای افزایش سرعت همگرایی، برای هر کدام از متغیرهایمان حدی را تعیین می کنیم. محدوده هایی که برای هر کدام از المانها در نظر گرفته شده برای خازن ها از 0 تا $500nF$ و برای سلف ها از 0 تا $300mH$ است.

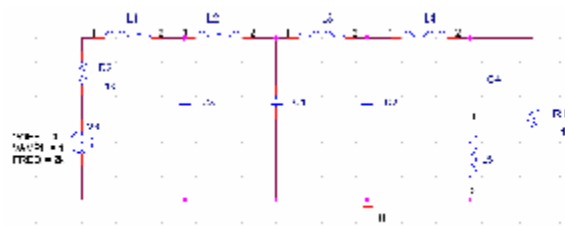
تابع برازندگی را در این مساله برابر با مجموع مربعات خطای حاصل از تفاوت اندازه تابع پله در فرکانسهای مشخص از اندازه پاسخ مدار شبیه سازی شده برای المانها در همان فرکانسها در نظر گرفتیم.

همانطور که در شکل 4 مشاهده می شود، پس از تعیین موارد فوق و تعیین توپولوژی مناسب، مدار را در یک فایل txt شبیه سازی می کنیم. این فایل به عنوان ورودی نرم افزار مطلب تعریف می شود. برنامه الگوریتم ژنتیک، مقادیری تصادفی را به عنوان مقادیر اولیه به متغیرهای ورودی خود که سلف ها و خازن ها هستند اختصاص می دهد و آنها را در فایل متنی ورودی جایگزین نموده سپس Hspice را اجرا می کند؛ پس از اجرا، نرم افزار Hspice مدار را به صورت بسیار دقیق تحلیل کرده و نتایج خروجی را در فایلی با پسوند LIS می ریزد، Matlab فایل مزبور را باز نموده مقادیر مطلوب خود را که برای بررسی برازندگی به آنها نیاز دارد استخراج می کند و پس از تحلیل برازندگی کروموزوم ها و اعمال عملگرهای جهش، برش و... نسل جدید را ایجاد می کند. مقادیر ایجاد شده در نسل جدید بعنوان ورودی های مدار به netlist برگردانده شده، Hspice مجدداً مدار مورد نظر که اکنون المان هایش مقادیر جدیدی گرفته اند را simulate می کند و مراحل قبلی به تناوب تا پایان نسلهای تعیین شده و یا برآورده شدن معیار همگرایی تکرار می شود.

حالتی کاربرد دارد که دستیابی به تابعی بر حسب پارامترهای مدار که باید تعیین شوند امکان پذیر و ساده باشد. امکان بدست آوردن چنین تابعی معمولاً تنها برای مدارهای ساده امکان پذیر است. اشکال دیگری که این روش دارد این است که اغلب رابطه هایی که مورد استفاده قرار می گیرند خود دارای تقریب های فراوان بوده و مناسب این گونه طراحی ها نمی باشد. مزیت این روش سرعت به جواب رسیدن الگوریتم است و دیگر اینکه اگر بتوانیم به تابعی بر حسب متغیرها دست یابیم آنگاه نحوه عملکرد هر پارامتر در مدار و میزان اهمیت هر المان در خروجی مدار مشخص می شود و در این حالت می توانیم (برای مثال در الگوریتم ژنتیک) با قرار دادن تعداد بیت های بیشتر برای متغیر موثرتر سرعت همگرایی را بالا ببریم و به نتایج بهتری برسیم. روش دیگر این است که الگوریتم فقط مقادیر قطعات را تعیین می کند و توپولوژی مدار و تعداد المان ها توسط طراح تعیین می شود سپس مداری با اندازه های بهینه برای المان هایش تولید می شود. مزیت این روش دقت و سادگی آن بوده و اشکال آن زمانگیر بودن آن است.

2- طرح مساله

در این مقاله قصد داریم فیلتری طراحی کنیم که پاسخ فرکانسی آن تا حد امکان به پاسخ فرکانسی یک تابع پله نزدیک باشد. فیلتری که فرکانس قطع بالای آن $1kHz$ باشد. برای چنین هدفی از مداری استفاده کردیم با 9 المان که طراحی آن بصورت دستی مشکل باشد.



شکل 3- توپولوژی انتخاب شده برای طراحی فیلتر ایده آل

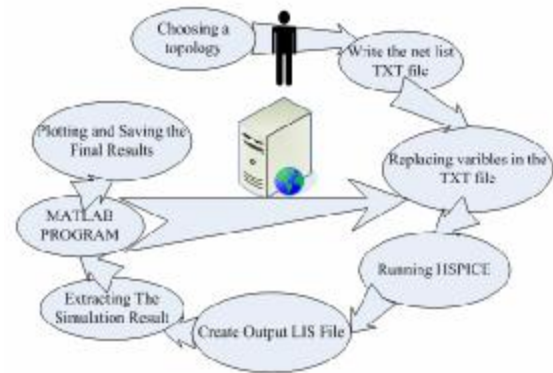
3- روند طراحی

برای بهینه سازی و طراحی یک مدار با روشهای نوین موارد زیر لازم است:

1- تعیین نوع الگوریتم. 2- نرم افزاری که الگوریتم در آن

همانطور که مشاهده می شود در نتایج بدست آمده فیلتر حاصل خواسته های ما را برآورده نمی کند. همچنین تغییرات تابع برازندگی هنوز به حالت پایدار نرسیده است. این مطلب می تواند مربوط به کم بودن تعداد نسل ها و جمعیت اولیه باشد که سبب شده همگرایی مناسب حاصل نشود. برای بررسی این مطلب الگوریتم ژنتیک را یکبار دیگر با مقادیر جدید اجرا می کنیم.

در این مرحله تعداد نسل ها را 100 نسل و جمعیت اولیه را 15 در نظر می گیریم و با ثابت نگه داشتن سایر مقادیر بار دیگر الگوریتم را اجرا می کنیم.



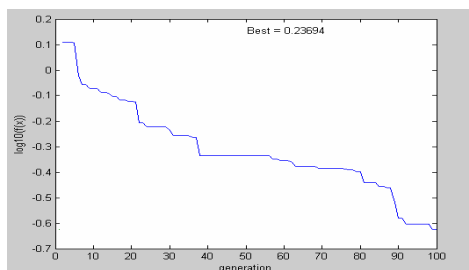
شکل 4- چرخه کلی طراحی مدار با الگوریتم ژنتیک [1]

3-2- نتایج حاصل از طراحی با الگوریتم

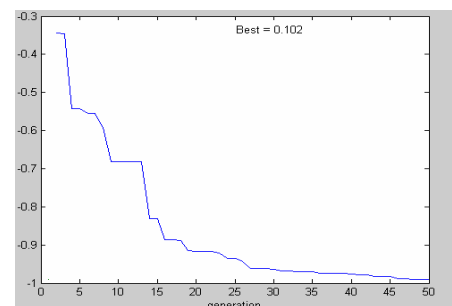
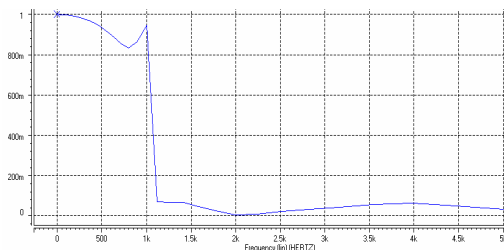
ژنتیک در هر مرحله

مقادیری که برای هر یک از پارامترهای الگوریتم در مرحله اول انتخاب کردیم به این ترتیب است:

تعداد کروموزوم ها در هر نسل 10 کروموزوم و تعداد نسلهایی که الگوریتم اجرا می شود 50 نسل



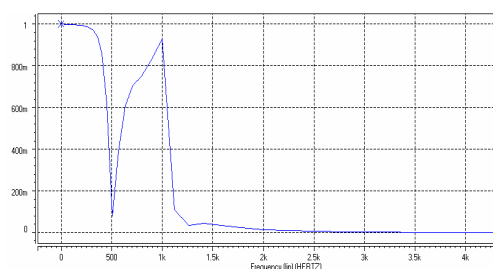
شکل 7- نمودار تغییرات تابع برازندگی به ازاء افزایش نسل در مرحله 2



شکل 5- نمودار تغییرات تابع برازندگی به ازاء افزایش نسل در مرحله 1

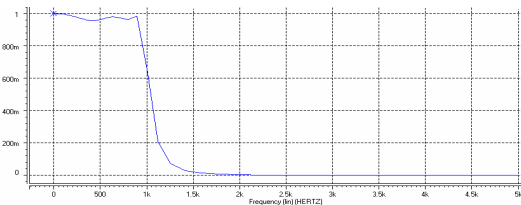
شکل 8- نمودار ولتاژ خروجی فیلتر طراحی شده در مرحله 2

در این مرحله می بینیم که نتیجه بصورت واضحی بهبود یافته است. اما نه فیلتر هنوز کاملاً ایده آل شده و نه منحنی کاهش تابع برازندگی به حالت تعادل رسیده است.



شکل 6- نمودار ولتاژ خروجی فیلتر طراحی شده در مرحله 1

در مرحله بعد با 200 نسل میزان نخبه گرایی را از 0.9 به 0.8 کاهش داده و احتمال آنرا افزایش دادیم.



شکل 12- نمودار ولتاژ خروجی فیلتر طراحی شده در مرحله 4

مشاهده می شود که در این مرحله با افزایش احتمال عملگر جهش جستجو گسترده تر شده و در 200 نسل همواره تابع برازندگی در حال کم شدن است و در نهایت بعد از گذشت نسلهای تعیین شده به مقادیری از المانها می رسیم که تقریباً خواسته های ما را برآورده می کند.

C1	C2	C3	C4	L1	L2	L3	L4	L5
2.93 pf	328.93 pf	394.42 pf	201.8 pf	17 mH	156mH	244.86 mH	256mH	879uH

جدول 1- مقادیر المانهای فیلتر در مرحله آخر GA

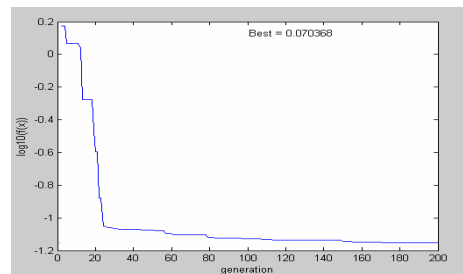
3-3- طراحی مدار به روش PSO

برای طراحی یک مدار به روش هوش جمعی دقیقاً مشابه روش الگوریتم ژنتیک عمل می کنیم با این تفاوت که در این روش دیگر نیازی به تشکیل کروموزومها، اعمال عملگرهایی چون جهش و برش، نخبه گرایی و سایر مراحل حل یک مساله با الگوریتم ژنتیک نداریم.

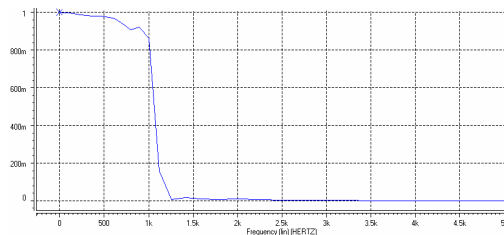
از مزایای این روش سادگی برنامه و عدم نیاز به تعیین و اعمال عملگرهاست. در این روش کافی است تعداد افراد، متغیرهایی که باید محاسبه شوند، تعداد تکرار و ماکسیمم تغییر مکان هر particle را مشخص کنیم. در این روش قادر به محدود کردن محدوده جستجو نیستیم. تنها محدودیتی که در این روش می توانیم اعمال کنیم، محدود کردن مقادیر رندومی است که در ابتدای برنامه برای هر کدام از particle ها تعیین می شود. در این بخش از مقاله تابع ارزیابی، نرم افزارهای بکار رفته و محدودیتی که برای مقادیر اولیه در نظر گرفتیم مشابه الگوریتم ژنتیک است.

3-4- نتایج حاصل از طراحی مدار با PSO

نتایجی که در این روش با استفاده از مقادیر معین شده زیر بدست آمده در شکلهای 13 و 14 مشاهده می شود. تعداد افراد (particles) را 20 فرد و تعداد تکرار را



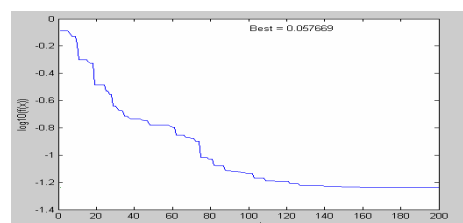
شکل 9- نمودار تغییرات تابع برازندگی به ازاء افزایش نسل در مرحله 3



شکل 10- نمودار ولتاژ خروجی فیلتر طراحی شده در مرحله 3

همانطور که مشاهده می شود در این مرحله خروجی فیلتر خیلی بهتر شده و تابع برازندگی تقریباً همگرا شده و به حالت تعادل رسیده است. تغییر نتیجه و بهبود آنرا می توان به افزایش احتمال نخبه گرایی نسبت دهیم. همانطور که دیدیم نخبه گرایی بطور واضحی سرعت همگرایی را افزایش می دهد. در عین حال از آنجایی که الگوریتم اینبار به سرعت همگرا شده و در عین حال علی رغم اینکه تعداد نسلها 200 نسل است هنوز به المانهای ایده آل نرسیده ایم، می توان نتیجه گرفت که احتمالاً به علت بالا رفتن احتمال نخبه گرایی الگوریتم در نقطه بهینه محلی به دام افتاده است.

در مرحله آخر تاثیر افزایش احتمال عملگر جهش را بر روی الگوریتم ژنتیک بررسی می کنیم. در این مرحله اندازه عملگر جهش از 0.7 به 0.6 کاهش می یابد و احتمال آن افزایش می یابد



شکل 11- نمودار تغییرات تابع برازندگی به ازاء افزایش نسل در مرحله 4

دو روش می توانند به عنوان ابزاری برای تعیین مقدار المان ها بکار گرفته شوند و به نتایج قابل قبولی دست یابند.

دسترسی به این نتایج بدون نیاز به اطلاعات مداری و هرآنچه برای طراحی یک مدار (و در اینجا بطور خاص در مورد طراحی یک فیلتر) بصورت کلاسیک نیاز است حاصل می شود.

همچنین در این پروژه اثر افزایش احتمال عملکرد عملگرهای الگوریتم ژنتیک در نتیجه بهینه سازی و طراحی یک مدار بررسی شد. و در نهایت با به کار بردن روش PSO مشاهده کردیم که این روش علیرغم سادگی الگوریتمش می تواند همپای الگوریتم ژنتیک و با سرعتی بالاتر به حل چنین مسائلی بپردازد.

5- سپاسگزاری

با سپاس فراوان از جناب آقای مهندس امیرحسین قدس که با راهنماییهای مفیدشان ما را در این پروژه یاری کردند.

6- مراجع

[1] امین ابراهیمی خانکوک و حمیدرضا رضایی دهرسخ , طراحی یک میکسر Double Balanced به کمک الگوریتم ژنتیک و دانشکده مهندسی دانشگاه فردوسی مشهد و 1384

[2] john R. koza , Forrest H Bennett David Andre , Martin A. Keane, Genetic Programming Darwinian Invention and problem solving ,third edition,

[3] Matlab , version 6.5 the math works, 2002

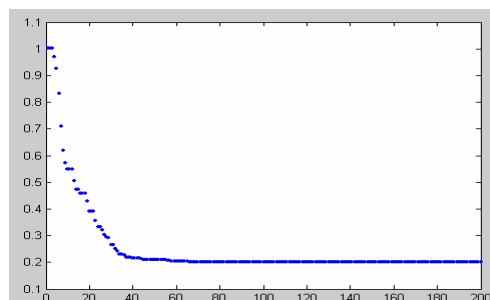
[4] J . singh, pso Matlab Toolbox , PSOTOOLBOX open source 2003

[5] A. chipperfield , genetic algorithm users guide for use with MATLAB , version 1.2

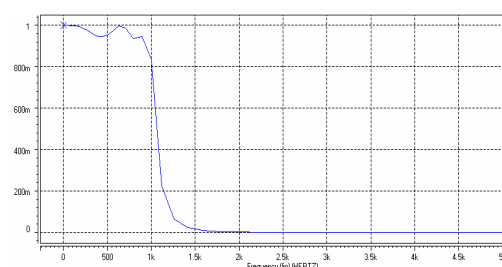
[6] Star-Hspice manual, Release 2001.2,

[7] <http://www.petroleumtimes.com/article>

200 تکرار، همچنین ماکسیمم تغییرات هر فرد را (V_{max}) 5 در نظر گرفتیم. همانطور که مشاهده می شود، تابع برازندگی به سرعت در چند تکرار اول نسبت به مقدار اولیه اش کم می شود و پس از آن سرعت تغییراتش کاهش می یابد. سرعت همگرایی این الگوریتم به نسبت الگوریتم ژنتیک بالاتر است.



شکل 13- نمودار تغییرات تابع برازندگی به ازاء تعداد تکرار متوالی



شکل 14- نمودار ولتاژ خروجی فیلتر طراحی شده به روش pso

C1	C2	C3	C4	L1	L2	L3	L4	L5
8.42 nf	248 nf	314 nf	240 nf	25.7 mH	2.10 mH	3.11 mH	247 mH	278 mH

جدول 2- مقادیر المانهای فیلتر در مرحله آخر PSO

4- نتیجه گیری

در این گزارش الگوریتم های ژنتیک و PSO با نرم افزارهای شبیه ساز ترکیب شده و به صورت ابزاری دقیق برای بهینه سازی یک فیلتر پایین گذر مورد استفاده قرار گرفت. بنابر نتایج حاصله چنانچه توپولوژی مدار از قبل معلوم باشد این