

حل مسئله رنگ آمیزی گراف به وسیله PSO

کارو لوکس^(۲)

lucas@ipm.ir

امین فاضل دهکردی^(۱)

afazel@mehr.sharif.edu

^(۱) دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف

^(۲) گروه مهندسی برق و کامپیوتر، دانشکده فنی دانشگاه تهران

چکیده: در این مقاله، روشی برای حل مسئله رنگ آمیزی گراف، با استفاده از الگوریتم PSO ارائه شده است. PSO یک روش بهینه سازی تکاملی است که از رفتار اجتماعی گروه پرندگان یا دسته ماهی ها الهام گرفته شده است. الگوریتم PSO معمولاً برای بهینه سازی توابع غیرخطی با متغیرهای پیوسته به کار می رود، در حالی که در مسئله رنگ آمیزی گراف با یک فضای جستجوی گسسته سر و کار داریم. بنابراین با اعمال تغییراتی در الگوریتم PSO پایه و اضافه کردن توابعی جهت بررسی و نسبت دهی رنگ ها به رأس های گراف مورد نظر، از آن برای حل مسئله رنگ آمیزی گراف استفاده شده است. کارایی الگوریتم پیشنهادی به کمک یک مجموعه استاندارد از مثال های مسئله رنگ آمیزی گراف بررسی شده است.

کلمات کلیدی: الگوریتم PSO، مسئله رنگ آمیزی گراف، الگوریتم های تکاملی.

۱. مقدمه

تکنیک های محاسبات تکاملی، برخلاف الگوریتم های جستجوی متداول، روی یک مجموعه از جواب ها در فضای جستجو عمل می کنند و با استفاده از همکاری و رقابتی که بین جواب ها ایجاد می کنند، می توانند خیلی سریع جواب بهینه را برای مسائل بهینه سازی پیچیده پیدا کنند. این تکنیک ها به طور عمده از فرایند تکامل در طبیعت الهام گرفته شده اند که چهار مورد مشهور آن، الگوریتم ژنتیک، برنامه نویسی تکاملی، استراتژی های تکاملی و برنامه نویسی ژنتیکی می باشند. سوای این تکنیک های محاسبات تکاملی الهام گرفته از فرایند تکامل در طبیعت، یک سری تکنیک های محاسباتی جدید ابداع شده اند که رفتار اجتماعی را

شبیه‌سازی کرده‌اند نظیر اجتماع مورچگان^۱ و PSO^۲. عمده دلیلی که می‌توان برای رفتار اجتماعی موجودات زنده آورد، بهینگی آن می‌باشد. به این ترتیب منطقی به نظر می‌رسد که برای حل مسائل بهینه‌سازی مهندسی، این رفتارهای اجتماعی را شبیه‌سازی کنیم.

الگوریتم PSO، رفتار اجتماعی گروه پرندگان یا دسته ماهی‌ها را شبیه‌سازی می‌کند که در سال ۱۹۹۵ میلادی توسط Kennedy و Eberhart [۱] ارائه شد. الگوریتم PSO تنها یک الگوریتم برای بهینه‌سازی نمی‌باشد، بلکه یک ابزار برای نمایش شناخت اجتماعی انسان‌ها و عامل‌های هوشمند^۳ بر اساس روانشناسی اجتماعی می‌باشد. بعضی از دانشمندان معتقدند که دانش به وسیله اثر متقابل رفتارهای اجتماعی بهینه می‌شود و تفکر نه تنها یک عمل خصوصی بلکه اجتماعی نیز می‌باشد. PSO بعنوان الگوریتم بهینه‌سازی، یک جستجوی مبتنی بر جمعیت را فراهم می‌کند که در آن هر فرد^۴ با گذشت زمان موقعیت خود را تغییر می‌دهد. در یک سیستم PSO، افراد در یک فضای جستجوی چندبعدی پرواز (شنا) می‌کنند. در طول این پرواز، هر فرد موقعیت خود را با توجه به تجربه‌های خود و تجربه‌های همسایگان خود تنظیم می‌کند و در واقع از بهترین موقعیت خود و اطرافیانش استفاده می‌کند. بنابراین همانند الگوریتم‌های ژنتیک، PSO متدهای جستجوی محلی و جستجوی سراسری را برای رسیدن به جواب بهینه ترکیب می‌کند که در واقع کوششی است برای به وجود آوردن حالت تعادل بین اکتشاف^۵ و بهره‌برداری^۶.

الگوریتم PSO خیلی ساده و از دیدگاه محاسباتی (سرعت و حافظه) خیلی کم هزینه می‌باشد و کاربرد عمده آن در تعیین اوزان شبکه عصبی و سیستم‌های کنترلی و هر جایی که بتوان الگوریتم ژنتیک را به کار برد، می‌باشد. در سال‌های اخیر مقالات زیادی [۲]، [۳] استفاده از PSO را به جای روش یادگیری پس‌انتشار^۷ در شبکه‌های عصبی مورد بررسی قرار داده‌اند و در اکثر موارد PSO به عنوان روشی سریع و قابل قبول گزارش شده است.

۲. رنگ آمیزی گراف

یک رنگ آمیزی مناسب از گراف $G = (V, E)$ عبارت است از تابعی از رأس‌های گراف به مجموعه‌ای از رنگ‌ها به طوری که هر دو رأس مجاور در گراف دارای رنگ‌های متفاوت باشند. اگر $|C| = k$ باشد، می‌گوییم که گراف G ، k -رنگ پذیر است. حداقل تعداد رنگ‌های ممکن به طوری که یک رنگ آمیزی مناسب از گراف G حاصل شود را عدد کروماتیک^۸ گراف G گویند.

مسئله پیدا کردن عدد کروماتیک و یک رنگ آمیزی مناسب برای یک گراف به علت کاربردهای فراوان آن در زمینه‌هایی مانند ارتباطات و زمان‌بندی بسیار مورد توجه بوده است. این مسئله نیز مانند بسیاری از مسائل نظریه گراف، NP-complete می‌باشد. هنگامی که فرض‌هایی برای مسئله رنگ آمیزی گراف در نظر گرفته شود انواع مختلفی از این مسئله به وجود می‌آید [۴]. در این زمینه

^۱ Ant Colony

^۲ Particle Swarm Optimization

^۳ Intelligent Agent

^۴ Particle

^۵ Exploration

^۶ Exploitation

^۷ Back Propagation

^۸ Chromatic Number

تحقیقات جامعی صورت پذیرفته است که نتیجه آن‌ها تعداد بسیار وسیعی از الگوریتم‌های دقیق و تقریبی، هیوریستیک^۱ و متاهیوریستیک^۲ می‌باشد [۵]، [۶].

۳. الگوریتم PSO

الگوریتم PSO در ابتدا توسط یک گروه^۳ از افراد که به طور تصادفی تولید شده‌اند مقداردهی می‌شود و جستجو برای پیدا کردن جواب بهینه آغاز می‌گردد. هر فرد در حافظه خود بهترین مسیری را که تا کنون بر اساس تابع برازش^۴ طی کرده است را حفظ می‌کند و همچنین مقدار تابع برازش را نیز نگه می‌دارد. این مقدار pbest نام دارد. مقدار دیگری که در عمل یک فرد از این جامعه مؤثر است بهترین مقدار تابع برازشی است که به وسیله یکی از افراد این اجتماع حاصل شده است. این مقدار نیز gbest نام دارد. از نظر مفهومی، pbest برای هر فرد در واقع حافظه اتوبیولوژیکی آن فرد محسوب می‌شود و تغییر موقعیت افراد بر اساس pbest در واقع پاسخ به احساس غربتی است که افراد هنگام دوری از محلی که در آن بیشتر ارضاء می‌شوند، دارند. gbest همان دانش عمومی جمعیت است و وقتی که افراد موقعیت خود را بر اساس gbest تغییر می‌دهند در واقع تلاش می‌کنند که سطح دانش خود را به سطح دانش جمعیت برسانند.

با توجه به مقادیر pbest و gbest، هر فرد از روابط زیر برای تعیین موقعیت بعدی خود استفاده می‌کند:

$$v_{t+1} = w_t \cdot v_t + c_1 \cdot \text{rand}() (pbest - x_t) + c_2 \cdot \text{rand}() (gbest - x_t) \quad (1)$$

$$x_{t+1} = x_t + v_{t+1} \quad (2)$$

در روابط بالا c_1 و c_2 پارامترهای یادگیری هستند. $\text{rand}()$ تابعی برای تولید اعداد تصادفی در محدوده صفر و یک است. x_t موقعیت فعلی و v_t سرعت حرکت افراد می‌باشد و معمولاً توسط یک مقدار ماکزیمم (V_{Max}) محدود می‌شود یعنی اگر v_{t+1} بزرگتر از V_{Max} شود، آن را برابر با V_{Max} قرار می‌دهیم. در واقع با این کار توانایی الگوریتم PSO را در جستجو به صورت سراسری محدود می‌کنیم. w_t یک پارامتر کنترلی است که تأثیر سرعت فعلی را بر سرعت بعدی کنترل می‌کند و یک حالت تعادل بین توانایی الگوریتم در جستجو به صورت محلی و جستجو به صورت سراسری ایجاد می‌نماید و در نتیجه به طور میانگین در زمان کمتری به جواب می‌رسیم. چون w_t به طور مستقیم و V_{Max} به طور غیر مستقیم، تعادل بین هر دو صورت جستجو را ایجاد می‌نمایند، بهتر است که از V_{Max} صرف‌نظر کنیم و ایجاد این تعادل را به عهده w_t بسپاریم. منطقی به نظر می‌رسد که w_t به صورت تدریجی کاهش یابد تا در ابتدا جستجو به صورت سراسری صورت بگیرد و به تدریج به سمت جستجوی محلی پیش برویم [۷].

بر اساس این که gbest را برای هر فرد در یک شعاع همسایگی و یا در کل جمعیت در نظر بگیریم، الگوریتم PSO به ترتیب دارای دو نسخه محلی و سراسری خواهد بود. نسخه سراسری، که gbest را برای هر فرد در کل جمعیت در نظر می‌گیرد، سریعتر است ولی

¹ Heuristic

² Metaheuristic

³ Swarm

⁴ Fitness

احتمال گیر افتادن در مینیمم محلی برای بعضی از مسائل وجود دارد. نسخه محلی، که در آن gbest برای هر فرد در یک شعاع همسایگی تعریف می شود، کمی کندتر می باشد ولی در عوض به آسانی در دام مینیمم محلی نمی افتد. بنابراین خوب است که در ابتدای جستجو از نسخه سراسری بهره ببریم تا سریعتر به جواب برسیم و آنگاه از نسخه محلی برای پالایش جواب استفاده کنیم. ساختار کلی الگوریتم PSO مشابه با الگوریتم ژنتیک است. به این صورت که هر دو جمعیت اولیه را به طور تصادفی مقداردهی می کنند، هر دو از یک تابع برازش برای محاسبه میزان مطلوبیت جواب ها استفاده می کنند و در هر دو هیچ گونه تضمینی در مورد پیدا کردن جواب بهینه نیست. همچنین هر دو الگوریتم توانایی ایجاد تعادل بین جستجو به صورت محلی و سراسری را دارند. در PSO این کار توسط پارامترهای w_t و V_{Max} و در الگوریتم ژنتیک توسط عملگر جهش صورت می گیرد. ولی مکانیزم اشتراک اطلاعات در PSO کاملاً با الگوریتم ژنتیک متفاوت است، به این صورت که در الگوریتم ژنتیک کروموزم ها اطلاعات را با هم به اشتراک می گذارند در حالی که در PSO فقط gbest اطلاعات را به دیگر افراد جمعیت انتقال می دهد. از مزیت های PSO می توان به پیاده سازی راحت تر و تعداد پارامترهای کمتر اشاره کرد. از لحاظ محاسباتی هم PSO کم هزینه تر می باشد. الگوریتم PSO، اصولاً برای بهینه سازی مسائل خطی با متغیرهای پیوسته توسعه داده شده است ولی با اعمال تغییرات اندک و تعریف مجدد عملگرهای محاسباتی، می توان آن را در مواردی که متغیرها گسسته می باشند نیز به کار گرفت.

۴. استفاده از PSO برای حل مسئله رنگ آمیزی گراف

برای حل مسئله موقعیت هر فرد جمعیت در PSO را می توان به عنوان یک رشته از رنگ های موجود در نظر گرفت که به ترتیب از قبل تعیین شده ای به رأس های گراف نسبت داده می شوند، به عنوان مثال اگر موقعیت فردی در فضای چهار بعدی برای حل مسئله رنگ آمیزی گرافی با چهار رأس برابر با $\langle 1, 2, 3, 4 \rangle$ باشد بیان می کند که رأس اول و چهارم گراف دارای رنگ ۱ و رأس های دوم و سوم به ترتیب دارای رنگ های ۲ و ۳ می باشند. پس بردار موقعیت در واقع یک بردار N بعدی است که N تعداد رأس های گراف مورد نظر می باشد.

برای تعریف تابع برازش نیاز به تابعی است که بیان کند آیا نسبت دهی رنگ ها به رأس های گراف محدودیت های مسئله رنگ آمیزی گراف را ارضاء می کند یا نه. یعنی اینکه آیا در نسبت دهی رنگ ها هر دو رأس مجاور دارای رنگ های متفاوت می باشند یا خیر. این تابع مقدار conflict را بر می گرداند که بیان کننده تعداد رأس هایی است که دارای رنگ های یکسان می باشند. از آنجایی که موقعیت هر فرد برابر با رنگ هایی است که به یال ها نسبت داده می شوند و موقعیت هر فرد با توجه به روابط (۱) و (۲) می تواند منفی نیز بشوند نیاز به تابعی است که همیشه تناظری بین موقعیت و رنگ های نسبت داده شده ایجاد کند. این تابع تضمین می کند که رنگ های نسبت داده شده همیشه از عدد ۱ شروع شوند.

اگر موقعیت یک فرد از جمعیت $x = (n_1, n_2, \dots, n_N)$ باشد، تابع برازش برای x برابر است با:

$$f(x) = \begin{cases} \max_{i=1}^N (n_i) & \text{if conflict} = 0 \\ \text{conflict} \times p + \max_{i=1}^N (n_i) & \text{else} \end{cases} \quad (3)$$

که p برابر است با درجه اهمیت که ارضاء نشدن محدودیت مسأله می تواند داشته باشد. فضای جستجو در مسأله رنگ آمیزی گراف یک فضای گسسته می باشد بدین معنی که مؤلفه های هر فرد از جمعیت نمی تواند هر مقدار دلخواهی داشته باشد و مقادیر مجاز فقط محدود به اعداد طبیعی ۱ تا N می باشند بنابراین برای حل مسأله رنگ آمیزی گراف، با یک PSO گسسته^۱ [۸] سروکار داریم.

۵. نتایج حاصل از آزمایشات

برای تست الگوریتم شرح داده شده در بخش قبل، از گراف هایی با ۳، ۴، ۵ و ۱۰ رأس استفاده شده است. جمعیت در نظر گرفته شده برای PSO با افزایش تعداد رأس ها، از ۲۰ فرد تا ۳۵۰ فرد افزایش پیدا می کند که موقعیت اولیه آنها به طور تصادفی انتخاب می شود. شرط توقف الگوریتم رسیدن تابع برازش به عدد کروماتیک گراف یا یک ماکزیمم تعداد تکرار می باشد. الگوریتم ده بار با پارامترهای مذکور به ازای هر گراف اجرا شده است که تعداد موفقیت ها یا شکست ها در رسیدن به عدد کروماتیک گراف در جدول ۱ آمده است. همان طور که مشاهده می شود الگوریتم به جز یک شکست که در گراف با ۱۰ یال به دست آورده است در بقیه موارد به خوبی عمل کرده است. آزمایش های مختلف نشان داده است که مقدار جمعیت عامل مؤثری در جواب نهایی می باشد و با افزایش جمعیت، الگوریتم در رسیدن به جواب بهینه بهتر عمل می کند هر چند که با افزایش آن، سرعت اجرا پایین می آید. استفاده از W_t به صورت یک مقدار متغیر نزولی و همچنین محدود کردن V_{t+1} به یک مقدار V_{Max} عوامل مؤثر دیگری در اجرای الگوریتم می باشند.

جدول ۱: نتایج به دست آمده از ده بار اجرای الگوریتم بر روی گراف های با تعداد رأس های مختلف

تعداد رأس ها	تعداد یال ها	عدد کروماتیک	تعداد موفقیت ها (شکست ها)	
۱	۳	۲	۱۰	۱
۲	۳	۳	۱۰	۲
۳	۴	۲	۱۰	۳
۴	۴	۴	۱۰	۴
۵	۵	۲	۱۰	۵
۶	۱۰	۳	۱۰ (۹)	۶

^۱ discrete PSO

۶. جمع‌بندی و نتیجه‌گیری

در این مقاله پس از بیان مسأله رنگ‌آمیزی گراف، روشی برای حل آن با استفاده از الگوریتم PSO ارائه شده است. همانطور که بیان شد روش پیشنهادی در حل گراف با ابعاد کوچک به خوبی عمل کرده است و کار آینده استفاده از آن در حل مسائل با ابعاد بزرگتر می‌تواند باشد. با این حال می‌توان با اعمال تغییراتی در این روش برای سایر مسائل بهینه‌سازی گسسته‌ای که الگوریتم مناسبی برایشان یافت نشده، از آن استفاده کرد.

۷. منابع:

- [1] Kennedy, J. and Eberhart, R.C., "*Particle Swarm Optimization*", Proceedings of the 4th IEEE International Conference on Neural Networks, Vol. IV, Piscataway, NJ, IEEE Service Center, 1995, pp. 1942-1948.
- [2] Eberhart, R.C. and Shi, Y., "*Evolving Artificial Neural Networks*", Proceedings of International Conference on Neural Networks and Brain, Beijing, P.R. China, 1998, pp. PL5-PL13.
- [3] Gudise, V.G., and Venayagamoorthy, G.K., "*Comparison of Particle Swarm Optimization and Back Propagation as Training Algorithms for Neural Networks*", Proceedings of the IEEE Swarm Intelligence Symposium 2003 (SIS 2003), Indianapolis, Indiana, USA, 2003, pp. 110-117.
- [4] Jensen T.R., Toft B., "*Graph Coloring Problems*", Wiley Interscience Series in Discrete Mathematics and Optimization, 1995.
- [5] Kubale M., "*Introduction to Computational Complexity and Algorithmic Graph Coloring*", Gdanskie Towarzystwo Naukowe, 1998.
- [6] de Werra D., "*Heuristics for Graph Coloring*", Computing Suppl. 7, 1990, pp. 191-208.
- [7] Shi, Y., and Eberhart, R.C., "*Parameter selection in particle swarm optimization*", Evolutionary Programming VII: Proceedings of the 7th Annual Conference on Evolutionary Programming, New York, 1998, pp. 591-600.
- [8] Clerc, M., "*Discrete Particle Swarm Optimization, Illustrated by the Traveling Salesman Problem*", online: <http://www.mauriceclerc.net>, 2000.