

# رویکرد تکاملی توأم برای مساله رنگ آمیزی گراف

دانیال شه میرزادی  
استاد

کارولوکس  
استاد

محمد بیگلربگیان  
استادیار

دانشکده فنی، دانشگاه تهران

## چکیده

این مقاله، ابتدا سعی بر معرفی مساله رنگ آمیزی نقشه (GCP) و ادبیات آن نموده و سپس بر روشهای توسعه یافته برای حل این مساله مینماید. از آنجا که این مسئله، یکی از مسائل نمونه NP-hard میباشد، استفاده از روشهای هوشمند در حل آن، کارآیی بیشتری از خود نشان داده است. در این مقاله، برای حل مسئله مذکور، یک الگوریتم تکاملی دو جانبه ارائه شده است. این الگوریتم، شامل دو جمعیت میباشد که هر یک با اعمال اپراتورهای ژنتیک، رو به بهبود میروند و در نهایت، از کنار هم قرار دادن این جمعیتها، جواب کل مسئله، حاصل میگردد. حجم محاسبات مربوط به حل مسئله با روش توسعه یافته در این مقاله با حجم محاسبات مورد نیاز برای حل مساله توسط یک الگوریتم ژنتیک متناظر که در [1]، توسعه یافته، مقایسه گردیده است. نتایج بدست آمده، نشان میدهد که روش توسعه یافته، دارای مزیت چشمگیری نسبت به روشهای متعارف میباشد و میتوان از آن در حل مسائل بهینه سازی پیچیده، استفاده نمود.

## کلمات کلیدی

سیمبایاسیس، تکامل دو جانبه، بهینه سازی، مساله رنگ آمیزی نقشه،

## A Co-evolutionary Approach to Graph Coloring Problem

C. Lucas  
Professor

D. Shahmirzadi  
Professor

M. Biglarbegian  
Assistant Professor

Faculty of Engineering, University of Tehran

## Abstract

*This paper begins with a review of the Graph Colouring Problem and its literature. Next developed methods for solving the problem are examined. Intelligent methods have demonstrated superior efficiency, since the problem is a NP-hard benchmark. A co-evolutionary algorithm for solving this problem is presented in this paper. The algorithm consists of two populations that simultaneously evolve using genetic operations. Eventually, the complete solution is found through merging individual members of the two populations. The effectiveness of the procedure has been compared to that of a simple genetic algorithm. The obtained results show significant advantage over the traditional routines.*

## Key Words

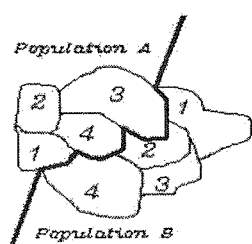
*Symbiosis, Co-evolution, Optimization, Graph-Coloring Problem,*

## مقدمه

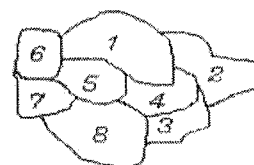
مسئله گراف کالرینگ، یکی از مسائلی ارضا قیود (CSP) موجو در ادبیات هوش مصنوعی میباشد. این مسئله شامل رنگ آمیزی یک نقشه با حداقل تعداد رنگ میباشد، بطوریکه هیچ دو شهر مجاور نقشه، دارای رنگ یکسان نباشد. از آنجا که جغرافیای یک نقشه را میتوان بشکل یک گراف معادل، در نظر گرفت، در حالات خاص، میتوان روشهای موجود در تئوری گرافها را برای آن، گسترش داد [2]. در عین حال، بسیاری از روشهای سنتی موجود در هوش مصنوعی، برای حل این مساله، بکار گرفته شده است [3,4]. با توجه به اینکه مسئله گراف کالرینگ، دارای پیچیدگی NP-hard میباشد [5, 6]، توانایی روشهای مذکور، با ازدیاد تعداد شهرهای موجود در گراف، رو به کاهش میرود. ظاهراً، در حل مسائل NP-hard، استفاده از روشهای هوشمندتری چون شبکه های عصبی و الگوریتمهای ژنتیک، میتواند سودمند باشد [7]، که برای بسیاری از مسائل از این دست، کارآیی این روشها، نشان داده شده است [8,9,10]. مسئله گراف کالرینگ نیز توسط الگوریتمهای ژنتیک و تکاملی مختلفی مورد بررسی قرار گرفته و راه حلهایی نیز برای آن ارائه شده است [1,11,12,13]. در این مقاله، سعی در ارائه یک الگوریتم تکاملی توام، برای حل مساله گراف کالرینگ، گردیده است. الگوریتم تکاملی توام، در حقیقت، بکارگیری الگوریتمهای تکاملی موجود، بر روی بیش از یک جمعیت و ارتباط بین آنها، میباشد. بسته به اینکه، بین این جمعیتها، رابطه رقابتی بیشتر باشد یا رابطه همکاری، الگوریتم تکاملی توسعه یافته، بترتیب از الگوریتمهایی تحت عنوان سیمبایاسیس دور و یا نزدیک میشود. در الگوریتمهای سیمبایاسیس، بهبود خواص جمعیتها، با همکاری و در جهت کمک به یکدیگر انجام میگردد.

## الگوریتم توسعه یافته

همانطور که گفته شد، اساس الگوریتم سیمبایاسیس توسعه یافته در این مقاله، الگوریتم ژنتیک ارائه شده در [1]، برای حل مسئله مپ کالرینگ میباشد. برای شرح الگوریتم تکاملی دوجانبه مورد نظر در این مقاله، نقشه نشان داده شده در شکل 1 را در نظر میگیریم. در الگوریتم ژنتیک ارائه شده در [1]، تمامی 8 شهر موجود در شکل 1، بعنوان یک جمعیت در نظر گرفته شده است ولی در این مقاله، ما جمعیت کل مساله را به دو جمعیت کوچکتر (و تا حد امکان هم اندازه) تقسیم میکنیم. در شکل 2، نمونه ای از تقسیم جمعیت شکل 1 به دو جمعیت، نشان داده شده است.

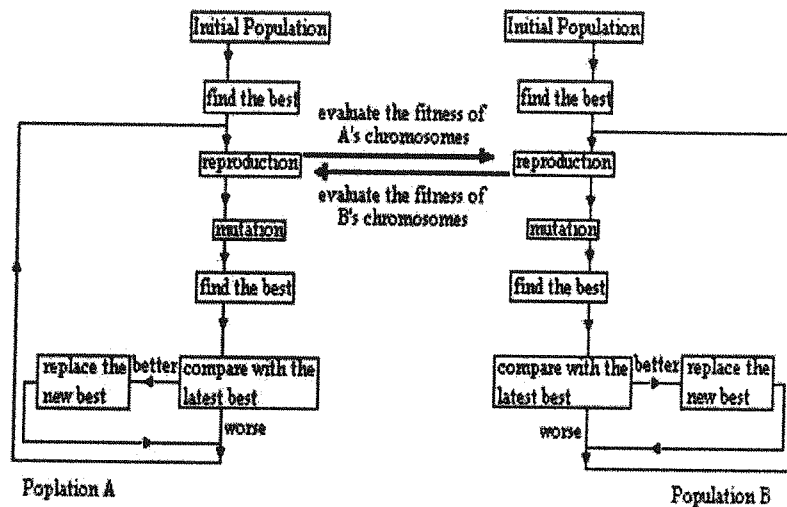


شکل (۲)



شکل (۱)

در حقیقت، هدف ما از الگوریتم مذکور، رنگ آمیزی دو تکه نقشه بطور جداگانه میباشد، بطوریکه پس از اتصالشان به یکدیگر نیز، هیچ تداخل رنگی وجود نداشته باشد. برای این منظور، ما یک الگوریتم ژنتیک بر روی هر یک از این دو جمعیت، بطور جداگانه، پیاده مینماییم. الگوریتم ژنتیک ما، شامل اپراتورهای تولید مجدد (Cross over) و جهش (Mutation) میباشد (الگوریتم ارائه شده در [1] نیز اندکی تغییر داده شده است تا نتایج آن با الگوریتم این مقاله، قابل مقایسه گردد). نحوه ارتباط دو جمعیت مذکور در الگوریتم توسعه یافته، در مرحله ارزیابی کروموزومهای هر جمعیت میباشد، بدینمعنی که، همانطور که خواهیم دید، تابع برازش (Fitness Function) بکار رفته در این الگوریتم، با توجه به کل جمعیتها میباشد. بنابراین، میتوان الگوریتم مورد نظر را، مانند آنچه در شکل 3 آمده است نمایش داد.



شکل (۳)

پیدا کردن بهترین کروموزوم هر نسل، صرفاً بمعنی استفاده از ثبات (Register) در جستجوی جواب میباشد تا در مواردی که بعلت حجم زیاد محاسبات، مجبور به قطع الگوریتم، قبل از پیدا شدن جواب ایده آل گشتیم، بهترین جواب پیدا شده را داشته باشیم. حال به تشریح جزئیات پیاده سازی الگوریتم میپردازیم. برای پیاده سازی الگوریتم، ابتدا باید نقشه شهرهای مورد نظر را، به شکلی کد نماییم تا بتوانیم آنرا به الگوریتم توسعه یافته، بخورانیم. اینکار در [1]، توسط یک ماتریس انجام گردیده است. در این مقاله، ما از همان ایده، استفاده کرده و یک نقشه را با سه ماتریس کد مینماییم. بدینصورت که دو ماتریس برای نشان دادن وضعیت شهرها در هر یک از جمعیت‌های A و B و یک ماتریس نیز برای تعیین وضعیت همسایگی بین جمعیت‌های A و B در مرز تماس آنها در نظر گرفته میشود. بعنوان نمونه، در شکل 4، ماتریسهای متناظر با نقشه نشان داده شده در شکل 2، نمایش داده شده است:

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$Adj = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

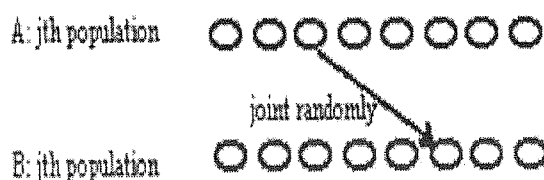
شکل (۴)

نحوه تشکیل ماتریسهای A و B، بدینصورت میباشد که ابتدا یک ماتریس مربعی  $n \times n$  تشکیل میدهیم که n تعداد شهرهای موجود در جمعیت مورد نظر میباشد. در سطر و ستون این ماتریس، شهرها را بترتیب شماره گذاریشان، قرار میدهیم و بنابراین آرایه  $A_{ij}$  این ماتریس، نمایانگر تلاقی دو شهر  $i$ ام و  $j$ ام میباشد. حال بسته به اینکه این دو شهر، با هم مجاور باشند یا نه، بترتیب این آرایه، 1 یا 0 میگردد. البته از آنجا که مجاورت، یک رابطه جابجایی است، بدیهی است که اگر مقدار  $A_{ij}$  برابر 1 شود، مقدار  $A_{ji}$  هم برابر 1 میشود ولی برای کاهش حجم محاسبات، ما یکی از این دو را 1 و دیگری را 0 در نظر میگیریم که در بهترین حالت، میتوان یک ماتریس بالا مثلثی (یا پایین مثلثی) بدست آورد. به این نکته نیز باید توجه نمود که همسایگی یک شهر با خودش، از لحاظ تداخل رنگهای آن، بی معنی است و بنابراین تمامی آرایه های موجود در روی قطر اصلی ماتریسهای A و B، صفر میباشد. همانطور که گفتیم، ماتریس همسایگی ADJ، نیز بیانگر جغرافیای شهرهای موجود روی مرز دو جمعیت A و B میباشد. برای تشکیل آرایه های این ماتریس نیز، ابتدا شهرهای A و B را بر حسب شماره گذاریشان، بترتیب

در ستون و سطر ماتریس ADJ در نظر گرفته و بنابراین ADJij بیانگر تلاقی شهر i ام جمعیت A با شهر j ام جمعیت B میباشد و بسته به اینکه این دو شهر با هم در تماس باشند یا نه، آرایه مذکور بترتیب ۱ یا ۰، خواهد بود. بنابراین، تنها یکسانی رنگ شهرهایی که آرایه متناظرشان، ۱ میباشد، باید مورد بررسی قرار گیرد. حال، برای ارزیابی کروموزومها و میزان نزدیکیشان به جواب مسئله، باید تابع مطلوبیتی در نظر گرفت. تابع مطلوبیت در الگوریتم توسعه یافته، باید طوری در نظر گرفته شود که ارتباطی همکارانه، بین دو جمعیت در حال بهبود، برقرار سازد و در نهایت با رسیدن آن به یک مقدار مشخص، جواب مسئله کامل بدست آید. همانطور که میدانیم، از اتصال هر کروموزوم از جمعیت A و هر کروموزوم از جمعیت B، میتوان یک رنگبندی برای کل نقشه، بدست آورد. از طرفی، از آنجا که نگرش ما نسبت به دو جمعیت A و B، هیچ تفاوتی نمیکند، در نهایت ما برای ارزیابی کروموزوم A ام از جمعیت A، آنرا با یکی از کروموزومهای نسل j ام از جمعیت B، متصل مینماییم و عکس تعداد شهرهای مجاور هم رنگ موجود در کل نقشه حاصل از اتصال فوق را، بعنوان تابع مطلوبیت کروموزوم مذکور، در نظر میگیریم، یعنی داریم:

$$ff(chA_{j,i}) = \frac{1}{N \left( chA_{j,i}, chB_{i,rand} \right)} \quad (1)$$

که در رابطه مذکور، n تعداد کروموزومهای موجود در هر یک از جمعیتهای A و B میباشد و N نیز تعداد شهرهای مجاور هم رنگ، در کل نقشه حاصل از دو کروموزوم A(j,i) و B(j,rand) میباشد. اندیسهای I و j نیز بترتیب شماره جمعیت و شماره کروموزوم آن جمعیت، میباشد. بدیهی است که هر چه مقدار N، کمتر گردد، تابع مطلوبیت افزایش یافته و در حالت جواب که هیچ تلاقی رنگی در شهرها، وجود ندارد، N برابر صفر و تابع مطلوبیت، بینهایت میگردد. در شکل ۵، این مطلب نشان داده شده است:



شکل (۵)

برای ارزیابی کروموزومهای جمعیت B نیز، عینا مانند جمعیت A، عمل مینماییم. البته شیوه های مشابهی برای ارزیابی کروموزومهای جمعیتهای A و B، بررسی شده و از آنجا که حجم محاسبات کل الگوریتم با توجه به حجم محاسبات ارائه شده در [1]، بیشتر بوده، روش بالا انتخاب شده است. مثلا میتوانیم کروموزومهای A ام جمعیت j ام A را با تمامی n کروموزوم جمعیت j ام B، تلفیق و مینیمم یا میانگین مقدار تلاقی رنگهای کل نقشه در این تلفیقات را بعنوان تابع خوبیت کروموزوم Aij مذکور، در نظر گرفت ولی بدلیل ذکر شده، این روشها مناسب ارزیابی نشد. در این مرحله، ما قادر به پیاده سازی اپراتور تولید مجدد هستیم. در این اپراتور که پس از تولید جمعیت آغازین اعمال میگردد، ما سعی در انتخاب کروموزومهای بهتر نسل قبل برای وجود در نسل بعد، داریم. در حقیقت، احتمال بقای کروموزومهای هر نسل در نسل بعد، با توجه به میزان خوبیتشان، تعیین میگردد. همانطور که میدانیم، خوبی یک کروموزوم بسته به بیشتر بودن مقدار تابع مطلوبیت آن دارد. بنابراین، احتمال اعمال شده در اپراتور تولید مجدد یک کروموزوم در نسل بعد، مانند زیر در نظر گرفته میشود:

$$P_{reproduction}(chA_{j,i}) = \frac{ff(chA_{j,i})}{\sum_{i=1}^n ff(chA_{j,i})} \quad (2)$$

سپس نوبت به اپراتور جهش میگردد که برای اعمال آن، باید مشخص کرد که رنگ کدام کروموزوم و کدام شهر از آن

کروموزوم را می‌خواهیم با کدامیک از رنگهای موجود، عوض نماییم. انتخاب این سه پارامتر، بدلیل عدم وجود هر گونه رجحیتی، در فضای احتمالی همگن، با روابط زیر، صورت می‌گیرد:

$$P_{\text{mutation-color}} = \frac{1}{n} \quad (3)$$

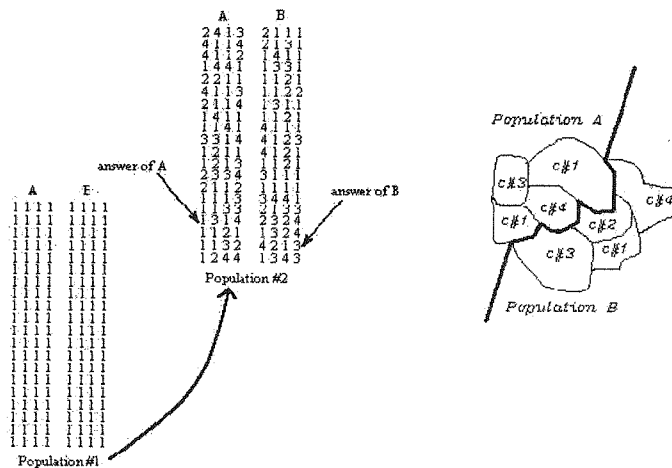
$$P_{\text{mutation-city}} = \frac{1}{m} \quad (4)$$

$$P_{\text{mutation-city}} = \frac{1}{c} \quad (5)$$

که در روابط فوق،  $n$  تعداد کروموزومهای موجود در هر جمعیت،  $m$  تعداد شهرهای موجود در هر کروموزوم و  $c$  تعداد رنگهای موجود برای رنگ آمیزی نقشه، میباشد. در این مرحله، ما قادریم تا الگوریتم مورد نظر را پیاده‌سازی نماییم.

### پارامترهای الگوریتم و اجرای آن

همانطور که گفتیم، ما می‌خواهیم نتایج بدست آمده از الگوریتم تکاملی توام ارائه شده در این مقاله را با نتایج الگوریتم ژنتیک ارائه شده در [1] (که اندکی تغییرات در آن ایجاد نموده ایم)، مقایسه نماییم. جمعیت آغازین انتخاب شده در اجرای الگوریتم، بصورتی انتخاب شده است تا توانایی الگوریتم در بدترین شرایط، ارزیابی شود. بدین معنی که تمامی شهرهای موجود در کروموزومهای جمعیت آغازین، هم‌رنگ انتخاب شده اند. تعداد کروموزومهای موجود در هر نسل، 20 میباشد. در گذار از هر نسل به نسل بعد، اپراتور تولید مجدد، یکبار و اپراتور جهش، 80 بار اعمال میگردد. نتیجه یکی از اجراهای الگوریتم و نحوه رسیدن به جواب برای رنگ آمیزی نقشه شکل 2، در شکل 6، نشان داده شده است. در رنگ آمیزی این نقشه، از 4 رنگ استفاده شده و همانطور که در شکل 6 مشاهده میگردد، کروموزوم جواب در نسل 2، پیدا شده است.

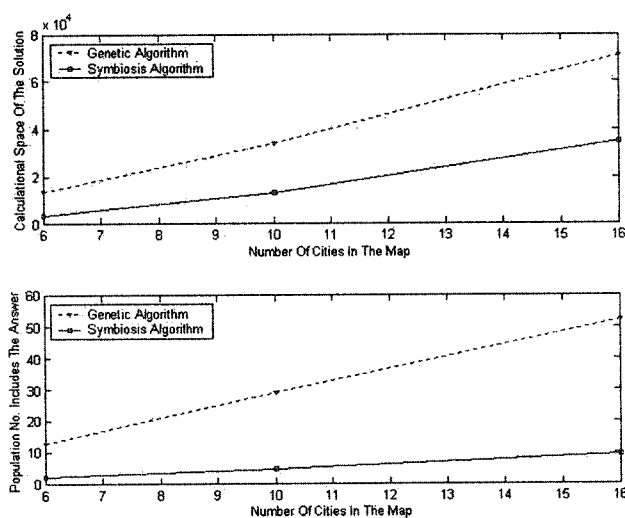


شکل (6)

### مقایسه الگوریتم سیمبایاسیس با الگوریتم ژنتیک مشابه

همانطور که گفتیم، مسئله گراف کالرینگ، دارای پیچیدگی NP-hard میباشد. این مطلب، بدین معنی است که با افزایش تعداد شهرهای موجود در نقشه، روند حجم محاسبات آن، بصورت غیر چند جمله ای، افزایش می‌یابد. بنابراین، از آنجا که الگوریتم سیمبایاسیس توسعه یافته، بر روی حجم کمتری از مسئله، اعمال میگردد، انتظار میرود تا حجم محاسبات مورد نیاز و تعداد نسل تا رسیدن به جواب، در الگوریتم ارائه شده در این مقاله، کمتر از کمیتهای مشابهشان در الگوریتم ژنتیک ارائه شده در [1]، باشد. مطلب فوق، با بکارگیری دو الگوریتم مذکور، بر روی چند نقشه با ابعاد مختلف و مشاهده نتایج، تصدیق شده که نتایج مربوطه، در شکل 7، آمده است. البته از آنجا که خروجی الگوریتمهای مذکور، تصادفی میباشد، نتایج بدست آورده شده

در شکل 7، متوسط 100 بار اجرای الگوریتم، در هر یک از حالات میباشد.



شکل (۷)

## نتیجه گیری

الگوریتمهای ژنتیک و تکاملی، در سالهای اخیر، بعنوان ابزارهای بسیار کارآمد در بهینه سازی، وسیعاً مورد استفاده قرار گرفته‌اند. توانایی این الگوریتمها در گریز از نقاط بهینه محلی و همچنین عدم نیاز به مشتق گیری محلی و استفاده از ماتریسهای جاکوبی و هسین و قابلیت تعمیم به مسائل تصمیم گیری چند معیاره، از عوامل بسیار مهم جذابیت آنها میباشد. در برابر این امتیازات چشمگیر، این ابزارها دارای نقاط ضعفی مشابه نقاط ضعف تکامل در طبیعت هستند. از جمله پیچیدگی زمانی آنها که رسیدن به پاسخ مطلوب را مستلزم طی نسلهای طولانی از جمعیت‌های غیر بهینه میسازد. روشهای افزایش کارایی این ابزارها هم میتواند با الهام گرفتن از فرایندهای طبیعی، کشف شود. در این مقاله، روش تکامل توام، بعنوان یک توسعه مناسب از ابزار یاد شده، برای حل مسائل پیچیده مطرح شده است. مسئله نمونه انتخاب شده (مسئله رنگ آمیزی گراف)، در مرجع [1]، با استفاده از الگوریتم ژنتیک حل شده است و مسئله پیچیدگی، همانگونه که بصورت نظری قابل اثبات است، بصورت تجربی هم کاملاً مشهود میباشد. تجربه بکارگیری روش پیشنهادی، نشان دهنده بهبود قابل ملاحظه توانایی الگوریتم، در رویارویی با مسئله پیچیدگی، میباشد. نتایج مثبت حاصل از این تجربه، حاکی از امکان بکارگیری روش پیشنهادی در حل مسائل پیچیده بهینه‌سازی و تصمیم گیری میباشد.

## مراجع

- [1] د. شهمرزادی "ارائه یک الگوریتم ژنتیک برای حل مسئله Map Coloring" چهارمین کنفرانس دانشجویی - صنعتی سیستمهای هوشمند، اسفند 1380، دانشگاه خواجه نصیرالدین طوسی، تهران، ایران
- [2] C. Lin, G. Hu, Z. Zhang "A Six-Color Theorem for the Edge-Face Coloring of Plane Graphs" Discrete Mathematics, 141 (1995), PP:291-297.
- [3] A. Hertz, D. De werra "Using Tabu Search Techniques for Graph Coloring" Computing, No.39, PP: 345-351, 1987.
- [4] M. Chams, A. Hertz, D. De werra "Some Experiments With Simulated Annealing for Coloring Graphs" European Journal Operational Research 32(1987), PP: 260-266.
- [5] H. P. Schwefel, G. Rudolph "Contemporary Evolution Strategies" 3 rd International Conference on Artificial Life (Lecture Notes in Artificial Intelligence, Vol. 929), F. Moran, A. Moreno, J. J. Merelo, P. Chacon, Eds. Berlin, Germany: Springer, PP:893-907, 1995.
- [6] M. R. Gary, D. S. Johnson "Computers and Intractability: A Guide to the Theory of NP-Completeness", W. H. Freeman and Company, 1979.
- [7] W. M. Spears "Using Neural Networks and Genetic Algorithms as Heuristics for NP Complete Problems" Ms Thesis in Computer Science, Fall 1989, George Mason University.
- [8] W. A. Greene "Genetic Algorithms for Partitioning Sets" International Journal on Artificial Intelligence Tools,

Vol. 10, No.s 1&2, (2001), PP: 225-241.

- [9] B. Freisleben, P. Merz "A Genetic Local Search Algorithm for Solving Symmetric and Asymmetric Traveling Salesman Problems" Proceeding of ICEC'96 , PP: 616-621, 1996.
- [10] D. R. Jones and M. A. Beltramo "Solving Partitioning Problems with Genetic Algorithms" Proceeding of 4 th ICGA, PP: 442-449, 1991.
- [11] L. Davis, Handbook of Genetic Algorithms, Chapter 6, "Order-Based Genetic Algorithms and the Graph Coloring Problem" Van Nostrand Reinhold, 1991.
- [12] C. Fleurent, J. A. Ferland "Genetic and Hybrid Algorithms for Graph Coloring" Annals of Operations and Research, 63 (1996), PP: 437-461.
- [13] F. F. Ali, Z. Nakao, R. B. Tan, Y. Chen "An Evolutionary Approach for Graph Coloring" Proceeding of SMC'99 Conference on Systems, Man and Cybernetics, Vol. 5, 1999.
- [14] K. Tagawa, K. Kanesige, K. Inoue, H. Haneda "Distance Based Genetic Algorithm: An Application for the Graph Coloring Problem" Proceeding of the CEC'99 Congress on Evolutionary Computing, Vol. 3, 1999.