

راهنمای سوال الگوریتم ژنتیک

در اینجا می‌خواهیم نحوه حل سوال مربوط به رنگ‌آمیزی گراف در الگوریتم ژنتیک را شرح دهیم. فرض کنید می‌خواهیم نقشه‌ی استان‌های ایران را با ۴ رنگ، رنگ‌آمیزی کنیم به طوری که هیچ ۲ استان مجاوری رنگ مشابه نداشته باشند. خروجی نهایی مورد نظر ما مشابه تصویر زیر می‌باشد :



این مسئله یک مسئله‌ی NP-Complete می‌باشد. یکی از روش‌های حل این مسئله، استفاده از الگوریتم ژنتیک می‌باشد.

در حالت کلی، مراحل حل این مسئله به کمک الگوریتم ژنتیک به صورت زیر است :

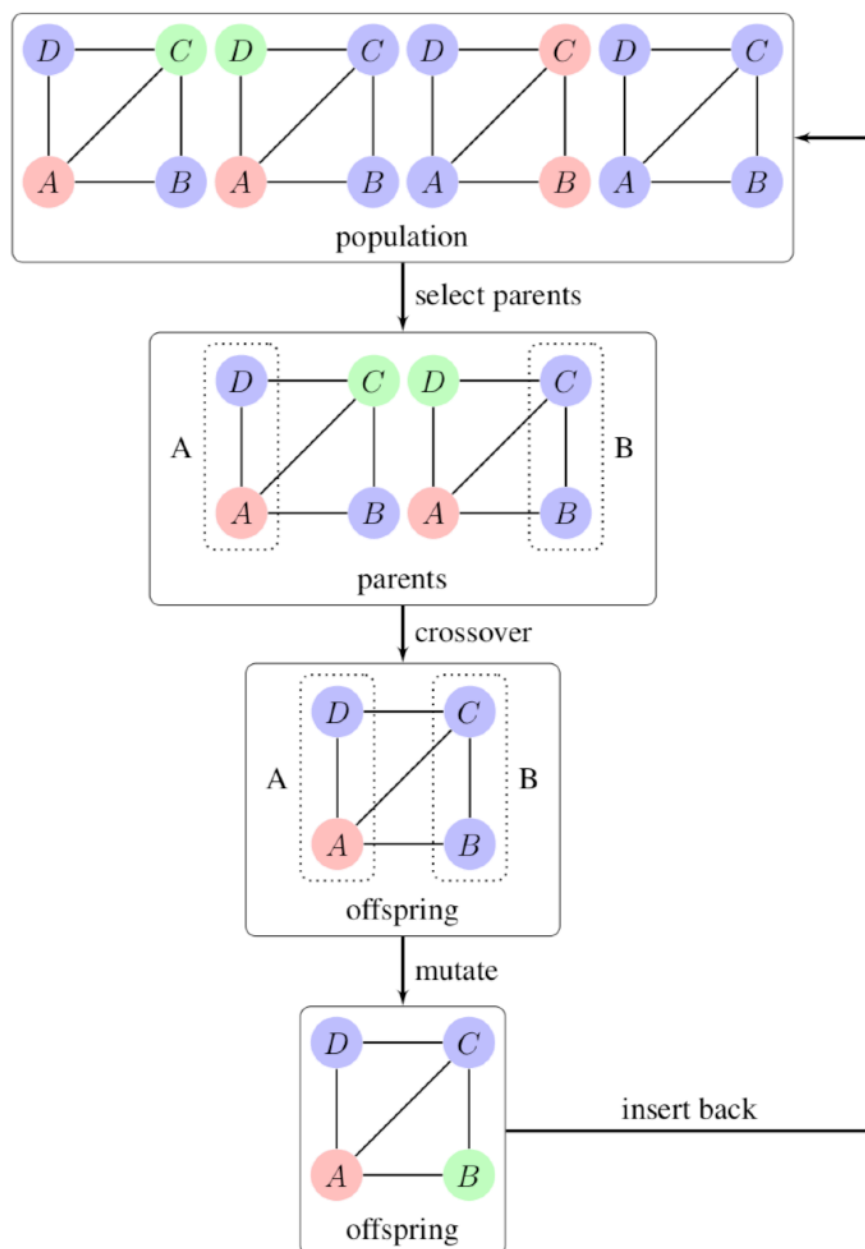


Figure 2.2: Example iteration of an evolutionary algorithm

جهت حل مسئله‌ی رنگ‌آمیزی گراف، مراحل زیر را دنبال کنید :

مرحله ۱ (تولید جماعت اولیه) : جماعت اولیه را با population نام‌گذاری می‌کنیم. توجه کنید که تعداد جمعیت اولیه (populationSize) از پارامترهای قابل تنظیم مسئله می‌باشد. هریک از اعضای population، یک chromosome نام دارد. در ابتدا تمام گره‌های هریک از chromosome ها را به صورت رندوم، رنگ‌آمیزی کنید.

مرحله ۲ (تعیین میزان شایستگی هر chromosome) : تابع fitness function، بیانگر میزان شایستگی هر chromosome در مسیر رسیدن به پاسخ مسئله می‌باشد که از آن برای تعیین والدین جهت تولید نسل بعدی استفاده می‌کنیم.

برای هریک از chromosome ها، تابع fitness function را به صورت زیر تعریف کنید :

فرض کنید $G = (V, E)$ گرافی با n گره و m یال باشد. اگر رنگ متعلق به گره‌ی i را با $c(i)$ نمایش دهیم، یک رنگ‌آمیزی گراف به صورت $C(G) = \{c(0), c(1), \dots, c(n)\}$ می‌باشد.

تابع δ را به صورت زیر تعریف می‌کنیم :

$$\text{for all } i, j \in E, \delta(i, j) = 1 \text{ if } c(i) \neq c(j) \text{ and } \delta(i, j) = 0 \text{ if } c(i) = c(j)$$

و با استفاده از این تابع، fitness function مورد نظر ما به صورت زیر تعریف می‌شود :

$$F(C(G)) = \frac{\sum_{i, j \in E} \delta(i, j)}{m}$$

برای هر chromosome، مقدار fitness را با توجه به این تابع به دست می‌آوریم.

مرحله ۳ (انتخاب والدین) : در این مرحله از روش tournament selection استفاده می‌کنیم. در این روش تعداد k عضو را به صورت random انتخاب کرده و بهترین آن‌ها را برمی‌گزینیم. مثلاً اگر جماعت اولیه برابر ۱۰۰ باشد و k را برابر ۴ قرار دهیم، ۲۵ tournament اتفاق می‌افتد و ۲۵ عضو برگزیده می‌شوند. توجه کنید که k از پارامترهای قابل تنظیم می‌باشد و tournamentSize نام دارد. جماعت برگزیده، والدین نامیده می‌شوند.

مرحله ۴ (تولید نسل جدید) : در مرحله‌ی بعد، باید تولید نسل جدید انجام گیرد. به این مرحله crossover گفته می‌شود. در این مرحله باید نسل جدید از طریق ترکیب هر ۲ والد برگزیده از مرحله‌ی قبل صورت گیرد. الگوریتم زیر، حالت ساده‌ای از تولید فرزندان را نشان می‌دهد :

```
for i = 1 to populationLength :
    select x randomly from parents
    select y randomly from parents where (y != x)
    newChromosome[i] = crossover(x,y)
end for;
```

در این مرحله، باید تابع crossover را تعریف کنیم.

روش‌های مختلفی برای تعریف تابع crossover وجود دارد. در اینجا ما هر عضو از population را به دو زیرگراف دلخواه تقسیم کرده به طوری که اشتراک این دو زیرگراف تهی و اجتماع این دو زیرگراف یک chromosome جدید را به وجود آورد.

مرحله ۵ (جهش): در این مرحله، تابع mutation را تعریف می‌کنیم. فرض کنید هر chromosome دارای تعداد n گرهی قابل رنگ آمیزی باشد. اصطلاحاً گفته می‌شود که هر کروموزوم، n ژن در اختیار دارد. همچنین پارامتر mutationRate را به عنوان یک پارامتر قابل تنظیم، تعریف می‌کنیم. تعداد ژن‌های جهش یافته از رابطه‌ی زیر به دست می‌آید:

$$\text{mutatedGenomes} = \text{populationLength} * n * \text{mutationRate}$$

و سپس به تعداد mutatedGenomes، ژن‌های موجود در population را به صورت رندوم تغییر می‌دهیم. در مثال رنگ آمیزی نقشه ایران، تعداد ۳۱ استان در اختیار داریم و فرض می‌کنیم جماعت اولیه را برابر ۱۰۰ در نظر گرفته‌ایم و mutationRate را برابر ۰.۰۲ قرار داده‌ایم. در این حالت رنگ ۶۲ ژن را به صورت رندوم تغییر می‌دهیم.

مرحله ۶ (بازگشت): در این مرحله، جماعت جدید ایجاد شده را به عنوان جماعت اولیه قرار می‌دهیم و کلیه مراحل را به اندازه‌ی متغیر numberOfGenerations که از پارامترهای قابل تنظیم مسئله می‌باشد، تکرار می‌کنیم.

از این الگوریتم جهت رنگ آمیزی نقشه‌ی ایران با تعداد ۴ رنگ، استفاده کنید.

مسئله را با تمام پارامترهای زیر، حل کنید و در نهایت درمورد تاثیر هر پارامتر در همگرایی پاسخ، بحث کنید.

numberOfGenerations = 50, 500, 5000
populationSize = 10, 100, 1000
tournamentSize = 2 for populationSize=10
tournamentSize = 2,5,10 for populationSize=100 and 1000
mutationRate = 0.01, 0.02, 0.05, 0.1

در هر generation، بهترین، بدترین و میانگین تابع شایستگی را ذخیره کرده و در نهایت نمودار میزان این ۳ متغیر در طول generations را نمایش دهید.
