گزارش پروژه سوم

نگار میرگتی ۸۱۰۱۹۴۴۱۳

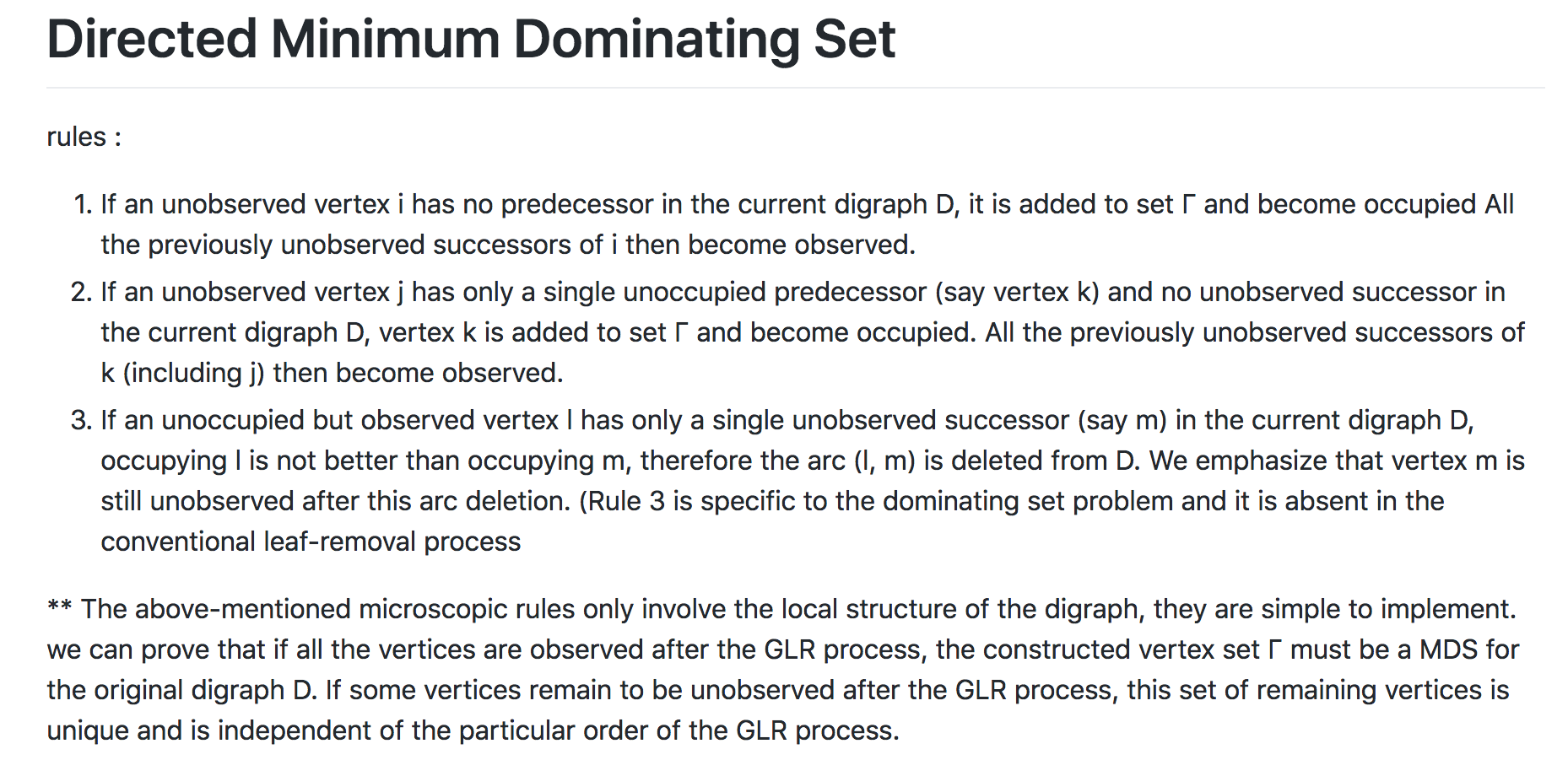
1. کوچکترین مجموعه غالب :
   1. توضیح پیاده سازی : ابتدا اگر راس ایزوله ای وجود داشته باشد به مجموعه جواب اضافه میشود. سپس در یک حلقه تا زمانی که همه ی رئوس مارک نشده باشند یک راس که دارای بیشترین همسایه های مارک نشده باشد انتخاب می شود و به مجموعه جواب اضافه می شود. همچنین خود راس و همه ی همسایه هایش مارک می شوند.

خروجی الگوریتم به ازای گراف شطرنج ۱۲\*۱۲ : ۷

پیچیدگی الگوریتم : O(V + E) حلقه ی کلی برنامه حداکثر V بار تکرار می شود. درون حلقه هر بار باید به ازاء هر راس تعداد همسایه های مارک نشده اش بدست آید و سپس راس ماکسیمم محاسبه شود.

* 1. اولین پیاده سازی : مشابه قسمت قبل با این تفاوت که در هر مرحله راسی که تعداد همسایه های مارک نشده اش که یال از آنها به این راس وارد می شود بیشترین است انتخاب می شود. پیچیدگی نیز مشابه قسمت قبل است.

راه حلی بهتر: (CA3\_1\_directed.R, not completed yet)



* 1. الگوریتم حریصانه پیاده سازی شده مانند الگوریتم قسمت اول می باشد با این تفاوت که تا زمانی که به α درصد نرسیده ایم رئوس به مجموعه غالب اضافه می شوند.
  2. مساله را می توان به صورت مقابل مدل کرد که در آن wi وزن رئوس است و xi بیانگر عضو بودن یا نبودن هر یک از رئوس در مجموعه غالب می باشد. این مدل integer programming است.

Minimum Weighted Dominating Set

Min (Z = Σwixi)

Σaijxj >= 1 (for all 1 <= i <= n)

xi ∈ {0, 1} (for all 1 <= i <= n)

1. رنگ آمیزی :
2. پیچیدگی الگوریتم :حلقه ی کلی برنامه V بار تکرار می شود. درون حلقه رنگ های مربوط به همه ی همسایه های این راس بدست می آید(deg(v)) و اگر از بین مجموعه رنگ های فعلی رنگی وجود داشت که در این مجموعه نبود به آن راس داده می شود(V^2) در غیر این صورت رنگ جدیدی به مجموعه رنگ های استفاده شده اضافه می شود. در نهایت پیچیدگی برابر خواهد بود با : O(V\*(V^2 + Δ(G))) در صورتیکه از ساختمان داده ی مناسب تری استفاده شود پیچیدگی مربوط به بخش پیدا کردن تفاضل مجموعه رنگ های استفاده شده و مجموعه رنگ های همسایه ها را میتوان به VLogV یا حتی V کاهش داد.
3. مساله به صورت مقابل مدل می شود.

Wj = {0, 1} , Wj = 1 if color j is used and otherwise 0.

target : minimize number of colors used -> min{ ΣWj }

Constraints:

∀j ∈V Σxij = 1 : every node is assigned 1 and only 1 color

∀u, v∈E, j∈C xuj + xvj ≤ 1: avoiding color conflicts

∀i∈V, j∈C xij ≤ wj  : if any node is colored with color j then wj=1

مقایسه خروجی دو الگوریتم : برای گراف با سایز مطرح شده در صورت پروژه بنظر می آید در بی نهایت جواب می دهد که فکر میکنم به این دلیل است که تمام متغیر های x در تابع هدف وجود دارند. بنابراین روی گراف خیلی کوچکتر تست کردم و تفاوت زمان اجرا برای همین گراف کوچک بسیار زیاد است!

g1 <- erdos.renyi.game(30, 65, type = "gnm", directed = F, loops = F)

> greedy\_coloring(g1)

Time difference of 0.1208119 secs

num\_of\_colors in greedy coloring = 4

> ip\_ans <- ip\_coloring(g1)

Time difference of 24.41889 secs

Success: the objective function is 3

پ.ن : در ابتدا این پروژه را با حدود ۱۹ ساعت تاخیر آپلود کرده بودم. دراین ورژن جدید فقط پیاده سازی بخش ب سوال ۲ را اضافه کردم.