گزارش پروژه سوم

نگار میرگتی ۸۱۰۱۹۴۴۱۳

1. کوچکترین مجموعه غالب :
   a. 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.

   b. 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)

Directed Minimum Dominating Set

rules :

1. If an unobserved vertex i has no predecessor in the current digraph D, it is added to set Γ and become occupied All the previously unobserved successors of i then become observed.
2. If an unobserved vertex j has only a single unoccupied predecessor (say vertex k) and no unobserved successor in the current digraph D, vertex k is added to set Γ and become occupied. All the previously unobserved successors of k (including j) then become observed.
3. If an unoccupied but observed vertex l has only a single unobserved successor (say m) in the current digraph D, occupying l is not better than occupying m, therefore the arc (l, m) is deleted from D. We emphasize that vertex m is still unobserved after this arc deletion. (Rule 3 is specific to the dominating set problem and it is absent in the conventional leaf-removal process

** The above-mentioned microscopic rules only involve the local structure of the digraph, they are simple to implement. we can prove that if all the vertices are observed after the GLR process, the constructed vertex set Γ must be a MDS for the original digraph D. If some vertices remain to be unobserved after the GLR process, this set of remaining vertices is unique and is independent of the particular order of the GLR process.

   c. 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.
   d. مساله را می توان به صورت مقابل مدل کرد که در آن w_i وزن رئوس است و x_i بیانگر عضو بودن یا نبودن هر یک از رئوس در مجموعه غالب می باشد. این مدل integer programming است.

Minimum Weighted Dominating Set

Min $(Z = \Sigma w_i x_i)$

$\Sigma a_{ij} x_j >= 1$ (for all $1 <= i <= n$)

$x_i \in \{0, 1\}$ (for all $1 <= i <= n$)

2. رنگ آمیزی :

a. 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.

b. مساله به صورت مقابل مدل می شود.

$W_j = \{0, 1\}$, $W_j = 1$ if color j is used and otherwise 0.

target : minimize number of colors used -> $\min\{ \sum W_j \}$

Constraints:

$\forall j \in V$ $\sum x_{ij} = 1$: every node is assigned 1 and only 1 color
$\forall u, v \in E, j \in C$ $x_{uj} + x_{vj} \leq 1$: avoiding color conflicts
$\forall i \in V, j \in C$ $x_{ij} \leq w_j$: if any node is colored with color j then $w_j=1$