



# بهینه‌سازی ترکیبیاتی

محمد هادی فروغمنداعرابی  
بهار ۱۳۹۶

## الگوریتم پیدا کردن تطابق بیشینه در گراف دوبخشی

جلسه دهم

نگارنده: محمد طه طوغانی، سینا اکبری

### ۱ روش

در جلسه گذشته نشان داده شد که:

قضیه ۱.  $M$  تطابق بیشینه است اگر و فقط اگر، مسیر  $M$  - افزایشی وجود نداشته باشد. بنابراین، با توجه به قضیه فوق، برای پیدا کردن تطابق بیشینه مانند زیر عمل می‌کنیم:

---

#### Algorithm 1 PSEUDO CODE

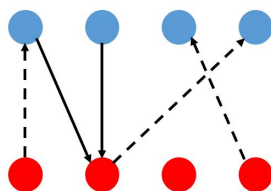
---

```
 $M \leftarrow \emptyset$   
while True do  
    FIND AN AUGMENTING PATH P  
    if P IS FOUND then  
         $M \leftarrow M \Delta P$   
    else  
        break
```

---

برای درک بهتر، گراف را به صورت جهت دار در نظر می‌گیریم به این صورت که یال‌های تطابق جهت پائین به بالا دارند و بقیه یال‌ها جهت بالا به پائین، مانند شکل زیر:

لم ۲. مسیر  $M$  - افزایشی: مسیر جهت‌دار از راس نپوشیده بالا به راس نپوشیده پائین در گراف دوبخشی (یافتن این مسیر با الگوریتم DFS)



نتیجه ۳. تکرار حلقه الگوریتم ذکر شده از مرتبه تعداد رئوس گراف است (چرا؟) و اجرای هر مرتبه الگوریتم  $DFS$  از مرتبه جمع تعداد یال‌ها و رئوس می‌باشد، لذا مرتبه زمانی اجرای الگوریتم از  $O(|V|(|V| + |E|))$  می‌باشد که با فرض این که  $|V|$  از مرتبه  $O(|E|)$  باشد، از مرتبه  $O(|V||E|)$  می‌باشد. لازم به ذکر است که الگوریتم دیگری نیز وجود دارد که مرتبه زمانی آن  $O(|E|\sqrt{|V|})$  است. سه آرایه به صورت زیر، در الگوریتم پیدا کردن مسیر  $M$  - افزایشی تعریف می‌شوند:

$$G[\text{راس پائین}][\text{راس بالا}] = \begin{cases} True & ; \text{دو راس متصل هستند} \\ False & ; \text{دو راس متصل نیستند} \end{cases}$$

$$M[\text{راس پائین}] = \begin{cases} \text{پوشیده شده} & ; \text{راس مقابل در تطابق} \\ -1 & ; \text{پوشیده نشده} \end{cases}$$

برای استفاده در  $dfs$  [راس بالا]  $Mark$

### Algorithm 2 MAX MATCHING

```

check ← True
while check do
    check ← False
    for 0 ≤ i < n do
        mark[ ] ← False
        if i IS NOT MATCHED then
            check ← check OR AUG(i)
    
```

### Algorithm 3 AUGMENTING PATH

```

bool G[UPPER VERTEX][LOWER VERTEX]
bool Mark[UPPER VERTEX]
int M[LOWER VERTEX]
function AUG(v)
    if Mark[v] then
        return False
    Mark[v] ← True
    for 0 ≤ j < n do
        if G[v][j] then
            if (M[j] = -1) || (AUG(M[j])) then
                M[j] ← v
                return True
    return False
    
```

## ۲ بهبود الگوریتم

قضیه ۴. اگر از راس  $V$  در  $M$ ، مسیر  $M$  - افزایشی پیدا نشد، پس از اعمال تعدادی مسیر  $M$  - افزایشی، باز هم از  $V$  مسیر  $M$  - افزایشی پیدا نخواهد شد.

نتیجه ۵. می‌توان الگوریتم پیدا کردن بیشینه تطابق را به صورت زیر بهبود بخشید:



---

**Algorithm 4** MAX MATCHING

---

```
for  $0 \leq i < n$  do  
   $mark[ ] \leftarrow False$   
   $AUG(i)$ 
```

---