Bab 6 Matching

Farida Hanum

Aplikasi: Masalah Pernikahan (*Marriage Problem*), Penjadwalan Tugas, Penjadwalan Waktu dll.

- 6.1 Pendahuluan
- 6.2 *Matching* (Berbobot) Maksimum dalam Graf (Berbobot) *Bipartite*
- 6.3 *Matching* maksimum dalam graf umum

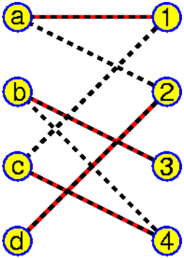
6.1 Pendahuluan

Definition

Suatu *matching* dalam graf G adalah suatu subgraf 1-*regular* dari G, yaitu suatu subgraf yang diinduksi dari kumpulan pasangan *edge* yang tak*adjacent*.

Definition

Suatu *matching* dengan kardinalitas maksimum di G disebut *matching* maksimum dari G.



Matching

Theorem

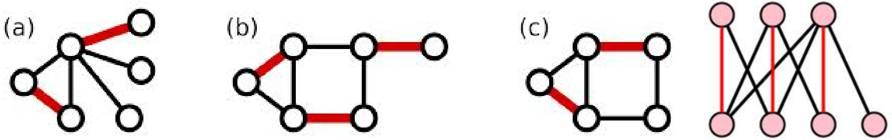
Setiap matching yang berorder p mempunyai paling banyak $p/2$ edge.

Definition

Matching berorder p yang mempunyai tepat $p/2$ *edge* disebut *matching* yang **sempurna** (*perfect*).

Theorem

Jika suatu graf berorder p mempunyai matching sempurna, maka p pasti bilangan genap; tapi tidak sebaliknya.

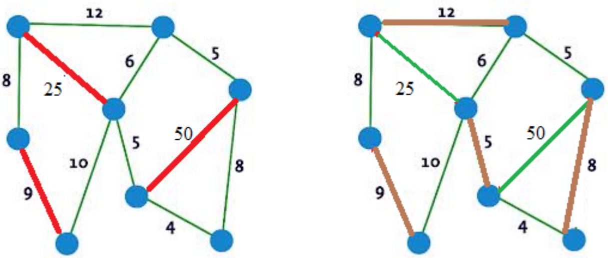


Matching Berbobot Maksimum

Definition

Suatu *matching* dalam graf berbobot dikatakan *matching* dengan **bobot maksimum** jika jumlah bobot setiap *edge*nya adalah maksimum.

Suatu *matching* dengan bobot maksimum belum tentu merupakan *matching* maksimum pada *underlying graph*nya.



matching berbobot maksimum, tapi bukan *matching* maksimum

matching maksimum tapi bukan *matching* berbobot maksimum

Masalah yang terkait dengan matching

1. menentukan *matching* maksimum dalam graf umum,
2. menentukan *matching* maksimum dalam graf *bipartite*,
3. menentukan *matching* berbobot maksimum dalam graf *bipartite* berbobot,
4. menentukan *matching* berbobot maksimum dalam graf (umum) berbobot

(yang dibahas hanya dua *matching* dalam graf *bipartite* saja)

Definisi

Definition

Misalkan M adalah suatu *matching* pada digraf G.

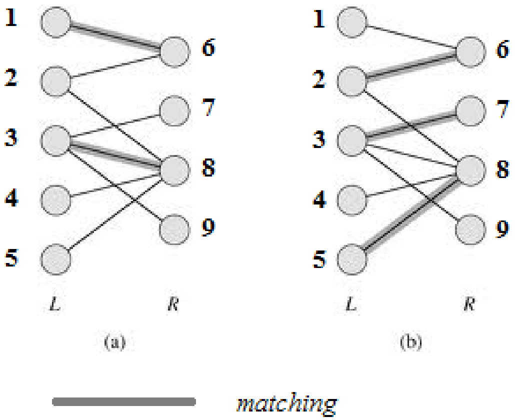
- Suatu *edge* di G yang ada di M dinamakan **sisi yang terpasangkan** (*matched edge*); sedangkan sisi $e \notin M$ dikatakan **sisi yang tidak terpasangkan** (*unmatched edge*).
- Suatu verteks v di G dinamakan **verteks yang terpasangkan relatif terhadap** *matching* M jika v *incident* dengan suatu *edge* di M; jika tidak, maka v dinamakan verteks *single* relatif terhadap *matching* M.

Definition

- Suatu **alternating path** dari G adalah suatu *path* dengan *edge-edge*nya bergantian terpasangkan dan takterpasangkan.
- Suatu *alternating path* taktrivial yang dimulai dan berakhir pada suatu verteks *single*, dinamakan **augmenting path.**

Alternating path dan Augmenting Path

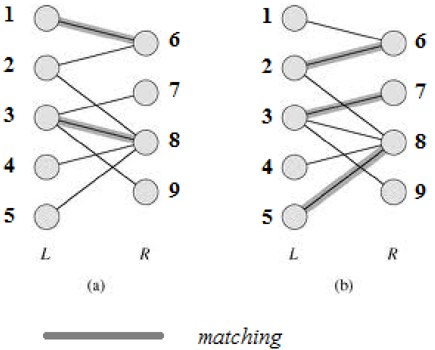
Berikan contoh *alternating path* dan *augmenting path*, *alternating path* tapi bukan *augmenting path*.



Teorema 6.2

Theorem

Suatu matching M pada graf G adalah **matching maksimum** *jika dan hanya jika* **tidak terdapat augmenting path** *relatif terhadap M di graf G.*



(a) bukan matching maksimum, (b) matching maksimum

Misalkan diberikan dua *subset* yang *disjoint* dari $V(G)$, yaitu U_1 dan U_2.

1. U_1 dikatakan **terpasangkan** dengan U_2 jika terdapat *matching* M di G sehingga setiap *edge* dari M *incident* dengan suatu verteks di U_1 dan verteks di U_2, dan setiap verteks di U_1 atau U_2 *incident* dengan *edge* dari M.
2. Jika $M \subseteq M^*$ dengan M^* adalah *matching* di G, maka U_1 juga dikatakan **terpasangkan dengan** U_2 **melalui** M^*.

Ilustrasi:

Definisi

Definition

Misalkan U adalah himpunan verteks dari G, dan $U \neq \varnothing$, dan misalkan *neighborhood* $N(U)$ menyatakan himpunan semua verteks di G yang *adjacent* dengan paling sedikit satu elemen di U, maka U dikatakan **takdefisien** jika

$$|N(S)| \geq |S|$$

untuk setiap *subset* S dari U.

Ilustrasi: –

Teorema 6.3

Theorem

Misalkan G graf bipartite dengan himpunan partisi V_1 dan V_2. Himpunan V_1 dapat dipasangkan dengan V_2 jika dan hanya jika V_1 takdefisien.

Example

Si Unyil mengundang beberapa temannya, yaitu Alvin, Bobo, Cuplis, Dora, Edun, dan Flinstone. Ia telah menyiapkan 6 macam minuman, yaitu es doger (*ed*), es jeruk (*ej*), es campur (*ec*), jus mangga (*jm*), susu coklat (*sc*), dan teh manis (*tm*) masing-masing 1 gelas. Kemudian Unyil menanyakan minuman kesukaan teman-temannya, dan jawabannya adalah: Alvin *ej, jm*; Bobo *ej, sc, tm*; Cuplis *jm, sc, tm*; Dora *ed, ej, ec, jm*; Edun *ej, jm, sc, tm*; dan Flinstone *ej, jm, sc*. Apakah mungkin semua teman Unyil memperoleh minuman kesukaannya?

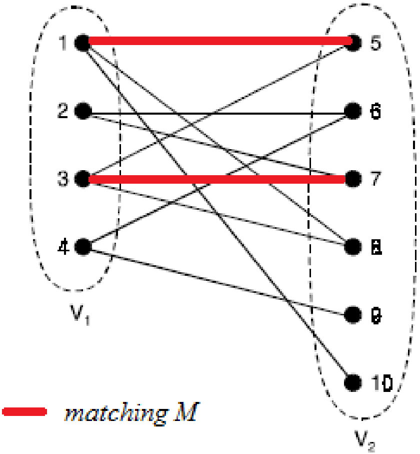
Algoritme penentuan matching maksimum untuk graf bipartite

Algoritme 6.1

1. Misalkan G graf (p, q) dan diberikan *matching* awal M.
2. Periksa apakah M adalah *matching* maksimum. Jika tidak, lanjutkan.
3. Tentukan suatu verteks *single* relatif terhadap *matching* M yang belum pernah menjadi verteks akar pada iterasi sebelumnya, misalkan v.
4. Konstruksi suatu *alternating tree* dengan verteks akar v.
5. Periksa apakah *alternating tree* tersebut dapat diperluas, dan dilanjutkan ke langkah selanjutnya. Jika tidak, berarti *alternating tree* tersebut adalah *augmenting path*.
6. Tentukan *alternating path* yang dapat diperluas, yaitu P, dan *augmen matching* M sepanjang P.

Contoh

Tentukan *matching* maksimum pada graf *bipartite* berikut:



Pelabelan Verteks

Definition

Misalkan $G = (V, E)$ graf *bipartite* berbobot w dengan $V = V_1 \cup V_2$. Suatu **pelabelan verteks yang fisibel** adalah suatu fungsi bernilai real ℓ pada himpunan verteks $V_1 \cup V_2$ sehingga untuk setiap $v \in V_1$ dan $u \in V_2$ berlaku

$$\ell(v) + \ell(u) \geq w(vu).$$

Notasi $\ell(x)$ disebut **label** dari verteks x.

Definition

Misalkan $\ell(v) = \max\{w(vu) \,|\, u \in V_2\}$ untuk semua $v \in V_1$, dan $\ell(u) = 0$, untuk semua $u \in V_2$, maka ℓ merupakan **pelabelan verteks yang fisibel**.

Spanning Subgraph H

Misalkan ℓ adalah pelabelan verteks yang fisibel, dan didefinisikan

$$E_\ell = \left\{ \begin{array}{c} vu \in E(G) \,|\, v \in V_1 \wedge u \in V_2 \\ \wedge \quad \ell(v) + \ell(u) = w(vu) \end{array} \right\}.$$

Misalkan H_ℓ adalah *spanning subgraph* dari G dengan himpunan sisi E_ℓ. Teorema 6.6 berikut berguna untuk Algoritme 6.2.

Theorem

Misalkan ℓ adalah pelabelan verteks yang fisibel pada graf G yang lengkap bipartite dan berbobot. Jika H_ℓ memuat matching yang sempurna (perfect) M', maka M' adalah matching berbobot maksimum dari G.

Algoritme penentuan matching berbobot maksimum untuk graf bipartite lengkap berbobot

Algoritme 6.2 (Kuhn-Munkres)

1. Inisialisasi:
   Untuk setiap $v \in V_1, \ell(v) \leftarrow \max\{w(uv) \,|\, u \in V_2\}$
   Untuk setiap $u \in V_2, \ell(u) \leftarrow 0$
   Misalkan H_ℓ adalah *spanning subgraph* dengan himpunan sisi E_ℓ.
   Misalkan G_ℓ adalah *underlying graph* dari H_ℓ.
2. Gunakan Algoritme 6.1 untuk memperoleh *matching* maksimum, misalkan M.

3. Periksa apakah M *matching* berbobot maksimum.
   a. Jika setiap verteks di V_1 sudah terpasangkan relatif terhadap M, proses selesai. Output: M.
   b. Jika tidak, misalkan x adalah verteks *single* pertama di V_1.
   c. Konstruksi suatu *alternating tree* T yang berakar di verteks x.
   Jika suatu *augmenting path* P dapat ditentukan, maka *augmen matching* M sepanjang P, dan kembali ke Langkah 3a.
   Jika tidak, misalkan T adalah *alternating tree* relatif terhadap M yang berakar di x yang tidak dapat diperluas lagi di G_ℓ.

4. Pelabelan verteks fisibel yang baru:

Hitung

$$m_\ell \leftarrow \min\{\ell(v) + \ell(u) - w(vu) \mid\}$$

untuk $v \in V_1 \cap V(T)$ dan $u \in V_2 - V(T)$ dan tentukan nilai

$$\ell(v) \leftarrow \begin{cases} \ell(v) - m_\ell; & \text{untuk } v \in V_1 \cap V(T) \\ \ell(v) + m_\ell; & \text{untuk } v \in V_2 \cap V(T) \\ \ell(v); & \text{selainnya} \end{cases}$$

Konstruksi G_ℓ yang baru dan kembali ke Langkah 3c.

Contoh:

Misalkan diberikan matriks bobot penugasan pekerja dengan pekerjaan sebagai berikut:

$$\begin{array}{cc} & \text{pekerjaan} \\ \text{pekerja} & \begin{pmatrix} 8 & 3 & 2 & 10 & 5 \\ 10 & 7 & 10 & 6 & 6 \\ 4 & 9 & 4 & 2 & 9 \\ 8 & 10 & 5 & 3 & 3 \\ 9 & 5 & 8 & 5 & 9 \end{pmatrix} \end{array}$$

Tentukan penugasan yang berbobot **maksimum** dengan catatan satu pekerja hanya menangani satu pekerjaan.

Matching Berbobot Minimum

Algoritme Kuhn-Munkres tersebut dapat pula digunakan untuk menentukan *matching* berbobot (misalkan biaya) **minimum** dengan cara:

1. menentukan $m = \max\{w(uv)\}$,
2. mengurangi m dengan bobot sisi, sehingga diperoleh matriks bobot yang baru,
3. gunakan Algoritme 6.2 untuk menentukan bobot maksimum.

Hasil yang diperoleh adalah *matching* berbobot minimum dengan matriks bobot asal.

Ilustrasi:

$$\underset{\text{minimum}}{\overset{\text{bobot } w(uv)}{\begin{pmatrix} 8 & 3 & \boxed{2} \\ \boxed{5} & 7 & 10 \\ \boxed{3} & 4 & 5 \end{pmatrix}}} \xrightarrow{\quad} m = 10 \quad \underset{\text{maksimum}}{\overset{m - w(uv)}{\begin{pmatrix} 2 & 7 & \boxed{8} \\ \boxed{5} & 3 & 0 \\ \boxed{7} & 6 & 5 \end{pmatrix}}}$$

PR

Soal Latihan 6.2 nomor 3, 5, 6