

E.A.6.7 (HC-VIP)

1.1 Modellazione

Dati i parametri $I, V, G = (I, \text{bus})$ t.c.

- $\text{casa} \in I$
- $V \subseteq I / \{\text{casa}\}$
- $\text{bus} \subseteq I \times I$
- $|V| \leq \frac{|I|}{2}$

Si definiscono le variabili

- $P = |I| - 1$ è il numero di *bus* da prendere (o il numero di *passi* da effettuare) per visitare tutti i nodi escluso *casa*
- $\mathcal{P} = \{1, \dots, P\}$
- $\mathcal{I} = I \setminus \{\text{casa}\}$
- $\text{LP} = \{X_i^p \mid i \in \mathcal{I} \wedge p \in \mathcal{P}\}$ l'insieme di lettere proposizionali dove
 - X_i^p è vera se l'indirizzo i è stato visitato al passo p

Il problema si può modellare nel seguente modo

$$\phi = \phi_{\text{ALO_ind}} \wedge \phi_{\text{AMO_ind}} \wedge \phi_{\text{ALO_pass}} \wedge \phi_{\text{AMO_pass}} \wedge \phi_{\text{casa_1}} \wedge \phi_{\text{casa_2}} \wedge \phi_{\text{bus}} \wedge \phi_{\text{VIP}}$$

(ALO) Ad ogni passo viene visitato almeno un indirizzo

$$\phi_{\text{ALO_ind}} = \bigwedge_{p \in \mathcal{P}} \bigvee_{i \in \mathcal{I}} X_i^p \quad (1)$$

(AMO) Ad ogni passo viene visitato al più un indirizzo

$$\phi_{\text{AMO_ind}} = \bigwedge_{\substack{p \in \mathcal{P} \\ i_1, i_2 \in \mathcal{I} \\ i_1 < i_2}} X_{i_1}^p \rightarrow \neg X_{i_2}^p \quad (2)$$

(ALO) Ogni indirizzo viene visitato ad almeno un passo

$$\phi_{\text{ALO_pass}} = \bigwedge_{i \in \mathcal{I}} \bigvee_{p \in \mathcal{P}} X_i^p \quad (3)$$

(AMO) Ogni indirzzo viene visitatao ad al più un passo

$$\phi_{\text{AMO_pass}} = \bigwedge_{\substack{i \in \mathcal{I} \\ p_1, p_2 \in \mathcal{P} \\ p_1 < p_2}} X_i^{p_1} \rightarrow \neg X_i^{p_2} \quad (4)$$

1. Partenza da casa la mattina (se l'indirizzo i **non è raggiungibile** tramite *bus* da *casa*, allora non può essere il primo passo)

$$\phi_{\text{casa_1}} = \bigwedge_{\substack{i \in \mathcal{I} \\ (\text{casa}, i) \notin \text{bus}}} \neg X_i^1 \quad (5)$$

2. Ritorno a casa la sera (se dall'indirizzo i **non si può raggiungere** *casa* tramite *bus*, i non può essere l'ultimo passo)

$$\phi_{casa_2} = \bigwedge_{\substack{i \in \mathcal{I} \\ (i, casa) \notin bus}} \neg X_i^P \quad (6)$$

3. Dal cliente al passo p bisogna usare un *bus* per raggiungere il cliente al passo $p + 1$
- Se al passo p si visita l'indirizzo i , al passo $p + 1$ **non si può raggiungere** l'indirizzo j se non c'è un *bus* da i a j

$$\phi_{bus} = \bigwedge_{\substack{p \in \mathcal{P} \setminus \{P\} \\ i, j \in \mathcal{I} \\ (i, j) \notin bus}} X_i^p \rightarrow \neg X_j^{p+1} \quad (7)$$

4. I clienti VIP devono essere visitati nella prima metà del percorso
- I clienti VIP **non** si possono visitare nella seconda metà del percorso

$$\phi_{VIP} = \bigwedge_{\substack{v \in \mathcal{V} \\ p \in \mathcal{P} \\ p > \lceil \frac{P}{2} \rceil}} \neg X_v^p \quad (8)$$

1.2 Istanziamento

1.2.1 Parametri e variabili

- $I = \{casa, i_2, i_3, i_4, i_5\}$
- $V = \{i_2\}$
- $bus = \{$
 $(casa, i_2), (casa, i_3), (i_2, i_4), (i_2, i_5), (i_3, casa),$
 $(i_3, i_5), (i_4, casa), (i_4, i_5), (i_5, i_2), (i_5, i_3)$
 $\}$
- $P = 4$
- $\mathcal{P} = \{1, 2, 3, 4\}$
- $\mathcal{I} = \{i_2, i_3, i_4, i_5\}$
- LP = $\{$
 $X_{i_2}^1, X_{i_3}^1, X_{i_4}^1, X_{i_5}^1, X_{i_2}^2, X_{i_3}^2, X_{i_4}^2, X_{i_5}^2,$
 $X_{i_2}^3, X_{i_3}^3, X_{i_4}^3, X_{i_5}^3, X_{i_2}^4, X_{i_3}^4, X_{i_4}^4, X_{i_5}^4$
 $\}$

1.2.2 Vincoli

(ALO) Ad ogni passo viene visitato almeno un indirizzo

$$\begin{aligned} \phi_{ALO, ind} = & \\ & (X_{i_2}^1 \vee X_{i_3}^1 \vee X_{i_4}^1 \vee X_{i_5}^1) \wedge (X_{i_2}^2 \vee X_{i_3}^2 \vee X_{i_4}^2 \vee X_{i_5}^2) \wedge \\ & (X_{i_2}^3 \vee X_{i_3}^3 \vee X_{i_4}^3 \vee X_{i_5}^3) \wedge (X_{i_2}^4 \vee X_{i_3}^4 \vee X_{i_4}^4 \vee X_{i_5}^4) \end{aligned}$$

(AMO) Ad ogni passo viene visitato al più un indirizzo

$$\begin{aligned} \phi_{AMO, ind} = & \\ & (X_{i_2}^1 \rightarrow \neg X_{i_3}^1) \wedge (X_{i_2}^1 \rightarrow \neg X_{i_4}^1) \wedge (X_{i_2}^1 \rightarrow \neg X_{i_5}^1) \wedge \\ & (X_{i_3}^1 \rightarrow \neg X_{i_4}^1) \wedge (X_{i_3}^1 \rightarrow \neg X_{i_5}^1) \wedge (X_{i_4}^1 \rightarrow \neg X_{i_5}^1) \wedge \\ & (X_{i_2}^2 \rightarrow \neg X_{i_3}^2) \wedge (X_{i_2}^2 \rightarrow \neg X_{i_4}^2) \wedge (X_{i_2}^2 \rightarrow \neg X_{i_5}^2) \wedge \\ & (X_{i_3}^2 \rightarrow \neg X_{i_4}^2) \wedge (X_{i_3}^2 \rightarrow \neg X_{i_5}^2) \wedge (X_{i_4}^2 \rightarrow \neg X_{i_5}^2) \wedge \\ & (X_{i_2}^3 \rightarrow \neg X_{i_3}^3) \wedge (X_{i_2}^3 \rightarrow \neg X_{i_4}^3) \wedge (X_{i_2}^3 \rightarrow \neg X_{i_5}^3) \wedge \\ & (X_{i_3}^3 \rightarrow \neg X_{i_4}^3) \wedge (X_{i_3}^3 \rightarrow \neg X_{i_5}^3) \wedge (X_{i_4}^3 \rightarrow \neg X_{i_5}^3) \wedge \\ & (X_{i_2}^4 \rightarrow \neg X_{i_3}^4) \wedge (X_{i_2}^4 \rightarrow \neg X_{i_4}^4) \wedge (X_{i_2}^4 \rightarrow \neg X_{i_5}^4) \wedge \\ & (X_{i_3}^4 \rightarrow \neg X_{i_4}^4) \wedge (X_{i_3}^4 \rightarrow \neg X_{i_5}^4) \wedge (X_{i_4}^4 \rightarrow \neg X_{i_5}^4) \end{aligned}$$

(ALO) Ogni indirizzo viene visitato ad almeno un passo

$$\begin{aligned}\phi_{\text{ALO_pass}} = & \\ & (X_{i_2}^1 \vee X_{i_2}^2 \vee X_{i_2}^3 \vee X_{i_2}^4) \wedge (X_{i_3}^1 \vee X_{i_3}^2 \vee X_{i_3}^3 \vee X_{i_3}^4) \wedge \\ & (X_{i_4}^1 \vee X_{i_4}^2 \vee X_{i_4}^3 \vee X_{i_4}^4) \wedge (X_{i_5}^1 \vee X_{i_5}^2 \vee X_{i_5}^3 \vee X_{i_5}^4)\end{aligned}$$

(AMO) Ogni indirzzo viene visitatao ad al più un passo

$$\begin{aligned}\phi_{\text{AMO_pass}} = & \\ & (X_{i_2}^1 \rightarrow \neg X_{i_2}^2) \wedge (X_{i_2}^1 \rightarrow \neg X_{i_2}^3) \wedge (X_{i_2}^1 \rightarrow \neg X_{i_2}^4) \wedge \\ & (X_{i_2}^2 \rightarrow \neg X_{i_2}^3) \wedge (X_{i_2}^2 \rightarrow \neg X_{i_2}^4) \wedge (X_{i_2}^3 \rightarrow \neg X_{i_2}^4) \wedge \\ & (X_{i_3}^1 \rightarrow \neg X_{i_3}^2) \wedge (X_{i_3}^1 \rightarrow \neg X_{i_3}^3) \wedge (X_{i_3}^1 \rightarrow \neg X_{i_3}^4) \wedge \\ & (X_{i_3}^2 \rightarrow \neg X_{i_3}^3) \wedge (X_{i_3}^2 \rightarrow \neg X_{i_3}^4) \wedge (X_{i_3}^3 \rightarrow \neg X_{i_3}^4) \wedge \\ & (X_{i_4}^1 \rightarrow \neg X_{i_4}^2) \wedge (X_{i_4}^1 \rightarrow \neg X_{i_4}^3) \wedge (X_{i_4}^1 \rightarrow \neg X_{i_4}^4) \wedge \\ & (X_{i_4}^2 \rightarrow \neg X_{i_4}^3) \wedge (X_{i_4}^2 \rightarrow \neg X_{i_4}^4) \wedge (X_{i_4}^3 \rightarrow \neg X_{i_4}^4) \wedge \\ & (X_{i_5}^1 \rightarrow \neg X_{i_5}^2) \wedge (X_{i_5}^1 \rightarrow \neg X_{i_5}^3) \wedge (X_{i_5}^1 \rightarrow \neg X_{i_5}^4) \wedge \\ & (X_{i_5}^2 \rightarrow \neg X_{i_5}^3) \wedge (X_{i_5}^2 \rightarrow \neg X_{i_5}^4) \wedge (X_{i_5}^3 \rightarrow \neg X_{i_5}^4)\end{aligned}$$

1. Partenza da casa la mattina (se l'indirizzo i **non è raggiungibile** tramite *bus* da *casa*, allora non può essere il primo passo)

$$\phi_{\text{casa}_1} = \neg X_4^1 \wedge \neg X_5^1$$

2. Ritorno a casa la sera (se dall'indirizzo i **non si può raggiungere** *casa* tramite *bus*, i non può essere l'ultimo passo)

$$\phi_{\text{casa}_2} = \neg X_2^4 \wedge \neg X_5^4$$

3. Dal cliente al passo p bisogna usare un *bus* per raggiungere il cliente al passo $p + 1$

- Se al passo p si visita l'indirizzo i , al passo $p + 1$ **non si può raggiungere** l'indirizzo j se non c'è un *bus* da i a j

$$\begin{aligned}\phi_{\text{bus}} = & \\ & (X_2^1 \rightarrow \neg X_2^2) \wedge (X_2^1 \rightarrow \neg X_3^2) \wedge (X_3^1 \rightarrow \neg X_2^2) \wedge (X_3^1 \rightarrow \neg X_3^2) \wedge \\ & (X_3^1 \rightarrow \neg X_4^2) \wedge (X_4^1 \rightarrow \neg X_2^2) \wedge (X_4^1 \rightarrow \neg X_3^2) \wedge (X_4^1 \rightarrow \neg X_4^2) \wedge \\ & (X_5^1 \rightarrow \neg X_4^2) \wedge (X_5^1 \rightarrow \neg X_5^2) \wedge (X_2^2 \rightarrow \neg X_2^3) \wedge (X_2^2 \rightarrow \neg X_3^3) \wedge \\ & (X_3^2 \rightarrow \neg X_2^3) \wedge (X_3^2 \rightarrow \neg X_3^3) \wedge (X_3^2 \rightarrow \neg X_4^3) \wedge (X_4^2 \rightarrow \neg X_2^3) \wedge \\ & (X_4^2 \rightarrow \neg X_3^3) \wedge (X_4^2 \rightarrow \neg X_4^3) \wedge (X_5^2 \rightarrow \neg X_4^3) \wedge (X_5^2 \rightarrow \neg X_5^3) \wedge \\ & (X_2^3 \rightarrow \neg X_2^4) \wedge (X_2^3 \rightarrow \neg X_3^4) \wedge (X_3^3 \rightarrow \neg X_2^4) \wedge (X_3^3 \rightarrow \neg X_3^4) \wedge\end{aligned}$$

$$\begin{aligned}
& (X_3^3 \rightarrow \neg X_4^4) \wedge (X_4^3 \rightarrow \neg X_2^4) \wedge (X_4^3 \rightarrow \neg X_3^4) \wedge (X_4^3 \rightarrow \neg X_4^4) \wedge \\
& (X_5^3 \rightarrow \neg X_4^4) \wedge (X_5^3 \rightarrow \neg X_5^4) \wedge (X_2^4 \rightarrow \neg X_2^5) \wedge (X_2^4 \rightarrow \neg X_3^5) \wedge \\
& (X_3^4 \rightarrow \neg X_2^5) \wedge (X_3^4 \rightarrow \neg X_3^5) \wedge (X_3^4 \rightarrow \neg X_4^5) \wedge (X_4^4 \rightarrow \neg X_2^5) \wedge \\
& (X_4^4 \rightarrow \neg X_3^5) \wedge (X_4^4 \rightarrow \neg X_4^5) \wedge (X_5^4 \rightarrow \neg X_4^5) \wedge (X_5^4 \rightarrow \neg X_5^5)
\end{aligned}$$

4. I clienti VIP devono essere visitati nella prima metà del percorso
- I clienti VIP **non** si possono visitare nella seconda metà del percorso

$$\phi_{\text{VIP}} = \neg X_2^3 \wedge \neg X_2^4$$

1.3 Codifica (a questo giro in Rust)

```
use crate::encoder::*;
use serde::Serialize;

#[derive(Clone, Copy, Hash, PartialEq, Eq, PartialOrd, Ord,
Serialize, Debug)]
pub struct X(usize, usize);

pub fn encode_instance(
    addresses: usize, // esclusa "casa"
    buses: &[(usize, usize)],
    vips: &[usize],
) → (String, Vec<X>) {
    use Literal::Neg;
    let steps = addresses;

    let mut encoder = EncoderSAT::new();

    // ALO_ind
    for p in 1..=steps {
        encoder.add((2..=addresses + 1).map(|i| X(i,
p).into()).collect())
    }

    // AMO_ind
    for p in 1..=steps {
        for i1 in 2..=addresses + 1 {
            for i2 in i1 + 2..addresses + 1 {
                encoder.add(vec![Neg(X(i1, p)), Neg(X(i2, p))])
            }
        }
    }

    // ALO_pass
    for i in 2..=addresses + 1 {
        encoder.add((1..=steps).map(|p| X(i,
p).into()).collect());
    }

    // AMO_pass
    for i in 2..=addresses + 1 {
        for p1 in 1..=steps {
            for p2 in p1 + 1..=steps {
                encoder.add(vec![Neg(X(i, p1)), Neg(X(i, p2))]);
            }
        }
    }

    // casa_1
    for i in 2..=addresses + 1 {
        if !buses.contains(&(1, i)) {
            encoder.add(vec![Neg(X(i, 1))]);
        }
    }
}
```

```

    }
}

// casa_2
for i in 2..=addresses + 1 {
    if !buses.contains(&(i, 1)) {
        encoder.add(vec![Neg(X(i, steps))]);
    }
}

// bus
for p in 1..steps {
    for i in 2..=addresses + 1 {
        for j in 2..=addresses + 1 {
            if !buses.contains(&(i, j)) {
                encoder.add(vec![Neg(X(i, p)), Neg(X(j, p +
1))]);
            }
        }
    }
}

// VIP
for &v in vips {
    for p in steps.div_ceil(2) + 1..=steps {
        encoder.add(vec![Neg(X(v, p))]);
    }
}

encoder.end()
}

```