Reuven Bar-Yehuda and Dror Rawitz

On the Equivalence Between the Local Ratio Technique and the Primal-Dual Schema
Overview

Definitions and Background

The Local-Ratio Technique

Approximation using LP Duality

The Primal-Dual Schema

Examples: GHS, PVS, s-t cut, Min-2SAT, Interval Scheduling

Fractional Primal-Dual

Practical Primal-Dual [BR04b]
Definitions are similar for maximization problems

\( \{ f \} = \{ x : f \} \)

Abuse notation by treating \( x \) as

Remarks:

\( \mathcal{F} \not\in \{ f \} \setminus x \quad x \in \mathcal{F} \quad \forall x \in \mathcal{F} \)  \( \mathcal{F} \supseteq \mathcal{A} \quad x \cdot \mathcal{L} \quad x \cdot \mathcal{L} \quad \mathcal{F} \sqsubseteq x \cdot \mathcal{L} \)

\( \mathcal{F} \sqsubseteq x \quad x \cdot \mathcal{L} \quad \mathcal{F} \sqsubseteq x \cdot \mathcal{L} \)

\( x \equiv \) optimal

\( x \) is minimal

\( x \) is \( r \)-approximate

\( x \) is \( r \)-approximate

\( \min \) solution

weight function

set of constraints on \( x \) \( \in \{ 0, 1 \} \)

\( \mathcal{F} \quad (m, \mathcal{F}) \quad \) Minimization Problem

\( w \)
Generalized Hitting Set

Instance:
\[ S = \{ s_1, \ldots, s_m \} \]

Solution:
\[ \Omega \cup U_0 \]
\[ s_0 \in S \]

Measure:
\[ (s)m \bigcap_{s \in S} + (n)m \bigcap_{s \in \Omega} \]

Remarks:
- We must pay for sets that are not hit by \( \Omega \)

We get \( \exists \Omega \), we get generalized vertex cover
- We get \( \infty \), we get hitting set

Generalized Hitting Set
Generalized Hitting Set

Example:

\[ U = \{f_1, f_2, f_3, f_4, f_5, g_8\} \]

\[ S = \{f_1, f_2, f_3, f_4, f_5, g_8\} \]

Feasible solution:

\[ U_0 = \{f_2, f_3\} \]
\[ S_0 = \{f_2, f_3, f_5\} \]

Minimal solution:

\[ U_0 = \{f_2, f_3\} \]
\[ S_0 = \{f_2, f_3, f_5\} \]

Optimal solution:

\[ U_0 = \{f_2, f_3\} \]
\[ S_0 = \{f_2, f_3, f_5\} \]

Observation:

\[ \forall s \in S \in \mathcal{A} \quad \max_{s \in S} \frac{|S|}{|s|} \geq |S \cup \{s\}| + |\mathcal{L} \cup s| \]

\[ \emptyset = \mathcal{L} \cup s \quad \text{then} \quad \mathcal{S} \cap \{\mathcal{L}, \mathcal{S}\} (\mathcal{S}, \mathcal{L}) \]

Optimal solution:

\[ \{\{3, 3\}\} = \mathcal{S} \quad \text{and} \quad \{2\} = \mathcal{L} \]

Minimal solution:

\[ \emptyset = \mathcal{S} \quad \text{and} \quad \{3, 3\} = \mathcal{L} \]

Feasible solution:

\[ \{\{3, 3\}\} = \mathcal{S} \quad \text{and} \quad \{3, 3\} = \mathcal{L} \]

Example:

Generalized Hitting Set
Primal-Dual Schema

Local Ratio Technique

Related Work

\[ \text{FVS/Min. Sol.} \Rightarrow \text{FVS} \]

\[ \text{Network Design/Min. Sol.} \Rightarrow \text{VC} \]

\[ \text{Local Ratio Technique} \]

\[ \text{Primal-Dual Schema} \]
Related Work

PDSurvey\textsuperscript{[Wil02]} \quad \begin{align*}
\text{LR Survey} &\quad \text{PD Survey} \\
\text{Scheduling split intervals} &\quad \text{with demands}
\end{align*}

\text{Fractional LR} \quad \Rightarrow \quad \text{Fractional PD} \text{[BR04b]}

\text{Dual and Primal Fitting} \text{[FR03]} \quad \Leftrightarrow \quad \text{Dual Fitting} \text{[CHV79,J+03]}

\text{Dual Fitting} \text{[CHV79,J+03]} \quad \Leftrightarrow \quad \text{Relaxed Primal} \text{[JV01]}

\text{Relaxed Primal} \text{[JV01]} \quad \equiv \quad \text{Negative Weights} \text{[BR04a]}

\text{Negative Weights} \text{[BR04a]} \quad \equiv \quad \text{Local Ratio}

\text{Local Ratio} \quad \equiv \quad \text{Pimal-Dual}

\text{Pimal-Dual}
The Local Ratio Technique

We pay \( r \cdot q \), and the optimum drops by at least \( q \).

Breakdown of \( w \) and \( b \) is determined by algorithm.

Solution is constructed such that breakdown exists.

\[
\begin{align*}
\mathbf{1} \cdot \mathbf{w} \leq x \mathbf{m} & \iff \\
\mathbf{B} \cdot \mathbf{r} \geq x \mathbf{m} & \forall \mathbf{A} \\
\mathbf{B} \leq \mathbf{B}_{\text{Opt}} & \forall \mathbf{A} \\
\mathbf{B} + \cdots + \mathbf{B} = \mathbf{B} \\
\mathbf{n} + \cdots + \mathbf{n} = \mathbf{n}
\end{align*}
\]

Local Ratio:

\[
\begin{align*}
\mathbf{1} \cdot \mathbf{w} \leq x \mathbf{m} & \iff \\
\mathbf{B} \cdot \mathbf{r} \geq x \mathbf{m} & \mathbf{B} \leq \mathbf{B}_{\text{Opt}}
\end{align*}
\]

Usually:

The Local Ratio Technique
The Local Ratio Technique

Local Ratio Theorem:

Let \( w, w_1 \) and \( w_2 \) be functions s.t. \[ w = w_1 + w_2 \].

\[ x \text{ is an } r \text{-approx w.r.t. } (F, w_1) \text{ and } (F, w_2) \text{ if and only if } \exists \varphi \geq 0 \text{ such that } \varphi' \cdot r \geq x \cdot \varphi \geq \varphi' \]

Intuition: Find a good weight function \( w_1 \), and solve recursively for \( w_2 \).

Definition: \( w_1 \) is called \( r \)-effective if \( \forall \varphi \exists \text{ such that } \varphi \cdot r \geq x \) minimal solution \( x \).

\[ m = m_1 - m_2 = \text{find a good weight function } m_1, \text{ and solve recursively for } m_2 \]

Intuition: Let \( m, m_1, m_2 \) and \( \varphi \) be functions s.t. \( m = m_1 + m_2 \).

Local Ratio Technique: 

The Local Ratio Technique
Let \( s \subseteq S \), and define:

\[
\{ \{ s \} \cap s \subseteq t \mid \exists \varepsilon \}
\]

\( = (t)^1 \mathcal{m} \) and denote:

Generalized Hitting Set - Local-Ratio

Observation:

A minimal solution \( x \) is \( \varepsilon \)-effective if

\[
3 \cdot (|s| \max_{s \in S \supseteq \varepsilon} = \nabla)
\]

or otherwise:

\[
\{ \} \cup \forall s \in t \ni \varepsilon
\]

\( = (\varepsilon)^1 \mathcal{m} \)
Generalized Hitting Set - Local-Ratio

Let \( S \subseteq U \), and define:

\[
\varphi_1(t) = \begin{cases} 
\|S\| & \text{if } \|S\| \leq \min \{ t \subseteq n : (n)m \} \\
\min \{ t \subseteq n : (n)m \} & \text{otherwise}
\end{cases}
\]

-approxAlgorithm:

**Algorithm GHS** \((U, S, w)\):

1. If \( S = \emptyset \), return \((U, S, w)\).
2. Choose a set \( s \in S \).
3. Let be \((S', \Omega)\) after the removal of \( t \).
4. \((s)m \cap \{ s \subseteq n : (n)m \} \) min \( t \to \Omega \).
5. \((s)m \cap \{ s \subseteq n : (n)m \} \) min \( \Omega \to \emptyset \).
6. \((S, \Omega) \to (S', \Omega) \) if feasible, else go to 7.
7. Return \((S', \Omega) \).
8. Return \((U, S, w) \).

\( \forall \)-approx Algorithm

Let \( S \subseteq U \) and define:

\[
\varphi_1(t) = \begin{cases} 
0 & \text{if } S = \emptyset \\
\|S\| & \text{otherwise}
\end{cases}
\]

\( \forall \)-approx Algorithm
Algorithm GHS2 $(U, S, w)$:

1. For each set $s \in S$:
   
   
   \[
   \min_{u \in s} w(u)
   \]
   
   2. For each set $s \in S$
   
   \[
   \min_{u \in s} w(u) > 0
   \]
   
   3. Return
   
   \[
   m - m \rightarrow m
   \]

Remark:

Generalization of $\bigtriangleup$-approximation algorithm for hitting set

\[
\{0 < (n) \wedge s \in n \wedge s = n\} = \bigtriangleup \bigcap \{0 = (n) \wedge n = \bigtriangleup\}
\]

\[
\{s \in n : (n) \wedge \min_{u \in s} w(u) \rightarrow \infty\}
\]

Generalized Hitting Set - Local-Ratio

An iterative version of the $\bigtriangleup$-approximation algorithm.
The Local Ratio Technique

Usually, local ratio algorithms are recursive and reveal intuition. They are elegant!

- are elegant
- reveal intuition
- construct a new weight function in each iteration
- use reverse deletion (solution correction)
- are recursive

The Local Ratio Technique
Instance: digraph $G = (V; E)$, $s; t \in V$.

Solution: $F \subseteq E$ s.t. there is no path $s \rightarrow t$ in $(G \setminus E, \Lambda)$.

Measure: \[
\sum_{e \in E} m
\]

Observation 1: Minimal solution = Cut

Observation 2: $s \not\rightarrow t$ times $S \leftarrow d$ $S$

$s \not\rightarrow t$ times $S \leftarrow d$ $S$
Conclusion: \( \varrho \) is \( t \)-effective

\[
I = (1 - \eta) - \eta = (\exists) \varrho \quad \bigwedge_{(S', S) \in \theta} \quad \text{For any } (S', S) \in \theta
\]

\[
\begin{cases}
0, & \text{otherwise}
\end{cases}
\]

\[
\begin{cases}
(s \leftarrow_d t) \quad \varrho \in \theta \quad 1, \\
\varrho \in \theta \quad 1, \\
\varrho \in \theta \quad I
\end{cases}
\]

Minimum \( s \)-cut
Algorithm FF:

1. Let $F = f_{e: w(e) = 0}$

2. If $F$ is feasible:
   - Return a minimal subset of $F$

3. Let $P$ be a path from $s$ to $t$ in $(V, E)$

4. $\forall m \in P, e \in E$
   - $\min_e \{w(e) : e \in P\}$

5. Return $\text{ FF}(w, P)$

Remarks:
- $w$ capacities of residual graph
- First LR analysis which uses negative weights
- Also: preflow-push and assignment problem [BR04a]

Ford & Fulkerson’s Algorithm
Using Linear Programming for Approximation

- Primal-dual based algorithms
- Rounding: solve $\text{P}$ and then round solution

Techniques:

Idea: Find $x$ such that $wx \geq \text{Opt(P)} \cdot \text{Opt(\text{P})}$

Integrality gap $\max \text{Opt(\text{IP})}/\text{Opt(\text{P})} \geq \text{Opt(\text{IP})}/\text{Opt(\text{P})}$

$\text{P}$ is the LP-relaxation of $\text{IP}$

**Techniques:**

- Rounding: solve $\text{P}$ and then round solution
- Primal-dual based algorithms

Idea: Find $x$ such that $wx \geq \text{Opt(P)}$
\[
16 = 9 + 10 \leq (3x - 2x \in + x \in) + (3x \in + 2x - x \in) \leq 3x \in + 2x + x \in
\]

A better lower bound:

\[
10 \leq 3x \in + 2x - x \in \leq 3x \in + 2x + x \in
\]

A lower bound:

\[
0 \leq 3x \in + 2x - x \in, 1 \in
\]

\[
9 \leq 3x - 2x \in + x \in
\]

\[
0 \leq 3x \in + 2x - x \in \quad \text{s.t.}
\]

\[
\min \quad 3x \in + 2x + x \in
\]

LP Duality - Example
LP Duality - Example

Lower bound is $10y_1 + 6y_2$.

\[ 3x_1 + 2x + x_3 \geq (3x - 2x_2 + x_4) y_1 + (3x + 3x_1 - x_2) y_2 \]

such that

Assign a non-negative coefficient $y_i$ to every primal inequality

\[ 0 \leq \varepsilon x_1, x_2, x_3 \]

\[ 6 \geq \varepsilon x - 2x_2 + x_4 \]

s.t.

\[ 10 \geq \varepsilon x_3 + 2x - x_1 \]

min

\[ \varepsilon x_2 + 2x + x_1 \]
The problem of finding the best lower bound can be formulated as a linear program.

**Primal**

\[
	ext{min } 7x_1 + 3x_2 + 5x_3 \\
\text{s.t. } x_1 + 3x_2 + 2x_3 \leq 10 \\
\quad x_1 \geq 0, x_2 \geq 0, x_3 \geq 0
\]

**Dual**

\[
\text{max } 0 \geq 7y_1 + 5y_2 + 6y_3 \\
\text{s.t. } y_1 + 5y_2 + 2y_3 \leq 7 \\
\quad 3x_1 + 2x_2 + x_3 \geq 0
\]
LP Duality

\[ \min \quad P \quad \sum_{j=1}^{n} w_j x_j \]
\[ \text{s.t.} \quad P \quad \sum_{i=1}^{p} a_{ij} x_j \leq b_i \quad \forall i \]
\[ x_j \geq 0 \quad \forall j \]

\[ \max \quad \forall j \quad w_j \]
\[ \text{s.t.} \quad \sum_{i=1}^{m} a_{ij} y_i \leq b_j \quad \forall j \]
\[ y_i \geq 0 \quad \forall i \]

Or, in a compact formulation:

\[ \forall x, y : \quad w^T x \leq b^T y \]

\[ \text{Thus, } \text{Opt}(\text{Primal}) \geq \text{Opt}(\text{Dual}) \]

\[ \text{The dual of the dual is the primal.} \]

\[ \text{For every } x \text{ and } y : \quad w^T x \leq b^T y \]

\[ \text{Thus, } \text{Opt}(\text{Primal}) = \text{Opt}(\text{Dual}). \]

\[ \forall x, y : \quad w^T x \leq b^T y \]

\[ \text{Opt}(\text{Primal}) = \text{Opt}(\text{Dual}). \]
\[ \text{Max Flow vs. Min s-t cut:} \] 

\[ \text{Opt(Dual)} = \text{Opt(Min s-t cut).} \]

\[ \text{When } x^e \in \{0,1\} \text{ w.e } \text{get Min s-t cut.} \]

\[ \text{Implicit } x^e \geq 1. \]

\[ \forall e \in E \quad 0 \leq x^e \]

\[ \forall P \in \text{Path}(s,t) \quad 1 \geq \sum_{e \in P} x^e \geq \sum_{e \in P} f^e \]

\[ \text{Dual:} \quad \min \quad \sum_{e \in E} w^e x^e \]

\[ \text{Max Flow:} \quad \max \quad \sum_{e \in p} f^e \]

\[ \text{Min Flow vs. Min s-t cut} \]
\[ q = \sum_{j=1}^{n} w_j x_j \quad \text{s.t.} \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \]

\[ m = \sum_{i=1}^{m} w_i y_i \quad \text{s.t.} \quad \sum_{i=1}^{m} a_{ij} y_i \leq w_j \]

**Dual Conds.:**

\[ 0 < q, \quad A \]

\[ 0 < m, \quad w \]

**Primal Conds.:**

\[ x \text{ and } y \text{ are optimal} \]

**Complementary Slackness Conditions:**

\[ \begin{align*}
    q & = \sum_{j=1}^{n} w_j x_j \quad \text{s.t.} \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \\
    m & = \sum_{i=1}^{m} w_i y_i \quad \text{s.t.} \quad \sum_{i=1}^{m} a_{ij} y_i \leq w_j
\end{align*} \]

**Strong Duality Theorem:**

\[ q^* = \max_{x \in P} q \quad \text{and} \quad m^* = \min_{y \in D} m \]

**Weak Duality Theorem:**

\[ q^* \geq \max_{x \in P} q \quad \text{and} \quad m^* \leq \min_{y \in D} m \]

**Dual:** Packing of Primal constraints in order to get a lower bound

**LP Duality - Summary**
Problem: Find an integral primal solution $x$ and a dual solution $y$ such that:

$w^T x - b^T y \geq 0$

By the Weak Duality Theorem:

$w^T x \geq \inf \{ q : q \geq 0, \quad x^T q \geq 1 \}$

Question: How do we find such solutions?

Idea: Find an integral primal solution $x$.
Primal-Dual Algorithms

Idea: Find an integral primal solution \( x \) and a dual solution \( y \) that satisfy:

**Relaxed Primal:**
\[
\bar{w}_j \geq 0 \\
\sum_{i=1}^{m} a_{ij} y_i \geq \bar{r}_1 \\
\sum_{j=1}^{n} x_j \geq \bar{r}_2 \\
\sum_{j=1}^{n} w_j x_j = \sum_{i=1}^{m} r_i y_i = \sum_{j=1}^{n} r_j x_j
\]

**Relaxed Dual:**
\[
\sum_{i=1}^{m} b_i \geq 0 \\
\sum_{j=1}^{n} a_{ij} x_j \geq \sum_{i=1}^{m} b_i \\
\sum_{j=1}^{n} w_j x_j = \sum_{i=1}^{m} r_i y_i = \sum_{j=1}^{n} r_j x_j
\]

Remark: \( \bar{r}_1 = \bar{r}_2 = \bar{r} \) in primal-dual schema.

Question: How do we find such solutions?

In this case:
\[
\bar{w}_j \geq 0 \\
\sum_{i=1}^{m} a_{ij} y_i \geq \bar{r}_1 \\
\sum_{j=1}^{n} x_j \geq \bar{r}_2 \\
\sum_{j=1}^{n} w_j x_j = \sum_{i=1}^{m} r_i y_i = \sum_{j=1}^{n} r_j x_j
\]

Relaxed Dual:
\[
\sum_{i=1}^{m} b_i \geq 0 \\
\sum_{j=1}^{n} a_{ij} x_j \geq \sum_{i=1}^{m} b_i \\
\sum_{j=1}^{n} w_j x_j = \sum_{i=1}^{m} r_i y_i = \sum_{j=1}^{n} r_j x_j
\]

Relaxed Primal:
\[
\sum_{i=1}^{m} b_i \geq 0 \\
\sum_{j=1}^{n} a_{ij} x_j \geq \sum_{i=1}^{m} b_i \\
\sum_{j=1}^{n} w_j x_j = \sum_{i=1}^{m} r_i y_i = \sum_{j=1}^{n} r_j x_j
\]

Find an integral primal solution \( x \) and a dual solution \( y \) that satisfy:

Idea: Primal-Dual Algorithms
In each advancement step we use only "good" constraints:

\[ \sum_{i=1}^{n} y_i = \sum_{j=1}^{m} a_{ij} x_j \]

\[ \Rightarrow \quad \sum_{j=1}^{m} x_j = \sum_{i=1}^{n} x_i \]

\[ = \quad \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} y_i \]

\[ \quad \text{i.e., } y \text{ is changed such that the relaxed dual conditions are satisfied.} \]

\[ q \cdot x \geq \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} y_i \geq q \quad \Rightarrow \quad 0 < \hat{y} \]

\[ \text{The Primal-Dual Schema} \]

\[ \text{i.e., } x \text{ contains elements that are "paid for", i.e., } x \text{ contains elements that are } \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} y_i \]

\[ = \quad \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} y_i \]

\[ \quad \text{i.e., } y \text{ is changed such that the relaxed dual conditions are satisfied.} \]

\[ q \cdot x \geq \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} y_i \geq q \quad \Rightarrow \quad 0 < \hat{y} \]

\[ \text{In each advancement step we use only "good" constraints:} \]

\[ \text{The Primal-Dual Schema} \]
1. Primal: \( f \in x \) only if dual constraint is tight

2. Relaxed Dual: \( 0 \preceq \bar{f} \) only if primal constraint is "good"

3. In each iteration we raise \( s \bar{f} \) until some constraint becomes tight.

**Analysis:**

8. Return \( x \)

7. \( \{t\} \ \text{is feasible then } \{t\} \ \text{is feasible} \)

6. For each \( t \in T \)

5. \( \{t\} \cap x \rightarrow x \)

4. \( \{s\} \in \mathcal{S} \) \( \begin{cases} n_0 \in n : s \bar{f} \{s\} \in \mathcal{S} \end{cases} - (n)m \) min \( \rightarrow t \)

3. For each set \( s \in \mathcal{S} \):

2. \( \emptyset \rightarrow x, 0 \rightarrow f \)

1. \( \nabla \)-approximation Algorithm (\( \nabla \)-type \( [\text{G}W97] \))
The Primal-Dual Schema

- Algorithm may construct new valid inequalities
- In [BT98], algorithm may construct new valid inequalities
- In each iteration: several dual variables are raised until some constraint becomes tight
- Reversal deletion
- A violation oracle decides which variables to raise
- A single dual variable is changed in each iteration
- Changes weights

The Primal-Dual Schema
Definition: An inequality is called \( r \)-effective if for any minimal solution \( x \), \( x \geq \varepsilon \).

Remark: Weight updates no need to remember the dual variables.

\[
\{0 < (n)m \mid s \in nA : s\} = S \text{ and } \{0 = (n)m : n\} = \bigwedge
\]

Approx Algorithm:

1. For each set \( s \in S \),
2. \( \{s\} \cap \{s \in n : (n)m\} \text{ min} \to \{s\} \)
3. \( s \text{ } - (n)m \to (n)m \)
4. \( s \text{ } - (s)m \to (s)m \)
5. Return \( \{u : w(u) = 0\} \) and \( S_0 = \{s : \exists u \in s; w(u) > 0\} \).

Observation: \( (s)e \leq \sum_{s \in n} x_s \), if \( x \geq \varepsilon \).

\( \forall \cdot \varepsilon \geq ax \geq \varepsilon \)
\( a \text{ is } \rho\text{-effective (with } \rho \text{ as a witness)} \)

\[ \equiv \]

\( a \text{ is } \rho\text{-effective} \implies a \rho \geq ax \)

**Lemma:**

\[ \{s\} \cap s \in \tau, 1, \text{ otherwise} \]

\[ (t)g \]

\[ (e = s\bar{h}) \]

\( I \preceq s x + n x \overset{s \in n}{\lor} \)

**Example:**

\( \exists x \text{ minimal solution } x \)

\( \text{A \ minimal solution } x \text{ such that } \rho \in \mathcal{E} \)

\[ + \mathbb{R} \leftarrow \{u_1, \ldots, u_I\} : \rho \]

\( \rho' \triangleright x \quad \rho \text{-effective} \)

\( \rho' \triangleright x \quad \rho \text{-effective} \)

**Definition:**

\( \text{Valid Inequalities vs. Weight Functions} \)
Equivalence

[BGW97] primal-dual algorithm based on hitting set problem

[BT98] primal-dual algorithm for covering problems (uses inequalities)

[Bar00] local ratio algorithm

Our Results [BR01b]:

Frameworks that extend [BT98] and [Bar00]

[BT98] \subseteq [CW97] \subseteq [GW97]

Global analysis vs. local analysis

Equivalence is constructive

Frameworks are equivalent

Integrality gap – bounds approximation ratio of LR algorithms

Equivalence

Bar00] local ratio algorithm. For covering problems

[BT98] primal-dual algorithm. For covering problems (uses inequalities)

[CW97] primal-dual algorithm based on the hitting set problem

Equivalence

31
Minimum s-t cut

Instance: digraph $G = (V; E)$, $s; t \in V$, edge capacities $w : E \to \mathbb{R}^+$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Measure: edge capacities $w : E \to \mathbb{R}^+$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V$. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digraph $C : (V; \lambda E) = C$, $s; t \in V. Solution: digra
Minimum $s$-$t$ cut

\[
I = \begin{cases}
\varnothing \in \mathcal{P} & \text{if } (S, S) \in \varnothing \\
0 & \text{otherwise}
\end{cases}
\]

**Remarks:**

PD schema analysis which uses negative coefficients

PD schema analysis of a non-dual ascent algorithm

\[
I = \sum_{x \in \mathcal{P}} \varnothing - \sum_{x \in \mathcal{P}} \varnothing

I = (I - \varnothing) - \varnothing = (\varnothing) \in \mathcal{P}
\]
We do not need to know the value of $\ell$. (\begin{align*}
\ell + |\Lambda| - |\mathcal{E}| & \geq a(x(n)) \mathcal{E} \subseteq \lambda \mathcal{F} \\
& \text{and (a) if } \lambda \mathcal{E} \subseteq \mathcal{F} \text{ then } \lambda \mathcal{F} \subseteq \mathcal{E} \\text{effective}
\end{align*})

Conclusion:

where $\ell$ is the size of a minimum cardinality FVS

$(\ell + |\Lambda| - |\mathcal{E}|) \geq (a) \mathcal{E} \subseteq \lambda \mathcal{F} > \ell + |\Lambda| - |\mathcal{E}|$

If $\Lambda \subseteq \mathcal{F}$ and $\mathcal{F}$ is a minimal FVS then

Proposition ([BG94],[CGHW98]):

(a)$m \mathcal{F} \subseteq \lambda \mathcal{F}$

Measure:

Solution: (\begin{align*}
\mathcal{F} \setminus \Lambda \cup & \text{ s.t. } \Lambda \subseteq \mathcal{F} \\
\Lambda & \text{ is acyclic}
\end{align*})

Instance: $\mathcal{F}$

Feedback Vertex Set
**Algorithm (CDZ01):**

(a) $m \in \mathbb{N}$

Measure: $\ell \subseteq \mathcal{H}$ is acyclic

Solution: $(\mathcal{A} \setminus \Lambda) \subseteq \mathcal{H}$ s.t. $\Lambda \supseteq \mathcal{H}$

Instance: Tournament $\mathcal{G}$ such that $(\forall, \Lambda) = \mathcal{G}$

Feedback Vertex Set in Tournaments

**This work [BR01b]:**

1. 2.5-approx LR phase that disposes of forbidden sub-tournaments.
2. Finds an optimal solution in tournaments without forbidden sub-tournaments.

A primal-dual analysis of this algorithm.

Equivalence continues to hold.
Minimum SAT

Instance: 2CNF formula

Solution: assignment that satisfies

Measure: assignment that satisfies

Solution: 2CNF formula

Minimum 2SAT
Single resource

A set of activities \( A_1, \ldots, A_m \)

An instance \( I \in A_i \) has \([s(I), e(I))\) and \( p(I) \geq 0 \)

A feasible schedule:

1. at most one instance of every activity
2. at most one instance for all time instants \( t \).

\[
\begin{align*}
\text{max} \quad & \sum_{I} w_I x_I \\
\text{s.t.} \quad & \sum_{I: I \in A_i} x_I \leq 1 \quad \forall i \\
& \sum_{I: s(I) \leq t < e(I)} x_I \leq 1 \quad \forall t \\
& x_I \in \{0, 1\} \quad \forall I
\end{align*}
\]
is 2-effective

\[ (r)I \subseteq I(x) \]

is 2-effective

\[ (r)I \cap (r)A \forall I \in I: \exists \theta = (I)^1d \]

otherwise \[ 0, \quad \] \[ I \]

\[ (r)I \cap (r)A \forall I \in I: \exists \theta = (I)^1d \]

\( f \) set of instances intersecting

\( f \) activity to which \( f \) belongs

\( f \) instance with minimum end-time

Interval Scheduling

2-approximation algorithm

BBFNS00
Let $x$ be a solution of LP-relaxation. We find integral $x$ and $y$ such that $x \cdot y \geq m \cdot n$. $y$ is a solution to dual of LP-relaxation. Let $P'$ be the set of above constraints satisfied by $x$. The primal constraints $x \cdot y \geq m \cdot n$ such that $y$ is not a solution to dual of LP-relaxation. We find integral $x$ and $y$ such that $x \cdot y \geq m \cdot n$. Let $x^*$ be an optimal solution of LP-relaxation. New extension of primal-dual schema:
Algorithms that relax primal conditions (e.g., $\rho > 1$), and algorithms that use dual fitting can be interpreted combinatorially [FR03].

Remarks:

- Global analysis needs certain conditions (e.g., for clean-up phase).
- Algorithm that relax primal conditions (i.e., $\rho > 1$).

<table>
<thead>
<tr>
<th>Local analysis</th>
<th>Global analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight function</td>
<td>Valid inequality</td>
</tr>
<tr>
<td>Local Ratio Theorem</td>
<td>LP-Duality</td>
</tr>
<tr>
<td>Local Ratio</td>
<td>Dual-Primal</td>
</tr>
</tbody>
</table>

Analysis: Global vs. Local
Our framework extends known frameworks for covering problems. Constructive equivalence

Scope:

- Fractional
- Non-boolean
- Maximization problems
- Negative weights/coefficients
- Clean-up phase
- Covering and non-covering

Conclusion