Partial vs. Complete Domination: t-DOMINATING SET

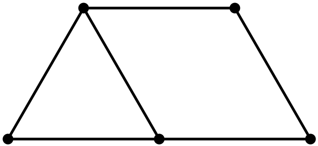
Joachim Kneis Daniel Mölle Peter Rossmanith

Theory Group, RWTH Aachen University

January 22, 2007

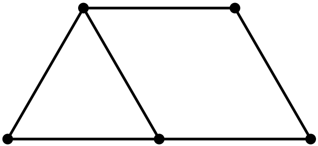
Complete vs. Partial Solutions

VERTEX COVER (VC):



Cover all edges.

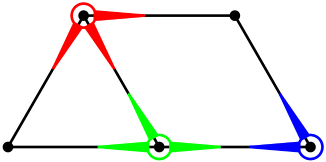
DOMINATING SET (DS):



Dominate all nodes.

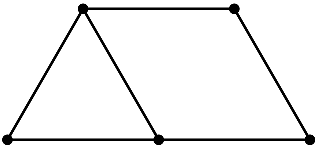
Complete vs. Partial Solutions

VERTEX COVER (VC):



Cover all edges.

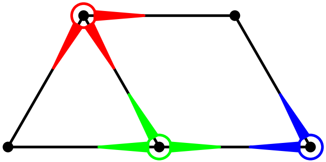
DOMINATING SET (DS):



Dominate all nodes.

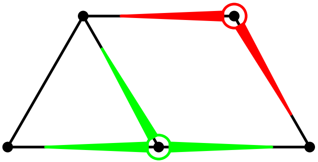
Complete vs. Partial Solutions

VERTEX COVER (VC):



Cover all edges.

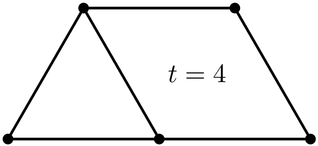
DOMINATING SET (DS):



Dominate all nodes.

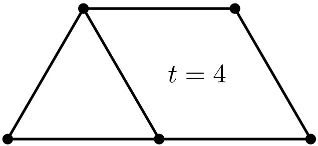
Complete vs. Partial Solutions

t-VERTEX COVER (*t*-VC):



Cover *t* edges.

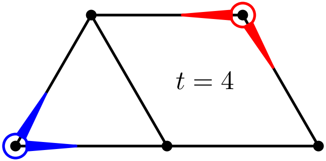
t-DOMINATING SET (*t*-DS):



Dominate *t* nodes.

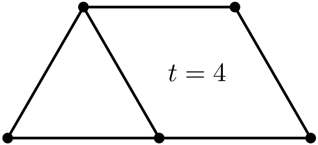
Complete vs. Partial Solutions

t-VERTEX COVER (*t*-VC):



Cover *t* edges.

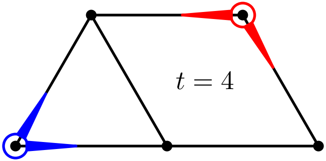
t-DOMINATING SET (*t*-DS):



Dominate *t* nodes.

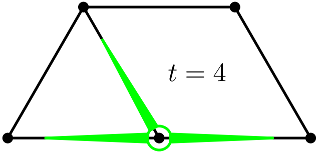
Complete vs. Partial Solutions

t-VERTEX COVER (*t*-VC):



Cover *t* edges.

t-DOMINATING SET (*t*-DS):



Dominate *t* nodes.

Parameterized Complexity

Downey and Fellows: *Parameterized Complexity*, 1999.
Flum and Grohe: *Parameterized Complexity Theory*, 2006.
Niedermeier: *Invitation to Fixed-Parameter Algorithms*, 2006.

- Analyze complexity in n and a parameter k
- L with parameter k is in FPT $:\Leftrightarrow$
  L can be solved in time $O(f(k) \cdot poly(n))$
- Hierarchy: FPT $\subseteq$ W[1] $\subseteq$ W[2] $\subseteq \ldots$
- Seemingly, L cannot be in FPT if it is W[1]-hard

Parameterized Complexity

Downey and Fellows: *Parameterized Complexity*, 1999.
Flum and Grohe: *Parameterized Complexity Theory*, 2006.
Niedermeier: *Invitation to Fixed-Parameter Algorithms*, 2006.

- Analyze complexity in n and a parameter k
- L with parameter k is in FPT $:\Leftrightarrow$
  L can be solved in time $O(f(k) \cdot poly(n))$
- Hierarchy: FPT $\subseteq$ W[1] $\subseteq$ W[2] $\subseteq \ldots$
- Seemingly, L cannot be in FPT if it is W[1]-hard

Parameterized Complexity

Downey and Fellows: *Parameterized Complexity*, 1999.
Flum and Grohe: *Parameterized Complexity Theory*, 2006.
Niedermeier: *Invitation to Fixed-Parameter Algorithms*, 2006.

- Analyze complexity in n and a parameter k
- L with parameter k is in FPT $:\Leftrightarrow$
  L can be solved in time $O(f(k) \cdot poly(n))$
- Hierarchy: FPT $\subseteq$ W[1] $\subseteq$ W[2] $\subseteq \ldots$
- Seemingly, L cannot be in FPT if it is W[1]-hard

Parameterized Complexity

Downey and Fellows: *Parameterized Complexity*, 1999.
Flum and Grohe: *Parameterized Complexity Theory*, 2006.
Niedermeier: *Invitation to Fixed-Parameter Algorithms*, 2006.

- Analyze complexity in n and a parameter k
- L with parameter k is in FPT $:\Leftrightarrow$
  L can be solved in time $O(f(k) \cdot poly(n))$
- Hierarchy: FPT $\subseteq$ W[1] $\subseteq$ W[2] $\subseteq \ldots$
- Seemingly, L cannot be in FPT if it is W[1]-hard

Parameterized Complexity

Downey and Fellows: *Parameterized Complexity*, 1999.
Flum and Grohe: *Parameterized Complexity Theory*, 2006.
Niedermeier: *Invitation to Fixed-Parameter Algorithms*, 2006.

- Analyze complexity in n and a parameter k
- L with parameter k is in FPT $:\Leftrightarrow$
  L can be solved in time $O(f(k) \cdot poly(n))$
- Hierarchy: FPT $\subseteq$ W[1] $\subseteq$ W[2] $\subseteq \ldots$
- Seemingly, L cannot be in FPT if it is W[1]-hard

Complete vs. Partial Solutions

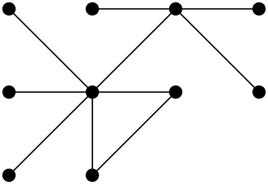
Problem	Parameter	Best result	Reference
VC	k	$O(1.2738^k + kn)$	Chen et al.
t-VC	k	W[1]-hard	Guo et al.
t-VC	t	$O(2.0911^t n(n+m)k)$	ISAAC'06
DS	k	W[2]-complete	DF99
t-DS	k	W[2]-hard	(obvious)
t-DS	t	$O((4+\varepsilon)^t poly(n))$	(today)

Algorithmics for t-Vertex Cover

Method	Result	Reference
Color-Coding	$5.4366^t \cdot poly(n)$	M. Bläser
Random Separation	$4.0000^t \cdot poly(n)$	Cai et al.
Randomized Branching	$2.0911^t \cdot poly(n)$	ISAAC'06

Divide-and-Color

LONGEST PATH: Does $G = (V, E)$ contain a k-node path?

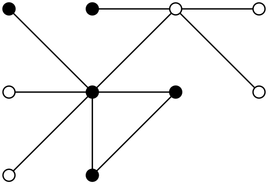


1. Randomly color G in black and white.
2. Recursively check for a black $\lceil k/2 \rceil$-node path and a white $\lfloor k/2 \rfloor$-node path that combine to form a k-node path in G.

$\longrightarrow$ Randomized $O(4^k \cdot poly(n))$ algorithm [WG 2006]

Divide-and-Color

LONGEST PATH: Does $G = (V, E)$ contain a k-node path?

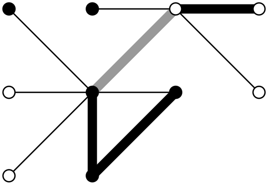


1. Randomly color G in black and white.
2. Recursively check for a black $\lceil k/2 \rceil$-node path and a white $\lfloor k/2 \rfloor$-node path that combine to form a k-node path in G.

$\longrightarrow$ Randomized $O(4^k \cdot poly(n))$ algorithm [WG 2006]

Divide-and-Color

LONGEST PATH: Does $G = (V, E)$ contain a k-node path?

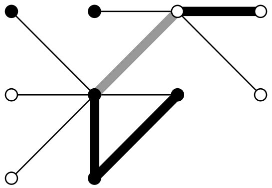


1. Randomly color G in black and white.
2. Recursively check for a black $\lceil k/2 \rceil$-node path and a white $\lfloor k/2 \rfloor$-node path that combine to form a k-node path in G.

$\longrightarrow$ Randomized $O(4^k \cdot poly(n))$ algorithm [WG 2006]

Divide-and-Color

LONGEST PATH: Does $G = (V, E)$ contain a k-node path?



1. Randomly color G in black and white.
2. Recursively check for a black $\lceil k/2 \rceil$-node path and a white $\lfloor k/2 \rfloor$-node path that combine to form a k-node path in G.

$\longrightarrow$ Randomized $O(4^k \cdot poly(n))$ algorithm [WG 2006]

Divide-and-Color

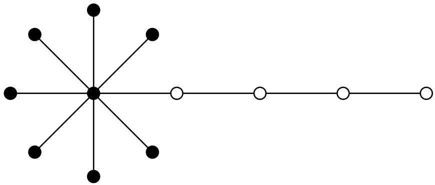
Applied to t-Dominating Set:

1. Randomly color G in black and white.
2. Recursively check for a black $\lceil t/2 \rceil$-DS and a white $\lfloor t/2 \rfloor$-DS whose combined size is $\leq k$.

$\longrightarrow$ Randomized $O(4^t \cdot poly(n))$ algorithm?

Unbalanced Solutions

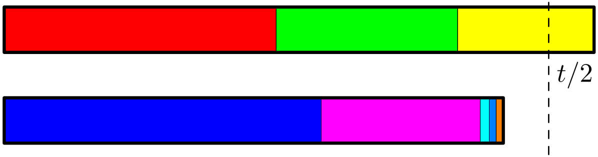
Let $k = 2$, $t = 10$:



Every t-DS of size k is *unbalanced*.

Unbalanced Solutions

What if all minimum solutions are unbalanced?

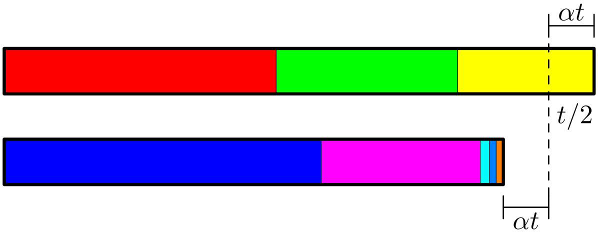


- PARTITION: Imbalance requires large numbers
- t-DS: Imbalance requires high-degree nodes

$\longrightarrow$ A small fraction of a solution yields a $t/2$-DS

Unbalanced Solutions

We define α-balance:



Lemma: $\neg\exists$ α-balanced solution $\Rightarrow$ $\exists$ $t/2$-DS of size $\beta := \lceil \frac{1}{2\alpha} \rceil$

Towards an Algorithm

TDS(G, t):

if $\exists$ small solution D **then return** D; **fi**;
if $|V| = \emptyset$ **then return** ∞; **fi**
$k_{opt} := \infty$;
for $4 \cdot 2^t$ **times do**
  color G;
  $k_{opt} \leftarrow$ <Handle the unbalanced case>;
  $k_{opt} \leftarrow$ <Handle the balanced case>;
endfor;
return k_{opt};

Correctness

Lemma

TDS(G, t) returns the size of a minimum t-DS with prob. $\geq \frac{1}{2}$.

Unbalanced case:

- Good coloring: 2^{-t}
- Recursive call: $\frac{1}{2}$
- Total failure probability:

$$(1 - 2^{-t} \cdot \frac{1}{2})^{4 \cdot 2^t} \leq e^{-2}$$

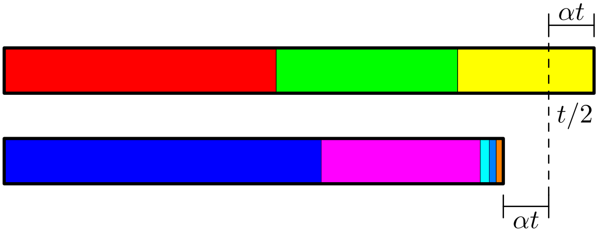
Balanced case:

- Good coloring: 2^{-t}
- Recursive call: $\frac{1}{2} \cdot \frac{1}{2}$
- Total failure probability:

$$(1 - 2^{-t} \cdot \frac{1}{4})^{4 \cdot 2^t} \leq e^{-1}$$

Runtime Bound

Recall α-balance:



Lemma

TDS(G, t) *performs* $\leq 4^{(1+\alpha)t} \cdot t^6$ *recursive calls.*

Main Result

Theorem

Let $0 < \alpha \leq 1/25$. t-DOMINATING SET *can be solved with exponential small error probability in time*

$$O((4+6\alpha)^t \cdot t^6 \cdot n^{\lfloor \frac{1}{2\alpha} \rfloor + 1}).$$

Derandomized: $O((16+\varepsilon)^t poly(n))$