Linear-Time Computation of a Linear Kernel for Dominating Set on Planar Graphs

René van Bevern[1] Sepp Hartung[1] Frank Kammer[2]
Rolf Niedermeier[1] Mathias Weller[1]

[1]TU Berlin, Germany

[2]University of Augsburg, Germany

Worker 2011

Planar Dominating Set

Planar Dominating Set

Input: A planar graph G and an integer k.

Question: Is there a dominating set of size k for G?

- $\gamma(G) =$ size of minimum dominating set of G

Known Results:

- Garey & Johnson, 1979: NP-hardness
- Baker, J. ACM 1994: PTAS
- subexponential algorithms:
  - Alber, Bodlaender, Fernau, Kloks & Niedermeier, Algorithmica 2002

Problem Kernels

Focusing on Kernel Size:

Alber, Fellows & Niedermeier, J. ACM 2004:
335γ-vertex kernel in $O(n^3)$ time

Chen, Fernau, Kanj & Xia, SIAM J. Comput. 2007:
67γ-vertex kernel in $O(n^3)$ time

Focusing on Larger Graph Classes:

Fomin & Thilikos, ICALP 2004:
$O(\gamma + g)$-vertex kernel in $O(gn^3)$ time on graphs of genus g

Philip, Raman & Sikdar, ESA 2009:
$O(\gamma^{2(d+1)^2})$-vertex kernel in $O(2^d dn^2)$ time on d-degenerated graphs

Fomin, Lokshtanov, Saurabh & Thilikos, SODA 2010:
$O(\gamma)$-vertex kernel in polynomial time on apex-minor free graphs

Our Results

van Bevern, Hartung, Kammer, Niedermeier & Weller, IPEC 2011:
O(γ)-vertex kernel in O(n) time

Motivation:

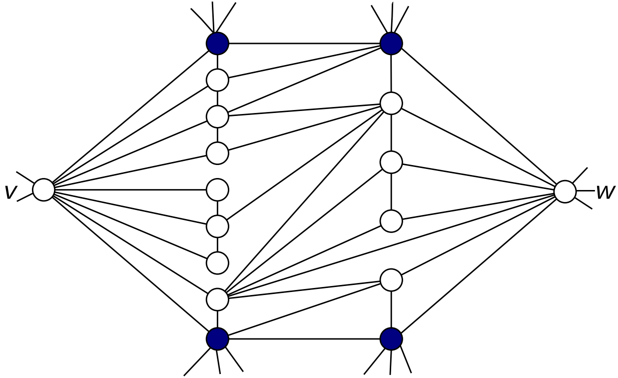
- kernel size $\Leftrightarrow$ running time of kernelization
  - Which kernel size can be achieved in linear time?
  - combine **size-optimized** with **time-optimized** kernelization algorithm
    - with Chen et al., 2007: 67γ-vertex kernel in $O(\gamma^3 + n)$ time

Main Features:

- rework and more refined analysis of the interaction of the reduction rules of Alber et al., 2004
- non-exhaustive application of rules $\rightarrow$ "knowing when to stop"

Alber et al. kernel I

Main Observation: A graph is decomposable into $O(\gamma)$ regions.



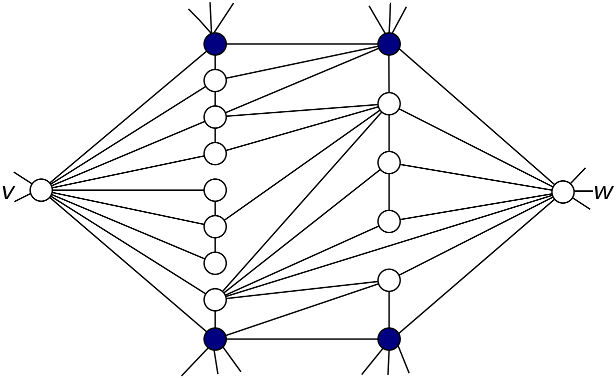
Definition

A region between two vertices v and w is a closed subset of the plane such that

1. all vertices are contained in $N(v) \cup N(w)$, and
2. the boundary is formed by two simple paths of length at most three.

Alber et al. kernel I

Main Observation: A graph is decomposable into $O(\gamma)$ regions.

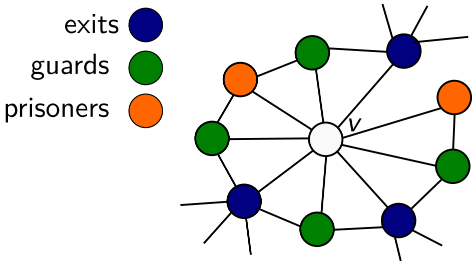


Problem kernel is obtained by two reduction rules:

1. **Private Neighborhood**: Reduces number of vertices outside of regions to $O(\gamma)$
2. **Joint Neighborhood**: Shrink each region to $O(1)$ vertices

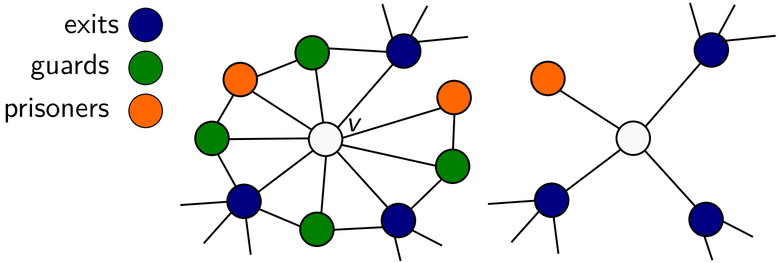
Alber et al.: Private Neighborhood Rule

- partition the neighborhood of a vertex v into three sets



Alber et al.: Private Neighborhood Rule

- partition the neighborhood of a vertex v into three sets

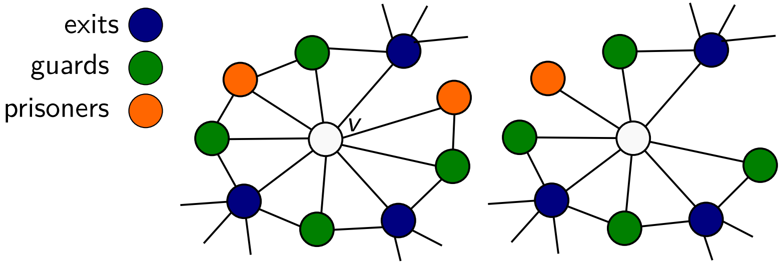


Private Neighborhood Rule

Delete guards and prisoners and add a degree-one neighbor.

Alber et al.: Private Neighborhood Rule

- partition the neighborhood of a vertex v into three sets



Private Neighborhood Rule

Delete guards and prisoners and add a degree-one neighbor.

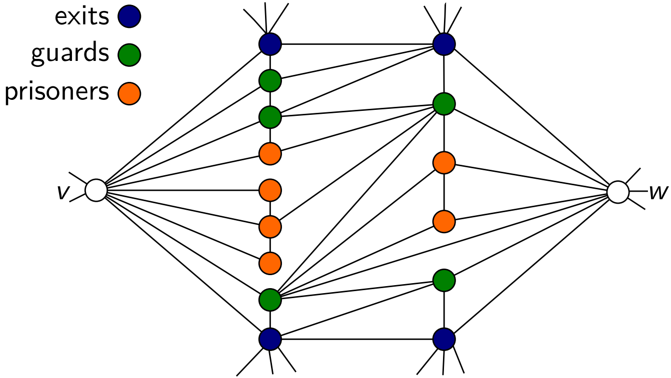
Modified Private Neighborhood Rule

Delete prisoners and add a degree-one neighbor.

- avoids **cascading effects** $\rightarrow$ exhaustive application in $O(n)$ time

Alber et al.: Joint Neighborhood Rule I

Joint Neighborhood Rule: **Finds and Shrinks Regions**

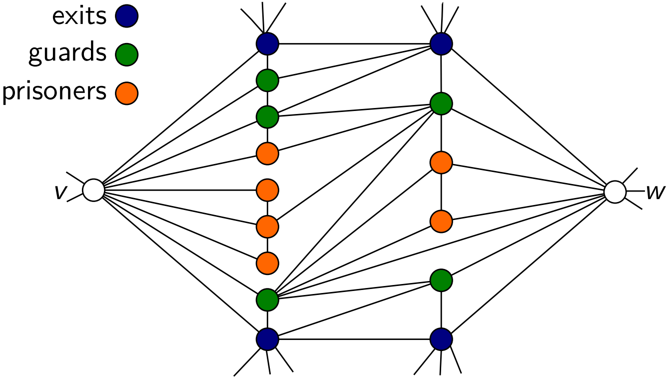


Shrink every **potential region:**

- Alber et al.: For all vertices v, w delete vertices from $N(v) \cup N(w)$ $\rightarrow \Omega(n^2)$. Rule applies $O(n)$ times $\rightarrow$ $O(n^3)$ time

Alber et al.: Joint Neighborhood Rule I

Joint Neighborhood Rule: **Finds and Shrinks Regions**



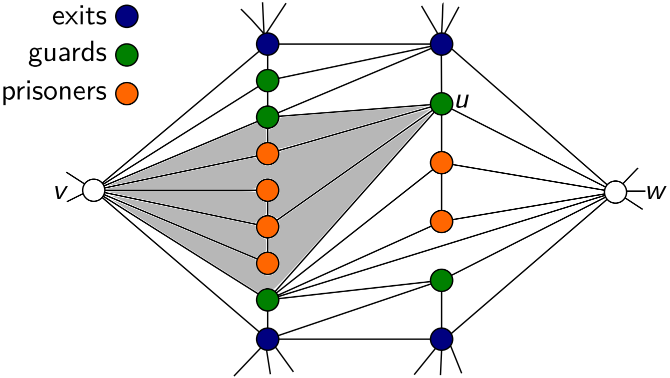
Shrink every **potential region:**

- Modified Joint Neighborhood Rule:
  - DFS: starting in v, list all vertices up to a depth of three
  - follow only constant degree vertices

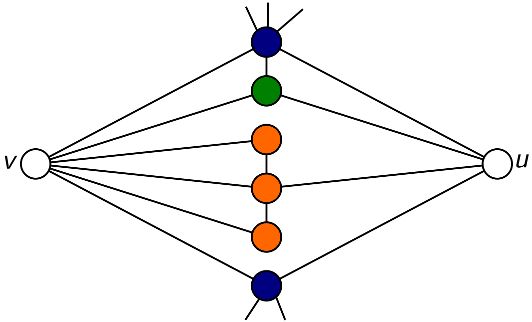
Alber et al.: Joint Neighborhood Rule II

Our Main Observation:

- the neighbors of inner vertices lie in the region → adjacent to v or w
- inner vertices with large degree form **restricted regions**



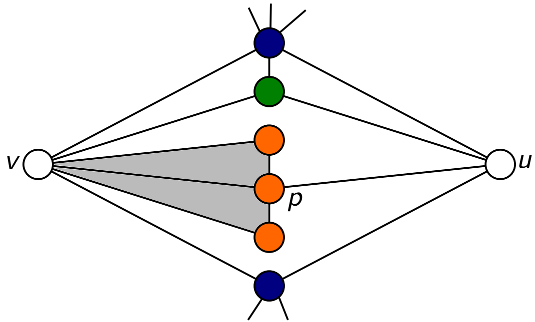
Restricted Regions I



Observations:

- all inner vertices have to be adjacent to v
- boundary paths have length at most two

Restricted Regions I



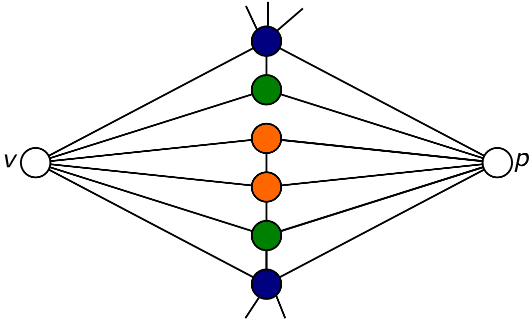
Observations:

- all inner vertices have to be adjacent to v
- boundary paths have length at most two
- a common neighbor $p \in N(u) \cap N(v)$ can still have **unbounded degree**
  - $\rightarrow$ consider the restricted region between v and p

Restricted Region II

Observations:

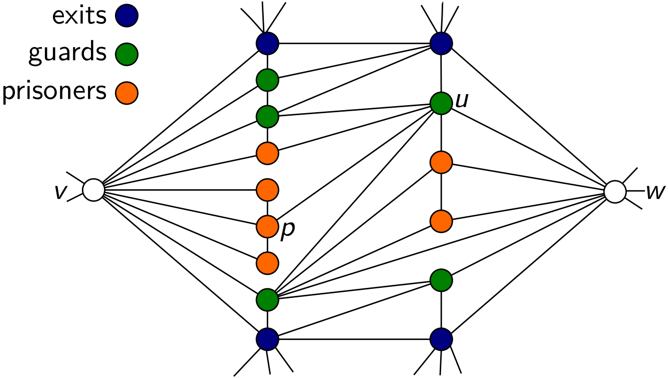
- all inner vertices have to be **adjacent to v and p**
  - planarity: **inner vertices have constant degree!**



- $v - p$ region can be explored in time bounded by the number of inner vertices→ **Modified Joint Neighborhood Rule**

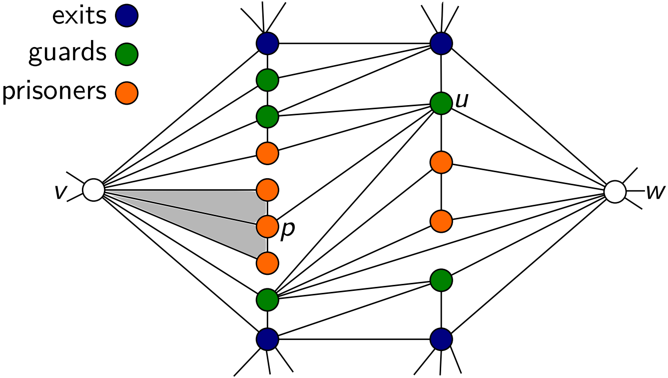
Kernelization Algorithm

1. Modified Private Neighborhood Rule: Shrinks vertices outside of regions



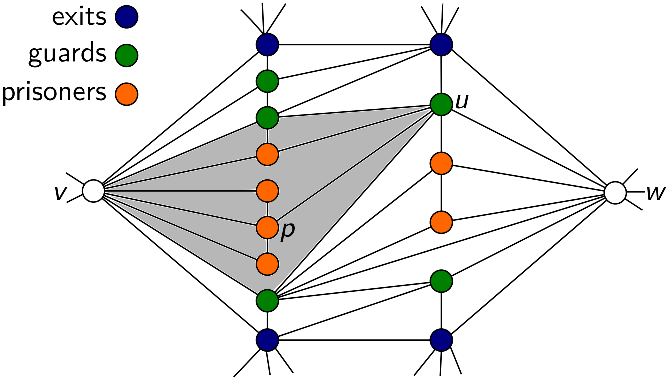
Kernelization Algorithm

1. Modified Private Neighborhood Rule: Shrinks vertices outside of regions
2. Modified Joint Neighborhood Rule: Shrinks $v - p$ region $\rightarrow$ $\deg(p)$ constant



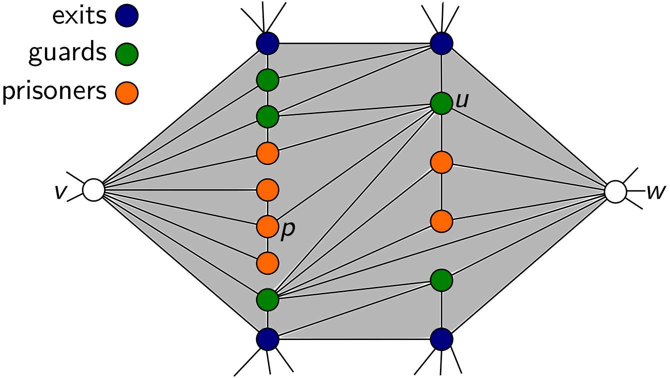
Kernelization Algorithm

1. Modified Private Neighborhood Rule: Shrinks vertices outside of regions
2. Modified Joint Neighborhood Rule: Shrinks $v - p$ region $\rightarrow$ $\deg(p)$ constant
3. Modified Joint Neighborhood Rule: Shrinks $v - u$ region $\rightarrow$ $\deg(u)$ constant



Kernelization Algorithm

1. Modified Private Neighborhood Rule: Shrinks vertices outside of regions
2. Modified Joint Neighborhood Rule: Shrinks $v - p$ region $\rightarrow \deg(p)$ constant
3. Modified Joint Neighborhood Rule: Shrinks $v - u$ region $\rightarrow \deg(u)$ constant
4. Modified Joint Neighborhood Rule: Shrinks $v - w$ region to constant size



Conclusion

- kernel size and kernel time race
  - combining them yields fast and effective preprocessing for hard problems
- approaches to get time-optimized kernels:
  1. Speed-up known kernels (our work)
  2. Develop new time-optimized kernels
     - Hagerup, IPEC 2011, Linear-Time Kernelization for Planar Dominating Set

Outlook & Open Question

- implementation of our approach
  - 335γ kernel: ALBER, BETZLER & NIEDERMEIER, ANN. OPER. RES., 2006

- Exhaustive application of reduction rules of Alber et al. (or even Chen et al.) in linear time?
- Can our approach be extended to other problems?
  - CONNECTED VERTEX COVER
  - GUO & NIEDERMEIER, ICALP 2007
- Lower Bounds: Does every FPT-problem admit a kernel in $O(n)$ time?
  - near-linear time: best kernel in $O(p(k) + n)$ or $O(p(k) \cdot n)$ time?