A Linear Kernel for the Differential of a Graph

Can we beat combinatorial kernels?

Sergio Bermudo

Henning Fernau

Universidad Pablo de Olavide   

Universität Trier 

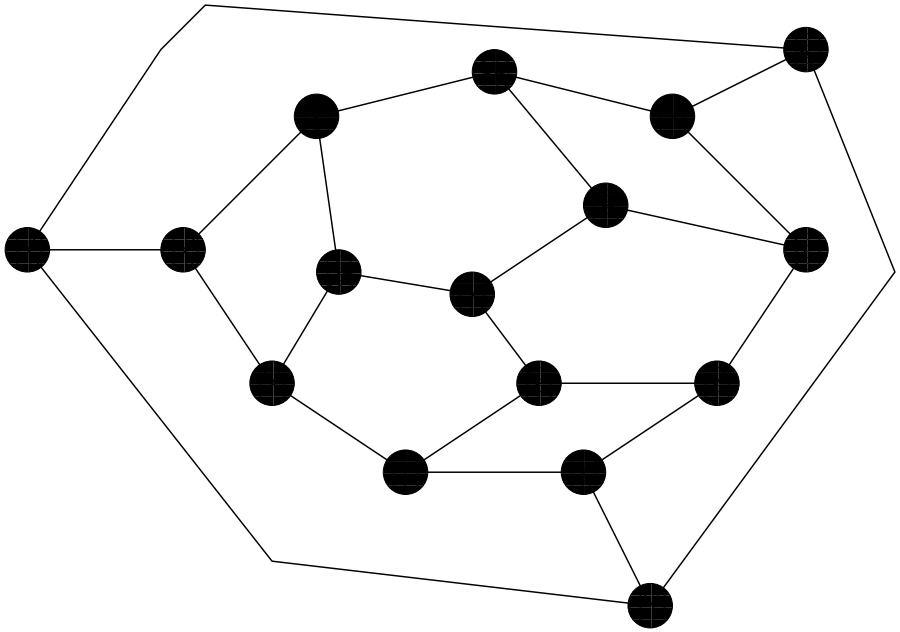

sbernav@upo.es,fernau@uni-trier.de

WorKer, Vienna, August 2011

Overview on the talk

- The differential of a graph
- Classical complexity
- Parameterized complexity
- Kernel results
- Measure and Conquer: Exact algorithms

How to influence a small network

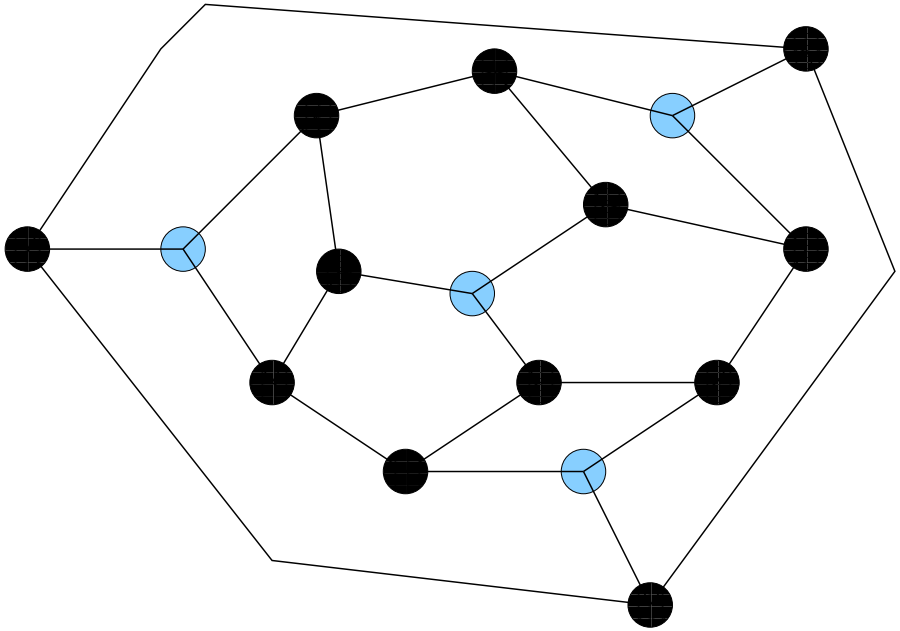


Conflicting requirements:

— large influence
— small budget

Aim: Find the important points to influence as many as possible at small costs.

The differential as a simplified model



— Buy influence points S

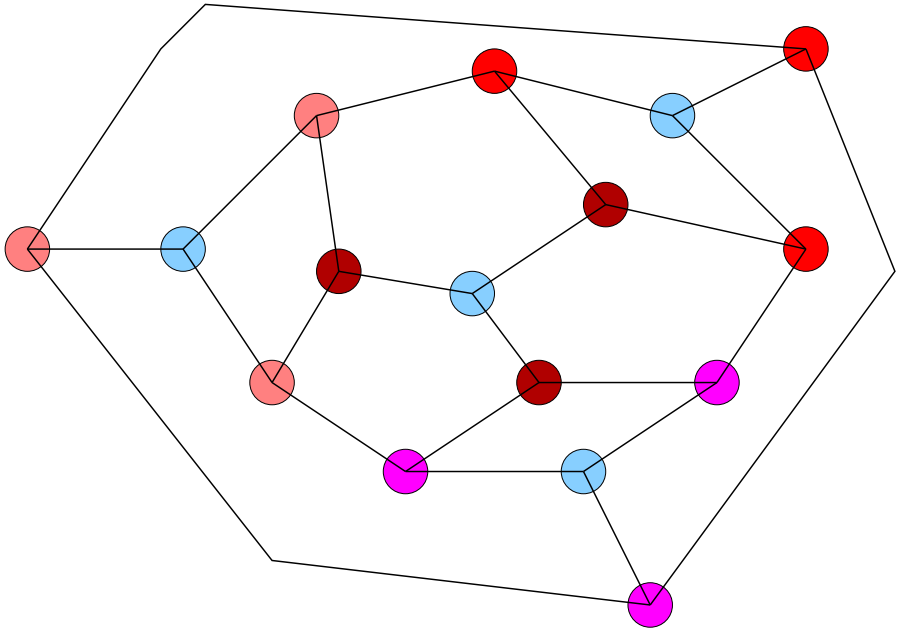
— They "earn" you their neighbors.

— Net gain: $|B(S)| - |S|$.

Aim: Maximize net gain.

Example net gain: 8.

The differential as a simplified model



— Buy influence points S

— They "earn" you their neighbors.

— Net gain: $|B(S)| - |S|$.

Aim: Maximize net gain.

Example net gain: 8.

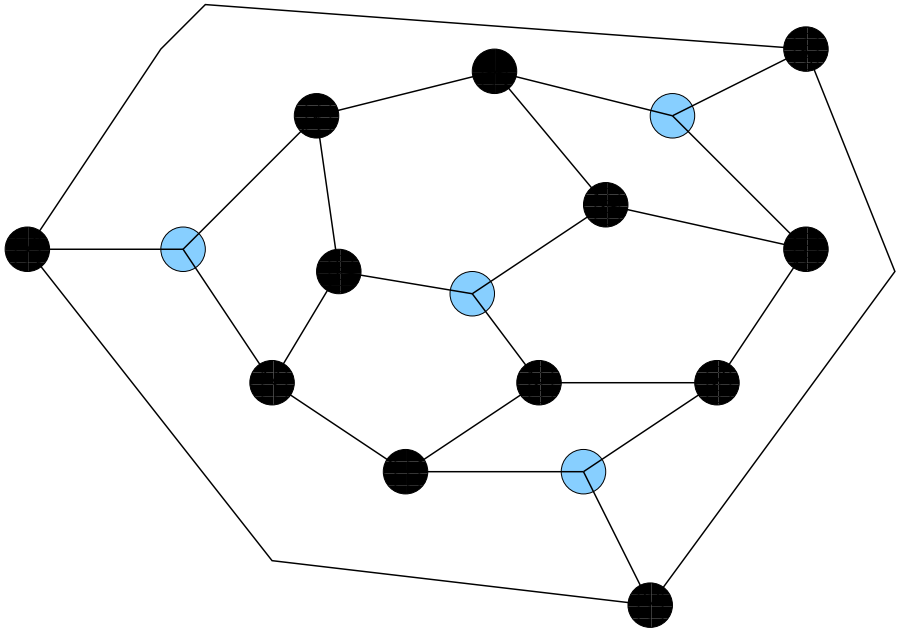
Our Problem: Finding a small differential set

A set S delivering the largest $\partial(S)$ is called a ∂-*set*, and we set $\partial(\Gamma)$ to $\partial(S)$.

Motivation from viral marketing

J. L. Mashburn, T. W. Haynes, S. M. Hedetniemi, S. T. Hedetniemi, P. J. Slater, Differentials in graphs, Utilitas Mathematica 69 (2006) 43–54.

Determining the differential of a graph

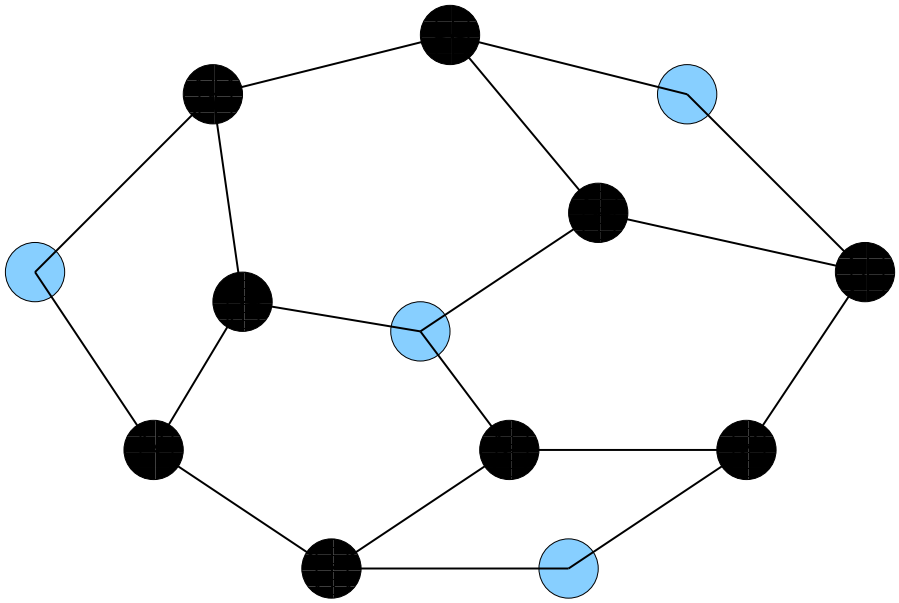


— Differential set S

Example value: 8.

Let us delete the outermost three vertices. . .

Determining the differential of a graph



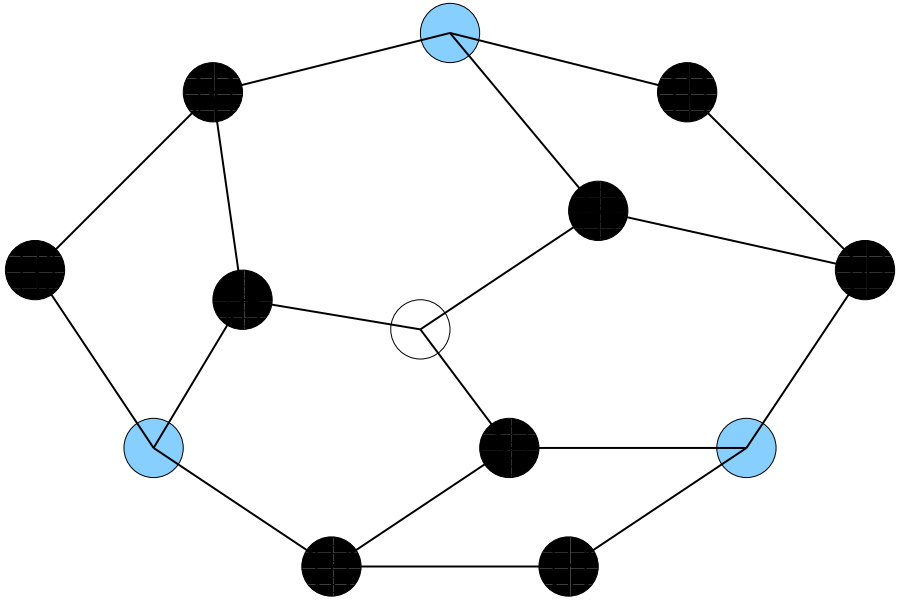
— The set S

Example value: 5.

All vertices are either in S or neighbors of S-vertices.

Is this optimal?

Determining the differential of a graph



— Differential set S

Example value: 6.

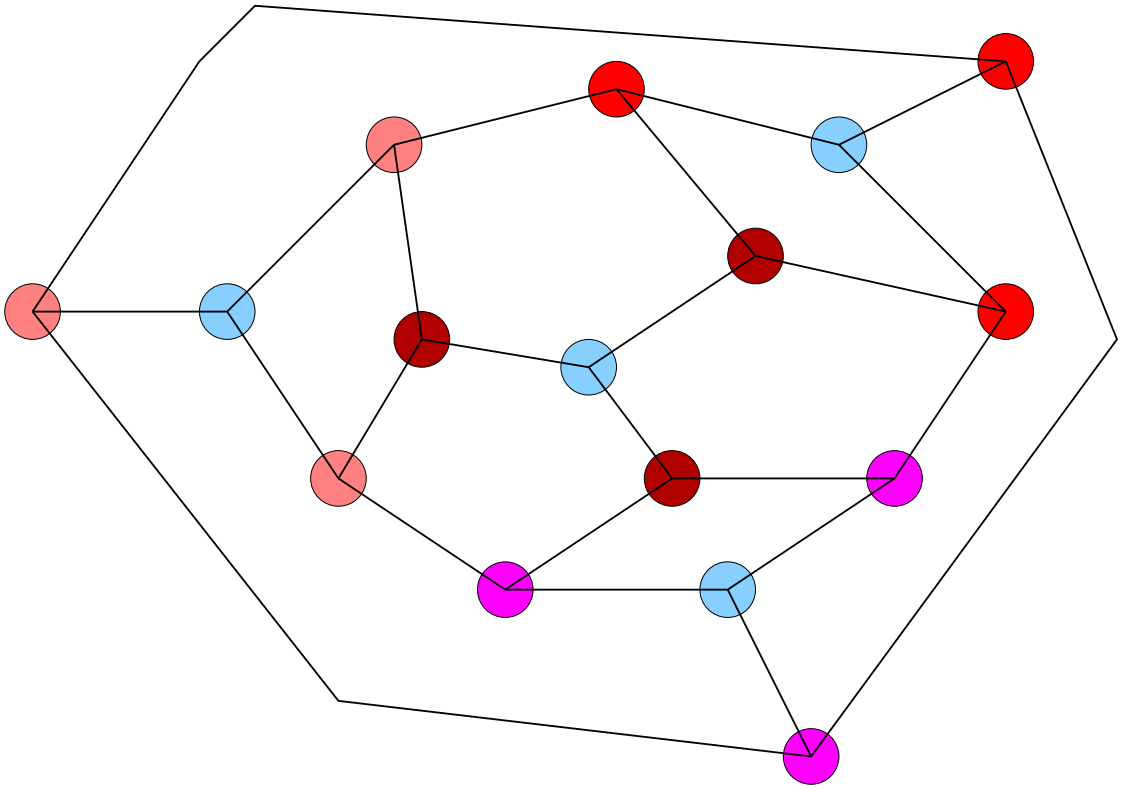
The white vertex is not in $B(S)$!

An alternative definition

Call any star S_d with $d \geq 2$ a *big star*.
Given the graph $\Gamma = (V, E)$, a *big star packing* is a vertex-disjoint collection $\mathcal{S} = \{X_i \mid 1 \leq i \leq k\}$ of big stars $X_i \subseteq V$.
$\partial(\mathcal{S}) = \sum_{S \in \mathcal{S}} (|S| - 2)$: The differential of a big star packing.
$SP(\Gamma, \mathcal{S})$ reads as: "$\mathcal{S}$ is a big star packing of Γ."

Proposition: $\partial(\Gamma) = \max\{\partial(\mathcal{S}) : SP(\Gamma, \mathcal{S})\}$.

Can you see the stars?



Overview on the talk

- The differential of a graph
- Classical complexity
- Parameterized complexity
- Kernel results
- Measure and Conquer: Exact algorithms

NP-hardness results for special graph classes.

1. MDS on split graphs is NP-complete.
2. MDS on cubic graphs is NP-complete.

The reductions use variants of the following problem:

3-DIMENSIONAL MATCHING WITH MAXIMUM DEGREE THREE (3DM3)

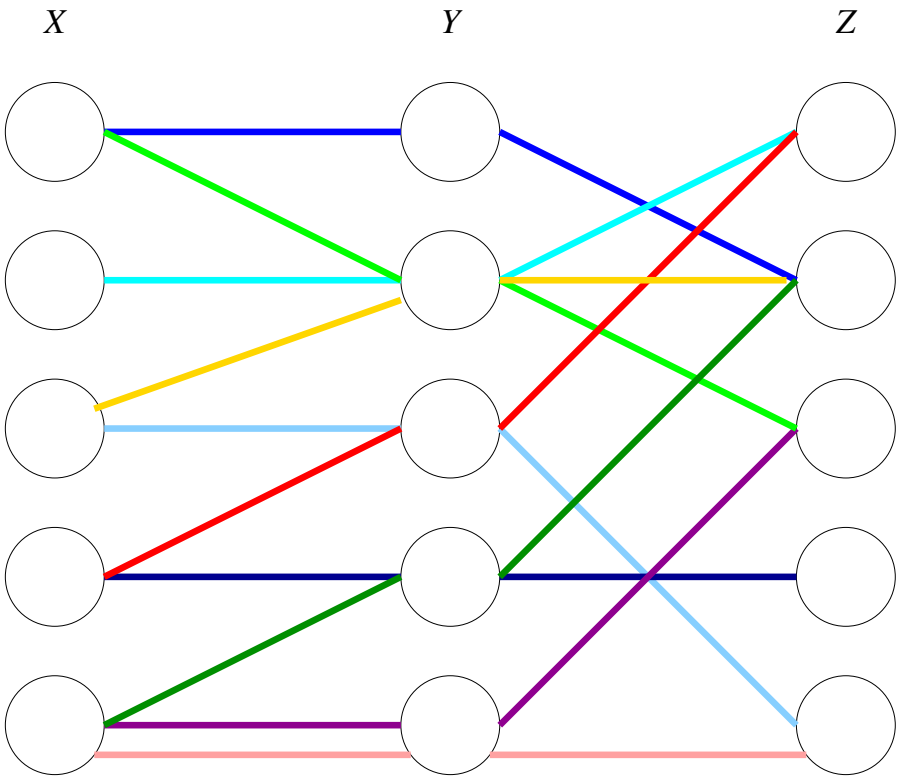
Given: Three sets X, Y, Z with $|X| = |Y| = |Z| = q$, and a relation $R \subseteq X \times Y \times Z$, such that, for each $v \in G = X \cup Y \cup Z$, there are at most three triples in R that contain v

Parameter: a positive integer k

Question: Are there at least k triples $r_1, \ldots, r_k$ in R such that each element of G occurs exactly once in one of the triples?

Known to be NP-hard for $k = q$ (perfect matching, 3DPM3).

3-DIMENSIONAL MATCHING WITH MAXIMUM DEGREE THREE



3-DIMENSIONAL MATCHING

— The blue will do.

— Each element participates in at most three triples.

A simple graph representation for the split graph case

Split graphs: Vertex set can be split into a clique and an independent set.

Given: $G = X \cup Y \cup Z$ with $|X| = |Y| = |Z| = q$, a relation $R \subseteq X \times Y \times Z$, such that ...; where $k = q$ (perfect matching).
Constructed split graph: $\Gamma = (V, E)$ with $V = G \cup R$ and $E = E_1 \cup E_2$, where
$E_1 = \{v(x, y, z) : v \in G, (x, y, z) \in R, v \in \{x, y, z\}\}$ and
$E_2 = \{(x, y, z)(x', y', z') : (x, y, z) \in R, (x', y', z') \in R \setminus \{(x, y, z)\}\}$.

Theorem: MDS on split graphs is NP-complete.

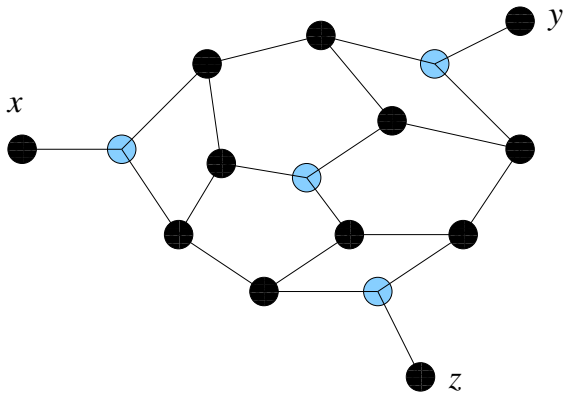
Cubic graph representations are more complicated

The previous construction is “nearly cubic”, apart from:

(1) The 3DPM3 instance might contain elements in G that occur less than thrice in R. $\rightsquigarrow$ Separate construction of a “normal form 3DPM3”, called 3DPM3’

(2) The edge set E_2 destroys cubicity.

$\Rightarrow$ Need for a special gadget for the relation triples $(x, y, z) \in R$:



Theorem: MDS on cubic graphs is NP-complete.

Overview on the talk

- The differential of a graph
- Classical complexity
- Parameterized complexity
- Kernel results
- Measure and Conquer: Exact algorithms

Recalling MAXSNP

Introduced in: C. H. Papadimitriou, M. Yannakakis, Optimization, approximation, and complexity classes, Journal of Computer and System Sciences 43 (1991) 425–440.

A syntactically defined approximation complexity class.

MAXSNP is closed under L-reductions.

All MAXSNP-complete problems are constant-factor approximable, but unlikely to possess approximation schemes.

Well-known MAXNP-complete problems:
— MAX 3DM3
— MAX IS-B (independent set on graphs whose degree is bounded by $B \geq 3$)

MAX DIF and MAXSNP: Our results

- MAX DIF in split graphs is MAXSNP-hard.
- MAX DIF in subcubic graphs is MAXSNP-complete.
- For any $d \geq 3$, the problem MAX DIF, restricted to graphs of maximum degree d, is MAXSNP-complete.

The hardness reductions start from MAX 3DM3.
Membership in MAXSNP is shown by a reduction using MAX IS-B.

Consequences for Parameterized Complexity

L. Cai, J. Chen, On fixed-parameter tractability and approximability of NP optimization problems, Journal of Computer and System Sciences 54 (1997) 465–474.

Corollary: For any $d \geq 3$, the problem k-MDS, restricted to graphs of maximum degree d, is in FPT.

L. Cai, D. Juedes, On the existence of subexponential parameterized algorithms, Journal of Computer and System Sciences 67 (2003) 789–807.

Corollary: For any $d \geq 3$, there is no algorithm running in time $\mathcal{O}^*(2^{o(k)})$ that decides, given a graph $\Gamma = (V, E)$ of maximum degree d and a parameter k, if $\partial(\Gamma) \geq k$ and also produces a witness set $S \subseteq V$ with $\partial(S) \geq k$ together with a possible YES-answer, unless 3-SAT can be deterministically solved in time $\mathcal{O}(2^{o(n)})$, where n is the number of variables occurring in an input formula.

Overview on the talk

- The differential of a graph
- Classical complexity
- Parameterized complexity
- Kernel results
- Measure and Conquer: Exact algorithms

Parameterized complexity in a nutshell

Problems with a parameter k

Running time $\mathcal{O}(f(k)p(n))$

Complexity class: FPT

Standard approaches: kernelization and search-trees; this could be combined

Kernelization: pol.-time reduction $(I, k) \mapsto (I', k')$ s.t. $\max\{|I'|, k'\} \leq g(k)$.
Direct answers (YES or NO) can be integrated on top.

Well-known: A parameterized problem is in FPT iff it admits a kernelization.

A Combinatorial kernel for k-MDS

Theorem (B&F 2011): In a connected graph Γ of order n, $n \geq 3$, $\partial(\Gamma) \geq \left\lceil \frac{n}{5} \right\rceil$.
This suggests the following kernelization:

1. Remove all isolated vertices and edges.

2. If the remaining graph Γ' has order at least $5k$, then answer YES.

3. Otherwise, Γ' has at most $5k$ vertices.

This kernel is of a purely combinatorial nature and
hence, it looks a bit like cheating: it involves no “real algorithmics”.

Our combinatorial result I

The structure behind stars:

D: star centers
$B(D)$: neighbors of star centers, but not in D
$C(D)$: the remaining vertices

Combinatorial trick: Consider D with $\partial(D) = \partial(\Gamma)$ s.t. $|D|$ is minimum.

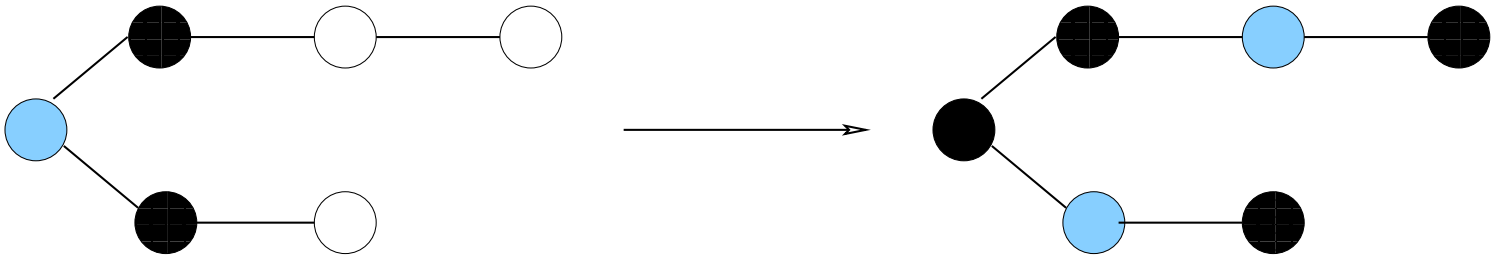
(a) for all $v \in D$, $\delta_{B(D)}(v) \geq 2$, (big star packing)

(b) for all $v \in B(D)$, $\delta_{C(D)}(v) \leq 2$, (otherwise, form a new star)

(c) for all $v \in C(D)$, $\delta_{C(D)}(v) \leq 1$. (otherwise, form a new star)

Our combinatorial result II

Small stars may attain the bound but do not violate it:



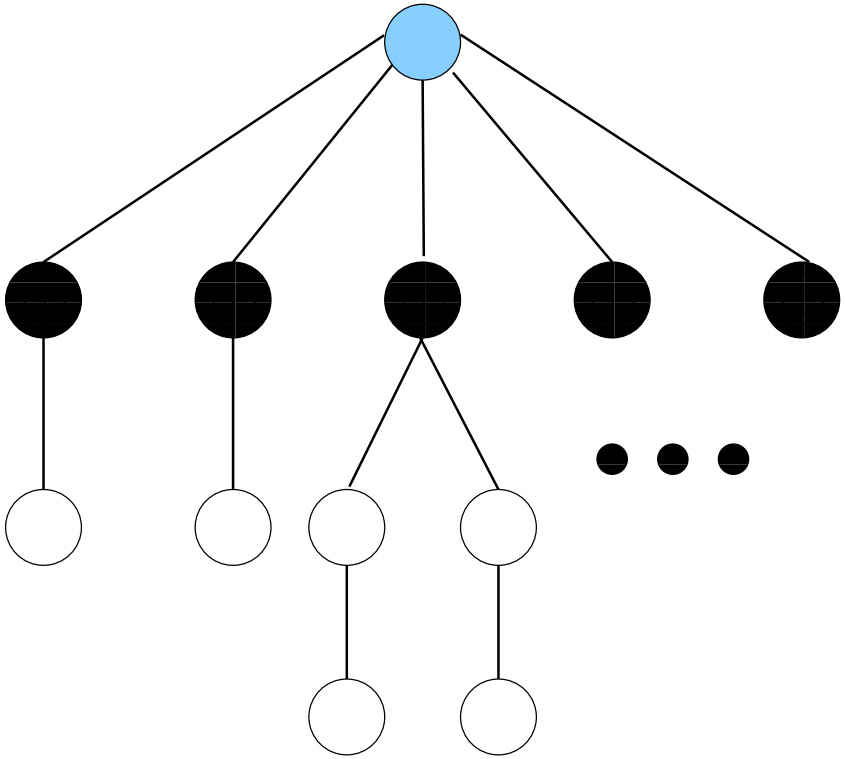
Similar reasoning for other cases

$\rightsquigarrow$ An S_2 star in $\mathcal{S}(D)$ can have at most two "neighbors" in $C(D)$.

$\rightsquigarrow$ $5k$ bound confirmed in this case.

Our combinatorial result III: Big stars S_d

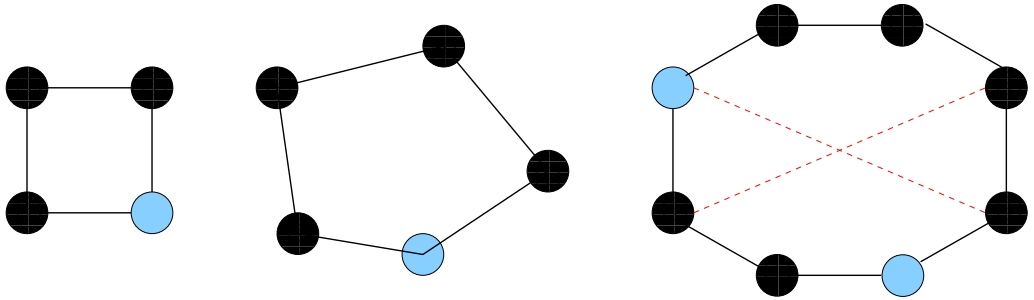
Worst case example (this attains the $5k$ for $d \geq 2$):



Comments

The many leaves in the worst-case example are necessary. We have shown:

Theorem: For any connected graph Γ of order n that has minimum degree two, $\partial(\Gamma) \geq \frac{3n}{11}$, apart from five exceptional graphs listed below, where $\partial(\Gamma) = n/4$ or $\partial(\Gamma) = n/5$.



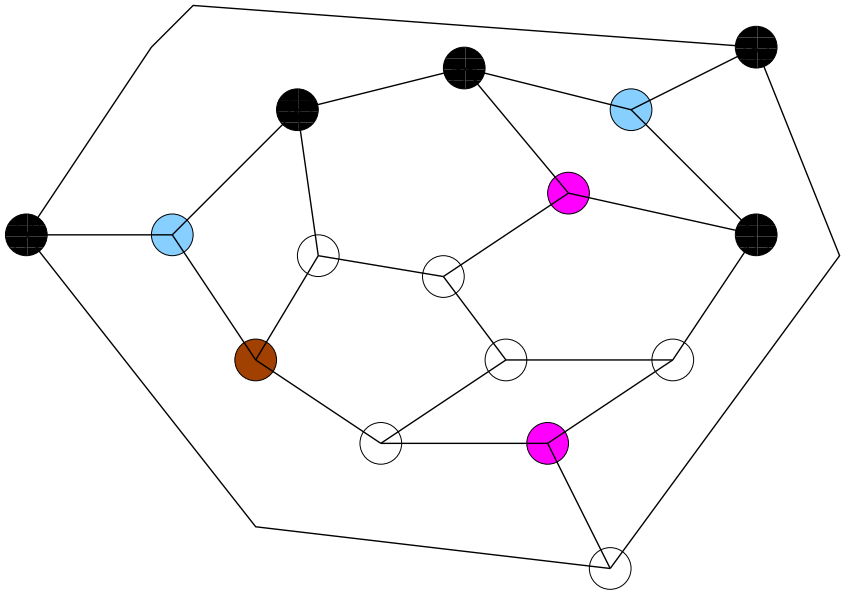
Red dotted edges may be present or not.

We have found NO kernel exploit for this.

Overview on the talk

- The differential of a graph
- Classical complexity
- Parameterized complexity
- Measure and Conquer: Exact algorithms

A colorful situation in the search tree



○: undecided, nothing known

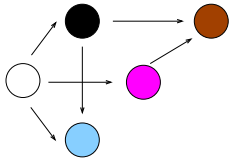
●: in the differential set

●: influenced

●: influenced; decided not to go into the differential set

●: not influenced; decided not to go into the differential set

Possible transitions:



Branch on ○ or ● vertices as long as there are ○ or ● vertices.

Measure and Conquer: MAXIMUM DIFFERENTIAL SET

Algorithm 1 A simple algorithm for MAXIMUM DIFFERENTIAL SET

1: **if** possible choose a $v \in$ ○ such that $|N(v) \cap ($○ $\cup$ ●$)| \geq 3$; **then**
2: Binary branch on v (i.e., set v ● in one branch, ● in the other)
3: **else if** possible choose a $v \in$ ● such that $|N(v) \cap ($○ $\cup$ ●$)| \geq 4$; **then**
4: Binary branch on v.
5: **else**
6: Solve the remaining instance using an INDEPENDENT SET algorithm.

Very similar to the DOMINATING SET algorithm as shown in:
H. F,, D. Raible, Searching trees: an essay, in: J. Chen, S. B. Cooper (Eds.), Theory and Applications of Models of Computation TAMC, Vol. 5532 of LNCS, Springer, 2009, pp. 59–70.

DIFFERENTIAL SET: **Analysis of the branching process in steps 2 and 4.**

Use the *measure* $\mu = |\bigcirc| + \omega \cdot (|\bullet| + |\blacksquare|) \leq n$

Let $n_\bigcirc = |N(v) \cap \bigcirc|$ and $n_\bullet = |N(v) \cap \bullet|$.

If v goes into the DS in **step** 2, we first reduce μ by one as v vanishes.

$\rightsquigarrow$ Vertices in $N(v) \cap \bigcirc$ will be influenced and hence moved to the set $\blacksquare$.

$\rightsquigarrow$ μ is reduced by $n_\bigcirc \cdot (1 - \omega)$.

Similarly, vertices from $N(v) \cap \bullet$ are influenced, so they disappear from μ.

$\rightsquigarrow$ μ is reduced by $n_\bullet \omega$.

If v is not put into DS, reduce μ by $(1 - \omega)$: v is moved from $\bigcirc$ to $\bullet$.

$\rightsquigarrow$ branching vector:

$$(1 + n_\bigcirc(1 - \omega) + n_\bullet \omega, (1 - \omega)) \tag{1}$$

where $n_\bigcirc + n_\bullet \geq 3$ due to step 1.

A similar analysis for **step** 4 yields (where $n_\bigcirc + n_\bullet \geq 4$ due to step 3):

$$(\omega + n_\bigcirc(1 - \omega) + n_\bullet \omega, \omega) \tag{2}$$

DIFFERENTIAL SET: **Analysis of the branching process in steps 2 and 4.**

Unfortunately, depending on $n_{\circ}$ and $n_{\bullet}$ we have an *infinite number of branching vectors*. $\rightsquigarrow$ Consider only the *worst case branches*.

For (1) these are the ones with $n_{\circ} + n_{\bullet} = 3$ and for (2) $n_{\circ} + n_{\bullet} = 4$.

$\rightsquigarrow$ a finite set of recurrences $R_1(\omega), \ldots, R_9(\omega)$ depending on ω.

Now, choose ω in a way such that the maximum root of the evolving characteristic polynomials is minimum. In this case we easily see that $\omega := 0.5$.

Then the worst case branching vector for both cases is $(2.5, 0.5)$.

$\rightsquigarrow$ The number of leaves of the search tree evolving from this branching vector can be bounded by $\mathcal{O}^*(1.7549^{\mu})$.

Thus, our algorithm may break the 2^n-barrier using a simple measure.

Solving the remaining cases with INDEPENDENT SET

In step 6, we create an INDEPENDENT SET instance $G_I = (V_I, E_I)$, where

$V_I = \{v \in$ ● $: |N(v) \cap ($○ $\cup$ ● $)| = 3\} \cup \{v \in$ ○ $: |N(v) \cap ($○ $\cup$ ● $)| = 2\}$.

$u, v \in V_I$ are said to be *in conflict* if $N[u] \cap N[v] \cap ($○ $\cup$ ● $) \neq \emptyset$.
Let $E_I = \{\{u, v\} : u, v \in V_I$ and they are in conflict$\}$.
Finding a maximum independent set in G_I is equivalent to finding a maximum differential set in the remaining subgraph in step 6.

(1) The running time of the algorithm finding a maximum independent set in a graph of order n' is $\mathcal{O}^*(1.21^{n'})$ (a brief history of this result can be found in F. V. Fomin, D. Kratsch, Exact Exponential Algorithms, Texts in Theoretical Computer Science, Springer, 2010.).

(2) $n' \leq |$● $| + |$○ $| \leq \frac{\mu}{\omega_1} = 2\mu$. Hence, the time of step 6 is $\mathcal{O}^*(1.47^{\mu})$.

Theorem: MDS can be solved in time $\mathcal{O}(1.755^n)$ on arbitrary graphs of order n.

Algorithmic parameterized consequence

Corollary: k-DS can be solved in time $\mathcal{O}^*(16.65^k)$, using polynomial space only.

Can we do better?

Yes, we can!

However, we seem to be forced to mix kernelization ideas with branching.

Pending path rules

P_n: Path on n vertices
A path subgraph is called *pending* if it is connected only via a bridge to the remaining graph (component). It is *end-pending* if the path vertex that is endpoint of that bridge is also endpoint of the path. The bridge endpoint that is not on the path is called a *path-connecting vertex*.

Pending P_3 Rule: Remove an end-pending P_3 and decrement the parameter.

Lemma: If the Pending P_3 Rule does not apply, then we know the following:

- The only end-pending paths are P_1- and P_2-paths.
- There are no pending P_n for $n \geq 6$.
- Pending P_n for $n = 3, 4, 5$ are not end-pending.
- Path connecting vertices not belonging to some path are of degree ≥ 3.

Reduction rules

1. We can savely remove all isolated vertices and isolated edges.
2. Mark all path-connecting vertices that have at least two P_1- or P_2-neighbors.
3. Merge all marked vertices and replace the (at least two) path neighbors by four new neigbors w, x, y, z with (only further) edges wx, yz, updating the parameter k accordingly. Finally, unmark the marked vertex again.

Comments on correctness:
Marking should mean that (w.l.o.g.), a marked vertex should be put into the differential set.
Assume that, after merging, the only remaining marked vertex x has $\ell \geq 2$ "path neighbors" (P_1-paths or P_2-paths are counted).
Hence, when putting x (finally) into the differential set, k can be decreased by at least $\ell - 1$ due to these path neighbors.
Similarly, putting x into the differential set after the surgery decreases k by three.
So, we should update k as $k := k - \ell + 4$.
In the unmarked instance, x should be in a maximum differential set.

What do we get? A new way of using M&C!

Let (G, k) be the reduced instance.
If we shave off all leaves L and P_2-paths P, we get a graph G' with $\delta(G') \geq 2$.
If $|V(G')| \geq \frac{11}{3}k$, we can return YES.
Otherwise, $|V(G')| < \frac{11}{3}k$.
If $|L| \geq 2k$, we can form at least k S_2-stars, so return YES.
Otherwise, $|P| + |L| < 2k$.
If $P = \emptyset$, we start our branching algorithm with at most $3.\overline{6} \cdot k$ ○-vertices and at most $2k$ ●-vertices. Hence, the measure is at most $4.\overline{6} \cdot k$.

$\rightsquigarrow$ Running time: $\mathcal{O}^*(13.75^k)$.

This might also motivate a refined measure, as ω is playing a new role now.

$P \neq \emptyset$

We like to count end-pending P_2's also only by $\omega = .5$. This is justified by the following new branching rule on a P_2-path $x - y$ pending at z via the bridge yz.
This new branching rule is performed before the other branchings.

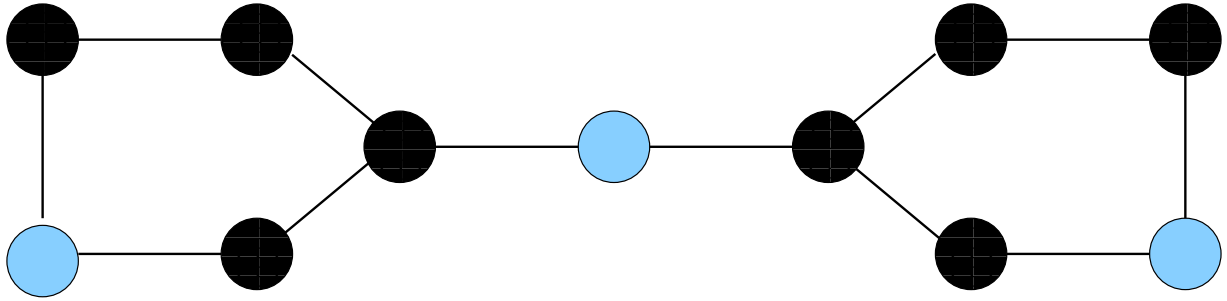
Either $y \in D$ (in that case, we can remove x, y, z) or $z \in D$, which allows to remove y, z. Doing this branching first, we can assume that z is a ○-vertex. In the second branch, z becomes ●.
Hence, this is a $(1.5, 1.5)$-branch, with a nice branching number of 1.5875.

After this first branching phase, we can of course color neighbors of ● by ● or ● to be consistent with the situation of the usual branching.

Comments on correctness:
Consider an optimum solution D where $z \notin D$. If $z \in C(D)$, then D is not optimum, as we can add y to D, yielding a new S_2-star. If $z \in B(D)$ because of an S_2-star, we can replace that S_2-star vwz by xyz, i.e., $D' = (D - w) + y$ gives the same differential. As we have no pendant P_5-paths, vwz is no pendant P_3-path. If $z \in B(D)$ because of an S_d-star with $d > 2$, we can turn this into an S_{d-1}-star by adding the star xyz, i.e., $D' = D + y$ gives the same differential.
In conclusion, if $z \notin D$, we can assume $y \in D$.

Subcubic graphs 1

Theorem: (B/F 2011) Let Γ be a subcubic graph of order $n \geq 12$ and minimum degree $\delta \geq 2$. If Γ does not have any pendant subgraph isomorphic to Γ^i as drawn below (pendant from the central vertex v_i), then $\partial(\Gamma) \geq \frac{2n}{7}$.



Since v_i is in $D \cup B(D)$ for any optimal D, these graphs can be preprocessed by reduction rules.

Subcubic graphs 2

Doing the "shaving" as before, we can derive a bound of $3.5 \cdot k$ for the "core". This core has hence (using results by Fomin / Høie) pathwidth of roughly $\frac{7}{12}k$. Putting back the pending paths, this estimate will not change.

On graphs of pathwidth at most d, the differential of a graph can be computed in time $\mathcal{O}^*(3^d)$. Hence, we derive:

Corollary: On subcubic graphs, it can be decided in time (roughly) $\mathcal{O}^*(3^{\frac{7}{12}k})$, i.e., in time $\mathcal{O}^*(1.8982^k)$, if a given graph has a differential of at least k. However, this approach uses exponential space.

A small catch

Merging of vertices doomed to be in the differential set is not possible.
However, we can do the following:

A P_3 or P_4 or P_5 pending at a non-endpoint can be first replaced by a P_3 pending at its midpoint and then (temporarily) be removed, decrementing the parameter.

This “rule” might trigger new rule applications.

A conceptual thought

We cannot claim to have derived a kernel in the classical sense, or at least this kernel (that we could obtain by putting back all paths that we cut off) looks uninteresting.

Rather, we construct an auxiliary graph with the property that a YES in the auxiliary graph **implies** a YES of the original instance.

Otherwise, we can bound some “important vertices” (what we called *core* above) so that these yield a bound on the pathwidth of the original (!) graph.

Notice that the pending P_3 considered on the previous slide can be “integrated” into the core, as the 3 vertices per parameter point are a good ratio.

Polynomial space

The pathwidth-based algorithm can be modified to use polynomial space only, at the expense of a bigger running time of $\mathcal{O}^*(3.61^k)$. Dyn.Prog. base 9, not 3

Notice that alternatively, we can refine our M&C analysis.
The second branch is never used, so we can modify / simplify our measure:

$\mu_3 = |\bigcirc| + \omega \cdot |\bullet| \leq n.$

Moreover, we could analyze the neighborhood further, for instance, if a $\bigcirc$ vertex has three $\bullet$ neighbors v_1, v_2, v_3, then each of these v_i must have one $\bigcirc$ neighbor u_i different from v, as otherwise v can be put into the differential set as a reduction rule. Even with these u_i, when v is NOT put into the differential set, then all u_i (which might be identical) are to be put in the differential set. This gives a branching vector of $(1 + 3\omega, 2 + \omega)$ in this case.
Other branches would be $(4, 1 - \omega)$, $(3 + \omega, 1 - \omega)$ or $(2 + 2\omega, 1 - \omega)$. With this analysis, $\omega = 0.15$ ist best, yielding a branching number of 1.6119. $P \neq \emptyset$ is also o.k.

This quick analysis would yield an $\mathcal{O}^*(1.62^n)$ exact and an $\mathcal{O}^*(8.57^k)$ parameterized algorithm;however, this is amenable to improved analysis of local situations.

Further achievements

Similar (bad) complexity results for determining the minimum ∂ set.
MAXSNP-classification for the related *enclaveless number* computation.
This parameter is also known as *nonblocker*:

F. Dehne, M. Fellows, H. F., E. Prieto, and F. Rosamond. NONBLOCKER: parameterized algorithmics for MINIMUM DOMINATING SET. In J. Štuller, J. Wiedermann, G. Tel, J. Pokorný, and M. Bielikova, editors, *Software Seminar SOFSEM*, volume 3831 of *LNCS*, pages 237–245. Springer, 2006.

Again, we only know of a combinatorial kernel.

Our exact algorithm transfers to the vertex-weighted case, bringing the problem closer to reality.
Also the kernel result transfers.

Questions 1 These apply both to k-MDS as to k-EN (nonblocker).

- Can we derive better kernels with more algorithmic arguments?

  The derived bounds are provably optimal as combinatorial results.

  Although the YES can be easily testified, see our combinatorial proof.

- Can we obtain better parameterized algorithms?

  For example, by genuinely FPT Measure&Conquer?

- Better kernels for smaller graph classes, for instance with degree bounds?

Questions 2

What other types of combinatorial results can be used to obtain kernel, or more generally, FPT results?

Ex.: Lozin / Rautenbach used Ramsey theory (no good constants to be expected....)

Can we make use, e.g., of the Moore bound?

Graphs of bounded diameter d and maximum degree Δ can only exist if the # vertices is bounded by a function $f_{\text{Moore}}(d, \Delta)$.

How to use the Moore bound

Closing the day with VERTEX COVER...

After applying Buss's rule, we know that $\Delta(G) \leq k$.

Not hard to see: If G admits a VC of size k, then $d(G) \leq 2k$.

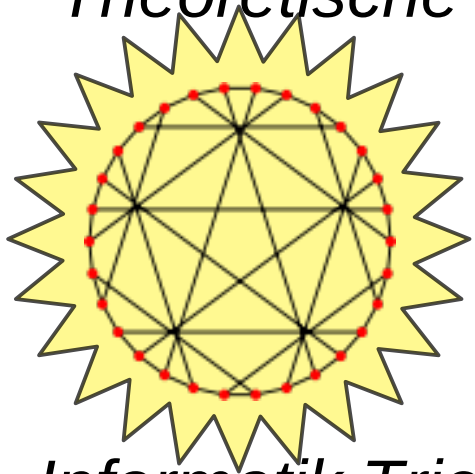
$\rightsquigarrow$ If G contains more than $f_{\text{Moore}}(2k, k)$ vertices, then NO.

Hence, we have a kernel (of size $f_{\text{Moore}}(2k, k)$).

Unfortunately, $f_{\text{Moore}}(d, \Delta) = 1 + \Delta(\sum_{i=0}^{d-1}(\Delta - 1)^i)$.

Thanks for your attention !

Theoretische



Informatik Trier