FPT-inspired approximation algorithms

Henning Fernau
Universität Trier, Germany
fernau@uni-trier.de

Workshop on Parametric Complexity; Singapore, August 2017
Overview on the talk

- Some origins
- Kernelization and approximation
- Local Ratio techniques
- More genuine reductions for approximation
- Conclusions
Some origins

- First approximation papers:

- \(\mathcal{NP}\)-completeness theory:
  early 1970s: Levin, Karp, Cook, …

- First \(\mathcal{FPT}\) results:
  K. Mehlhorn’s book on Data Structures and Algorithms 1984;
  Kuo / Fuchs 1987 “efficient for arrays with a moderate number of faulty cells”

- \(\mathcal{FPT}\) theory:
  Downey / Fellows 1992: Fixed parameter tractability and completeness (Congressus Numerantium, Vol. 87), as well as Conference on Structures in Complexity Theory
Some origins

Graham 1966:

In conclusion, one might ask just how "typical" the examples are for which $\omega' / \omega_o$ is close to the upper bound $2 - (1/n)$. While very little work has been done on this aspect, empirical results (using computer simulation (see Ref. 1)) indicate that examples in which $\omega' / \omega_o \geq 1.1$ are quite common.

IV. ACKNOWLEDGMENTS

I wish to acknowledge here the stimulating discussions I have had on this subject with S. Lin and with G. K. Manacher, who originally brought these questions to my attention.

REFERENCES

Approximation in a nutshell

Deals with **optimization problems** $P$: either minimization or maximization.

Recall: $\mathcal{NP}$-theory deals with **decision problems**.

Approximation algorithms produce solutions $S$, not necessarily optimal.

Usual quality measure of an approximation algorithm $A$ for $P$:
Worst-case ratio between the “size” of the obtained solution $S$ and the “size” of some best solution $S^*$, or its reciprocal value.
If this ratio is bounded by some constant, $A$ places $P$ within $\mathcal{APX}$. 
Parameterized complexity in a nutshell

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

Problem kernel of size $g(k)$, computable in time $q(n)$.

Characterization: Both approaches yield the same class $\mathcal{FPT}$.
Additive bonus: running time $\mathcal{O}(f(k) + p(n))$

Complexity class: $\mathcal{FPT}$

Standard approaches: search trees & kernelization

(Parameterized) language: $\mathcal{L}(\mathcal{P}) := \{(I, k) \mid (I, k) \text{ is a YES-instance of } \mathcal{P}\}$
Overview on the talk

- Some origins
- Kernelization and approximation
- Local Ratio techniques
- More genuine reductions for approximation
- Conclusions
Recipe for kernelization

- Take some reduction rules of your choice.
- Select some good mathematical meat, e.g., from your favorite combinatorics shop.
- Stir well for best algorithmic results.
Our question: Can we use this as a Recipe for approximation?

- Take some reduction rules of your choice.
- Select some good mathematical meat, e.g., from your favorite combinatorics shop.
- Stir well for best algorithmic results.
Our running example: Total Domination

Let $G = (V, E)$ be a graph. $D \subseteq V$ is a total dominating set if

$$\forall v \in V \exists u \in D \cap N(v)$$

Lots of things are known.

Our recipe might work.

(Henning / Yeo 2013)

Total Domination is a tough problem

Finding a total dominating set of size at most $k$ is $\mathcal{NP}$-complete.

Hard to tell: Is the following solution optimal?

Easy to check: No subset is a total dominating set (minimality).
Total Domination is a tough problem
even worse...

- Probably, there is no $\mathcal{FPT}$ algorithm for TDS (under standard parameterization).

- Probably, there is no constant-factor approximation algorithm (in poly-time).

Turn your attention around:
A **harmless set (with unanimity thresholds)** in a graph $G = (V, E)$ is a set $H \subseteq V$ such that $V \setminus H$ is a total dominating set.

We will see: HS is in $\mathcal{FPT}$ and in $\mathcal{APX}$.
News from the combinatorics shop

What bounds are known for the total domination number $\gamma_t(G)$ of a graph $G$ of order $n$, or for the harmless set number $hs(G)$?

This depends on the minimum degree $\delta(G)$.

<table>
<thead>
<tr>
<th>$\delta(G)$</th>
<th>$\gamma_t(G)$ bounds</th>
<th>$hs(G)$ bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0$</td>
<td>No solution exists.</td>
<td>No solution exists.</td>
</tr>
<tr>
<td>$\geq 1$</td>
<td>$\gamma_t(G) \leq \frac{2}{3} n$</td>
<td>$hs(G) \geq \frac{1}{3} n$</td>
</tr>
<tr>
<td>$\geq 2$</td>
<td>$\gamma_t(G) \leq \frac{4}{7} n$</td>
<td>$hs(G) \geq \frac{3}{7} n$</td>
</tr>
<tr>
<td>$\geq 3$</td>
<td>$\gamma_t(G) \leq \frac{1}{2} n$</td>
<td>$hs(G) \geq \frac{1}{2} n$</td>
</tr>
</tbody>
</table>

∽ Easy kernel exploit: If $k < \frac{1}{3} n$, answer YES $\rightarrow$ kernel size $3k$.

Less clear for approximation of the Maximum Harmless Set problem.

Henning Fernau: FPT-inspired approximation algorithms; Workshop on Parametric Complexity 2017
An approximation algorithm (Bazgan & Chopin, Discrete Optimization 2014)

Lemma: Let $G = (V, E)$ and $G' = (V, E')$ be graphs with $E' \subseteq E$. If $D' \subseteq V$ is a total dominating set of $G'$, then $D'$ is a total dominating set of $G$.

1. Let $G = (V, E)$ be (w.l.o.g.) a connected graph of order $n$.
2. Compute a spanning tree $T = (V, E_T)$ of $G$.
3. Compute an optimum TDS $D_T$ for $T$ (by dynamic programming).
4. $H := V \setminus D_T$ is a harmless set for $G$ satisfying

$$|H| = hs(T) \geq \frac{1}{3}n \geq \frac{1}{3}|H^*|,$$

where $H^*$ is a maximum harmless set for $G$.

$\Rightarrow$ Thm. Maximum Harmless Set can be approximated up to a factor of 3.
Possible improvements

**Aim:** Get smaller kernels / better approximation

Find a rule to cope with vertices of degree one.

Namely, with a minimum degree of two, we might improve on our previous bounds, getting the factor of 3 down to $\frac{7}{3}$ (or better?).

- No general degree-one rule is known.
- Even if: How could this be used for approximation?
- Even then, what about vertices of degree two?
- Or, are there nicer combinatorial results?
Useful reductions for approximation

• If $u$ and $v$ are two vertices of degree one in $G$ with the same neighbor $w$, then delete $u$, yielding graph $G'$. This creates a bijection between leaves and their support vertices.

• Assume that $G = (V, E)$ is a graph that contains a path $x - u - v - w - y$, where $u, v, w$ are three consecutive vertices of degree two, where $\delta(y) \geq 2$. Then, construct the graph $G'' = (V', E')$ by

  – deleting $x, u, v, w$ and
  – connecting $y$ to all vertices in $N(x) \setminus \{u\}$.

This rule is reminiscent of the folding rule for Vertex Cover.

• We do need more rules in the end . . .
Rules at work: Identify a local situation
Rules at work: ... and apply the first rule
Rules at work: Identify a local situation
Rules at work: ... and apply the second rule
Important properties

Both rules satisfy the following properties (as we can show):

- Both can be implemented efficiently.

- If $G$ gets transformed to $G'$ and if we could obtain a factor-$\alpha$ approximation for $G'$, then we could also produce a factor-$\alpha$ approximation for $G$. This includes the exact case, when $\alpha = 1$. 
What about the reduced instance?

The reduced instance satisfies:

- There are no three consecutive vertices of degree two.
- No vertex of degree one has a twin.

But how could we make use of these properties?

The combinatorics shop offers (Lam and Wei 2007):

**Thm.** If a graph $G'$ of order $n$ has no vertices of degree one and no three consecutive vertices of degree two, then $\gamma_t(G') \leq \frac{1}{2} n$.

How can we get rid of all the leaves?
Another idea …
Further modify the graph, so that the new graph satisfies the conditions set up by Lam and Wei.

- Introduce two copies of the reduced graph.
- Delete all leaves and introduce edges between the leaf-neighbors so that degrees are maintained.
- Apply the algorithmic result of Lam and Wei.
- Use this (approximative) solution to get, first a 2-approximative solution of the reduced graph, and then (undoing the reductions) a 2-approximative solution of the original graph.

Working on the example
We obtain a relatively small total dominating set
A missing bit

**Theorem 1.** Let $G$ be a graph of order $n_G$ and let $H$ be a graph of order $n_H$ obtained from $G$ by deleting $d$ vertices and possibly adding some edges. Let $D_G$ and $D_H$ be TDS solutions of $G$ and $H$ s.t. $a = |D_G| - |D_H| \leq d$.

If $|D_H| \leq c \cdot n_H$ and $d \leq \gamma_t(G')$, then $V(G') \setminus D_G$ is a harmless set of $G$ whose size $n_G - |D_G|$ is within a factor of $(1 - c)^{-1}$ from optimum.

**Proof.** As $n_H = n_G - d$, $|D_G| = |D_H| + a \leq c(n_G - d) + a = cn_G + (a - cd) \leq cn_G + d - cd = cn_G + (1 - c)d \leq cn_G + (1 - c)\gamma_t(G)$. Hence, $n_G - |D_G| \geq n_G - cn_G - (1 - c)\gamma_t(G) = (1 - c)(n_G - \gamma_t(G))$. This immediately yields an approximation factor of $(1 - c)^{-1}$. \[\square\]
An approximate solution for the two copies of course gives an approximate solution for each component.
Undoing the reductions

This solution is guaranteed to be 2-approximate.
This also shows that our previous solution was NOT optimal.
Summary: When does this approach work?

- Introduce a notion of approximation-preserving reduction. Often, these rules will also work as kernelization rules.

- Look for algorithmic combinatorial results. This aspect was irrelevant for kernelization.

- If “no elements” (vertices) give a trivial feasible solution to the maximization problem, then analogues like the “missing bit” can be shown and employed.

- This does NOT work out for minimization problems.

- In the paper: Harmless Set, $\kappa$-Nonblocker, Differential
Some maths that we hid so far... (for maximization problems)

An $\alpha$-preserving reduction, with $\alpha \geq 1$, is a pair of mappings $\text{inst}_P : I_P \to I_P$ and $\text{sol}_P$ which, given $y' \in \text{SOL}_P(\text{inst}_P(x))$, produces some $y \in \text{SOL}_P(x)$ s.t. there are $a, b \geq 0$ satisfying $a \leq \alpha \cdot b$ and the following:

1. $m^*_P(\text{inst}_P(x)) + a \geq m^*_P(x)$,

2. for each $y' \in \text{SOL}_P(\text{inst}_P(x))$, the corresponding solution $y = \text{sol}_P(y')$ satisfies: $m_P(\text{inst}_P(x), y') + b \leq m_P(x, y)$.

**Thm.** If the pair $(\text{inst}_P, \text{sol}_P)$ describes an $\alpha$-preserving reduction and if, given some instance $x, y' \in \text{SOL}_P(\text{inst}_P(x))$ is an $\alpha$-approximate solution for $\text{inst}_P(x)$, then $y = \text{sol}_P(y')$ is an $\alpha$-approximate solution for $x$.

This property needs to be shown for each reduction rule.
The differential of a graph


Let $G = (V, E)$ be a graph. For $D \subseteq V$, $B(D) := N(D) \setminus D$ and $C(D) := V \setminus (D \cup B(D))$.

The *differential of $D$* is $\partial(D) = |B(D)| - |D|$ and the *differential of $G$*, written $\partial(G)$, is equal to $\max\{\partial(D) \text{ such that } D \subseteq V\}$.

Mashburn, Haynes, Hedetniemi (2x) and Slater, *Utilitas Mathematica*, 2006.
Here, $\partial(D) = 9 - 3 = 6$. Is this maximal?

**Theorem 2.** For any connected graph $G$ of order $n \geq 3$, $\partial(G') \geq n/5$.

**Theorem 3.** For any connected graph $G$ of order $n$ that has minimum degree two, $\partial(G') \geq \frac{3n}{11}$, apart from five exceptional graphs listed below.

Red dotted edges may be present or not.
Combinatorial experiences with “dual parameters”: Roman domination

A \textit{Roman domination} function of a graph \( G = (V, E) \) is \( R : V \to \{0, 1, 2\} \) with

\[
\forall v \in V : R(v) = 0 \Rightarrow \exists x \in N(v) : R(x) = 2.
\]

\textbf{ROMAN DOMINATION (ROMAN)}

\textbf{Given:} A graph \( G = (V, E) \)

\textbf{Parameter:} a positive integer \( k \)

\textbf{Question:} Is there a \textit{Roman domination} function \( R \) such that

\[
R(V) := \sum_{x \in V} R(x) \leq k?
\]

\textbf{Graph parameter:} \( \gamma_R \)

\textbf{F. International Journal of Computer Mathematics 2008}

\textbf{Henning Fernau: FPT-inspired approximation algorithms; Workshop on Parametric Complexity 2017}
Historical Background The Roman Empire in the times of Constantine
A side-track: Who was Constantine?
You can find Constantine’s palace in Trier
A pure graph model
Constantine’s solution

![Graph of regions connected with distances](image)
Britain in danger
Another solution
Combinatorial experiences with “dual parameters”: Roman domination

Bermudo, F., Sigarreta in Applicable Analysis and Discrete Mathematics 2014:

**Theorem 4.** If $G$ is a graph of order $n$, then $\gamma_R(G) = n - \partial(G)$.

Hence, our previous combinatorial result Thm. 3 also follows from:

We simply forgot about parameterized dualization!

**Theorem 5.** *(Bermudo, F., TCS 2015)* There exists a kernelization algorithm for DIF that runs in linear time and that, given $(G, k)$, provides a kernel $(G', k)$ with $|V'| \leq 4k$, or that directly (and correctly) answers YES.

**Notabene:** The same reasoning can be used to prove a factor-4 approximation.

**Question:** Can we make full algorithmic use of Thm. 3?

Henning Fernau: FPT-inspired approximation algorithms; Workshop on Parametric Complexity 2017
Sketch of the Leaf Reduction

Sketch of the Hair and the Hair-Leaf Reductions.
Reductions for DFFERENTIAL

1. **Leaf Reduction.** If there are two leaves adjacent to the same vertex, then connect these leaves by an edge.

2. **Hair Reduction.** If there are two hairs connected to the same vertex, then remove the two hair leaves.

3. **Hair-Leaf Reduction.** If there is a leaf and a hair connected to the same vertex, then remove the hair leaf.

4. **Long Hair Reduction.** If there is a long hair $u - v - w - \cdots$, then remove $u, v, w$.

5. **Neighbor Hair Reduction.** If there is a hair $u - v - \cdots$ connected to a vertex $w$ and another hair $u' - v' - \cdots$ connected to a neighbor $w'$ of $w$, then remove the edge $ww'$. 
More to it . . .

- The presented rules (only) guarantee a factor-4 approximation.
- On top, *as usual with reductions*, they provide **additional structure**.

1. To each vertex, at most one leaf or one hair is attached, but not both together.
2. If we remove all leaves and all hairs from $G$, then the remaining graph, henceforth called *nucleus*, has minimum degree of at least two.
3. If a hair is attached to a vertex $u$ in the nucleus, then no hair is attached to any neighbor of $u$ within the nucleus.
Making use of local structures

Not diving into details here, but:
We could present *local improvement rules* that finally allowed us to conclude:

**Theorem 6.** \textsc{Differential} is factor-$\frac{11}{3}$ polynomial-time approximable.

Better combinatorial bounds for Roman Domination for 2-connected graphs and for graphs of minimum degree at least three by Liu and Chang (2012).

Can these bounds turned into kernelization or approximation algorithms?
To take away

- Reduction rules can be a useful tool for approximation algorithms.
- They might be a nice bridge to parameterized approximation. This is still to be shown.
- But, they are also useful for poly-time approximation. We have proven several combinatorial approximation algorithms here.
- Reduction rules are easy to implement. The final combinatorial bit could be more complicated. Often: Greedy combined with local search.
Overview on the talk

- Some origins
- Kernelization and approximation
- Local Ratio techniques
- More genuine reductions for approximation
- Conclusions
The Local Ratio technique

- Meant to easily analyze approximation algorithms that do local transformations.
- In the end, these transformations can be viewed as reduction rules.
- Hence, this method is easy to implement.
- Sometimes allows a combinatorial interpretation of LP-based methods.
- Self-monitoring could be implemented on top.
  But this is not mentioned in the classical papers.
**Vertex Cover again**

**Local transformation:**
Take any edge and put both endpoints in the approximate cover.

Doing this exhaustively will yield a vertex cover $C$.

As for each edge, one of the endpoints must be in the cover, it is easily seen by induction that the approximate cover $C$ has at most twice as many vertices as an optimum one.
A nice summary of the method


The main idea can be phrased as follows: iteratively, pay two dollars (at most) to reduce the total optimum by one dollar (at least), so the rate of payment is no more than twice the rate of the optimum reduction. This implies a total payment (i.e., approximation cost) \( \leq \) twice the optimum cost.

This paper also contains a factor-2 approximation algorithm for **Feedback Vertex Set**.
Primal / Dual on top

LP methods are quite common to design approximation algorithms.

These methods often lack a direct combinatorial interpretation.

Notable exception: The primal / dual method, see:


Overview on the talk

- Some origins
- Kernelization and approximation
- Local Ratio techniques
- More genuine reductions for approximation
- Conclusions
**Vertex Cover again**

Now: Focus on graphs of bounded average degree

Importance: a simple model of “realistic graphs”
Far more important than graphs of bounded degree, but far less studied . . .

Easy to generate random instances, as an upper bound on # edges is known.

We start with an example, observing *rules at work*. 
Example: Dealing with leaves, take credit of two edges
Example: Folding subsequent vertices of deg. 2, take credit of two edges
**Example:** Dealing with a vertex of deg. 2 in a triangle, no credit needed
**Example:** Triangle situations may propagate . . . Where? Why? We remove 5 vertices and 8 edges and hence need a credit of two edges.
**Example:** Deal with vertex $v = 23$ of deg. 3 s.t. $\exists u \in N(v) : N(v) \subseteq N[u]$. Mostly, we can even pay back to the bank in this case. For instance, in this example, removing one vertex and four edges is paying back.
**Example:** Triangle Rule plus propagation; one edge is missing

A 4-regular graph remains, $K_4 \times K_4$, easy to solve.
Example: Our solution: it is off by at most one . . .
Example: A slightly better optimum solution
Example: Comparing solutions
Our reduction rules

Small-Degree rules (as traditional) . . .

**Degree-0 Rule:** Delete isolated vertices; they do not go into the cover.

**Degree-1 Rule:** If \( v \) has a neighbor of degree one, put \( v \) into the cover and decrement the parameter.

*Caution:* When aiming at an \( \alpha \)-approximation for average-degree-bounded graphs, this is a dangerous rule, as it will increase the average degree. Therefore, we will avoid this case when stating our theorem, but “in practice”, this is apparently no problem.

Notice that these rules are *exact rules*.
The bank business

Problem: Deleting low-degree vertices may destroy a given upper-bound on the average degree of a graph, as “too few” edges are removed this way.
Solution: When working from dealing with low-degree vertices (with nice approximation rules) to higher-degree vertices, we take some credit from the bank. They will be payed back when processing vertices of higher degree.

This strategy allows us to relax the upper-bound condition “on the way”.

Otherwise, we often use the following:

Lemma 1. In a graph $G$ with average degree $d_{avg}$, if one deletes $x$ vertices and at least $\frac{d_{avg}}{2}x$ edges, the remaining graph $G'$ has average degree $d'_{avg} \leq d_{avg}$.

Henning Fernau: FPT-inspired approximation algorithms; Workshop on Parametric Complexity 2017
What we want to show with a rule-based algorithm

**Theorem 7.** For any graph $G$ with average degree $d_{\text{avg}} \leq 4$ and minimum degree $\delta \geq 2$, there exists a polynomial-time factor-1.5 approximation algorithm for MINIMUM VERTEX COVER.

The rules have to maintain

- the approximation ratio and
- the degree constraints
Approximation preserving reductions

Warm-up: Exact Reduction rules

Degree-0 Rule: Delete isolates

Degree-1 Rule:

Degree-2 Rule:

Observe: Increase of degrees is possible.
Approximation preserving reductions The case $\frac{3}{2}$:

Triangle Rule well known:

Degree-3 Rule: plus one
The bank business or: How to maintain degree constraints

The lower degree constraint can be maintained by executing the Degree-1 Rule implicitly.

A worsening step puts two adjacent vertices into the cover (this has approximation ratio 2).

If we put $p$ vertices into the cover due to exact reduction rules, and if this reduces the total number of vertices in the graph by $q$ and the total number of edges by $r$, then we can delete also $p$ pairs of adjacent vertices by worsening steps and we will do this such that there are at least $2(q + 2p) - r$ edges incident to at least one of the vertices in the $p$ pairs.

Indeed, if we delete $q + p$ vertices and at least $2(q + 2p)$ edges, by Lemma 1 it follows that the average degree remains below 4.

Example: If a Degree-1 Rule puts $v$ with $d(v) = 4$ into the cover, 4 edges and 2 vertices are removed, which is fine. If $d(v) = 2$, then two edges are missing. We take a credit from the bank. When later removing, say, a pair of adjacent vertices of degrees 3 and 4, then we have (altogether) deleted 4 vertices and $2 + 6 = 8$ edges, paying back this credit.

Basic Proof Idea: Show that rules are always applicable if bank is empty or that we can alternatively pay back some credit.

For approximation ratio 1.5, we can always afford a worsening step to pay back two.
Folding Rule

A famous rule to deal with vertices of degree two. (Stege 2000; Chen, Kanj, Xia 2001)

We remove 2 vertices and 2 edges from the graph.
Take a credit of two from the bank.
After an appropriate worsening step, we will have removed 4 vertices and at least 8 edges.
We also put 3 vertices in the cover, knowing that any best solution needs at least 2.
**Example:** Exact rules can compensate degree constraints: Folding and Degree-3 Rule together remove 3 vertices and $6 = 2 + 4$ edges.
A rough outline of our approximation algorithm

1. If there is a vertex $v$ of degree two in $G$, process $v$, possibly taking credit from the bank, and recurse.

2. If there is no vertex of degree two in $G$ but a vertex $v$ of degree three, process $v$, possibly taking credit from the bank, and recurse.

3. If $G$ has minimum degree four and if the bank has given some credit, pay back by performing a worsening step and recurse.

There are two ways of leaving this algorithm:

- If the graph $G$ has minimum degree four and the bank is empty, then we can use the known approximation algorithm mentioned in Lemma 2 to obtained the desired approximation ratio, as we can show that the fact that if the bank is empty, then the average degree of $G$ is bounded above by four, which means that $G$ is 4-regular.

- If we leave this algorithm with a bank still owing money, this means (as we can show) that the graph has no edges. Due to the minimum degree invariant, this also means that the graph has no vertices. So, the whole graph has been processed.
Experimental insights from thousands of random graphs (Florian KK)

- Graphs of avg. deg. two are always optimally solved.
  No surprise, but good test for the implementation.

- Graphs of avg. deg. 2.5 are always optimally solved.
  More of a surprise, as counterexamples can be constructed.

- Graphs of avg. deg. 3 are mostly optimally solved.
  All factors that the algorithm could guarantee were < 1.01.

- Graphs of avg. deg. 3.5 are never optimally solved (according to guarantee).
  Nearly all factors that the algorithm could guarantee were < 1.05.

- Graphs of avg. deg. 4 give solutions with a guaranteed factor < 1.1.

- Some reduction rules never (or hardly ever) apply.
More experimental insights from thousands of random graphs

- The algorithm was also run on graphs of higher average degree.
- Our theorem does not guarantee that such graphs can be solved.
- However, in practice graphs up to average degree of eight are completely solved by the algorithm.
- Only, the approximation factor guaranteed by the algorithm gets closer to 1.5.
- Still probably a nice preprocessing for parameterized approximation algorithms for all realistic graphs.
Previous results


**Lemma 2.** There exist polynomial time approximation algorithms for **MINIMUM VERTEX COVER** for any graph with maximum degree 4 that guarantee factors of $3/2$ (Hochbaum) or even of $9/7 + \varepsilon$, $\varepsilon > 0$ (Berman / Fujito).

Surprisingly, not much known for bounded average degree.


They get $\frac{4d_{avg}+1}{2d_{avg}+3} = 1.54$ for $d_{avg} = 4$.

**IMHO,** these types of results should be more prominent in the literature.
Conclusions

Advantages of rule-based approach

- Easy to implement; the bank can keep track both of the edges to delete (in addition) and of the number of affordable worsening steps.
- Easy to combine with other (exact) approaches.
- Possible to monitor the approximation factor.
- Last two items allow trade-offs of precision and running time.
Prospects

- Via Local Ratio, rule-based approach encaptures some LP techniques. Can this be further extended?
- Implement self-monitoring on top would sell well to practitioners.
- Implementations and practical comparisons of algorithms should be useful.
Further combinations

- Combine self-monitoring approximation algorithms with exact methods like search trees to speed up exact algorithms in practice.

- There are many more methods in approximation that $\mathcal{FPT}$ should make use of, just recall:

Thanks for your attention!