A Practical Analysis of Kernelization Techniques for the Maximum Cut Problem

NII Shonan Meeting | March 5, 2019
Damir Ferizovic, Sebastian Lamm, Matthias Mnich, Christian Schulz, Darren Strash
Max-Cut: Definition and Example

- Given $G = (V, E)$, find $S \subseteq V$ such that $|E(S, V \setminus S)|$ is maximum.
- Notation: $mc(G) := \max_{S \subseteq V} |E(S, V \setminus S)|$
Max-Cut: Definition and Example

- Given $G = (V, E)$, find $S \subseteq V$ such that $|E(S, V \setminus S)|$ is maximum.
- Notation: $mc(G) := \max_{S \subseteq V} |E(S, V \setminus S)|$

Motivation
Analysis of Existing Work
Our Contribution
Results
Conclusion

Sebastian Lamm – Kernelization and Max-Cut
March 5, 2019
Max-Cut: Definition and Example

- Given $G = (V, E)$, find $S \subseteq V$ such that $|E(S, V \setminus S)|$ is maximum
- Notation: $mc(G) := \max_{S \subseteq V} |E(S, V \setminus S)|$

Motivation
Analysis of Existing Work
Our Contribution
Results
Conclusion
March 5, 2019 2/20
Max-Cut: Definition and Example

- Given $G = (V, E)$, find $S \subseteq V$ such that $|E(S, V \setminus S)|$ is maximum.
- Notation: $mc(G) := \max_{S \subseteq V} |E(S, V \setminus S)|$

S = \{v_2, v_3, v_4, v_6\}
$\rightarrow |E(S, V \setminus S)| = 7$
$\rightarrow$ maximum?
Max-Cut: Definition and Example

- Given $G = (V, E)$, find $S \subseteq V$ such that $|E(S, V \setminus S)|$ is maximum
- Notation: $mc(G) := \max_{S \subseteq V} |E(S, V \setminus S)|$

Motivation

Analysis of Existing Work

Our Contribution

Results

Conclusion

Sebastian Lamm – Kernelization and Max-Cut

March 5, 2019
Max-Cut: Definition and Example

Given $G = (V, E)$, find $S \subseteq V$ such that $|E(S, V \setminus S)|$ is maximum

Notation: $mc(G) := \max_{S \subseteq V} |E(S, V \setminus S)|$

$S = \{v_3, v_4, v_6, v_7\}$

$\rightarrow |E(S, V \setminus S)| = 8$
Max-Cut: Definition and Example

- Given $G = (V, E)$, find $S \subseteq V$ such that $|E(S, V \setminus S)|$ is maximum.
- Notation: $mc(G) := \max_{S \subseteq V} |E(S, V \setminus S)|$

Motivation
Analysis of Existing Work
Our Contribution
Results
Conclusion

Sebastian Lamm – Kernelization and Max-Cut
March 5, 2019

$S = \{v_3, v_4, v_6, v_7\}$
$\rightarrow |E(S, V \setminus S)| = 8$
$\rightarrow$ maximum?
Max-Cut: Importance of Studying it

- Member of Karp’s 21 **NP-complete** problems
- Used in...

**Circuit design**  **Statistical physics**  **Social networks**
Kernelization: Definition and Example

- Kernelization: Compress graph while preserving optimality

\[ G_0 = G : \]

Motivation

Analysis of Existing Work

Our Contribution

Results

Conclusion

Sebastian Lamm – Kernelization and Max-Cut

March 5, 2019
Kernelization: Definition and Example

- Kernelization: Compress graph while preserving optimality

\[ G_0 = G : \]

\[ \begin{align*}
  v_3 \\
  v_1 & \quad v_4 \\
  v_6 & \quad v_7
\end{align*} \]

\[ \begin{align*}
  v_2 & \quad v_5
\end{align*} \]
Kernelization: Definition and Example

- Kernelization: Compress graph while preserving optimality

\[ G_1 : \]

\[ mc(G_0) = mc(G_1) + 2 \]
Kernelization: Definition and Example

Kernelization: Compress graph while preserving optimality

\[ G_1 : \]

\[ mc(G_0) = mc(G_1) + 2 \]
\[ mc(G_1) = 6 \]
Kernelization: Definition and Example

Kernelization: Compress graph while preserving optimality

\[ G_0 = G : \]

\[ mc(G_0) = 6 + 2 = 8 \]
Max-Cut: Current Research on Kernelization

- Previous work mostly of theoretical nature
  - Analyze problem $k + mc$-lowerbound$(G)$
  - Different reformulations (Etscheid and Mnich 2018, Madathil, Saurabh, and Zehavi 2018, Prieto 2005)

- Research on practicality missing
  - Present for other problems
  - INDEPENDENT SET, VERTEX COVER (Hespe, Schulz, and Strash 2018, Akiba and Iwata 2016)
Max-Cut: Reduction Rule 8 in Etscheid and Mnich 2018
Max-Cut: Reduction Rule 8 in Etscheid and Mnich 2018

Motivation

Analysis of Existing Work

Our Contribution

Results

Conclusion
Max-Cut: Reduction Rule 8 in Etscheid and Mnich 2018

1. \( N_G(x) \cap S = N_G(X) \cap S \)
2. \( |X| > \frac{|K| + |N_G(X) \cap S|}{2} \geq 1 \)
Max-Cut: Reduction Rule 8 in Etscheid and Mnich 2018

1. $N_G(x) \cap S = N_G(X) \cap S \checkmark$

2. $|X| > \frac{|K| + |N_G(X) \cap S|}{2} \geq 1 \times$

Motivation
Analysis of Existing Work
Our Contribution
Results
Conclusion

March 5, 2019

Sebastian Lamm – Kernelization and Max-Cut
Max-Cut: Reduction Rule 8 in Etscheid and Mnich 2018

\[ N_G(x) \cap S = N_G(X) \cap S \]
\[ |X| > \frac{|K| + |N_G(X) \cap S|}{2} \geq 1 \]
Max-Cut: Reduction Rule 8 in Etscheid and Mnich 2018

\[ N_G(x) \cap S = N_G(X) \cap S \]
\[ |X| > \frac{|K| + |N_G(X) \cap S|}{2} \geq 1 \]
- Kernelization mostly **driven by rule 8**

- Weak-points in practice
  - Reliance on clique-forest
  - Parameter $k$ large in practice
    - **Kernel size $O(k)$ too large**
    - $O(k \cdot |E(G)|)$ time too slow
Overview of Our Contributions

- Implemented and evaluated work of Etscheid and Mnich 2018
- **Generalized existing reduction rules**
  - Rules not dependent on a subgraph anymore
- **Developed new reduction rules**
  - Simplistic but significant improvement in practice
  - Identified inclusions
- **Efficient implementation**
  - Timestamping system
- Benchmark over a variety of instances
Overview of Our Contributions

- Five new unweighted reduction rules
  - Rule to compress induced 3-paths
  - Two rules reducing cliques \((R8, S2)\)
  - Two antagonizing rules – merge and divide of cliques

- Briefly investigated: Weighted path compression
External/Internal Vertices

\[ S \iff \in C_{ext}(G[S]) \iff \in C_{int}(G[S]) \]

Motivation
Analysis of Existing Work
Our Contribution
Results
Conclusion

Sebastian Lamm – Kernelization and Max-Cut
March 5, 2019
Rule Generalization: R8
– “Sharing Adjacencies”
Rule Generalization: R8
– “Sharing Adjacencies”
Rule Generalization: R8
– “Sharing Adjacencies”

1. \( N_G(X) \cup X = N_G(x) \cup \{x\} \)
2. \( |X| > \max\{|N_G(X)|, 1\} \)

Motivation
Analysis of Existing Work
Our Contribution
Results
Conclusion

Sebastian Lamm – Kernelization and Max-Cut

March 5, 2019
Rule Generalization: R8 – “Sharing Adjacencies”

Motivation

Analysis of Existing Work

Our Contribution

Results

Conclusion

1. \( N_G(X) \cup X = N_G(x) \cup \{x\} \) ✓
2. \( |X| > \max\{|N_G(X)|, 1\} \) ✓
Rule Generalization: R8
– “Sharing Adjacencies”

Motivation

Analysis of Existing Work

Our Contribution

Results

Conclusion

Sebastian Lamm – Kernelization and Max-Cut

March 5, 2019

1. $N_G(X) \cup X = N_G(x) \cup \{x\}$ ✓
2. $|X| > \max\{|N_G(X)|, 1\}$ ✓
New Reduction Rule: S2
– “Semi-Isolated Cliques”

Clique $G[S]$ with $|C_{ext}(G[S])| \leq \left\lceil \frac{|S|}{2} \right\rceil$
New Reduction Rule: S2 – “Semi-IsolatedCliques”

\[
\text{Clique } G[S] \text{ with } |C_{\text{ext}}(G[S])| \leq \left\lfloor \frac{|S|}{2} \right\rfloor
\]
New Reduction Rule: S2
– “Semi-Isolated Cliques”

Clique \( G[S] \) with \( |C_{ext}(G[S])| \leq \left\lceil \frac{|S|}{2} \right\rceil \) ✓
New Reduction Rule: S2
– “Semi-Isolated Cliques”

Clique $G[S]$ with $|C_{ext}(G[S])| \leq \left\lceil \frac{|S|}{2} \right\rceil$ ✓
New Reduction Rule: S2
– “Semi-Isolated Cliques”

=Clique $G[S]$ with $|C_{ext}(G[S])| \leq \left\lfloor \frac{|S|}{2} \right\rfloor$

Motivation
Analysis of Existing Work
Our Contribution
Results
Conclusion
Sebastian Lamm – Kernelization and Max-Cut
Techniques Used for Performance

- **Avoid time-intensive checks**
  - Vertex $v$ internal in clique: $\forall w \in N_G(v) : \text{Deg}(v) \leq \text{Deg}(w)$

- **Avoid checking the same reduction rules**
  - Timestamp of most recent change in neighborhood for each vertex
  - Keep timestamp $T$ for each rule:
    - All vertices with timestamp $\leq T$ already processed
  - Update vertex on change

![Diagram](image)

$T - 4$  $T - 1$  $T + 3$

$v_3$  $v_1$  $v_2$
Experiments on KaGen Graphs

- Random graphs by KaGen, 150 per each graph type. \( |V| = 2048 \)
- **Total runtime: 16 sec.** (68 min. by Etscheid and Mnich 2018!)

Kernelization efficiency for KaGen graphs; metric: \( e(G) = 1 - \frac{|V(G_{\text{Kor}})|}{|V(G)|} \)

Motivation

Analysis of Existing Work

Our Contribution

Results

Conclusion

Sebastian Lamm – Kernelization and Max-Cut

March 5, 2019
Experiments on KaGen Graphs

- Improvement over Etscheid and Mnich 2018. $|V| = 2048$
- Discrepancy between theory and practice

Absolute difference in efficiency: $e_{\text{absDiff}} = e(G_{\text{new}}) - e(G_{\text{old}})$
Experiments on KaGen Graphs

- Improvement on our results with weighted path compression

Absolute difference in efficiency: \( e_{\text{absDiff}} = e(G_{\text{newWeighted}}) - e(G_{\text{new}}) \)

Motivation

Analysis of Existing Work

Our Contribution

Results

Conclusion

Sebastian Lamm – Kernelization and Max-Cut

March 5, 2019

16/20
## Experiments – BiqMac Solver

| Name           | $|V(G)|$ | $deg_{avg}$ | $e(G)$ | $T_{BM}(G)$ | $T_{BM}(G_{ker})$ |
|----------------|------|------------|--------|-------------|-------------------|
| ego-facebook   | 2888 | 1.03       | 1.00   | -           | 0.01 [∞]          |
| road-euroroad  | 1174 | 1.21       | 0.79   | -           | -                 |
| rt-twitter-copen | 761 | 1.35       | 0.85   | -           | 1.77 [∞]          |
| bio-diseasome  | 516  | 2.30       | 0.93   | -           | 0.07 [∞]          |
| ca-netscience  | 379  | 2.41       | 0.77   | -           | 0.67 [∞]          |
| g000302        | 317  | 1.50       | 0.21   | 1.88        | 0.74 [2.53]       |
| g001918        | 777  | 1.59       | 0.12   | 31.11       | 17.45 [1.78]      |
| g000981        | 110  | 1.71       | 0.28   | 531.47      | 21.53 [24.68]     |
| imgseg_105019  | 3548 | 1.22       | 0.93   | f           | 13748.62 [∞]      |
| imgseg_35058   | 1274 | 1.42       | 0.37   | -           | -                 |
| imgseg_374020  | 5735 | 1.52       | 0.82   | f           | -                 |

Times in seconds. 10 hour time limit with 5 iterations.

---

**Motivation**

- Our Contribution

**Analysis of Existing Work**

- Results

**Conclusion**

Sebastian Lamm – Kernelization and Max-Cut

March 5, 2019
| Name                  | $|V(G)|$ | $\text{deg}_{\text{avg}}$ | $e(G)$ | $T_{LS}(G)$ | $T_{LS}(G_{\text{ker}})$ |
|-----------------------|-------|---------------------|-------|-------------|---------------------|
| ego-facebook          | 2888  | 1.03                | 1.00  | 20.09       | 0.09 [228.91]       |
| road-europaroad       | 1174  | 1.21                | 0.79  | -           | -                   |
| rt-twitter-copen      | 761   | 1.35                | 0.85  | -           | 834.71 [∞]          |
| bio-diseasome         | 516   | 2.30                | 0.93  | -           | 4.91 [∞]            |
| ca-netscience         | 379   | 2.41                | 0.77  | -           | 956.03 [∞]          |
| g000302               | 317   | 1.50                | 0.21  | 0.58        | 0.49 [1.17]         |
| g001918               | 777   | 1.59                | 0.12  | 1.47        | 1.41 [1.04]         |
| g000981               | 110   | 1.71                | 0.28  | 10.73       | 4.73 [2.27]         |
| imgseg_105019         | 3548  | 1.22                | 0.93  | 234.01      | 22.68 [10.32]       |
| imgseg_35058          | 1274  | 1.42                | 0.37  | 34.93       | 24.71 [1.41]        |
| imgseg_374020         | 5735  | 1.52                | 0.82  | 1739.11     | 72.23 [24.08]       |

Times in seconds. 10 hour time limit with 5 iterations.
Experiments – Localsolver

Solution size over time by Localsolver: initial vs. kernelized

Motivation
Analysis of Existing Work
Our Contribution
Results
Conclusion

Sebastian Lamm – Kernelization and Max-Cut

March 5, 2019
Conclusion

Summary

- Previous work: Good in theory, bad in practice
- Set of new (unweighted) reduction rules
- Sparse graphs highly reducible
- Significant benefits for existing solvers

Future Work

- Add parallelism?
- New (weighted) reduction rules?
- Hybrid approach: Use solver for reductions?