Partial Optimality and Fast Lower Bounds for Weighted Correlation Clustering Supplementary Material

Jan-Hendrik Lange [1 2] **Andreas Karrenbauer** [1] **Bjoern Andres** [1 3 4]

A. Partial Optimality

A.1. Compulsory Edges

In this section, we discuss another partial optimality condition. To this end, we call a cut $B = (S_1, S_2)$ of G *minimal* if both S_1 and S_2 induce connected components.

Definition 5. For any graph $G = (V, E)$ and any $c \in \mathbb{R}^E$, an edge $f \in E$ is called *compulsory* w.r.t. (G, c) iff, for every minimal cut B of G with $f \in E_B$, it holds that

$$\sum_{e \in E_B} c_e > 0. \tag{26}$$

Lemma 6. *If $f \in E$ is compulsory, then $x^*_f = 0$ for any optimal solution x^* of* P_{MC}.

Proof. Suppose $f = uv \in E$ is compulsory and $x^*_f = 1$ in an optimal solution x^* of P_{MC}. The components of the partition corresponding to x^* can be divided into two groups $\mathcal{S} \,\dot\cup\, \mathcal{T} = \Pi$ such that

$$B = \Big(\bigcup_{S \in \mathcal{S}} S, \bigcup_{T \in \mathcal{T}} T \Big)$$

is a minimal cut and the nodes u and v belong to some component $S \in \mathcal{S}$, respectively $T \in \mathcal{T}$. The edge set of B can be written as

$$E_B = \bigcup_{S \in \mathcal{S}, T \in \mathcal{T}} E(S, T),$$

where $E(S, T)$ denotes the set of edges between components S and T. Therefore, since f is compulsory, we have

$$0 < \sum_{e \in E_B} c_e = \sum_{S \in \mathcal{S}} \sum_{T \in \mathcal{T}} \sum_{e \in E(S,T)} c_e.$$

This implies that there exist some $S' \in \mathcal{S}$ and $T' \in \mathcal{T}$ such that

$$\sum_{e \in E(S', T')} c_e > 0.$$

Hence, merging the clusters S and T strictly improves the objective, which contradicts the optimality of x^*. ☐

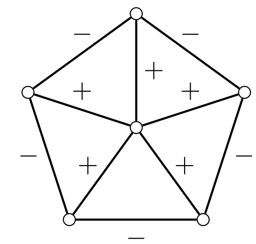

Figure 2. The figure shows a conflicted odd wheel of length 5. The edges in the outer cycle are repulsive and the spokes are attractive.

Unfortunately, finding compulsory edges is likely intractable, as computing minimum st-cuts w.r.t. signed capacity functions is NP-hard (by reduction from MAX CUT) on general graphs.

B. Dual Lower Bounds

B.1. Extension to Odd Wheels

In this section, we show that our dual heuristic algorithm can be extended so as to take into account so-called *odd wheel* inequalities.

A *wheel* is a graph $W = (V_W, E_C \cup E_w)$ with $V_W = \{v_1, \dots, v_{|E_C|}\} \cup \{w\}$ such that $(\{v_1, \dots, v_{|E_C|}\}, E_C)$ is a cycle and $E_w = \{\{v_i, w\} \mid 1 \leq i \leq k\}$ is the set of *spokes*. A wheel is called *odd* if $|E_C| = |E_w|$ is odd. With any wheel W we associate the wheel inequality

$$\sum_{e \in E_C} x_e - \sum_{e \in E_w} x_e \leq \left\lfloor \frac{|E_C|}{2} \right\rfloor. \tag{27}$$

Wheel inequalities are valid for the multicut polytope MC and can be separated in polynomial time. A wheel inequality

[1]Max Planck Institute for Informatics, Saarbrücken, Germany [2]Saarland University, Saarbrücken, Germany [3]Bosch Center for AI, Renningen, Germany [4]University of Tübingen, Germany. Correspondence to: Jan-Hendrik Lange <jlange@mpi-inf.mpg.de>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

is facet-defining for MC iff the associated wheel is odd (Chopra & Rao, 1993).

In analogy to conflicted cycles, we call a wheel $W = (V_W, E_C \cup E_w)$ *conflicted* w.r.t. (G, c) if $E_C \subset E^-$ and $E_w \subset E^+$. In other words, the cycle edges E_C are all repulsive and the spokes are all attractive, cf. Figure 2. Now, for any conflicted wheel $W = (V_W, E_C \cup E_w)$, the definition $\hat{x}_e := 1 - x_e$ for all $e \in E^-$ transforms the associated wheel inequality (27) to the covering inequality

$$\sum_{e \in E_C \cup E_w} \hat{x}_e \geq \left\lceil \frac{|E_C|}{2} \right\rceil . \tag{28}$$

Therefore, the cycle covering relaxation (22) can be tightened by including the conflicted (odd) wheel inequalities (28). This correponds to introducing additional wheel variables y_W with coefficients $\lceil |E_C|/2 \rceil$ into the dual (24). Note that in our heuristic approach we only increase dual variables, but never decrease any. Thus, since any conflicted odd wheel contains conflicted cycles, it only makes sense to increase wheel variables y_W before the main loop of ICP (or in between). We search for conflicted odd wheels by, for each center vertex $w \in V$, finding an odd cycle in the subgraph (U, E^-) with $U = \{u \neq w \mid uw \in E^+\}$. Similar to the cycle case, we iteratively pack conflicted odd wheels until no more conflicted odd wheels are left. Afterwards, we enter the main loop of ICP.

In Table 4 we empirically evaluate the effect of increasing conflicted odd wheel variables as described above, both in terms of runtime and obtained gaps. It can be seen that the runtime generally increases, in particular on the larger instances. However, the gaps are not improved with the exception of the complete graphs from *Modularity Clustering*.

C. Experiments

C.1. Partial Optimality

In Table 5 we evaluate the effect of our partial optimality reductions on the primal heuristics. We only apply the local conditions that can be checked in linear time and omit general uv-cuts. It can be observed that the differences in runtime are insignificant for the small graphs, but substantial for KLj on the instances *Epinions* and *Slashdot*.

C.2. Re-weighting

In Table 6 we provide a comparison between our heuristic re-weighting of instances and the reparameterization with MPC according to Swoboda & Andres (2017). It can be seen that the solutions found by GAEC+KLj have slightly lower optimality gaps when the instances were reparameterized by MPC. An exception are the complete graphs from *Modularity Clustering*.

References

Chopra, S. and Rao, M. The partition problem. *Mathematical Programming*, 59(1–3):87–115, 1993. doi: 10.1007/BF01581239.

Swoboda, P. and Andres, B. A message passing algorithm for the minimum cost multicut problem. In *CVPR*, 2017. doi: 10.1109/CVPR.2017.530.

Table 4. Reported below are lower bounds found by iterative cycle packing (ICP) and ICP with added conflicted odd wheels (OW+ICP), relative to the objective value of the best known feasible solution, as well as average total running times until convergence.

	Image Seg.		*Knott-3D-150*		*Knott-3D-300*		*Knott-3D-450*		*Mod. Clust.*		*Epinions*		*Slashdot*	
	t [s]	Gap	t [s]	Gap	t [s]	Gap	t [s]	Gap	t [s]	Gap	t [s]	Gap	t [s]	Gap
ICP	0.10	0.21%	0.04	0.14%	0.54	0.16%	3.31	0.18%	0.03	11.08%	65	1.04%	24	8.60%
OW+ICP	0.10	0.20%	0.06	0.17%	0.77	0.17%	7.70	0.19%	0.03	9.51%	340	3.38%	63	8.51%

Table 5. Compared below are the average runtimes of the primal heuristics (no prefix) to the average runtimes of the reduction algorithm with subsequent application of the primal heuristics (prefix RED).

	Image Seg.		*Knott-3D-150*		*Knott-3D-300*		*Knott-3D-450*		*Mod. Clust.*		*Epinions*		*Slashdot*	
	t [s]	Gap	t [s]	Gap	t [s]	Gap	t [s]	Gap	t [s]	Gap	t [s]	Gap	t [s]	Gap
GAEC	0.01	0.53%	0.01	0.28%	0.06	0.47%	0.21	0.25%	0.01	8.09%	2.19	1.75%	1.52	7.76%
RED+GAEC	0.01	0.54%	0.01	0.26%	0.08	0.54%	0.23	0.33%	0.01	8.30%	2.04	1.60%	1.56	7.54%
GAEC+KLj	0.03	0.34%	0.02	0.11%	0.42	0.15%	6.46	0.07%	0.01	1.28%	8324	0.18%	9217	6.69%
RED+GAEC+KLj	0.03	0.29%	0.02	0.10%	0.36	0.15%	4.19	0.07%	0.01	1.40%	1168	0.18%	2306	6.55%

Table 6. The table provides a comparison between our heuristic re-weighting and a reparameterization with dual solutions obtained from MPC. We report the gaps w.r.t. the original cost function obtained by relation to the objective value of the best known lower bound.

	Image Seg.	*Knott-3D-150*	*Knott-3D-300*	*Knott-3D-450*	*Mod. Clust.*	*Epinions*	*Slashdot*
GAEC+KLj	0.34%	0.11%	0.15%	0.07%	1.28%	0.18%	6.69%
ICP+GAEC+KLj	0.10%	0.00%	0.03%	0.05%	1.14%	0.12%	6.58%
MPC+GAEC+KLj	0.05%	0.01%	0.01%	0.01%	1.32%	0.09%	6.69%