Department of Industrial Engineering and Management

Technical Report

No. 2013-5

Redundant constraints in the standard formulation for the clique partitioning problem

Atsushi Miyauchi and Noriyoshi Sukegawa

May, 2013

Tokyo Institute of Technology

2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552, JAPAN
http://www.me.titech.ac.jp/index-e.html

Redundant constraints in the standard formulation for the clique partitioning problem

Atsushi Miyauchi[1,1,*], Noriyoshi Sukegawa[2]

Graduate School of Decision Science and Technology, Tokyo Institute of Technology, Ookayama 2-12-1, Meguro-ku, Tokyo 152-8552, Japan.

Abstract

Grötschel and Wakabayashi (1989) experimentally confirmed that their cutting plane algorithms for the standard formulation for the clique partitioning problem terminate when only a small fraction of constraints are added. Motivated by this result, we theoretically derive a certain class of redundant constraints in the formulation. More than half of the constraints belong to the class for some instances.

Keywords: Mathematical programming, Graph partitioning, Transitivity constraints

1. Introduction

The clique partitioning problem (CPP, for short) is one of the most fundamental graph partitioning problems. We consider a complete weighted undirected graph $G = (V, E, c)$ consisting of $n := |V|$ vertices, $n(n-1)/2$ edges and a weight $c : E \to \mathbb{R}$. Note that the range of c can include both positive and negative values. For convenience, $c(\{i, j\})$ is denoted by c_{ij} in what follows. A set A of edges is called *clique partitioning* if there exists a partition $\{V_1, V_2, \ldots, V_p\}$ of V such that

$$A = \bigcup_{l=1}^{p} \{\{i, j\} \in E \mid i, j \in V_l\}.$$

The goal of the CPP is to find a clique partitioning A that maximizes the total weight $\sum_{\{i,j\}\in A} c_{ij}$. The CPP is an NP-hard problem [1] with a wide variety of applications such as qualitative data analysis [2, 3], microarray analysis [4], group technology [5, 6], and community detection [7, 8].

*Corresponding author

Email addresses: `miyauchi.a.aa@m.titech.ac.jp` (Atsushi Miyauchi), `sukegawa.n.aa@m.titech.ac.jp` (Noriyoshi Sukegawa)

[1]Postal Address: Ookayama 2-12-1-W9-521, Meguro-ku, Tokyo 152-8552, Japan.

[2]Postal Address: Ookayama 2-12-1-W9-520, Meguro-ku, Tokyo 152-8552, Japan.

The standard formulation for the CPP is the one proposed by Grötschel and Wakabayashi [2]. Let $V := \{1, 2, \ldots, n\}$. By introducing variables x_{ij} $(i < j)$ equal to 1 if $\{i, j\} \in A$, 0 otherwise, the formulation can be written as

$$
\begin{aligned}
\text{(P): max.} \quad & \sum_{1 \le i < j \le n} c_{ij} x_{ij} \\
\text{s. t.} \quad & x_{ij} + x_{jk} - x_{ik} \le 1 \quad \forall 1 \le i < j < k \le n, \\
& x_{ij} - x_{jk} + x_{ik} \le 1 \quad \forall 1 \le i < j < k \le n, \\
& -x_{ij} + x_{jk} + x_{ik} \le 1 \quad \forall 1 \le i < j < k \le n, \\
& x_{ij} \in \{0, 1\} \qquad\qquad \forall 1 \le i < j \le n.
\end{aligned}
$$

The first three sets of constraints, called *transitivity constraints*, stipulate that for any triples of edges $\{i, j\}$, $\{j, k\}$ and $\{i, k\}$, if $\{i, j\} \in A$ and $\{j, k\} \in A$, then $\{i, k\} \in A$. This formulation has been frequently discussed and employed to construct heuristics and exact algorithms [2, 5, 7, 8]. However, (P) is prohibitive even for relatively small graphs, because the number of transitivity constraints grows rapidly with n.

Needless to say, (P) generally contains some redundant transitivity constraints, that is, we can obtain the same set of optimal solutions even if some constraints are removed from (P). Indeed, Grötschel and Wakabayashi [2] experimentally confirmed that, for many instances arising in qualitative data analysis, their cutting plane algorithms terminate when only a small fraction of transitivity constraints are added.

In this study, we theoretically derive a certain class of redundant transitivity constraints in (P). By using our result, the number of constraints treated in heuristics and exact algorithms can be reduced. We provide the number of the redundant constraints and its proportion for various instances. Although the proportion depends on a instance, more than half of the constraints belong to the class for some cases.

We note that the analysis in the present study is based on Dinh and Thai [9]. They have similar results for the modularity maximization problem, which is a special case of the CPP. Our result substantially extends its application range.

2. Main result

2.1. Redundant constraints

Focusing on the sign of c, we present the following class of constraints in (P).

$$
(*) \begin{cases}
x_{ij} + x_{jk} - x_{ik} \le 1 & \forall 1 \le i < j < k \le n,\ c_{ij} < 0 \wedge c_{jk} < 0, \\
x_{ij} - x_{jk} + x_{ik} \le 1 & \forall 1 \le i < j < k \le n,\ c_{ij} < 0 \wedge c_{ik} < 0, \\
-x_{ij} + x_{jk} + x_{ik} \le 1 & \forall 1 \le i < j < k \le n,\ c_{jk} < 0 \wedge c_{ik} < 0.
\end{cases}
$$

We consider the following formulation, which is derived by removing all constraints included in $(*)$ from (P).

$$
\begin{aligned}
\text{(RP): max.} \quad & \sum_{1 \le i < j \le n} c_{ij} x_{ij} \\
\text{s. t.} \quad & x_{ij} + x_{jk} - x_{ik} \le 1 \quad \forall 1 \le i < j < k \le n,\ c_{ij} \ge 0 \vee c_{jk} \ge 0, \\
& x_{ij} - x_{jk} + x_{ik} \le 1 \quad \forall 1 \le i < j < k \le n,\ c_{ij} \ge 0 \vee c_{ik} \ge 0, \\
& -x_{ij} + x_{jk} + x_{ik} \le 1 \quad \forall 1 \le i < j < k \le n,\ c_{jk} \ge 0 \vee c_{ik} \ge 0, \\
& x_{ij} \in \{0, 1\} \qquad \forall 1 \le i < j \le n.
\end{aligned}
$$

The following theorem states that all constraints included in $(*)$ are redundant.

Theorem 1. (P) *and* (RP) *have the same set of optimal solutions.*

Proof. It suffices to show that an arbitrary optimal solution

$$\boldsymbol{x}^* = (x^*_{ij})_{1 \le i < j \le n}$$

of (RP) is feasible for (P). Here, we set

$$E_+ = \{\{i, j\} \in E \mid c_{ij} \ge 0\}$$

and

$$E^* = \{\{i, j\} \in E \mid x^*_{ij} = 1\}.$$

Let $\{(V_1, E^*_1), (V_2, E^*_2), \ldots, (V_p, E^*_p)\}$ be the set of the connected components of (V, E^*). It is enough to show that (V_l, E^*_l) is complete for each $l = 1, 2, \ldots, p$. In what follows, let (V_l, E^*_l) be a fixed component. We initially present the following fact.

Lemma 1. $(V_l, E^*_l \cap E_+)$ *is connected.*

Proof. It suffices to show that for an arbitrary partition $\{S, T\}$ of V_l, there exists an edge in $E^*_l \cap E_+$ whose one endpoint in S and the other in T. From the definition of (V_l, E^*_l), there exists at least one edge in E^*_l which satisfies the above condition. Here, suppose that all these edges are not in E_+, that is, we assume that the weights of these edges are all negative. Let us focus on a feasible solution of (RP) obtained by changing the values of variables corresponding to those edges from 1 to 0 on $\boldsymbol{x}^*$. It is seen that the objective value of this solution is strictly greater than that of $\boldsymbol{x}^*$. This contradicts the optimality of $\boldsymbol{x}^*$. □

By this fact, for arbitrary distinct vertices $i, j \in V_l$, there exists a path $i = u_0, u_1, \ldots, u_q = j$ consisting of edges in $E^*_l \cap E_+$. The transitivity constraint

$$x_{u_0 u_1} + x_{u_1 u_2} - x_{u_0 u_2} \le 1$$

is included in (RP) since $\{u_0, u_1\} \in E_+$ (and $\{u_1, u_2\} \in E_+$). Thus, using $x^*_{u_0 u_1} = x^*_{u_1 u_2} = 1$, we have $x^*_{u_0 u_2} = 1$. Similarly, the transitivity constraint

$$x_{u_0 u_2} + x_{u_2 u_3} - x_{u_0 u_3} \le 1$$

Table 1: The number of the redundant constraints and its proportion.

Instance	n	#(P)	#(∗)	%
KKV [10]	24	6,072	863	14.21
KIN [11]	38	25,308	4,097	16.19
GRO [10]	43	37,023	5,381	14.53
BUR [12]	55	78,705	12,107	15.38
MIL [13]	60	102,660	20,613	20.08
LEE [12]	70	164,220	31,046	18.91
Wildcats [2]	30	12,180	2,137	17.55
Workers [2]	34	17,952	3,056	17.02
Cetacea [2]	36	21,420	11,822	55.19
Micro [2]	40	29,640	7,439	25.10
Soybean [3]	47	48,645	17,474	35.92
UNO [2]	54	74,412	28,656	38.51
Human [14]	132	1,123,980	672,882	59.87
UNO1b [2]	139	1,313,967	403,059	30.67
UNO1a [2]	158	1,934,868	773,245	39.96

is also included in (RP) since $\{u_2, u_3\} \in E_+$. (Note that $\{u_0, u_2\}$ is not necessarily in E_+.) Hence, substituting $x^*_{u_0u_2} = x^*_{u_2u_3} = 1$, we obtain $x^*_{u_0u_3} = 1$. Repeating this operation, we derive $x^*_{u_0u_q} = x^*_{ij} = 1$. This implies that (V_l, E^*_l) is complete. □

2.2. Examples

We confirm the number of constraints included in (∗) and its proportion for 15 instances. The results are shown in Table 1. The first column lists the name and source of each instance. The columns of #(P) and #(∗) give the number of transitivity constraints in (P) and (∗), respectively. The column of % provides the proportion of constraints in (∗). All instances arise in applications of the CPP; the first 6 instances arise in group technology, and the remainders arise in aggregation of equivalence relations.

The proportion of constraints in (∗) is not high for instances arising in group technology. On the other hand, relatively many constraints belong to (∗) for instances arising in aggregation of equivalence relations. In particular, 55.19% and 59.87% of transitivity constraints are included in (∗) for Cetacea and Human, respectively.

3. Conclusions

In this study, we introduced a class of redundant transitivity constraints in the standard formulation for the CPP. Our result can be seen as a theoretical evidence for the experimental result shown by Grötschel and Wakabayashi [2].

In closing, we note that the redundant constraints, which we revealed above, are also redundant in its linear programming relaxation. An exact claim and its proof are given in the Appendix.

Appendix. Redundant constraints in linear programming relaxation

The linear relaxation problems ($\overline{\mathrm{P}}$) and ($\overline{\mathrm{RP}}$) are derived by replacing the set of constraints $x_{ij} \in \{0,1\}$ by $x_{ij} \in [0,1]$ in (P) and (RP), respectively. As mentioned in Section 3, the following theorem states that all constraints included in ($*$) are also redundant in ($\overline{\mathrm{P}}$).

Theorem 2. ($\overline{\mathrm{P}}$) *and* ($\overline{\mathrm{RP}}$) *have the same set of optimal solutions.*

Proof. It suffices to show that an arbitrary optimal solution

$$\boldsymbol{x}^* = (x^*_{ij})_{1 \le i < j \le n}$$

of ($\overline{\mathrm{RP}}$) is feasible for ($\overline{\mathrm{P}}$) as well as the proof of Theorem 1. For convenience, we consider the residual graph of $\boldsymbol{x}^*$, denoted by $\boldsymbol{d}^*$. Namely,

$$\boldsymbol{d}^* = (d^*_{ij})_{1 \le i < j \le n} = \mathbf{1} - \boldsymbol{x}^*.$$

Then, it can be seen that the transitivity constraints for $\boldsymbol{x}^*$ in ($\overline{\mathrm{P}}$) correspond to the triangle inequalities for $\boldsymbol{d}^*$. For instance,

$$x^*_{ij} + x^*_{jk} - x^*_{ik} \le 1 \;\Leftrightarrow\; d^*_{ik} \le d^*_{ij} + d^*_{jk}.$$

for the first set of transitivity constraints. Hence, it is enough to show that $\boldsymbol{d}^*$ satisfies the triangle inequalities. Here, we set

$$\overline{E}^* = \{\{i,j\} \in E \mid d^*_{ij} < 1 \; (\Leftrightarrow x^*_{ij} > 0)\}.$$

Let $\{(V_1, \overline{E}^*_1), (V_2, \overline{E}^*_2), \ldots, (V_p, \overline{E}^*_p)\}$ be the set of the connected components of $(V, \overline{E}^*)$. It suffices to confirm that the triangle inequalities for all triples of nodes in each connected component are satisfied. The reason is that the other inequalities are always satisfied because there are at least two terms equal to 1 in each inequality. As well as Lemma 1, we have the following fact. A proof is omitted since its principle is entirely similar to that of Lemma 1.

Lemma 2. $(V_l, \overline{E}^*_l \cap E_+)$ *is connected.*

Therefore, for arbitrary distinct vertices $i, j \in V_l$, there exists at least one path consisting of edges in E_+. A shortest one of these paths and its distance are denoted by $i = u_0, u_1, \ldots, u_q = j$ and d'_{ij}, respectively. Now, repeatedly applying the triangle inequalities, corresponding to the transitivity constraints in ($\overline{\mathrm{RP}}$), to $\boldsymbol{d}^*$ as

$$\begin{aligned} d'_{ij} &= d^*_{u_0 u_1} + d^*_{u_1 u_2} + \cdots + d^*_{u_{q-1} u_q} \\ &\ge d^*_{u_0 u_2} + d^*_{u_2 u_3} + \cdots + d^*_{u_{q-1} u_q} \\ &\ge \cdots \ge d^*_{u_0 u_{q-1}} + d^*_{u_{q-1} u_q} \ge d^*_{u_0 u_q} = d^*_{ij}, \end{aligned}$$

we have $d'_{ij} \geq d^*_{ij}$. Here, let $d_{ij} = \min\{d'_{ij}, 1\}$ and construct $\boldsymbol{d} = (d_{ij})_{i<j}$ by gathering d_{ij} for all $i, j \in V_l$. Clearly, $d_{ij} \geq d^*_{ij}$. Furthermore, it can be seen that $\boldsymbol{d}$ satisfies the triangle inequalities since

$$\begin{aligned} d_{ij} + d_{jk} &= \min\{d'_{ij}, 1\} + \min\{d'_{jk}, 1\} \\ &\geq \min\{d'_{ij} + d'_{jk}, 1\} \\ &\geq \min\{d'_{ik}, 1\} = d_{ik}. \end{aligned}$$

The first inequality follows since $\boldsymbol{d}' = (d'_{ij})_{i<j}$ is nonnegative. The second inequality follows since $\boldsymbol{d}'$ satisfies the triangle inequalities. In point of fact, $d_{ij} = d^*_{ij}$ for all $i, j \in V_l$. If this is shown, then the proof is completed since $\boldsymbol{d}^*$ satisfies the triangle inequalities in the component.

Suppose that there exist some $i, j \in V_l$ such that $d_{ij} \neq d^*_{ij}$, namely $d_{ij} > d^*_{ij}$. Then, we have $\{i, j\} \notin E_+$. The reason is that, if $\{i, j\} \in E_+$, then $d_{ij} = d^*_{ij}$ since $d_{ij} \leq d^*_{ij}$. Let us focus on a feasible solution

$$\boldsymbol{x}' = \begin{cases} 1 - d_{ij} & \text{if } i, j \in V_l, \\ x^*_{ij} & \text{otherwise}, \end{cases}$$

of ($\overline{\text{RP}}$). By a simple calculation, it can be confirmed that the objective value of $\boldsymbol{x}'$ is strictly greater than that of $\boldsymbol{x}^*$. This contradicts the optimality of $\boldsymbol{x}^*$. □

References

[1] Y. Wakabayashi, Aggregation of binary relations: algorithmic and polyhedral investigations, Ph.D. Thesis, Universität Augsburg, 1986.

[2] M. Grötschel and Y. Wakabayashi, A cutting plane algorithm for a clustering problem, *Mathematical Programming* 45 (1989) 59–96.

[3] M. J. Brusco and H. F. Köhn, Clustering qualitative data based on binary equivalence relations: neighborhood search heuristics for the clique partitioning problem, *Psychometrika* 74 (2009) 685–703.

[4] H. Wang, B. Alidaee, F. Glover and G. Kochenberger, Clustering of microarray data via clique partitioning, *Journal of Combinatorial Optimization* 10 (2005) 77–92.

[5] M. Oosten, J. H. G. C. Rutten and F. C. R. Spieksma, The clique partitioning problem: Facets and patching facets, *Networks* 38 (2001) 209–226.

[6] H. Wang, B. Alidaee, F. Glover and G. Kochenberger, Solving group technology problems via clique partitioning, *International Journal of Flexible Manufacturing Systems* 18 (2006) 77–97.

[7] G. Agarwal and D. Kempe, Modularity-maximizing graph communities via mathematical programming, *The European Physical Journal B* 66 (2008) 409–418.

[8] D. Aloise, S. Cafieri, G. Caporossi, P. Hansen, L. Liberti and S. Perron, Column generation algorithms for exact modularity maximization in networks, *Physical Review E* 82 (2010) 046112.

[9] T. N. Dinh and M. T. Thai, Finding community structure with performance guarantees in complex networks, e-print `arXiv:1108.4034` (2011).

[10] R. K. Kumar, A. Kusiak and A. Vannelli, Grouping of parts and components in flexible manufacturing systems, *European Journal of Operational Research* 24 (1986) 387–397.

[11] J. R. King, Machine component group formation in group technology, *Omega* 8 (1980) 193–199.

[12] D. F. Rogers and S. S. Kulkarni, Optimal bivariate clustering and a genetic algorithm with an application in cellular manufacturing, *European Journal of Operational Research* 160 (2005) 423–444.

[13] J. Miltenburg and W. Zhang, A comparative evaluation of nine well-known algorithms for solving the cell formation in group technology, *Journal of Operations Management* 10 (1991) 44–72.

[14] C. L. Blake and C. J. Merz, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml/] (last accessed on Dec. 10, 2012), University of California, Irvine, School of Information and Computer Sciences (1998).