Sufficient Conditions for Redundant Transitivity Constraints in the Linear Ordering Problem

Noriyoshi Sukegawa

Joint work with Shinji Mizuno

(Tokyo Institute of Technology, Japan)
Optimization problem with many constraints

Suppose that we have some optimization problem in hand

- if the size is small, one can use an optimization software directly
- if otherwise, that is the size is too large,
  we need to introduce some techniques like cutting plane method
Optimization problem with many constraints

Suppose that we have some optimization problem in hand

▶ if the size is small, one can use an optimization software directly
▶ if otherwise, that is the size is too large, we need to introduce some techniques like cutting plane method

ex) Standard formulation for the traveling salesman problem

- # variables = $O(n^2)$ / # constraints = $O(2^n)$
Question

Do we need to consider all the constraints?
Do we need to consider all the constraints?

▶ to describe the feasible region, maybe yes.
▶ however, to solve the problem, of course no.
  □ Some of the constraints will be automatically satisfied as we optimize the objective function
  □ There are several “redundant” constraints
Question

Do we need to consider all the constraints?

- to describe the feasible region, maybe yes.
- however, to solve the problem, of course no.
  - Some of the constraints will be automatically satisfied as we optimize the objective function
  - There are several “redundant” constraints

Today, we want to discuss such a redundancy!!
Redundancy

We say that a set of constraints is **redundant** if the optimal solution set remains **unchanged** even if we delete it from the current formulation.
Redundancy

We say that a set of constraints is **redundant** if the optimal solution set remains **unchanged** even if we delete it from the current formulation.
Redundancy

We say that a set of constraints is **redundant** if the optimal solution set remains **unchanged** even if we delete it from the current formulation.
Abstract

We consider the linear ordering problem

- NP-hard in general
- classical combinatorial optimization problem
We consider the linear ordering problem

- NP-hard in general
- classical combinatorial optimization problem

We focus on the standard formulation

- # variables = $O(n^2)$ / # constraints = $O(n^3)$
- $n = 200$, about 3 million constrains!!
Abstract

We consider the linear ordering problem

- NP-hard in general
- classical combinatorial optimization problem

We focus on the standard formulation

- \# variables = \(O(n^2)\) / \# constraints = \(O(n^3)\)
- \(n = 200\), about 3 million constraints!!

We give sufficient conditions for the constraints to be redundant
Content

- Motivation
- Linear Ordering Problem
- The standard formulation
- Main result
- Concluding remarks
Motivation

Dinh & Thai ['12]

▶ consider the modularity maximization problem
▶ deal with the standard formulation
  □ # variables = $O(n^2)$ / # constraints = $O(n^3)$
  □ $n = 200$, about 3 million constraints!!
▶ and give a sufficient condition for the redundant constraints
Motivation

Dinh & Thai ['12]

- consider the modularity maximization problem
- deal with the standard formulation
  - # variables = $O(n^2)$ / # constraints = $O(n^3)$
  - $n = 200$, about 3 million constraints!!
- and give a sufficient condition for the redundant constraints

Miyauchi & S ['14] generalize and slightly improve their result
Numerical experiments

Removing the redundant constraints lessens the CPU time by Gurobi Optimizer (compared to when we use all the constraints)

- for well structured instances
  - about 90% of the constraints are revealed to be redundant
  - in addition, the CPU time lessens about 80%!!
Numerical experiments

Removing the redundant constraints lessens the CPU time by **Gurobi Optimizer** (compared to when we use all the constraints)

- for well structured instances
  - about 90% of the constraints are revealed to be redundant
  - in addition, the CPU time lessens about 80%!!

Of course, one can implement a specialized cutting plane algorithms, but using these results would be much **easier**
Question

Can we reveal such a redundancy in other problems?

- to begin with, we consider the linear ordering problem
- to obtain the result, we utilize the analysis in Miyauchi & S [’14]
Question

Can we reveal such a redundancy in other problems?

- to begin with, we consider the linear ordering problem
- to obtain the result, we utilize the analysis in Miyauchi & S ['14]

In Miyauchi & S ['14], they deal with the generalized problem (clique partitioning problem) whose solutions are transitive binary relation

- In the linear ordering problem, solutions are transitive binary relation
- We would apply similar analysis
Content

- Motivation
- Linear Ordering Problem
- The standard formulation
- Main result
- Concluding remarks
Let $G = (V, A)$ be a complete directed graph

- $T \subseteq A$ is **tournament** if either $(i, j) \in T$ or $(j, i) \in T$
- $T$ is **transitive** if $(i, j) \in T$ and $(j, k) \in T$ implies $(j, k) \in T$
- linear ordering $\iff$ transitive tournament
Linear Ordering Problem

In the linear ordering problem,

- we are given an arc weight $w_{ij} \in \mathbb{R}$ for each $(i,j) \in A$
- the goal is to find a maximum weight linear ordering
- Pairwise comparison $\Rightarrow$ Linear ordering which maximizes the “fitness”

Applications:

- **voting system** ($w_{ij}$: # voters who prefer candidate $i$ to candidate $j$)
- **Ranking of sports team** ($w_{ij}$: score team $i$ get from team $j$)
Linear Ordering Problem

In the linear ordering problem,

- we are given an arc weight $w_{ij} \in \mathbb{R}$ for each $(i,j) \in A$
- the goal is to find a maximum weight linear ordering
- Pairwise comparison $\Rightarrow$ Linear ordering which maximizes the “fitness”
Content

▶ Motivation
▶ Linear Ordering Problem
▶ The standard formulation
▶ Main result
▶ Concluding remarks
Standard formulation

Introduce $x_{ij} \in \{0, 1\}$ takes 1 if $(i, j) \in T$

- then the problem can be formulated as a simple 0-1 ILP.

$$(P): \quad \text{maximize} \quad \sum_{(i,j) \in A} w_{ij} x_{ij}$$

subject to

\begin{align*}
    x_{ij} + x_{ji} &= 1 \\
    x_{ij} + x_{jk} + x_{ki} &\leq 2 \\
    x_{ij} &\in \{0, 1\}
\end{align*}$$
Standard formulation

Introduce $x_{ij} \in \{0, 1\}$ takes 1 if $(i, j) \in T$

then the problem can be formulated as a simple 0-1 ILP.

(P):

maximize $\sum_{(i,j) \in A} w_{ij} x_{ij}$

subject to

$x_{ij} + x_{ji} = 1$

$x_{ij} + x_{jk} + x_{ki} \leq 2$

$x_{ij} \in \{0, 1\}$

inequality constraints are called transitivity constraints
Transitivity constraint

# transitivity constraints = $2 \binom{n}{3} = O(n^3)$

- On the other hand, # variables = $2 \binom{n}{2} = O(n^2)$

- Hence, clearly, # constraints is quite large for # variables
Transitivity constraint

# transitivity constraints = $2 \binom{n}{3} = O(n^3)$

- On the other hand, # variables = $2 \binom{n}{2} = O(n^2)$
- Hence, clearly, # constraints is quite large for # variables

However, some of the constraints might be redundant.

- Intuitively, transitivity constraints corresponding to rock – scissors – papers relation are likely to be violated
- Conversely, reverse of rock – scissors – papers relation are NOT
Content

- Motivation
- Linear Ordering Problem
- The standard formulation
- Main result
- Concluding remarks
Main result

Depending on a given arc weight \( w \), Let us divide \( A \) as

\[ A^+ = \{(i,j) \in A: w_{ij} \geq w_{ji}\} \]

(“win” or “tie”)

Roughly speaking, we want to take many arc from \( A^+ \) as much as possible
Main result

Depending on a given arc weight $w$, Let us divide $A$ as

- $A^+ = \{(i,j) \in A: w_{ij} \geq w_{ji}\}$  \hspace{1cm} ("win" or "tie")
- $l_{ijk} = \# \{(i,j), (j,k), (k,i)\} \cap A^+ \in \{0, 1, 2, 3\}$
- $l_{ijk}$ captures "how the constraint is likely to be violated"
- $l_{ijk} = 3$ corresponds rock – scissors – paper relation

\[ l_{ijk} = 2 \]

\[ l_{ijk} = 1 \]

Main result

Depending on a given arc weight $w$, Let us divide $A$ as

- $A^+ = \{(i,j) \in A: w_{ij} \geq w_{ji}\}$ ("win" or "tie")
- $l_{ijk} = \# \{(i,j), (j,k), (k,i)\} \cap A^+ \in \{0, 1, 2, 3\}$
- $l_{ijk}$ captures "how the constraint is likely to be violated"
- $l_{ijk} = 3$ corresponds rock – scissors – paper relation

Theorem 1

Transitivity constraints $x_{ij} + x_{jk} + x_{ki} \leq 2$ with $l_{ijk} = 0$ are redundant
Outline of the proof

Let \((R)\) be a problem obtained from \((P)\) by deleting all the transitivity constraints with \(l_{ijk} = 0\).
Outline of the proof

Let \((R)\) be a problem obtained from \((P)\) by deleting all the transitivity constraints with \(l_{ijk} = 0\).

Now we want to show that any optimal solution \(x^*\) to \((R)\) is also feasible at original problem \((P)\).
Outline of the proof

Let \((R)\) be a problem obtained from \((P)\) by deleting all the transitivity constraints with \(l_{ijk} = 0\).

Now we want to show that any optimal solution \(x^*\) to \((R)\) is also feasible at original problem \((P)\).

\[ T^* = \{(i, j) \in A: x^*_{ij} = 1\} \]

\[ \text{it is enough to show that } T^* \text{ contain no directed cycle} \]
Outline of the proof

Let \((R)\) be a problem obtained from \((P)\) by deleting all the transitivity constraints with \(l_{ijk} = 0\).

Now we want to show that any optimal solution \(x^*\) to \((R)\) is also feasible at original problem \((P)\).

- \(T^* = \{(i,j) \in A: x^*_{ij} = 1\}\)
- it is enough to show that \(T^*\) contain no directed cycle

Lemma 1

If there exists a path from \(s\) to \(t\) on \(T^*\), there also exists one on \(T^* \cap A^+\)
Proof

Suppose to the contrary that $T^*$ contain directed cycle

- by Lemma1, there exists a directed cycle on $T^* \cap A^+$

If there exists a path from $s$ to $t$ on $T^*$, there also exists one on $T^* \cap A^+$
Proof

Suppose to the contrary that $T^*$ contain directed cycle

- by Lemma 1, there exists a directed cycle on $T^* \cap A^+$

To meet $x_{s1} + x_{12} + x_{2s} \leq 2$ ($l_{s12} \geq 2$),

since $(s, 1), (1, 2) \in T^*$, we have $(2, s) \notin T^*$

In other words, $(s, 2) \in T^*$

Lemma 1

If there exists a path from $s$ to $t$ on $T^*$, there also exists one on $T^* \cap A^+$
Proof

Suppose to the contrary that $T^*$ contain directed cycle

- by Lemma1, there exists a directed cycle on $T^* \cap A^+$

To meet $x_{s2} + x_{23} + x_{3s} \leq 2$ ($l_{s23} \geq 1$),

since $(s, 2), (2, 3) \in T^*$, we have $(3, s) \notin T^*$

In other words, $(s, 3) \in T^*$

Lemma1

If there exists a path from $s$ to $t$ on $T^*$, there also exists one on $T^* \cap A^+$
Proof

Suppose to the contrary that $T^*$ contain directed cycle

- by Lemma 1, there exists a directed cycle on $T^* \cap A^+$
- using constraints with $l_{ijk} \geq 1$, we have $(s, t) \in T^*$
- Contradicts to $(s, t) \notin T^*$

If there exists a path from $s$ to $t$ on $T^*$, there also exists one on $T^* \cap A^+$
Limitation

# transitivity constraints with \( l_{ijk} = 0 \) is not so large

- Are there more redundant constraints?
- For instance, \( l_{ijk} = 1 \) is also redundant?

\[ l_{ijk} \leq 0 \subseteq l_{ijk} \leq 1 \subseteq l_{ijk} \leq 2 \]
Limitation

# transitivity constraints with \( l_{ijk} = 0 \) is not so large

- Are there more redundant constraints?
- For instance, \( l_{ijk} = 1 \) is also redundant?
- Unfortunately, this is not true since we have a counterexample

\[
\begin{align*}
    l_{ijk} &\leq 0 \subseteq l_{ijk} \leq 1 \subseteq l_{ijk} \leq 2 \\
    l_{123} = l_{234} = l_{345} = l_{451} = l_{512} = 3
\end{align*}
\]

"pathological" directed cycle

\[
\begin{array}{cccccc}
    & 1 & 2 & 3 & 4 & 5 \\
\hline
1 & - & M & 0 & 1 & 1 \\
2 & 1 & - & M & 0 & 1 \\
3 & 1 & 1 & - & M & 0 \\
4 & 0 & 1 & 1 & - & M \\
5 & M & 0 & 1 & 1 & - \\
\end{array}
\]

\( M \): sufficiently large

(rock – scissors – paper relation)
Further results

Theorem 2

If there is no "pathological" directed cycle in \((V, A^+)\),
the transitivity constraints \(x_{ij} + x_{jk} + x_{ki} \leq 2\) with \(l_{ijk} \leq 1\) are redundant.

Let \(d\) be the length of the longest directed cycle in \((V, A^+)\).

Theorem 3

When \(d = 4\), the transitivity constraints \(x_{ij} + x_{jk} + x_{ki} \leq 2\) with \(l_{ijk} \leq 1\)
are redundant. When \(d = 3\), the ones with \(l_{ijk} \leq 2\) are redundant.
Content

- Motivation
- Linear Ordering Problem
- The standard formulation
- Main result
- Concluding remarks
Brief numerical experiments

We solved the instances from LOP library LOLIB
<http://www.iwr.uni-heidelberg.de/groups/comopt/software/LOLIB/>

- Using Theorem 1 reduces the CPU time but not so much
- Although our sufficient conditions are not satisfied, the transitivity constraints with $l_{ijk} \leq 1$ are always redundant
- The transitivity constraints with $l_{ijk} \leq 1$ account for exactly half of all the transitivity constraints
- The CPU time is also about half (compared to the case when we feed all the constraints to an optimization software)
Conclusion

We discussed the redundancy of the transitivity constraints of the standard formulation for the linear ordering problem. Our analysis is based on that of Miyauchi & S[‘14].

We propose to classify the transitivity constraints into four classes in a hierarchical way and give a sufficient condition for each class to be redundant.

However, currently, the contribution on the CPU time is not so big.
Future work

We showed that the transitivity constraints with $l_{ijk} \leq 1$ is not necessarily redundant by giving an example. There, we use big coefficients.

▶ What if the coefficients are “balanced”?

▶ ex) if $w_{ij} \in \{0, 1\}$, we obtain the Slater’s problem which is NP-hard

Address another well-known problem, say TSP

▶ cf) If we do not take the objective function into account and do not introduce auxiliary variables, the standard formulation for TSP must have $O(2^n)$ inequalities (Kaibel and Weltge[’14])
merci beaucoup