数理解析研究所講究録
第2069巻 2018年 62-73

An Application of Polyhedral Relaxations to Optimal Contribution Selection of Tree Breeding Problem

Sena Safarina[1] and Makoto Yamashita[1]

[1] *Department of Mathematical and Computing Science, Tokyo Institute of Technology, 2-12-1-W8-29 Ookayama, Meguro-ku, Tokyo 152-8552, Japan.*

1 Introduction

An optimal contribution selection (OCS) is a mathematical optimization problem that aims to maximize the total benefit under a constraint for genetic diversity. Based on the contribution of candidates, OCS problems can be classified into unequal and equal deployment problems. While an unequal deployment problem (UDP) does not require the same contribution for the candidates, an equal deployment (EDP) demands the chosen candidate contribute the same amount.

A mathematical optimization formulation for UDP is proposed by Meuwissen [7] that also can be found on [14]. However, this research is concerned on EDP of form:

$$\begin{array}{lll} \text{maximize} & : & \boldsymbol{g}^T\boldsymbol{x} \\ \text{subject to} & : & \boldsymbol{e}^T\boldsymbol{x} = 1, \\ & & \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \leq 2\theta, \\ & & x_i \in \left\{0, \frac{1}{N}\right\} \text{ for } i = 1, \ldots, m. \end{array} \tag{1}$$

The objective is to maximize the total benefit $\boldsymbol{g}^T\boldsymbol{x}$ where $\boldsymbol{g} = \{g_1, g_2, \ldots, g_m\}$ is an estimate breeding value (EBV) representing the quality of each tree candidate $\boldsymbol{x}$. The constraint $\boldsymbol{e}^T\boldsymbol{x} = 1$ shows that the total contribution of all candidates is unity due to the vector all of ones $\boldsymbol{e} \in \mathbb{R}^m$. Our important constraint $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \leq 2\theta$ requires the genetic diversity be under an appropriate level $\theta \in \mathbb{R}_{++}$ in which $\mathbb{R}_{++}$ is the set of strictly positive real numbers $]0, \infty[$The genetic diversity constraint is proposed by [2] while the construction of numerical relationship matrix $\boldsymbol{A} \in \mathbb{R}^{m \times m}$ is proposed by [13]. Moreover, [11] observed that the matrix $\boldsymbol{A}$ is always semi-definite positive so that the problem can be solved by semi-definite programming (SDP) approaches. The last constraint defines 0 for candidate x_i with no contribution, and $\frac{1}{N}$ for that with the contribution. Here, $N \in \mathbb{R}_{++}$ is the parameter to indicate the number of candidate we will choose candidates, and $m \in \mathbb{R}_{++}$ is the number of whole candidates.

The OCS problem has been solved trough a software `GENCONT` [8] to control inbreeding in the selection. `GENCONT` is based on Lagrangian multiplier method which fixes the solution that exceeds lower or upper bounds $\left(0 \leq x_i \leq \frac{1}{N}\right)$ at only its lower and upper bound. Thus, even though `GENCONT` outputs the solution in only a few seconds, it often outputs only suboptimal solution for OCS problem rather an optimal solution. To resolve such a problem arising from `GENCONT`, different software, `OPSEL` [10], was proposed by Mullin. `OPSEL` is an implementation of branch and bound with an outer approximation method. Using this implementation, they successfully computed optimal solutions. However, `OPSEL` generates huge number of constraints in the framework of branch and bound, therefore, computing the solution by `OPSEL` takes long computation time. Hence, it is necessary to find a different approach to solve the problem efficiently.

The quadratic constraint in formulation (1) can be described with a second-order cone. Through the Cholesky factorization $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{U}^T$, the following condition can be derived:

$$\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \leq 2\theta N^2 \Leftrightarrow \sum_i \left(\boldsymbol{U}_i^T x_i\right)^2 \leq 2\theta N^2 \Leftrightarrow \left(\sqrt{2\theta}N, \boldsymbol{U}^T\right) \in \mathcal{K}^m,$$

where $\mathcal{K}^m$ is the $(m+1)$-dimensional second-order cone defined by

$$\mathcal{K}^m := \left\{(v_0, \boldsymbol{v}) \in \mathbb{R}_+ \times \mathbb{R}^r : \sum_{k=1}^m v_k^2 \leq v_0^2\right\}.$$

Introducing a new variable $\boldsymbol{y} = N\boldsymbol{x}$, we get MI-SOCP formulation of our OCS problem (1):

$$\begin{array}{lll} \text{maximize} & : & \frac{\boldsymbol{g}^T\boldsymbol{y}}{N} \\ \text{subject to} & : & \boldsymbol{e}^T\boldsymbol{y} = N, \\ & & \left(\sqrt{2\theta}N, \boldsymbol{U}^T\boldsymbol{y}\right) \in \mathcal{K}^m, \\ & & y_i \in \{0,1\} \text{ for } i = 1,\ldots,m. \end{array} \tag{2}$$

However, the non-linearity on MI-SOCP formulation leads to heavy computation time. Therefore, we discuss approaches based on polyhedral programming relaxation, an implementation of lifted polyhedral programming (LPP) relaxation and a cone decomposition relaxation, to reduce the long computation time.

We conducted numerical experiment for the existing implementations: `GENCONT` and `OPSEL`. Since we use `CPLEX` [5] can handle integer constraint on (2), we also implemented (1) on `CPLEX` to compare the effectiveness with our proposed methods.

The remaining of our paper is organized as follow. In Section 2 and 3, we explain our proposed approaches based on LPP relaxation and cone decomposition, respectively. We present the numerical result for all methods in Section 4. Finally, in Section 5, we conclude our research and discuss for future studies.

2 Lifted Polyhedral Programming Relaxation

Lifted polyhedral programming relaxation [1][4][12] is an approach to solve SOCP problems by employing polyhedral relaxation as illustrated in Figure 1. The second-order cone $\mathcal{K}^2$ on Figure 1(a) is approximated by a construction of polyhedron, to generate linear constraints since the nonlinearity of $\mathcal{K}^2$ makes MI-SOCP problems hard. More precisely, we generate many planes as in Figure 1(b). Thus, we obtain a mixed-integer *linear* programming problem as the resultant problem which can decrease the heavy computation time.

The paper [1] proposed to replace $\mathcal{K}^m$ with a polyhedron $\mathcal{K}^m_\epsilon$ using a tightness $\epsilon > 0$ so that satisfies:

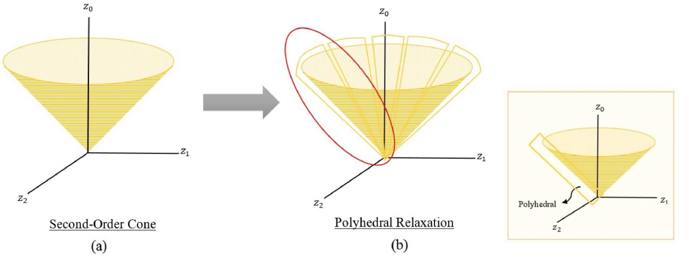

Figure 1: Polyhedral Relaxation

$$\mathcal{K}^m \subsetneq \mathcal{K}^m_\epsilon \subsetneq \{(v_0, \boldsymbol{v}) \in \mathbb{R}_+ \times \mathbb{R}^m : ||\boldsymbol{v}||_2 \leq (1+\epsilon)v_0\}.$$

In addition, the polyhedral relaxation $\mathcal{K}^m_\epsilon$ is given as below [12]:

$$\begin{aligned}
\mathcal{K}^m_\epsilon := \{(v_0, \boldsymbol{v}) \in \mathbb{R}_+ \times \mathbb{R}^m : \exists (\delta^j)_{j=0}^J \in \mathbb{R}^{(T(m))} \text{ s.t.} \\
& v_0 = \delta_1^J, \\
& \delta_i^0 = v_i \text{ for } i \in \{1, \cdots, m\}, \\
& \left(\delta_{2i-1}^j, \delta_{2i}^j, \delta_i^{j+1}\right) \in \mathcal{W}_{s_j(\epsilon)} \text{ for } i \in \left\{1, \cdots, \left\lfloor \frac{t_j}{2} \right\rfloor \right\}, \ j \in \{0, \cdots, J-1\}, \\
& \delta_{t_j}^j = \delta_{\lceil t_j/2 \rceil}^{j+1} \text{ for } j \in \{0, \cdots, J-1\} \text{ s.t. } t_j \text{ is odd}\}
\end{aligned}$$

with $J = \lceil \log_2(m) \rceil$, and $\{t_j\}_{j=0}^J$ is defined recursively as follow:

$$\begin{cases} t_0 = m, \\ t_{j+1} = \lceil \frac{t_j}{2} \rceil \quad \text{for } j \in \{0, ..., J-1\}. \end{cases}$$

Using this definition, we also define $T(m) = \sum_{j=0}^J t_j$. In the definition of polyhedral relaxation, $\mathcal{W}_s$ expresses a polyhedron to approximate the second-order cone $\mathcal{K}^m$ defined by the following constraints:

$$\begin{aligned}
\mathcal{W}_s := \big\{(v_0, v_1, v_2) \in \mathbb{R}_+ \times \mathbb{R}^2 : \exists(\alpha, \beta) \in \mathbb{R}^{2s} \text{ s.t}\\
v_0 &= \alpha_s \cos\left(\frac{\pi}{2^s}\right) + \beta_s \sin\left(\frac{\pi}{2^s}\right),\\
\alpha_1 &= v_1 \cos(\pi) + v_2 \sin(\pi),\\
\beta_1 &\geq |v_2 \cos(\pi) - v_1 \sin(\pi)|,\\
\alpha_{i+1} &= \alpha_i \cos\left(\frac{\pi}{2^i}\right) + \beta_i \sin\left(\frac{\pi}{2^i}\right),\\
\beta_{i+1} &\geq \left|\beta_i \cos\left(\frac{\pi}{2^i}\right) - \alpha_i \sin\left(\frac{\pi}{2^i}\right)\right|,\\
&\text{for } i \in \{1, ..., s-1\}\big\}
\end{aligned}$$

where s is the number of the attached planes. When we use $\mathcal{W}_{s_j(\epsilon)}$ $s_j(\epsilon)$ is defined with

$$s_j(\epsilon) = \left\lceil \frac{j+1}{2} \right\rceil - \left\lceil \log_4\left(\frac{16}{9}\pi^{-2}\log(1+\epsilon)\right)\right\rceil \text{ for } j \in \{0, \ldots, J-1\}.$$

Using the relation we can have a good approximation with small $\epsilon > 0$ as illustrated in Figure 2.

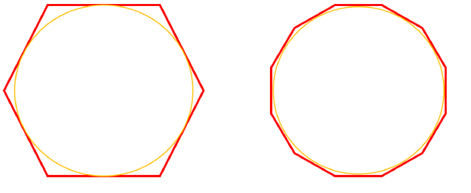

Figure 2: Effect of choosing different ϵ

We implemented this approach for different parameter values 2θ and small $\epsilon > 0$ using **Matlab R2016a**. We also set the duality gap 10% as the stopping criterion in **CPLEX**, so the accuracy of the obtained objective value is 10%.

Table 1 shows the numerical results of LPP relaxation for $2\theta = \{0.03, 0.05, 0.08\}$ with different $(1+\epsilon)2\theta$. The first column in the table is the problem size m, the second is the parameter for the diversity constraint 2θ, the third indicates ϵ to generate the LPP constraints. The ϵ is expressed by the change of group coancestry threshold from 2θ to $(1+\epsilon)2\theta$. For example, when we relax $2\theta = 0.03$ to $(1+\epsilon)2\theta = 0.0301$, it means that we set $\epsilon = 0.33 \times 10^2$. The four, fifth, and six columns show the computation time to build the mathematical model, the time to solve the mathematical model and the total computation time, respectively. The last two columns show the obtained results: the group coancestry $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}$ and the objective value $\boldsymbol{g}^T\boldsymbol{x}$.

The result on Table 1 was generated on different environment due to out of memory (OOM) while solving the problem with $2\theta = 0.03$. More precisely, we used a Debian Linux server on Opteron 4386 (3.10 GHz) and 128 GB memory space for the problem with $2\theta = 0.03$. The remaining problem was solved by a 64-bit Windows 10 PC on Xeon CPU E2-1231 (3.40 GHz) with 8 GB memory space.

Table 1: LPP results for $2\theta = \{0.03, 0.05, 0.08\}$

Pedigree m	2θ	$(1+\epsilon)2\theta$	Time (s) Builder	Time (s) Solver	Time (s) Total	Group Coancestry	Objective Value
200	0.03	0.0306	0.04	179.14	179.18	0.0306	22.9232
200		0.0301	0.47	575.19	575.66	0.0300	22.3038
200	0.05	0.0510	0.04	0.23	0.27	0.0506	27.8762
200		0.0505	0.04	0.30	0.35	0.0498	27.8536
200	0.08	0.0810	0.04	0.07	0.11	0.0574	28.0676
200		0.0807	0.04	0.09	0.14	0.0574	28.0676
1050	0.03	0.0306					OOM
1050		0.0301					OOM
1050	0.05	0.0510	0.28	5.69	5.97	0.0506	21.7713
1050		0.0505	0.31	17.67	17.99	0.0501	21.8048
1050	0.08	0.0810	0.27	2.51	2.79	0.0806	27.5894
1050		0.0807	0.32	4.62	4.95	0.0797	27.4881
2045	0.03	0.0306	0.92	7.03	7.96	0.0304	360.5734
2045		0.0301	1.06	212.84	213.90	0.0300	380.4484
2045	0.05	0.0510	0.94	3.85	4.80	0.0508	420.8250
2045		0.0505	1.06	6.79	7.86	0.0498	420.0940
2045	0.08	0.0810	0.96	1.89	2.86	0.0812	444.7350
2045		0.0807	1.05	1.96	3.01	0.0788	443.0992
5050	0.03	0.0306					OOM
5050		0.0301					OOM
5050	0.05	0.0510	9.44	620.12	629.56	0.0507	29.0492
5050		0.0505	13.55	836.25	849.81	0.0501	28.8007
5050	0.08	0.0810	7.03	41.94	48.97	0.0807	37.7933
5050		0.0807	10.66	36.45	47.12	0.0784	37.3102

From Table 1, we observed that LPP cannot obtain the solution for some problems with very tight 2θ due to OOM. The utilization of the tight 2θ makes its polyhedral relaxation $(1+\epsilon)2\theta$ generate large number of LPP constraints. Besides, LPP does not obtain optimal solution for larger ϵ. As example, the problem with $(1+\epsilon) = 0.0306$ is not optimal since the solution of group coancestry $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} > 2\theta$ which violates our quadratic constraint. Thus, we need to determine another tighter ϵ that will consume times. Therefore, we need another implementation to reduce the large number of the constraints.

We combine the implementation of LPP approach with an active constraint selection method to select important constraint for our OCS problem.

Definition 2.1 (Active Constraint) *Let $a_i^T x \le b_i$ $(i = 1, \dots, p)$ be inequality constraints in an optimal optimization problem with $a_i \in \mathbb{R}^q$ and $b_i \in \mathbb{R}$ $(i = 1, \dots, p)$, and let x^* be an optimal solution of the optimization problem. We set a threshold $\epsilon > 0$.*

The inequality constraint $a_i^T x \le b_i$ is said to be active at x^* if $|a_i^T x^* - b_i| < \epsilon$. Otherwise, the constraint $a_i^T x \le b_i$ is called inactive.

Algorithm 2.2 (Active constraint selection method) *If an inequality constraint $a_i^T x \le b_i$ is active at some optimal solution x^* obtained in a preliminary experiment, we replace $a_i^T x \le b_i$ with the equality constraint $a_i^T x = b_i$.*

Using such method, we conducted preliminary experiments and found that the constraint

$$\beta_1 \ge -v_2 \cos(\pi) + v_1 \sin(\pi)$$

in the polyhedron $\mathcal{W}_s$ always active at the obtained optimal solution. Therefore, we replace the constraint $\beta_1 \ge |v_2 \cos(\pi) - v_1 \sin(\pi)|$ by the equality $\beta_1 = -v_2 \cos(\pi) + v_1 \sin(\pi)$. This replacement can reduce the number of inequalities from which we expect the reduction of computation time.

Table 2 shows the computation time of the framework of LPP+active constraint. However, the implementation of LPP relaxation combining with active selection method (LPP-AS) requires longer computation time than LPP implementation itself. In addition, the number of problems that generated OOM is increased compared with the previous one. For example, we do not include the result for the problem with $m = \{10100, 15222\}$ since it failed to obtain the solution due to OOM. We consider that using LPP and LPP-AS is hard depending on the chosen ϵ (see Table 1). We should determine good ϵ to obtain the optimal solution in a practical time. Therefore, we propose another implementation for solving OSC problem in the next section.

Table 2: LPP-AS results with $2\theta = \{0.03, 0.05, 0.08\}$

Pedigree m	2θ	$(1+\epsilon)2\theta$	Time (Builder)	Time (Solver)	Time (Total)	Group Coancestry	Objective Value
200	0.03	0.0301					OOM
200	0.05	0.0505	0.05	0.20	0.25	0.0500	27.8516
200	0.08	0.0807	0.02	0.05	0.08	0.0574	28.0676
1050	0.03	0.0301					OOM
1050	0.05	0.0501	0.36	28.50	28.86	0.0499	21.5067
1050	0.08	0.0807	0.34	2.32	2.66	0.0797	27.4657
2045	0.03	0.0301	1.08	442.56	443.64	0.0300	373.9100
2045	0.05	0.0505	1.00	7.52	8.53	0.0496	418.7352
2045	0.08	0.0807	1.14	4.62	5.76	0.0804	444.2776
5050	0.03	0.0301					OOM
5050	0.05	0.0505	15.47	3373.64	3389.12	0.0497	28.5624
5050	0.08	0.0807	11.0807	294.79	305.87	0.0803	37.3127

3 Cone Decomposition Method

The concept of cone decomposition method is similar to LPP implementation. We employ a polyhedral relaxation to solve the problem. However, the cone decomposition derives different formulation on lifted polyhedral relaxation to approximate the solution. An m-dimensional second-order cone can be decomposed by the following theorem.

Theorem 3.1 *[6] Let*

$$\hat{\boldsymbol{H}}^d := \left\{ (v_0, \boldsymbol{v}, \boldsymbol{w}) \in \mathbb{R}^{(2m+1)} : v_j^2 \leq w_j v_0, \forall j \in \{1, \dots, d\}, \sum_{j=1}^{d} w_j \leq v_0 \right\},$$

then $\mathcal{K}^d = Proj_{(v_0,\boldsymbol{v})}(\hat{\boldsymbol{H}}^d)$ and hence $\hat{\boldsymbol{H}}^d$ is a lifted reformulation of $\mathcal{K}^d$ with d rotated two-dimensional conic quadratic constraints, one linear constraint, and d auxiliary variables. $Proj_{(v_0,\boldsymbol{v})}$ is the orthogonal projection onto the space $(v_0, \boldsymbol{v})$ variables.

The utilization of the theorem makes another reformulation on our OCS (3) as follow:

$$\begin{array}{ll} \text{maximize} & : \frac{\boldsymbol{g}^T \boldsymbol{y}}{N} \\ \text{subject to} & : \boldsymbol{e}^T \boldsymbol{y} = N, \\ & z_i^2 \leq w_i z_0 \text{ for } i = 1, \dots, m, \\ & \sum_{i=1}^{m} w_i \leq z_0, \\ & y_i \in \{0, 1\} \text{ for } i = 1, \dots, m \end{array} \tag{3}$$

where $z_i = U_i^T y_i$ for $i = 1, \dots, m$ and $z_0 = \sqrt{2\theta N^2}$.

Regarding the quadratic constraint in new formulation (2), we consider implementing another method to convert it into linear constraints. Our approach is based on a Lagrange multiplier method and outer approximation.

The following is an algorithm for an implementation of cone decomposition method based on the theorem, the definition, an outer approximation method [3].

Algorithm 3.2 *A combination of cone decomposition method with Lagrangian multiplier and Outer approximation method for OCS problem.*

Step 1

We compute an initial solution $(\hat{y}_i^0, \hat{z}_i^0, \hat{w}_i^0)$ by omitting the quadratic constraint $z_i^2 \leq w_i y_0$ on formulation (3). Let $k = 0$.

Step 2

If $(\hat{z}_i^k)^2 \leq \hat{w}_i^k y_0$ is violated, we compute the projection of $(\hat{w}_i^k, z_i^k)$ onto $z_i^2 \leq w_i z_0$ by solving the following subproblem with the Lagrangian multiplier method.

$$\begin{array}{ll} \textit{maximize} & : \frac{1}{2}\left(\boldsymbol{z} - \hat{y}_i^k\right)^2 + \frac{1}{2}\left(\boldsymbol{w} - \hat{w}_i^k\right)^2 \\ \textit{subject to} & : \boldsymbol{z}^2 \leq \boldsymbol{w} z_0 \end{array}$$

Let a solution of this subproblem be $(\bar{y}_i^k, \bar{z}_i^k, \bar{w}_i^k)$.

Step 3

To apply an outer approximation method [3], we generate the following constraint

$$\begin{pmatrix} z_i^k - \bar{z}_i^k \\ w_i^k - \bar{w}_i^k \end{pmatrix}^T \begin{pmatrix} \hat{z}_i^k - \bar{z}_i^k \\ \hat{w}_i^k - \bar{w}_i^k \end{pmatrix} \leq 0.$$

We add the above constraint if $\bar{z}_i^2 - \bar{w}_i^2 y_0 > 10^{-8}$.

Step 4

Repeat Step 2 and 3 to obtain an optimal solution if the following condition is not hold:

$$||\hat{z}^2 - \bar{z}|| < 10^{-8} \textit{and } ||\hat{w}^2 - \bar{w}|| < 10^{-8} \tag{4}$$

Using Algorithm 3.2, we conduct numerical experiment to compare the performance of our proposed methods with the existing methods.

4 Numerical Result

In the numerical test, we compared the performance of our proposed method with the existing method, OPSEL and GENCONT. We also compared their performances with the optimization solver CPLEX. We used the data from https://doi.org/10.5061/dryad.9pn5m, which was generated by the simulation POPSIM [9], for $m = \{200, 1050, 2045, 5050, 10100, 15222\}$, $N = \{50, 100\}$, and $gap = \{1\%, 5\%\}$. Moreover, we set the computation time limit to 3 hours for all methods except for LPP and LPP-AS.

The numerical experiment was done by using a 64-bit Windows 10 PC on Xeon CPU E3-1231 (3.40 GHz) with 8 memory space. We implemented the proposed methods using Matlab R2016a. In addition, we handled the optimization problems that involve integer constraint using CPLEX.

Table 3 shows the result from a breeding selection solver GENCONT for all m except $m = \{10100, 15222\}$ due to OOM. From Table 3, we observe that the number of selected candidates did not correspond to the given parameter N. This means that GENCONT failed to obtain the optimal solution since the constraint $\boldsymbol{e}^T\boldsymbol{x} = N$ is not satisfied.

Table 3: The result from GENCONT

$N = 50$					
pedigree m	2θ	$\boldsymbol{g}^T\boldsymbol{x}$	$\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}$	Time (s)	# Selected N
200	0.0334	11.472	0.03340	3.54	64
1050	0.0627	25.91	0.06270	7.20	81
2045	0.0711	438.36	0.07109	111.52	71
5050	0.1081	43.44	0.10810	1561.43	78
$N = 100$					
pedigree m	2θ	$\boldsymbol{g}^T\boldsymbol{x}$	$\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}$	Time (s)	# Selected N
200	0.0258	8.89	0.02580	0.48	93
1050	0.0539	24.07	0.0539	4.77	94
2045	0.0628	432.75	0.06279	106.48	74
5050	0.0994	42.08	0.09940	1533.31	81

Table 4: The comparison of the convex relaxation approaches ($N = 50$)

Algorithm	m	2θ	$2\theta(1+\epsilon)$	gap = 5%			gap = 1%		
				$g^T x$	$x^T Ax$	time (s)	$g^T x$	$x^T Ax$	time (s)
CPLEX				24.86	0.03340	5.65	25.21	0.03340	10800.34
OPSEL				25.12	0.03340	5.510	25.18	0.03340	596.57
LPP	200	0.0334	0.033802	25.01	0.03380	10.02	25.32	0.03380	1684.03
LPP-AS			0.033802			TLE			TLE
CDM				25.03	0.03340	2.006	25.19	0.03340	1.83
CPLEX				24.97	0.06243	4.32	25.01	0.06264	10804.34
OPSEL				24.39	0.06254	594.09	24.85	0.06268	7672.56
LPP	1050	0.0627	0.063455	24.87	0.06298	561.74	25.03	0.06362	746.58
LPP-AS			0.063455			TLE			TLE
CDM				24.75	0.06212	7.91	25.02	0.06269	12.78
CPLEX				436.69	0.06860	3.65	436.69	0.06860	3.72
OPSEL				432.94	0.06700	7.09	435.87	0.07020	14.42
LPP	2045	0.0711	0.071956			OOM			OOM
LPP-AS			0.071956			TLE			TLE
CDM				434.26	0.06760	2.05	436.81	0.06860	2.44
CPLEX				41.40	0.10029	2306.99	42.54	0.10724	10809.92
OPSEL				41.574	0.10471	236.70	42.66	0.10814	7277.70
LPP	5050	0.1081	0.109401			OOM			OOM
LPP-AS			0.109401			OOM			OOM
CDM				42.42	0.10677	150.52	42.71	0.10809	190.65
CPLEX				46.13	0.06916	543.07	46.49	0.06990	10845.44
OPSEL				46.00	0.07005	4509.83	46.21	0.06975	8300.24
LPP	10100	0.0701	0.070944			OOM			OOM
LPP-AS			0.070944			OOM			OOM
CDM				46.32	0.06978	922.71	46.33	0.06993	1492.90
CPLEX				460.21	0.03880	1881.41	460.21	0.03880	10826.79
OPSEL				474.10	0.03853	2654.49	478.65	0.03873	10557.52
LPP	15222	0.0388	0.039267			OOM			OOM
LPP-AS			0.039267			OOM			OOM
CDM				452.49	0.03880	551.01	458.83	0.03820	891.515

Table 4 presents the result for the case $N = 50$. We fixed $\epsilon = 0.006$ to generate the solution from the problem with LPP relaxation and its modification (LPP-AS). From Table 4, LPP and LPP-AS failed to obtain the solution due to OOM and time limit exceeded (TLE), even when the time limitation was set into 2 days. This condition is different with the result on Section 2 since we set tighter gap for the solution on here. Moreover this problem is a consequence of very tight ϵ. The chosen $\epsilon = 0.006$ makes LPP cannot generate optimal solution. For example, the problem with $m = 200$ has the value of genetic pedigree $x^T Ax = 0.00380$ in which larger than $2\theta = 0.0334$. Thus, we have to try and find another ϵ, as in Table 5. In contrast to LPP and LPP-AS, the cone decomposition method (CDM) takes shorter computation time than other methods.

Lastly, Table 5 indicates the solution for all methods with $N = 100$. Similarly to the result on the previous table, LPP and LPP-AS got no sensitive solutions for some problems due to TLE and OOM. Besides, the objective solution of LPP-AS for $Z = 100$, which is equal to -1.16, is invalid due to time limitation. In this table, CDM gives better performance on computation time than others. Based on the above observation, CDM is the most effective method to solve OCS problem.

Table 5: The comparison of the convex relaxation approaches ($N = 100$)

Algorithm	Z	2θ	$2\theta(1+\epsilon)$	gap = 5%			gap = 1%		
				$g^T x$	$x^T Ax$	time (s)	$g^T x$	$x^T Ax$	time (s)
CPLEX				23.43	0.02580	2.46	23.55	0.02580	10800.67
OPSEL				23.14	0.02575	1.30	23.54	0.02580	566.89
LPP	200	0.0258	0.026111	23.29	0.02575	1408.79			TLE
LPP-AS			0.026111	23.559	0.02580	2.18	23.42	0.02580	10800.52
CDM				23.55	0.02580	2.18	23.55	0.02580	2.28
CPLEX				22.52	0.05379	2.14	22.52	0.0538	6.18
OPSEL				21.79	0.05358	6.07	22.25	0.05382	193.08
LPP	1050	0.0539	0.054549			TLE			TLE
LPP-AS			0.054549	-1.16	0.03065	12210.27	-1.16	0.03065	21602.16
CDM				22.50	0.05356	15.34	22.504	0.05356	15.14
CPLEX				420.37	0.06140	6.35	420.37	0.06140	6.88
OPSEL				419.53	0.06155	7.93	419.53	0.06155	7.96
LPP	2045	0.0628	0.063556			TLE			TLE
LPP-AS			0.063556	409.96	0.05665	23015.45	409.96	0.05665	32407.06
CDM				418.67	0.06010	2.68	418.67	0.06010	2.73
CPLEX						OOM			OOM
OPSEL				40.13	0.09860	134.55	40.47	0.09936	367.29
LPP	5050	0.0994	0.100698			OOM			OOM
LPP-AS			0.100698			OOM			OOM
CDM				40.49	0.09832	184.23	40.49	0.09832	200.40
CPLEX				44.50	0.06099	961.96	44.44	0.06061	11321.99
OPSEL				443.36	0.06020	584.77	44.44	0.06100	7538.99
LPP	10100	0.0610	0.061734			OOM			OOM
LPP-AS			0.061734			OOM			OOM
CDM				46.32	0.06978	922.71	44.43	0.06082	1137.68
CPLEX				460.21	0.03880	1881.41	460.21	0.03880	10826.79
OPSEL				474.10	0.03853	2654.49	478.65	0.03873	10557.52
LPP	15222	0.0300	0.030361			OOM			OOM
LPP-AS			0.030361			OOM			OOM
CDM				452.49	0.03880	551.01	458.83	0.03820	891.515

5 Conclusion and Future Work

In this study, we proposed the implementation of polyhedral relaxation, which is LPP, LPP-AS, and cone decomposition methods, to optimal contribution selection of tree breeding problem. The computation time problem difficulty from `OPSEL` makes us consider to propose the efficiency methods for solving OCS. We compared the efficiency of our proposed implementation with the existing breeding selection software (`GENCONT` and `OPSEL`) and also with the optimization solver `CPLEX`.

Based on the numerical result, we observed that our proposed relaxations, LPP and LPP-AS, failed to obtain the solution for the problem with larger m due to time limitation and memory size of our environment. This condition occurred since we used very tight ϵ which will increase larger number of constraints. Besides, very tight gaps (5% and 1%) make this method harder. Compared with LPP and LPP-AS, `GENCONT` solved the problem quickly, however it only generated suboptimal solution rather than the optimal one. We also need to consider on choosing best epsilon so that LPP and LPP-AS can guarantee the optimal solution. Therefore, we can conclude that CDM is better than other implementation in our numerical experiment since CDM can efficiently obtain the optimal solution of OCS problem.

In future study, we will consider another problem of OCS that involves not only simple binary constraints but also semi-integer constraints.

References

[1] A. Ben-Tal and A. Nemirovski. On polyhedral approximations of the second-order cone. *Mathematics of Operations Research*, 26(2):193–205, 2001.

[2] C. C. Cockerham. Group inbreeding and coancestry. *Genetics*, 56(1):89, 1967.

[3] M. A. Duran and I. E. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical programming*, 36(3):307–339, 1986.

[4] F. Glineur et al. Computational experiments with a linear approximation of second-order cone optimization. 2000.

[5] I. ILOG. Cplex optimization studio. *URL: http://www-01. ibm. com/software/commerce/optimization/cplex-optimizer*, 2014.

[6] M. Lubin, E. Yamangil, R. Bent, and J. P. Vielma. Extended formulations in mixed-integer convex programming. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 102–113. Springer, 2016.

[7] T. H. Meuwissen. Maximizing the response of selection with a predefined rate of inbreeding. *Journal of animal science*, 75(4):934–940, 1997.

[8] T. H. Meuwissen. Gencont: an operational tool for controlling inbreeding in selection and conservation schemes. In *Proceedings of the 7th Congress on Genetics Applied to Livestock Production*, pages 19–23, 2002.

[9] T. J. Mullin. *Using simulation to optimise tree breeding programmes in Europe: an introduction to POPSIM.* Skogforsk, 2010.

[10] T. J. Mullin, J. Ahlinder, M. Yamashita, and O. Rosvall. Opsel 1.0: A computer program for optimal selection in forest tree breeding by mathematical programming. *Arbetsrapport frr an Skogforsk, Uppsala, Sweden*, 2013.

[11] R. Pong-Wong and J. A. Woolliams. Optimisation of contribution of candidate parents to maximise genetic gain and restricting inbreeding using semidefinite programming (open access publication). *Genetics Selection Evolution*, 39(1):3, 2007.

[12] J. P. Vielma, S. Ahmed, and G. L. Nemhauser. A lifted linear programming branch-and-bound algorithm for mixed-integer conic quadratic programs. *INFORMS Journal on Computing*, 20(3):438–450, 2008.

[13] S. Wright. Coefficients of inbreeding and relationship. *The American Naturalist*, 56(645):330–338, 1922.

[14] M. Yamashita, T. J. Mullin, and S. Safarina. An efficient second-order cone programming approach for optimal selection in tree breeding. *arXiv preprint arXiv:1506.04487*, 2015.