The Implementation of the Primal-Dual Interior-Point Method for the Semidefinite Programs and its Engineering Applications

Katsuki Fujisawa

Department of Architecture and Architectural Systems, Kyoto University.

In resent years, semidefinite program (SDP) has been intensively studied both in theoretical and practical aspects of various fields including interior-point methods, combinatorial optimization and the control and systems theory. The SDPA (SemiDefinite Programming Algorithm) [4] is a C++ implementation of a Mehrotra-type primal-dual predictor-corrector interior-point method for solving the standard form semidefinite program. The SDPA incorporates data structures for handling sparse matrices and an efficient method proposed by Fujisawa *et al.* [5] for computing search directions when problems to be solved are large scale and sparse. Finally, we report numerical experiments of the SDP for the structural optimization under multiple eigenvalue constraints.

1 Introduction.

The main purpose of this paper is to explain the implementation of SDPA (SemiDefinite Programming Algorithm) [4] for semidefinite programs and report some numerical experiments of SDPA. Besides SDPA, there are some computer programs SDPpack [2], SDPSOL [19], CSDP [3], SDPHA [15] and SDPT3 [17] for semidefinite programs which are available through the Internet. The SDPA is a C++ implementation of a Mehrotra-type [10] primal-dual predictor-corrector interior-point method. Although three types of search directions, the HRVW/KSH/M direction [9], the NT direction [13] and the AHO direction [1] are available in SDPA, we employed the HRVW/KSH/M direction in our numerical experiments because its computation is the cheapest among the three directions (particularly, for sparse data matrices) when we employ the method proposed by Fujisawa et al. [5]. Monteiro et al. [12] recently showed that in theory, the NT direction requires less computation for dense matrices. However, their method needs large amount of computational memory and does not efficiently exploit the sparse data structures. Actually, according to their numerical results, the computation of the HRVW/KSH/M direction is favorable compared to the the computation of the NT and AHO directions.

Let $R^{n\times n}$ and $\mathcal{S}^n \subset R^{n\times n}$ denote the set of all $n\times n$ real matrices and the set of all $n\times n$ real symmetric matrices, respectively. We use the notation $\boldsymbol{U}\bullet\boldsymbol{V}$ for the inner product of $\boldsymbol{U},\ \boldsymbol{V} \in R^{n\times n}$, *i.e.*, $\boldsymbol{U}\bullet\boldsymbol{V} = \sum_{i=1}^n\sum_{j=1}^n U_{ij}V_{ij}$, where U_{ij} and V_{ij} denote the (i,j)th element of $\boldsymbol{U}$ and $\boldsymbol{V}$, respectively. We write $\boldsymbol{X} \succeq \boldsymbol{O}$ and $\boldsymbol{X} \succ \boldsymbol{O}$ when $\boldsymbol{X} \in \mathcal{S}^n$ is positive semidefinite and positive definite, respectively.

Let $\boldsymbol{A}_i \in \mathcal{S}^n$ $(0 \leq i \leq m)$ and $b_i \in R$ $(1 \leq i \leq m)$. SDPA solves the standard form semidefinite program and its dual:

$$\left.\begin{array}{l}\mathcal{P}\text{: minimize } \boldsymbol{A}_0\bullet\boldsymbol{X} \text{ subject to } \boldsymbol{A}_i\bullet\boldsymbol{X} = b_i \ (1\leq i\leq m),\ \boldsymbol{X}\succeq\boldsymbol{O}.\\ \mathcal{D}\text{: maximize } \displaystyle\sum_{i=1}^m b_i y_i \text{ subject to } \sum_{i=1}^m \boldsymbol{A}_i y_i + \boldsymbol{Z} = \boldsymbol{A}_0,\ \boldsymbol{Z}\succeq\boldsymbol{O}.\end{array}\right\} \qquad (1)$$

For simplicity, we say that $(\boldsymbol{X},\boldsymbol{y},\boldsymbol{Z})$ is a feasible solution (an interior-feasible solution, or an optimal solution, respectively) of the SDP (1) if $\boldsymbol{X}$ is a feasible solution (an interior-feasible solution, *i.e.,* a feasible solution satisfying $\boldsymbol{X}\succ\boldsymbol{O}$ or a minimizing solution, respectively) of $\mathcal{P}$ and $(\boldsymbol{y},\boldsymbol{Z})$ is a feasible solution (an interior-feasible solution, *i.e.,* a feasible solution satisfying $\boldsymbol{Z}\succ\boldsymbol{O}$ or a maximizing solution, respectively) of $\mathcal{D}$.

In Section 2, we present some issues on the implementation of SDPA which are relevant for our numerical experiments. Section 3 is devoted to SDPs for structural optimization under multiple eigenvalue constraints. In Section 4, we present numerical results of the SDPA. Section 5 gives conclusions.

2 Some Issues of the Implementation of SDPA.

In this section, we describe the data structures for handling block diagonal matrices and the HRVW/KSH/M search direction both used in SDPA. Finally, we explain the algorithmic framework of SDPA.

2.1 Data Structures

The SDPA can handles block diagonal matrices. In terms of the number of blocks, denoted by nBLOCK, and the block structure vector, denoted by bLOCKsTRUCT, we express a common matrix data structure for the constraint matrices $\boldsymbol{A}_0,\ \boldsymbol{A}_1,\ \ldots,\ \boldsymbol{A}_m$. If we deal with a block diagonal matrix $\boldsymbol{A}$ of the form

$$\left.\begin{array}{ll} \boldsymbol{A} \;= & \begin{pmatrix} \boldsymbol{B}_1 & \boldsymbol{O} \, \boldsymbol{O} \cdots & \boldsymbol{O} \\ \boldsymbol{O} \, \boldsymbol{B}_2 & \boldsymbol{O} \cdots & \boldsymbol{O} \\ \cdot \quad \cdot & \cdots\cdot & \boldsymbol{O} \\ \boldsymbol{O} \quad \boldsymbol{O} & \boldsymbol{O} \cdots & \boldsymbol{B}_\ell \end{pmatrix}, \\ \boldsymbol{B}_i : & \text{a } p_i \times p_i \text{ symmetric matrix } \ (i = 1, 2, \ldots, \ell), \end{array}\right\} \tag{2}$$

we define the number nBLOCK of blocks and the block structure vector bLOCKsTRCTURE as follows:

$$\begin{aligned} \text{nBLOCK} \; &= \ell, \\ \text{bLOCKsTRUCT} \; &= (\beta_1,\ \beta_2,\ \ldots,\ \beta_\ell), \\ \beta_i &= \begin{cases} \ \ p_i \text{ if } \boldsymbol{B}_i \text{ is a symmetric matrix}, \\ -p_i \text{ if } \boldsymbol{B}_i \text{ is a diagonal matrix}. \end{cases} \end{aligned}$$

For example, if $\boldsymbol{A}$ is of the form

$$\begin{pmatrix} 1\,2\,3\,0\,0\,0\,0 \\ 2\,4\,5\,0\,0\,0\,0 \\ 3\,5\,6\,0\,0\,0\,0 \\ 0\,0\,0\,1\,2\,0\,0 \\ 0\,0\,0\,2\,3\,0\,0 \\ 0\,0\,0\,0\,0\,4\,0 \\ 0\,0\,0\,0\,0\,0\,5 \end{pmatrix}, \tag{3}$$

we have

$$\text{nBLOCK} = 3 \quad \text{and} \quad \text{bLOCKsTRUCT} = (3,\ 2,\ -2).$$

2.2 Search Direction.

The HRVW/KSH/M direction is the solution $(d\boldsymbol{X}, d\boldsymbol{y}, d\boldsymbol{Z})$ of the system of equations

$$\boldsymbol{A}_i \bullet d\boldsymbol{X} = p_i \ (1 \le i \le m),\ d\boldsymbol{X} \in \mathcal{S}^n, \tag{4}$$

$$\sum_{i=1}^m \boldsymbol{A}_i dy_i + d\boldsymbol{Z} = \boldsymbol{D},\ d\boldsymbol{Z} \in \mathcal{S}^n, \tag{5}$$

$$\widehat{d\boldsymbol{X}} \boldsymbol{Z} + \boldsymbol{X} d\boldsymbol{Z} = \boldsymbol{K},\ \widehat{d\boldsymbol{X}} \in R^{n\times n},\ d\boldsymbol{X} = (\widehat{d\boldsymbol{X}} + \widehat{d\boldsymbol{X}}^T)/2. \tag{6}$$

Here $p_i \in R$, $\boldsymbol{D} \in \mathcal{S}^n$ and $\boldsymbol{K} \in R^{n\times n}$ denote a scalar constant, an $n \times n$ constant symmetric matrix, and an $n \times n$ constant matrix, respectively, which are determined by the current point $(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{Z})$ and some other factors. Note that $\widehat{d\boldsymbol{X}} \in R^{n\times n}$ serves as an auxiliary variable matrix. Under the linear independence assumption on the set $\{\boldsymbol{A}_i : 1 \le i \le m\}$ of constraint matrices, we know [9] that for any $\boldsymbol{X} \succ \boldsymbol{O}$, $\boldsymbol{Z} \succ \boldsymbol{O}$, $p_i \in R$ $(1 \le i \le m)$, $\boldsymbol{D} \in \mathcal{S}^n$, and $\boldsymbol{K} \in R^{n\times n}$, the system of equations (4), (5) and (6) has a unique solution $(d\boldsymbol{X}, d\boldsymbol{y}, d\boldsymbol{Z})$.

We can reduce the system of equations (4), (5) and (6) to

$$\boldsymbol{B}d\boldsymbol{y} = \boldsymbol{b}, \tag{7}$$

$$\left.\begin{array}{l} d\boldsymbol{Z} = \boldsymbol{D} - \displaystyle\sum_{i=1}^{m} \boldsymbol{A}_i dy_i, \\ \widehat{d\boldsymbol{X}} = (\boldsymbol{K} - \boldsymbol{X}d\boldsymbol{Z})\,\boldsymbol{Z}^{-1},\ d\boldsymbol{X} = (\widehat{d\boldsymbol{X}} + \widehat{d\boldsymbol{X}}^T)/2, \end{array}\right\} \tag{8}$$

where

$$\left.\begin{array}{l} B_{ij} = \boldsymbol{X}\boldsymbol{A}_i\boldsymbol{Z}^{-1} \bullet \boldsymbol{A}_j \ (1 \le i \le m,\ 1 \le j \le m) \\ b_i = p_i - (\boldsymbol{K} - \boldsymbol{X}\boldsymbol{D})\boldsymbol{Z}^{-1} \bullet \boldsymbol{A}_i \ (1 \le i \le m). \end{array}\right\} \tag{9}$$

The matrices $\boldsymbol{X}$, $\boldsymbol{Z}^{-1}$ and $\boldsymbol{B}$ are symmetric and dense in general even when all $\boldsymbol{A}_i$ $(1 \le i \le m)$ are sparse. Hence solving the system of equations (7) in $d\boldsymbol{y}$ by using a direct method such as the Cholesky factorization and the LDL^T factorization of $\boldsymbol{B}$ requires $O(m^3)$ arithmetic operations. On the other hand, if we treat all $\boldsymbol{A}_i$ $(1 \le i \le m)$ as dense matrices and if we use the above formulae (9) for the coefficient matrix $\boldsymbol{B}$ in a straightforward way, the computation of $\boldsymbol{B}$ requires $O(mn^3 + m^2n^2)$ arithmetic operations. Therefore computing the coefficient matrix $\boldsymbol{B}$ is more crucial than solving $\boldsymbol{B}d\boldsymbol{y} = \boldsymbol{b}$ in the entire computation of the HRVW/KSH/M direction.

In their paper [5], Fujisawa, Kojima and Nakata proposed three distinct formulae $\mathcal{F}_1$, $\mathcal{F}_2$ and $\mathcal{F}_3$ for computing $\boldsymbol{B}$, and their efficient combination $\mathcal{F}_*$. They demonstrated through numerical experiments that the combined formula $\mathcal{F}_*$ worked very efficiently when some of $\boldsymbol{A}_i$ $(1 \le i \le m)$ are sparse. We incorporated their formula $\mathcal{F}_*$ into SDPA. See the paper [5] for more details.

2.3 The Algorithmic Framework of SDPA.

Step 0 : Set an initial point $(\boldsymbol{X}^0, \boldsymbol{y}^0, \boldsymbol{Z}^0)$ with $\boldsymbol{X}^0 \succ \boldsymbol{O}, \boldsymbol{Z}^0 \succ \boldsymbol{O}$. Decide on the search direction to use. Set the parameters: $0.0 < \epsilon^*$, $1 < \omega^*$, $0.01 \le \beta^* \le 0.10$ and $\beta^* \le \bar{\beta} \le 0.20$. (The default values of these parameters are: $\epsilon^* = 1.0E-5$, $\omega^* = 2.0$, $\beta^* = 0.05$ and $\bar{\beta} = 0.1$). Let $k = 0$.

Step 1 : If the current iterate $(\boldsymbol{X}^k, \boldsymbol{y}^k, \boldsymbol{Z}^k)$ is feasible and the relative gap

$$\frac{|P - D|}{\max\{1.0,\ (|P| + |D|)/2\}}$$

gets smaller than ϵ^*, then stop the iteration. Here P and D denote the primal and the dual objective values, respectively. If we detect that there is no feasible solution $(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{Z})$ such that $\omega^*\boldsymbol{X}^0 \succeq \boldsymbol{X} \succeq \boldsymbol{O}$ and $\omega^*\boldsymbol{Z}^0 \succeq \boldsymbol{Z} \succeq \boldsymbol{O}$, then stop the iteration. See [9] for details on how to get such information on infeasibility.

Step 2 : (Predictor Step) Let

$$\beta_p = \begin{cases} 0 \text{ if the current iterate is feasible,} \\ \bar{\beta} \text{ otherwise.} \end{cases}$$

Solve the system of equations (4), (5) and (6) with $\boldsymbol{K} = \beta_p(\boldsymbol{X} \bullet \boldsymbol{Z}/n)\boldsymbol{I} - \boldsymbol{X}\boldsymbol{Z}$ to compute the predictor direction $(d\boldsymbol{X}_p, d\boldsymbol{y}_p, d\boldsymbol{Z}_p)$.

Step 3 : (Corrector Step) Let

$$\beta = \frac{(\boldsymbol{X} + \bar{\alpha}_p d\boldsymbol{X}_p) \bullet (\boldsymbol{Z} + \bar{\alpha}_d d\boldsymbol{Z}_p)}{(\boldsymbol{X} \bullet \boldsymbol{Z})},$$

where $\bar{\alpha}_p$ and $\bar{\alpha}_d$ are computed as in [6]. Choose the parameter β_c as follows:

$$\beta_c = \begin{cases} \max\{\beta^*, \beta^2\} & \text{if the current iterate is feasible and } \beta \le 1.0, \\ \max\{\bar{\beta}, \beta^2\} & \text{if the current iterate is infeasible and } \beta \le 1.0, \\ 1.0 & \text{otherwise.} \end{cases}$$

Compute the corrector direction $(d\boldsymbol{X}_c, d\boldsymbol{y}_c, d\boldsymbol{Z}_c)$ by solving the system of equations (4), (5) and (6) with $\boldsymbol{K} = \beta_c(\boldsymbol{X} \bullet \boldsymbol{Z}/n)\boldsymbol{I} - \boldsymbol{X}\boldsymbol{Z} - d\boldsymbol{X}_p d\boldsymbol{Z}_p$.

Step 4 : Set the next iterate $(\boldsymbol{X}^{k+1}, \boldsymbol{y}^{k+1}, \boldsymbol{Z}^{k+1})$ such that

$$\boldsymbol{X}^{k+1} = \boldsymbol{X}^k + \alpha_p d\boldsymbol{X}_c \;\text{ and }\; (\boldsymbol{y}^{k+1}, \boldsymbol{Z}^{k+1}) = (\boldsymbol{y}^k, \boldsymbol{Z}^k) + \alpha_d(d\boldsymbol{y}_c, d\boldsymbol{Z}_c),$$

where α_p and α_d are computed as in [6].

Step 5 : $k \longleftarrow k+1$ and go to Step 1.

3 Semidefinite Programming for Structural Optimization under Multiple Eigenvalue Constraints.

The eigenvalues of free vibration as well as the linear buckling load factor, which are calculated by solving linear eigenvalue problems, are important performance measures of the structures. Therefore, there have been many researches for optimization of structures under eigenvalue constraints. Consider a truss with fixed locations of nodes and members which can exist. The vector of member cross-sectional areas is denoted by $\mathbf{A} = \{A_i\}$. Let $\mathbf{K}$ and $\mathbf{M}_s$ denote the stiffness matrix and the mass matrix due to the structural mass which are functions of $\mathbf{A}$. The mass matrix for nonstructural mass is denoted by $\mathbf{M}_0$. The eigenvalue problem of vibration is formulated as

$$\mathbf{K}\Phi_r = \Omega_r(\mathbf{M}_s + \mathbf{M}_0)\Phi_r, \quad (r = 1, 2, \ldots, N^d) \tag{10}$$

where Ω_r and Φ_r are the rth eigenvalue and eigenvector, and N^d is the number of freedom of displacements. The eigenvector Φ_r is normalized by

$$\Phi_r^T \mathbf{M} \Phi_r = 1. \quad (r = 1, 2, \ldots, N^d) \tag{11}$$

Let $\bar{\Omega}$ denote the specified lower bound of the eigenvalues. The topology optimization problem for specified fundamental eigenvalue is formulated as

$$\text{OPE:} \quad \text{Minimize} \quad \sum_{i=1}^{N^m} L_i A_i \tag{12}$$

$$\text{subject to} \quad \Omega_r \ge \bar{\Omega}, \quad (r = 1, 2, \ldots, N^d) \tag{13}$$

$$A_i \ge 0, \quad (i = 1, 2, \cdots, N^m) \tag{14}$$

where L_i is the length of the ith member, and N^m is the number of members. The optimal topology is found by removing the members with $A_i \le \epsilon$, where ϵ is a small positive lower bound. A small positive lower bound is usually given for A_i to prevent instability of the structure.

It is well known that optimum designs for specified fundamental eigenvalue often have multiple or repeated eigenvalues. If the fundamental eigenvalue of the optimum design is simple, OPE may easily be solved by using a nonlinear programming or an optimality criteria approach [18, 16], because there is no difficulty in calculating the sensitivity coefficients of Ω_1 with respect to A_i. In the case of multiple eigenvalue, only directional sensitivity coefficients can be calculated [7]. Although some formulations of sensitivity analysis of repeated eigenvalue has been presented [8, 11], it is not clear if those formulations can be used for optimizing large structures. In this paper, the eigenvalue problem (10) is converted into a standard form, and solved by using semi-definite programming.

Consider a structure where $\Omega_1 \geq \bar{\Omega}$ is satisfied. In this case the Rayleigh's principle leads to the following inequality for any admissible mode ψ:

$$\psi^T(\mathbf{K} - \bar{\Omega}\mathbf{M})\psi \geq 0 \tag{15}$$

where (11) has been used. This inequality implies that the matrix $\mathbf{K} - \bar{\Omega}\mathbf{M}$ is positive semi-definite, and formulations of SDP may be possible. The matrices $\mathbf{K}_i$ and $\mathbf{M}_i$ are defined as

$$\mathbf{K}_i = \frac{\partial \mathbf{K}}{\partial A_i}, \quad \mathbf{M}_i = \frac{\partial \mathbf{M}_s}{\partial A_i}. \tag{16}$$

Since $\mathbf{K}$ and $\mathbf{M}_s$ are linear functions of A_i for trusses, those are written as

$$\mathbf{K} = \sum_{i=1}^{N^m} \mathbf{K}_i A_i, \quad \mathbf{M}_s = \sum_{i=1}^{N^m} \mathbf{M}_i A_i \tag{17}$$

The problems $\mathcal{P}$ and $\mathcal{D}$ for this case are formulated as

$$\begin{aligned}
\mathcal{P}\text{'}: \quad & \text{Minimize} \quad \sum_{i=1}^{N^m} L_i y_i & (18)\\
& \text{subject to} \quad \mathbf{X} = \sum_{i=1}^{N^m} (\mathbf{K}_i - \bar{\Omega}\mathbf{M}_i) y_i - \bar{\Omega}\mathbf{M}_0 & (19)\\
& \qquad \mathbf{X} \in \mathcal{S}^n, \mathbf{X} \succeq \boldsymbol{O}. & (20)\\
\mathcal{D}\text{'}: \quad & \text{Maximize} \quad \bar{\Omega}\mathbf{M}_0 \bullet \mathbf{Y} & (21)\\
& \text{subject to} \quad (\mathbf{K}_i - \bar{\Omega}\mathbf{M}_i) \bullet \mathbf{Y} = L_i & (22)\\
& \qquad \mathbf{Y} \in \mathcal{S}^n, \mathbf{Y} \succeq \boldsymbol{O}. & (23)\\
& & (24)
\end{aligned}$$

Problems $\mathcal{P}'$ and $\mathcal{D}'$ are solved successively to find optimal solutions. It is important to note here that the design sensitivity coefficients of eigenvalues with respect to the design variables are not needed in the optimization process. Therefore, there is no difficulty, as shown in the examples, in finding the solutions with multiple fundamental eigenvalues.

4 Numerical Results.

Optimal topologies are found for plane trusses, and the computational efficiency and accuracy of the results are compared among the proposed method using SDP and Parametric Programming (PP) [14]. In the following example, the material of the members are steel where elastic modulus E is 205.8 GPa and the mass density ρ is 7.86×10^{-3} kg/cm^2. In SDP, E and ρ are scaled so that $E = 1000.0$ is satisfied to prevent divergence in the process of finding a feasible solution. The specified eigenvalue is 1000.0 rad^2/s^2 for all the cases. The computation has been carried out on Sun Ultra 2 Model 1300 (UltraSPARC-II 296MHz).

Table 1: Comparison of performances of SDP and PP.

		SDP	Parametric
	Volume (cm^3)	6.4493×10^5	6.4479×10^5
Arch	Ω_1 (rad^2/s^2)	1000.0	982.58
($N^m = 174$)	Ω_2 (rad^2/s^2)	1000.0	1028.2
($N^d = 106$)	$\bar{A}_i$ (cm^2)	0.0	0.001
	CPU (s)	38.00(SDPA)	1382.81
		599.01(CSDP)	

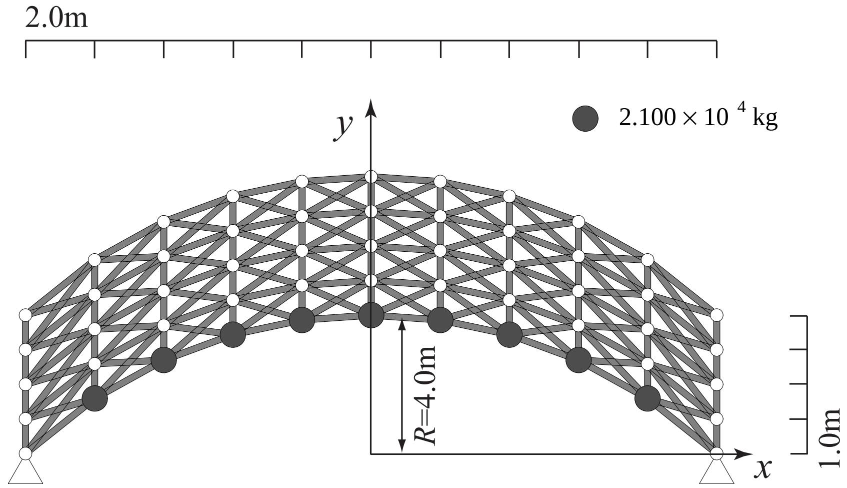

Figure 1: A plane arch grid.

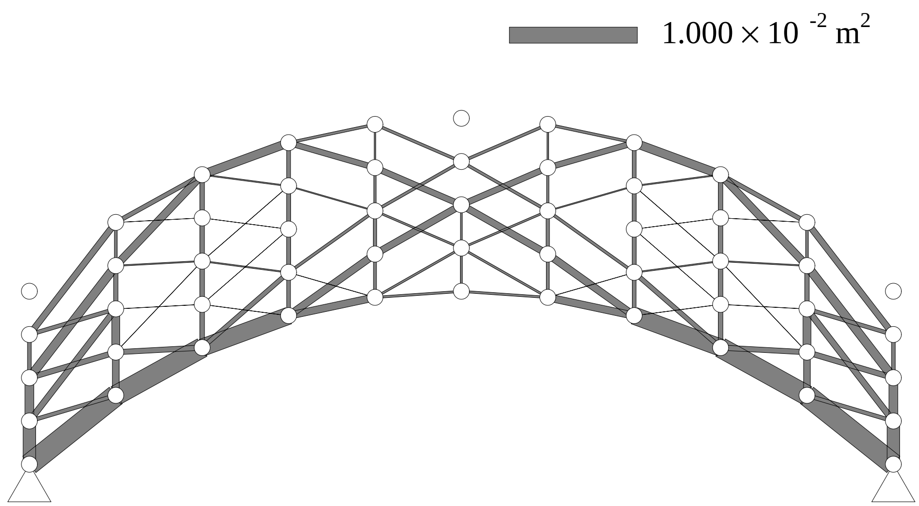

Figure 2: Optimal topology of the plane arch grid.

4.1 A plane arch grid with repeated eigenvalue

Consider a plane arch grid as shown in Fig. 1. Nonstructural masses are located at the nodes along the lowest circle. The optimal topology found by SDP after removing the members with $A_i \leq \epsilon$ is as shown in Fig. 2. The results by SDP and PP are listed in Table 1. The cross-sectional areas are linked in PP so that only symmetric designs are allowed in the optimal topology. A symmetric solution has been successfully found by SDP without assigning those additional constraints.

The multiplicity of the fundamental eigenvalues of the optimal truss is two. It should be noted here that the CPU time of PP is very large compared with that of SDP, because the number of linear equations to be solved simultaneously in PP is almost proportional to the multiplicity of the fundamental eigenvalue. We can say that the SDPA [4] is much faster than the CSDP [3] from Table 1.

5 Conclusions

Fujisawa et al. [6] reported that the SDPA is much faster than other softwares for SDPs because the SDPA incorporates data structures for handling sparse matrices and an efficient method proposed by Fujisawa *et al.* [5] for computing search directions when problems to be solved are large scale and sparse. Sparsity is one particular form of problem structure we exploit here because it is an important feature arising from SDP relaxation in combinatorial optimization and it has a straightforward tractable structure. However, we need an accumulation of more numerical results and knowledge for exploiting other problem structures and we consider them for future work.

The optimum design problem has been formulated as a SDP, and an algorithm has been presented for topology optimization of trusses for specified fundamental eigenvalue of vibration. The proposed algorithm is very effective for the cases of optimum designs with multiple eigenvalues, because sensitivity coefficients of the eigenvalues with respect to the design variables are not needed in the algorithm. We have seen in other examples the optimum design with up to five-fold fundamental eigenvalues can be found without any difficulty. Note that no drastic increase has been observed in CPU time for the case of multiple eigenvalues. In addition of these advantages, symmetric solutions are found without assigning any side constraints for imposing symmetricity of the cross-sectional areas.

References

[1] F. Alizadeh, J.-P.A. Haeberly and M.L. Overton, "Primal-dual interior-point methods for semidefinite programming," Working Paper, 1994.

[2] F. Alizadeh, J.-P.A. Haeberly, M.V. Nayakkankuppam, M.L. Overton and S. Schmieta, "SDPpack – User's Guide –," Comp. Sci. Dept., New York University, New York, June 1997. Available at http://cs.nyu.edu/cs/faculty/overton/sdppack/sdppack.html.

[3] B. Borchers, "CSDP, a C library for semidefinite programming," Department of Mathematics, New Mexico Institute of Mining and Technology, 801 Leroy Place Socorro, New Mexico 87801, March 1997. Available at http://www.nmt.edu/~borchers/csdp.html.

[4] K. Fujisawa, M. Kojima and K. Nakata, "SDPA (Semidefinite Programming Algorithm) – User's Manual –," Technical Report B-308, Tokyo Institute of Technology, Oh-Okayama, Meguro, Tokyo 152, Japan, December 1995, Revised August 1996.

[5] K. Fujisawa, M. Kojima and K. Nakata, "Exploiting sparsity in primal-dual interior-point methods for semidefinite programming," *Mathematical Programming* **79** (1997) 235–253.

[6] K. Fujisawa, M. Fukuda, M. Kojima and K. Nakata, "Numerical Evaluation of the SDPA (SemiDefinite Programming Algorithm)", Technical Report B-330, Tokyo Institute of Technology, Oh-Okayama, Meguro, Tokyo 152, Japan, August 1997, Revised on June 1998.

[7] E.J. Haug and J. Cea (eds.), Optimization of Distributed Parameter Structures, Sijthoff & Noordhoff, 1981.

[8] E.J. Haug and K.K. Choi, Systematic occurrence of repeated eigenvalues in structural optimization, J. Optimization Theory and Appl., Vol. 38, pp. 251-274, 1982.

[9] M. Kojima, S. Shindoh and S. Hara, "Interior-point methods for the monotone semidefinite linear complementarity problems," *SIAM Journal on Optimization* **7** (1997) 86–125.

[10] S. Mehrotra, "On the implementation of a primal-dual interior point method," *SIAM Journal on Optimization* **2** (1992) 575–601.

[11] A.P. Seyranian, E. Lund and N. Olhoff, Multiple eigenvalues in structural optimization problem, Structural Optimization, Vol. 8, pp. 207-227, 1994.

[12] R.D.C. Monteiro, P.R. Zanjácomo, "Implementation of primal-dual methods for semidefinite programming based on Monteiro and Tsuchiya directions and thier variants," Technical Report, School Industrial and Systems Engineering, Georgia Tech., Atlanta, GA 30332, July 1997, Revised August 1997.

[13] Yu.E. Nesterov and M.J. Todd, "Self-scaled barriers and interior-point methods in convex programming," *Mathematics of Operations Research* **22** (1997) 1–42.

[14] Tsuneyoshi Nakamura and M. Ohsaki, A natural generator of optimum topology of plane trusses for specified fundamental frequency, Comput. Meth. Appl. Mech. Engng., Vol.94, pp. 113-129, 1992.

[15] F.A. Potra, R. Sheng and N. Brixius, "SDPHA – a MATLAB implementation of homogeneous interior-point algorithms for semidefinite programming," Department of Mathematics, University of Iowa, Iowa City, IA 52242, April 1997. Available at http://www.math.uiowa.edu/~rsheng/SDPHA/sdpha.html.

[16] E. Sadek, An optimality criterion method for dynamic optimization of structures, Int. J. Numer. Meth. Engng., Vol. 28, pp. 579-592, 1989.

[17] K.C. Toh, M.J. Todd and R.H. Tütüncü, "SDPT3 – a MATLAB software package for semidefinite programming," Dept. of Mathematics, National University of Singapore, 10 Kent Ridge Crescent, Singapore, December 1996. Available at http://www.math.nus.sg/~mattohkc/index.html.

[18] V.B. Venkayya and V.A. Tishler, Optimization of structures with frequency constraints, in: Computer Methods in Nonlinear Solids Structural Mechanics ASME-AMD-54, ASME, New York, pp. 239-259, 1983.

[19] S.-P. Wu and S. Boyd. "SDPSOL: a parser/solver for SDP and MAXDET problems with matrix structure", Department of Electrical Engineering, Stanford University, June, 1996. Available at http://www-isl.stanford.edu/people/boyd/SDPSOL.html