Semidefinite Programming in Timetabling II: Algorithms

Jakub Mareček · Andrew J. Parkes

Introduction Semidefinite programming (SDP) is a subfield of convex optimisation (Wolkowicz, Saigal, & Vandenberghe, 2000). It has recently gained considerable attention, as it makes it possible to derive strong lower bounds for minimisation problems in combinatorial optimisation (Goemans & Rendl, 2000), as well as to obtain very good solutions using randomised rounding. In some of the present-best approximation algorithms, both lower and upper bounds are obtained in this fashion. There seem to be, however, only few applications to practical scheduling, timetabling, or rostering problems.

At PATAT 2010, (Burke, Mareček, & Parkes, 2011) presented an SDP relaxation of bounded graph colouring, which can be used to detection the infeasibility in many timetabling problems. Subsequently, they have introduced relaxation for a number of timetabling problems in an extended version of their paper. In this abstract, we present augmented Lagrangian methods, also known as boundary point methods or proximal methods, for solving such relaxations.

Semidefinite Programming Semidefinite programming (SDP, Bellman & Fan, 1963; Alizadeh, 1995; Wolkowicz et al., 2000) is a popular generalisation of linear programming, replacing the vector variable with a square symmetric matrix variable and the polyhedral symmetric convex cone of the positive orthant with the non-polyhedral symmetric convex cone of positive semidefinite matrices. The primal-dual pair in the standard form is:

$$z_p = \min_{X \in \mathscr{S}^n} \langle C, X \rangle \text{ s.t. } \mathscr{A}_A(X) = b \text{ and } X \succeq 0 \tag{P SDP}$$

$$z_d = \max_{y \in \mathbb{R}^m, S \in \mathscr{S}^n} b^T y \text{ s.t. } \mathscr{A}_A^*(y) + S = C \text{ and } S \succeq 0 \tag{D SDP}$$

where X is a primal variable in the set of $n \times n$ symmetric matrices $\mathscr{S}^n$, y and S are the corresponding dual variables, b is an m-vector, C, A_i are compatible matrices, and linear operator $\mathscr{A}_A(X)$ maps symmetric $n \times n$ matrices to vectors in $\mathbb{R}^m$. The ith element $\mathscr{A}_A(X)_i = \langle A_i, X \rangle$, and the adjoint is $\mathscr{A}_A^*(y) = \sum_i y_i A_i$. $M \succeq N$ or $M - N \succeq$ denotes $M - N$ is positive semidefinite. Note that an $n \times n$ matrix, M, is positive semidefinite if and only if $y^T M y \geq 0$ for all $y \in \mathbb{R}^n$.

Contact author: J. Mareček, E-mail: jakub@marecek.cz
School of Mathematics, The University of Edinburgh, JCMB among the King's Buildings, Edinburgh EH9 3JZ.

A. Parkes, E-mail: ajp@cs.nott.ac.uk
School of Computer Science, The University of Nottingham, Nottingham NG8 1BB, UK.

All relaxations of (Burke et al., 2011), use linear equalities given by the adjacency matrix of the conflict graph, further linear equalities and further linear inequalities. These are best treated explictly in the primal-dual pair:

$$z_p = \min_{X \in \mathscr{S}^n} \langle C, X \rangle \text{ s.t. } \mathscr{A}_{A_1}(X) = b_1 \text{ and } \mathscr{A}_{A_2}(X) = b_2 \text{ and } \mathscr{A}_B(X) \geq d \text{ and } X \succeq 0$$

$$z_d = \max_{y_1 \in \mathbb{R}^m, y_2 \in \mathbb{R}^p, v \in \mathbb{R}^q, S \in \mathscr{S}^n} b_1^T y_1 + b_2^T y_2 + d^T v \qquad (1)$$

$$\text{s.t. } \mathscr{A}_{A_1}^*(y_1) + \mathscr{A}_{A_2}^*(y_2) + \mathscr{A}_B^*(v) + S = C \text{ and } S \succeq 0 \text{ and } v \geq 0.$$

where $\mathscr{A}_{A_1}(X), b_1, \mathscr{A}_{A_2}(X), b_2$ are given by the conflict graph and further linear equalities, specific to a particular timetabling problem, respectively, d is a q-vector, and linear operator $\mathscr{A}_B(X)$ maps $n \times n$ matrices to q-vectors similarly to $\mathscr{A}_A$ above. As all linear combinations with non-negative coefficients of positive semidefinite matrices are positive semidefinite, $X \succeq 0$ should again be seen as a restriction to a convex cone. This extends the approach of Wen, Goldfarb, and Yin (2010), who treat linear inequalities explicitly.

Semidefinite Programming Solvers Traditionally, SDP is solved using primal-dual interior point methods (Wright, 1997): From the KKT conditions, comprising of the primal (P SDP) and dual (D SDP) problems and the complementarity condition $ZX = 0$, one derives the "Newton system" by relaxing the complementarity condition to $ZX = \mu I$ or similar. These methods are also referred to as second-order, as they employ the second-order partial derivatives, unlike first-order methods, which use only first derivatives. Povh, Rendl, and Wiegele (2006) observe that implementations of second-order methods for computing theta-like SDP relaxations are currently limited to graphs of about 10,000 edges, which amounts to little more than 100 vertices in the dense graphs found in timetabling applications.

First-order Lagrangian methods have long been used as an alternative. In iteration k of solving a semidefinite program in standard form, one updates X^k to X^{k+1} as follows:

$$(y^{k+1}, S^{k+1}) = \text{argmin}_{y,S} -b^T y + \langle X^k, \mathscr{A}_A^*(y) + S - C \rangle \qquad (2)$$

$$X^{k+1} = X^k + \mu^{-1}(\mathscr{A}_A^*(y^{k+1}) + S^{k+1} - C) \qquad (3)$$

This approach suffers from two major drawbacks: the convergence may be frail and the minimisation of the Lagrangian (2) may turn out to be expensive. The first drawback may be alleviated by augmenting the Lagrangian (Powell, 1969; Hestenes, 1969) with a Frobenius norm term:

$$L_\mu(X, y, S) = -b^T y + \langle X, \mathscr{A}_A^*(y) + S - C \rangle + \frac{1}{2\mu} ||\mathscr{A}_A^*(y) + S - C||_F^2 \qquad (4)$$

The second drawback can be alleviated by minimising the Lagrangian first for y and only subsequently for S. Using fixed point arguments, one can still show the convergence of such two-step minimisation.

Our Solver The augmented Lagrangian of the dual (1) is:

$$\begin{aligned} L_\mu(X, y_1, y_2, v, S) = &- b_1^T y_1 - b_2^T y_2 - d^T v \qquad (5) \\ &+ \langle X, \mathscr{A}_{A_1}^*(y_1) + \mathscr{A}_{A_2}^*(y_2) + \mathscr{A}_B^*(v) + S - C \rangle \\ &+ \frac{1}{2\mu} ||\mathscr{A}_{A_1}^*(y_1) + \mathscr{A}_{A_2}^*(y_2) + \mathscr{A}_B^*(v) + S - C||_F^2 \end{aligned}$$

The multiple splitting is elaborated in Algorithm Schema 1.

Algorithm Schema 1 AugmentedLagrangianMethod$(A_1, A_2, B, C, b_1, b_2, d)$

1: **Input:** Instance I = $(A_1, A_2, B, C, b_1, b_2, d)$ of SDP (1)
2: **Output:** Primal solution Y, computed up to a certain precision
3: Set iteration counter $k = 0$
4: Initialise $X^k \succeq 0$ with a heuristically obtained colouring
5: Compute matching values of dual variables $y_1^k, y_2^k, v^k \geq 0$, and $S^k \succeq 0$
6: **while** the precision is insufficient **do**
7: Increase iteration counter k
8: Update y_1^{k+1} $= \operatorname{argmin}_{y_1 \in \mathbb{R}^m} L_\mu(X^k, y_1, y_2^k, v^k, S^k)$ $= -(A_1 A_1^T)^{-1}(\mu(A_1(X^k) - b_1) + A_1(A_2^T(y_2^k) + B^T(v^k) + S^k - C))$
9: Update y_2^{k+1} $= \operatorname{argmin}_{y_2 \in \mathbb{R}^m} L_\mu(X^k, y_1^{k+1}, y_2, v^k, S^k)$ $= -(A_2 A_2^T)^{-1}(\mu(A_2(X^k) - b_2) + A_2(A_1^T(y_1^{k+1}) + B^T(v^k) + S^k - C))$
10:
11: Update v^{k+1} $= \operatorname{argmin}_{v \in \mathbb{R}^q, v \geq 0} L_\mu(X^k, y_1^{k+1}, y_2^{k+1}, v, S^k)$ $= \operatorname{argmin}_{v \in \mathbb{R}^q, v \geq 0} \left(\left(B \left(X^k + \frac{1}{\mu} \left(A_1^T(y_1^{k+1}) + A_2^T(y_2^{k+1}) + S^k - C \right) \right) - d \right)^T v + \frac{1}{2\mu} v^T (BB^T) v \right)$ which is a single-cone second-order cone program (Alizadeh & Goldfarb, 2003)
12: Update S^{k+1} $= \operatorname{argmin}_{S \succeq 0} L_\mu(X^k, y_1^{k+1}, y_2^{k+1}, v^{k+1}, S)$ $= \operatorname{argmin}_{S \in \mathscr{S}^n, S \succeq 0} \left\| S - \left(C - A_1^T(y_1^{k+1}) - A_2^T(y_2^{k+1}) - B^T(v^{k+1}) - \mu X^k \right) \right\|_F^2$ which can be solved by spectral decomposition of the term subtracted from S (Stewart, 1993)
13: Choose any step-length $\mu \geq 0$
14: Update X^{k+1} $= X^k + \frac{A_1^T(y_1^{k+1}) + A_2^T(y_2^{k+1}) + B^T(v^{k+1}) + S^{k+1} - C)}{\mu}$
15: **end while**
16: Return X

An important aspect of implementing the augmented Lagrangian method is problem-specific simplification of linear algebra involved. In relaxations of bounded graph colouring of a graph on n vertices, one can exploit properties of the relaxation to:

- not compute $(A_1 A_1^T)^{-1}$
- compute $A_1^T y_1$ in time $O(m)$
- compute $(A_2 A_2^T)^{-1}$ in time $O(n^2)$
- compute $A_2^T y_2$ in time $O(n)$
- compute $(AB^T)^{-1}$ in time $O(n)$
- compute $B^T v$ in time $O(n)$
- evaluate the augmented Lagrangian and its gradient at a given v in time n^2

This allows for an efficient implementation using a variant of limited memory BGFS seach with projection to non-negative v to minimise the quadratic program on Line 11. The bulk of the run-time is hence spent an eigenvalue decomposition in Line 12. Our particular implementation relies on Intel Math Kernels.

Recovering an Assignment Finally, since the seminal paper of Karger, Motwani, and Sudan (Karger, Motwani, & Sudan, 1998), there has been a continuing interest in algorithms recovering a colouring from semidefinite relaxations. Typically, such algorithms are based on simple randomised iterative rounding of the semidefinite programming relaxation. One such algorithm, specialised to simple timetabling is displayed in Algorithm Schema 2.

Conclusions SDP solvers are less well-developed than LP solvers, in general. Compared to interior point methods, augmented Lagrangian drastically reduce the dependence of per-iteration run-time on the number of constraints. Preliminary computational results suggest this method is practical for instances with up to millions of edges in the conflict graph.

Algorithm Schema 2 `IterativeRounding`(X) based on Karger, Motwani, and Sudan

Input: Matrix variable X of the solution to the SDP (**??**) of dimensions $n \times n$, bound m, number $a_{\max}$ of randomisations to test, plus the input to Simple Timetabling, if required
Output: Partition P of the set $V = 1, 2, \ldots, n$
Compute vector $v, X = v^T v$ using Cholesky decomposition
for Each attempted randomisation $a = 1, \ldots, a_{\max}$ **do**
    Initialise $P_a = \emptyset, i = 1, X = V$
    while There are uncoloured vertices in X **do**
        Pick a suitable $c = \sqrt{\frac{2(k-2)}{k \log_e \Delta}}$ for Δ being the maximum degree of the vertices in X
        Generate a random vector r of dimension $|X|$
        Pick $R_i \subseteq X$ of at most m elements in the descending order of $v_i r_i$, where (1) positive and (2) independent of previously chosen and, in Simple Timetabling, (3) the respective events fit within the rooms and (4) require only features available
        Update $P_a = P_a \cup \{\{R_i\}\}, X = X \setminus R_i$, $i = i + 1$
    end while
end for
Return P_a of minimum cardinality

References

Alizadeh, F. (1995). Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM J. Optim.*, *5*(1), 13–51.

Alizadeh, F., & Goldfarb, D. (2003). Second-order cone programming. *Math. Program.*, *95*(1, Ser. B), 3–51. (ISMP 2000, Part 3 (Atlanta, GA))

Bellman, R., & Fan, K. (1963). On systems of linear inequalities in Hermitian matrix variables. In *Proc. Sympos. Pure Math., Vol. VII* (pp. 1–11). Providence, R.I.: Amer. Math. Soc.

Burke, E. K., Mareček, J., & Parkes, A. J. (2011). Semidefinite programming relaxations in timetabling. Available from `http://cs.nott.ac.uk/~jxm/timetabling/bounding-bounded.pdf`

Goemans, M. X., & Rendl, F. (2000). Combinatorial optimization. In *Handbook of semidefinite programming* (Vol. 27, pp. 343–360). Boston, MA: Kluwer Acad. Publ.

Hestenes, M. R. (1969). Multiplier and gradient methods. *J. Optimization Theory Appl.*, *4*, 303–320.

Karger, D., Motwani, R., & Sudan, M. (1998). Approximate graph coloring by semidefinite programming. *J. ACM*, *45*(2), 246–265.

Povh, J., Rendl, F., & Wiegele, A. (2006). A boundary point method to solve semidefinite programs. *Computing*, *78*(3), 277–286.

Powell, M. J. D. (1969). A method for nonlinear constraints in minimization problems. In *Optimization (Sympos., Univ. Keele, Keele, 1968)* (pp. 283–298). London: Academic Press.

Stewart, G. W. (1993). On the early history of the singular value decomposition. *SIAM Rev.*, *35*(4), 551–566.

Wen, Z., Goldfarb, D., & Yin, W. (2010). Alternating direction augmented lagrangian methods for semidefinite programming. *Math Program Comput*.

Wolkowicz, H., Saigal, R., & Vandenberghe, L. (Eds.). (2000). *Handbook of semidefinite programming*. Boston, MA: Kluwer Academic Publishers. (Theory, algorithms, and applications)

Wright, S. J. (1997). *Primal-dual interior-point methods*. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM).