Proof. We set $\delta = n^{0.613}$ and apply Theorem 11.33 to obtain a subgraph H of G and a colouring of $G - H$ with $\frac{2n}{\delta} = 2n^{0.387}$ colours. We embed H in $\mathcal{R}^n$ using (approximately) unit-length vectors so that the dot product corresponding to every edge is less than $-\frac{1}{2} + \frac{1}{10n}$. We set $j = \lceil \log_3 \delta \rceil + 5$, choose j random hyperplanes through the origin, and assign vertices in the same orthant the same colour. This provides a colouring of all the vertices of H which are not incident to a bad edge.

As discussed earlier, the expected number of bad edges is at most $|E(H)|(\frac{1}{3}+\frac{1}{n})^j < \frac{|E(H)|}{16\delta} \leq \frac{n}{8}$. If we obtain more than $\frac{n}{4}$ bad edges, we make a new independent choice of the hyperplanes. The probability that this happens is at most $\frac{1}{2}$. So we expect to perform this step at most twice. Once we have j hyperplanes for which there are at most $\frac{n}{4}$ bad edges, we have a semi-colouring of G with $2^j + \frac{2n}{\delta} = O(n^{0.387})$ colours, as we simply leave the vertices in bad edges uncoloured. □

This is a randomized algorithm, which can easily be derandomized using the method of conditional expectations.

A better algorithm

We can now discuss the proof of Theorem 11.30. Because of our auxiliary results, we need only show:

Theorem 11.37 *There is a polynomial-time algorithm which semi-colours any 3-colourable graph of maximum degree $n^{\frac{3}{4}}$ using $O(n^{\frac{1}{4}}(\log\ n)^{\frac{1}{2}})$ colours.*

Proof. Once again, we construct an embedding of G in $\mathcal{R}^n$ so that the x_i are unit-length and the dot product corresponding to every edge is not much more than $-\frac{1}{2}$. Now, however, we choose $Cn^{\frac{1}{4}}(\log n)^{\frac{1}{2}}$ random unit vectors $y_1, \ldots, y_k$ for an appropriate constant C and colour x_i with colour j if it is closer to y_j than any other y_l. Kargar, Motwani and Sudan verified that the expected number of bad edges under this procedure is less than $\frac{n}{8}$. So, repeating the process until there are at most $\frac{n}{4}$ such edges, which means on average we perform two or fewer iterations, we will have obtained the desired semi-colouring. □

This randomized algorithm can be derandomized, but the details are tricky; see [26].

11.8 The Theta Body

We now discuss Grötschel, Lovász and Schrijver's algorithm for optimizing over perfect graphs. To do so, we need to consider the stable set polytope, the fractional stable set polytope, and the theta body. We recall that the stable set polytope of G, denoted by $STAB(G)$ and defined in Chapter 2, consists of all those vectors which can be written as convex combinations of the incidence vectors of stable sets. The fractional stable set polytope, denoted by $QSTAB(G)$ and also defined in Chapter 2, consists of all those nonnegative vectors $z = (z_1, \ldots, z_n)$ in $\mathcal{R}^n$ such that for every clique Q of G, $\sum_{v_i \in Q} z_i \leq 1$. The theta body, which we define now, is a convex body which contains $STAB(G)$ and is contained in $QSTAB(G)$.

To define the theta body, $TH(G)$, of an n vertex graph G, we consider the following set of constraints on $n+1$ vectors of dimension $n+1$:

$$x^0 \cdot x^0 = 1, \forall v_i \in V, x^0 \cdot x^i = x^i \cdot x^i, \text{ and } \forall v_i, v_j \in E(G),\ x^i \cdot x^j = 0.$$

Then, $TH(G)$ consists of the vectors $(z_1, \ldots, z_n)$ such that there are $n+1$ vectors satisfying the above constraints for which $x^i \cdot x^i = z_i$. We can maximize over the theta body by maximizing $\sum_{i=1}^{n} w_i(x^i \cdot x^i)$ subject to the above constraints.

We note that for any stable set S, we can obtain a solution to our constraint set showing that the characteristic vector of S is in $TH(G)$ as follows:

(i) Set x^0 to be the vector whose first coordinate is one and whose other coordinates are 0.
(ii) For $v_i \notin S$ set $x^i = 0$.
(iii) For $v_i \in S$ set $x^i = x^0$.

More generally, if z can be expressed as the convex combination of the characteristic vectors of stable sets then it can be so expressed using the characteristic vectors of at most $n+1$ stable sets (since we can restrict ourselves to linearly independent stable sets), say, $z = \sum_{i=1}^{n+1} a_i \mathcal{X}^i$ for characteristic vectors $\mathcal{X}^1, \ldots, \mathcal{X}^{n+1}$ of stable sets $S_1, \ldots, S_{n+1}$. Let x^0 be the vector with $x_i^0 = \sqrt{a_i}$ and for $1 \leq j \leq n$ let x^j be the vector with $x_i^j = \sqrt{a_i}$ if v_j is in S_i and 0 otherwise. It is easy to verify that these vectors certify that z is in $TH(G)$. So, we have shown that $STAB(G)$ is indeed contained in $TH(G)$.

To see that $TH(G)$ is contained in $QSTAB(G)$, we consider a vector $(z_1, \ldots, z_n)$ in $TH(G)$ and the corresponding vectors $x^0, \ldots, x^n$ satisfying the above constraints with $z_i = x^i \cdot x^i$. Note first that letting y^i be the unit vector with the same direction as x^i, choosing r_i so that $x^i = r_i y^i$, and letting l_i be the length of the projection of y^i onto x^0, we have:

$$l_i = (x^0 \cdot y^i) = \frac{1}{r_i}(x^0 \cdot x^i) = \frac{1}{r_i}(x^i \cdot x^i) = r_i.$$

Since $z_i = r_i^2$ we obtain that $z_i = (x^0 \cdot y^i)^2$.

Now for any set Y of unit-length vectors every pair of which forms a 90° angle, and any unit-length vector x^0 in $\mathcal{R}^n$, basic linear algebra yields $\sum_{y \in Y}(x^0 \cdot y)^2 \leq 1$. Applying this for every set Y of vertices corresponding to a clique, we see by the above remarks that $(z_1, \ldots, z_n)$ is in $QSTAB(G)$, as required.

So, we have obtained $STAB(G) \subseteq TH(G) \subseteq QSTAB(G)$. For perfect graphs, as $STAB(G) = QSTAB(G)$, these three bodies are the same. As discussed above, we can solve WOPT on $TH(G)$ for arbitrary graphs by solving the corresponding SDP, which has a polynomial number of constraints (this is one advantage it has over the LP formulation; of course, there are also an infinite number of half-planes enforcing semi-definiteness, but as we have seen we can efficiently test if any of these are violated). As discussed in Section 11.2.2, since $TH(G)$ is the constraint set of an LP when G is perfect, we can use an algorithm which solves WOPT to solve SOPT over $TH(G)$ for such graphs. In particular, for any perfect graph G, we can find a maximum-weight stable set in polynomial time for any weighting of $V(G)$.

We now present an algorithm for colouring perfect graphs with ω colours. The core of the algorithm is a subroutine from [14] which expresses a point of a feasible region R of an LP as a convex combination of vertices of R (we omit its description).

Our approach to colouring perfect graphs is recursive. We simply find a stable set S which meets every maximum clique in the input graph G. Then, $G - S$ is a perfect graph with $\chi(G-S) = \chi(G) - 1$ so a recursive call allows us to $(\omega(G)-1)$-colour it. Adding S as a new colour class yields the desired $\omega(G)$.

To find our desired stable set S, we note that every stable set with nonzero weight in a fractional $\omega(G)$-colouring of G meets every maximum clique K, for we have also fractionally ω-coloured K. So we need only find a fractional $\omega(G)$-colouring of G. But having expressed $(\frac{1}{\omega}, \ldots, \frac{1}{\omega})$ as a convex combination of characteristic vectors of stable sets $\sum_S a_S X_S$, we can obtain the desired fractional colouring by assigning each stable set S the weight ωa_S.

So, our algorithm proceeds as follows:

1. Compute $\omega(G)$ by optimizing $\sum x^i$ over $TH(\overline{G})$.
2. Apply our subroutine with $R = TH(G) = STAB(G)$ to obtain an expression for $(\frac{1}{\omega}, \ldots, \frac{1}{\omega})$ as a convex combination of stable sets of G.
3. Let S be any stable set assigned a nonzero multiplier in this expression.
4. Use S as a colour class and recursively colour $G - S$.

The above discussion shows the algorithm is correct. It is easy to verify that it runs in polynomial time.

Acknowledgements

I would like to thank U. Feige, M. Goemans, M. Laurent, L. Lovász, C. McDiarmid and B. Shepherd for careful readings of the paper and helpful comments. In particular, Monique is largely responsible for the current form of Section 11.8 and Michel provided the proof of 11.28 as well as making many other suggestions with respect to my treatment of SDP theory. I also had useful discussions with J. Cheriyan, M. Todd and Henry Wolkowizc.

References

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

[2] G. Blache, M. Karpinski and J. Wirtgen. On approximation intractability of the bandwidth problem, *Electronic Colloquium on Computational Complexity* **5** (1998).

[3] J. Bourgain. On Lipschitz embedding of finite metric spaces, *Israel J. Math.* **52** (1985), 46–52.

[4] S. Boyd, L. el Ghaoui, E Feron and V. Balkrishnan. *Linear Matrix Inequalities in System and Control Theory*, SIAM, 1994.

[5] P. Chian, J. Chvatalova, A. Dewdney and N. Gibbs. The bandwidth problem for graphs and matrices–a survey, *J. Graph Theory* **6** (1982), 223–254.

[6] S. Cook. The complexity of Theorem Proving Procedures, in *Proceedings of STOC* (1971), 151–158.

[7] C. Delorme and S. Poljak. Laplacian eigenvalues and the maximum cut problem, *Math. Programming* **62** (1993), 557–574.

[8] P. Erdös and Selfridge. On a combinatorial game, *J. Combin. Theory Sec. A* **14** (1973), 298–301.

[9] U. Feige, Approximating the bandwidth via volume respecting embeddings, in *the Proceedings of STOC* (1998), 90–99.

[10] M. Goemans. Semidefinite programming in combinatorial optimization, *Math. Programming* **79**(1997), 143–161.

[11] M. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, *J. ACM* **42** (1995), 1115–1145.

[12] G. Golub and C. Van Loan. *Matrix Computations* John Hopkins University Press, Baltimore, MD (1989).