Journal of Computational Mathematics, Vol. 24, No. 6, 2006, 749–760.

AN EFFECTIVE CONTINUOUS ALGORITHM FOR APPROXIMATE SOLUTIONS OF LARGE SCALE MAX-CUT PROBLEMS *1)

Cheng-xian Xu, Xiao-liang He and Feng-min Xu
(*Department of Mathematics, Xi'an Jiaotong University, Xi'an 710049, China*)

Abstract

An effective continuous algorithm is proposed to find approximate solutions of NP-hard max-cut problems. The algorithm relaxes the max-cut problem into a continuous nonlinear programming problem by replacing n discrete constraints in the original problem with one single continuous constraint. A feasible direction method is designed to solve the resulting nonlinear programming problem. The method employs only the gradient evaluations of the objective function, and no any matrix calculations and no line searches are required. This greatly reduces the calculation cost of the method, and is suitable for the solution of large size max-cut problems. The convergence properties of the proposed method to KKT points of the nonlinear programming are analyzed. If the solution obtained by the proposed method is a global solution of the nonlinear programming problem, the solution will provide an upper bound on the max-cut value. Then an approximate solution to the max-cut problem is generated from the solution of the nonlinear programming and provides a lower bound on the max-cut value. Numerical experiments and comparisons on some max-cut test problems (small and large size) show that the proposed algorithm is efficient to get the exact solutions for all small test problems and well satisfied solutions for most of the large size test problems with less calculation costs.

Mathematics subject classification: 90C27.
Key words: Max-cut problems, Algorithm, Feasible direction method, Laplacian matrix, Eigenvectors.

1. Introduction

The max-cut problem is to partition the vertex set of an undirected graph, denoted by $G(V,E)$, into two parts in order to maximize the sum of the weights on the edges between these two parts, where V with $|V|=n$ is the set of n vertices and E the edge set of the graph. This problem has long been known to be NP-hard, and it is solvable in polynomial time only for some special classes of graphs [10]. Because of its theoretical and practical importance, and because efficient algorithms for NP-hard combinatorial optimization problems are unlikely to exist, many approximate algorithms (see [11],[15],[21],[23]) have been proposed to solve max-cut problems at an approximation factor ρ, that is, to find a *cut* $(S,\bar{S})$ such that $w(S,\bar{S}) \geq \rho w^*$, where S and $\bar{S}=V\backslash S$ denote the cut, $w(S,\bar{S})$ is the value of the *cut* $(S,\bar{S})$, w^* is the max-cut value, and ρ is generally called the performance guarantee of an algorithm. Among these approximate algorithms, the most famous is the randomization algorithm with performance guarantee $\rho = 0.87856$ proposed by Goemans and Williamson [9]. The algorithm relaxes each binary variable in $\{-1,1\}$ to a unit vector in space R^n to form a semi-definite programming

* Received November 30, 2005.
1) This work is supported by National Natural Science Foundation of China at 10231060.

problem, hence increasing the problem dimension from n to $n \times n$, and the resulting SDP problem is then solved using any existing semi-definite programming algorithms, for example, interior algorithms. Then an approximate solution to the max-cut problem is generated from the optimal solution of the relaxed SDP problem using a randomization algorithm. Although extremely interesting because Goemans and Williamson's algorithm has the best worst case performance guarantee, it is of complex design and its computation time may prohibitive from large scale max-cut problems [7]. For solving large scale max-cut problems, some nonlinear programming methods are proposed in [6],[7],[12],[18]. The strengthened semi-definite programming relaxation [4] and the rank two relaxation [7] of max-cut problems are modifications of Goemans and Williamson's work. The algorithm in [22] generates an approximate solution to the max-cut problem by minimizing the largest eigenvalue of the matrix that is the sum of the Laplacian matrix of the graph and a variable diagonal matrix. Since the algorithm calculates the largest eigenvalues of a sequence of given matrices satisfying the constraints and the objective function in minimization is not differentiable everywhere, it is of complex design and not applicable for the solution of large scale max-cut problems.

In this paper, we present an effective continuous algorithm for approximate solutions of large scale max-cut problems. The algorithm relaxes the max-cut problem into a continuous nonlinear programming problem that finds the largest eigenvalue of the Laplacian matrix of the underlying graph by maximizing a convex quadratic function subject to a single constraint. The constraint restricts the length of the variable vectors. An efficient feasible direction method is used to perform the maximization of the resulting nonlinear programming problem. The method only employs the gradient evaluations of the objective function and no any matrix calculations and no line searches are required. This greatly reduces the calculation cost in the implementation of the algorithm and increases the efficiency, and makes the algorithm applicable to large scale max-cut problems. The convergence of the feasible direction method to KKT points of the nonlinear programming is proved. If the solution obtained by the feasible direction method is a global solution of the resulting nonlinear programming, the solution provides an upper bound on the optimal value of the max-cut. A feasible solution to the max-cut problem can then be generated from the solution of the nonlinear programming, and provides a lower bound for the max-cut value. Numerical experiments and comparisons on some well-known max-cut test problems (small size) and on some large size problems that are randomly generated by the procedure **rudy** are made to show the efficiency of the proposed method on both the computation time and resulting solutions.

Let $w_{ij} = w_{ji}$ be the weight on edge $e_{ij} \in E$ of a graph $G(V, E)$, where $w_{ij} = 0$ if there is no edge connecting vertices V_i and V_j. Using the Laplacian matrix of the graph $L = \frac{1}{4}(\mathrm{Diag}(We) - W) = (L_{ij})_{n\times n}$ with weight matrix $W = (w_{ij})_{n\times n}$, the max-cut problem can be expressed as

$$(\mathrm{MC}) : \begin{cases} \mathrm{Max} \ \ x^T L x \\ s.t. \ \ x_i^2 = 1, \quad i = 1, \cdots, n, \end{cases}$$

where

$$L_{ij} = \begin{cases} -w_{ij}, & i \neq j, \\ \sum_{k\neq i}^{n} w_{ik}, & i = j, \end{cases}$$

The Laplacian matrix L is positive semi-definite. The constraints in (MC) restrict each variable taking values either 1 or -1, and hence it is a combinatorial optimization problem. Goemans and Williamson in [9] relaxe the problem to formulate a semi-definite programming problem by