3.4 Progress in implementation and application

3.4.1 Combinatorial and global optimization

As we mentioned before, although they are "polynomially" solvable, semidefinite programs with dimension above 1000 have been extremely hard to solve in practice. Quite recently, Benson, I and Zhang [W1][W4], based on [J15], have developed two *c*-programs to solve semidefinite relaxations for max-cut and other combinatorial optimization (available on *http : //dollar.biz.uiowa.edu/col/*) with dimension up-to 9000. To the best of our knowledge, our result is the first computational evidence of the effectiveness of these approximations in solving large-scale problems. More precisely, we applied the program to solving the well-known $G - set$ problems, see Helmberg and Rendl [10]. These problems were generated by **rudy**, a machine independent graph generator written by G. Rinaldi. This set of problems became a standard test set for graph optimization, and they had never been solved before by interior-point algorithms.

Our program was run on the HP model C240 computer. Solving the semidefinite relaxation stopped when its relative duality gap dropped below 10^{-3}. The following table show the performance of the code for solving them as maximum-cut. Similar results obtained for solving them as maximum graph-partition and maximum box-QP.

The first column, *Name*, of the tables is the problem index. Then, *Dim* is the number of nodes in the graph, and *Spar* is the density ratio of the graph edges. *SDPObj* is the optimal value of the SDP relaxation, which should be an upper bound for the maximal value of the original problem. The fifth column, $Vobj$, lists the objective value of the approximate solution resulted from a randomization procedure, that is, a feasible objective value for the original problem. The sixth column, *Ratio*, depicts the ratio: $Vobj/SDPobj$. If this ratio equals 1, then the approximate solution is actually 100% optimal. Most of problems have ratios above 0.9, indicating that their solutions are at least 90% optimal. Some of them are below 0.9, since these graphs have negative edge weights for which the theory can guarantee only 60% optimality.

The next three columns represent run-time statistics: the number of iterations and the CPU time (in seconds) used in solving the SDP relaxation; total computational time, including the randomization, in solving each of this set of problems. All problems are being solved in a reasonable computational time; and the sparser of the problem, the faster of the solution time.

3.4.2 Nonlinear optimization

On linear and nonlinear programming, Andersen and I [J11] have developed a generalization of the recent Ye-Todd-Mizuno homogeneous and self-dual linear programming (LP) algorithm. The generalized algorithm solves the linear and nonlinear convex programming problem without any regularity assumption concerning the existence of optimal, feasible, or interior feasible solutions. It also detects infeasibility based on a provable criterion. (It was a major problematical issue to detect infeasibility for practical interior-point methods. Some theoretical results hold only for feasible cases. Other approaches to detect infeasibility are somewhat difficult to implement in practice.)

Kortanek, Xu and I [J2] present theoretical and computational results for solving the