Solving large scale Max Cut problems via tabu search

Article *in* Journal of Heuristics · August 2013

DOI: 10.1007/s10732-011-9189-8

CITATIONS
26

READS
193

5 authors, including:

Gary Kochenberger
University of Colorado
122 PUBLICATIONS **3,059** CITATIONS
SEE PROFILE

Jin-Kao Hao
University of Angers
350 PUBLICATIONS **5,880** CITATIONS
SEE PROFILE

Zhipeng Lü
Huazhong University of Science and Technology
59 PUBLICATIONS **1,043** CITATIONS
SEE PROFILE

Haibo Wang
Texas A&M International University
448 PUBLICATIONS **6,262** CITATIONS
SEE PROFILE

J Heuristics
DOI 10.1007/s10732-011-9189-8

Solving large scale Max Cut problems via tabu search

Gary A. Kochenberger · Jin-Kao Hao · Zhipeng Lü · Haibo Wang · Fred Glover

Received: 31 August 2010 / Accepted: 1 July 2011

Abstract In recent years many algorithms have been proposed in the literature for solving the Max-Cut problem. In this paper we report on the application of a new Tabu Search algorithm to large scale Max-cut test problems. Our method provides best known solutions for many well-known test problems of size up to 10,000 variables, although it is designed for the general unconstrained quadratic binary program (UBQP), and is not specialized in any way for the Max-Cut problem.

Keywords Max Cut problem · Metaheuristics · Combinatorial optimization

1 Introduction

Given an undirected graph $G(V, E)$ with edge weights w_{ij}, the NP-hard Max-Cut (MC) problem seeks a partition $S_1 \subset V$ and $S_2 = V - S_1$ such that the weight of the cut, defined as the sum of the weights on the edges connecting the two sets, is maximized. This problem has long served as a challenging test for researchers testing new

G.A. Kochenberger
University of Colorado, Denver, CO, USA
e-mail: gary.kochenberger@ucdenver.edu

J.-K. Hao · Z. Lü
LERIA, Universite d'Angers, 49045, Angers Cedex 01, France

J.-K. Hao
e-mail: hao@info.iniv-angers.fr

H. Wang
Texas A&M International University, Laredo, TX, USA
e-mail: hwang@tamiu.edu

F. Glover (✉)
OptTek Systems, Boulder, CO 80302, USA
e-mail: glover@opttek.com

Published online: 14 September 2011

methods for combinatorial algorithms. The problem also has well known practical applications in several areas including statistical physics, VLSI design, classification, and social network analysis. Other applications, as discussed by Boros and Hammer (1991), include fault test generation, multiprocessor assignments in distributed networks, image enhancement, and maximum likelihood rankings in statistics. Additional applications are discussed in the survey paper by Poljak and Tuza (1995).

The computational challenge of the Max-Cut problem has motivated a variety of solution approaches ranging from approximation algorithms based on semidefinite programming, to metaheuristic methods, to exact methods. The approximation algorithms, represented by the groundbreaking work of Goemans and Williamson (1995) and the later work by Karish et al. (2000) provide a performance guarantee but are generally outperformed by other methods in computational testing. Recent reports on exact methods include the cut and price approach by Krishnan and Mitchell (2006) and the branch and bound approach of Rendl et al. (2008). While these approaches represent advances over the approximation algorithms in finding optimal solutions to Max-Cut instances, their applications have proven to be limited to problems with no more than a few hundred nodes.

For larger MC instances, those with thousands of nodes, metaheuristic methods are required. In this regard, several advances have been recently reported in the literature, the most notable of which are the rank-2 relaxation method, called CirCut, of Burer et al. (2001), the hybrid randomized method of Festa et al. (2002), and the scatter search method of Marti et al. (2009). Computational testing indicates that these methods generally outperform other methods in the literature with the scatter search method providing the best overall performance on test problems with up to 3000 nodes.

2 Unconstrained binary quadratic programming and the Max-Cut problem

Much of the published research on the Max-Cut problem present algorithms specially crafted for the problem at hand. Our approach taken here is quite different. In this paper we report on the application of a new tabu search method for the Unconstrained Binary Quadratic Program (UBQP) to the Max-Cut problem. Our approach is not specialized in any way for the Max-Cut problem but instead is designed for the general UBQP.

It is well known (see for example Boros and Hammer 1991; Helmberg and Rendl 1996) that the Max-Cut problem can be modeled as an integer program of the form:

$$\max \frac{1}{2} \sum_{1 \le i < j \le n} w_{ij}(1 - y_i y_j)$$

$$\text{subject to:} \quad y_i \in \{-1, 1\} \quad \forall i \in V$$

To solve this problem, we first introduce a change of variables ($y_i = 2x_i - 1$) leading to the unconstrained quadratic program in binary variables

$$\max \sum_{i<j} w_{ij}(x_i + x_j - 2x_i x_j); \quad x_j \in \{0, 1\}$$

This problem is of the form max $x'Qx$ and thus is solvable by the recently developed Diversification Driven Tabu Search (DDTS) method by Glover et al. (2010) for the general UBQP problem.

The DDTS method repeatedly alternates between a simple version of tabu search (TS) and a diversification phase founded on a memory-based perturbation operator. Starting from an initial random solution, DDTS uses the TS procedure to reach a local optimum. Then, the perturbation operator is applied to displace the solution to a new region, whereupon a new round of TS is launched. To facilitate achieving effective diversification, the perturbation operator is guided by information from a special memory structure.

This tabu search procedure uses a neighborhood defined by the single 1-flip moves, which consists of changing (flipping) the value of a single variable x_j to its complement value $1 - x_j$. The implementation of this neighborhood uses a fast incremental evaluation technique to calculate the cost of candidate moves. The diversification strategy utilizes a memory- based perturbation operator composed of three parts: a flip frequency memory, an elite solution memory, and an elite value frequency memory. These memory structures are used jointly by the perturbation operator to enhance the diversification of the search process. Complete details of this method are given in Glover et al. (2010).

3 Computational results

Due to the fact that the leading exact methods can only handle problems containing up to a few hundred binary variables, we restrict our comparisons to the best the literature has to offer coming from metaheuristic methods, which have demonstrated an ability to handle problems an order of magnitude larger.

Our computational testing was carried out on the "*G*" problems available at http://www.stanford.edu/~yyye/yyye/Gset/. This set of test problem of medium to large size is widely adopted in the literature to facilitate computational testing and comparisons. In all, we considered 69 problems ranging in size from n = 800 variables to n = 10,000 variables.

The recent paper of Marti et al. (2009) describes the use of a special purpose Scatter Search (SS) method for the Max-Cut problem and provides comparative results with widely regarded solution procedures such as GRASP Festa et al. (2002) and CirCut (Burer et al. 2001) on some of these same "G" problems. Their scatter search method, which we refer to by MDL, proved to outperform competing metaheuristic methods on most test problems considered and thus we use the MDL results as our main benchmark for making comparisons with our method. Our testing however, went beyond comparisons with MDL in terms of problem size, as results for the MDL method were reported only for problems up to $n = 3000$. To provide a benchmark for comparison for larger problems (up to $n = 10{,}000$ variables) we include published results from Choi and Ye (2000), which is the only published account we can find that reports on these larger instances.

Results from our testing are shown in Table 1 where the column titled CY refers to results published by Choi and Ye, the column titled MDL gives the results published

Table 1 Computational results

Problem ID	# Vars	Best known OBJ	SDP bound	CY OBJ	MDL OBJ	KHLWG OBJ
G1	800	11624	12078	–	11624	11624
G2	800	11620	12084	–	11620	11620
G3	800	11622	12077	–	11622	11620
G4	800	11646	–	–	11646	11646
G5	800	11631	–	–	11631	11631
G6	800	2178	–	–	2165	2178
G7	800	2003	–	–	1982	2006
G8	800	2003	–	–	1986	2005
G9	800	2048	–	–	2040	2054
G10	800	1994	–	–	1993	2000
G11	800	564	627	542	562	564
G12	800	556	621	540	552	556
G13	800	580	645	564	578	580
G14	800	3060	3187	2982	3060	3061
G15	800	3049	3169	2975	3049	3050
G16	800	3045	3172	–	3045	3052
G17	800	3043	–	–	3043	3046
G18	800	988	–	–	988	991
G19	800	903	–	–	903	904
G20	800	941	–	876	941	941
G21	800	931	–	855	930	931
G22	2000	13346	14123	12989	13346	13359
G23	2000	13317	14129	13006	13317	13342
G24	2000	13314	14131	12985	13303	13337
G25	2000	13326	–	–	13320	13332
G26	2000	13314	–	–	13294	13328
G27	2000	3318	–	–	3318	3336
G28	2000	3285	–	–	3285	3295
G29	2000	3389	–	–	3389	3391
G30	2000	3403	–	3080	3403	3403
G31	2000	3288	–	2936	3288	3288
G32	2000	1398	1560	1338	1398	1406
G33	2000	1376	1537	1330	1362	1378
G34	2000	1372	1541	1334	1364	1378
G35	2000	7670	8000	–	7668	7678
G36	2000	7660	7996	–	7660	7670
G37	2000	7666	8009	–	7664	7682
G38	2000	7681	–	–	7681	7683
G39	2000	2395	–	–	2393	2397
G40	2000	2387	–	–	2374	2390

Table 1 (*Continued*)

Problem ID	# Vars	Best known OBJ	SDP bound	CY OBJ	MDL OBJ	KHLWG OBJ
G41	2000	2398	–	–	2386	2400
G42	2000	2469	–	–	2457	2469
G43	1000	6659	7027	–	6656	6660
G44	1000	6648	7022	–	6648	6639
G45	1000	6652	7020	–	6642	6652
G46	1000	6645	–	–	6634	6649
G47	1000	6656	–	–	6649	6665
G48	3000	6000	–	6000	6000	6000
G49	3000	6000	–	6000	6000	6000
G50	3000	5880	–	5880	5880	5880
G51	1000	3846	–	–	3846	3847
G52	1000	3849	–	–	3849	3849
G53	1000	3846	–	–	3846	3848
G54	1000	3846	–	–	3846	3851
G55	5000	9960	–	9960	–	10236
G56	5000	3649	–	3649	–	3934
G57	5000	3220	–	3220	–	3460
G58	5000	–	–	–	–	19248
G59	5000	–	–	–	–	6019
G60	7000	13658	–	13658	–	14057
G61	7000	5273	–	5273	–	5680
G62	7000	4612	–	4612	–	4822
G63	7000	8059	–	8059	–	26963
G64	7000	7861	–	7861	–	8610
G65	8000	13286	–	13286	–	5518
G66	9000	–	–	–	–	6304
G67	10000	–	–	–	–	6894
G70	10000	9499	–	9499	–	9458
G72	10000	6644	–	6644	–	6922

by Marti, Duarte and Laguna, and the right most column, titled KHLWG, gives our results. The Semi-Definite relaxation results have been reported in the literature for some of these problems and we list these in the forth column of the table to provide yet another benchmark for comparison. The third column in Table 1 gives the best known solutions reported in the literature prior to this paper. In Table 1 we report results from our method for each of the 69 problems under consideration. A "dash" in the table denotes that a result for that particular problem was not available. For example, MDL does not report results for problems G55 to G72.

To make an allowance for the different computers used by MDL and our DDTS algorithm, we used the standard SPEC benchmark data. The results reported by MDL

were obtained by letting the method run for 1/2 hour of CPU time on each problem. The comparable time on our computer, according to the SPEC 2000 and SPEC 2006 benchmarks is 2.36 hours of CPU time. Accordingly, the results we present in the table for problems G1-G54 refer to the best solution found by our method in 2.36 hours of CPU time. For the larger problems not addressed by MDL, we increased the CPU run time to allow our algorithm more time to search for good solutions as the problem size increased. Accordingly, for the $n = 5000$ and 7000 variable problems, we report the best solutions found in 8 hours of CPU run time. For the 8,000, 9,000, and 10,000 variable problems, the CPU run time limits were 12 hours, 20 hours, and 24 hours respectively.

Table 1 enables a comparison of the DDTS metaheuristic with the scatter search metaheuristic of MDL on problems G1–G54. In a comparable time limit, our approach found better solutions on 40 of the 54 problems. On 12 problems both methods found the same solution and on 2 of the problems MDL found a better solution.

Over the entire test bed of 69 problems, Table 1 shows that our method matched best known solutions on 19 of the 69 problems, found new best known solutions on 46 problems, and failed to find best known solutions on 4 problems. On those problems for which a SDP bound is available, our objective function values are very close to the bounds, adding further evidence of the quality of our results.

4 Summary & conclusion

We demonstrate that a modern tabu search metaheuristic designed for the general unconstrained binary quadratic program can produce high quality solutions to large Max-Cut problems. Comparisons with other approaches reported in the literature show that our method outperformed the leading competitive methods by a wide margin and in fact found new best known solutions for most problems attempted. This is particularly noteworthy in that the solution approach we employed is not specialized in any way for the Max-Cut problem, but handles problems of a much larger class. Our results suggest that further enhancements to solution methodologies for the unconstrained binary quadratic program hold considerable promise for solving even larger Max-Cut problems. We plan to report on such work in future papers.

References

Boros, E., Hammer, P.: The Max-Cut problem and quadratic 0-1 optimization; polyhedral aspects, relaxations and bounds. Ann. Oper. Res. **33**, 151–180 (1991)

Burer, S., Monteiro, D., Zang, Y.: Rank-two relaxation heuristic for Max-Cut and other binary quadratic programs. SIAM J. Optim. **12**, 503–521 (2001)

Choi, C., Ye, Y.: Solving sparse semidefinite programs using the dual scaling algorithm with an iterative solver. Working Paper, Department of Management Sciences, The University of Iowa (2000)

Festa, P., Pardalos, P., Resende, M., Ribeiro, C.: Randomized heuristics for the Max-Cut problem. Optim. Methods Softw. **7**, 1033–1058 (2002)

Glover, F., Lü, Z., Hao, J.K.: Diversification-driven tabu search for unconstrained binary quadratic problems. 4OR, Q. J. Oper. Res. (2010). doi:10.1007/s10288-009-0115-y

Goemans, M., Williamson, D.: Improved approximation algorithms for Max-Cut and satisfiability problems using Semidefinite programming. J. ACM **42**, 1115–1145 (1995)

Helmberg, C., Rendl, F.: Solving quadratic (0-1) problems by semidefinite programs and cutting planes. Working Paper, Technische Universitat Graz (1996)

Karish, S., Rendl, F., Clausen, J.: Solving graph bisection problems with semidefinite programming. SIAM J. Comput. **12**, 177–191 (2000)

Krishnan, K., Mitchell, J.: A semidefinite programming based polyhedral cut and price approach for the Max-Cut problem. Comput. Optim. Appl. **33**, 51–71 (2006)

Marti, R., Duarte, A., Laguna, M.: Advanced scatter search for the Max-Cut problem. INFORMS J. Comput. **21**, 26–38 (2009)

Rendl, F., Rinaldi, G., Wiegele, A.: Solving Max-Cut to optimality by intersecting semidefinite and polyhedral relaxations. Math. Program. **121**, 307–335 (2008)

Poljak, S., Tuza, Z.: The Max-Cut problem: a survey. In: Cook, W., Lovasz, L., Seymour, P. (eds.) Special Year on Combinatorial Optimization. DIMACS Series in Discrete Mathematics and Computer Science. Am. Math. Soc., Providence (1995)