

An effective modeling and solution approach for the generalized independent set problem

Gary Kochenberger · Bahram Alidaee ·
Fred Glover · Haibo Wang

Revised: 18 April 2006 / Accepted: 24 April 2006
© Springer-Verlag 2006

Abstract The generalized independent set (GIS) problem was first introduced by Hochbaum and Pathria (Forest Sci **43**(4), 544–554, 1997) and independently explored in greater detail by Hochbaum (Manage Sci **50**(6), 709–123, 2004). This problem, with applications in forest management and a variety of related areas, is a generalization of the classical maximum independent set problem. In this paper we highlight a natural, nonlinear formulation for the problem that is an attractive alternative to the linear model found in the literature. The effectiveness of this alternative formulation is demonstrated by computational experience on test problems of varying size and density, disclosing a dramatic reduction in the time to obtain optimal and near optimal solutions and an ability to solve much larger problems.

G. Kochenberger (✉)
School of Business, University of Colorado at Denver, Denver, CO, USA
e-mail: Gary.Kochenberger@Cudenver.edu

B. Alidaee
School of Business, University of Mississippi, University, MS 38677-1848, USA
e-mail: Balidaee@bus.olemiss.edu

F. Glover
University of Colorado, Boulder, CO 80302 USA
e-mail: Fred.Glover@Colorado.edu

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

1 Introduction

The generalized independent set (GIS) problem extends the standard independent set problem by allowing adjacent vertices to be in the set chosen provided an edge-based penalty is incurred. It is useful to think of the standard independent set problem as a special case of GIS with infinite edge weights. First introduced in the context of forest management applications, the GIS problem provides a model that embraces many other combinatorial problems, including set partitioning, coloring, graph partitioning and discrete tomography problems. This generality identifies the GIS problem as one whose effective solution can be of interest in many applications.

The GIS problem can be formally defined relative to a graph $G = (V, E)$ where V is the set of n vertices and E is the set of edges. A weight (profit) $w_i > 0$ is associated with each vertex $i \in V$ and a weight (cost) $c_{ij} > 0$ is associated with each edge $(i, j) \in E$. The problem is to find a subset of the vertices $S \subseteq V$ that maximizes the difference between the sum of the weights of the vertices in S and the sum of the costs of those edges that have both endpoints in S .

The optimization model for GIS that appears in the literature is a large linear 0/1 program containing both node and edge variables and a constraint for every edge in the graph (see Sect. 2 below). Since even modest sized graphs of modest density have many edges, this linear model is typically large, having many variables and constraints. As a result, the linear 0/1 model can be challenging to solve, even for a state-of-the-art integer programming method such as embodied in the CPLEX software.

Our alternative approach is motivated by the fact that in recent years several papers have reported on the attractiveness of nonlinear alternatives to linear models for certain combinatorial optimization problems (see, for instance, [1, 6–10]). In some cases equivalent unconstrained nonlinear models are constructed by absorbing constraints into the objective function and employing infeasibility penalties instead of imposing the constraints explicitly. In other cases the nonlinear alternative emerges simply by adopting a nonlinear perspective that leads to a direct reformulation of the model. Such is the case for the problem we consider here. These direct nonlinear reformulations are typically much smaller than their linear counterparts in terms of the number of variables and/or the number of constraints. Often the nonlinear alternatives have no constraints other than binary restrictions.

Advances in metaheuristic methodologies for solving difficult combinatorial problems often enable such nonlinear alternatives to be efficiently solved to optimality or near optimality. Though no guarantee exists that optimality can be achieved by metaheuristic methods within a finite time period, the finiteness guarantee associated with exact methods permits “finite” to be astronomically large, so that in practical terms a metaheuristic that obtains good solutions quickly can be greatly preferable to an exact method that is unable to find comparable solutions when given a much longer time to run. In this paper we highlight a natural, nonlinear model for GIS as an alternative to the linear

model given in the literature, and demonstrate, that an effective metaheuristic approach exists for the alternative model that makes it highly attractive by comparison with the classical linear one.

2 Modeling representations for GIS

2.1 The linear model [4]

For each vertex $i \in V$, let x_i denote a binary variable that takes the value 1 if and only if vertex i is chosen to be in the subset S . Similarly, let z_{ij} denote a binary variable that takes the value 1 if and only if edge $(i, j) \in E$ has both endpoints in the set S . The linear model for GIS can then be stated as:

$$\text{GIS_Linear : } \max \sum_{i \in V} w_i x_i - \sum_{(i,j) \in E} c_{ij} z_{ij}$$

subject to

$$\begin{aligned} x_i + x_j - z_{ij} &\leq 1 && \forall (i, j) \in E, \\ x_i, z_{ij} &\in \{0, 1\} && \forall i, j \in V. \end{aligned} \tag{1}$$

Except for small instances of GIS, this is a large, 0/1 linear program containing many variables and constraints.

2.2 Nonlinear alternative

The preceding linear 0/1 model can be viewed as the linearization of an equivalent nonlinear model, which results by noting that the constraints shown in (1) imply that when both x_i and x_j are equal to 1, z_{ij} must be 1 as well. Otherwise the z_{ij} variables will be equal to 0 since the c_{ij} coefficients are positive. Thus, the z_{ij} variables can be replaced by the product $x_i x_j$ and the equivalent nonlinear model is:

$$\text{GIS_Nonlinear : } \max \sum_{i \in V} w_i x_i - \sum_{(i,j) \in E} c_{ij} x_i x_j$$

$$x_i \in \{0, 1\}$$

Remarks

- GIS_Nonlinear is an unconstrained binary quadratic program of the form

$$\text{UBQP : } \max_x x^t Q x$$

x binary

where Q is an n -by- n symmetric matrix.

- Except for the binary restrictions, this model is completely unconstrained. Moreover, this model contains the original *node* variables only.

- GIS_Nonlinear is much smaller than its linear counterpart in terms of both the number of variables and the number of constraints.

2.3 Solving GIS_Nonlinear:

Several methods have been proposed recently in the literature for solving UBQP. A brief summary of these methods appears in Kochenberger et al. [6]. In principle, any of these approaches could be applied to GIS_Nonlinear. For small instances, the exact method of Pardalos and Rodgers [11] could be employed. Our experience with this method is that it works well on small problems with up to 50 nodes (or so). Beyond that its performance degrades rapidly. For larger instances, corresponding to graphs with hundreds of nodes and many thousands of edges, metaheuristic methods are required. For the computational experience reported here, we used a basic adaptive memory, tabu search (TS) approach described in Glover et al. [2,3]. A brief overview of the method is given next.

2.4 Overview of TS solution method for UBQP

Our TS method for UBQP is centered around the use of strategic oscillation, which constitutes one of the primary strategies of tabu search. The variant of strategic oscillation we employ may be sketched in overview as follows. The method alternates between constructive phases that progressively set variables to 1 (whose steps we call “add moves”) and destructive phases that progressively set variables to 0 (whose steps we call “drops moves”). To control the underlying search process, we use a memory structure that is updated at *critical events*, identified by conditions that generate a subclass of locally optimal solutions. Solutions corresponding to critical events are called *critical solutions*.

A parameter *span* is used to indicate the amplitude of oscillation about a critical event. We begin with *span* equal to 1 and gradually increase it to some limiting value. For each value of *span*, a series of alternating constructive and destructive phases is executed before progressing to the next value. At the limiting point, *span* is gradually decreased, allowing again for a series of alternating constructive and destructive phases. When *span* reaches a value of 1, a *complete span cycle* has been executed and the next cycle is launched. The search process is typically allowed to run for a pre-set number of span cycles.

Information stored at critical events is used to influence the search process by penalizing potentially attractive add moves (during a constructive phase) and inducing drop moves (during a destructive phase) associated with assignments of values to variables in recent critical solutions. Cumulative critical event information is used to introduce a long term bias into the search process by means of additional penalties and inducements similar to those discussed above. A complete description of the framework for the method is given in Glover et al. [3].

Table 1 Results of GIS test problems (CPLEX 8.1)

Test ID	Size	Density	CPLEX Results for the linear model					TS results for nonlinear model		
			No. of vars	No. of con	OFV	Time (s)	No. of nodes	No. of vars	OFV	Time (s)
GIS50A	50	0.20082	296	246	954	0.11	29	50	954	1
GIS50B	50	0.40816	550	500	604	1.41	170	50	604	1
GIS50C	50	0.60163	787	737	428	2.82	374	50	428	1
GIS50D	50	0.80327	1,034	984	370	3.54	306	50	370	1
GIS100A	100	0.19212	1,051	951	1,394	80.74	14,371	100	1,394	1
GIS100B	100	0.40061	2,083	1,983	802	155.08	14,350	100	802	1
GIS100C	100	0.59677	3,054	2,954	686	89.74	4,170	100	686	1
GIS100D	100	0.80566	4,088	3,988	410	91.15	3,081	100	410	1
GIS125A	125	0.19368	1,626	1,501	1,526	666.22	80,819	125	1,526	2
GIS125B	125	0.40219	3,242	3,117	926	807.32	52,252	125	926	2
GIS125C	125	0.59523	4,738	4,613	628	347.90	12,749	125	628	2
GIS125D	125	0.80142	6,336	6,211	500	370.91	8,321	125	500	2
GIS175A	175	0.19580	3,156	2,981	1,786	32,952	916,840	175	1,786	8
GIS175B	175	0.40151	6,288	6,113	1,022	11,210	389,614	175	1,022	8
GIS175C	175	0.59783	9,277	9,102	652	4,193	86,396	175	652	8
GIS175D	175	0.80302	12,401	12,226	502	2,286	30,013	175	502	8
GIS200A	200	0.19553	4,091	3,891	1,602 ^a	92,268	2,117,072	200	1,720	22
GIS200B	200	0.39955	8,151	7,951	1,044	66,121	1,649,765	200	1,044	22
GIS200C	200	0.60121	12,164	11,964	712	12,603	198,529	200	712	22
GIS200D	200	0.80387	16,197	15,997	512	4,988	50,586	200	512	22
GIS400A	400	0.20102	16,441	16,041	1,912 ^a	138,568	818,414	400	2,270	50
GIS400B	400	0.39940	32,272	31,872	1,042 ^a	208,874	534,217	400	1,258	50
GIS400C	400	0.59984	48,267	47,867	738 ^a	157,215	249,850	400	824	50
GIS400D	400	0.80103	64,322	63,922	464 ^a	149,080	287,113	400	598	50

^aCPLEX unable to find the optimal solution due to the limit of memory on the system

As shown in the next section, this method, which was designed for the generic UBQP problem, works very well on GIS_Nonlinear, in spite of not being specialized or tuned to handle this special subclass of the UBQP problem domain.

3 Computational experience

To illustrate the attractiveness of the nonlinear model and its solution via our TS method, we generated a set of 24 test problems of size 50–400 nodes with densities ranging from roughly 20 to 80%. Node and edge weights were randomly generated in range (2–98).

To present a benchmark for comparison, the linear 0/1 version of each test problem was submitted to the CPLEX software. Table 1 gives the problem characteristics and the results obtained from the linear model (via CPLEX 8.1) and the nonlinear model (via our TS method). All computations were carried out on a Sun Enterprise 450 Server with 1 GB of memory, 16 GB disk space, and a 167 MHz UltraSPARC processor.

The first three columns in Table 1 give the problem ID, number of nodes in the graph and the edge density. The last three columns in the table give the

results obtained from the nonlinear model via our TS solution method. The columns in the middle pertain to the results produced from the linear model via CPLEX.

The linear model for each problem was run on CPLEX 8.1 until CPLEX either reported an optimal solution or terminated due to an “out of memory” error. In those cases where CPLEX was not able to terminate naturally, we report the best objective function value (OFV) it found during the run. CPLEX was able to solve 19 of the 24 problems. As expected the run times ramped up substantially with problem size. CPLEX was not able to terminate naturally for one of the 200 node problems (GIS200A) nor for any of the 400 node problems. These larger problems had run times approaching 60 h of computer time.

The nonlinear model for each problem was run for an arbitrary limit of 500 SPAN cycles of our TS method for the general UBQP problem. The times reported are those required to complete the 500 cycles and the OFV given are the best values found during the search process. For the 19 problems that CPLEX was able to solve to completion, our TS method, operating on the nonlinear model, produced the same (optimal) solution values but in a fraction of the time required by CPLEX applied to the linear counterpart. (The four small 50 node problems were trivial for CPLEX and serve as an exception to this.) For the five large problems that CPLEX was not able to solve to completion, the nonlinear alternative approach gave significantly better OFV than CPLEX. Moreover, our approach required less than a minute of computer time while CPLEX took 30–58 h per problem instance. (Effectively, when an exact method such as CPLEX fails to achieve optimality within a particular time period, it may be conceived as operating as a heuristic, since the solution effort is not being devoted to the task of verifying the optimality of an optimal solution already found.) It is important to point out that the size of the nonlinear model to be solved, i.e., the UBQP instance, is independent of the number of edges in the graph, depending only on the number of nodes. Thus a 20% dense problem with 400 nodes and an 80% dense problem with 400 nodes give rise to UBQP models having 400 variables in each case, differing only in the density of their Q matrices.

4 Summary

Our nonlinear model for the GIS problem is demonstrated to exhibit significant advantages over the standard 0/1 linear model. Computational experience with a set of test problems for graphs with up to 400 nodes and densities over 80% illustrates the attractiveness of the nonlinear modeling approach when accompanied by a straightforward TS method designed to handle more general UBQP problems. A notable feature of the nonlinear models is their ability to represent the problem with substantially fewer variables and constraints than their linear counterparts. The importance of this size differential is dramatically illustrated by problem GIS400D in Table 1, whose linear version contains 64,322 binary variables and 63,922 constraints while its nonlinear

alternative contains 400 binary variables and no constraints. In general, this difference in size outweighs the increased complexity of nonlinearity over linearity, given the existence (in the present context) of an effective method for handling the nonlinear structure. As our results show, approaching the GIS problem via the nonlinear model rather than the linear model greatly extends the size of GIS problems that can be readily solved.

References

1. Alidaee, B., F. Glover, Kochenberger, G., Rego, C.: A new modeling and solution approach for the number partitioning problem. *J. of Appl. Math. Decis. Sci.* **9**(2), 113–121 (2005)
2. Glover, F., Kochenberger, G., Alidaee, B.: Adaptive memory tabu search for binary quadratic programming. *Manage. Sci.* **44**(3), 336–345 (1998a)
3. Glover, F., Kochenberger, G., Alidaee, B., Amini, M.: Tabu search with critical event memory: an enhanced application for binary quadratic programming. In: Voss, S., Martello, S., Osman, I., Roucairol, C. (ed.) *Meta Heuristics: Theory and Applications*, pp. 93–109, Kluwer, Dordrecht (1998b)
4. Hochbaum, D.: Selection, provisioning, shared fixed costs, maximum closure, and implications on algorithmic method today. *Manage. Sci.* **50**(6), 709–723 (2004)
5. Hochbaum, D., Pathria, A.: Forest harvesting and minimum cuts. *Forest Sci.* **43**(4), 544–554 (1997)
6. Kochenberger, G.A., Glover, F., Alidaee, B., Rego, C.: A unified modeling and solution framework for combinatorial optimization problems. *OR Spectrum* **26**, 237–250 (2004)
7. Kochenberger, G., Glover, F., Alidaee, B., Rego, C.: An unconstrained quadratic binary approach to the vertex coloring problem. *Ann. OR* **139**, 229–241 (2005a)
8. Kochenberger, G., Glover, F., Alidaee, B., Lewis, K.: Using the unconstrained quadratic program to model and solve max 2-Sat problems. *Int. J. Oper. Res.* **1**(1 & 2), 89–100 (2005b)
9. Kochenberger, G., Glover, F., Alidaee, B., Wang, H.: Clustering of microarray data via clique partitioning. *J. Comb. Optim.* **10**, 77–92 (2005c)
10. Lewis, M., Alidaee, B., Kochenberger, G.: Using xOx to model and solve the uncapacitated task allocation problem. *OR Lett.* **33**, 176–182 (2005)
11. Pardalos, P., Rodgers, G.: Computational aspects of a B&B algorithm for quadratic zero-one programming. *Computing* **45**, 131–144 (1990)