An unconstrained binary quadratic programming for the maximum independent set problem

Sidi Mohamed Douiri, Souad Elbernoussi

Research laboratory Mathematics Computing and Applications
Faculty of Sciences, University Mohammed V-Agdal
4 Avenue Ibn Battouta, B.P. 1014 RP, Rabat, Morocco
douirisidimohamed@hotmail.fr; souad.elbernoussi@gmail.com

Received: 21 September 2011 / **Revised:** 16 August 2012 / **Published online:** 22 November 2012

Abstract. For a given graph $G = (V, E)$ the maximum independent set problem is to find the largest subset of pairwise nonadjacent vertices. We propose a new model which is a reformulation of the maximum independent set problem as an unconstrained quadratic binary programming, and we resolve it afterward by means of a genetic algorithm. The efficiency of the approach is confirmed by results of numerical experiments on DIMACS benchmarks.

Keywords: maximum independent set, unconstrained binary quadratic programming, genetic algorithm.

1 Introduction

The maximum independent set problem (MIS) is one of the central combinatorial optimization problems and shown to be NP-hard [1]. This problem is relevant for many practical applications in computer science and operation research, and engineering [2], such as register allocation in a compiler, assigning channels to radio station, scheduling exam, graph coloring, and the reader collision problem [3]. Some studies were made on the basis of the greedy algorithms and tabu search [4, 5], or using the intersection graph of an axis-parallel rectangles [6]. In [7] the authors proposed a method based on an improvement of the maximum independent set algorithm given by F. Glover in [8]. Other studies based on a method that utilizes the polynomially solvable critical independent set problem [9]. The unconstrained binary quadratic programming problem is to maximize (or minimize) the function:

$$f(x) = x^{\mathrm{T}} Q x,$$

where $Q = (q_{ij})$ is an $n \times n$ matrix of constants and x is an n-vector of binary variables. This formulation has an ability to model a wide range of different problems on many areas as traffic management [10], machine scheduling [11], UBQP gives evidence to its relevance and effectiveness in the face of known problems by their complexity such

© Vilnius University, 2012

as the set-partitioning problem [12], the set packing problem [13], the vertex coloring problem [14], and the linear ordering problem [15]. Given its NP-hard nature [1], various approaches have been proposed for solving this model using exact methods [16, 17] and metaheuristic methods as memetic algorithms [18, 19], scatter search [20], simulated annealing [21], adaptive memory approaches based on tabu search [22–24], and recently combination of GRASP and tabu search [25].

This paper presents an efficient method for solving the maximum independent set problem (MIS) via its modeling as a unconstrained binary quadratic programming (UBQP).

This paper is organized as follows: we define the maximum independent set problem in Section 2. Section 3 presents the transformation of linear programming to unconstrained binary quadratic programming associated to the maximum independent set problem. In Section 4 the ingredients of our algorithm are described, including an adapted genetic algorithm, and Section 5 draws some conclusions.

2 Problem definition

The maximum independent set problem (MIS) may be written in a form of a linear problem with binary variables as follows:

$$(\text{LP}_{\text{MIS}}) \begin{cases} \max x_0 = \sum_{i=1}^n x_i, \\ x_i + x_j \le 1, \ \ \{i,j\} \in E, \\ x \ \text{binary}, \end{cases} \tag{1}$$

where $x_i^2 = x_i$ and $x_i \in (0, 1)$.

The problem (LP_{MIS}) is constituted of a linear objective function with one types of constraints $x_i + x_j \le 1$ and the number of these inequality constraints is equal to $\operatorname{card}(E)$. Our purpose is to reformulate the problem (LP_{MIS}) in a binary quadratic problem without constraints in the form:

$$(\text{UBQP}_{\text{MIS}}) \begin{cases} \max x_0 = x^{\mathrm{T}} Q x, \\ x \ \text{binary}, \end{cases}$$

where Q is a square symmetric matrix of dimension $\operatorname{card}(V)$. For this we apply a transformation on the constraints set.

3 Transformation

We introduce the constraints in the objective function in the following way. The objective is transformed under shape $x^{\mathrm{T}} D x$ by (1), then the constraints $x_i + x_j \le 1$ are introduced by the product $x_i x_j$. The problem (LP_{MIS}) is thus replaced by the quadratic problem without constraint:

$$(\text{UBQP}_{\text{MIS}}) \begin{cases} \max x_0 = \sum_{i=1}^n x_i - \sum_{(i,j) \in E} x_i x_j, \\ x \ \text{binary}. \end{cases}$$

Example. We consider the undirected graph in Fig. 1, on this graph we obtain three independent sets: $C_1 = (2, 4, 5, 7, 9)$, $C_2 = (1, 6, 8)$ and $C_3 = (3)$.

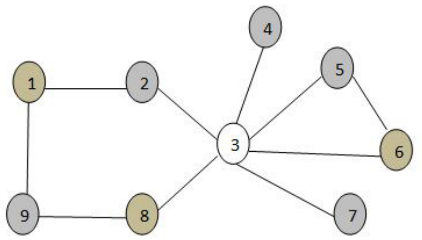

Fig. 1. Independent sets of a graph.

We reformulate it into an unconstrained binary quadratic programming problem. The example satisfies the following linear programming:

$$(\mathrm{LP_{MIS}}) \begin{cases} \max x_0 = \sum_{i=1}^{9} x_i, & x_3 + x_6 \le 1, \\ x_1 + x_2 \le 1, & x_3 + x_7 \le 1, \\ x_1 + x_9 \le 1, & x_3 + x_8 \le 1, \\ x_2 + x_3 \le 1, & x_5 + x_6 \le 1, \\ x_3 + x_4 \le 1, & x_8 + x_9 \le 1, \\ x_3 + x_5 \le 1, & x \text{ binary}. \end{cases}$$

We apply the transformation and obtain:

$$Q = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & -1 & -1 & -1 & -1 & -1 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}.$$

The independent set $C_1 = (2, 4, 5, 7, 9)$, i.e., $x = (010110101)$ gives an optimal solution for $\mathrm{UQP_{MIS}}$ with $x_0 = 5$, if we add or remove one or more vertex to C_1 the value of $\mathrm{UQP_{MIS}}$ will be inferior, for example if we add to C_1 the vertex 3, i.e., $x = (011110101)$ we find $x_0 = -2$.

4 Solving $\mathrm{UBQP_{MIS}}$

After having transformed our MIS problem in an unconstrained binary quadratic form, we try to solve it using an adapted genetic algorithm well by choosing aptly the operators which will be ideal to the MIS problem.

4.1 Genetic algorithm

Our resolution approach of $\mathrm{UBQP_{MIS}}$ problem is based on genetic algorithm (GA). The GA is a research process based on the laws of natural selection and genetics. Generally, a GA consists of three simple operations. Their functioning is extremely simple, we start with an initial population, we evaluate the performance of each individual, and we create a new population of potential solutions using evolutionary operators: selection, crossover and mutation, GA cycle is repeated until a desired stopping criterion is reached.

The individuals of initial population are generated randomly with equal probability such as the genes are assigned a value of 0 or 1. To build a diversified initial population, each individual added to the population must be different to all the existing solutions of the population. Using the best known lower bound based on degrees of vertices d_i given by Caro and Tuza [26]:

$$\sum_{i\in V}\frac{1}{d_i+1}\le \mathrm{card}(\mathrm{MIS}(G))$$

During the initialization of the population we fix the size of the independent set to $p = \lfloor\sum_{i\in V} 1/(d_i+1)\rfloor$ that is, the number of 1 in each individual of the population.

We consider the objective function $x^{\mathrm{T}}Qx$ as evaluation function (fitness) of each individual in the population.

For the selection, first, we randomly choose two individuals, then we apply the tournament selection operator in order to keep the best individual. The comparison between two individuals is carried out according to their fitness.

Crossover is a recombination operator that combines parts of two individual parents to produce offspring that contain some genetic information from both parents. A probability parameter p_c, is set to determine the crossover rate. We opt for a special crossing, we choose two random integers r, s in each parents P_1 and P_1 such as r, $s \in [1,n]$ who represent points inter-genes. We opt for a crossing which aims to permute two blocks of genes of each parents pair. The two selected blocks have the same number of genes which are worth one and the same size. By this crossing we always keep the number of 1 on each new individual obtained.

$$C_1(i)=\begin{cases}P_1(i) & \text{if } i\notin[r_1,s_1],\\ P_2(j) & \text{if } i\in[r_1,s_1] \text{ and } j\in[r_2,s_2],\end{cases}$$

$$C_2(i)=\begin{cases}P_2(i) & \text{if } i\notin[r_2,s_2],\\ P_1(j) & \text{if } i\in[r_2,s_2] \text{ and } j\in[r_1,s_1].\end{cases}$$

Table 1 presents a comparison between the results obtained using a 1-point crossover, a 2-point crossover and the proposed crossover for 25 tests. The tests were performed for a crossover rate equal to $p_c = 0.8$ and a mutation rate equal to $p_m = 0.2$. The average values (X_{avg}) solutions obtained with the proposed crossover are clearly superior compared to the two types of crossover.

The mutation is used with the aim to further explore the search space and reaching solutions that the crossing cannot touch them. We choose randomly two distinct genes that we permute them after, usually with small probability p_m.

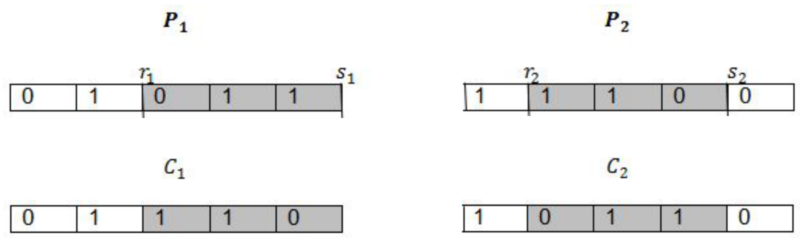

Fig. 2. Used crossover operator.

Table 1. Comparison of solution quality for different type of crossover operations.

	One point crossover	Two point crossover	Proposed crossover
Graph	X_{avg}	X_{avg}	X_{avg}
anna	77.84	76.53	**79.04**
david	33.71	33.60	**34.18**
huck	**27**	26.14	**27**
fpsol2.i.1	301.28	303.59	**306.47**
inithx.i.1	563.06	561.25	**564.80**
zeroin.i.3	118.72	120.55	**122.36**
myciel4	**11**	**11**	**11**
myciel5	20.61	18.27	**23**
myciel6	**47**	45.93	**47**
myciel7	93.65	93.12	**94.07**
mug100-25	**32.19**	31.45	31.86
mulsol.i.5	86.04	84.27	**86.73**

Remark. We fix $p = \lfloor \sum_{i \in V} 1/(d_i + 1) \rfloor$, then we apply the transformation to obtain UBQP_{MIS} problem, after we run the genetic algorithm, and each time we find a maximum independent set we increment the number p by one in order to find an independent set larger.

Algorithm 1. UBQP_{MIS} algorithm.

1. Give the problem UBQP_{MIS}.
2. Put $p = \lfloor \sum_{i \in V} 1/(d_i + 1) \rfloor$.
3. While $T \leq T_{\max}$.
4. Evaluate each individual by calculating its fitness $x_0 = x^{\mathrm{T}} Q x$.
5. Make a selection by tournament.
6. Apply the crossing.
7. Apply the mutation with a very low probability.
8. Evaluate the individuals.
9. If we find a feasible solution x_0 then $p \leftarrow p + 1$.
10. End while.

4.2 Obtained results

In this section, we present the experimental results after running our algorithm on instances available in the literature (see the web page `http://mat.gsia.cmu.edu/COLOR03/` for a complete description of the instances) and compare them with those given by [7,9]. The platform of our experiments is a personal computer windows 7, AMD Athlon(tm) X2 dual-core QL-65 (2cpus) 2. GHz with 4 GB RAM. The results obtained by our algorithm are reported in the Table 2. For each instance, we indicate the number of vertices n, the number of edges d_0, the values X_{S}, X_{DBG} and X_{UBQP} respectively denote the cardinal of the maximal independent set given by S. Butenko et al. [9], DBG algorithm [7], and $\mathrm{UBQP}_{\mathrm{MIS}}$ algorithm, the symbol '–' means that the information is not available. The parameters of the GA for the $\mathrm{UBQP}_{\mathrm{MIS}}$ problem are: population size$= 100$, crossover rate $p_c = 0.8$, mutation rate $p_m = 0.2$, and the maximum iterations number $T_{\max} = 200$.

Table 2. Computational results.

Graph	n	d_0	X_{S}	X_{DBG}	X_{UBQP}	$Time_{\mathrm{UBQP}}$ (s)
anna	138	493	80	80	80	0.87
david	87	406	36	36	36	0.49
fpsol2.i.1	496	11654	307	307	307	12.39
fpsol2.i.2	451	8691	261	261	261	10.77
fpsol2.i.3	425	8688	238	238	238	9.18
huck	74	301	27	27	27	0.51
inithx.i.1	864	18707	566	566	566	15.26
inithx.i.2	645	13979	365	365	365	10.94
inithx.i.3	621	13969	360	360	360	7.89
jean	80	254	38	38	38	0.60
zeroin.i.1	211	4100	120	120	120	4.22
zeroin.i.2	211	3541	127	127	127	4.05
zeroin.i.3	206	3540	123	123	123	5.13
1-FullIns5	282	3247	–	138	138	6.67
3-FullIns4	405	3524	–	193	193	8.42
2-Inser4	149	541	–	74	74	1.05
3-Inser3	56	110	–	27	27	0.38
4-Inser3	79	156	–	39	39	0.96
games120	120	638	–	22	22	1.20
mulsol.i.1	197	3925	–	100	100	5.26
mulsol.i.5	186	3973	–	88	88	4.81
mug88-1	88	146	–	–	29	1.17
mug88-25	88	146	–	–	29	0.95
mug100-1	100	166	–	–	33	1.78
mug100-25	100	166	–	–	33	2.34
myciel3	11	20	–	–	5	0.21
myciel4	23	71	–	–	11	0.33
myciel5	47	236	–	–	23	0.57
myciel6	95	755	–	–	47	0.84

Graph	n	d_0	X_{S}	X_{DBG}	X_{UBQP}	$Time_{\mathrm{UBQP}}$ (s)
myciel7	191	2360	–	–	95	3.96
dsjc125.1	125	736	–	–	32	2.03
dsjc250.1	250	3218	–	–	39	5.49
le450_15a	450	8168	–	–	71	9.41
le450_15d	450	16750	–	–	37	12.16
le450_25a	450	8260	–	–	85	11.07
le450_25d	450	17425	–	–	40	13.22

5 Conclusion

In this paper we have presented a method to reformulate the linear program of maximum independent set problem as an unconstrained binary quadratic problem, the proposed algorithm proves to be highly effective in solving a range of benchmark instances from the literature. Most of these instances have been easily resolved with a reasonable execution time. For the first 13 instances we obtained identical results than those in [9], our results are also identical to the first 21 instances given in [7], to show the effectiveness of our algorithm we tested 15 other instances.

References

1. M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York, 1979.

2. A. Al-Jaber, A. Sharieh, Algorithms based on weight factors for maximum independent set, *DIRASAT, Nat. Sci.*, **27**(1), pp. 74–90, 2000.

3. M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, New York, 1983.

4. T.A. Feo, M.G.C. Resende, S.H. Smith, A greedy randomized adaptive search procedure for maximum independent set, *Oper. Res.*, **42**(5), pp. 860–878, 1994.

5. C. Friden, A.Hertz, D. de Werra, Tabaris: An exact algorithm based on tabu search for finding a maximum independent set in a graph, *Comput. Oper. Res.*, **155**, pp. 43–445, 1990.

6. T.M. Chan, A note on maximum independent sets in rectangle intersection graphs, *Inf. Process. Lett.*, **89**(1), pp. 19–23, 2004.

7. S.M. Douiri, S. El Bernoussi, A new heuristic for the sum coloring problem, *Appl. Math. Sci.*, **5**(63), pp. 3121–3129, 2011.

8. F. Glover, Tutorial on surrogate constraint approaches for optimization in graphs, *J. Heuristics*, pp. 175–228, 2003.

9. S. Butenko, S. Trukhanov, Using critical sets to solve the maximum independent set problem, *Oper. Res. Lett.*, **35**(4), pp. 519–524, 2007.

10. G. Gallo, P. Hammer, B. Simeone, Quadratic knapsack problems, *Math. Program. Study*, **12**, pp. 132–149, 1980.

11. B. Alidaee, G.A. Kochenberger, A. Ahmadian, 0-1 Quadratic programming approach for the optimal solution of two scheduling problems, *Int. J. Syst. Sci.*, **25**, pp. 401–408, 1994.

12. M. Lewis, G.A. Kochenberger, B. Alidaee, A new modelling and solution approach for the set-partitioning problem, *Comput. Oper. Res.*, **35**(3), pp. 807–813, 2008.

13. B. Alidaee, G.A. Kochenberger, K. Lewis, M. Lewis, H. Wang, A new approach for modelling and solving set packing problems, *Eur. J. Oper. Res.*, **86**(2), pp. 504–512, 2008.

14. G.A. Kochenberger, F. Glover, B. Alidaee, C. Rego, An unconstrained quadratic binary programming approach to the vertex coloring problem, *Ann. Oper. Res.*, **139**, pp. 229–241, 2005.

15. M. Lewis, B. Alidaee, F. Glover, G.A. Kochenberger, A note on xQx as a modelling and solution framework for the linear ordering problem, *Int. J. Oper. Res.*, **5**(2), pp. 152–162, 2009.

16. P. Pardalos, G.P. Rodgers, Computational aspects of a branch and bound algorithm for quadratic zero-one programming, *Computing*, **45**, pp. 131–144, 1990.

17. C. Helmberg, F. Rendl, Solving quadratic $(0, 1)$-problem by semidefinite programs and cutting planes, *Math. Program.*, **82**, pp. 291–315, 1998.

18. P. Merz, K. Katayama, Memetic algorithms for the unconstrained binary quadratic programming problem, *Biosystems*, **78**, pp. 99–118, 2004.

19. Z. Lü, F. Glover, J.K. Hao, A hybrid metaheuristic approach to solving the UBQP problem, *Eur. J. Oper. Res.*, **207**(3), pp. 1254–1262, 2010.

20. M. Amini, B. Alidaee, G.A. Kochenberger, *A Scatter Search Approach to Unconstrained Quadratic Binary Programs, New Methods in Optimization*, McGraw-Hill, New York, NY, 1999, pp. 317–330.

21. K. Katayama, H. Narihisa, Performance of simulated annealing-based heuristic for the unconstrained binary quadratic programming problem, *Eur. J. Oper. Res.*, **134**, pp. 103–119, 2001.

22. F. Glover, G.A. Kochenberger, B. Alidaee, Adaptive memory tabu search for binary quadratic programs, *Manage. Sci.*, **44**, pp. 336–345, 1998.

23. G. Palubeckis, Multistart tabu search strategies for the unconstrained binary quadratic optimization problem, *Ann. Oper. Res.*, **131**, pp. 259–282, 2004.

24. Y. Wang, Z. Lü, F. Glover, J.K. Hao, Backbone guided tabu search for solving the UBQP problem, *J. Heuristics*, 2011, doi:10.1007/s10732-011-9164-4.

25. Y. Wang, Z. Lü, F. Glover, J.K. Hao, Probabilistic GRASP-Tabu Search algorithms for the UBQP problem, *Comput. Oper. Res.*, 2012, doi:10.1016/j.cor.2011.12.006.

26. Y. Caro, Z. Tuza, Improved lower bounds on k-independence, *J. Graph Theory*, **15**, pp. 99–107, 1991.