OR Spectrum (2004) 26: 237–250
DOI: 10.1007/s00291-003-0153-3

OR Spectrum

A unified modeling and solution framework for combinatorial optimization problems*

Gary A. Kochenberger[1], **Fred Glover**[2], **Bahram Alidaee**[3], **and Cesar Rego**[3]

[1] School of Business, University of Colorado at Denver, Denver, CO 80217, USA (e-mail: Gary.Kochenberger@cudenver.edu)
[2] School of Business, University of Colorado at Boulder, Boulder, CO 80304, USA (e-mail: Fred.Glover@Colorado.edu)
[3] Hearin Center for Enterprise Science, University of Mississippi, University, MS 38677, USA (e-mail: {Balidaee,Crego}@bus.olemiss.edu)

Abstract. Combinatorial optimization problems are often too complex to be solved within reasonable time limits by exact methods, in spite of the theoretical guarantee that such methods will ultimately obtain an optimal solution. Instead, heuristic methods, which do not offer a convergence guarantee, but which have greater flexibility to take advantage of special properties of the search space, are commonly a preferred alternative. The standard procedure is to craft a heuristic method to suit the particular characteristics of the problem at hand, exploiting to the extent possible the structure available. Such tailored methods, however, typically have limited usefulness in other problems domains.

An alternative to this problem specific solution approach is a more general methodology that recasts a given problem into a common modeling format, permitting solutions to be derived by a common, rather than tailor-made, heuristic method. Because such general purpose heuristic approaches forego the opportunity to capitalize on domain-specific knowledge, they are characteristically unable to provide the effectiveness or efficiency of special purpose approaches. Indeed, they are typically regarded to have little value except for dealing with small or simple problems.

This paper reports on recent work that calls this commonly held view into question. We describe how a particular unified modeling framework, coupled with latest advances in heuristic search methods, makes it possible to solve problems from a wide range of important model classes.

Keywords: Integer programming – Combinatorial optimization – Metaheuristics

* This research was supported in part by ONR grants N000140010598 and N000140310621.

Correspondence to: G.A. Kochenberger

Introduction

The optimization folklore strongly emphasizes the unproductive consequences of converting problems from a specific class to a more general representation, since the “domain-specific structure” of the original setting then becomes invisible and can not be exploited by a method for the more general problem representation. Nevertheless, there is a strong motivation to attempt such a conversion in many applications to avoid the necessity to develop a new method for each new class. We demonstrate the existence of a general problem representation that frequently overcomes the limitation commonly ascribed to such models. Contrary to expectation, when a specially structured problem is translated into this general form, it often does not become much harder to solve, and sometimes becomes even easier to solve, provided the right type of solution approach is applied.

Our research over the past few years has revealed that this unified approach is surprisingly successful for a wide range of important problems, often surpassing the performance of established special-purpose methods for particular problem classes. As such, this unified approach holds great promise as a practical method for solving a variety of important problems.

The unified model

The model with this appealing property is the unconstrained quadratic binary programming problem, accompanied by the device of introducing quadratic infeasibility penalty functions to handle constraints. Not only is this model capable of representing many “special case” problem classes, but it can be advantageously exploited by adaptive memory (tabu search) metaheuristics and associated evolutionary (scatter search) methods. Computational outcomes disclose the effectiveness of this combined modeling and solution approach for problems from a diverse collection of challenging settings.

The unconstrained quadratic program can be written in the form:

$$UQP : \min f(x) = xQx$$

where Q is an n by n matrix of constants and x is an n-vector of binary variables. UQP is notable for its ability to represent a significant variety of important problems. The applicability of this representation has been reported in diverse settings such as spin glasses and circuit board layout (De Simone et al. [11], Grotschel et al. [19] and Palubeckis [33]), financial analysis (Laughunn [30], McBride and Yormak [31]), computer aided design (Krarup and Pruzan [29]), traffic management (Gallo et al. [13], Witsgall [40]), machine scheduling (Alidaee, Kochenberger, and Ahmadian [1]), cellular radio channel allocation (Chardaire and Sutter [9]), molecular conformation (Phillips and Rosen [37]) and the prediction of epileptic seizures (Iasemidus et al. [25]. Moreover, many satisfiability problems (Hammer and Rudeanu [21], Boros and Hammer [5], Boros and Prekopa [7]) as well as combinatorial optimization problems pertaining to graphs such as determining maximum cliques, maximum cuts, maximum vertex packing, minimum coverings, maximum independent sets, and maximum independent weighted sets are known to be capable of being formulated by the UQP problem (see for instance Boros and Hammer [5], Bourjolly et al. [8], Hammer et al. [20] as well as Du and Pardalos [12], Pardalos and Rodgers [34, 35], and Pardalos and Xue [36]).

The application potential of UQP is substantially greater than this, however, due to reformulation methods that enable certain constrained models to be re-cast in the form of UQP. Hammer and Rudeanu [21], Hansen [22], and Hansen et al. [23] show that any quadratic (or linear) objective in bounded integer variables and constrained by linear equations can be reformulated as a UQP model. Nonetheless, few applications of this idea appear in the literature. Our purpose here, based on extensive experience with a wide variety of problems, is to establish that such reformulation into the UQP format is not merely a representational novelty, but a unified framework of practical consequences.

Transformation to xQx

Many practical combinatorial optimization problems can be modeled as constrained optimization problems of the form

$$\min x_0 = xQx$$

$subject\ to$

$$Ax = b, \quad x\ binary$$

The foregoing model accommodates both quadratic and linear objective functions since the linear case results when Q is a diagonal matrix (observing that $x_j^2 = x_j$ when x_j is a 0-1 variable). Problems with inequality constraints can also be put into this form by introducing so-called *slack variables* to convert the inequalities into equations, and representing these bounded slack variables by a binary expansion. These constrained quadratic optimization models are then converted into equivalent UQP models by adding a quadratic infeasibility penalty function to the objective function as an alternative to explicitly imposing the constraints $Ax = b$. The general approach to such re-casting, which we call transformation # 1, is given below:

Transformation 1. We choose a positive scalar P, to yield

$$\begin{aligned} x_0 &= xQx + P\,(Ax - b)^t\,(Ax - b) \\ &= xQx + xDx + c \\ &= x\hat{Q}x + c \end{aligned}$$

where the matrix D and the additive constant c result directly from the matrix multiplication indicated. We can drop the additive constant, whereupon the equivalent unconstrained version of our constrained problem becomes

$$UQP(PEN) : \min x\hat{Q}x, \ x\ binary$$

From a theoretical standpoint, a suitable choice of the penalty scalar P can always be chosen so that the optimal solution to $UQP(PEN)$ is the optimal solution to the original constrained problem (Hammer and Rudeanu [21). From a practical standpoint, however, experience has shown that penalty-based conversions in other settings have uniformly proved to be highly unstable, engendering numerical difficulties and poor solution performance when the penalties are large, and producing invalid representations of the original problem when the penalties are smaller. Finding a proper trade-off between penalty size (and the design of a method to exploit the penalized representation) has turned out to be feasible only in the case of linear and convex programming domains, where penalty considerations are much simpler than in combinatorial optimization. By contrast, however, our experience with penalty-based representations of the UQP model for combinatorial optimization problems has shown them to be easy to work with and highly robust. As reported in [27], valid and computationally stable penalty values can be found without difficulty and a wide range of such values work well.

In addition to the modeling possibilities introduced by Transformation 1, a very important special class of constraints that arise in many applications can be handled by an alternative approach, given below.

Transformation 2. This approach is convenient for problems with considerations that isolate two specific alternatives and prohibit both from being chosen. That is, for a given pair of alternatives, one or the other but not both *may* be chosen. If x_j and x_k are binary variables denoting whether or not alternatives j and k are chosen, the standard constraint that allows one choice but precludes both is:

$$x_j + x_k \leq 1$$

Then, for a positive scalar P, adding the penalty function Px_jx_k to the objective function is a simple alternative to imposing the constraint in a traditional manner. For problems with a linear objective function, the scalar P (with respect to transformation # 2) can be chosen as small as the largest objective function coefficient [5]. This penalty function has sometimes been used by to convert certain optimization problems on graphs into an equivalent UQP model as referenced in the previous section. Its potential application, however, goes far beyond these settings as demonstrated in this paper. Variable upper bound constraints of the form $x_{ij} \leq y_i$ can be accommodated by Transformation 2 by first replacing each y_i variable by $1 - y_i'$, where y_i' is the complementary variable that equals 1 when $y_i = 0$ and equals 0 when $y_i = 1$. The opportunity to employ this modeling device in the context of Transformation 2 makes it possible to conveniently model a variety of additional problem types.

The constraint associated with Transformation # 2 appears in many important applications which leads us to single it our here as an important alternative to Transformation # 1. We note, however, that many other problem specific special cases exist that can be employed to quickly yield quadratic equivalent representations. We illustrate this later in the paper when we discuss results we have obtained for the max 2-SAT problem.

Examples

Before highlighting some of the problem classes to which we have successfully applied the foregoing transformation approaches, we give two small examples from classical NP-hard problem settings to provide concrete illustrations.

Example 1. Set Partitioning. The classical set partitioning problem is found in applications that range from vehicle routing to crew scheduling [26, 32]. As an illustration, consider the following small example:

$$\min x_0 = 3x_1 + 2x_2 + x_3 + x_4 + 3x_5 + 2x_6$$

subject to

$$\begin{aligned} x_1 + x_3 + x_6 &= 1 \\ x_2 + x_3 + x_5 + x_6 &= 1 \\ x_3 + x_4 + x_5 &= 1 \\ x_1 + x_2 + x_4 + x_6 &= 1 \end{aligned}$$

and x binary. Applying Transformation 1 with $P = 10$ gives the equivalent UQP model:

$UQP(PEN)$: $\min x\hat{Q}x,\ x\ binary$

where the additive constant, c, is 40 and

$$\hat{Q} = \begin{bmatrix} -17 & 10 & 10 & 10 & 0 & 20 \\ 10 & -18 & 10 & 10 & 10 & 20 \\ 10 & 10 & -29 & 10 & 20 & 20 \\ 10 & 10 & 10 & -19 & 10 & 10 \\ 0 & 10 & 20 & 10 & -17 & 10 \\ 20 & 20 & 20 & 10 & 10 & -28 \end{bmatrix}$$

Solving $UQP(PEN)$ by the Tabu Search method of Glover et al. [17, 18] we obtain an optimal solution $x_1 = x_5 = 1$, (all other variables equal to 0) for which $x_0 = 6$. In the straightforward application of Transformation 1 to this example, it is to be noted that the replacement of the original problem formulation by the $UQP(PEN)$ model did not involve the introduction of new variables. In many applications, Transformation 1 and Transformation 2 can be used in concert to produce an equivalent UQP model, as demonstrated next.

Example 2. The K-Coloring Problem.
Vertex coloring problems seek to assign colors to nodes of a graph such that adjacent nodes are assigned different colors. The K-coloring problem attempts to find such a coloring using exactly K colors. A wide range of applications, ranging from frequency assignment problems to printed circuit board design problems [10, 39], can be represented by the K-coloring model.

Such problems can be modeled as satisfiability problems using the assignment variables as follows:

Let x_{ij} to be 1 if node i is assigned color j, and to be 0 otherwise.

Since each node must be colored, we have

$$\sum_{j=1}^{K} x_{ij} = 1 \quad i = 1, ..., \ n \tag{1}$$

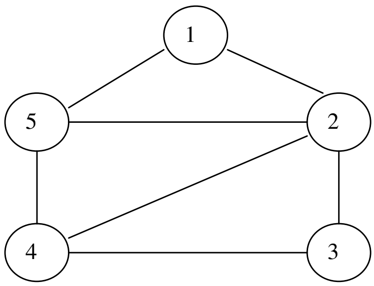
where n is the number of nodes in the graph. A feasible coloring requires that adjacent nodes are assigned different colors. This is accomplished by imposing the constraints

$$x_{ip} + x_{jp} \leq 1 \quad p = 1, ..., K \tag{2}$$

for all adjacent nodes (i, j) in the graph.

This problem can be re-cast into the form of UQP by using Transformation 1 on the assignment constraints of (1) and Transformation 2 on the adjacency constraints of (2). No new variables are required. Since the model of (1) and (2) has no explicit objective function, any positive value for the penalty, P, will do. The following example gives a concrete illustration of the re-formulation process.

Consider the following graph and assume we want find a feasible coloring of the nodes using 3 colors.



Our satisfiablity problem is that of finding a solution to:

$$x_{i1} + x_{i2} + x_{i3} = 1 \qquad i = 1, \ 5 \tag{3}$$

$$x_{ip} + x_{jp} \leq \ 1 \qquad p = 1, \ 3 \tag{4}$$

(for all adjacent nodes i and j)

In this traditional form, the model has 15 variables and 26 constraints. To recast this problem into the form of UQP, we use Transformation 1 on the equations of (3) and Transformation 2 on the inequalities of (4). Arbitrarily choosing the penalty P to be 4, we get the equivalent problem:

$UQP(Pen) : \min x\hat{Q}x$

where the $\hat{Q}$ matrix is:

$$\hat{Q} = \begin{bmatrix} -4 & 4 & 4 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 4 & -4 & 4 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \\ 4 & 4 & -4 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \\ 4 & 0 & 0 & -4 & 4 & 4 & 4 & 0 & 0 & 4 & 0 & 0 & 4 & 0 & 0 \\ 0 & 4 & 0 & 4 & -4 & 4 & 0 & 4 & 0 & 0 & 4 & 0 & 0 & 4 & 0 \\ 0 & 0 & 4 & 4 & 4 & -4 & 0 & 0 & 4 & 0 & 0 & 4 & 0 & 0 & 4 \\ 0 & 0 & 0 & 4 & 0 & 0 & -4 & 4 & 4 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 & 4 & -4 & 4 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & -4 & 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 4 & 0 & 0 & -4 & 4 & 4 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 & 0 & 4 & 0 & 4 & -4 & 4 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 4 & 4 & 4 & -4 & 0 & 0 & 4 \\ 4 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & -4 & 4 & 4 \\ 0 & 4 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 4 & -4 & 4 \\ 0 & 0 & 4 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & -4 \end{bmatrix}$$

Solving this unconstrained model, $x\hat{Q}x$, yields the feasible coloring:

$x_{11}, x_{22}, x_{33}, x_{41}, x_{53}, = 1 \ \ all\ other \ \ x_{ij} = 0$

This approach to coloring problems has proven to be very effective for a wide variety of coloring instances from the literature. Later in this paper we present some computational results for several standard k-coloring problems. An extensive presentation of the xQx approach to a variety of coloring problems, including a generalization of the K-coloring problem considered here, is given in Kochenberger, Glover, Alidaee, and Rego [28].

Solution approaches to UQP

Due to its computational challenge and application potential, UQP has been the focus of a considerable number of research studies in recent years, including both exact and heuristic solution approaches. Recent papers report on the branch and bound (exact) approaches as well as a variety of modern heuristic methods including simulated annealing, genetic algorithms, tabu search, and scatter search (see [27] for references to these and other works). Each of these approaches exhibits some degree of success. However, the exact methods degrade rapidly with problem size, and have meaningful application to general UQP problems with no more than 100 variables. (A notable exception to this for the Ising spin glass problem is discussed in [11].) For larger problems, heuristic methods are usually required. Several proposed heuristics, including the DDT method of Boros, Hammer, and Sun [6] and the "one-pass" procedures of Glover, Alidaee, Rego, and Kochenberger [16] have proven to be effective in certain instances. Two methods we have found to be particularly successful for a wide variety of problems are based on tabu search [14, 17, 18] and on the related evolutionary strategy scatter search [15]. In the following section we highlight our tabu search approach which was used to produce the computational results referenced later in this paper.

Although not pursued by us here, we note that an alternative approach is to solve UQP as a continuous non-linear optimization problem within the unit cube. This allows other heuristic/approximation methods based on continuous optimization methodologies to be applied (see [4, 7, 38]).

Tabu search overview

Our TS method for UQP 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 completed and the next cycle is launched.

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 subtle 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, Kochenberger, Alidaee, and Amini [17].

Computational experience

Our results of applying the tabu search and associated scatter search metaheuristics to combinatorial problems recast in UQP form have been uniformly attractive in terms of both solution quality and computation times. Although our methods are designed for the completely general form of UQP, without any specialization to take advantage of particular types of problems reformulated in this general representation, our outcomes have typically proved competitive with or even superior to those of specialized methods designed for the specific problem structure at hand. Our broad base of experience with UQP as a modeling and solution framework includes a substantial range of problem classes including:

Quadratic assignment problems
Capital budgeting problems
Multiple knapsack problems
Task allocation problems (distributed computer systems)
Maximum diversity problems
P-median problems
Asymmetric assignment problems
Symmetric assignment problems
Side constrained assignment problems
Quadratic knapsack problems
Constraint satisfaction problems (CSPs)
Discrete tomography problems
Set partitioning problems
Set packing problems
Warehouse location problems
Maximum clique problems
Maximum independent set problems
Maximum cut problems
Graph coloring problems
Number partitioning problems
Linear ordering problems
Clique partitioning problems
SAT problems

We are currently solving problems via UQP with more than 35,000 variables in the quadratic representation and are working on enhancements that will permit larger instances to be solved. For each of the problem classes listed above we have considerable computational experience showing that this approach consistently produces high quality solutions within very modest computational time. Below we present some representative results for two of the problem classes. Details of our experience with other problems will be presented in future papers.

K-coloring results

Earlier in the paper we presented a small example of the 3-coloring problem. To test the potential attractiveness of the UQP modeling and solution approach to K-coloring problems, 15 standard test problems from the literature were recast into the form of UQP and solved by our tabu search method. Table 1 gives a description of the problems and presents the results. All computations were carried out on a 1.7 gigahertz PC.

The first four columns of Table 1 indicate the problem identifier along with the size of the graphs and the number of colors (K) to be used. The last three columns give the number of variables involved, whether or not a feasible coloring was found utilizing K colors, and the time our tabu search method took to find a solution. Note that feasible colorings (solutions) were quickly found in all 15 cases. In fact, the solutions shown in Table 1 are known to be optimal.

Table 1. K-coloring test problems from http://mat.gsia.cmu.edu/COLOR/instances.html

ID	# Vertices	# Edges	# K	xQx Variables	xQx feasible	xQx Time
Myciel3	11	20	4	44	Yes	<1 sec
Myciel4	23	71	5	115	Yes	<1 sec
Myciel5	47	236	6	282	Yes	<1 sec
Myciel6	95	755	7	665	Yes	<1 sec
Myciel7	191	2360	8	1528	Yes	<1 sec
Anna	138	493	11	1518	Yes	47 sec
David	87	406	11	957	Yes	1 min, 13 sec
Huck	74	301	11	814	Yes	2 sec
Jean	80	254	10	800	Yes	<1 sec
Games120	120	638	9	1080	Yes	<1 sec
Queen5_5	25	160	5	125	Yes	<1 sec
Queen6_6	36	290	7	252	Yes	<1 sec
Queen7_7	49	476	7	343	Yes	<1 sec
Queen8_12	96	1368	12	1162	Yes	<1 sec
Queen8_8	64	728	9	576	Yes	<1 sec

Max 2-SAT results

Several authors (Hammer and Rudeanu [21], Hansen and Jaumard [24], Boros and Hammer [5]) have established the connection between SAT problems and nonlinear penalty functions. The special case of Max 2-SAT is particularly well suited for this approach as it leads naturally to an xQx representation. Our experience, as shown below, indicates that this is a very attractive way to approach this class of problems.

For a 2-SAT problem, a given clause could have zero, one, or two negations, each with a corresponding (classical) linear constraint. Each linear constraint, in turn, has an exact quadratic penalty that serves as an alternative to the linear constraint. The three possibilities and their constraint/penalty pairs are:

(a) *No negations:*
Classical constraint: $x_i + x_j \geq 1$
Exact Penalty: $(1 - x_i - x_j + x_i x_j)$
(b) *One negation:*
Classical constraint: $x_i + \bar{x}_j \geq 1$
Exact Penalty: $(x_j - x_i x_j)$
(c) *Two negations:*
Classical constraint: $\bar{x}_i + \bar{x}_j \geq 1$
Exact Penalty: $(x_i x_j)$

It is easy to see that the quadratic penalties shown are zero for feasible solutions and positive for infeasible solutions. Thus, these special penalties can be used to readily construct a penalty function of the form of xQx (simply by adding the

Table 2. Problems from Borchers and Furman [3]

n	m	Best known solution	xQx solution	xQx time	Maxsat[3] solution	Maxsat time
50	100	4	4	< 1	4	0.4
50	150	8	8	< 1	8	1.5
50	200	16	16	< 1	16	116.2
50	250	22	22	< 1	22	652.4
50	300	32	32	< 1	32	8,763
50	350	41	41	< 1	NA	$>$ 12 hr
50	400	45	45	< 1	NA	$>$ 12 hr
50	450	63	63	< 1	NA	$>$ 12 hr
50	500	66	66	< 1	NA	$>$ 12 hr
100	200	5	5	< 2	5	3.2
100	300	15	15	< 2	15	13,770
100	400	29	29	< 2	NA	$>$ 12 hr
100	500	44	44	< 2	NA	$>$ 12 hr
100	600	?	65	< 2	NA	$>$ 12 hr
150	300	4	4	< 3	4	4.1
150	450	22	22	< 3	NA	$>$ 12 hr
150	600	38	38	< 3	NA	$>$ 12 hr

penalties together) which we seek to minimize. We have found this approach to be very effective on a variety of test problems. Table 2 shows the results we obtained via this approach on a set of test problems from the literature.

Remarks.

1. All times in seconds unless noted otherwise.
2. Maxsat is an exact method developed by Borchers & Furman
3. Maxsat results obtained on IBM RS/6000-590
4. xQx results obtained on a 1.6 MHZ PC.
5. Each xQx run was for 50 SPAN cycles
6. Problem 100_600 was previously unsolved.

As shown in Table 2, by re-casting each Max 2-SAT instance into the form of xQx and solving the resulting unconstrained quadratic binary program with our Tabu Search heuristic, we were able to find best known solutions very quickly to all test problems considered. By way of contrast, the method of Borchers and Furman took a very long time on several problems and was unable to find best known results for several instances in the allotted 12 hour time limit. In addition to the problems of Table 2 above, we have successfully applied this approach to randomly generated problems with as many as 1000 variables and more than 10,000 clauses where best known results are found in roughly one minute of computation time.

The results shown in Tables 1 and 2 above serve as strong evidence of the attractiveness of the xQx approach for the problems considered. Considering both solution quality and the time taken to produce these solutions, this approach is very competitive with special purpose methods constructed specifically for vertex coloring and max 2-Sat problems. We note in passing that similar performance relative to special purpose methods has been obtained for the other problem classes singled out earlier in the paper as well.

Summary

We have demonstrated how a variety of disparate combinatorial problems can be solved by first re-casting them into the common modeling framework of the unconstrained quadratic binary program. Once in this unified form, the problems can be solved effectively by adaptive memory tabu search metaheuristics and associated evolutionary (scatter search) procedures.

Our findings challenge the conventional wisdom that places high priority on preserving linearity and exploiting specific structure. Although the merits of such a priority are well-founded in many cases, the UQP domain appears to offer a partial exception. In forming $UQP(PEN)$, we destroy any linearity that the original problem may have exhibited. Moreover, any exploitable structure that may have existed originally is "folded" into the $\hat{Q}$ matrix, and the general solution procedure we apply takes no advantage of it. Nonetheless, our solution outcomes have been remarkably successful, yielding results that rival the effectiveness of the best specialized methods.

This combined modeling/solution approach provides a unifying theme that can be applied in principle to all linearly constrained quadratic and linear programs in bounded integer variables, and the computational findings for a broad spectrum of problem classes raises the possibility that similarly successful results may be obtained for even wider ranges of problems. As our methods for UQP continue to improve with ongoing research, the UQP model offers a representational tool of particular promise.

References

1. Alidaee B, Kochenberger G, Ahmadian A (1994) 0-1 quadratic programming approach for the optimal solution of two scheduling problems. International Journal of Systems Science 25: 401–408
2. Amini M, Alidaee B, Kochenberger G (1999) A scatter search approach to unconstrained quadratic binary programs. In: Corne D, Dorigo M, Glover F (eds) New methods in optimization, pp 317–330. McGraw-Hill, New York
3. Borchers B, Furman J (1999) A two-phase exact algorithm for max-Sat and weighted max SAT. Journal of Combinatorial Optimization 2: 299–306
4. Boros E, Hammer P (1991) The max-cut problem and quadratic 0-1 optimization, polyhedral aspects, relaxations and bounds. Annals of OR 33: 151–225
5. Boros E, Hammer P (2002) Pseudo-Boolean optimization. Discrete Applied Mathematics 123(1–3): 155–225

6. Boros E, Hammer P, Sun X (1989) The DDT method for quadratic 0-1 minimization. RUTCOR Research Center, RRR 39-89
7. Boros E, Prekopa A (1989) Probabilistic bounds and algorithms for the maximum satisfiability problem. Annals of OR 21: 109–126
8. Bourjolly JM, Gill P, Laporte G, Mercure H (1994) A quadratic 0/1 optimization algorithm for the maximum clique and stable set problems. Working paper, University of Montreal
9. Chardaire P, Sutter A (1994) A decomposition method for quadratic zero-one programming. Management Science 41(4): 704–712
10. Cho J-D, Raje S, Sarrafzadeh M (1998) Fast approximation algorithms on maxcut, K-coloring, and K-color ordering for VLSI applications. IEE Transactions on Computers 47: 1253–1266
11. De Simone C, Diehl M, Junger M, Mutzel P, Reinelt G, Rinaldi G (1995) Exact ground state4s of ising spin glasses: new experimental results with a branch and cut algorithm. Journal of Statistical Physics 80: 487–496
12. Du D, Pardalos P (eds) (1998–99) Handbook of combinatorial optimization, (4 vol). Kluwer, Boston
13. Gallo G, Hammer P, Simeone B (1980) Quadratic knapsack problems. Mathematical Programming 12: 132–149
14. Glover F, Laguna M (1997) Tabu search. Kluwer, Boston
15. Glover F (1998) A template for scatter search and path relinking. School of Business, University of Colorado, Technical Report
16. Glover F, Alidaee B, Rego C, Kochenberger G (2002) One-pass heuristics for large-scale unconstrained binary quadratic programs. EJOR 137: 272–287
17. Glover F, Kochenberger G, Alidaee B, Amini MM (1999) Tabu with search critical event memory: an enhanced application for binary quadratic programs. In: Voss S, Martello S, Osman I, Roucairol C (eds) Metaheuristics: advances and trends in local search paradigms for optimization. Kluwer, Boston
18. Glover F, Kochenberger G, Alidaee B (1998) Adaptive memory tabu search for binary quadratic programs. Management Science 44(3): 336–345
19. Grotschel M, Junger M, Reinelt G (1988) An application of combinatorial optimization to statistical physics and circuit layout design. Operations Research 36(3): 493–513
20. Hammer P, Hansen P, Simone B (1981) Upper planes of quadratic 0/1 functions and stability in graphs. In: Mangasarian O, Meyer R, Robinson S (eds) Nonlinear programming, Vol 4, pp 395–414. Academic Press, New York
21. Hammer P, Rudeanu S (1968) Boolean methods in operations research. Springer, Berlin Heidelberg New York
22. Hansen PB (1979) Methods of nonlinear 0-1 programming. Annals Discrete Mathematic 5: 53–70
23. Hansen P, Jaumard B, Mathon V (1993) Constrained nonlinear 0-1 programming. INFORMS Journal on Computing 5(2): 97–119
24. Hansen P, Jaumard B (1990) Algorithms for the maximum satisfiability problem. Computing 44: 279–303
25. Iasemidis LD, Shiau DS, Sackellares JC, Pardalos P (2000) Transition to epileptic seizures: optimization. DIMACS Series in Discrete Math and Theoretical Computer Science 55: 55–73
26. Joseph A (2002) A concurrent processing framework for the set partitioning problem. Computers & Operations Research 29: 1375–1391
27. Kochenberger G, Glover F, Alidaee B, Rego C (2002) Solving combinatorial optimization problems via reformulation and adaptive memory metaheuristics. Working Paper, University of Colorado at Denver

28. Kochenberger G, Glover F, Alidaee B, Rego C(2002) An unconstrained quadratic binary programming approach to the vertex coloring problem. Working Paper, University of Colorado at Denver
29. Krarup J, Pruzan A (1978) Computer aided layout design. Mathematical Programming Study 9: 75–94
30. Laughunn DJ () Quadratic binary programming. Operations Research 14: 454–461
31. McBride RD, Yormack JS (1980) An implicit enumeration algorithm for quadratic integer programming. Management Science 26: 282–296
32. Mingozzi A, Boschetti M, Ricciardelli S, Blanco L (1999) A set partitioning approach to the crew scheduling problem. Operations Research 47(6): 873–888
33. Palubeckis G (1995) A heuristic-branch and bound algorithm for the unconstrained quadratic zero-one programming problem. Computing 54(4): 284–301
34. Pardalos P, Rodgers GP (1990) Computational aspects of a branch and bound algorithm for quadratic zero-one programming. Computing 45: 131-144
35. Pardalos P, Rodgers GP (1992) A branch and bound algorithm for maximum clique problem. Computers & OR 19: 363–375
36. Pardalos P, Xue J (1994) The maximum clique problem. The Journal of Global Optimization 4: 301–328
37. Phillips AT, Rosen JB (1994) A quadratic assignment formulation of the molecular conformation problem. The Journal of Global Optimization 4: 229–241
38. Rosenberg IG (1972) 0-1 optimization and non-linear programming. Revue Francaise d'Automatique, d'Informatique et de Rescherche Operationnelle (Série Blueu) 2: 95–97
39. Tsuda N (2000) Fault-tolerant processor arrays using additional bypass linking allocated by graph-node coloring. IEEE Transactions on Computers 49: 431–442
40. Witsgall C (1975) Mathematical methods of site selection for electronic system (EMS). NBS Internal Report