A NEW MODELING AND SOLUTION APPROACH FOR THE NUMBER PARTITIONING PROBLEM

BAHRAM ALIDAEE, FRED GLOVER, GARY A. KOCHENBERGER, AND CESAR REGO

Received 24 November 2003 and in revised form 9 May 2004

The number partitioning problem has proven to be a challenging problem for both exact and heuristic solution methods. We present a new modeling and solution approach that consists of recasting the problem as an unconstrained quadratic binary program that can be solved by efficient metaheuristic methods. Our approach readily accommodates both the common two-subset partition case as well as the more general case of multiple subsets. Preliminary computational experience is presented illustrating the attractiveness of the method.

1. Introduction

The number partitioning problem (NPP), being one of Garey and Johnson's [3] six basic NP complete problems, has been the subject of considerable research in recent years. In the simplest case, the problem consists of partitioning a set of numbers into two subsets such that the sums of the numbers in each subset are as close as possible. The more general case of NPP seeks to partition the original set into n subsets ($n > 2$) such that the sums for each set are as close to each other as possible. Most of the work reported in the literature has been directed to the two-subset case. This problem is known to be computationally challenging and, except for small instances, is most productively approached by heuristic means.

The literature contains several papers describing methods for solving NPP. One of the most widely referenced methods is the *differencing method* (DM) of Karmarkar and Karp [11] which is $O(n\log n)$ for the two-subset case. Several computational studies, including those of Johnson et al. [10] and Arguello et al. [1], report that DM outperforms, in terms of both solution quality and computation time, alternative methods such as simulated annealing and GRASP. Due to it's effectiveness, Korf [13] employs DM to produce initial solutions for an improvement method that is reported to be very successful on a large variety of partitioning problems. An analysis of various published methods for NPP can be found in the paper by Gent and Walsh [4].

In this paper, we present a new way of modeling and solving a variety of NPPs. Several versions of the NPP, each a generalization of the one preceding it, are considered. We

DOI: 10.1155/JAMDS.2005.113

show that each version of NPP can be cast into the common modeling framework of the unconstrained quadratic binary program (UQP). In turn, this common formulation enables solutions by recently developed metaheuristics for this model.

The UQP can be written in the form

$$\text{UQP}: \min f(x) = xQx, \tag{1.1}$$

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 number of important problems. A discussion of a wide variety of applications can be found in Kochenberger et al. [12].

Our purpose here is to show how this versatile model can be used to model and solve NPPs. In the sections below, we illustrate the use of UQP for various NPPs. Preliminary computational experience is presented for the two-subset case.

2. The two-subset case

The most common version of NPP involves partitioning a set of numbers into two subsets such that the subset sums are as close to each other as possible. We model this problem as follows.

Consider a set of numbers $S = \{s_1, s_2, s_3, \ldots, s_m\}$. The goal is to partition S into two subsets such that the subset sums are as close to each other as possible. Let $x_j = 1$ if s_j is assigned to subset 1, 0 otherwise. Then sum_1, subset 1's sum, is $\text{sum}_1 = \sum_{j=1}^{m} s_j x_j$ and the sum for subset 2 is $\text{sum}_2 = \sum_{j=1}^{m} s_j - \sum_{j=1}^{m} s_j x_j$. The difference in the sums is then given by

$$\text{diff} = \sum_{j=1}^{m} s_j - 2\sum_{j=1}^{m} s_j x_j = c - 2\sum_{j=1}^{m} s_j x_j. \tag{2.1}$$

We approach the goal of minimizing the absolute value of diff by minimizing

$$\text{diff}^2 = \left\{c - 2\sum_{j=1}^{m} s_j x_j\right\}^2 = c^2 + 4xQx, \tag{2.2}$$

where

$$q_{ii} = s_i(s_i - c), \qquad q_{ij} = s_i s_j. \tag{2.3}$$

Dropping the additive and multiplicative constants, our optimization problem becomes simply

$$\text{UQP}: \min xQx. \tag{2.4}$$

As a foundation for applying the UQP model to solve NPP, we first review solution methodologies created for UQP.

2.1. Solution approaches for 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. These various papers approach UQP by branch and bound, decomposition, semidefinite programing and cutting planes, tabu search, simulated annealing, evolutionary methods such as genetic algorithms and scatter search, as well as simple one-pass heuristic methods. Each of these approaches exhibits some degree of success and could in principle be utilized to solve problems reformulated as UQP. However, the exact methods degrade rapidly with problem size, and have meaningful application to general UQP problems with no more than a few hundred variables. For larger problems, heuristic methods are required.

Below, we highlight our tabu search heuristic [5, 6] which has proven to be very successful on a wide variety of UQP instances and that was used to produce the computational results presented later in the paper. Reference [12] gives an overview of other solution approaches for UQP.

2.2. 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. The search process is typically allowed to run for a preset 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 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 by Glover et al. [6].

Example 2.1. We illustrate the approach on a simple example. Consider the set of 8 numbers

$$S = \{25, 7, 13, 31, 42, 17, 21, 10\}. \tag{2.5}$$

Following the development above, we have that $c^2 = 27,556$ and the equivalent UQP

Table 2.1. Subset sum differences and times.

ID	Sum difference via DM	DM time	Sum difference via xQx	xQx time
NP25.1	30	$< 1s$	0	$< 1s$
NP25.2	36	$< 1s$	0	$< 1s$
NP25.3	35	$< 1s$	1	$< 1s$
NP25.4	28	$< 1s$	0	$< 1s$
NP25.5	26	$< 1s$	0	$< 1s$
NP75.1	23	$< 1s$	1	$1s$
NP75.2	31	$< 1s$	1	$1s$
NP75.3	25	$< 1s$	1	$1s$
NP75.4	24	$< 1s$	0	$1s$
NP75.5	23	$< 1s$	1	$1s$

problem is $\min x_0 = xQx$ with

$$Q = \begin{bmatrix} -3525 & 175 & 325 & 775 & 1050 & 425 & 525 & 250 \\ 175 & -1113 & 91 & 217 & 294 & 119 & 147 & 70 \\ 325 & 91 & -1989 & 403 & 546 & 221 & 273 & 130 \\ 775 & 217 & 403 & -4185 & 1302 & 527 & 651 & 310 \\ 1050 & 294 & 546 & 1302 & -5208 & 714 & 882 & 420 \\ 425 & 119 & 221 & 527 & 714 & -2533 & 357 & 170 \\ 525 & 147 & 273 & 651 & 882 & 357 & -3045 & 210 \\ 250 & 70 & 130 & 310 & 420 & 170 & 210 & -1560 \end{bmatrix}. \tag{2.6}$$

Solving with our tabu search heuristic gives $x = (0,0,0,1,1,0,0,1)$ for which $x_0 = -6889$ yielding perfectly matched sums of 83.

2.3. Computational experience. The standard comparison in the literature for new approaches to the NPP is with the differencing method (DM) of Karmarker and Karp [11], which has proven to be both fast and effective in several comparative studies. Due to its prominence in the literature, we include it here as a benchmark as well. The results reported here are on modest-sized random problems of size $m = 25$ and $m = 75$. Five instances of each size are considered with the elements drawn randomly from the interval (50,100). Each of the 10 problems was solved by our UQP approach as well as the method of Karmarker and Karp. The results are shown in Table 2.1. For each problem, our tabu search heuristic was run for 20 "span" cycles.

The solutions shown in Table 2.1 indicate that our method dominates DM in terms of solution quality. For five of the problem instances, our approach found partitions with equal sums (differences of 0). For the other five problems, our method produced partitions whose sums differed by only 1. We suspect that these later results are optimal as well.

In contrast to this, DM was unable to produce equal-sum partitions for any of the problems, yielding solutions with unequal sums of substantial margins across all ten problems. Solution times, shown in seconds on a Pentium 333 laptop, are roughly the same for both methods.

3. Partitioning with multiple subsets

The problem of the previous section can be generalized to accommodate $n > 2$ partitions. As before, the goal is to assign numbers to subsets such that the subset sums are as close to each other as possible. For this more general case, we start with a constrained model containing assignment equations ensuring that each number is assigned to one of the subsets. This constrained model is then recast into the form of UQP by introducing quadratic infeasibility penalties into the objective function as an alternative to the explicit imposition of the assignment constraints. This approach, shown by Kochenberger et al. [12] to be very successful in a wide variety of other problem classes, is presented below.

As before, we start with a set of m numbers $S = \{s_1, s_2, \dots, s_m\}$ to be partitioned into n subsets. Let $x_{ij} = 1$ if element s_i is assigned to subset j, 0 otherwise.

The sum for subset j (sum_j) is given by

$$\text{sum}_j = \sum_{i=1}^{n} s_i x_{ij}, \tag{3.1}$$

and we seek to minimize

$$x_0 = (\text{sum}_1 - \text{sum}_2)^2 + (\text{sum}_1 - \text{sum}_3)^2 + \cdots + (\text{sum}_{m-1} - \text{sum}_m)^2 \tag{3.2}$$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, m. \tag{3.3}$$

Since the objective of (3.2) is quadratic in the assignment variables, we can rewrite (3.2) as $x_0 = xQx$, where $x = (x_{11}, x_{12}, \dots, x_{1n}, x_{21}, \dots, x_{mn})$ is the binary vector of length m by n and Q is the square, symmetric matrix resulting from combining terms in (3.2). Our model is now of the form

$$\min x_0 = xQx \tag{3.4}$$

subject to

$$Ax = b, \quad x \text{ binary}. \tag{3.5}$$

This constrained model, following the general reformulation put forth by Hammer and Rudeanu [7], can then be recast as an unconstrained quadratic model by imposing the constraints implicitly via quadratic penalties added to the objective function. Specifically, for a positive scalar P, we have

$$x_0 = xQx + P(Ax - b)^t(Ax - b) = xQx + xDx + c = x\hat{Q}x + c, \tag{3.6}$$

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

$$\mathrm{UQP(PEN)} : \min x\hat{Q}x, \quad x \text{ binary}, \tag{3.7}$$

and we see that the multiple-subset case considered in this section, like the two-subset case of Section 2, can be modeled by UQP. A suitable choice of the scalar penalty P, for the general application of this reformulation approach, can always be chosen so that the optimal solution to UQP(PEN) is the optimal solution to the original constrained problem. (Hammer and Rudeanu [7], Hansen [8], Hansen et al. [9], and Boros and Hammer [2]). We illustrate the procedure in the following example.

3.1. Multiple-subset examples. Consider the set of numbers $S = \{15,3,7,11,7,5,21,9,13,7,5,15,23,14,13,13,27,15,9,9,17,10,11,19,8\}$ to be partitioned into subsets of size $n = 3,4$, and 5 such that their subset sums are as close as possible (i.e., 3 separate problems).

For each case, the transformation to $x\hat{Q}x$ is accomplished using an arbitrarily chosen penalty value of $P = 900$. Solving UQP for each case results in the assignments shown in Table 3.1. For each case, the numbers in the "subset assignment" column are the subset indices of the assignments made. For example, for the $n = 3$ case, the first number (15) is assigned to subset 3, the second number (3) is assigned to subset 1, and so forth. The assignments shown are very well balanced with subset sums of 102, 102, and 102, respectively for the $n = 3$ case, 76, 76, 77, and 77 for the $n = 4$ case, and 62, 61, 61, 61 and 61 for the $n = 5$ case.

The results given in Table 3.1 show the assignment of all elements (numbers), confirming that the penalty used was in fact sufficiently large to yield feasible assignments. Note that the UQP problems solved to produce the results of Table 3.1 were of size 75 variables, 100 variables, and 125 variables, respectively. The largest of these were solved in less than 2 seconds on a Pentium 333 laptop by our tabu search heuristic [6].

4. Nonhomogeneous case

To motivate this section, consider a machine-loading problem where a set of m jobs must be assigned to n machines such that the aggregate loading of the machines is as level as possible. If the machines are identical, job times are independent of the assignments made and the problem can be correctly modeled and solved by the representation given in the previous section.

A more general machine-loading problem, however, would allow for nonidentical machines and machine-dependent job times. The development of the previous section can easily be modified to accommodate this more general problem setting as indicated below.

Our problem is to level the loading of m jobs assigned to n machines where t_{ij} is equal to the time required to accomplish job i on machine j. Following our earlier development, let $x_{ij} = 1$ if job i is assigned to machine j, 0 otherwise. Then the total time (load) of the

Table 3.1. Multiple-subset results.

Index number	Number	Subset assignment		
		$n = 3$ case	$n = 4$ case	$n = 5$ case
1	15	3	2	2
2	3	1	3	1
3	7	3	1	3
4	11	2	2	1
5	7	1	3	5
6	5	1	4	2
7	21	2	3	1
8	9	2	2	4
9	13	2	2	3
10	7	3	4	3
11	5	3	1	5
12	15	3	3	4
13	23	3	4	3
14	14	2	4	5
15	13	3	3	5
16	13	1	4	5
17	27	1	1	4
18	15	2	4	2
19	9	2	2	2
20	9	1	1	5
21	17	3	1	2
22	10	2	3	4
23	11	1	1	3
24	19	1	2	1
25	8	1	3	1

jobs assigned to machine j is

$$T_j = \sum_{i=1}^{m} t_{ij}x_{ij}, \quad j = 1, n, \tag{4.1}$$

and we want to minimize the aggregate squared deviation,

$$T_0 = (T_1 - T_2)^2 + (T_1 - T_3)^2 + \cdots + (T_{n-1} - T_n)^2 \tag{4.2}$$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, m. \tag{4.3}$$

By exactly the development of the previous section, this model can be recast into the form of $\min x\hat{Q}x$. Note that for the machine-loading problem context considered here, we may have additional constraints that job assignments must satisfy in addition to simply ensuring that each job gets assigned. Such constraints, provided that they are linear, can be "folded" into the $\hat{Q}$ matrix via additional quadratic penalties to once again yield an equivalent representation in the form of $x\hat{Q}x$.

4.1. Related applications. Our focus in the development given above was to indicate how various versions of the NPP seeking equal subset sums could be modeled and solved via the common UQP framework. Additional partitioning problems, closely related to those considered above, can also be productively addressed by this approach. For instance, consider the problem where each subset has a predetermined *target* sum, tar_j, and we seek a partition of the original set into n subsets that minimizes the aggregate deviation from these target values.

Utilizing sum_j and x_{ij} as defined earlier, we have the problem

$$\min x_0 = (\text{sum}_1 - \text{tar}_1)^2 + (\text{sum}_2 - \text{tar}_2)^2 + \cdots + (\text{sum}_n - \text{tar}_n)^2 \tag{4.4}$$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, m, \tag{4.5}$$

which by the developments illustrated earlier can be readily recast into UQP.

5. Summary

In this paper, we have introduced the unconstrained quadratic program (UQP) as a new and fruitful representation of various forms of the NPP. This model (xQx) is shown to be robust in its ability to accommodate problem variations, affording an opportunity to solve any or all problem versions by a single solution method.

In addition to introducing a new model for this class of problems, we have presented preliminary computational experience demonstrating the attractiveness of our tabu search method for solving NPPs via the xQx representation. Our outcomes show that this approach is both conceptually and computationally attractive.

References

[1] M. Arguello, T. Feo, and O. Goldschmidt, *Randomized methods for the number partitioning problem*, Comput. Oper. Res. **23** (1996), no. 2, 103–111.

[2] E. Boros and P. L. Hammer, *Pseudo-boolean optimization*, Discrete Appl. Math. **123** (2002), no. 1-3, 155–225.

[3] M. R. Garey and D. S. Johnson, *Computers and Intractability. A Guide to the Theory of NP-Completeness*, A Series of Books in the Mathematical Sciences, W. H. Freeman and Company, California, 1979.

[4] I. P. Gent and T. Walsh, *Analysis of heuristics for number partitioning*, Comput. Intelligence **14** (1998), no. 3, 430–451.

[5] F. Glover, G. Kochenberger, and B. Alidaee, *Adaptive memory tabu search for binary quadratic programs*, Management Sci. **44** (1998), no. 3, 336–345.

[6] F. Glover, G. Kochenberger, B. Alidaee, and M. M. Amini, *Tabu search with critical event memory: an enhanced application for binary quadratic programs*, Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization (Sophia-Antipolis, 1997) (S. Voss, S. Martello, I. Osman, and C. Roucairol, eds.), Kluwer Academic, Massachusetts, 1999, pp. 93–109.

[7] P. L. Hammer and S. Rudeanu, *Boolean Methods in Operations Research and Related Areas*, Econometrics and Operations Research, vol. 7, Springer-Verlag, New York, 1968.

[8] P. B. Hansen, *Methods of nonlinear* $0-1$ *programming*, Ann. Discrete Math. **5** (1979), 53–70.

[9] P. B. Hansen, B. Jaumard, and V. Mathon, *Constrained nonlinear* $0-1$ *programming*, ORSA J. Comput. **5** (1993), no. 2, 97–119.

[10] D. Johnson, C. Aragon, L. McGeoh, and C. Schevon, *Optimization by simulated annealing: an experimental evaluation; part II, graph coloring and number partitioning*, Oper. Res. **39** (1991), no. 3, 378–406.

[11] N. Karmarkar and R. Karp, *The differencing method of set partitioning*, Tech. Rep. UCB/CSD 82/113, Computer Science Division, University of California, California, 1982.

[12] G. Kochenberger, F. Glover, B. Alidaee, and C. Rego, *An unconstrained quadratic binary programming approach to modeling and solving combinatorial optimization problems*, University of Colorado Working Paper, University of Colorado, Colorado, 2002.

[13] R. E. Korf, *A complete anytime algorithm for number partitioning*, Artificial Intelligence **106** (1998), no. 2, 181–203.

Bahram Alidaee: Hearin Center for Enterprise Science, School of Business Administration, University of Mississippi, MS 38677, USA
E-mail address: balidaee@bus.olemiss.edu

Fred Glover: Leeds School of Business, University of Colorado at Boulder, Boulder, CO 80309-0419, USA
E-mail address: fred.glover@colorado.edu

Gary A. Kochenberger: School of Business, University of Colorado at Denver and Health Sciences Center, Denver, CO 80217-3364, USA
E-mail address: gkochenberger@cudenver.edu

Cesar Rego: Hearin Center for Enterprise Science, School of Business Administration, University of Mississippi, MS 38677, USA
E-mail address: crego@bus.olemiss.edu