Research Note

Fast 2-flip move evaluations for binary unconstrained quadratic optimization problems

Fred Glover[1] and Jin-Kao Hao[2]

[1]OptTek Systems, Inc.
1919 Seventh Street
Boulder, CO 80302, USA
glover@opttek.com

[2]Laboratoire d'Etude et de Recherche en Informatique (LERIA)
Université d'Angers
2 Boulevard Lavoisier
49045 Angers Cedex 01, France
jin-kao.hao@univ-angers.fr

Abstract

We provide a method for efficiently evaluating 2-flip moves that simultaneously change the values of two 0-1 variables in search methods for binary unconstrained quadratic optimization problems (UQP). We extend a framework recently proposed by the authors for creating efficient evaluations of 1-flip moves to yield a method requiring significantly less computation time than a direct sequential application of 1-flips. A Tabu Search algorithm that combines 1-flip and 2-flip moves, in a study currently in process, has made use of this extension to produce very competitive results on some UQP benchmark instances.

Keywords: zero-one optimization; unconstrained quadratic programming; metaheuristics; computational efficiency; 2-flip move; tabu search.

Corresponding author: Fred Glover

Biographical notes: Fred Glover holds the title of Distinguished Professor at the University of

Colorado and is Chief Technology Officer for OptTek Systems, Inc. He has authored or co-authored more than 370 published articles and eight books in the fields of mathematical optimization, computer science and artificial intelligence. Dr. Glover is the recipient of the distinguished von Neumann Theory Prize, an elected member of the National Academy of Engineering, and has received honorary awards and fellowships from the American Association for the Advancement of Science (AAAS), the NATO Division of Scientific Affairs, the Miller Institute of Basic Research in Science and numerous other organizations.

Dr. Jin-Kao Hao holds a full Professor position in the Computer Science Department of the University of Angers (France) and is currently the Director of the LERIA Laboratory. His research lies in the design of effective heuristic and metaheuristic algorithms for solving large-scale combinatorial search problems. He is interested in various application areas including bioinformatics, telecommunication networks and transportation. He has co-authored more than one hundred peer-reviewed publications in international journals, book chapters and conference proceedings.

---

1 Introduction

The binary unconstrained quadratic programming problem may be written

$$\text{UQP: Minimize } x_o = xQx \qquad (1)$$
$$x \text{ binary}$$

where Q is an n by n matrix of constants and x is an n-vector of binary (zero-one) variables. The formulation UQP is notable for its ability to represent a wide range of important problems, including those from social psychology (Harary, 1953), financial analysis (Laughunn, 1970; McBride and Yormak, 1980), computer aided design (Krarup and Pruzan, 1978), traffic management (Gallo, Hammer, and Simeone, 1980; Witsgall, 1975), machine scheduling (Alidaee, Kochenberger, and Ahmadian, 1994), cellular radio channel allocation (Chardaire and Sutter, 1994) and molecular conformation (Phillips and Rosen, 1994). Moreover, many 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 as documented in papers of Pardalos and Rodgers (1990), and Pardalos and Xue (1994). A review of additional applications and formulations can be found in Kochenberger et al. (2004).

In the paper of Glover and Hao (2009), we exposed a method for efficiently evaluating 1-flip moves that change the value of a single 0-1 variable for the UQP problem. In this note, we extend this method to provide fast evaluations for 2-flip moves that change the values of two 0-1 variables.

While the 1-flip move is frequently employed by many search algorithms for the UQP problem, the 2-flip move constitutes an interesting and complementary alternative. The availability of an accelerated method for these 2-flip moves offers new possibilities for creating methods that seek to exploit various combined neighborhoods, allowing the search to explore more candidate solutions within a given time limit. As a consequence of our analysis we also show how to conveniently evaluate 3-flip and higher order moves, although the efficient update we present for 2-flip moves becomes more complex and requires more effort for these more advanced moves.

2 Basic notions of 2-flip moves

We assume the problem is represented by storing Q as a lower triangular matrix as in Glover and Hao (2009). Starting from the 1-flip perspective, let x' and x'' represent two binary solutions where x'' is obtained from x' by flipping the value of a single variable x_i from 0 to 1 or from 1 to 0 (according to whether $x_i = x_i'$ is 0 or 1). Define $x_o' = x'Qx'$ and $x_o'' = x''Qx''$. Then the objective function change produced by flipping x_i, given by $\Delta_i = x_o'' - x_o'$, discloses whether the move that replaces x' by x'' will cause x_o to improve or deteriorate (respectively, decrease or increase) relative to the minimization objective.

As observed in Glover and Hao (2009), Δ_i can be expressed as

$$\Delta_i = (x_i'' - x_i')\,(Q_{ii} + \sum(Q_{(i,h)}\, x_h') : h \in N - \{i\})) \qquad (2)$$

where the notation $Q_{(i,h)}$ refers to Q_{ih} if $i > h$ and to Q_{hi} if $h > i$. A very efficient update of Δ_i exists by storing and updating the summation term in (2), differentiating its two components for $h < i$ and $h > i$.

In the case of a 2-flip neighborhood, we are interested in the change in x_o that results by flipping 2 variables, x_k and x_j, and will refer to this objective function change by Δ_{kj}. Although we will provide a method that is more efficient than evaluating pairs of 1-flips in succession, it is convenient to consider the 2-flip process from a sequential perspective. Accordingly, we assume $k > j$ and refer to x_k and x_j respectively as the "first" and "second" variables flipped, as a basis for determining Δ_{kj}. Moreover, for a given x_k we will consider all variables x_j, $j < k$, that may be flipped together with x_k, and undertake to identify the best of these flipping companions, whose index we denote by $j^* = j(k)$ (a function of k).

Analysis and illustration

We accompany the analysis underlying our derivation with an example, to make the ideas clearer. In the process we will also introduce additional useful notation. Since Q is stored as a lower triangular matrix, we define $N_k = \{j \in N:\ j < k \text{ and } Q_{kj} \neq 0\}$. The set N_k corresponds to the "column indexes" associated with row k in the 1-flip evaluation method, which are accessed by a linked list in performing 1-flip updates as shown in Glover and Hao (2009).

We also introduce a term λ_{kj} which is defined as a function of the values x_k' and x_j', as follows:

For $x_k' = 0$:

$$\lambda_{kj} = \Delta_j + Q_{kj} \quad \text{if } x_j' = 0$$
$$\lambda_{kj} = \Delta_j - Q_{kj} \quad \text{if } x_j' = 1$$

(3)

For $x_k' = 1$:

$$\lambda_{kj} = \Delta_j - Q_{kj} \quad \text{if } x_j' = 0$$
$$\lambda_{kj} = \Delta_j + Q_{kj} \quad \text{if } x_j' = 1$$

(Equivalently, $\lambda_{kj} = \Delta_j + \delta_{kj}Q_{kj}$, where $\delta_{kj} = 1$ if $x_k' = x_j'$ and $\delta_{kj} = -1$ if $x_k' \neq x_j'$.)

Fundamental Relationship:

The value Δ_{kj} (= $x''Qx'' - x'Qx'$) that gives the change in x_o when x_k and x_j are both flipped is given by

$$\Delta_{kj} = \Delta_k + \lambda_{kj} \qquad (4)$$

Justification: After first flipping x_k and changing x_o by the amount Δ_k, the result of additionally flipping x_j can be given by (2) for j = i, provided we first update the portion of (2) that refers to x_k' by replacing x_k' with x_k''. The unique term changed is $Q_{(j,k)}\,x_k'$, or simply $Q_{kj}x_k'$ for k > j, which modifies (2) by adding the term λ_{kj} as identified in (4) above.

To illustrate how we may take advantage of (3) and (4), suppose k = 17 (i.e., the "first" variable flipped is x_{17}), and assume the full set of cross product terms in which x_{17} appears with a non-zero coefficient is given by:

$$x_{17}\,(6x_2 - 7x_5 + 11x_6 - 12x_8 + 12x_9)$$

Then $N_{17} = \{2, 5, 6, 8, 9\}$, and the non-zero coefficients Q_{kj} for k = 17 and $j \in N_k$ are listed in

Table 1 below, together with the current values x_j' of the associated variables x_j. The table also shows the single flip evaluations Δ_j for each of these variables and we assume for the illustration that $x_{17}' = 0$ and $\Delta_{17} = 3$. The λ_{kj} values shown in Table 1 can then be verified to result from the formula given in (3) and (4) under these assumptions.

j =	2	5	6	8	9
Q_{kj} =	6	– 7	11	– 12	12
x_j' =	0	0	1	0	1
Δ_j =	1	5	4	4	2
λ_{kj} =	7	– 2	– 7	– 8	– 10

Table 1

By our argument that justifies (4), we may interpret the value λ_{kj} as the amount of change that will be produced in the quantity Δ_j if we first flip x_k and then flip x_j. This interpretation can be readily confirmed by reference to the values given in the example. To see this, consider again the expression

$$x_{17}\,(6x_2 - 7x_5 + 11x_6 - 12x_8 + 12x_9).$$

If x_{17} is flipped to change $x_{17}' = 0$ to $x_{17}' = 1$ then by additionally flipping x_2 from 0 to 1 the preceding expression makes it clear that $Q_{17,2} = 6$ units will be added to Δ_2, thus increasing the change in x_o by 6 units beyond the change produced by flipping x_2 itself. Similarly, if the flip of x_{17} is followed by additionally flipping x_5 then $Q_{17,5} = -7$ units will be added to Δ_5, in this instance reducing the value of x_o by 7 units beyond the change produced by flipping x_5. In the case of flipping x_6, whose current value is given by $x_6' = 1$, examination of the preceding expression discloses that $Q_{17,6} = 11$ units will be subtracted (rather than added) to produce the additional change in x_o. The interpretation of the rest of the λ_{kj} values follows similarly.

Based on the foregoing analysis, we conclude that for a given index k, the smallest of the λ_{kj} values (under the objective of minimizing x_o) identifies the best variable x_{j^*} to flip together with x_k. Hence in the case of our example $j^* = 9$, yielding the smallest λ_{kj} value $\lambda_{k9} = -10$. Note that this value can be obtained from the calculation of (3) without bothering to compute Δ_{kj} from (4).

To determine whether the 2-flip produced by flipping both x_k and x_{j^*} (here x_{17} and x_9) is negative and hence profitable, we complete the step given in (4) by now adding the value of $\Delta_{17} = 3$ to λ_9, yielding a final value of $\Delta_{17,9} = -7$, which verifies that the 2-flip will indeed improve x_o.

(It may incidentally be noted that none of the 1-flips were profitable in this example – i.e., Δ_k and all Δ_j values were positive – but several of the 2-flips yield negative Δ_{kj} values, disclosing that they improve x_o.)

3 Generating and updating the 2-flip evaluations: implications for computation

The first step for producing fast 2-flip calculations is evidently to generate and update the Δ_k values by the rules in Glover and Hao (2009), and for each Δ_k to calculate the λ_{kj} values, $j \in N_k$, as illustrated in the preceding section. These λ_{kj} values are then maintained so that they may simply be looked up in the process of evaluating the 2-flip moves.

We can improve this process when Q is sparse, however, by employing the following approach. Instead of recalculating the λ_{kj} values after each 2-flip move, we keep a special record that makes it possible to merely update the relevant subset of λ_{kj} values that change from one iteration to the next. This updating process can be performed as follows.

Together with recording the λ_{kj} values for each $k \in N$ and for $j \in N_k$, we also identify the associated index j^* (= $j(k)$) = arg min(λ_{kj}, $j \in N_k$) and the value $v_k = \lambda_{kj^*}$. The best 2-flip that can be obtained when x_k is the first flip variable will change x_o by the amount

$$\Delta_{kj^*} = \Delta_k + v_k. \qquad (5)$$

By this relationship, the 2-flip evaluation can be made by simply scanning the indexes $k \in N$ and performing the calculation of (5) for each. The structure of the lower triangular matrix of course implies that we only calculate a 2-flip evaluation once for each k and j (for $k > 1$), rather than calculating it for both of the pairs (k,j) and (j,k).

Effects of the update calculations

In brief, to carry out this accelerated updating process for 2-flip evaluations we must increase the memory used by the 1-flip evaluations to additionally store the λ_{kj} values. These can be recorded and accessed in the same way as the Q_{kj} values (hence effectively doubling the memory devoted to storing the sparse matrix Q in lower triangular form). The two n-dimensional arrays for identifying $j^* = j(k)$ (for each k) and the value v_k are also added.

This added memory makes it possible to evaluate all 2-flips with only a little more than twice the effort required to evaluate all 1-flips. This represents an appreciable savings in computational effort by comparison with the approximate squaring of the effort required to examine 1-flips that a more direct 2-flip implementation would entail.

We could in fact effectively perform the analysis that identifies the j* and v_k ($=\lambda_{kj*}$) values without having to store the full set of updated λ_{kj} values, and simply rely on the formulas (3) and (4) to derive the final evaluations. This adds some computational effort, but can be used as an alternative to save additional memory if Q is dense. We also observe that in the case of a tabu search implementation we need to determine an additional instance of the winning index j* = j(k) and the associated value v_k, derived from the λ_{kj} values by restricting attention to those variables x_j that are not tabu. (The ordinary "unrestricted" derivation remains relevant for the case of an aspiration level that may overrule a variable's tabu status.) This can be done without doubling the effort, since the "tabu search j*" can be determined on the same pass of the indexes $j \in N_k$ (for non-zero Q_{kj}'s) used to update the λ_{kj} values and determine the unrestricted j*.

4 Extensions to 3-flip (and higher order) moves

The type of "incremental" analysis used in identifying the change in x_o represented by the value Δ_{kj} for 2-flip moves can readily be extended to identify an analogous value Δ_{kjp} for 3-flip moves, where we assume k > j > p. In particular, supposing that we have first flipped x_k and x_j to obtain the change Δ_{kj} identified in (4), we conclude by flipping x_p and determine its effect by considering the result of applying (2) for i = p, subject to first modifying the expression (2) to account for the changes induced by having replaced x_k' and x_j' by x_k'' and x_j''.

We note that the terms of (2) affected by the change in x_k and x_j are precisely those of $Q_{kp}x_k'$ and $Q_{jp}x_j'$, and hence the result is to create a modified instance of Δ_p given by

$$\lambda_{kjp} = \Delta_p + Q_{kp}(x_k'' - x_k') + Q_{jp}(x_j'' - x_j')$$

or equivalently

$$\lambda_{kjp} = \Delta_p + \delta_{kp}Q_{kp} + \delta_{jp}Q_{jp}$$

for δ_{kp} and δ_{jp} given as in the expression for δ_{kj} following (3). In sum, then the value Δ_{kjp} is given by

$$\Delta_{kjp} = \Delta_{kj} + \lambda_{kjp.}$$

This same incremental analysis can be easily applied to higher order moves. For example, in the case of 4-flips the evaluations have the form

$$\Delta_{kjpq} = \Delta_{kjp} + \lambda_{kjpq}$$

and the general pattern is apparent. To exploit such evaluations without excessive computational effort, candidate list strategies of the type proposed with tabu search can be employed. (See, e.g., Glover and Laguna (1997).)

5 Concluding remarks

Moves consisting of 1-flips and 2-flips moves evidently define different and complementary neighborhoods. Although these moves can independently be employed in search processes, a combination of them may lead to more effective search. This is exemplified in one of our ongoing studies where 1-flip and 2-flip moves are made probabilistically within an algorithm based on tabu search. Combined with an extended memory-based diversification strategy, this joint use of both moves allows the algorithm to reach highly competitive results on a set of benchmark instances from the literature. More specifically, testing on the nine instances of UQP transformed from the Set-Partitioning Problem (Lewis, Kochenberger, and Alidaee, 2008), this algorithm improves three previous best objective values while equaling the best solution for the remaining instances. The same algorithm using only 1-flip move fails to match the previous results under the same testing conditions.

Finally, we note that there exist several other ways to establish combined neighborhoods from 1-flip and 2-flip moves, as described in Lü, Glover, and Hao (2009). For instance, a search algorithm can explore the two moves in a circular (token ring) manner, or employ the union of both moves. It may also switch between the moves using a systematic strategy that incorporates learning, rather then simply using a probabilistic criterion for switching. A more thorough analysis of these alternatives, as well as those that arise by applying our incremental analysis to more complex moves, provides interesting directions for future research.

Acknowledgment: We are indebted to a reviewer whose insightful comments have improved the presentation of our paper. This work is partially supported by a "Chaire d'excellence" from "Pays de la Loire" Region (France)" and regional MILES (2007-2009) and RaDaPop projects (2009-2011).

References:

Alidaee, B, G. Kochenberger, and A. Ahmadian (1994) "0-1 Quadratic Programming Approach for the Optimal Solution of Two Scheduling Problems," *International Journal of Systems Science*, 25, 401-408.

Chardaire, P, and A. Sutter (1994) "A Decomposition Method for Quadratic Zero-One Programming," *Management Science*, 4 1(4), 704-712.

Gallo, G, P. Hammer, and B. Simeone (1980), "Quadratic Knapsack Problems," *Mathematical Programming*, 12, 132-149.

Glover F, and J.K. Hao (2009) Efficient Evaluations for Solving large 0-1 Unconstrained Quadratic Optimization Problems. *International Journal of Metaheuristics,* 1(1).

Glover, F, and M. Laguna (1997) *Tabu Search*, Kluwer Academic Publishers.

Harary, F (1953) "On the Notion of Balanced of a Signed Graph," *Michigan Mathematical Journal*, 2, 143-146.

Kochenberger, G., F. Glover, B., Alidaee and C. Rego (2004) "A Unified Modeling and Solution Framework for Combinatorial Optimization Problems," *OR Spectrum,* 26, 237-250.

Krarup, J, and A. Pruzan (1978) "Computer Aided Layout Design," *Mathematical Programming Study*, 9, 75-94.

Laughunn, D.J. (1970) "Quadratic Binary Programming," *Operations Research*, 14, 454-461.

Lewis, M., G., Kochenberger, B. Alidaee (2008) "A New Modeling and Solution Approach for the Set-Partitioning Problem," *Computers and Operations Research*, 35(3), 807-813.

Lü, Z., F., Glover, and J.K., Hao (2009) "Neighborhood Combination for Unconstrained Binary Quadratic Programming," Proceedings of the 8th Metaheuristic International Conference (*MIC 2009*), Hamburg, Germany, 13-16 July, *2009*.

McBride, R.D., and J.S. Yormack (1980) "An Implicit Enumeration Algorithm for Quadratic Integer Programming," *Management Science*, 26, 282-296.

Pardalos, P, and G.P. Rodgers (1990) "Computational Aspects of a Branch and Bound Algorithm for Quadratic Zero-one Programming," *Computing,* 45, 131-144.

Pardalos, F, and J. Xue (1994) "The Maximum Clique Problem," The Journal of Global Optimization, 4, 301-328.

Phillips, A.T., and J.B. Rosen (1994) "A Quadratic Assignment Formulation of the Molecular Conformation Problem," *Journal of Global Optimization*, 4, 229-241.

Witsgall, C. (1975) "Mathematical Methods of site Selection for Electronic System (EMS)," NBS Internal Report.