

# **xQx as a Modeling and Solution Framework for the Linear Ordering Problem**

**Mark Lewis<sup>a</sup>, Bahram Alidaee<sup>b</sup>, Fred Glover<sup>c</sup>, Gary Kochenberger<sup>d,\*</sup>**

<sup>a</sup> School of Professional Studies, Missouri Western State University, St. Joseph, MO

<sup>b</sup> Hearin Center for Enterprise Science, School of Business, University of Mississippi, University, MS

<sup>c</sup> Leeds School of Business, University of Colorado, Boulder, CO

<sup>d</sup> School of Business, University of Colorado, Denver, CO

(September, 2005)

**Abstract.** This paper illustrates how large instances of the linear ordering problem can be effectively modeled and solved as unconstrained quadratic binary programs (UQP). Computational experience comparing a basic tabu search code for UQP to the state-of-the-art commercial code (CPLEX) demonstrates the viability and attractiveness of our approach.

**Key words:** Linear ordering; integer programming; metaheuristics

**MSC Classification:** 90C10, 90C20, 90C59

\* Corresponding author.

*Email addresses:* mlewis14@missouriwestern.edu, balidaee@bus.olemiss.edu, fred.glover@colorado.edu, Gary.kochenberger@cudenver.edu

---

## **1. Introduction**

The linear ordering problem (LOP) is well-known for its diversity of application as well as its computational challenge. This important problem, which is known to be NP-hard, has attracted considerable attention in the literature. In this paper we present a general method for solving 0-1 integer linear programs, apply this general approach to LOP and show its attractiveness by comparison to an industry standard commercial solver.

Applications of the linear ordering model have been reported in a wide variety of problem settings including economics, logistics, social sciences, and discrete

mathematics (e.g. see Laguna and Martí (2003), Mitchell and Borchers (2000), Mitchell, et al. (1998), Steiner and Stephenson (2000)). The most oft-quoted example is from economics where the problem concerns the triangulation of input-output matrices as seen in Grötschel, et al. (1984), Mitchell and Borchers (2000). Linear ordering applications typically involve the ranking of pair-wise comparisons as used in analytical hierarchical processing, with LOP providing an alternative method of analyzing the preference data that maximizes consistency (see Laguna, et al. (1999)). Applications in bioinformatics are seen in Bar-Joseph, et al. (2001) and Biedl, et al. (2001), which discuss the optimal ordering of leaves in binary trees resulting from the hierarchical clustering of genes. A potential application in web site design, see for example, Cooley (2003) maximizes an ordering of pages based on clickstream or shopping cart data.

We will illustrate how the unconstrained quadratic binary program (UQP) can effectively serve as an attractive new approach for modeling and solving linear ordering problems from the standpoint of being a general purpose model design applicable to many other kinds of problems. In a theoretical sense, the UQP formulation is as general as the 0-1 pure integer linear programming (ILP) problem, and hence it becomes valuable to know the range of 0-1 ILP problems for which UQP also yields an exploitable design. In particular, we are interested to discover those 0-1 problems whose UQP formulation gives a basis for applying methods that yield high quality solutions efficiently. From this standpoint, we are motivated to compare the UQP model against classical 0-1 ILP models, applied to problems from different settings to determine which problems are more exploitable by UQP and which are more exploitable by 0-1 ILP.

We focus on the LOP problem as one that is challenging and has a significant range of applications, and compare the use of the UQP model solved by a tabu search method (which has been shown to be effective for this formulation) against the use of the 0-1 ILP model solved by CPLEX (which is a leading method for exploiting this latter formulation). Our results indicate that the UQP model has significant advantages over the 0-1 ILP model in this context, and yields particularly good outcomes for large, dense instances of LOP.

It is to be noted that CPLEX is an exact optimizing method, guaranteeing an optimal solution in a finite (though possibly large) time, while tabu search is a heuristic

approach that does not offer such a guarantee. However, we know of no general purpose heuristic for the 0-1 ILP formulation that dominates CPLEX on a notable range of problems, in part because CPLEX has been progressively modified over the years by teams of skilled researchers to incorporate a variety of components that may appropriately be considered heuristic. For the restricted goal of finding a first-feasible solution, the feasibility pump method of Fischetti, et al. (2005) has proved better than CPLEX, but cannot compete with CPLEX for finding optimal or near-optimal solutions.

It is on the dimension of quickly finding elite (i.e., very high quality) solutions that we seek to compare the UQP formulation to the 0-1 ILP formulation, and for this purpose CPLEX is among the best methods available. Our previous findings of the power of the UQP formulation on a wide range of other applications gives added impetus for investigating whether it may also prove comparably effective for the LOP problem.

## 2. Mathematical Model

Given an  $n \times n$  matrix of weights  $C = \{c_{ij}\}$ , we wish to find the permutation of rows and columns of  $C$  such that the sum of the weights of the upper triangular matrix are maximized. Letting  $x_{ij} = 1$  if item  $i$  precedes item  $j$  in the ordering, the standard binary integer programming model (see Grötschel, et al. (1984)) for LOP is:

$$\text{LO\_IP: Max } \sum_{i < j}^n c_{ij} x_{ij} + \sum_{j < i}^n c_{ij} (1 - x_{ij}) \quad (1)$$

$$\text{s.t. } x_{ij} + x_{jk} - x_{ik} \leq 1 \quad \forall (i,j,k): i < j < k \quad (2)$$

$$x_{ij} + x_{jk} - x_{ik} \geq 0 \quad \forall (i,j,k): i < j < k \quad (3)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i,j): i < j \quad (4)$$

Even modest sized instances of LO\_IP yield models with many variables ( $O(n^2)$ ) and many constraints ( $O(n^3)$ ).

## 2.1 Solving LOP

Exact methods for solving LOP, which concentrate on LP-based methods and facet generation, can be found in Grötschel, et al. (1984) and Mitchell and Borchers (2000). Such methods, as expected, work reasonably well on small to medium sized problems but performance predictively drops off as problem size increases. Even for modest sized problems, exact methods tend to perform poorly on high density instances of LOP. Due to the computational difficulty of solving LOP exactly (i.e., to proven optimality), several heuristic methods have recently been proposed for this problem. Most notable are the scatter search approaches found in Laguna and Martí (2003) & Laguna, et al. (1999) and the Lagrangian based approach given in Belloni and Lucena (2003). These methods are reported to perform well on medium to large scale problems.

It is to be emphasized that the solution approaches highlighted above are specifically designed for LOP. The approach we take in this paper is fundamentally different from previously published methods. Rather than attempt to solve LOP as a large 0-1 linear integer program or create specialized heuristics with tailored data structures, we re-cast and solve LOP as an unconstrained quadratic binary program (UQP). Several recent articles (see Kochenberger, et al. ((to appear)), Kochenberger, et al. (2004) and Lewis, et al. (2005)) have reported on the successful application of this unified framework for combinatorial optimization problems that is a marked departure from the conventional practice of tailoring a solution method for the problem structure at hand. With the unified approach, problems are re-cast into the common modeling form of an unconstrained quadratic binary program (UQP)

$$\begin{aligned} UQP : \text{opt } x^t Qx \\ x \text{ binary} \end{aligned}$$

This problem, in turn, is solved by a solution method designed not for the original problem structure but for the generic UQP model. In this manner, a single optimizer, i.e., one designed to solve UQP, can be used to solve a wide variety of combinatorial problems. Our main objective in this paper is to illustrate how this approach can be easily and successfully employed to solve the linear ordering problem.

The transformation from a given mathematical structure into the unified framework of UQP is accomplished by imposing quadratic infeasibility penalties as an alternative to the explicit imposition of the original problem constraints. This approach can, in principle, be applied to any problem with linear constraints, bounded integer variables, and a linear or quadratic objective function. In this regard, we refer to a penalty function,  $g(x)$ , as being a *valid infeasibility penalty (VIP)* if  $g(x)$  is zero when  $x$  is feasible and positive otherwise. Further, if by the selection of an appropriate positive scalar penalty  $P$ , the combined objective function (i.e., including the penalties) is optimized over binary  $x$  when  $g(x) = 0$ , we call the term  $Pg(x)$  a *sufficient infeasible penalty (SIP)*. (The minus sign in the combined objective function is of course replaced by a plus sign if the goal is one of minimization.). For models with linear constraints, it is always possible to find *quadratic VIPs* and associated penalties  $P$  to yield corresponding *SIPs*, and thus the constrained problem can be re-cast into the form of UQP by identifying the appropriate  $Q$  matrix so that  $x^t Q x$  represents the combined function that includes the  $Pg(x)$  term. Generally, VIPs will not be known and will have to be discovered. Such discovery is straightforward as outlined in Boros and Hammer (2002), Hammer and Rudeanu (1968) and Hansen (1970). For certain problem classes, however, VIPs are known in advance, making the re-casting into the form of UQP even easier. The LOP is of this nature. To see this, refer to the constraints of (2) and (3) and consider the penalty function

$$g(x) = g(x_{ij}, x_{jk}, x_{ik}) = (x_{ik} + x_{ij}x_{jk} - x_{ij}x_{ik} - x_{jk}x_{ik}) \quad (5)$$

Examining the truth table for  $x_{ij}, x_{jk}$ , and  $x_{ik}$  reveals that  $g(x)$  is a VIP since it is zero for  $x$  satisfying (2) and (3) and otherwise positive. Thus by identifying an appropriate positive penalty  $P$  and subtracting  $Pg(x)$  (for  $g(x)$  given in (5)) from the objective function of (1) for each  $i < j < k$ , LO\_IP is directly recast into the form of

$$LO\_UQP: \max x^t Q x \\ x \text{ binary}$$

where  $x$  is a vector with  $n' = n(n-2)/2$  variables and  $Q$  is an  $n' \times n'$  symmetric matrix.

Note that in the recasting of LO\_IP into LO\_UQP we go from a linear model with many constraints to a quadratic model with no constraints and the transformation of

LO\_IP into LO\_UQP is accomplished without the introduction of new variables. Also, the original objective function information of LO\_IP appears on the main diagonal of  $Q$ , with the scalar penalty,  $P$ , sufficiently large so that solutions to LO\_UQP are feasible for LO\_IP. In the computational testing reported in section 3, choosing  $P = 20$  (twice the maximum  $c_{ij}$  value) was sufficient in that our solutions were feasible in all cases. Additional testing indicated that even smaller values of  $P$  worked as well.

## 2.2 Example:

The procedure given above is illustrated by the following example found in Laguna, et al. (1999). Consider the initial table of weights

		1	2	3	4
1		0	12	5	3
2		4	0	2	6
3		8	3	0	9
4		11	4	2	0

where the initial permutation is  $p = (1,2,3,4)$  with value 37.

The LO\_IP model is:

$$\begin{aligned}
 &\text{Max } x_0 = 32 + 8x_{12} - 3x_{13} - 8x_{14} - 1x_{23} + 2x_{24} + 7x_{34} \\
 &\text{st} \\
 &x_{12} + x_{23} - x_{13} \leq 1 \quad x_{12} + x_{23} - x_{13} \geq 0 \\
 &x_{12} + x_{24} - x_{14} \leq 1 \quad x_{12} + x_{24} - x_{14} \geq 0 \\
 &x_{13} + x_{34} - x_{14} \leq 1 \quad x_{13} + x_{34} - x_{14} \geq 0 \\
 &x_{23} + x_{34} - x_{24} \leq 1 \quad x_{23} + x_{34} - x_{24} \geq 0
 \end{aligned}$$

Utilizing the penalties given by (5), the equivalent  $xQx$  model is completely specified by the 6x6  $Q$  matrix:

$$Q = \begin{bmatrix} 8 & P & P & -P & -P & 0 \\ P & -3-2P & P & P & 0 & -P \\ P & P & -8-4P & 0 & P & P \\ -P & P & 0 & -1 & P & -P \\ -P & 0 & P & P & -2-2P & P \\ 0 & -P & P & -P & P & 7 \end{bmatrix}$$

Choosing the scalar penalty,  $P$ , to be 10 and solving the problem:

$$\text{Max } x_0 = 32 + x^t Q x$$

yields the value 15 for  $x^t Q x$  with  $x_{12}$  and  $x_{34} = 1$  (all other variables equal to 0). Thus, the optimal objective function value is  $15 + 32 = 47$  and the corresponding permutation is:  $p = (3,4,1,2)$  yielding the table shown below.

		3	4	1	2
3		0	9	8	3
4		2	0	11	4
1		5	3	0	12
2		2	6	4	0

### 3. Computational Experience

To test the UQP approach to LOP, we randomly generated a set of linear ordering test problems. The publicly available test problems provided by LOLIB (1997) are known to be fairly easy to solve exactly with a commercial solver, as reported in Belloni and Lucena (2003) and as a result, several authors (Belloni and Lucena (2003), Mitchell and Borchers (2000)) have proposed problem generators intended to produce difficult instances of LOP. In generating the problems considered here, we followed the approach used in Mitchell and Borchers (2000) but purposely set size and density parameters to create problems of even greater difficulty. The main diagonal elements of the generated weight matrix  $C$  are zero while the off diagonal elements are randomly selected from a uniform distribution between zero and ten. Because almost every pair of distinct elements in  $C$  have non-zero weights, these problems are very dense and more difficult than those found in LOLIB or elsewhere in the literature. As shown below,

these problems also proved to be difficult for CPLEX, even when set to emphasize quickly finding good solutions at the expense of proving optimality.

Table 1 describes the characteristics of the twelve problems generated. To provide a standard and readily available basis of comparison, each problem was also solved in the original form of LO\_IP utilizing CPLEX 8.1. CPLEX was unable to make MIP presolve or aggregator reductions on any of the problems.

Each of our test problems was re-cast into LO\_UQP using a scalar penalty value of  $P = 20$  and solved by a basic multi-start tabu search meta heuristic with path re-linking (called MSTR), described in Lewis (2005). The basic move structure in MSTR is defined by a simple 1-opt method with a set of elite solutions maintained as the search process unfolds. The multi-start approach perturbs one of the elite solutions saved for path-relinking by setting those variables that are often at a level of one to zero (and vice-versa) and giving them a longer tabu tenure. Path re-linking is applied periodically between the current solution and the elite set and also to new incumbent solutions as an intensification strategy. In all cases the results obtained, as given in the following section, were feasible with respect to the original constrained LO\_IP formulation.

Problem ID	n	# binary variables	# CPLEX rows
LO_LAK_20	20	190	2280
LO_LAK_30	30	435	8,120
LO_LAK_40	40	780	19,760
LO_LAK_50	50	1225	39,200
LO_LAK_60	60	1770	68,440
LO_LAK_70	70	2415	109,480
LO_LAK_80	80	3160	164,320
LO_LAK_90	90	4005	234,960
LO_LAK_100	100	4950	323,400
LO_LAK_150	150	11175	1,102,600
LO_LAK_200	200	19900	2,626,800

TABLE 1. Problem Characteristics

### 3.1. Results of Testing

All problems were run using a Dell PowerEdge Server having four 2.0 GHz processors and 4 GB of RAM. Parallel processing was not utilized. To provide a more meaningful comparison, CPLEX v8.1 mip parameters were set to emphasize quickly finding feasible solutions (as opposed to proving optimality). Table 2 presents the results from solving the twelve LO\_LAK problems with both tabu search and CPLEX. The total times are in cpu seconds and for CPLEX indicate either the time to optimality or an out of memory error condition. MSTR total times reflect the time to execute five million tabu search iterations. Table 2 also illustrates that as the problem size increases, the lp-relaxations are increasingly difficult for CPLEX to solve, resulting in fewer nodes investigated.

MSTR finds optimal solutions quickly for the first two smaller problems. While for LO\_LAK\_40, LO\_LAK\_50 and LO\_LAK\_60, MSTR finds comparable solutions much faster than CPLEX, e.g. for LO\_LAK 50, it took CPLEX approximately 60 times as long to find the MSTR solution. For the larger problems, MSTR solutions were on average 31% better than CPLEX and comparable solutions were obtained orders of magnitude faster, e.g. for LAK\_80 MSTR finds the solution 1148 in 60 seconds (vs. 930 in 7601 sec). Table 2 also illustrates the general range of problem sizes solvable and although larger problem instances can be solved with our current approach, instances with  $n$  greater than 500 will require different data structures or larger amounts of memory.

## 4. Summary and Conclusions

The results reported in this paper are consistent with the competitive results obtained by the UQP approach to other problem classes (see, for instance Kochenberger, et al. ((to appear)) and Lewis, et al. (2005)). Our performance here serves to further emphasize the viable, robust nature of UQP as a fruitful modeling and solution framework for various combinatorial optimization problems. As new and improved optimizers become available for solving UQP, this unified framework will develop into an increasingly attractive approach for rapidly finding high-quality solutions to such problems.

Problem ID	CPLEX					MSTR		
	Tree Nodes	Time to Sol	Total Time	Final MIP Gap	Solution	Time to Sol	Total Time	Solution
LAK_20	0	1	1	0%	160	1	60	160
LAK_30	94	114	150	0%	240	7	150	240
LAK_40	850	2,363	3,310	0%	611	103	364	600
LAK_50	2651	171,145	175,671	6%	732	676	1332	705
LAK_60	550	129,657	175,491	14%	1084	534	5402	1010
LAK_70	48	34,588	54,398	87%	970	5111	5402	1307
LAK_80	30	7,601	175,158	134%	930	5045	5711	1453
LAK_90	10	18,237	170,105	133%	1150	5742	7704	1687
LAK_100	2	24,632	88,108	155%	1363	7994	8923	1943
LAK_150	-	-	-	-	-	14950	16063	2857
LAK_200	-	-	-	-	-	21175	50005	2875

TABLE 2. Comparison of CPLEX and Multi-start Tabu Search

## References

- Bar-Joseph Z., Gifford D. K., and Jaakkola T. S.,(2001) Fast optimal leaf ordering for hierarchical clustering, *Bioinformatics* 17 S22-S29.
- Belloni A. and Lucena A.,(2003) Langrangian heuristics for the linear ordering problem, *Metaheuristics: Computer Decision-Making* 37-63.
- Biedl T., Brejova B., Demaine E., A. Hamel A., and T. Vinar T.,(2001) Optimal Arrangement of Leaves in the Tree Representing Hierarchical Clustering of Gene Expression Data, Technical report 2001-14, University of Waterloo, Canada.
- Boros E. and Hammer P. L.,(2002) Pseudo-Boolean Optimization, *Discrete Allied Math* 123 155-225.
- Cooley R.,(2003) The Use of Web Structure and Content to Identify Subjectively Interesting Web Usage Patterns, *ACM Transactions on Internet Technology* 3 93-116.
- Fischetti M., Lodi A., and Glover F.,(2005) The Feasibility Pump, *Mathematical Programming - Series A* DOI 10.1007/s10107-004-0570-3.
- Grötschel M., Jünger M., and Reinelt G.,(1984) A Cutting Plane Algorithm for the Linear Ordering Problem, *Operations Research* 32 1195-1220.

- Hammer P. and Rudeanu S., 1968, *Boolean Methods in OR and Related Areas*, (Eds.) Berlin, Heidelberg, New York: Springer-Verlag.
- Hansen P.,(1970) Methods of Nonlinear 0-1 Programming, *Annals of Discrete Math* 5 53-70.
- Kochenberger G., Glover F., Alidaee B., Rego C.,(2005) An Unconstrained Quadratic Binary Approach to the Vertex Coloring Problem, *Annals of OR* 139 229-241.
- Kochenberger G. A., Glover F., Alidaee B., Rego C.,(2004) A unified modeling and solution framework for combinatorial optimization problems, *OR Spectrum* 26 237-250.
- Laguna M. and Martí R., 2003, *Scatter search: methodology and implementations in C*, (Eds.) Boston: Kluwer Academic Publishers.
- Laguna M., Martí R., and Campos V.,(1999) Intensification and diversification with elite tabu search solutions for the linear ordering problem, *Computers and Operations Research* 26 1217-1230.
- Lewis M.,(2005) A path-relinking multi-start tabu search metaheuristic for unconstrained quadratic binary optimization, *Hearin Center Technical Report HCES-01-05*.
- Lewis M., Alidaee B., and Kochenberger G.,(2005) Using xQx to Model and Solve the Uncapacitated Task Allocation Problem, *Operations Research Letters* 33 176-182.
- LOLIB,(1997) Linear Ordering Library, <http://www.iwr.uni-heidelberg.de/iwr/comopt/soft.LOLIB/LOLIB.html>.
- Mitchell J. E. and Borchers B., 2000, *High Performance Optimization*, (Eds.) H. Frenk. Boston: Kluwer Academic Publishers.
- Mitchell J. E. and Borchers B., "Solving Linear Ordering Problems With A Combined Interior Point/Simplex Cutting Plane Algorithm," in *High Performance Optimization*, H. F. e. al., Ed.: Kluwer Academic Publishers, 2000, pp. 349-366.
- Mitchell J. E., Pardalos P. P., and Resende M. G. C., "Interior point methods for combinatorial optimization," in *Handbook of Combinatorial Optimization*, vol. 1, D. Z. Du and P. Pardalos, Eds.: Kluwer Academic Publishers, 1998, pp. 189-297.
- Steiner G. and Stephenson P.,(2000) Subset-Restricted Interchange For Dynamic Min-Max Scheduling Problems, *SIAM J. Discrete Math* 13 419-435.