

Modeling Applications of Linear Programming: Matching, Covering, Packing, and Linear Ordering

Thomas R. Cameron

March 3, 2026

1 The Matching Problem

Given a graph $G = (V, E)$, a matching is a subset of edges $M \subseteq E$ such that no two edges in M share common vertices. The matching problem on the graph G seeks a matching of maximum cardinality. For each $e \in E$, let x_e denote a binary variable that indicates whether the edge e is in a matching of G . Then, the matching problem can be modeled by (1a)–(1c).

$$\text{maximize } z = \sum_{e \in E} x_e \quad (1a)$$

$$\text{subject to } \sum_{e \in E: v \in e} x_e \leq 1, \forall v \in V, \quad (1b)$$

$$x_e \in \{0, 1\}, \forall e \in E \quad (1c)$$

Note that the objective function of the matching problem, denoted w , is maximizing the number of edges included in the matching. Furthermore, constraint (1b), says that no vertex can be included in more than one edge in the matching. Finally, constraint (1c) says that each edge of G is either included or not included in the matching. Since each constraint is a linear inequality and the variables have integrality constraints the matching problem is an example of an integer linear optimization problem. We say that a binary vector \mathbf{x} of cardinality $|E|$ is a feasible solution to the matching problem if constraints (1b)–(1c) are satisfied. Moreover, we say that \mathbf{x} is optimal if there is no other feasible solution with a larger objective function value.

As an example, consider a math department consisting of 10 courses it would like to offer at a single time slot. Also, the math department has 10 professors available during that time slot whose teaching qualifications are listed in Table 1. The goal is to offer as many of the 10 courses possible during the single time slot.

We can represent the given information as a bipartite graph as shown in Figure 1. Note that for each $v \in \{0, 3, 5, 8\}$, we have $N(v) = \{B, E\}$. Therefore, at least two of the courses cannot be offered. Also, note that the vertices A and G are the only vertices whose neighborhood includes $\{1, 7, 9\}$. Hence, at least one of those courses cannot be offered, so at most 7 course can be offered. The following collection of edges

$$\{A, 1\}, \{B, 0\}, \{C, 4\}, \{D, 6\}, \{E, 3\}, \{F, 2\}, \{G, 7\}$$

is a matching of cardinality 7, which is therefore optimal. A vertex cover is a subset of vertices such that every edge contains a vertex from that subset. The König-Egerváry Theorem states that, for bipartite graphs, the maximum matching and minimum vertex cover have the same cardinality. Note that $\{A, B, E, G, 2, 4, 6\}$ is a vertex cover of cardinality 7.

Last Name	Courses
Aguilera	401, 402, 407, 409
Backman	400, 403, 404, 405, 408
Carter	404, 406
Dykstra	404, 406
Elster	400, 402, 403, 405, 408
Fernandez	402, 404, 406
Gooden	401, 406, 407, 409
Hernandez	404, 406
Innis	402, 406
Johnson	402, 404

Table 1: Teaching qualifications for math department.

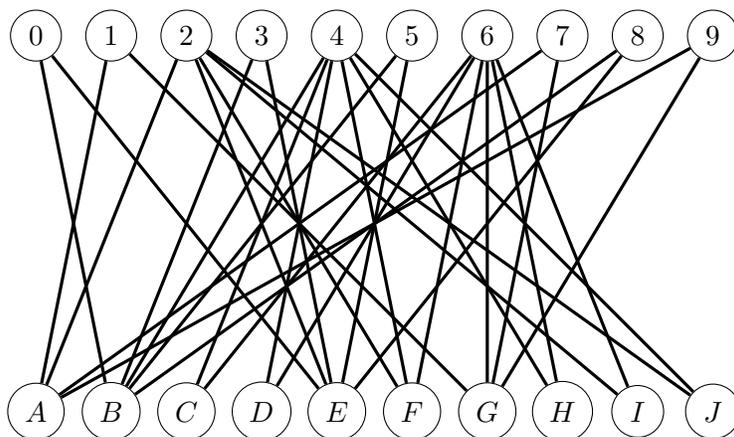


Figure 1: Graph of teaching qualifications for math department.

2 The Set Cover and Set Packing Problems

Let \mathcal{U} denote a finite set of cardinality $n \geq 0$ and let $\mathcal{S} = \{S_1, \dots, S_k\}$ denote a collection of subsets of \mathcal{U} . A set cover of \mathcal{U} is a subfamily $\mathcal{C} \subseteq \mathcal{S}$ whose union is \mathcal{U} . The set cover problem on \mathcal{U} seeks a set cover of minimum cardinality. For each $i \in \{1, \dots, k\}$, let x_i denote a binary variable that indicates whether the set S_i is included in the set cover. Then, the set cover problem can be modeled by (2a)–(2c).

$$\text{minimize } w = \sum_{i=1}^k x_i \quad (2a)$$

$$\text{subject to } \sum_{i: v \in S_i} x_i \geq 1, \forall v \in \mathcal{U}, \quad (2b)$$

$$x_i \in \{0, 1\}, \forall i \in \{1, \dots, k\} \quad (2c)$$

We say that $\mathcal{C} \subseteq \mathcal{S}$ is a set packing with respect to \mathcal{U} if the subsets in \mathcal{C} are pairwise disjoint. The set packing problem on \mathcal{U} seeks a set packing of maximum cardinality. For each $i \in \{1, \dots, k\}$, let x_i denote a binary variable that indicates whether the set S_i is included in the set packing.

Then, the set packing problem can be modeled by (3a)–(3c).

$$\text{maximize } z = \sum_{i=1}^k x_i \quad (3a)$$

$$\text{subject to } \sum_{i: v \in S_i} x_i \leq 1, \forall v \in \mathcal{U}, \quad (3b)$$

$$x_i \in \{0, 1\}, \forall i \in \{1, \dots, k\} \quad (3c)$$

As an example, consider the universe $\mathcal{U} = \{1, 2, 3, 4, 5\}$ and the collection of subsets $\mathcal{S} = \{\{1, 2, 3\}, \{2, 4\}, \{3, 4\}, \{4, 5\}\}$ as shown in Figure 2. Note that a minimum cover is given by $\mathcal{C} =$

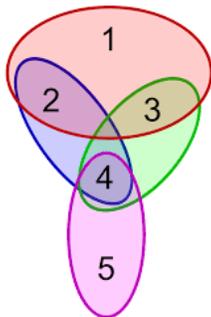


Figure 2: Universe and collection of subsets.

$\{\{1, 2, 3\}, \{4, 5\}\}$. Moreover, \mathcal{C} is also a maximum packing.

3 The Linear Ordering Problem

Let $n \geq 2$ and $a_{ij} \in \mathbb{R}_{\geq 0}$ for $1 \leq i, j \leq n$. Then, the LOP determines a *symmetric re-ordering* of the matrix $A = [a_{ij}]$, i.e., a similarity transformation of A via a permutation matrix, that results in a maximum upper-triangular sum. Equivalently, if A is the adjacency matrix of a simple digraph Γ , then the LOP seeks to find a spanning acyclic tournament sub-digraph of Γ with maximal weight sum, where we use the convention that an edge has weight zero if and only if the edge does not exist. The integer program formulation of this problem, denoted $LOP(A)$, is defined below:

$$\text{maximize } \sum_{i \neq j: i, j \in [n]} a_{ij} x_{ij} \quad (4a)$$

$$\text{subject to } x_{ij} + x_{ji} = 1, \quad \forall i < j: i, j \in [n], \quad (4b)$$

$$x_{ij} + x_{jk} + x_{ki} \leq 2, \quad \forall i < j, i < k, j \neq k: i, j, k \in [n], \quad (4c)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \neq j: i, j \in [n], \quad (4d)$$

where the x_{ij} are decision variables that indicate whether the edge (i, j) , with weight a_{ij} , should be included in the sub-digraph of Γ with maximum weight sum. Note that constraint (4b) ensures that the sub-digraph is a spanning tournament and constraint (4c) forces the tournament to contain no dicycles.

In Figure 3, the rankings of several features used in the U.S. World News College Rankings are reported for 10 liberal arts colleges, listed in alphabetical order, for the year 2013. Numbering the

College	Selectivity	Faculty Resource	Student/Faculty Ratio	Graduate Retention	Financial Resources
Amherst	5	7	9/1	1	10
Bowdoin	8	14	10/1	6	14
Carleton	12	16	9/1	4	27
Claremont	14	4	9/1	11	21
Haveford	2	5	8/1	6	15
Middlebury	6	17	9/1	11	3
Pomona	2	20	8/1	1	6
Swarthmore	6	7	8/1	4	9
Wellesley	12	12	8/1	14	10
Williams	4	3	7/1	1	6

Figure 3: College Features from U.S. World News 2013 Rankings

colleges as they appear from top to bottom in Figure 3, i.e., in alphabetical order, let a_{ij} denote the number of features that college i outperforms college j , where 0.5 is awarded to both schools in the case of a tie. Then, the adjacency matrix $A = [a_{ij}]$ is defined by

$$A = \begin{bmatrix} 0 & 5 & 4.5 & 3.5 & 2 & 3.5 & 1.5 & 2.5 & 3.5 & 0.5 \\ 0 & 0 & 3 & 3 & 1.5 & 2 & 1 & 0 & 2 & 0 \\ 0.5 & 2 & 0 & 2.5 & 1 & 2.5 & 1 & 0.5 & 1.5 & 0 \\ 1.5 & 2 & 2.5 & 0 & 1 & 2 & 1 & 1 & 2 & 0 \\ 3 & 3.5 & 4 & 4 & 0 & 4 & 2 & 2.5 & 3.5 & 1 \\ 1.5 & 3 & 2.5 & 3 & 1 & 0 & 1 & 1.5 & 3 & 0 \\ 3.5 & 4 & 4 & 4 & 3 & 4 & 0 & 3.5 & 3.5 & 2 \\ 2.5 & 5 & 4.5 & 4 & 2.5 & 3.5 & 1.5 & 0 & 4.5 & 0 \\ 1.5 & 3 & 3.5 & 3 & 1.5 & 2 & 1.5 & 0.5 & 0 & 0 \\ 4.5 & 5 & 5 & 5 & 4 & 5 & 3 & 5 & 5 & 0 \end{bmatrix},$$

and an optimal solution of $LOP(A)$ will correspond to an optimal aggregate of the college feature rankings in Figure 3. For instance, using the *Gurobi* optimization suite, we are able to find an optimal ranking

$$\sigma = (10, 7, 8, 5, 1, 6, 9, 2, 4, 3)$$

of $LOP(A)$, which indicates that with respect to the given data, Williams can be considered the best college and Carleton the worst college. Moreover, the symmetric re-ordering of A with respect to σ results in the following matrix

$$A_\sigma = \begin{bmatrix} 0 & 3 & 4 & 4.5 & 5 & 5 & 5 & 5 & 5 & 5 \\ 2 & 0 & 3 & 3.5 & 3.5 & 4 & 3.5 & 4 & 4 & 4 \\ 1 & 2 & 0 & 3 & 2.5 & 4 & 3.5 & 3.5 & 4 & 4 \\ 0.5 & 1.5 & 2 & 0 & 2.5 & 3.5 & 3.5 & 5 & 4.5 & 3 \\ 0 & 1.5 & 2.5 & 2.5 & 0 & 3.5 & 4.5 & 5 & 4.5 & 4 \\ 0 & 1 & 1 & 1.5 & 1.5 & 0 & 3 & 3 & 2.5 & 3 \\ 0 & 1.5 & 1.5 & 1.5 & 0.5 & 2 & 0 & 3 & 3.5 & 3 \\ 0 & 1 & 1.5 & 0 & 0 & 2 & 2 & 0 & 3 & 3 \\ 0 & 1 & 1 & 0.5 & 0.5 & 2.5 & 1.5 & 2 & 0 & 2 \\ 0 & 1 & 1 & 1.5 & 1 & 2 & 2 & 2 & 2.5 & 0 \end{bmatrix},$$

which can be formed by swapping the rows and columns indicated by the permutation σ .