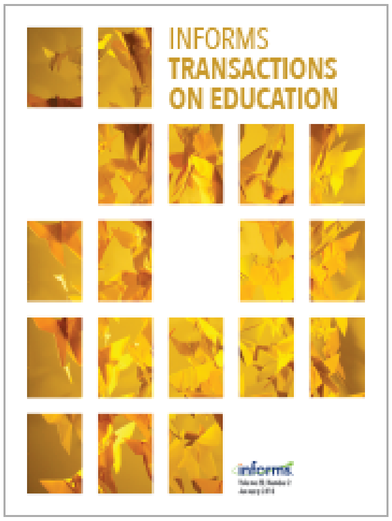


INFORMS Transactions on Education

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Puzzle—Brushing the Court Lines Optimally

Kenneth R. Baker, Michael J. Magazine

Vol. 16, No. 2, January 2016, pp. 79–83
ISSN 1532-0545 (online)

INFORMS Transactions on Education

http://dx.doi.org/10.1287/ited.2015.0150

Puzzle

Brushing the Court Lines Optimally

Kenneth R. Baker
Tuck School of Business, Dartmouth College, Hanover, New Hampshire 03755, ken.baker@dartmouth.edu

Michael J. Magazine
Lindner College of Business, University of Cincinnati, Cinncinnati, Ohio 45221, magazim@ucmail.uc.edu

Clay-court tennis players are familiar with the well-known courtesy, after finishing their match, of preparing the court for the next players who will use the court. Preparation consists of two parts: sweeping the court and brushing the lines. Sweeping the court is straightforward, but the quantitatively oriented player surely wants to know how to brush the lines (a nontrivial routing problem) in the most efficient fashion.

Keywords: integer programming; spreadsheet optimization; routing problems; post-optimality analysis; rural postman problem

History: Received: October 2015; accepted: October 2015. Published online in *Articles in Advance*.

Introduction

Clay-court tennis players are familiar with the well-known courtesy, after finishing their match, of preparing the court for the next players who will use the court. Preparation consists of two parts: sweeping the court, which makes the surface uniform but partially obscures the lines, and brushing the lines, which makes the tapes on the court clearly visible. Sweeping the court is straightforward, especially in facilities where there is a mandate to sweep in a particular orientation (usually east/west). Brushing the lines, on the other hand, presents a routing problem; and the quantitatively oriented player surely wants to know how to brush the lines in the most efficient fashion. To be specific, the problem is to find the shortest length of a walking path that permits brushing all the lines.

Here, we pose the problem of brushing the lines as an optimization problem that is amenable to a spreadsheet-based solution. We first present data for the problem, which is easy to find with an Internet search. Next, we state the problem in its basic form and clarify its assumptions. We then note its relation to models discussed in the literature, particularly the Traveling Salesperson Problem (TSP). Finally, we explore some interesting postoptimality topics.

We begin with the diagram of a standard tennis court. Figure 1 shows its official dimensions on one side of the net. The baseline measures 36 feet in length, and the service line (set back 21 feet from the net) measures 27 feet. The pair of sidelines forming the "alley" on either side measure 39 feet, with a separation of 4.5 feet. (No line exists where the net is located.) The lengths of all the lines on one side of the net sum to 240 feet. Typically, two brushes are available, so one player brushes the lines on one side of the net while a partner brushes the lines on the other side.

For modeling purposes, Figure 2 assigns numerical labels to each of the intersections formed by the network of court lines. In network terminology, the intersections are called *vertices*. As shown in the figure, the network contains 12 vertices. We can use the 12 vertex labels to identify the 12 line segments that must be brushed. A line segment, or in network terms, an *edge*, connects exactly two adjacent vertices. The baseline, which connects vertices 2 and 5, is not itself an edge but rather a set of connected *directed edges* that can be ordered to form a *path*. The optimization problem becomes one of finding a path of minimum length that contains all 12 segments. We refer to this as the *Brushing the Lines Problem* (BLP).

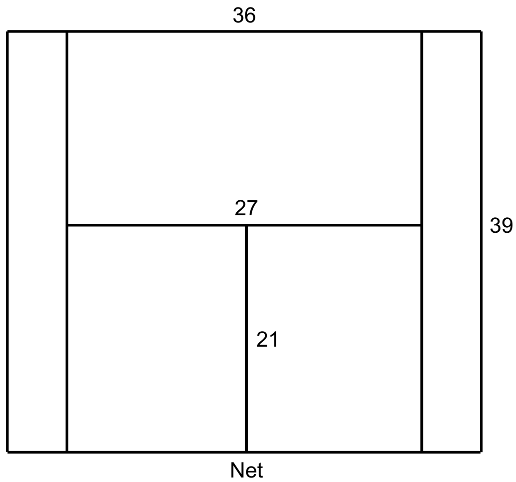
The Basic Problem

To make the problem clearer, it is helpful to clarify some assumptions.

What is the starting point for the brushing path? This is a relevant consideration because the choice of a starting point could certainly influence the solution. However, most tennis facilities have no standard location for storing the brush. Therefore, we assume that the starting

INFORMS Transactions on Education 16(2), pp. 79–83, ©2016 INFORMS

Figure 1 **Lengths (in Feet) of the Lines on a Tennis Court**

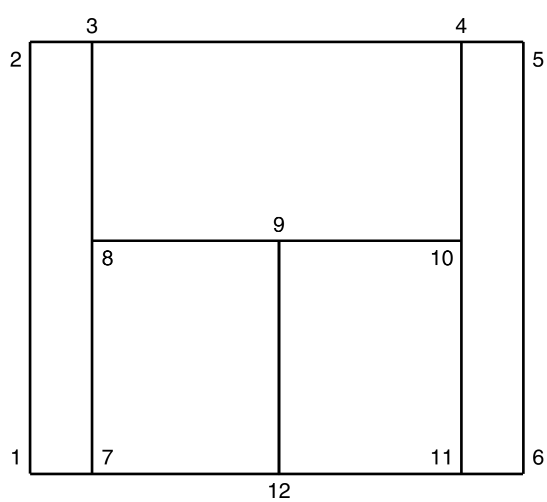


point for brushing can be at any vertex, and we measure the length of the path from that point. Similarly, there is no reason for the brushing path to return to its starting point, so a second assumption is that the finishing point for brushing can also be at any vertex.

Must the walking path follow the court lines? Empirically, at least, we sometimes see a player walk from vertex 1 to vertex 7 (or from 6 to 11). Although this edge connects two vertices, it does not correspond to a line in need of brushing, yet it may potentially be helpful at reducing the total distance walked. The implication, then, is that we need not limit the solution to the edges shown in Figure 2; any edge should be admissible. In other words, we assume that the underlying network of edges connects every pair of the 12 vertices in Figure 2.

Is it permissible to revisit an edge? It may be tempting to require that no edge appear on the path more than once. However, we might again draw on empirical observations that people often return to an edge even though no brushing is needed on the second visit. Indeed, a prohibition on revisiting an edge is unnecessary: If we can find a shorter path with edges revisited than any path without multiple visits, then we should surely prefer the shorter one. On the other hand, if we allow edges to be revisited and discover an optimal path without that property, then we know that the additional flexibility is redundant.

Figure 2 **The 12 Vertices of the Problem**



At this point, we can refine the problem statement for the BLP: Find the shortest walking path that traverses all the court lines in the larger network of edges connecting the 12 given vertices, from any starting vertex to any ending vertex. To build a suitable optimization model, we will focus on identifying decision variables and constraints, as well as on quantifying the objective function in terms of the decision variables.

Similarities to Other Problems

A standard early step in problem solving is to ask whether the problem at hand resembles a previously encountered or well-known problem for which a solution is already available (Polya 1945). In this case, the BLP is actually an instance of the Rural Postman Problem (RPP). However, the RPP is known to be NP-hard (Lenstra and Rinnooy Kan 1976) and no simple, easy solution algorithm exists. Although some integer programming formulations have been proposed for the general RPP (Eiselt et al. 1995), they do not lend themselves to solution with Excel's Solver because of the potentially large number of constraints. Other, more specialized, enumerative algorithms have been proposed, but those cannot be implemented in a spreadsheet-based approach without extensive additional coding.

The BLP is also similar to the Chinese Postman Problem (CPP), which seeks the minimum length of a path that visits every edge. The CPP has polynomial solutions as long as each edge can be traversed in either direction (Edmonds and Johnson 1973). The BLP differs in that we need not visit every edge but only a particular subset of edges. This difference makes the problem much more difficult.

The BLP resembles the Traveling Salesperson Problem (TSP), which seeks the shortest route that visits each vertex exactly once and returns to the starting vertex. The resemblance is visible at least in the sense that, for the TSP, the given information is an array of distances between vertex pairs and the objective is to find a minimum-length path containing all the vertices. (In our special case of the RPP, any path that traverses all the edges will necessarily pass through all the vertices.) Using the TSP as a guide, we introduce some formal notation. Let the index j denote a vertex ($j = 1, 2, \ldots, 12$), and let the pair (j, k) denote the

Table 1 **Distances (in Feet) Between Pairs of Vertices**

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.00	39.00	39.26	50.13	53.08	36.00	4.50	21.48	27.66	37.86	31.50	18.00
2	39.00	0.00	4.50	31.50	36.00	53.08	39.26	18.55	25.46	36.28	50.13	42.95
3	39.26	4.50	0.00	27.00	31.50	50.13	39.00	18.00	22.50	32.45	47.43	41.27
4	50.13	31.50	27.00	0.00	4.50	39.26	47.43	32.45	22.50	18.00	39.00	41.27
5	53.08	36.00	31.50	4.50	0.00	39.00	50.13	36.28	25.46	18.55	39.26	42.95
6	36.00	53.08	50.13	39.26	39.00	0.00	31.50	37.86	27.66	21.48	4.50	18.00
7	4.50	39.26	39.00	47.43	50.13	31.50	0.00	21.00	24.96	34.21	27.00	13.50
8	21.48	18.55	18.00	32.45	36.28	37.86	21.00	0.00	13.50	27.00	34.21	24.96
9	27.66	25.46	22.50	22.50	25.46	27.66	24.96	13.50	0.00	13.50	24.96	21.00
10	37.86	36.28	32.45	18.00	18.55	21.48	34.21	27.00	13.50	0.00	21.00	24.96
11	31.50	50.13	47.43	39.00	39.26	4.50	27.00	34.21	24.96	21.00	0.00	13.50
12	18.00	42.95	41.27	41.27	42.95	18.00	13.50	24.96	21.00	24.96	13.50	0.00

directed edge from vertex j to vertex k. In addition, let c_{jk} denote the distance from vertex j to vertex k. The distances c_{jk} are shown in Table 1.

We introduce the variables x_{jk}, defined as follows.

$$x_{jk} = 1 \text{ if the path includes a directed edge from } j \text{ to } k;$$
$$= 0 \text{ otherwise.}$$

Thus the objective of finding a minimum-length path in the BLP, as in the TSP, is equivalent to minimizing the sum $\sum_j \sum_k c_{jk} x_{jk}$.

The data set in Table 1 can be modified to help formulate the BLP conveniently. Anticipating the use of a spreadsheet model, we allow the variables x_{jj} to be in the formulation, but we discourage their use in the solution by resetting $c_{jj} = M$, which plays the role of a large cost sufficient to make the variables x_{jj} unattractive.

Whereas the BLP resembles the TSP, some differences are also noticeable. The TSP seeks a path that forms a tour (returns to its starting point), which is not the case in our basic version of the BLP. Nevertheless, it is useful to think about how the TSP adapts to a variation in which the path need not return to its starting point. For that variation, a "dummy" vertex is added to the network and assigned a distance of zero to and from each other vertex. Then an optimal tour is obtained for the expanded problem. In that solution, the edge that exits from the dummy vertex is interpreted as leading to the starting point on the optimal path, and the edge that enters the dummy vertex is interpreted as emanating from the finishing point. A similar enhancement of the BLP turns out to be convenient, so we formulate the problem as if we were searching for a tour that starts and ends at vertex 0. We thus add the dummy vertex 0 and corresponding distances $c_{0j} = c_{j0} = 0$, for $1 \leq j \leq 12$. With this modification, we can restate the problem as finding a minimum-length tour (starting and finishing at vertex 0), thus enabling us to interpret the result as a path in the original network. As a consequence,

Table 2 **Edges (j, k) to Be Brushed Define the Set E**

j	1	2	3	4	5	11	4	3	7	8	9	9
k	2	3	4	5	6	10	10	8	8	9	10	12

in our BLP formulation—as in the traditional TSP formulation—we may have to consider the possibility of additional constraints that prohibit subtours in the solution.

A second difference is that no specific edges are required to appear in TSP solutions; any combination of edges that forms a tour is admissible. In the BLP, by contrast, 12 of the 78 available edges must specifically be included in the path. Relative to the TSP, we can think of this requirement as an added set of specific *covering constraints*. The (undirected) edges that require brushing are listed as the set E in Table 2, with reference to the vertex numbering in Figure 2.

A third difference is that each vertex must be visited exactly once in the TSP; this implies that a given vertex has just one incoming edge and one outgoing edge. In the BLP as well, every vertex will be visited, but it is possible to visit a vertex more than once. Relative to the TSP, we can think of this possibility as a *flexibility condition*.

Finding a Solution

We can attack the BLP by adapting an approach that is suited for the TSP. A first step is to build on the assignment subproblem, for at the heart of the TSP formulation, we require

$$\sum_j x_{kj} = 1, \quad \text{for all } k,$$

$$\sum_k x_{jk} = 1, \quad \text{for all } j.$$

In other words, each vertex in the TSP has one entering edge and one leaving edge. However, in the BLP we must relax these constraints to accommodate the flexibility condition mentioned above. Thus, the heart of the BLP model becomes

$$\sum_j x_{jk} \geq 1, \quad \text{for all } k \geq 1,$$

$$\sum_k x_{jk} \geq 1, \quad \text{for all } j \geq 1.$$

For the dummy vertex, however, we require exactly one entering edge and one leaving edge:

$$\sum_j x_{j0} = 1,$$

$$\sum_k x_{0k} = 1.$$

Because of the flexibility condition, we must add a set of constraints to ensure consistency by requiring that the number of edges entering a vertex is the same as the number leaving that vertex. (This requirement is implied by the assignment equalities in the TSP but would not be automatic in the BLP.) Thus, we have

$$\sum_k x_{jk} = \sum_k x_{kj}, \quad \text{for all } j \geq 1.$$

In addition, the BLP includes covering constraints where the TSP has none, as mentioned above. Thus, we add constraints $x_{jk} + x_{kj} \geq 1$ for all edges (j, k) corresponding to set E.

At this point, we have an integer program that represents our basic version of the BLP but without consideration of subtours (that is, sets of vertices in the solution that are not connected to all other vertices in the solution). We may worry that, as in the TSP, our model may now need constraints to prohibit subtours. However, the BLP has a specific size and distance array, so its subtour elimination properties may not be typical of all RPP formulations. Indeed, something quite fortuitous occurs—when we solve the basic BLP, the optimal solution contains no subtours at all. The optimal path has a length of 280.5 feet. The optimal decision variables in cells in a run of our model[1] can be interpreted as the path

4-10-11-6-5-4-3-8-3-2-1-7-8-9-10-9-12.

This optimal path has some interesting features. Most surprising, perhaps, is that it revisits some edges: edges $(3, 8)$ and $(9, 10)$ are traversed twice (once in each direction). In addition, the only segments in the path that do not follow the court lines are two alley crossings, $(1, 7)$ and $(11, 6)$. The total length of those two segments (9 feet) and the two backtracking trips (31.5 feet) accounts for the optimal length as compared to the lower bound of 240 feet, which is the required brushing distance.

The manual construction of an optimal path from the optimal values of x_{jk} is not necessarily straightforward. In the optimal TSP solution, the set of x_{jk} corresponds to a unique path, given that it originates at node 0. In the BLP, the optimal set of x_{jk} values may imply more than one optimal path. For example, another interpretation of decision variables we encountered is the path

4-10-11-6-5-4-3-2-1-7-8-3-8-9-10-9-12.

This path differs from the previous path only in the position of the backtracking pair $(3, 8)$. Therefore, the optimal path is not unique.

Postoptimality Analyses

Having found an optimal solution, along with a slightly different alternative optimum, we can begin to explore alternative optima more systematically. To adapt our model for alternative optima, we can append an elimination constraint that renders the original optimal solution infeasible. Specifically, this elimination constraint sums the 18 positive values of x_{jk} in the original solution and constrains the total to be at most 17. In symbols:

$$x_{0,4} + x_{4,10} + x_{10,11} + \cdots + x_{9,12} + x_{12,0} \leq 17.$$

This constraint prohibits the original solution and allows us to search for some other solution that achieves a path length of 280.5. If we then wish to find a third alternative optimum, we could append a second elimination constraint, and we could in principle continue in this fashion until no additional alternative optima can be found. Nevertheless, it is possible to anticipate how many alternative optima we will find.

We can draw on some simple insights to organize the identification of alternative optima. First, we can observe the left-right symmetry in a tennis court. That observation leads to the insight that for every path starting at vertex 4, we can easily construct a symmetric path starting at vertex 3. Second, we note that in the two optimal sequences identified earlier (starting at vertex 4), we found two places in the sequence for the backtracking between vertices 3 and 8. Independent of that choice, we also have two ways of positioning the backtracking between vertices 9 and 10. A symmetric property corresponds to starting at vertex 3: one choice for backtracking between 4 and 10 and another choice for backtracking between 8 and 9.

Third, we observe that we can reverse any optimal path that starts at vertex 3 or vertex 4 and start at vertex 12 instead, thus producing another optimal path. At this stage, we note that an optimal path

[1] An Excel model for the BLP is available as supplemental material at http://dx.doi.org/10.1287/ited.2015.0150.

must start at vertex 4, vertex 3 (from symmetry), or vertex 12 (from reversibility). If we modify the original model to eliminate these options—that is, by appending constraints $x_{0,3} = x_{0,4} = x_{0,12} = 0$—then we find that the optimal solution increases to 285. This adapted form of the original model shows that to achieve optimality, the path must start at one of just three locations. If the path begins at any other point, the resulting solution will be suboptimal.

In summary, we have two choices for each of four factors:

- Where to start along the baseline.
- Where in the sequence to position the backtracking from the baseline.
- Where in the sequence to position the backtracking along a service line.
- Whether to reverse the sequence.

This analysis tells us that we have $2^4 = 16$ optimal sequences. (If this intuitive reasoning is not persuasive, then the elimination-constraint approach will reach the same conclusion.)

As a final observation, the paths listed earlier have an implication for an efficient strategy when only one brush is available for the entire court. One player can follow an optimal path, finish at vertex 12, hand the brush to a partner standing across the net at the same point, and the partner can follow the same path in reverse, starting at vertex 12 on the opposite side of the net. On the other hand, when two brushes are available, the partners can start at their respective baselines and finish by meeting at vertex 12 to discuss where to go for dinner.

Summary

We have introduced the Brushing the Lines Problem (BLP) and described its solution with a model that exploits the similarity between the BLP and the TSP. Our model contains 169 binary variables and 50 constraints, and it can even be solved in a spreadsheet using Excel's Solver. The same model can be adapted to study the rich population of alternative optima and to demonstrate, for example, that the optimal path can be generated by starting at one of only three key points on the court.

Supplemental Material

Supplemental material to this paper is available at http://dx.doi.org/10.1287/ited.2015.0150.

References

Edmonds J, Johnson E (1973) Matching, euler tours, and the chinese postman. *Math. Programming* 5(1):88–124.

Eiselt H, Gendreau M, LaPorte G (1995) Arc routing problems, Part II: The rural postman problem. *Oper. Res.* 43(3):399–414.

Lenstra JK, Rinnooy Kan A (1976) On general routing problems. *Networks* 6(3):273–280.

Polya G (1945) *How to Solve It* (Princeton University Press, Princeton, NJ).