A computer technique for optimizing the sites and heights of transmission line towers—a dynamic programming approach*

By G. Mitra and K. Wolfenden†

Given the survey data of a transmission line route and the choice of available towers of suspension type and of angle towers a dynamic programming algorithm is described which chooses and sites the towers (the location and angle of the angle towers being prescribed) in such a way that the overall cost of running the line from one end of the route to the other, subject to all the established design constraints, is a minimum. The EMA program has been successfully run at the University of London Atlas Computer using exacting test data supplied by C.E.G.B.

Given the survey data of a transmission line route and the choice of available towers of standard, i.e. suspension, type and of angle towers, it is required to choose and site the towers (the location and angle of the angle towers being prescribed) in such a way that the overall cost of running the line from one end of the route to the other is a minimum. The basic algorithm uses Bellman's "principle of optimality", the problem being considered as one of routing in a finite connected network. The profile of the line route is divided into a fine grid of possible tower sites, and the elevation corresponding to each allowable tower height at each such test site forms a node of the network in which the routing algorithm is recurrently applied. Subject to the constraints imposed by ground clearance requirements, including the effects of side slope and of special offending structures, single and double span limits, and the weight-span to wind-span ratio, the algorithm determines that connection between one node and a preceding node of the mesh which minimizes the cost of the line up to the current node. In this way a family of solutions is generated linking the terminal tower at the beginning of the line to the set of possible towers at any test site, and when the end of the line is reached the overall optimum solution is selected and the corresponding linkage traced back to the origin. Whenever the profile of a single straight line section is such that no solution is possible over this section using only suspension type towers, the program detects and monitors this situation. It then returns to a determined point on the line route from where it once again progresses, this time including the possibility of using an anchor tower at any subsequent test site.

The tower spotting problem was first set by one of us (K.W.) to a postgraduate student at the University of Leeds Electronic Computing Laboratory in 1962. The results of this exercise (Ranyard, 1963) and the installation of the Atlas Computer at the Institute of Computer Science encouraged us to tackle the problem afresh in late 1964 with a view to developing a standard program suited to British practice.

The dynamic programming approach was proposed by Shulman (1962) but he reported no implementation. Essentially the same approach has been adopted by Ahlborg and Palm (1962) whose program is now widely used for optimum tower spotting in Sweden. Other computer methods based on the established manual method of moving a transparent sag template along the profile have been successfully implemented (see Bartelink, 1964; Converti *et al.*, 1962; Hoare and Morwood, 1964; Popp *et al.*, 1963); all report a saving in cost of both the solution proposed and its derivation compared with those of traditional methods.

List of symbols

A_{hot}	The parameter of the parabola in which the line is assumed to hang at 122° F, the statutory design temperature. Under ice and wind loading at 22° F it is A_{cold}.
x	The horizontal distance from the beginning of the line section.
$s_1(x)$	The maximum single span limit at x.
s_2	The maximum double span limit.
X_{test}	An ordered set of test sites.
X_{tower}	An ordered set of tower sites.
$x_{1,i}$	The distance from x_i to the lowest point of the span (x_i, x_{i+1}) at 22° F.
$x_{-1,i}$	The distance from x_i to the lowest point of the span (x_{i-1}, x_i) at 22° F.
$z(x)$	The statutory clearance from the ground or any other object at x.
H_{tower}	An ordered set of available tower heights.
d	A horizontal displacement perpendicular to the direction of the line at x.
$v(x)$	The elevation above sea level of the lowest conductor of the line at x.
$u(x, d)$	The elevation above sea level of the ground or any offending structure at distance x and offset d.
$C(x, h)$	The cost of erecting a tower of height h at site x.

* This paper was first received on 29 Sept. 1966; the editors regret that it was not published earlier, owing to a misunderstanding.

† *University of London Institute of Computer Science*, 44 *Gordon Square, London, W.C.*1.

Mathematical formulation

The problem can now be formulated as follows. To choose an ordered set of tower sites

$$X_{\text{tower}} = \{x_i \mid i = 1, 2, \ldots M, \text{ and } x_{i-1} < x_i\}$$

from an ordered set of test sites

$$X_{\text{test}} = \{X_j \mid j = 1, 2, \ldots N, \text{ and } X_{j-1} < X_j\}$$

where $X_{\text{tower}} \subset X_{\text{test}}$, in such a way as to minimize the cost function

$$\sum_{i=1}^{M} C(x_i, h_i), \tag{1}$$

where the tower height h_i at x_i is selected from

$$H_{\text{tower}} = \{H_k \mid k = 1, 2, \ldots Q, \text{ and } H_{k-1} < H_k\}.$$

If the foundation costs are neglected the cost function simplifies to

$$\sum_{i=1}^{M} C(h_i) \tag{2}$$

and this is to be minimized subject to the following constraints, where $x_0 = 0$ and h_0 is specified:

(*a*) a maximum single span constraint

$$x_i - x_{i-1} \leqslant s_1(x_i), \quad i = 1, 2, \ldots M, \tag{3}$$

(*b*) a maximum double span constraint

$$x_{i+1} - x_{i-1} \leqslant s_2, \quad i = 1, 2, \ldots M-1, \tag{4}$$

(*c*) an uplift constraint which sets a limit on the possible angle of swing of the line under transverse wind loading

$$x_{1,i} + x_{-1,i} \geqslant W(x_{i+1} - x_{i-1}), \quad i = 1, 2, \ldots M-1, \tag{5}$$

W being a parameter of the line constants, and

(*d*) a statutory clearance constraint to conform with the British Standard Specifications for overhead transmission lines. Over ground with no side slope let $u(x, 0) = u(x)$, so that for a span $(i, i+1)$

$$v(x) = A_{\text{hot}}(x - x_i)(x - x_{i+1}) + \frac{v_{i+1} - v_i}{x_{i+1} - x_i}(x - x_i) + v_i \tag{6}$$

where $x_i \leqslant x \leqslant x_{i+1}$, $v_i = v(x_i)$, etc., and the actual clearance

$$v(x) - u(x) > z(x). \tag{7}$$

Whenever there is a side slope or an offending structure offset to the line route $u(x, d) \neq u(x, 0)$, and the appropriate critical clearance must be specially computed. Of course, it is only feasible to ensure that the statutory clearance is not violated at a discrete set of points, the clearance sites, along the line route, but provided these sites are sensibly chosen the likelihood of an invalid span being accepted can be ignored.

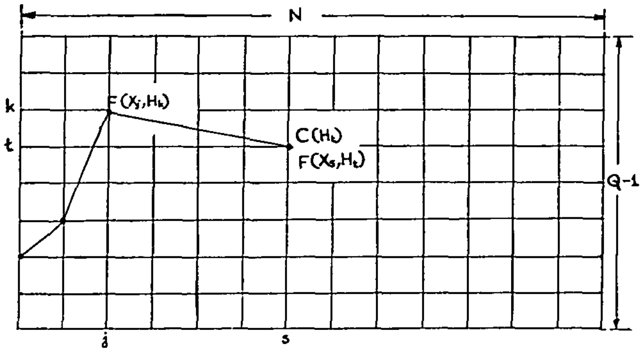

Fig. 1

This non-linear constrained optimization problem is solved by dynamic programming. It is considered as a routing problem in a finite network to which the principle of optimality (Bellman, 1957) is applied. Let

$$F(X_j, H_k), j = 1, 2, \ldots N; \text{ and } k = 1, 2, \ldots Q, \tag{8}$$

be the cost (including the cost $C(H_k)$ of the current tower) of constructing a line from the unique starting node to the tower of height H_k at site X_j using an optimal policy. The solution space is a two dimensional rectangular grid of $(N+1)Q$ nodes in which, for feasibility, not every node may be connected to every other node (**Fig. 1**).

The rule for assigning an optimal cost value $F(X_s, H_t)$ to any node (s, t) is to search for a linkage $(s, t) \rightarrow (j, k)$ satisfying the constraints of the problem and such that

$$F(X_s, H_t) = C(H_t) + \min F(X_j, H_k), \quad j = 0, 1, \ldots s-1; k = 1, 2, \ldots Q, \tag{9}$$

i.e. the minimum cost feasible linkage through a preceding node with determined optimal cost. This now specifies an optimal policy up to the node (s, t). Applying recurrence relation (9) repeatedly over the range $t = 1, 2, \ldots Q$ for $s = 1, 2, \ldots N$ successively [for $s = 0$, t is specified and $F(X_0, H_t) = C(H_t)$] a family of solutions is generated through to the end of each line section. It only remains to find $\min F(X_N, H_k)$, $k = 1, 2, \ldots Q$, and then to trace back to the origin those linkages which make up this overall optimum solution.

The data layout

The computer program starts with a data processing section designed to organize the raw input data most suitably for the subsequent application of the main algorithm. The line route is naturally divided into a number of sections from one angle tower to the next, and each such section is represented as one record in terms of a card file or, when processed, one fixed length record of a magnetic tape file.

(*a*) *Card file.* Every card in the card file has a special character marker in the first column which serves to identify the card type. This is followed by five numbers, representing, in general, the elevations at left 40 ft, at centre line and at right 40 ft, the clearance, and the chainage. The "*" character indicates a card whose information refers to a location on the route used both as a test site and as a clearance site, the "−" character indicates a card relating to a clearance site only. Such cards, arranged in ascending order of chainage, form the bulk of each section record, but at the end of the record there will be cards of two additional types. These are "+" cards detailing elevations in the neighbourhood of the last angle tower, necessary for the determination of its foundation level, and ">" cards detailing the line type, section number, angle of turn, etc.

(*b*) *Magnetic tape file.* On the magnetic tape file seven blocks are assigned to each section record which itself comprises test site, clearance site, and line detail sub-records. A test site sub-record occupies two blocks and consists of $2N$ alternate entries of chainage and the corresponding centre profile elevations. To determine the base setting levels of towers of different base widths at a given site it is only necessary to evaluate a single parameter relative to that site. This depends on the elevations of the neighbouring points and its value is computed by an interpolation routine, suitably scaled down to a decimal fraction, and then tagged on to the integral value of the site chainage. The clearance site sub-record occupies the next four blocks and 500 words of the seventh, the remaining twelve words being occupied by the line detail sub-record.

The EMA program

The data transfer, computation and decision processes are illustrated in the flow chart (**Fig. 2**) and its legends. The computation progresses by unit increments in s (loop 1) and t (loop 2), the indices of the current node, and the search for the preceding connecting node is carried out by adding unit increments to j (loop 3) and k (loop 4), the preceding tower site and height indices. These four loops constitute the core of the recursive algorithm.

In Routine R8 any trial linkage is specified as a vector $Y(y_1, y_2, y_3, y_4, y_5)$, where y_1, y_2, y_3, y_4 are the heights at which the insulators and conductors are supported at the previous and current towers and y_5 is the cost of building a line optimally up to the previous tower plus the cost of the current tower. This vector is fundamental to the testing of the linkage for feasibility and optimality. To facilitate the computation of its elements a pseudo truth table is compiled, the entries being the algebraic expressions for the vector elements determined by the values of the relevant boolean variables. Boolean/arithmetic expressions for the evaluation of the vector elements are synthesized from this table.

Optimality and sensitivity

The double span and uplift constraints (4) and (5) imply that the decision at s may not be completely justified unless the sub-optimals at feasible $j < s$ are also examined. Whenever such a constraint rules out a trial linkage, it is necessary to examine the next sub-optimal [or better still an ordered set of sub-optimals (Bellman and Kalaba, 1960)] associated with the preceding node in order to arrive at the true optimal solution up to the current node. Although the need for and the required order k of such sub-optimals cannot be predicted, they could be efficiently computed by a list processing technique. If the alternatives offered by the sub-optimals are to be considered, the network will lose its ordered tree structure and the resultant program, although powerful, will be extremely complicated.

It would appear a better strategy to make use of the sensitivity of the solution to the fineness of the grid and to seek to improve the solution by re-applying the original algorithm to a perturbation of the first solution on a still finer grid. In this way the elegance of the dynamic programming search procedure is preserved.

To study the sensitivity of the solution to variations in the average spacing between test sites a number of line sections were run with both 100 ft and 50 ft spacing. In all cases corresponding solutions had the same overall optimum cost although some towers were shifted by 50 ft and occasionally the heights of adjacent towers were interchanged in one solution relative to the other. One section of line was also run in the reverse direction, fixing the type and height of the angle tower at each end, and a solution identical with that for the forward run obtained.

In practice the solution obtained by one application, over a fairly fine grid, of the algorithm as described in the preceding section has proved to be adequate and its derivation economical of computer time.

Conclusions and results

It has been shown that the choice of the sites and heights of the towers in the design of an overhead transmission line can be posed as a combinatorial problem and solved by a discrete deterministic multistage decision technique.

The first rigorous test of the current program was made on a twenty-mile stretch of Dowlais–Cowbridge 400 kV supergrid line over hilly terrain with considerable side slope. The solution was carefully examined by C.E.G.B. from both an engineering and economic standpoint and they reported a saving of 7% on L2 type line and 5% on L6. The saving in cost per mile is thus of the order of £1000, while the computing cost per run is only £15 per mile. These results compare favourably with the savings produced by other computer solutions to the problem.

A subsequent original design exercise on a fifty-mile stretch of 400 kV line has produced even more satisfactory results, and plans to extend the scope of the program are now in hand.

Acknowledgements

We wish to thank Professor R. A. Buckingham for his encouragement and support, Mr. W. Oxburgh of Atlas Computing Service for his essential liaison work

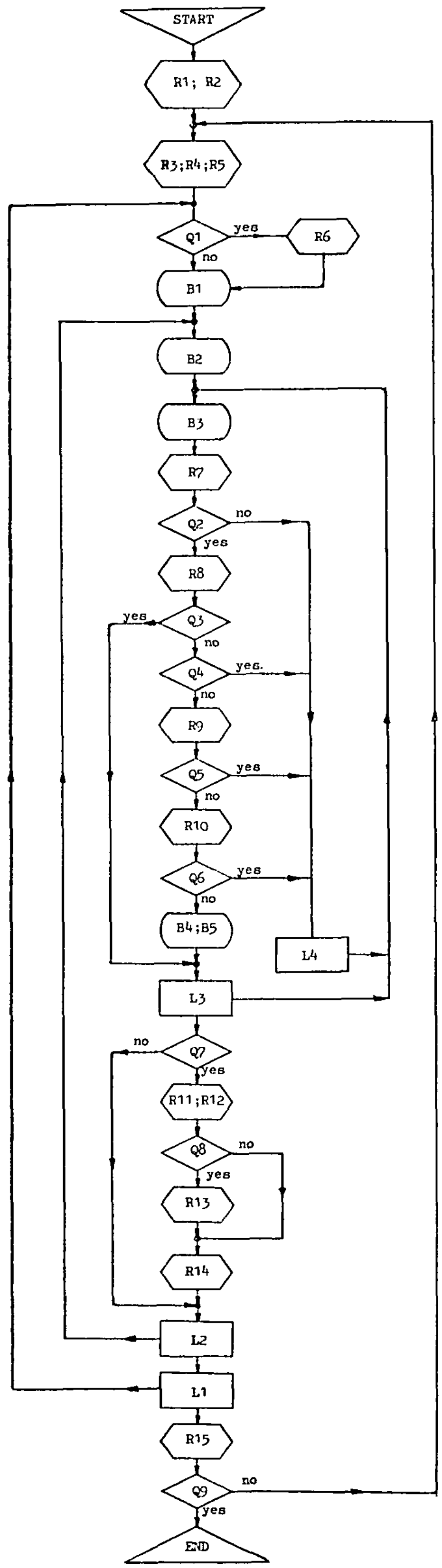

Fig. 2

Blocks

B1 Compute I and P the site indices over which the search for a previous tower is spread in the band of feasible connection.
B2 Initialize cost and solution markers before the search from a new node is started.
B3 Compute store index for the node to which connection is to be tried.
B4 Set the feasibility markers for the node solution and the site solution.
B5 Store the solution temporarily.

Decisions

Q1 Is there a change in maximum span in this section?
Q2 Is this connection worth trying, i.e. is there a feasible solution up to this node?
Q3 Is the present linkage of greater cost than a solution already obtained?
Q4 Is the double span constraint (4) violated?
Q5 Is the uplift constraint (5) violated?
Q6 Is the clearance marker reset by R10, i.e. is the clearance (7) violated?
Q7 Is there a feasible solution at this node?
Q8 Should there be a monitor output at this stage?
Q9 Is the end of line reached?

Loops

L1 Change test site index $1(1)N$.
L2 Change current tower height index $1(1)Q$.
L3 Change previous tower site index $1(1)P$.
L4 Change previous tower height index $1(1)Q$.

Subroutines

R1 Input the tower details.
R2 Generate the starting values controlled by input data.
R3 Bring down the section record from magnetic tape.
R4 Process the clearance-terrain data to transform them to elevation and side slope parameters.
R5 Initialize the starting parameters.
R6 Modify the span limit as required.
R7 Decode a solution with "unpack" instructions.
R8 Compute the elements of the connection vector Y.
R9 Compute the sag curve at 22 F° and check the uplift constraint.
R10 Compute the sag curve at 122 F° and check the clearance constraint.
R11 Re-store the solution prior to final storage.
R12 Mark the solutions to ensure the ordering of cost with height.
R13 Stepwise monitor output of the solution.
R14 Store a solution with "pack" instructions.
R15 Trace back linkage and output solution for each section.

with C.E.G.B., and Mr. H. V. Flaxman and his staff of the Transmission Project Group, C.E.G.B. for supplying the test data, for answering our many queries, and for their critical appraisal of the solutions proposed. To Mr. A. Wren of the University of Leeds we are indebted for much helpful criticism of an earlier draft of this paper.

References

AHLBORG, F., and PALM, Y. (1962). Optimum Tower Spotting on Transmission Lines by means of Electronic Computer, *Blue-White Series* No. 33, Swedish State Power Board.

BARTELINK, L. H. (1964). Spotting of Transmission Towers by Computer, *Canadian Engineering Journal*, July 1964, p. 25.

BELLMAN, R. (1957). *Dynamic Programming*, Princeton University Press, Princeton, p. 83.

BELLMAN, R., and KALABA, R. (1960). On *k*th Best Policies, *SIAM Journal*, Vol. 8, p. 582.

CONVERTI, V., HYLAND, E. J., and TICKLE, D. F. (1962). Optimized Transmission Tower Spotting on Digital Computer, *A.I.E.E., Power Apparatus and Systems*, No. 59, p. 55 (Trans. paper 60–1201).

HOARE, P. J., and MORWOOD, C. H. (1964). Overhead Transmission Line Optimized Tower Positioning by Digital Computer, *Journal of the Institution of Engineers, Australia*, Vol. 36, p. 115.

POPP, R. E., DABEKIS, C. J., and FULLERTON, F. M. (1963). Electronic Computer Program Permits Optimized Spotting of Electric Transmission Towers, *A.I.E.E. Power Apparatus and Systems*, No. 66, p. 360 (Trans. paper 63–64).

RANYARD, C. J. (1963). A Method to Minimize the Cost of Erecting the Towers in an Electric Power Transmission Line, Post-graduate Diploma Dissertation, University of Leeds, 1963, (see also this *Journal*, Vol. 10, p. 157).

SHULMAN, H. (1962). The Pole Spotting Problem, *SIAM Review*, Vol. 4, p. 12.

STAGG, G. W., and WATSON, M. (1964). Dynamic Programming—Transmission Line Design, *I.E.E.E. International Convention Record* (U.S.A.), Vol. 12, pt. 3, p. 55.

Book Reviews

Introduction to Dynamic Programming, by G. L. Nemhauser 1966; 256 pages. (New York: *John Wiley and Sons, Inc.*, 64s.)

An Introduction to Dynamic Programming—The Theory of Multistage Decision Processes, by O. L. R. Jacobs, 1967; 126 pages. (London: *Chapman and Hall Ltd.*, 30s.)

For a long time the field of dynamic programming was dominated by the pioneering works of Richard Bellman and his colleagues at the Rand Corporation. With increasing interest in the subject the range of authorship has widened and in the last few years a number of good books have appeared from other sources; the two under review are, in their different ways, welcome additions.

Dr. Nemhauser is evidently an enthusiast for dynamic programming and has written an outstanding book on both the theoretical and computational aspects of the subject, aimed at practising or aspiring operational research workers, management and social scientists, and engineers.

After a brief introduction on mathematical model building, the dynamic programming approach to optimization, and optimization techniques in general, he sets out to show when it is theoretically possible to use dynamic programming, considering first single-stage decision problems, then deterministic multistage problems with a finite number of stages and their possible decomposition into an equivalent series of single stage systems, and leading to the fundamental recursive equations. The following two chapters (Basic Computations and Computational Refinements, over 100 pages in all) show, by considering various models, how to transform a system model into multistage form and how to solve the recursive equations most efficiently; they contain a wealth of useful information, including three detailed flow diagrams.

The remaining chapters are Risk, Uncertainty, and Competition, an extension to stochastic and competitive models of the results obtained so far; Nonserial Systems, which examines processes with branches and feedback loops; and an elementary treatment of Infinite-Stage Systems. Finally, there is a quick review of applications.

Many worked examples are included in the text but for a proper understanding of dynamic programming some personal computational experience is necessary and, for the serious reader, there are nearly a hundred worthwhile exercises to choose from.

This well written book, with its useful bibliography, appears likely to remain my first choice on the subject for some time to come and will certainly be recommended to my own graduate students.

Dr. Jacobs' more modest book cannot help but suffer in comparison with the work just discussed. Nevertheless it is a good introduction to dynamic programming and must be recommended as such. It is meant for undergraduates and graduates new to the subject and presents the standard procedures, through a series of well chosen examples, in a natural order—discrete deterministic multistage decision processes, continuous deterministic multistage decision processes, stochastic decision processes, and adaptive decision processes. Unfortunately there are no exercises and the bibliography is barely adequate, although a few additional references are scattered throughout the text.

Both books rely on only a moderate level of mathematical attainment on the part of the reader.

K. WOLFENDEN (London)