Minimum Cost Structures by Dynamic Programming

FRED MOSES and GEORGE GOBLE

IN THE DESIGN of structures it is necessary to consider an array of possible choices in sizing elements such as beams and columns in finding an acceptable and economical design. It is only with the recent access to digital computers that designers can be relieved of some of the tedious calculations associated with structural design decisions. In further extending computer aided design, digital computers have also been used to directly determine optimum or least cost designs. Applications in steel structures have included bridge girders, wide flange beams, grillages and ship structures.[1,2,3]

One important design area that has received little attention is the optimizing of the total fabrication and material costs for a structural system composed of many elements such as multistory frames, transmission towers, and grillages. Fabrication costs can be reduced if many members or elements are constructed of a similar cross-section. This causes a penalty in extra material, since a minimum weight design often has members of many different cross-sections. This paper describes the use of a simple procedure known as Dynamic Programming to find the optimum selection of member sizes, which minimizes the overall cost including material and fabrication costs.

Dynamic Programming has previously seen application in structural design and construction decisions. It is used in the GAD program developed at Case Western Reserve for the Ohio Department of Highways to design highway bridge girders, including selecting steel cross-section and material strength.[1] It has also been used for optimum plate selection for ship structural design.[3] In construction management Dynamic Programming has been used in conjunction with C.P.M. and PERT techniques. Before describing the Dynamic Programming method, the present design problem will be described in more detail with an example.

Fred Moses is Associate Professor of Engineering, Case Western Reserve University, Cleveland, Ohio
George Goble is Professor of Structures, Case Western Reserve University, Cleveland, Ohio

DESIGN PROBLEM — BEAM SELECTION

The automated design programming to be established is deciding on the minimum cost sizes of beams or other elements to be fabricated for a given job. It assumes that, through either hand calculation or a computer design program aided by STRESS or STRUDL, the minimum allowable beam sizes have been determined. From a weight viewpoint the best design would be these minimum size beams. In many steel structures, because of fabrication costs, there are considerable savings from making many beams or elements of a similar size. This causes a weight penalty, but makes up for it by a cost saving in fabrication and erection. The Dynamic Programming procedure to be described sorts through the large number of possible combinations of beam selections and finds the minimum cost choice.

An important task for the designer and fabricator is to specify the magnitude of fabrication savings as a result of similarity in beam or element size. This is particularly true in constructions such as power plants with varying length beams in the frame, in which the saving due to having two different length beams be the same size is difficult to assess.

Example—Consider a structure with 15 beams to be fabricated. Based on layout and loading, the structural analysis has determined that there are five different beam sizes to be used, as shown in Table 1 with their respective steel costs per beam and the number of each

Table 1. Beam Design—Requirements and Costs

Beam Number (in order of decreasing weight)	Cost per Beam (assuming only one is fabricated)	Number of Such Beams in Original Minimum Size Selection
1	100	3
2	90	5
3	85	4
4	70	1
5	60	2

beam size required. A reasonable question from the designer or fabricator is whether or not the number of different beams to be fabricated can be reduced below the five specified. Thus, in some locations a heavier beam than specified will be used. This raises the steel cost for the structure, but reduces the overall cost by cutting fabrication and erection costs.

The type of information required from fabricators on cost reduction factors from using similar beams is shown in Table 2. The numbers are presented to illustrate the technique and may not necessarily be representative of an actual steel fabrication job. The overall percent saving levels off as the number of beams increases.

An example of a possible design is for the beams to be fabricated according to the original beam listing in Table 1. Then the total cost, C, is:

$$C = 3(100)(1 - 0.10) + 5(90)(1 - 0.14) + 4(85)(1 - 0.12) + 1(70)(1 - 0) + 2(60)(1 - 0.07)$$
$$= 270 + 387 + 299 + 70 + 112 = \$1{,}138$$

If all beams are fabricated as the heaviest size, the cost will be

$$C = 15(100)(1 - 0.22) = \$1{,}170$$

The optimum selection will be determined from the dynamic programming procedure which will be described in the following section.

Table 2. Fabrication Cost Reduction Factors

Number of Similar Beams Fabricated	Overall Percent Cost Reduction
1	0
2	7
3	10
4	12
5	14
6	16
8	19
9	20
10	21
12	22
15	22

DYNAMIC PROGRAMMING

Dynamic Programming has found many applications in systems optimization. For the purposes of this paper, the Dynamic Programming will be specialized for this particular type of selection problem. It may be stated as finding the values of x_1, x_2 up to x_n, which:

$$\text{Minimize } f = \sum_{i=1}^{n} f_i(x_i) = f_1(x_1) + f_2(x_2) + \ldots + f_n(x_n) \tag{1}$$

subject to a constraint

$$x_1 + x_2 + \ldots + x_n = b \tag{2}$$

In the beam selection program, $f_i(x_i)$ represents the cost of the beams of size i, including fabrication reductions due to using x_i of such beams. Each x_i indicates the number of beams of the given size to be fabricated. The constraint in Eq. (2) insures that all beams are to be fabricated. Since there is only one constraint, a classical Lagrange multiplier approach would seem feasible for minimization. In general this is not possible because of relative minima, discrete variable requirements and discontinuities in the variables and objective cost functions.

The Dynamic Programming (D.P.) solution searches the allocation of the x_i one at a time (hence the term dynamic) for a set of discrete values denoted by z, covering the range from 1 to the largest number of beams being fabricated for the job. This is done by constructing a sequence of cost functions $b_k(z)$, which consecutively adds each variable x_k to the solution. The function $b_1(z)$ is the cost at Stage 1 and considers only variables of size 1. The function $b_2(z)$ is Stage 2 and considers the allocation of both sizes 1 and 2. The kth stage is $b_k(z)$ and incorporates the selection of variables 1, 2, . . ., $k - 1$ and k. The cost function $b_k(z)$ is then the minimum cost if only k variables are used for a number of beams equal to z. The solution $b_k(z)$ is thus defined by:

$$b_k(z) = \min \sum_{i=1}^{k} f_i(x_i) \tag{3}$$

such that:

$$x_1 + x_2 + \ldots + x_k = z \tag{4}$$

The function $b_k(z)$ can be found by comparing values of $b_k(z)$ making use of $b_{k-1}(z)$ for all possible values of x_k. For example, the cost over the range x_k is:

$$b_k(z) = \min\,[f_k(x_k) + b_{k-1}(z - x_k)] \tag{5}$$

Equation (5) can be explained as follows: The function $b_k(z)$ is defined as the minimum cost using only the first k variables for a value z. The cost for the value z will be the cost of including variable x_k, which is $f_k(x_k)$ plus the cost of the remaining $k - 1$ variables optimally used to a number of elements equal to $z - x_k$. The optimum allocation of these $k - 1$ beams to a level $z - x_k$ already was found at the $k - 1$ stage and is $b_{k-1}(z - x_k)$. It is the need to use $b_{k-1}(z - x_k)$ that requires finding allocation functions over a continuous range of z values. To find $b_k(z)$, substitute the permissible values of x_k into Eq. (5) and find the value of x_k which minimizes the cost function. The evaluation of the $b_k(z)$ function is repeated for the range of z values to give a complete function. The procedure starts with Stage 1.

$$b_1(z) = f_1(x_1) \tag{6}$$

Since only activity 1 is available, no comparison need be made in Eq. (6).

For the range x_2,

$$b_2(z) = \min [f_2(x_2) + b_1(z - x_2)] \quad (7)$$

and finally, for the range x_n,

$$b_n(z) = \min [f_n(x_n) + b_{n-1}(z - x_n)] \quad (8)$$

The last function $b_n(z)$ gives the optimum cost when all variables are considered. In particular, setting z equal to the required number of elements b in Eq. (2) gives the optimum selection. The selection variables which produce the optimum $b_n(b)$ must be stored as the values of $x_1, x_2, \ldots, x_n$, which minimize each of the $b_k(z)$ functions. The minimization is simply done by substituting the possible values of x_k and comparing cost values. The D.P. procedure uses Bellman's "principle of optimality" which states: An optimal policy has the property that whatever the initial state and initial decisions are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.[4]

SOLUTION TO BEAM SELECTION

To find the lowest cost design, the D.P. allocation procedure is started from the heaviest beam first. The sequence is constructed of cost function $b_k(z)$, which was defined as the lowest cost selection using the beam categories 1 through k for z numbers of beams, where z will range from 1 to the maximum number of beams to be fabricated. The term k will range from 1 to the maximum number of beam sizes available to be considered for the structure.

Let

x_i = number of beams of category i, where i = 1, 2, 3, 4, and 5.

z = number of beams allocated

For simplication and to illustrate the calculations, the variables will be considered on the basis of the groups of design beams according to their size. Using Table 1, the minimum permissible size for each group of beams is:

$z = 1$ (3 beams)
$z = 2$ (8 beams)
$z = 3$ (12 beams)
$z = 4$ (13 beams)
$z = 5$ (15 beams)

Then, $b_k(z)$ is the minimum cost for z beams if beam sizes 1, 2, ...k are considered. The term k will range from 1 to 5 in this example. Selection will be in the order of heaviest beam first.

Table 3. Minimum Cost Allocations

Number of Beams z	$b_1(z)$	$b_2(z)$	$b_3(z)$	$b_4(z)$	$b_5(z)$
1	270				
2	647	647			
3	935	918	918		
4	1,012	980	980	980	
5	1,170	1,110	1,110	1,107	1,092

Thus:

$b_1(1)$ = (3 beams)($100/beam)(cost reduction 1 − 0.10) = $270

$b_1(2)$ = (8) (100)(1 − 0.19) = $647
$b_1(3)$ = (12)(100)(1 − 0.22) = $935
$b_1(4)$ = (13)(100)(1 − 0.22) = $1,012
$b_1(5)$ = (15)(100)(1 − 0.22) = $1,170

As in the general D.P. allocation in Eq. (7), over the range x_2,

$$b_2(z) = \min [f_2(x_2) + b_1(z - x_2)]$$

Thus, $b_2(1)$ = $270, since beam sizes x_2 are too small to be used in the first group ($z = 1$). The minimum value for $b_2(2)$ over the range x_2 is determined from the expressions

$x_2 = 0$: Cost = $b_1(2)$ = 647
$x_2 = 1$: Cost = (5)90 (1 − 0.14) + $b_1(1)$
= 386 + 270 = 656

Therefore,

$b_2(2)$ = $647 ($x_2 = 0$)

Similarly, for $b_2(3)$, over the range x_2,

$x_2 = 0$: Cost = (12)100(1 − 0.22) = $935
$x_2 = 1$: Cost = 4(90)(1 − 0.12) + $b_1(2)$ = $964
$x_2 = 2$: Cost = 9(90)(1 − 0.20) + $b_1(1)$ = $918

Therefore, the minimum is $b_2(3)$ = $918 ($x_2 = 2$, $x_1 = 1$, i.e., 3 beams @ Type 1 and 9 beams @ Type 2).

Continuing with the allocation procedures until all beam sizes have been considered gives the results in Table 3.

After going through the allocation of beam No. 5, the optimum selection has a cost of $1,092 and a beam schedule of:

3 @ Type 1	(cost = $270)
10 @ Type 2	710
2 @ Type 5	112
	$1,092

The minimum cost design for the example, \$1,092, compares with \$1,170 if only beams of Type 1 are used and \$1,137 if the design is based on minimum weight.

It is too soon to comment on this program's full capability or the actual construction saving that can be realized by its utilization. It does represent an efficient and easy programming technique for automating a decision which would otherwise require exhaustive calculations and comparisons. The procedure is easily programmed for the digital computer and requires relatively little computation or storage requirements. Further development of the program awaits the provision of descriptive fabrication cost estimates which for the example problem may not necessarily be representative. Only the percentage cost reductions are needed and not the actual cost values. One way to develop the cost reduction factors may be to vary these fabrication values for typical designs until a desired beam schedule is obtained. These factors can then be used for other designs.

REFERENCES

1. *Goble, G. G. and DeSantis, P. V.* Optimum Design of Composite Girders with Mixed Steels *Journal of the Structural Division, ASCE, Dec. 1966.*
2. *Moses, F. and Onoda, S.* Minimum Weight Design of Structures with Application to Elastic Grillages *Intl. Journ. for Num. Meth. in Eng.*, *Vol. 1, 1969.*
3. *Moses, F. and Tonnessen, A.* Dynamic Programming for Computing Optimal Plate Dimensions in Some Ship Structures *European Shipbuilding, No. 4, 1967.*
4. *Bellman, R. and Dreyfus, S.* Applied Dynamic Programming *Princeton University Press, 1962.*