JANAKI AMMAL, E. K. and KHOSLA, S. N.: Breaking barrier to polyploidy in the genus *Eucalyptus*. Proc. Indian Acad. Sci. B LXX Pt. VIII. 248—249. (1969). — KAPOOR, M. L. and KEDHARNATH, S.: Induced tetraploidy in Sandal Wood and *Eucalyptus*. Indian J. Genet. and Pl. Br. 34A, 408—412 (1974). — NILSSON-EHLE, H.: Über eine in der Natur gefundene Gigasform von *Populus tremula*. Hereditas **21**, 379—382 (1936). — RUGGERI, C.: Karyological notes on the genus *Eucalyptus* Proc. 5th World Forest Congress, Seattle U.S.A. II, 753—754 (1960). — WRIGHT, J. W.: Hybridization between species and races. Unasylva **18**, 30 (1964). — WRIGHT, J. W.: Introduction to Forest Genetics. 463 pp. Academic Press, New York (1976).

A Flexible Computer Algorithm for Designing Seed Orchards

By P. L. MARSH

Systems Analyst, School of Forest Resources,
North Carolina State University, Raleigh, N.C., USA

(Received 6th June 1984)

Abstract

The seed orchard design problem is shown to be equivalent to the three-dimensional traveling salesman problem found in Operations Research literature. The primary constraints, restricting inbreeding and promoting outcrossing, are used to create a "cost" matrix upon which an implicit enumeration scheme is used to find the solution. The resulting algorithm yields promising results for creation of basic seed orchard layouts. Extensions of the algorithm to design more complex orchard layouts are also discussed.

Key words: inbreeding, outcrossing, clonal performance, operation research permutated neighborhood, advanced generation breeding orchards.

Zusammenfassung

Das Problem der Anlage von Samenplantagen wird dem Problem des "Three-dimensional-traveling-Salesman" gleichgesetzt, das in der Operations-Forschungsliteratur zu finden ist.

Die primären Zwänge, Inzucht zu beschränken und Fremdung zu begünstigen, werden benutzt, um eine Kostenmatrix zu erstellen, auf die ein stillschweigend inbegriffenes Nummerierungsschema angewendet wird, um eine Lösung zu finden. Die daraus resultierenden algorithmischen Erträge versprechen Resultate, die zur Entwicklung von grundlegenden Samenplantagen-Anlagen dienen können.

Die Ausmaße des Algorithmus, um mehr komplexe Samenplantagen zu entwerfen, werden ebenfalls diskutiert.

Introduction

Tree improvement programs have become an integral part of plantation management of forest trees. Even though emphasis has recently been given to genetic engineering and vegetative propagation, the genetically improved plant material for most programs will continue to be obtained from seed orchards. While many criteria should be considered before designing an orchard *(Table 1)*, the objective is to randomly assign a set of clones to an area while simultaneously separating related ramets by a minimum distance to inhibitit inbreeding and varying the neighborhoods surrounding the ramets of each clone to promote outcrossing.

Since seed orchards are costly to establish, maintain and evaluate (JETT and ZOBEL, 1977), care must be taken to design the orchard for optimum seed production. Some of the earliest orchards consisted of row-plots, each involving ramets from a single clone, and completely random designs. These plans often led to inbreeding from inadequate separation among ramets of a clone. Ensuing simple fixed designs such as systematic (GIERTYCH, 1965) or shifting-block layouts (MALAC, 1962) were easily implemented but also failed to satisfy many of the design constraints of an orchard. Other designs such as the Fisher balanced lattice (LANGNER and STERN, 1955), cyclic balanced incomplete block (FREEMAN, 1969), and directional cyclic balanced incomplete block (DYSON and FREEMAN, 1968) quickly fell into disfavor because they are difficult to employ.

In 1967, LA BASTIDE devised a computer algorithm which develops a "permutated neighborhood" design that achieves most of the objectives of a seed orchard. In reality, it is a completely random design upon which restrictions are imposed to reduce inbreeding and increase outcrossing. The major drawback of the algorithm was its excessive computer runtime. In recent years, BELL and FLETCHER (1978) and HATCHER and WEIR (1981) have also employed computers to facilitate the tedious layout of more sophisticated "permutated neighborhood" seed orchard designs, but none of the methods fully meets the requirements of a seed orchard. In particular, the methods usually relax the outcrossing constraint and are adapted to a particular subset of the constraints listed in *Table 1*. The algorithms are inflexible and would require substantial modification time to accomodate a different subset of constraints.

The need exists for a flexible computer algorithm that can be employed regardless of the set of constraints. Computer design of seed orchards consists of two steps: (1) definition of a set of constraints to be adhered to during the design process, and (2) formulation of an efficient and stable algorithm that utilizes the set of constraints and produces a seed orchard layout having the desired qualities. The topic of this treatise is the description of an algorithm that can be used to design an orchard consisting of unrelated clones. Extensions to the algorithm to address more complex orchard designs are presented in the Conclusions.

Mathematical Formulation

The most commonly mentioned constraints for development of a seed orchard design are promoting outcrossing and restricting inbreeding *(Table 1)*. Outcrossing should be promoted to capitalize on genetic gain in seed orchards where progeny test information of the clones is lacking (ZOBEL *et al.*, 1958). The potential genetic gain can be greatly impaired if inbreeding freely occurs (GANSEL, 1971). The remaining constraints can be categorized either as exten-

Paper No. 9307 of the Journal Series of the North Carolina Agricultural Research Service, Raleigh, North Carolina 27695.

Table 1. — Design Constraints of a Forest Seed Orchard.

(1) Require little mathematical background to implement
(2) Inhibit inbreeding
(3) Promote outcrossing
(4) Permit comparisons of clonal performance
(5) Permit adaptation to any land shape or size
(6) Allow expansion
(7) Handle any number of clones and ramets per clone
(8) Accommodate an adequate number of clones to reduce the risk of graft-scion or cone production failure
(9) Permit use of related clones
(10) Allow unrestricted development of trees
(11) Permit efficient orchard management
(12) Allow for special constraints (such as a prevailing wind or an orchard composed of clones from a dioecious species)

sions of the inbreeding and outcrossing contraints or as restrictions based on orchard management decisions. Thus, they affect the mechanics of the algorithm but not its formulation.

The first step in writing an efficient computer algorithm for seed orchard design is the conversion of the constraints listed in *Table 1* into mathematical statements. In this treatise, only the derivation of the algorithm for an orchard with unrelated clones and with an equal number of ramets per clone is presented. Recommendations for extensions to the algorithm to accommodate different constraints are in the conclusions.

It is assumed that the algorithm will only be employed on clonal populations for which an acceptable layout exists and that the algorithm will find such a solution. Presently, no algorithm employs an 'a priori' check of the orchard parameters to ascertain the existence of a solution. Furthermore, none of the algorithms, including the one presented in this treatise, can guarantee that even if a feasible solution exists, it will be found within a reasonable time. Additional research is needed to solve these problems.

All permutated neighborhood algorithms presently in use design an orchard layout one planting position at a time. At each planting position, a scan is made of the previously assigned positions to ascertain which of the clones can occupy the n-th planting position without violating the constraints. A drawback to this method is that the clonal frequency distribution might become so unbalanced that completion of the layout is infeasible. An unbalanced distribution of the clonal ramets can lead to a bias when statstically analyzing clonal performance. This problem can be overcome by creating a linear arrangement of a set of ramets, at most one from each clone, at each interation of the algorithm. The sets, called templates, are aligned in a parallel fashion within the orchard ensuring that the ramets of a clone are evenly dispersed over the layout and will reduce the chances of the clonal distributions becoming unbalanced. The orchard can be tesselated[1] (GOLOMB, 1966, 1970) by a set of templates if the templates can be interfaced according to the constraints. The ensuing task is to construct the templates.

```
                 first template
            6    3    1    5    4    2

                   Matrix P
            new template position
            1    2    3    4    5    6
         1  1    0    0    0    1    1    3
         2  1    1    1    1    0    0    4
Clone    3  0    0    0    1    1    1    3
Number   4  1    1    1    0    0    0    3
         5  1    1    0    0    0    1    3
         6  0    0    1    1    1    1    4
            4    3    3    3    3    4
```

Figure 1. — Construction of the matrix P for an orchard with six clones.

In practice, inbreeding is sufficently controlled by maintaining an acceptable distance between related ramets. If there are k clones and t ($< = k$) are to be assigned to the template, a k by t matrix, called P, can be constructed to contain the inbreeding information. An element p_{il} of P is equal to one if clone i can occupy the l-th position of the template without violating the inbreeding constraint, and is zero otherwise. The row sums of P reflect the ease with which the clones can be assigned to the template. The column sums give an indication of the template positions that will be the most difficult to fill. *Figure 1* illustrates the construction of P given that 10 meters is the planting distance between trees and that 22 meters must be kept between related ramets. The top half of the figure represents the first template. Because of the minimum distance rule, the clones in positions one (clone 6) and two (clone 3) cannot occupy the first position in the next template. Thus the sixth and third elements in the first column of P are zero and the other elements in that column are one. The other columns of P are similarly constructed. The column sums of P indicate that positions one and six of the new template are the easiest to fill. The row sums indicate that clones 2 and 6 are restricted from fewer template positions by the minimum distance rule than the other clones.

The outcrossing constraint can be stated in a similar fashion as the inbreeding constraint. The variable n_{ij} has the value one when clone i has not been planted adjacent to clone j, and is zero otherwise. Given there are k clones, the k by k matrix, N, is formed by the n_{ij}'s. If there are no extenuating circumstances affecting the pollen dispersal (such as a prevailing wind), N is symmetric. Initially, this matrix will contain all ones except for zeroes along the main diagonal (clone i can never be planted adjacent to itself). As clones are assigned to a series of templates, zeroes replace the appropriate matrix elements. When this matrix becomes sufficiently sparse, it is reinitialized. The amount of outcrossing control exercised is determined by the frequency of reinitialization. *Figure 2* illustrates the construction of N. Note that since clones 3 and 5 are planted adjacent to clone 1 in the first template, elements (1,3), (3,1), (1,5), and (5,1) were changed to zero when updating N. These two clones will not be allowed to be assigned next to clone 1 again until N is reinitialized. The other rows of N were updated in a similar fashion.

The outcrossing information in matrix N can be implicitly combined with the inbreeding information in matrix P to form the k by k by t matrix C. The (i, j, l)-th element of C is obtained by multiplying the l-th element in the i-th row of P by the j-th element in the i-th row of N. The elements c_{ijl}, of C equal one if the i-th clone can occupy the l-th position of the template and can be planted adjacent to the j-th clone without violating the outcrossing contraint, and are zero otherwise. *Figure 3* illustrates the first cross-section of C constructed from the P and N matrices appearing in *Figures 1* and *2*, respectively. The first row of N is multiplied by the (1,1) element of P to yield the first row in the cross-section in C. The other rows of N are multiplied by the corresponding element in the first column of P. This cross-section indicates the clones that can occupy the first position of the next template.

The problem has now been reduced to finding a set of t elements of the matrix C each having the value one. It can be viewed as a three-dimensional traveling salesman

[1] Definition: Tesselate - to make a mosaic of a plane using small rectangular pieces.

```
              Matrix N (initial)
               neighboring clone
            1    2    3    4    5    6
          1 0    1    1    1    1    1
          2 1    0    1    1    1    1
Clone     3 1    1    0    1    1    1
number    4 1    1    1    0    1    1
          5 1    1    1    1    0    1
          6 1    1    1    1    1    0

                first template
            6    3    1    5    4    2

              Matrix N (updated)
              neighboring clone
            1    2    3    4    5    6
          1 0    1    0    1    0    1
          2 1    0    1    0    1    1
Clone     3 0    1    0    1    1    0
number    4 1    0    1    0    0    1
          5 0    1    1    0    0    1
          6 1    1    0    1    1    0
```

Figure 2. — Construction of the matrix N for an orchard with six clones.

```
          Matrix P                                  Matrix N
      template position                         neighboring clone
      1    2    3    4    5    6              1    2    3    4    5    6
c n 1 1    0    0    0    1    1      c n 1 0    1    0    1    0    1
l u 2 1    1    1    1    0    0      l u 2 1    0    1    0    1    1
o m 3 0    0    0    1    1    1      o m 3 0    1    0    1    1    0
n b 4 1    1    1    0    0    0      n b 4 1    0    1    0    0    1
e e 5 1    1    0    0    0    1      e e 5 0    1    1    0    0    1
  r 6 0    0    1    1    1    1        r 6 1    1    0    1    1    0

                    First cross-section of C
                    0    1    0    1    0    1
                    1    0    1    0    1    1
                    0    0    0    0    0    0
                    1    0    1    0    0    1
                    0    1    1    0    0    1
                    0    0    0    0    0    0
```

Figure 3. — First cross-section (the elements with 1 for their third subscript, i.e. template position) of C for an orchard with six clones.

problem like the one described by *Figure 4* (COOPER and STEINBERG, 1974). A feasible solution can be found if and only if (a) precisely t x_{ijl}'s with the value one can be selected and (b) the x_{ijl}'s with value one have a permutation of the integers 1, 2, . . . , k for their i-th subscript (and therefore the j-th subscript) and a permutation of the integers 1, 2, . . . , t for their l-th subscript. Since the three dimensional matrix C satisfies the Euclidean triangularity property, we are assured that an optimal (in this case, feasible) solution of the problem assigns each clone to the template no more than once (HARDGRAVE and NEMHAUSER, 1962). Constraint # 1 in *Figure 4* also ensures that the solution is a t-size tour of the clones. If $k > t$, then a t-sized subset of the k clones will be assigned to the template and the remaining k-t clones will not be assigned. If $k=t$, then each of the k clones will be assigned once to the template. Note that t can not exceed k and that the u_i's are nonrestricted variables, although no additional work is required if they are nonnegative (Wagner, 1969).

Method of Solution

The basic premise of the solution algorithm is that there

$$\text{Max} \sum_{l=1}^{t} \sum_{j=1}^{k} \sum_{i=1}^{k} C_{ijl} X_{ijl}$$

$$\begin{aligned} \text{s.t. } x = 0,\ 1 \text{ for all } i &= 1,\ 2,\ \dots,\ k \\ j &= 1,\ 2,\ \dots,\ k \\ l &= 1,\ 2,\ \dots,\ t \\ t &\leq k \end{aligned}$$

$$(1)\ u_i - u_j + nX_{ijl} \leq n - 1$$

$$\sum_{l=1}^{t} \sum_{j=1}^{k} X_{ijl} = 1 \text{ for } j = 1,\ 2,\ \dots,\ k$$

$$\sum_{l=1}^{t} \sum_{j=1}^{k} X_{ijl} = 1 \text{ for } i = 1,\ 2,\ \dots,\ k$$

Figure 4. — Mathematical Formulation of the Seed Orchard Design Problem.

exists a set of templates that can tesselate the orchard plot. The templates are permutations of the set of clones that satisfy the following two primary constraints:

1) No related ramets are within a set distance of each other. This will be referred to as the minimum distance rule (to inhibit inbreeding).
2) The neighborhoods around the ramets of each clone are different permutations of the remaining clones (to promote outcrossing).

The algorithm begins at one corner of the orchard and overlays the first template along the length of the rectangle. Additional templates are placed parallel to the first until all the rows along the orchard width are covered. If necessary, the algorithm returns to the top of the orchard and assigns templates to the next set of planting positions along the orchard's length. This process is repeated until the whole rectangle is tesselated. Graphically, the tesselation process is illustrated by *Figure 5*. Since many templates will be required to tesselate the rectangle, an efficient method must be derived to construct the templates.

The assignment of the clones within the template is a sequencing problem. No clone can be represented within a template more than once. Forcing the size of the template to be as large as possible (without exceeding the number of clones) will ensure that the ramets of a given clone are uniformly dispersed over the width of the orchard, use each clone approximately the same number of times within the orchard, and lessen the probability that either the minimum distance rule will be violated or the number of ramets per clone will become unbalanced. An even dispersal ensures that the statistical evaluation of clonal performance will not be biased by unknown variations of the orchard plot. The analysis will follow an incomplete block

SAS

	planting postions 1, 2, ... t,	← if necessary → t+1, ... 2t,	2t+1
ROW 1	1st template	N+1 template	
ROW 2	2nd template	N+2 template	
ROW N	Nth template	2*N template	

Figure 5. — Illustration of Tesselation of an Orchard with t Clones.

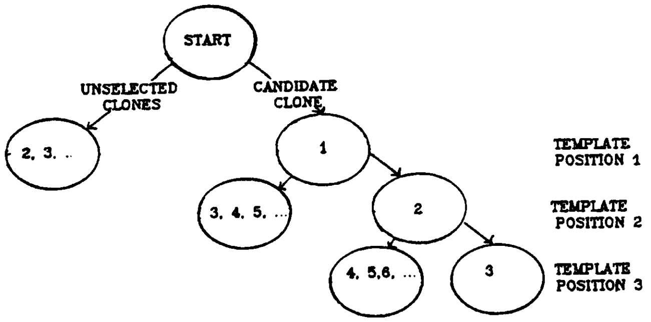

Figure 6. — Illustration of the Solution Tree.

design (if t=k, a randomized complete block design should be used).

The algorithm uses an implicit enumeration scheme similar to Balas' (1965) to find a solution to the sequencing problem. Starting at the top level of the solution tree, the algorithm selects the first clone that is eligible for the first position in the template. A clone is eligible for a position in the template if it is feasible for the position and has not been previously assigned to the template. This selection, as it is for all positions, is represented as a right branch on the solution tree *(Figure 6)*. The number of right branches from the top of the tree to the node is equal to the clone's position in the template. After the first position in the template has been filled, the algorithm tries to assign a clone to the second position. The clone must comply with the outcrossing constraint for the first clone, must not violate the minimum distance rule, and must not already have been used in the template. A successful placement adds another right branch to the solution tree. The algorithm branches right until either (1) a feasible solution, signified by a path of t right branches, has been found, or (2) one of the constraints bars passage to the next node. In the first case, the solution is assigned to the orchard layout and the algorithm proceeds accordingly. The second case signifies a failure to find an acceptable clone at a given position and causes the algorithm to return to the previous node and a branch left is executed. At the left branch, any of the clones that have not been selected (as signified by the nodes on the right branches on the path to the top of the tree) are eligible for selection. The next clone that meets the criterion for assignment is used and the algorithm proceeds in search of a feasible solution. If the algorithm has tried all possible clones for the first position and still cannot find a solution, the matrix N is reinitialized. By relaxing the outcrossing constraint, the cost matrix C can be redefined so that a feasible solution to the problem might be found. If a solution can now be found, the algorithm proceeds with tesselating the remainder of the plot. If no solution is found, the algorithm prints an error message and aborts since the minimum distance rule bars the construction of the new template. Empirically, the probability of this event occuring in practice is thought to be small. The algorithm will terminate in a finite number of steps since there are a finite (n!) number of nodes in the solution tree and no node is fathomed more than once. This method does not guarantee that the whole orchard will be tesselated even though a feasible tesselation might exist. As with other permutated neighborhood algorithms, the only recourse is to re-execute the algorithm.

Conclusions

The success of failure of the algorithm depends on whether an orchard layout that satisfies all the constraints can be designed in a reasonable time. By enforcing the minimum distance rule, the algorithm guarentees to inhibit inbreeding. The mechanics of the algorithm ensures that the orchard is compatible with secondary constraints such as expandability and the efficient use of land, clonal, and human resources. Since these constraints are satisfactorily met by the algorithm, the only other constraints that must be confirmed are the statistical comparison constraint (i.e. good dispersion of the ramets over the orchard plot) and the outcrossing constraint. Chi-square tests on various sample orchards confirmed these constraints were met by showing that (1) the clones were properly represented in the orchard, (2) the ramets of a clone were well-dispersed over the orchard, (3) no "trends" occurred in the placement of the ramets of the clones, and (4) each clone was adjacent to each of the other clones approximately the same number of times.

Extending the algorithm to handle more difficult orchard layouts is accomplished by t slight reconceptualization of the composition of the N and P matrices. The basic algorithm is independent of the constraints that are used. For an advanced generation orchard (where there are related clones), the clones should be segregated based on their parentage. The size of the P matrix remains k by t and the N matrix remains k by k, but now k equals the number of subpopulations. The construction of P and N follows accordingly and the creation of the templates becomes a two-stage process. Initially, the template is created by assigning a subpopulation to each of the positions in the template. After a feasible sequence has been found, a random clone from each of the assigned subpopulations is chosen and occupies its respective position. The N and P matrices are then updated based on the subpopulations occupying the orchard layout. Random clonal assignments to the template allows the subpopulations to remain balanced while the overall clonal population receives the proper weighted representation.

While altering the algorithm to allow the use of related ramets requires a reconceptualization of the meaning of N and P, special constraints such as a prevailing wind usually entail a reconceptualization of how the rows of these matrices are to be constructed. For example, if a prevailing wind is the primary pollen vector, the construction of P can be altered by using a varying minimum distance rule. An element in the matrix N would only be updated if the clones referred to lie in a certain direction to each other. Other contraints will affect the algorithm similarly. The construction of the templates will proceed normally.

Perhaps the most difficult alteration of the algorithm involves the creation of an orchard layout where the clones (or subpopulations) are not equally represented. A premultiplication of the cost matrix by a vector of weights will acheive the desired result, but the ascertation of the values to be used in the vector is obtuse. The weights should be such that, if there is only one ramet for a particular clone, then the expected placement of the ramet should be in the middle of the orchard. If a clone has two ramets, then one of the ramets will be expected to lie one-third into the layout, and the other clone should be expected to lie two-thirds into the layout. Other weighted representations follow accordingly.

The algorithm derived in this paper yields very promising results. Even large orchards can be designed in a reasonable time. User simplicity has been achieved since the number of clones, the identities of the clones, the planting dimensions and distances, the spacing between related ramets, and the degree of outcrossing control are the only inputs necessary for the algorithm to execute. Additional research is still necessary to provide the techniques needed for this algorithm to be applicable to all orchard layouts.

Sample Orchard Layout

The following sample orchard layout was designed using a population of 20 clones. The minimum separation of related ramets within the layout is three planting positions.

```
-------------------------------------------------------
|  1 |  7 | 13 | 14 |  2 |  9 | 15 |  3 | 12 |  4 | 16 |
-------------------------------------------------------
|  8 |  4 | 15 | 16 | 18 |  5 | 19 |  6 |  2 | 10 | 17 |
-------------------------------------------------------
|  6 |  9 | 11 | 19 |  7 | 17 | 13 | 16 |  5 | 14 | 15 |
-------------------------------------------------------
| 18 | 20 | 10 |  2 |  1 | 14 |  3 |  8 |  4 | 19 |  7 |
-------------------------------------------------------
| 12 |  8 |  5 | 13 | 20 |  4 |  9 |  7 | 10 |  3 | 16 |
-------------------------------------------------------
|  9 |  6 | 14 | 18 | 11 | 12 | 19 | 13 |  1 | 15 | 17 |
-------------------------------------------------------
|  4 |  7 |  3 |  9 |  6 | 10 |  2 | 20 | 18 |  5 | 11 |
-------------------------------------------------------
| 16 | 19 | 17 |  1 | 15 |  7 |  5 |  3 |  8 | 12 | 13 |
-------------------------------------------------------
|  8 | 14 | 20 | 11 | 16 | 17 |  9 | 15 | 19 |  6 |  1 |
-------------------------------------------------------
| 15 | 10 | 12 |  5 |  8 | 13 |  6 | 16 | 20 |  7 |  2 |
-------------------------------------------------------
|  5 | 18 |  9 |  3 | 10 | 20 |  4 |  1 | 14 | 13 | 11 |
-------------------------------------------------------
```

Bibliography

Balas, E.: An Additive Algorithm for Solving Linear Programming with Zero-One Variables. Op. Res. 13: 517—545 (1965). — Bell, G. D. and Fletcher, A. M.: Computer Organised Orchard Layouts (COOL) Based on the Permutated Neighborhood Design Concept. Silvae Genetica 27: 223—225 (1978). — Cooper, L. and Steinberg, D.: Methods and Applications of Linear Programming. W. B. Saunders and Company. Philadelphia, Pa. (1974). — Dyson, W. G. and Freeman, G. W.: Seed Orchard Designs for Sites with a Constant Prevailing Wind. Silvae Genetica 17: 12—15 (1968). — Freeman, G. H.: The Use of Cyclic Balanced Incomplete Block Designs for Non-directional Seed Orchards. Biometrics 25: 561—571 (1969). — Gansel, C. R.: Effects of Several Levels of Inbreeding on Growth and Oleoresin Yield in Slash Pine. Proceedings 11th Conference on Southern Forest Tree Improvement. Atlanta, Georgia: 173—177 (1971). — Giertych, M. M.: Systematic Lay-outs for Seed Orchards. Silvae Genetica 14: 91—94 (1965). — Golomb, S. W.: Tiling with Polyominoes. J. Combinatorial Theory 1: 280—296 (1966). — Golomb, S. W.: Tiling with Sets of Polyominoes. J. Combinatorial Theory 9: 60—71 (1970). — Hardgrave, W. W. and Nemhauser, G. L.: On the Relation Between the Traveling Salesman and the Longest Path Problem. Op. Res. 10: 647—657 (1962). — Hatcher, A. V. and Weir, R. J.: Design and Layout of Advanced Generation Seed Orchards, Presented at 21st Annual Southern Forest Tree Improvement Conference, Virginia Polytechnic Institute and State University, Blacksburg, Virginia: 205—212 (1981). — Jett, J. B. and Zobel, B. J.: Notes from Tree Improvement Cooperative Short Course. School of Forest Resources, North Carolina State University, Raleigh, North Carolina: 73—83 (1977). — La Bastide, J. G. A.: A Computer Program for the Lay-outs of Seed Orchards. Euphytica 16: 321—323 (1967). — Langner, W. and Stern, K.: Versuchstechnisches Problem bei der Anlage von Klon-Plantagen (An experimental problem concerning the arrangement of clone plantations). Zeitschr. für Forstgen. und Forstpflanz. 4: 81—88 (1955). — Malac, B. F.: Shifting-clone Design for a Superior Tree Seed Orchard. Woodland Research Notes 14, Union Camp. Corp., Savannah, Ga. (1962). — Wagner, H. M.: Principles of Operations Research. Prentice Hall, Inc. Englewood Cliffs, New Jersey (1969). — Zobel, B. J., Barber, J., Brown, C. L., and Perry, T. O.: Seed Orchards - Their Concepts and Management. Journal of Forestry 56: 815—825 (1958).

Genetic Variances and Covariances in Pinus contorta: estimates of genetic gains from index selection

By G. E. Rehfeldt[1])

(Received 14th June 1984)

Summary

Genetic variances and covariances were estimated for 6-year-old trees from 10 families within eight populations of *Pinus contorta* from the northern Rocky Mountains (USA). Analyses of 10 traits reflecting growth, quality, and adaptedness revealed substantial genetic variances for nearly all traits. As a consequence, univariate selection for height should provide genetic gains of about 5% per unit selection intensity. However, strong genetic correlations linked height with branch length (0.75), crown width (0.75), and the amount of shoot elongation that continues into mid-summer (0.70). Height was also related to the number of whorls on the current shoot and, consequently, to the number of branches. Thus, multi-trait selections are necessary to prevent genetic gains in productivity from being accompanied by inadvertent degeneration of traits related to adaptation and quality. Multi-trait selections, however, reduce anticipated genetic gains in productivity. Restricted selection indices are developed to exemplify procedures for holding constant the correlated responses while maximizing gains in height.

Key words: *Pinus contorta*, additive genetic variances, genetic correlations, genetic gains, restricted selection indices.

Zusammenfassung

Bei 10 Familien von 6 Jahre alten *Pinus contorta* innerhalb von 8 Populationen in den nördlichen Rocky Mountains (USA) wurden die genetischen Varianzen und Kovarianzen geschätzt. Die Analyse von 10 Merkmalen, die das Wachstum, die Qualität und das Anpassungsvermögen widerspiegeln, zeigte wesentliche genetische Varianzen für fast alle Merkmale. Als Konsequenz sollten allein für die Höhe genetische Gewinne von ungefähr 5% pro Selektionsintensitätseinheit möglich sein.

Enge genetische Korrelationen koppeln die Höhe mit der Astlänge (0,75), der Kronenbreite (0,75) und dem Ausmaß des Trieblängenwachstums, welches sich bis in die Sommermitte hinein fortsetzt (0,70). Die Höhe steht ebenfalls mit der Anzahl der Quirle des gegenwärtigen gleichlaufenden Triebes und folglich mit der Anzahl der Äste in Beziehung. Daher sind auf mehrere Merkmale gerichtete Selektionen nötig, um zu verhindern, daß mit genetischen Gewinnen bei der Produktivität gleichzeitig eine unbeabsichtigte Degeneration der Merkmale einhergeht, welche die Anpassungsfähigkeit und die Qualität bestimmen. Eine Selektion auf mehrere Merkmale hin reduziert beabsichtigte

[1]) Plant geneticist, Intermountain Forest and Range Experiment Station, USDA Forest Service, Ogden, Utah 84401, located at Intermountain Station's Forestry Sciences Laboratory, Moscow, Idaho 83843.