A technique for the composition of music in a computer

By S. Gill

This paper describes the principal features of a computer program that was written to generate music in the style of Schoenberg. The computer worked progressively through the composition, retaining at each stage several alternatives differing in the last few notes.

In response to an enquiry from the British Broadcasting Corporation, a program was written for the Pegasus computer to compose music of a particular limited type, and a short passage of the resulting music was broadcast during the programme "Machines Like Men" in the B.B.C. Television Service on Thursday 30th August, 1962. Part of the composition, translated into conventional musical notation and arranged for violin, viola, and bassoon is reproduced in Fig. 1.

Much work has already been done on the subject of musical composition by computer, notably by Hiller and Isaacson (1959). The problem is basically that of producing a detailed score which obeys certain rigid rules, and possesses a number of other desirable features, but which is otherwise arbitrary. The general procedure is to set up a routine that will generate random compositions obeying as many as possible of the rigid rules, and then by subsequent scanning to reject those which violate the remaining rules and to select the one having the most desirable features.

It is not practicable to generate entire pieces of music before operating the selection process, because that would require far too much material to be scanned, so the work has to be done in stages. Ideally, no doubt, the best procedure would be first to select the thematic material, then a skeleton plan of the work, and then to fill in more and more detail until ultimately the whole composition is expressed in terms of individual notes, rather in the way that a human composer might proceed. This, however, would call for a very sophisticated computer program that would need to deal with suitable representations of incompletely specified passages of

Fig. 1.—Part of the composition

music. So far as the author is aware no work on these lines has yet been published.

For the Pegasus program, the technique of serial composition was adopted, in which at any stage completely specified sequences exist from the beginning up to some point in the middle of the work, the remainder being so far uncomposed. Although many of the details of the program were somewhat arbitrary and not of lasting interest, the overall plan has novel features which it is the purpose of this paper to describe.

The "tree" process

The main difficulty with the process of alternate random generation and selection is that the computer may lead itself into a dead end. That is to say, it may find itself in a situation where part of the composition has been completely defined and cannot be altered, but which, if the rules are followed, could only lead to a very unsatisfactory state of affairs in the succeeding stages. It is, therefore, desirable to have some means of allowing the computer to "back-track", so that it can re-examine alternative choices at an earlier point in the composition.

However, to do so brings further complications into the program. Under what conditions should the computer go back, how far back should it go, how should it be prevented from making the same mistake twice, and so on? An alternative technique was, therefore, used in the Pegasus program, and appeared to have certain merits.

The technique adopted was to retain at any moment not one, but a small number (actually eight) of competitive versions of the partial composition, each completely specified up to a certain point, but not necessarily all the same length. The generation process took one of these partial compositions, or sequences, at random and extended it a short way according to the rules incorporated in the generating routine, making random choices where allowed. (In fact the extension was always by one quaver period or beat, although in principle the length of the extension could be varied from one step to another. Ideally perhaps one should try to keep constant the amount of *information* introduced by the random choices.) The result was then evaluated according to the remaining rules and criteria, and its value was compared with the values already found for the existing sequences. The sequence which had been chosen for extension was still retained in its unextended form as one of the candidates, so that at this point there was one extra sequence held in the machine. The weakest sequence was then rejected, and the whole process repeated.

The sequences were conveniently represented in the machine in the form of a tree, each sequence being linked backwards in time from the end to the beginning. Although eight alternatives were kept these did not all have to be stored independently, because in practice their earlier parts were common, so that they could all be linked back to the same initial passage. When a new sequence was generated it did not have to be copied out completely; it was merely necessary to record a new quaver beat to be added to its "parent" sequence, and to link the new beat back to this sequence. (Owing to the particular arrangement of block transfers in Pegasus the actual linking procedure was more complicated than this, but the differences are unimportant.) The result of this procedure was that the composition grew like a tree, continually throwing out new branches which grew to a greater or lesser extent depending on how successful they were in meeting the criteria laid down for evaluating sequences. From time to time branches died out, and finally a sufficient length of a single trunk was formed which was taken as the final composition.

Fig. 2 shows diagrammatically the first 100 steps in the development of a composition by this process. The order in which the various steps were taken is indicated by a serial number attached to each step. For example, the 20-th step extended by *one* quaver the sequence formed in the 8-th step, thus forming a new sequence with a length of three quavers. At any moment, only some of the sequences shown here were actually in the machine. Eight of the nodes in this diagram were current at any one time (except at the very beginning of the process), i.e. were available for extension. The only part of the tree existing at any moment consisted of these nodes and all the branches leading to them from the starting point. Thus, at one stage the eight current trial sequences were those formed in steps 23, 27, 35, 39, 44, 50 and 52; the storage of these entailed also storing steps 4, 15, 17, 26, 29, 37 and 43. The final composition was actually made up of steps 4, 26, 35, 52, 58, 74, 82, 87, 97 . . . (This example was derived from a diagnostic print-out obtained during an actual run.)

Arrangement of the process

The value of a sequence relates to the characteristics of the whole of that sequence from the beginning of the piece. Therefore, when evaluating a sequence obtained by attaching one further beat to its parent, the value of the parent itself must be incorporated. To this are added several terms expressing the extent to which the new beat satisfies the criteria for good sequences. Some of these terms may be negative so that the new sequence may have a lower value than its parent.

It was felt desirable to add a further feature to the process to discourage more strongly a sequence which, although itself valuable, continually failed to produce successful offspring. To do this a distinction was made between the "intrinsic" value (calculated as described above) of a sequence, and a "comparative" value, which was the one actually used for comparing sequences. The comparative value of a new sequence was initially set equal to its intrinsic value, but every time that the sequence was used as the parent of a new sequence, its comparative value was reduced by a fixed amount. Thus sequences which had already been extended in a number of ways were discouraged so as to give more
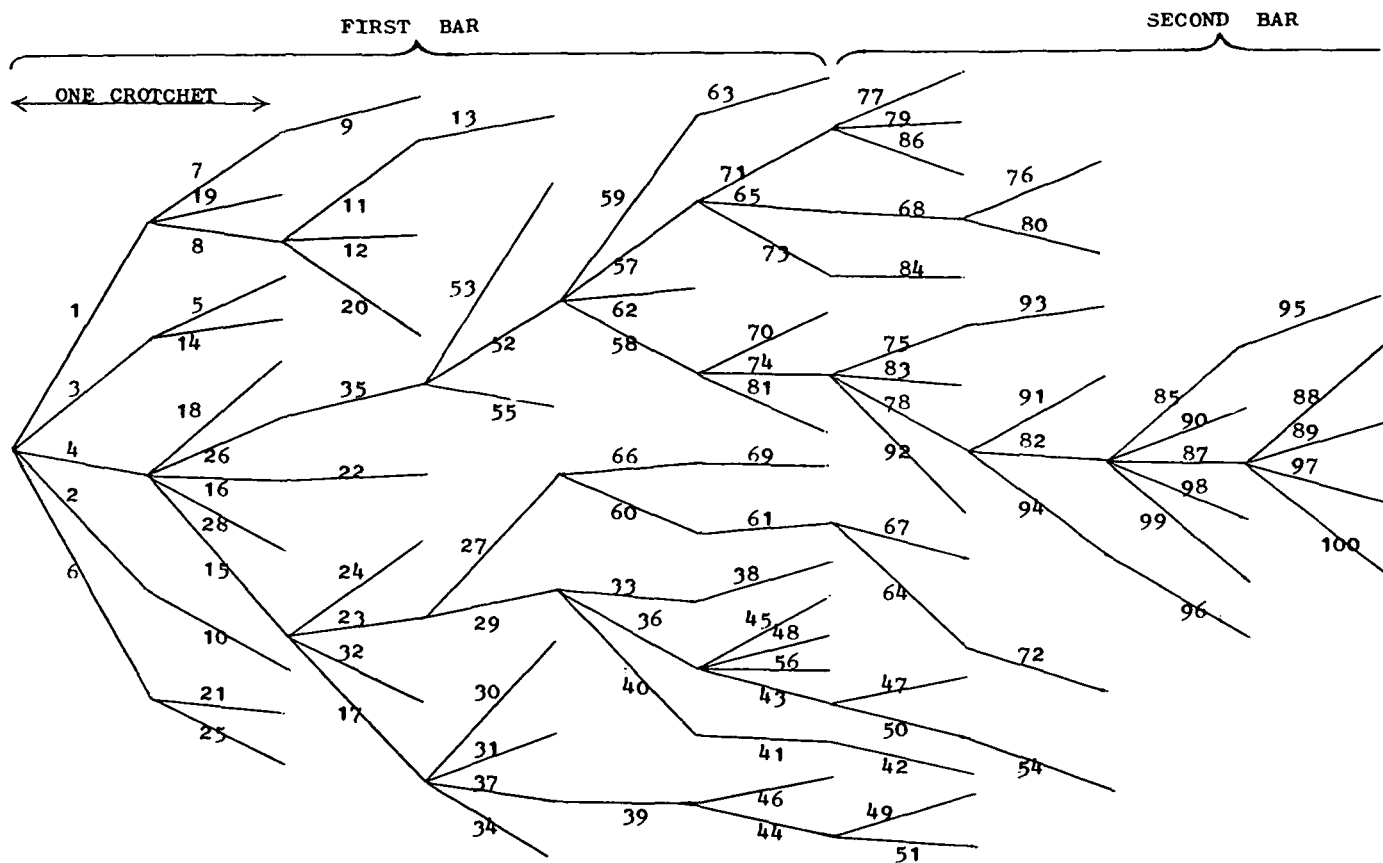
Fig. 2.—The first 100 steps

chance to the newcomers. When calculating the value of a new sequence it was the intrinsic value, not the comparative value, of the parent that was used. Unfortunately there was no time to make controlled experiments to determine the success of this device; it was put in purely as an article of faith.

To start the composition process, all eight trial sequences were initially set to the same state, representing a sequence of zero length. No other special arrangements had to be made before the generating and selecting cycle could be entered. However, an interesting problem arose in ensuring that the process would progress at a satisfactory rate. It is conceivable that if the criteria for evaluating sequences are too severe, the extended sequences or "offspring" will hardly ever succeed in ousting their parents, so that even after many hundreds of attempts the computer will not have got beyond the end of the first bar. Alternatively, if the criteria are too lax, almost any extension of a sequence will be accepted and the computer will very rapidly produce a long composition of poor quality.

The distinction between these two extremes lies in one single term contributing to the value of a sequence. This term is one which is contributed solely by the fact that the new sequence is one beat longer than its parent; in fact a single parameter is held in the machine to represent this term, and is merely added into the

value of every extended sequence. By decreasing this parameter the process is made more selective, and by increasing it the process is made to compose faster.

It was not easy to predict in advance the value of this parameter that would lead to a particular rate of composition, and, therefore, the parameter was adjusted by negative feedback so as to maintain a desirable rate. The observed rate was taken to be the mean rate of increase of the length of all eight trial sequences, smoothed over a period of the order of 50 program cycles. This was subtracted from a number set up on the hand keys, and the difference (suitably scaled) was used as the parameter controlling the composition rate. Thus the speed of composition could be controlled manually according to the amount of computer time available.

Composition rules

The rules adopted for the Pegasus program were aimed at producing music in the style of Schoenberg (the so-called 12-tone idiom). The music was (arbitrarily) for three voices in $\frac{3}{4}$ time, and no notes shorter than a quaver were allowed. Roughly speaking, each voice was constrained to follow the 12 degrees of the octave in a particular sequence ("tone row"), although the length of each note and the appearance of rests was open to choice. The same series of 12 degrees was

repeated throughout the work, although on each appearance it could be transposed up or down by any amount, reversed in sequence, and/or inverted (higher notes being replaced by lower and vice versa); also individual notes could be transposed up or down by complete octaves. Occasional deviations from this were allowed, by which a particular "tone row" could jump from one voice to another, but by and large the main element of choice lay in the disposition of notes and rests in time.

Some rather complicated evaluating criteria were supplied to endeavour to achieve a pleasing pattern of activity in the three voices. It was desired that each voice should rest for about a bar roughly once in every four or five bars, but preferably no two voices should be resting at the same time. It was also desired that, of the active voices, one should be moving fairly rapidly and another more slowly. These requirements were translated into suitable rules for calculating the value of a sequence, along with a few other terms designed to exercise some control over the length of skips (large changes of pitch) in each voice, avoidance of parallel octaves or near octaves, etc.

No serious attempt was made to produce an overall structure to the composition in this instance, since it was intended merely to be used as incidental music in a television programme. In particular there was no legislation for producing a satisfactory ending, although there would seem to · be no insuperable difficulty in introducing this by making the rules dependent on the point reached in the composition. In this instance Pegasus merely generated a bar or two beyond the required length, and the composition was trimmed to length later.

Storage within the computer

The storage of the music itself required one computer word per quaver. Thus a complete bar took six Pegasus words, which could be accommodated comfortably within the standard Pegasus block of eight words, leaving two words for linking-information etc. Owing to the characteristics of the Pegasus store the tree was constructed of complete blocks, and no provision was made for linking new sequences to a point in the middle of a bar. This meant that, if a parent sequence did not happen to end at the end of a bar, its last (incomplete) bar had to be copied in order to record the offspring.

The usual device of a "free list" (a simple linked chain of unused blocks) was used to keep track of available storage space. Each block in use contained a counter showing how many later blocks were linked to it, and when this count fell to zero and the block was no longer itself the last block of a trial sequence, the block was abandoned. This meant that it was returned to the free list, and the link-count in the block to which it had been linked was reduced by one (as a result of which this block might also be abandoned).

In addition to the tree comprising these blocks, a special "key block" was maintained for each of the

```
B.GCFC..O.E.  D.HCJ.K.....  L.N.I.O.MCO.  O.O.H.O.JEO.
FCACCC..F.A.  O.J.KCN...I.  D.E.MCNCHC..  JC..M...H.C.
O.O.O.O.G.L.  J...O.O.O.O.  O.O.O.KCGC..  M.KCMCHCK.F.


O.O.O.O.F...  GCA.FCK.AC..  ..GCL.J.O.O.  A.G.F...KC..
DCG...I.D...  E.FCGCO.GCFC  L.O.O.O.K...  I...N.KCJ.H.
GCJ.L.N.A.B.  CCB.A......O.  H.FCO...O.O.  O.O.O.O.O.O.


H.C...L.GCI.  D.FCACCCKCH.  N.MCGCHCO.J.  DCN...O.O.O.
M...JC....O.  O.O.O.O.O.O.  O.O.O.O.O.O.  HCHCC.D.E.O.
O.O.HCN.L.O.  J.GCF.G.F.DC  CC......ACFC  K.....I.O.O.


O.O.O.O.O.O.  O.I.O.F.O.A.  CC..E.F.N.H.  I.D.FCHC..O.
G...I.KCMCH.  JCM.NC..MC..  E...O.K...O.  O.O.J.A.....
O.O.NC..L.O.  J.H.G.AC..D.  O.FCO.H.C...  L...NCI.O.O.
```

Fig. 3.—Output notation

eight trial sequences. Besides a link to the end of the branch representing the sequence, this block contained a fair amount of detailed information concerning the sequence. This included its intrinsic value, the form of the tone row being used by each voice, its current position in that row, and all information about the voice's recent activity required to enable a suitable evaluation of its future activity to be made.

Output

At the end of the composition process, the sequence with highest value was chosen, and the tree was scanned backwards starting from this branch; during this scan all the links were reversed so that the composition was now linked forwards starting from the beginning. It was then possible for an output routine to work through the composition in the forward direction converting it to a suitable printed notation.

The output notation was constrained by the teleprinters available, and is illustrated in Fig. 3. Each voice is represented by one of three lines of print. The program was such that voices were only allowed a range of two octaves, and pitches were indicated by extending the usual nomenclature (A to G) to cover a second octave (i.e. up to N). Accidentals were indicated by a £ sign meaning a sharp (flats were unnecessary since in this style of music no distinction is made between A sharp and B flat). Inside the computer, pitches were represented as the number of semitones above a base pitch. For output, this number was multiplied by 7/12 and a suitable constant added. The integral part of the result gave the letter to be printed, and the size of the fractional part showed whether a sharp should be indicated. Two printed characters were allowed per quaver; the holding of a note for more than one quaver was indicated by a line of dots. Rests were indicated by the letter O.

Conclusion

Although the author was relatively unmoved by it, the resulting music appeared to have some small positive

132

merit to a professional musician. However, this was not achieved without an appreciable amount of experimentation. The miscellaneous parameters appearing in the rules of composition seemed to affect the results in a rather erratic way, and it was not easy to adjust them all by trial and error in order to produce acceptable results. Perhaps, in any future work of this kind, the idea of negative feedback should be used to control some of these parameters as well as the rate of composition. However, there would obviously be a risk of instability if this were carried too far.

The composition of music by means of a computer introduces yet another new complication in the already turbulent world of modern music. The author's experience has convinced him that the computer program is hardly a substitute for the human composer but is rather a new (and somewhat devious) medium of expression. Composers have already faced the challenge of expressing their work in the form of, e.g., schedules for the copying of sections of tape recordings through filters (in composing musique concrète), or punched tape to control electronic tone generators. Now they may also express themselves in the form of computer programs (or in the form of controlling parameters to be supplied to music-generating programs).

There is, however, one significant difference between composition by computing and by orthodox methods, namely that when using a computer the composer is much further removed from the final result (i.e. the music) than he is when writing an ordinary score. Experience will no doubt help a great deal, but it is doubtful whether composers will ever be able to foresee very clearly the full result of every choice that they make when feeding the computer.

Perhaps in the end we shall see musical composition taking the form of a co-operative venture between the human composer and the computer, with the computer supplying a number of plausible passages along lines suggested by the composer, who in turn selects the ones he wants and calls for further variations and ornamentations as required. This will, of course, call for suitably designed input and output devices. It would probably be done most effectively by means of a time-sharing program in a very powerful computer, since it would require spasmodic bursts of rapid computing.

Acknowledgements

Reference

HILLER, L. A., and ISAACSON, L. M. (1959). *Experimental Music*, New York: McGraw-Hill.

Correspondence

To the Editor,
The Computer Journal.
Sir,

"The LL^T and QR methods for symmetric tridiagonal matrices"

In their paper, "The LL^T and QR methods for symmetric tridiagonal matrices" (this *Journal*, Vol. 6, p. 99), James M. Ortega and H. F. Kaiser have successfully eliminated all square roots from the formulae for the LL^T-transformation (symmetric or "Choleski" modification of the LR-transformation) if applied to a symmetric tridiagonal matrix, and derive the formulae

$$
\left.
\begin{aligned}
q_1 &= a_1 \\
e_i &= b_i^2/q_i \\
\bar{a}_i &= q_i + e_i \\
q_{i+1} &= a_{i+1} - e_i \\
\bar{b}_1^2 &= q_{i+1}e_i \\
\bar{a}_n &= q_n
\end{aligned}
\right\} \text{ for } i = 1, 2, \ldots, n-1
$$

(q and e stand for the authors' d^2 and s^2 respectively).

This would seem to be a great achievement, but by comparison with the basic formulae of the quotient-difference algorithm (Rutishauser, 1957; Henrici 1958),

$$
\left.
\begin{aligned}
\bar{q}_1 &= q_1 + e_1 \\
\bar{e}_i &= q_{i+1}e_i/\bar{q}_i \\
\bar{q}_{i+1} &= q_{i+1} + e_{i+1} - \bar{e}_i
\end{aligned}
\right\} \begin{aligned} &\text{for } i = 1, 2, \ldots, n-1, \text{ with} \\ &e_n \text{ taken to be zero,} \end{aligned}
$$

it becomes obvious that the authors have nothing but re-established the quotient difference algorithm which since long has been used to compute eigenvalues of symmetric tridiagonal matrices, and which, incidentally, was the starting point from which the LR-transformation was derived by generalization.

The abbreviated QR-transformation on the other hand seems to be new and useful. However, in order to improve its numerical stability, it would be advantageous to carry not only the s_i^2, but also the corresponding $c_i^2 = 1 - s_i^2$ in the calculation.

Yours faithfully,
H. RUTISHAUSER.

Swiss Federal Institute of Technology,
Zurich, Switzerland.
13 May, 1963

RUTISHAUSER, H. (1957). *Der Quotienten-Differenzen Algorithmus*, Mitt. Inst. angew. Math. der ETH, Nr. 7, Birkhäuser Verlag, Basel.
HENRICI, P. (1958). *The Quotient-Difference Algorithm*, Nat. Bur. Standards, Applied Math. Series Vol. 49, p. 23–46.

133