Letters to the Editor

On "Systematic Generation of Hamiltonian Circuits"

EDITOR:

The paper by S. M. Roberts and Benito Flores entitled "Systematic Generation of Hamiltonian Circuits" [*Comm. ACM 9* (Sept. 1966), 690] contained several serious errors. After a description of a straightforward algorithm for the systematic generation of all Hamiltonian circuits of an undirected graph, it presented a flowchart for an inefficient program embodying the algorithm, erroneously claimed novel discovery of a standard method which removes some of its inefficiency, and failed entirely to remove its remaining inefficiency. It also rejected as "difficult to implement on computing machines" an algorithm in a book by Berge [1], while Berge had selected it in particular because it was "suitable for machine use."

One idea of possible value was the technique of reducing the extent of the search when it is possible to specify chains of nodes that are always to be linked together. However, since the user in such a case would treat the entire chain as a single arc, the inclusion of interior points of chains in this feature of the program only slows it down. The 57-node example in the article was quite deceptive, because it included four chains involving 42 of the nodes; i.e., the problem was really a 23-node problem as far as the computation was concerned.

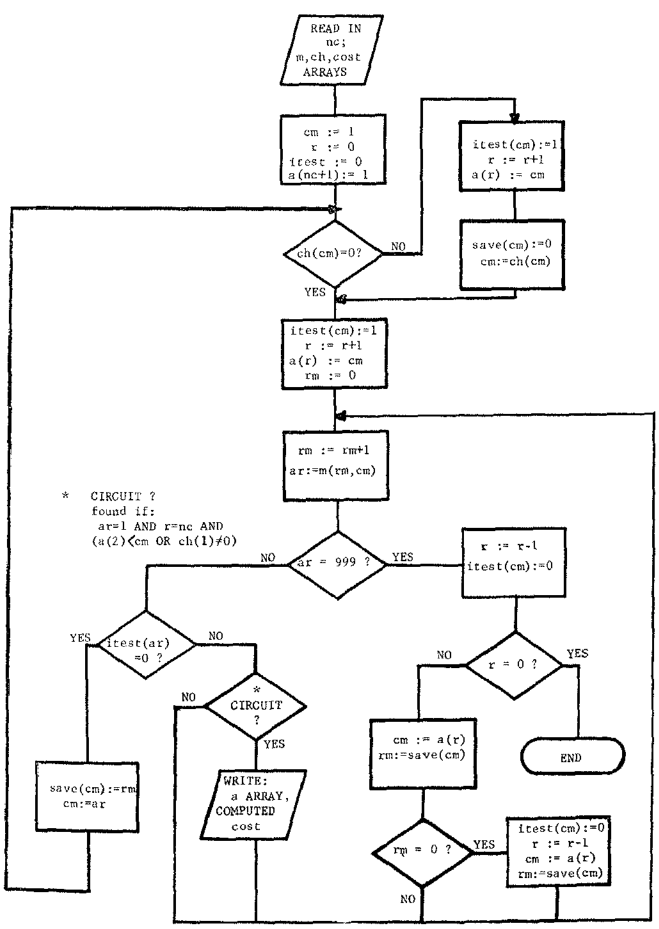

FIG. 1

The authors stated that their program was "fairly easy to describe and fairly difficult to implement." As the accompanying flowchart demonstrates, the program design could have been much simpler. For example, if they had known a recursive language like ALGOL 60 instead of FORTRAN IV, instead of searching for where they left off every time backing up was necessary, a recursive procedure would have preserved that information in a local variable. This concept could then have been translated to FORTRAN by one of the methods of recursion simulation (cf. [2]). This would amount to using a single-dimensional array to remember the row number being examined in each column.

The accompanying flowchart (Figure 1) embodies the improvements to the authors' flowchart so far discussed. Notice: (1) Chains are represented only by their endpoints, which point to each other in the array CH; (2) No restart feature is included; this is often a function of the operating system, and does not belong in a published flowchart; (3) The DO 100 loop at statement 70 in the paper was awkward and is eliminated.

The algorithm published by Berge, to which the authors alluded, involved reduction of the problem by a simple systematic method, and solution of the reduced problem by substitutions for the remaining unknowns. The authors were apparently confused by the Boolean equations; but these were a notational convenience which disappear when the method is programmed. In any case, Berge's method works as stated only for loosely connected graphs, and is not ideal for the traveling salesman problem.

In one step of the authors' algorithm, they checked interior points of chains unnecessarily for previous inclusion in the circuit; it is impossible that any of these points could have appeared unless both endpoints also appeared.

Some minor errors I noticed were:

(1) In Table IIIA, "1" was omitted from column 11.

(2) Although the check for prior inclusion of interior points of chains is superfluous, as noted above, it wouldn't have worked anyway, because the "Remove Chain" box in the flowchart fails to unmark these points in ITEST.

(3) Omitted from the initialization are $A(1) := 1$ when ISTART $= 0$ and recalculation of ITEST when ISTART $= 1$.

(4) Every circuit is printed twice (forward and backward); this extra printing could have been avoided easily, by requiring a check that $A(2) < A(\mathrm{NC})$ in the output routine.

Finally, the use of the ITEST array was claimed by the authors to be a novel contribution. Apparently they were not aware of the "Boolean array" in ALGOL 60, which was included in the language for just such purposes. The technique is commonly used in programming, and thus is not novel.

REFERENCES:

1. BERGE, C. *Theory of Graphs and its Applications*, transl. by A. Doig, John Wiley, New York, 1962, Ch. 11.

2. McKeeman, W. and Tesler, Larry. Algorithm 182: Nonrecursive adaptive integration. *Comm. ACM 6* (June 1963), 315.

Lawrence G. Tesler
Information Processing Corporation
Palo Alto, California

Mr. Roberts and Mr. Flores Reply

Editor:

Mr. Tesler makes violent charges without pin-pointing the specific object he is attacking. For example, in his first paragraph he claims the paper contains "several serious errors." He never states anywhere in his letter what these serious errors are. What does Tesler mean? Does he mean the program philosophy is incorrect? Does he mean the program will not work? Does he mean the program can not handle the problem it purports to deal with? In the first paragraph he objects to our flowchart and program because it "erroneously claimed novel discovery of a standard method which removes some of its *inefficiency*, and failed entirely to remove its remaining *inefficiency*." Tesler never bothers to support his contention that the method is "standard" nor does he describe specifically these two types of inefficiencies.

Tesler's letter is inconsistent. For example in his paragraph 1 he implies that we erred in rejecting Berge's algorithm, while in his paragraph 5, he agrees that the Berge algorithm is not really useful for these problems.

Mr. Tesler's main claim appears to be that he can draw a "simpler" flowchart than ours. He fails however to *emphasize* the fact that he has borrowed heavily our logic and our ideas. For example, he employs the combinatorial matrix, the chain idea, and the ITEST array idea. See also our DO 100 loop. Prodded by our previous correspondence he modestly allows in his Paragraph 4 that his flowchart is based on ours. Admittedly our code, as any code, is susceptible to improvement. Mr. Tesler and also one of our colleagues have suggested testing each node for duplication in the ITEST array before testing whether the node is an initial point of a chain. This is a good idea which will simplify the handling of chains.

In our earlier correspondence and even now, Tesler fails to distinguish between the conception of an idea and the implementation of the idea by a programming language. For example, he rejects our claim of novelty for the ITEST array idea because Algol language has Boolean capability. The fact remains nevertheless that the concept of the ITEST array to reduce $(N - 1)$ comparisons to one comparison is novel. That Tesler can find an Algol 60 Boolean array to implement the ITEST idea (*now that we have stated it*) does not tarnish the novelty of the idea. After all, the idea came first, the implementation second. By Mr. Tesler's logic we should all be Shakespeare's since we have at our command all the words available to the Bard.

Mr. Tesler has noted correctly some minor errors in the flowchart. The corrections are initialization of $A(1) = 1$, when ISTART = 0 and recalculation of ITEST array when ISTART = 1. Column 11, of Table IIIA, should contain a 1 not a 53.

The intent of the "REMOVE CHAIN" box is to remove the chain and, of course, to remove the corresponding entires from ITEST, since this is in fact what the program carries out. Perhaps indexing in this block would have clarified this.

To conclude QGRAPH does in fact generate, as claimed, all the Hamiltonian circuits and their costs, consistent with the M-matrix and chains.

S. M. Roberts
International Business Machines Corp.
Houston, Texas
and
Benito Flores
University of Houston
Houston, Texas

ESC Facility in USASCII

Editor:

The Morenoff and McLean paper (Jan. 67) is interesting as it demonstrates the Escape (ESC) Facility in the USASCII, which facility has not been exercised previously as it perhaps should. Certainly some code for non-numeric information processing could be permanently associated with one of the 255 indicators following the Escape character. It was proposed in 1962 in Stockholm that a block of these indicator codes should be reserved for association with alternate codes for programming languages. Subunits of such alternate codes might be identical to USASCII, but this is not vital.

In other words, the standardization process might, in the future, yield some alternate standard codes for special purposes which are in a sense subordinate to and linked through USASCII. Thus Mr. Morenoff's proposal is in no way in conflict with USASCII, and is indeed consonant with it.

However, the construction principles of his code are subject to scrutiny. One major premise is that functionally like sets of graphics (i.e., the alphabet) can be identified by bracketing between two binary numbers. For major sets this is also true in USASCII. A second premise is that within each such set "the ordering and sequencing of characters and words can be accomplished by simple binary comparisons of codes."

Here the author has fallen into a few traps, which might have been avoided by studying the bibliography of X3.2. The first is familiar to me because I did the same in the IBM 7030—that is, interspersing the upper and lower case representations of letters and thus giving each full graphic significance. Tain't so, and so the phone books show. With Mr. Morenoff's code, we would have:

De Carlo
De La Rue
De Long
DeLair
DeLancey
DeLaRue
Delancey
de Carlo
de la Rue
deLancey, and anguished subscribers.

The only proper method is to strip the case bit (b_6 in USASCII, b_1 here) and make a minor comparison upon equality. This may get a little more complicated in the case of italics, which obviously cannot be interspersed with the other graphemes in the proposed code.

Since the blank is high to digits and low to letters, we get:

A266		08
A2B	and	b̸8
A 66		
AB66		

The proposed code is perhaps more awkward than USASCII in comparing two numerals, since it requires radix point alignment and zerofill. Negative signs must be handled separately in either code.

There could be difficulties here with editing instructions.

Workers in the code microcosm will be sure to welcome Mr. Morenoff's interest.

R. W. Bemer
General Electric Co.
13430 N. Black Canyon
Phoenix, Arizona 85029

EDITORIAL

The European Computer Gap

by G. Salton

Among people concerned with the politics of science and technology, a good deal of discussion has centered recently on the relationship between the United States and a number of foreign countries, and on the effect abroad of American conditions of scientific research and development. Slogans such as the "technology gap" and the "brain drain" are being coined, reflecting the areas of principal concern, and the scientific achievements in Europe and elsewhere are often compared unfavorably with American performance. Computers and computer science play an important role in this situation, and it appears worthwhile, therefore, to examine the problem in more detail.

If the truth must be told once again, it is clear that in computer science at least, conditions in Europe appear in a less favorable light than those in the United States. In some countries—for example, England and Germany—a few independent computer manufacturers are on the scene, but no European country has an industry which is really strong and viable. In the universities, computer science is either ignored, or treated haphazardly without breadth and without consideration of its potential importance, and in most application areas computing theory and practice are lagging. Are the Europeans aware of the situation?—Indeed they are. Even in the popular press, more and more articles appear, drawing attention to the unhealthy conditions in the computer area, and demanding action of one sort or another. Committees are created to survey the situation, and in some countries emergency action programs are initiated, exemplified by the French "Plan Calcul," which is expected to revive an independent French computer industry. Many universities are also reexamining their position, and changes in the engineering or mathematics curricula designed to include computer-oriented courses are contemplated for the foreseeable future.

Without intending to be smug about our own situation, by disregarding, for example, our own shortcomings and failures it may still be appropriate to ask what exactly the troubles are with the computer field in Europe, and what can be done to lessen the contrast with American practice. This turns out to be a problem not at all easy to handle. Some people of course have ready answers, and take the simple view that Europeans can make easy headway by merely copying all things American. Typical instances of European decadence are given, for example, in a recent editorial in *Datamation* (September 1966), where a transplanted American is said to have explained the computer gap by pointing to the absence of the wall-mounted pencil sharpener, and to the fact that $1,000 and one week are needed to have a telephone installed in downtown Paris. Without discounting the accuracy of such stories, one is forced to say that they hardly touch the core of the problem.

Other observers acknowledge the importance of some of the well-known European contributions to the computer arts, for example, the page-turning procedures proposed with the original design of the Atlas, or the multi-processing algorithms included in the Gamma-60, but then add that in view of the obvious lack of know-how—demonstrated by the fact that these paper designs could never be converted into operational systems—it might have been more efficient to work on more down-to-earth proposals. Unhappily, equally notorious paper design failures can easily be quoted for the American scene, demonstrating presumably an equal lack of know-how on our side. Neither would it appear that the emphasis of European universities on "excellence-at-the-expense-of-reality"—to quote again from *Datamation*—could be responsible for the lag in computer science. After all, in our field, no one can ever be too excellent.

A more reasonable explanation, in my opinion, is given in an editorial in *Science* (17 February 1967) where the writer points to the big differences in social attitudes between Europeans and Americans. These attitudes, which of course affect business life as well as university life, are the product of long traditions in Europe, and are therefore particularly difficult to change. One of the more striking features of European life is the rigidity with which many problems are treated, compared with the flexibility and open-mindedness here. Many organizations operate in a strictly circumscribed environment with a single prespecified slot provided for each participant, and at most one well-established route to get from where one is to where one wants to be. Universities operate in many instances under semi-militaristic conditions, where the areas of competence are clearly defined and maintained, and cooperation across the invisable borders between the various groups and institutes can hardly be said to exist. There is also a chronic fear of competition, so much so that the best individuals or organizations are often not permitted to work in an area of interest if a previously established group has already laid a claim to it. All these rules and conventions make it difficult to approach new areas with the necessary speed and the appropriate resources, and even relatively minor alterations in established procedures—for example, the addition of a new course in a university—become topics for lengthy discussion and careful consideration.

To add to the effects of the previously cited inflexibility in the professional life, there exists also in Europe a well-established tradition of centralization, fortunately lacking in the United States. In education the central directives, formerly often provided by various religious organizations, are now mostly furnished by governmental bodies; the previously established centralized outlook is, however, generally maintained. Furthermore, in many cases much greater control is exercised by the responsible agencies over research and development work than is the case in the United States, and under normal circumstances only one such controlling body exists in any area without whose cooperation and approval little progress can be made.

Let us then be grateful for the multiplicity of avenues provided in American professional life, for the many competing outlooks, for the numerous support agencies, and for our lack of confining traditions. And let us hope that after water coolers and wall-mounted pencil sharpeners will have been installed, and the computer gap will have been narrowed, life in Europe will not have become Americanized to such an extent that many of us would no longer enjoy going there to work and interact with our European colleagues.