Programming Techniques

R. Morris Editor

Optimizing the Polyphase Sort

Donald L. Shell
General Electric Company, Schenectady
New York

Various dispersion algorithms for the polyphase sorting procedure are examined. The optimum algorithm based on minimizing the total number of unit strings read is displayed. The logic of this algorithm is rather complicated; hence, several other new dispersion algorithms with more straightforward logic are presented. Of the simple dispersion algorithms discussed, the Horizontal is best. It does approximately one-fourth to one and one-half percent less reading and writing than most algorithms in use today. An additional two and one-fourth to three percent improvement can be achieved by utilizing the Modified Optimum Algorithm. This algorithm is relatively straightforward, but it requires a fairly close estimate of the total number of unit strings before the dispersion begins.

Key Words and Phrases: sorting, polyphase sorting, dispersion algorithms, optimum dispersion algorithm, repetition operator

CR Categories: 5.31

1. Introduction

This paper presents an analysis of the well-known polyphase sort procedure. The polyphase sorting technique is discussed at some length in Flores' book [2] and elsewhere [1,3] in the literature. It is widely used in generalized sort programs provided to users by various computer manufacturers.

These various implementations utilize a variety of dispersion schemes; that is, methods of distributing unit strings onto various tapes prior to the actual polyphase merge procedure. Although these techniques are efficient, thus far there has been no known optimum procedure. In this paper we present the optimum scheme resulting from the minimization of the number of unit strings read and written during the polyphase procedure. The polyphase procedure is assumed to use a set of dummy strings and real strings in a manner analogous to that described by Malcolm [5]. This is consistent with the above minimization criterion.

This data is presented for orders R from 2–7 inclusive, where the polyphase merge of order R requires $R + 1$ tape drives.

The preparation of unit strings for the polyphase merge procedure is not discussed in this presentation. Instead, it is assumed that the unit strings are of equal length for purposes of the analysis. Gilstad [4] has discussed a procedure for polyphase merging where the tapes may be read in both the forward and backward direction. In the following analysis the assumption is made that all tapes are read in the forward direction only.

As is well known, there are certain numbers of unit strings which should be placed on various tape drives initially in order to achieve a precise polyphase merge with i levels or passes. These polyphase numbers $N_{i,k}(R)$ for order R and i levels are related to the Fibonacci numbers of order R in the manner described

Table I. Polyphase Numbers $N_{i,k}(R)$

$i\downarrow$ \ $k\rightarrow$	R = 2: 0	1	2	R = 7: 0	1	2	3	4	5	6	7
1	2	1	1	7	1	1	1	1	1	1	1
2	3	2	1	13	2	2	2	2	2	2	1
3	5	3	2	25	4	4	4	4	4	3	2
4	8	5	3	49	8	8	8	8	7	6	4
5	13	8	5	97	16	16	16	15	14	12	8
6	21	13	8	193	32	32	31	30	28	24	16
7	34	21	13	385	64	63	62	60	56	48	32
8	55	34	21	769	127	126	124	120	112	96	64
9	89	55	34	1531	253	251	247	239	223	191	127
10	144	89	55	3049	504	500	492	476	444	380	253
11	233	144	89	6073	1004	996	980	948	884	757	504
12	377	233	144	12097	2000	1984	1952	1888	1761	1508	1004
13	610	377	233	24097	3984	3952	3888	3761	3508	3004	2000
14	987	610	377	48001	7936	7872	7745	7492	6988	5984	3984
15	1597	987	610	95617	15808	15681	15428	14924	13920	11920	7936

i	R = 3: 0	1	2	3	R = 6: 0	1	2	3	4	5	6
1	3	1	1	1	6	1	1	1	1	1	1
2	5	2	2	1	11	2	2	2	2	2	1
3	9	4	3	2	21	4	4	4	4	3	2
4	17	7	6	4	41	8	8	8	7	6	4
5	31	13	11	7	81	16	16	15	14	12	8
6	57	24	20	13	161	32	31	30	28	24	16
7	105	44	37	24	321	63	62	60	56	48	32
8	193	81	68	44	636	125	123	119	111	95	63
9	355	149	125	81	1261	248	244	236	220	188	125
10	653	274	230	149	2501	492	484	468	436	373	248
11	1201	504	423	274	4961	976	960	928	865	740	492
12	2209	927	778	504	9841	1936	1904	1841	1716	1468	976
13	4063	1705	1431	927	19521	3840	3777	3652	3404	2912	1936
14	7473	3136	2632	1705	38721	7617	7492	7244	6752	5776	3840
15	13745	5768	4841	3136	76806	15109	14861	14369	13393	11457	7617

i	R = 4: 0	1	2	3	4	R = 5: 0	1	2	3	4	5
1	4	1	1	1	1	5	1	1	1	1	1
2	7	2	2	2	1	9	2	2	2	2	1
3	13	4	4	3	2	17	4	4	4	3	2
4	25	8	7	6	4	33	8	8	7	6	4
5	49	15	14	12	8	65	16	15	14	12	8
6	94	29	27	23	15	129	31	30	28	24	16
7	181	56	52	44	29	253	61	59	55	47	31
8	349	108	100	85	56	497	120	116	108	92	61
9	673	208	193	164	108	977	236	228	212	181	120
10	1297	401	372	316	208	1921	464	448	417	356	236
11	2500	773	717	609	401	3777	912	881	820	700	464
12	4819	1490	1382	1174	773	7425	1793	1732	1612	1376	912
13	9289	2872	2664	2263	1490	14597	3525	3405	3169	2705	1793
14	17905	5536	5135	4362	2872	28697	6930	6694	6230	5318	3525
15	34513	10671	9898	8408	5536	56417	13624	13160	12248	10455	6930

in Flores' book. For the purpose of this analysis $N_{i,0}$ is the total number of unit strings which can be handled by i levels, and $N_{i,k}$ is the number of strings which must initially be on tape unit k. The equation for computing N is presented in Section 4 below. The polyphase numbers for orders 2–7 and levels 1–15 are presented in Table 1.

2. Dispersion Procedures

Figure 1 illustrates the dispersion procedures which are discussed in the literature. This illustration is for order $R = 3$. In Figure 1, one should imagine that the vertical strips represent the various tapes which initially contain unit strings. On these tapes are string positions beginning with number 1 at the top down to however many positions are used on the tape. String position 1 is always assumed to be at the beginning of the tape when one is ready to start the merge procedure. Of course, as unit strings are added to a tape in the initial dispersion, they are added to the end of the tape. For purposes of analysis, however, one can think of these logically as being placed in string positions at the beginning of the tape. That is the way the illustrations are presented.

Figure 1 represents a level 5 situation. The $\times$-marks in the various unit string positions show the locations of strings which are on the tapes as a result of completing dispersion for level 4. Then the numbers

Fig. 1. Dispersion Sequence for Unit Strings.

String Position	Vertical			Diagonal			Horizontal			Rectangular		
1	23	28	31	31	30	26	29	30	31	30	31	×
2	22	27	30	29	28	23	26	27	28	28	29	×
3	21	26	29	27	25	20	23	24	25	26	27	×
4	20	25	×	24	22	×	21	22	×	24	25	×
5	19	24	×	21	19	×	19	20	×	22	23	×
6	18	×	×	18	×	×	18	×	×	20	21	×
7	×	×	×	×	×	×	×	×	×	18	19	×
8	×	×		×	×		×	×		×	×	
9	×	×		×	×		×	×		×	×	
10	×	×		×	×		×	×		×	×	
11	×	×		×	×		×	×		×	×	
12	×			×			×			×		
13	×			×			×			×		

Order 3, Level 5. Strings 18 through 31.

indicate, in order of occupation, the various string positions to disperse to a total of 31 strings, the maximum that can be handled by level 5.

The vertical dispersion procedure fills the longest tape first, then the next longest, etc., until all tapes have been filled for the level under consideration. This is the method presented by Reynolds [8]. The diagonal procedure places the strings on each tape in succession until the shortest one is filled, then on the remaining tapes in succession, etc., until dispersion is complete. This is the scheme suggested by Flores [2, p. 152]. The horizontal procedure fills in the various string positions on the longest tape first till the same number remain on the second longest, then on those two tapes in succession, etc. In this case, one can think of the strings flowing into the picture and filling it like water in a jar until string position 1 on each tape is filled. So far as the writer can determine, this procedure is not mentioned in the literature. The rectangular procedure assumes that the longest tape from the previous level is the shortest tape on the current level. This means that the same number of strings must be added to each of the other tapes. They are added one at a time to the various tapes in succession. This is the method discussed by Mendoza [7].

A "special" vertical procedure is discussed by Malcolm [5]. He adds strings vertically in an order dictated by the requirement that a specific tape must be the final collection tape. This special factor is ignored in all the other discussions. A "multiphase procedure" is discussed by Manker [6] for order 3 only. This assumes that the unit strings are dispersed evenly on two tapes. A special set of multiphase passes including some preliminary merging is then used to get an initial polyphase distribution on three tapes.

Heretofore, in the literature, it has been tacitly assumed that the Optimum Dispersion Procedure should minimize the number of levels. This means that one completes a picture similar to that in Figure 1 before considering the dispersion of strings in such a way as to require an additional level. Surprisingly it turns out that to achieve the absolute minimum amount of reading and writing of unit strings this assumption is erroneous. It is sometimes advantageous to utilize additional levels of the polyphase procedure in order to reduce the reading and writing volume. This result will be demonstrated below.

3. Polyphase Times

The time required to perform the polyphase merge is obviously a function of the number of levels *and* the number of unit strings read. This analysis concentrates on the number of unit strings read. The assumption is made that the time required is proportional to the unit strings which are read and written. In those situations where minimizing this time requires more than the minimum number of levels, a tradeoff is required. This tradeoff is left to the user to work out for his own specific environment.

In order to arrive at an understanding of the time calculation, one should refer to Figure 2. This illustration represents the string positions occupied by a sort of order 3 at each level. As one can see a sort requiring one pass has one string on each of three tape units, and so on through levels 2, 3, etc. The illustration also indicates how many times the string located in a given position on the tape unit must still be read in order to complete the sort. For example, with one pass to go, each string must be read once. If, however, two passes are required one can see that the first string on each tape must be read twice and the second string on each of two tapes must be read once. In general, the number of reads and writes required are determined iteratively by working backward from level 0 to 1, 2, 3, etc.

It may be noted that starting with any level the string in the nth-position from the beginning of the tape on all tapes will be read the same number of times. It is also apparent that in order to minimize time one should disperse the strings in such a way as to leave vacant, where possible, those positions which must be read the largest number of times. This means, for example, that with 6 levels and with 12 positions to be left unoccupied one should leave positions 1, 2, 3, and 5 vacant, but fill position 4 on each tape. Thus one can see that the optimum dispersion in general will be quite complicated.

4. Time Calculation Procedure

In this section a procedure for computing the time to perform the polyphase merge is presented. The unit of time is that required to read, process, and write one unit string.

The matrix of polyphase numbers for order R is

Fig. 2. Times To Read Each String To Complete Sort.

String Position	P	Level → 6	5	4	3	2	1	0
1	0	6 6 6	5 5 5	4 4 4	3 3 3	2 2 2	1 1 1	0
2	1	5 5 5	4 4 4	3 3 3	2 2 2	1 1		
3	1	5 5 5	4 4 4	3 3 3	2 2			
4	2	4 4 4	3 3 3	2 2 2	1			
5	1	5 5 5	4 4 4	3 3				
6	2	4 4 4	3 3 3	2 2				
7	2	4 4 4	3 3 3	2				
8	1	5 5 5	4 4					
9	2	4 4 4	3 3					
10	2	4 4 4	3 3					
11	3	3 3 3	2 2					
12	2	4 4 4	3					
13	3	3 3 3	2					
14	1	5 5						
15	2	4 4						
16	2	4 4						
17	3	3 3						
18	2	4 4						
19	3	3 3						
20	3	3 3						
21	2	4						
22	3	3						
23	3	3						
24	4	2						

computed in the usual way as follows:[1]

$$N_{i,k}(R):\ N_{o,o} = N_{o,1} = 1;$$
$$\underset{k=2}{\overset{R+1}{\Gamma}}\ (N_{o,k} = 0);$$
$$\underset{i=1}{\overset{I}{\Gamma}}\ (N_{i,o} = N_{i-1,o} + (R-1)N_{i-1,1}\ ;$$
$$\underset{k=1}{\overset{R}{\Gamma}}\ (N_{i,k} = N_{i-1,1} + N_{i-1,k+1});\ N_{i,R+1}\,0 =). \tag{1}$$

I is the number of levels for which N is being computed. $N_{i,o}$ is the maximum number of unit strings which may be sorted by the polyphase merge procedure of order R, using $(R+1)$ tapes and i levels. The initial dispersion scheme must place $N_{i,k}$ unit strings on tape unit k.

We now define a polyphase parameter vector for order R as follows:

$$P_j(R):\ P_1 = 0;\ \underset{i=1}{\overset{I-1}{\Gamma}}\ (k = N_{i,1}\ ;\ \underset{j=1}{\overset{N_{i,2}}{\Gamma}}\ (P_{j+k} = 1 + P_j)). \tag{2}$$

An example of P_j appears in Figure 2.

Now the total number of records read during the polyphase sort of $N_{i,o}$ unit strings with i levels is:

$$T_i(R):\ T_i = \sum_{k=1}^{R}\left(\sum_{j=1}^{N_{i,k}}(i - P_j)\right). \tag{3}$$

[1] A capital gamma is used as a repeat operator in a manner analogous to the use of capital sigma for a summation operator.

This result is obtained immediately by inspection of Figure 2.

This same result can be obtained by the following slightly different procedure. Define matrix $A_{i,j}(R)$ as the number of unit strings read $(i-j)$ times in the polyphase procedure using i levels.

$$A_{i,j}(R):\ \underset{i=0}{\overset{I}{\Gamma}}\left(\underset{j=0}{\overset{i}{\Gamma}}\ (A_{i,j} = 0);\ \underset{k=0}{\overset{R-1}{\Gamma}}\cdot\left(\underset{m=1+N_{i,R-k+1}}{\overset{N_{i,R-k}}{\Gamma}}\ (A_{i,P_m} = A_{i,P_m} + R - k)\right)\right).$$

Now $T_i = \sum_{j=0}^{i}(i-j)A_{i,j}$.

In order to compute the time to sort n unit strings with i levels we define the dispersion (or dummy) matrix D as follows:

$$D_{i;j,k}(R):\ D_{i,j,k} = \begin{cases} 1 \text{ if the } j\text{th-position on tape unit } k \text{ is occupied by a unit string,} \\ 0 \text{ if the position is a dummy (i.e. not occupied).} \end{cases}$$

Clearly, $n = \sum_{k=1}^{R}(\sum_{j=1}^{N_{i,k}} D_{i;j,k})$.

The time required to sort the n-unit strings with dispersion D is:

$$T_i(R, n, D):\ T_i = \sum_{k=1}^{R}\left(\sum_{j=1}^{N_{i,k}}(i - P_j)D_{i;j,k}\right). \tag{4}$$

This is the key expression for computing the relative sorting times corresponding to various dispersion schemes.

5. New Dispersion Algorithms

The optimum dispersion algorithm for a given number of unit strings and a given number of levels is achieved by filling in those string positions which correspond to the largest values of P_j. In principle one can evaluate eq. (4) for various levels and the best dispersion for each level to find the optimum procedure for a given number of unit strings. Computer programs have been used to do the equivalent of this and to determine for what range of numbers of unit strings a given number of levels is optimum. This data is presented in Table II.

The values in Table II show the maximum number of strings that can be optimally dispersed at each level for a given order. For example, suppose 100 unit strings must be merged using order 3. From Table I, it is clear that these strings can be handled with 7 levels. However, from Table II we see that 8 levels should be used since 95 strings is the maximum for which the 7-level dispersion is optimum. For large orders, the increase in level for optimum dispersion of large numbers of strings is rather dramatic.

It is easy to see that the problem associated with the bookkeeping of the dummies on the various tapes for the optimum procedure can become a very complicated one. It is necessary to know the extent to which all this work achieves advantage before undertaking it. An evaluation of this result is presented in Section 6. Even though this advantage can be as high as 10 to 15 percent in certain cases, it is probably still not worth the difficulty in the procedure. Nevertheless, a technique for achieving the optimum distribution is presented here.

To achieve the optimum one must know in advance how many unit strings are going to be sorted. It would be best to know this precisely, but in any case a close estimate would be necessary. Knowing this number, one can proceed with the following algorithms.

Optimum Dispersion Algorithm

Assume given:

R = order of polyphase merge,

n = number of unit strings,

$N_{i,j}(R)$ = polyphase numbers.

Step 1. Select the number of levels I to be used by reference to Table II. The optimum I is defined by R and n.

Step 2. Form the polyphase parameter vector P_j using eq. (2). There will be ($J = N_{I,1}$)-elements in the vector P_j.

Step 3. Form the vector Q_j. Each Q_j is a triple thus:

$\Gamma_{j=1}^{J} (Q_j = \{P_j; j; o\})$.

Step 4. Sort the elements of Q_j into ascending order. Now $Q_j = \{P'_j; j'; 0\}$.

Step 5. Set all $P'_j = 0$ in Q_j.

Step 6. Distribute unit strings to the various tapes filling position J' (from Q_J) first $(J - 1)'$ (from Q_{j-1}) next, etc., until the n unit strings are exhausted. At the same time enter $n_{j'}$ in the triple Q_j where $n_{j'}$ is the number of tapes with a unit string placed in position j'. One now has the unit strings distributed. The vector Q_j now consists of triples $Q_j = \{0; j'; n_{j'}\}$. See example in Figures 3 and 4.

Step 7. Sort the elements of Q_j into ascending order using j' as the key. Now $Q_j = \{0; j; n_j\}$.

At this point the optimum dispersion is complete and polyphase sorting can proceed. The information concerning the dispersion is contained in the vector Q_j. This is equivalent to having the dispersion matrix D used in eq. (4). An example of this dispersion is shown in Figure 3 for $R = 5$ and $n = 16$. Figure 4 shows the corresponding P vector and the Q vector following Steps 3–7.

This optimum procedure requires an excessively large vector P and complicated merging logic for large n. A simple modification greatly simplifies the logic. This modification requires that all empty string positions occur in the first $N_{I,R}$ (R) locations on the tapes. Thus after the first merge pass of the polyphase procedure only one tape will have vacant string positions. It also turns out that this approach never requires more than one level in excess of the minimum possible number of levels. The maximum number of strings for each order and level under these circumstances is presented in Table III. The procedure that should be used for the Modified Optimum Dispersion Algorithm is given below.

Table II. Maximum Number of Strings for Optimum Dispersion for Given Orders and Levels

Level I ↓ / Order R→	2	3	4	5	6	7
1	2	3	4	5	6	7
2	3	5	7	9	11	13
3	5	9	13	13	16	19
4	8	17	22	28	19	23
5	13	31	34	42	52	26
6	21	54	75	60	72	87
7	34	95	108	153	97	114
8	55	172	243	215	282	147
9	89	279	358	268	385	167
10	144	534	455	778	480	639
11	233	819	1196	1033	554	791
12	377	1634	1562	1248	1995	921
13	610	2400	4033	3909	2485	1016
14	987	4958	5378	4969		
15	1597	7028				

Modified Optimum Dispersion Algorithm

Assume given:

R = order of polyphase merge,

n = number of unit strings,

$N_{i,j}(R)$ = polyphase numbers.

Table III. Maximum Number of Strings for Modified Optimum Dispersion for Given Orders and Levels

Order $R \rightarrow$ / Level I ↓	2	3	4	5	6	7
1	2	3	4	5	6	7
2	3	5	7	9	11	13
3	5	9	13	17	21	25
4	8	17	25	32	39	47
5	13	31	48	62	78	93
6	21	56	92	122	153	185
7	34	105	172	242	304	363
8	55	189	333	467	604	732
9	89	346	631	920	1181	1450
10	144	630	1215	1792	2355	2879
11	233	1134	2320	3524	4645	5747
12	377	2090	4453	6884	9199	11352
13	610	3733	8514	13502	18240	22701
14	987	6932	16401			
15	1597	12459				

Fig. 3. Example of Optimum Dispersion.

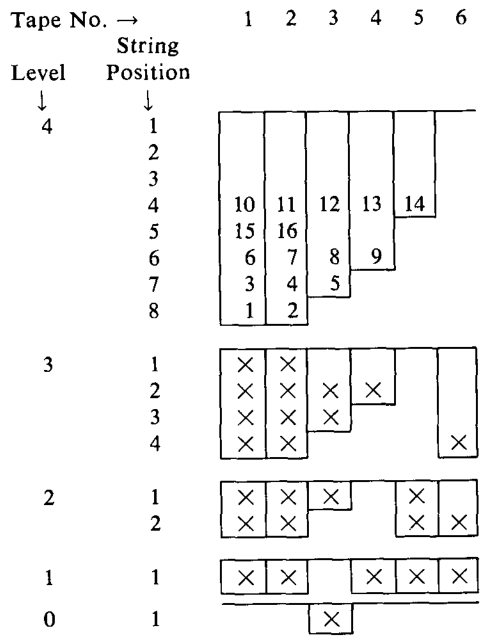

Table IV. Average Relative Speeds
Percentage Variation from Horizontal

Algorithm R ↓	Optimum	Mod. Opt.	Nominal	Mod. Nom.	Horizontal	Diagonal	Vertical	Rectangular	Worst
	←BETTER			→		←POORER			→
	1.53	1.34	0.55	0.27	0	0.09	0.20	0.87	2.00
	4.40	2.40	1.25	0.34	0	0.19	0.43	1.22	3.74
	7.93	2.72	1.54	0.45	0	0.14	0.81	1.16	4.87
	10.92	2.98	1.85	0.34	0	0.26	1.14	1.14	5.71
	14.06	2.88	1.92	0.45	0	0.26	1.55	0.87	6.62
	16.27	3.12	2.20	0.36	0	0.24	1.78	0.85	7.59

Step 1. Select the number of levels I to be used by reference to Table III. The optimum I is defined by R and n.

Step 2. Using $I - 1$ for I, form the polyphase parameter vector P_j using eq. (2). There will be $J = N_{I-1,1} = N_{I,R}$ elements in the vector P_j.

Step 3. Perform steps 3, 4, and 5 from the Optimum Dispersion Algorithm.

Step 4. Distribute $(N_{I,k} - N_{I,R})$-unit strings on tape k for $k = 1$ to $R - 1$, inclusive.

Step 5. Perform steps 6 and 7 from the Optimum Dispersion Procedure.

At this point the Modified Optimum Dispersion Algorithm is complete and polyphase merging can proceed. The data equivalent to the dispersion matrix D is contained in the vector Q_j.

If the number n of unit strings is not even approximately known in advance, one may use some variant of the above dispersion algorithms. If the choice is to fill the tapes for a given level before proceeding to the next level, one should use the Horizontal Algorithm or some variant of it. Under this circumstance the best one can do is to construct $N_{i,1} - N_{i-1,1}$ elements of the parameter vector P_j for level i. Then following a procedure analogous to the Optimum Dispersion Algorithm one can determine where to place the unit strings until the supply is exhausted or until the tapes are filled for level i. Then the process is repeated for the next level. Let us call this the Nominal Dispersion Algorithm.

A modification will simplify the logic of this process. In this Modified Nominal Dispersion Algorithm, distribute all unit strings as in the Horizontal Algorithm. Now after the dispersion is complete, one may determine the merging logic by the algorithm below.

Modified Nominal Dispersion Algorithm

Given:

R = the order of the merge,

n = the number of unit strings dispersed,

I = the level in which dispersion is completed

$N_{i,k}(R)$ = the polyphase numbers.

Step 1. Let $L = N_{I,R} - N_{I-1,R}$ and $T = N_{I,o} - RL$

Step 2. If $n \leq T$, dispersion is complete. Proceed

Fig. 4. Example of Optimum Dispersion—P and Q vectors.

j	P	Q following				
		Step 3	Step 4	Step 5	Step 6	Step 7
1	0	0,1,0	0,1,0	0,1,0	0,1,0	0,1,0
2	1	1,1,0	1,2,0	0,2,0	0,2,0	0,2,0
3	1	1,3,0	1,3,0	0,3,0	0,3,0	0,3,0
4	2	2,4,0	1,5,0	0,5,0	0,5,2	0,4,5
5	1	1,5,0	2,4,0	0,4,0	0,4,5	0,5,2
6	2	2,6,0	2,6,0	0,6,0	0,6,4	0,6,4
7	2	2,7,0	2,7,0	0,7,0	0,7,3	0,7,3
8	3	3,8,0	3,8,0	0,8,0	0,8,2	0,8,2

with the polyphase merge exactly as in the Horizontal Algorithm case. Otherwise go to step 3.

Step 3. Prepare the first L elements of P_j as in eq. (2).

Step 4. Perform steps 3, 4, and 5 of the Optimum Dispersion Algorithm for $J = L$.

Step 5. Compute t and r such that $n - T = Rt + r$ and $0 \leq r < R$.

Step 6. Construct $Q_j = \{0; j'; r\}$ for $j = L - t$ and $Q_j = \{0; j'; R\}$ for $j = L - t + 1$ to L.

Step 7. Perform step 7 of the Optimum Dispersion Algorithm.

The polyphase merge may now proceed as in the Modified Optimum Dispersion Algorithm case.

6. Evaluation of Dispersion Algorithms

In order to evaluate the various algorithms computer programs were prepared to calculate the number of unit strings which must be read and written to perform the polyphase merge. These calculations were performed for the number of unit strings n = 1–10000, order R = 2–7, and the nine dispersion algorithms: (1) Optimum, (2) Modified Optimum, (3) Nominal, (4) Modified Nominal, (5) Horizontal, (6) Diagonal, (7) Vertical, (8) Rectangular, and (9) Worst.

The first eight algorithms are described above. Number 9, the "Worst," was evaluated assuming the worst possible dispersion under the constraint of completing the tapes for level i before proceeding to level $i + 1$.

Of the simple algorithms (numbers 5–8) the Horizontal is best. Hence, it is used as a basis for comparison. To illustrate the calculation of relative speed let $t_a(n)$ equal the number of unit strings read using algorithm a for n strings, $t_h(n)$ equal the number of unit strings read using the Horizontal Algorithm for n strings. Now

$$\% \text{ Relative speed } = \frac{1}{100} \sum_{n=1}^{10000} \left(1 - \frac{t_a(n)}{t_h(n)}\right).$$

The results of this calculation are tabulated in Table IV.

Examination of these results indicates that current users of Diagonal, Vertical, or Rectangular Algorithms are doing from $\frac{1}{4}$ to $1\frac{1}{2}$ percent more work than necessary. It also shows that an additional $2\frac{1}{2}$ to 3 percent improvement can be made by using the Modified Optimum Algorithm for orders 3–7. The penalty for this gain is the use of the parameter vector P_j or some equivalent.

Referee's Comment. The horizontal distribution, the fact that additional levels can be advantageous even over "perfect" distributions and the essential idea of Figure 2 and the optimal procedure were all presented by B. Sackman and T. Singer of the ACM Sort Symposium in 1962. Curiously their paper was never published. I think it exists as a Mitre Corporation report, which can probably be obtained from Sackman or the Clearinghouse.

I don't believe Sackman and Singer noticed that the P vector simplifies the analysis. But one thing Sackman did discover, which is not in the present paper, is the idea of starting with non-Fibonacci distributions. For example, placing seven strings on tape 1 and four strings on tape 2 yields

Place dummy strings at the 5s; sorting nine elements with a score of 31 beats the optimum polyphase method, which has a score of 32! The question of optimum distribution is therefore not completely resolved.

Received June 1970; revised March 1971

References

1. Betz, D.K., and Carter, W.C. New merge sorting techniques. Proc. ACM 14th Nat. Conf., 1959.
2. Flores, I. *Computer Sorting*. Prentice-Hall, Englewood Cliffs, N.J., 1969.
3. Gilstad, R.L. Polyphase merge sorting—an advanced technique. Proc. EJCC Dec. 1960, Spartan Books, New York, 143–148.
4. Gilstad, R.L. Read-backward polyphase sorting. *Comm. ACM 6*, 5 (May 1963), 220–223.
5. Malcolm, W.D. String distribution for the polyphase sort. *Comm. ACM 6*, 5 (May 1963), 217–220.
6. Manker, H.H. Multiphase sorting. *Comm. ACM 6*, 5 (May 1963), 214–217.
7. Mendoza, A.G. A dispersion pass algorithm for the polyphase merge. *Comm. ACM 5*, 10 (Oct. 1962), 502–504.
8. Reynolds, S.W. A generalized merge algorithm. *Comm. ACM 4*, 8 (Aug. 1961), 347–349, addendum, 495.