Experiences in using a Decomposition program

By E. M. L. Beale, P. A. B. Hughes, and R. E. Small*

We had a particular problem in oilfield operations to solve. This problem naturally decomposed into sub-problems, and this led us to develop Decomposition within the linear programming system that we were using. It was of great help to us when writing the Decomposition program to have this problem in mind. The development of the method went hand in hand with the solution of the oilfield problem.

1. Outline of the problem

The problem concerned the production of oil from several different fields to meet a fixed overall target over a finite span of years. Typically we were concerned with seven different fields supplying three outlets (ports or refineries) over a span of twelve years.

The problem was formulated as a linear program, the object being to meet the overall target and to maximize an expression representing the net profit over the time span being considered. It naturally decomposed into sub-problems because the linking equations between the various fields were quite few in number.

Dantzig and Wolfe's Decomposition principle shows how to take advantage of the special structure of linear programming problems that can be considered as separate sub-problems with a relatively small number of linking equations. (Dantzig and Wolfe (1960), Dantzig and Wolfe (1961) and Dantzig (1963)). The linking equations are grouped together into what is known as the master problem, whilst each sub-problem contains the constraints and equations that can naturally be grouped together.

In this problem the operations of each field under consideration were expressed in separate sub-problems. The constraints in the sub-problems deal with the construction of new production facilities in each of the years being considered. These facilities include new oil wells, and also plant such as gas/oil separators which are required to handle the oil once it has reached the surface. There are also equations representing the productive capacity of both existing and new wells, and constraints on the upper limits of the capacity of the field. A solution to a sub-problem is a way of operating a field, i.e. a set of annual productions with the corresponding investments required to make the productions possible.

The master problem consists of the linking equations dealing with the supply of oil from the fields to the outlets. It also deals with the possibility of exploring for new oilfields. And, of course, it contains the main supply equations which say that the sum of productions from all the fields in any year must equal the overall target for that year.

* *All of C-E-I-R Ltd.*, 30–31 *Newman Street, London, W.*1.

2. The Decomposition principle

The principle of Decomposition is by now quite well known, but it is, perhaps, worth giving a very brief description of it before going on to describe its use. This will serve to clarify the terms that we are using.

It makes use of two fundamental facts. Firstly, a polyhedron can be represented either as the intersection of a set of half spaces or as the set of all possible linear combinations of its vertices. So the constraints that form a sub-problem can be replaced if one says that the variables of the sub-problem must take values represented by positive linear combinations of the vertices of the sub-problem. If the sub-problem is of any size, it is, of course, not feasible to consider all the vertices when making these linear combinations. A very restricted set can be used so long as it covers the region of the polyhedron with which we are concerned when seeking the optimum solution. A vertex of a sub-problem is the same as a basic feasible solution to that problem, and when we include it in the master program it is called a *proposal.*

We must now consider how we are going to chose relevant proposals from a sub-problem during the course of the computation. We use the second fundamental fact, that an optimal solution to a linear programming problem remains optimal if some or all of its constraints are withdrawn, and the variable terms in such constraints are multiplied by suitable Lagrange multipliers, or "π-values," and added to the objective function. So the procedure is to solve the master problem with a restricted set of proposals from the sub-problems, and obtain the corresponding π-values on the rows which are common to both the master problem and the sub-problem. The sub-problems can then be solved with the common row constraints replaced by additions (based on these π-values) to their objective functions. For each sub-problem the vertex corresponding to the optimum solution will form an additional proposal to the master problem. The master problem is then re-solved and the new set of π-values on the common rows are used to repeat the cycle. The step from one optimal solution of the master problem to the next one (based on a new set of proposals) is called a *major iteration*, to distinguish it from a normal iteration of the simplex method which we shall call a *minor iteration.*

This procedure terminates when none of the proposals generated by the sub-problems are worth introducing into the basis of master problem. When this is the case the π-values on the common rows in the master problem will remain unaltered, and so no new proposals will be generated if we make a further pass through the sub-problems. It can be shown quite easily that this whole procedure must terminate in a finite number of major iterations and that it must produce the overall optimum solution. Rigorous descriptions and proofs can be found in the references that we have given.

3. Development of the algorithm

We were faced with the oilfield production problem that we have described. As we have shown, it seemed to decompose naturally into sub-problems, and so we set about introducing Decomposition into our existing linear programming system, LP/90/94. A twelve-year problem could have been solved in un-decomposed form, but we believed that the machine time required would be excessive. As seems usual in such work, the answers were required "as soon as possible." Nevertheless we decided that it would be worth the effort of writing the decomposition program for this particular problem so long as we did not try to be too elaborate.

Wherever possible we made use of features of our existing LP code, some of which proved extremely helpful. At an early stage we took some short cuts for the convenience of programming. These fortunately proved an advantage when later we came to make changes to the system.

The first short cut was to keep the common rows as explicit separate rows in all the sub-problems, and to use a highly composite objective function. The alternative would have been, in each major iteration, to combine the contributions from the common rows to the sub-problem's objective functions into a single cost row. The former method, though a little more wasteful of computer storage, was easy to program as it used a scaling feature already present in LP/90/94 with only very minor alterations. This decision proved fortunate later, when we decided to submit intermediate solutions as proposals to the master problem. Any linear programming routine always produces a current right-hand side on all active rows of the problem. These current values on the composite cost rows of a sub-problem are precisely the coefficients that we require in the common rows of the master problem if we are to use the intermediate solution as a proposal.

Another of the short cuts was not to delete any proposals once they had been entered into the master problem. This decision was taken in the first instance out of laziness. It was justified on the grounds that in the early stages there would not be too many proposals in the master problem, whilst in the later stages, though there would be a fair accumulation of proposals, this would not matter too much since only a few minor iterations would be required in the master problem. In fact, we believe that the disadvantages of accumulating a large number of proposals are far outweighed by the advantages of giving the master problem a large range of choices.

A third short cut was used to obtain the final solutions to the sub-problems. At the overall optimum, we have in the master problem the weights that are attached to the various proposals from each sub-problem. The natural way to find the corresponding solution would be just to compute the weighted sum of these proposals from each sub-problem. This has the disadvantage that the program has to keep track of the complete solution of the sub-problem, each time that that solution is used as a proposal. In our problem, for example, the oil productions in each year from the field are the only entries required in the common rows (apart from the cost of this schedule which goes in the overall objective function). The complete solution would have all the details of the various investments required to make the schedule possible. We decided that rather than keep track of all this information each time a solution was used as a proposal, it was easiest to write a special piece of program to enable us to re-solve a sub-problem by working from the master problem solution. The program forms right-hand sides for the sub-problem common rows. In this case these right-hand sides form the yearly production schedule for the field (which comes from the multiplication of each proposal from that field that appears in the optimum basis, by its weight in the optimum solution, and the summing of the resultant vectors).

This solution method for the sub-problems was included for computational convenience. It naturally followed from the "getoff" feature of LP/90/94 that we were already using. An LP system must be able to save a problem in packed form on magnetic tape in such a way that it is easy to restart the problem from that tape. We used this feature to enable us to get a sub-problem off the computer, once it had been solved, and to write it on to tape in a form that made it easy to call back and restart the problem at the next major iteration.

So we started to try to solve some trial problems. The only special trick that we used was to incorporate what one might have considered as the most important sub-problem into the master problem. At first we put into the master problem the largest of the fields; later we changed the formulation and included Distribution and Exploration in the master and had all the fields in the sub-problems. This procedure obviously slows down the master problem, but we believe that it reduces the number of major iterations required, since it provides more realistic and stable π-values on the common rows. It gives the master problem some flexibility to adjust for deficiencies or peculiarities in the available sets of proposals from the sub-problems.

We soon found, as others have done before us, that a straightforward Decomposition system was so slow as to be more or less useless. The main trouble was that, despite including one of the fields in the master problem, the π-values on the common rows oscillated wildly from one major iteration to another. For the first few major iterations the master problem did not have a sufficient supply of sensible proposals from the sub-problems from which to chose. This meant that oil would be very much at a premium in some years, whilst there would be a glut in others. This behaviour was reflected by widely varying π-values in the common rows. These π-values are equivalent to the price which the master problem is prepared to pay for oil from the various fields in the years under consideration. If, for example, the π-value in the master problem optimum on the common row that defines the production from a field in a certain year is negative, then this will lead to zero production from that field in that year in the next optimum solution to the sub-problem. So we are trapped in a circle from which we can only slowly emerge if at all. A poor set of alternative proposals in the master problem gives rise to erratic π-values on the common rows. These in turn lead to nonsensical optimum solutions to those sub-problems (particularly to blank years) which form the additional set of proposals for that major iteration. This new set will tend to have blanks in the years where before there were gluts, and vice versa. So now the π-values will be inclined to veer in the opposite direction—and so on.

We decided that we must give the master problem a wider choice of alternative proposals, particularly during the first few major iterations. We altered the program so as to allow us to introduce intermediate solutions to the sub-problems as proposals to the master problem. When doing this, the short cuts that we had taken initially worked to our advantage, and the program alterations were quite simple. We allowed the frequency of the use of intermediate solutions to be controlled during computation from the on-line card reader. This was of great help in experimentation. We felt that during the early stages we should submit these intermediate solutions very frequently so as to give the master problem a chance to settle down quickly. As the solution proceeded, in the interests of keeping down the size of the master problem, we reduced the frequency. And then in the final stages, when we were very close to the overall optimum and when the sub-problems required only very few minor iterations during each major iteration, we increased the frequency again. This then gave the master problem a wide set of proposals in the region close to the optimum solution with which to form the overall optimum.

This change to our program helped to speed up the process, but not very much. In the early stages, a high proportion of the intermediate solutions were nonsense solutions. We realized, on looking at these proposals, that by iterating to the optimum solution of the sub-problems, often we were not only wasting time but we were perhaps doing actual harm by providing sets of bad proposals. Next we decided to introduce a facility to cut off sub-problems before they reached their optima. This proved of very great help in speeding up the process.

Just as for the frequency of the proposals, we included the cut off feature in such a way that it could be controlled from the on-line card reader. At the beginning of each major iteration we could specify an upper bound on the number of minor iterations per sub-problem. As the computation proceeded we progressively relaxed this bound so that after about the fourth major iteration it was usually inoperative. It was particularly useful during the first two major iterations. It had the double effect of helping the π-values to settle down faster, and of cutting out a large number of wasteful minor iterations in the sub-problems.

4. The matrix generator

Though we had made considerable reductions to the running time it was apparent that we could make further improvements if only we could provide good enough trial solutions. To start the process we must provide a set of π-values to form the initial objective functions for the sub-problems. In order to do this we solve the master problem with a set of trial solutions as proposals. The better the set of trial solutions that we provide, the more realistic will be the initial set of π-values, and hence the better the sets of proposals produced during the first major iteration.

It was clear that a single trial proposal from each sub-problem was quite useless unless it was realistically priced. Even then it was fairly useless since it did not provide the master problem with enough flexibility to deduce realistic differential coefficients, or π-values, on the common rows. We now concentrated on producing a comprehensive set of trial proposals, each realistically priced, of the form that naturally lead to useful linear combinations. Here we were greatly helped by the structure of the problem. The sub-problems, although formally LP problems, could in fact be solved by elementary means for given contributions to the common rows. If we specified for a field the production in each of the years, then it was quite easy to compute the investments required to make the production possible, and hence to compute its cost (or price). We suspect that this is not a particularly unusual feature of LP problems that lead naturally to a decomposed formulation.

We now concentrated on incorporating in the matrix generator program, routines to provide good sets of realistically priced trial solutions and good starting bases for the sub-problems. We were helped by our earlier decision on the solving of the sub-problems. This meant that we did not have to label the trial solutions with their component vectors in the sub-problems. We just needed the production schedule and its cost.

As we developed the techniques of providing sets of trial solutions, we became more skilled at giving the problem a good start. The importance of this is shown by the considerable saving in running time if one can

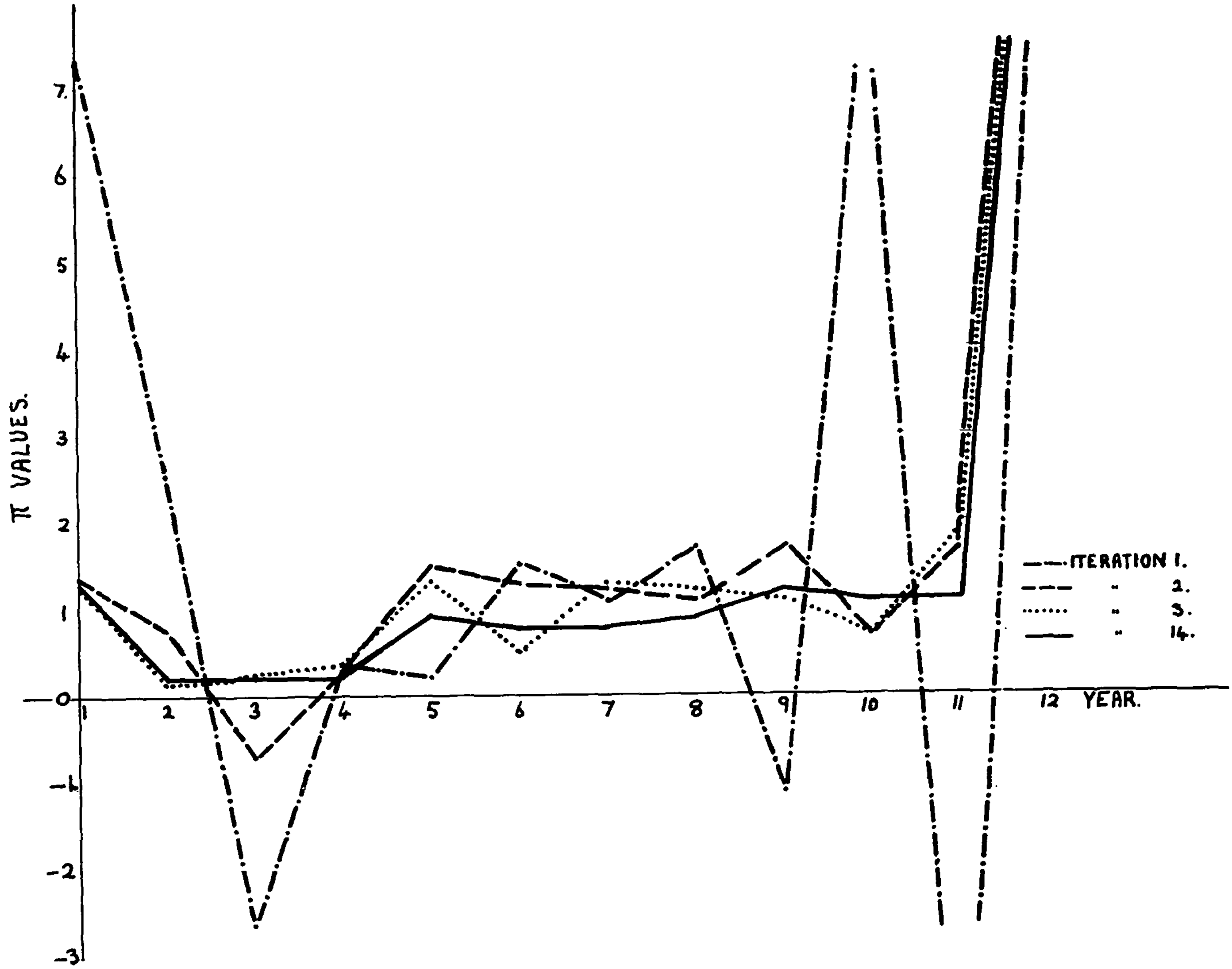

Fig. 1.—Common row π-values

generate π-values at the start of the problem that are quite like the π-values at the final optimum.

5. An example

The three sets of graphs, **Figs. 1, 2** and **3**, illustrate various intermediate stages in one of the sub-problems during a production run of the model. Each figure has four separate graphs. These relate to the first, second, third and fourteenth major iterations. The problem was considered close enough to the final optimum solution to make it not worth-while going on beyond fourteen major iterations.

Fig. 1 shows the π-values on the sub-problem common rows at the previous optimum to the master problem.

Fig. 2 shows the annual rates of production in the pool at the end of the iterating process for the sub-problem. The π-values shown in Fig. 1 are incorporated in the objective function for the sub-problem. The sub-problem is then solved, and the rates of production shown are part of the solution. This may or may not be an optimum solution, depending on whether or not the sub-problem has been cut off before reaching its optimum. In Fig. 2 the solutions for the first and second major iterations are not optima, and those for the third and fourteenth major iterations are optima.

Fig. 3 shows the annual rates of production for the pool in the optimum solution to the *master* problem at the end of the major iteration.

The sequence runs from Fig. 1 to Fig. 2 to Fig. 3. The π-values shown in Fig. 1 lead to the generation of the proposal shown in Fig. 2, and this proposal is then available to the master problem when deriving the optimum schedule for the pool as shown in Fig. 3.

The first set of π-values comes from the first optimum solution to the master problem. At that stage the master problem has available only trial solutions produced by the matrix generator. The smoothness of this

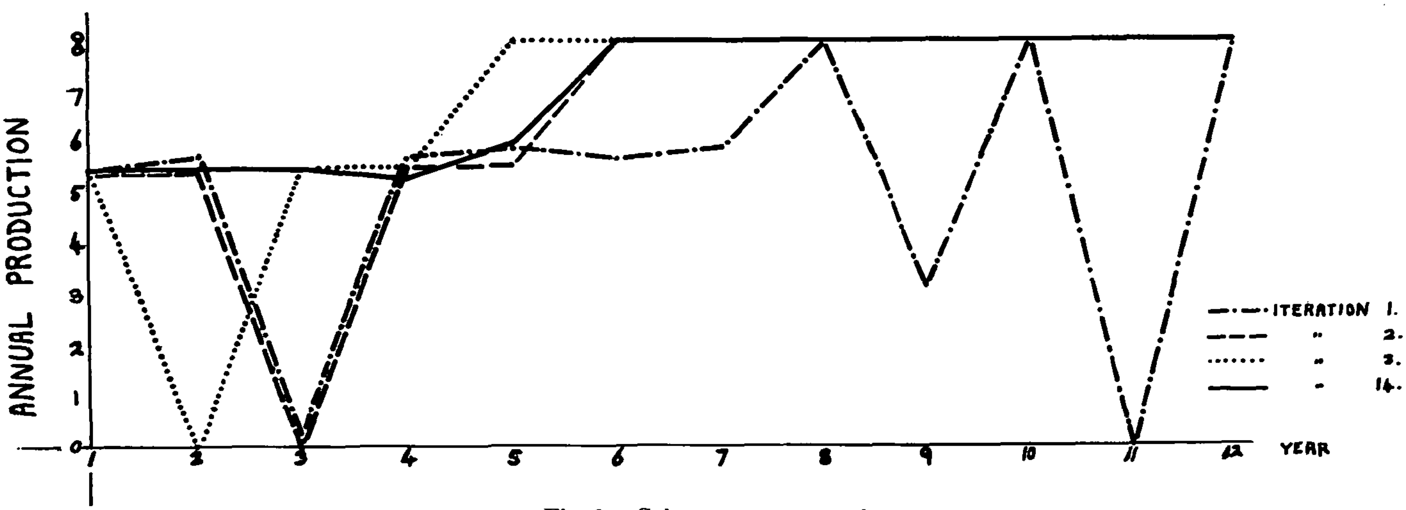

Fig. 2.—Sub-program proposal

first set of π-values is a measure of our success in making a good start towards solving the problem. In the example shown, the π-values are negative in the third, ninth and eleventh years. This leads to the first proposal shown in Fig. 2. It has zero rates of production in the third and eleventh years. The blank eleventh year is a nonsensical result and it is quickly eliminated from the proposals in subsequent major iterations. The sharp dip in production in the ninth year is also a nonsensical result. This rate of production would have been zero if we had allowed the sub-problem to continue to its optimum on the first major iteration. As it was, we cut off the sub-problem after ten minor iterations. This saves the time which would be wasted in the sub-problems when iterating under a very crude set of π-values. The set of π-values for the second major iteration has π-values for the ninth, tenth and eleventh years quite close to their final values. The fluctuations in the π-values for the end years is fairly soon smoothed out.

The dip in the third year is in a different category. Fig. 3 shows that, in the final optimum solution for this field, the rate of production does dip in the first few years. The rate of production in the first year under consideration was more or less fixed. In subsequent years the production could be switched to other fields by building extra capacity. That was, in this case, the optimum solution, and the reserves of the field shown in the example were saved for future use in meeting the high overall target of the last few years.

The sub-problem's optimum production rates for the early years were very sensitive to small changes in the π-values. Once the π-value for any particular year drops below the unit production cost, then the sub-problem's optimum solution will have zero production for that year. The second year in the third major

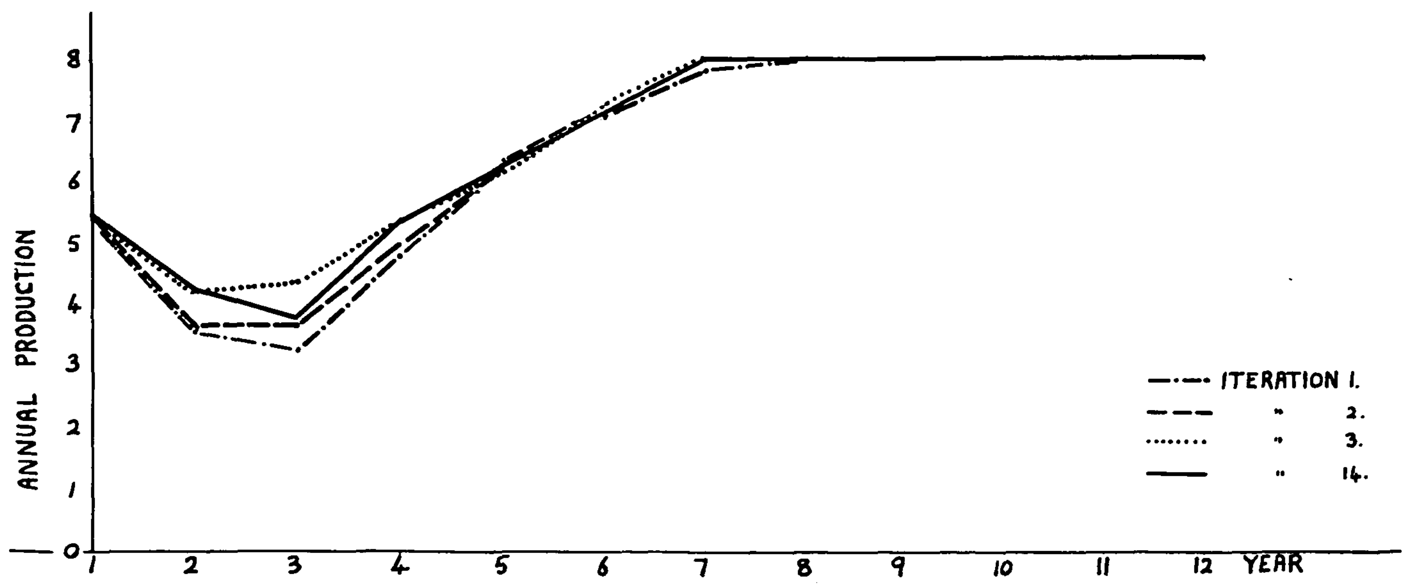

Fig. 3.—Master problem optimum

iteration is an example of a positive but small π-value leading to a zero rate of production.

In this example we have only shown a selection of the major iterations, and for each iteration only the last proposal from the sub-problem. We have not attempted to show in detail the full set of proposals nor their use in the master problem optima. But it is perhaps worth making a few points on this. The first proposal shown was never incorporated in a master problem optimum. The dips in the ninth and eleventh years make it a particularly unattractive schedule. The proposals from the second and third major iterations were included in master problem optima. They combined with some of the trial proposals that we had provided to give the dips in production shown in Fig. 3. Gradually the solution to the problem settled down. The trial solutions were used less and less, and the newly generated proposals more and more. The final master problem optimum used proposals from the fourth, sixth, seventh, eighth and tenth major iterations. Those proposals that entered with the largest weights were not, in fact, sub-problem optima. This illustrates the value of providing the master problem with a wide choice of proposals, and not confining the set to sub-problem optima.

6. Some features of the program

The Decomposition program has been added to LP/90/94 in such a way that all the regular features of the LP program are available during execution of the Decomposition program. Thus the maximum row size for the master problem or for any sub-problem (including the number of common rows) is 1,023. As we have already explained, there are two features of LP/90/94 which some people consider old-fashioned but which we find are extremely useful when one is extending the system to solve new types of problems: firstly, the use of six character names rather than sequence numbers to identify vectors, and secondly the routine use of control cards read on-line to govern the operation of the program. It proved very useful in the Decomposition program to be able to label proposals explicitly with the name of the sub-problem or origin and the number of the proposal. The label also incorporates a "flag" to indicate that this is a proposal. Two digits were assigned to the sub-problem number; this gives a capacity of 99 sub-problems.

The maximum number of rows which can be handled by this program is obviously about 99,000. LP/90/94 has no restriction on the number of vectors which may be used, provided that the whole problem will fit on one reel of magnetic tape. Facilities are included in the program to enable one to revise any part of the problem, and to remove existing proposals from the master problem if these are invalidated by such a revision.

The ability to control the operation of the program from the on-line card reader has a very great advantage over present controls in the development of operating techniques for this type of program. One can read the on-line printer to obtain information on the progress of the calculation, and place appropriate control cards in the card-reader at that time. One can also change one's mind very easily by substituting one card for another. All control cards are printed on- and off-line to provide a complete record of a run.

7. Some running experiences

A number of matrices have been decomposed, and their solution has been attempted by the use of this program. It soon became apparent that the control cards used were absolutely critical in minimizing the overall running time. For instance, it is obvious that if the master problem is filled with proposals, all very similar and all far from the optimum during the early stages of a run, the program will spend a large part of each minor iteration in the master problem pricing these proposals only to discover that they aren't worth considering. Similarly, until the master problem becomes feasible and the π-vector represents the marginal costs, in general the sub-problems will not move towards the desired optimum, and so these should be restricted in the number of minor iterations they are allowed to make. Once the master problem is feasible and the π-vector has settled down, the controls in use have less effect on the overall running time, though a rather poor choice will increase the number of major iterations performed in order to reach the global optimum.

The Decomposition program shows a saving in running time for problems of the order of 300 to 500 rows. The exact point at which the program becomes economic compared with LP/90/94 depends entirely on the problems under consideration. When using an IBM 7090, a particular 450-row problem which took 40 minutes to solve using LP/90/94 was solved in 37 minutes when decomposed.

Before we had the Decomposition program working we made a ten-year run on the oilfield problem in undecomposed form in about 5 hours. Subsequently we solved twelve-year runs on this problem when decomposed in about two hours. To make a direct comparison of these two times is unduly flattering to the Decomposition program. The twelve-year runs, though larger, were "easier" in the sense that the various field productions were more constrained. More importantly, we were able by that stage to specify very good starting bases and sets of trial solutions. We could not, of course, have specified trial solutions to the undecomposed ten-year problem but, in the light of subsequent runs, we could have specified a better starting basis.

References

DANTZIG, G. B. (1963). *Linear Programming and Extensions*, Princeton University Press.

DANTZIG, G. B., and WOLFE, P. (1960). "Decomposition Principle for Linear Programs," *Operations Research*, Vol. 8, pp. 101–11.

DANTZIG, G. B., and WOLFE, P. (1961). "The Decomposition Algorithm for Linear Programs," *Econometrica*, Vol. 29, pp. 767–78.