Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Letter to the Editor—Finding Minimal Cost-Time Ratio Circuits

Bennett Fox,

5. ———, "Systems of Distinct Representatives and Linear Algebra," *J. Res. NBS.* 71*B* (*Math and Math Phys.*) **4**, 241–245 (1967).
6. GALE, D., "Optimal Assignments in an Ordered Set: an Application of Matroid Theory," *J. of Combinatorial Theory* **4**, 176–180 (1968).
7. GILBERT, E. N., "Random Minimal Trees," *J. SIAM.* **13**, 376–387 (1965).
8. HELLMICH, K., "The Travel Route of Shortest Distance (Duration)," *Math. Technik Wirtschaft* **4**, 166–175 (1960). (In German.)
9. IVANESCU, P. L., *Pseudo-Boolean Programming and Its Applications* (Lecture Notes in Mathematics, #9), Springer-Verlag, New York, 1965.
10. KRUSKAL, J. B., "On the Shortest-Spanning Subtree of a Graph and the Traveling Salesman Problem," *Proc. Amer. Math. Soc.* **7**, 48–50 (1956).
11. NICHOLSON, T. A. J., "A Sequential Method for Discrete Optimization Problems and Its Application to the Assignment, Traveling Salesman, and Three Machine Scheduling Problems," *J. Inst. of Maths. and its Applications* **3**, 362–375 (1967).
12. OBRUCA, A. K., "Spanning Tree Manipulation and the Traveling Salesman Problem," *Comp. J.* **10**, 374–377 (1968).
13. "NBS Seminar on Matroids," *J. Res. NBS* 68*B* (*Math. and Math Phys.*) **1** & **2** (1965).
14. RILEY, W., III, "A New Approach to the Traveling Salesman Problem," *Bull. Opns. Res.* **7**, Sup. 1, B-11 (1959) (Abstract Only).
15. ROSSMAN, M. J., TWERY, R. J., AND STONE, F. D., "A Solution to the Traveling Salesman Problem by Combinatorial Programming" *Opns. Res.* **6**, 897 (1958) (Abstract Only).
16. WHITNEY, H., "On the Abstract Properties of Linear Dependence," *Amer. J. Math.* **57**, 509–533 (1935).
17. WILF, S. H., "A Mechanical Counting Method and Combinatorial Applications," *J. of Combinatorial Theory* **4**, 246–258 (1968).

FINDING MINIMAL COST-TIME RATIO CIRCUITS

Bennett Fox

The Rand Corporation, Santa Monica, California

(Received June 28, 1968)

In the treatment of routing problems, other authors have used column generators to introduce, into the basis of the master problem, the solution that corresponds to a cycle in a graph with minimal cost-to-time ratio. This subproblem is of independent interest and corresponds to deterministic Markov renewal programming; this note presents an efficient method for its solution, intuitive basis for which is a search for a way to route flow so that the cost-time trade-off is optimal. This flow-circulation problem is solved parametrically by the out-of-kilter algorithm.

WE SEEK a cycle in a network N with minimal cost-to-time ratio, a problem that has already been considered by DANTZIG, BLATTNER, AND RAO.[2]

It appears that the chief difference between their approach and ours is in the method for solving the subproblems, which must be solved parametrically in trial loss rates. Later, the methods will be briefly compared. Our solution's intuitive basis is a search for a way to route flow so that the cost-time trade-off is optimal. This flow circulation problem is solved parametrically by the out-of-kilter algorithm.

Without loss of generality, we suppose that there is at most one directed arc (i, j) leading from node i to node j. Each arc (i, j) in the network has a cost c_{ij} and a time t_{ij}. Let A denote any cycle contained in C, the set of cycles in N. Its cost and time are, respectively, $c(A) = \sum_{(i,j)\epsilon A} c_{ij}$ and $t(A) = \sum_{(i,j)\epsilon A} t_{ij}$. The loss rate $r(A) = c(A)/t(A)$ is to be minimized over $A \epsilon C$. Thus, we seek an A attaining $r^* = \min_{A \epsilon C} r(A)$. To do this, it is convenient to introduce the functions $v(\beta, A) = c(A) - \beta t(A)$ and $z(\beta) = \min_{A \epsilon C} v(\beta, A)$. It is easy to verify that $z(\cdot)$ is piecewise linear, decreasing, and concave.

Consider the β, say β^*, for which $z(\beta) = 0$. Then

$$0 = \min_A \{c(A) - \beta^* t(A)\} = \min_A \{[c(A)/t(A)] - \beta^*\} = r^* - \beta^*.$$

Thus, $\beta^* = r^*$ and a cycle A^* minimizing $v(\beta^*, A)$ is a cycle of minimum cost-to-time ratio.

THE ALGORITHM

We are now in a position to give an algorithm for finding A^*. Deferring the details for the subproblem to the next section, we give the *main routine*:

1. $k = 1$.
2. For some A, say A_1, compute $r(A)$.
3. Set $\beta_k = r(A_k)$.
4. Test whether $z(\beta_k)$ vanishes.
5. If $z(\beta_k) = 0$, terminate with the corresponding cycle A_k that minimizes $v(\beta_k, A)$; A_k is optimal.
6. Find a cycle A_{k+1} such that $r(A_{k+1}) < \beta_k$ and return to 3 with k replaced by $k+1$.

The algorithm is finite, since at any nonterminal step $\beta_{k+1} < \beta_k$, $A_1, A_2, \cdots, A_k$ are distinct, and there is only a finite number of cycles.

It remains to give a method for steps 4 and 6. We use the out-of-kilter algorithm (Ford and Fulkerson,[4] Clasen[1]), taking advantage, for $k > 1$, of the preceding solution. As we shall see in the section after next, we can expect most arcs to remain in kilter after the change from β_k to β_{k+1}. The labeling procedure simplifies, since the arc flows are 0 or 1 and we start with a feasible circulation.

THE SUBPROBLEM

To do step 4 of the main routine, we define relative arc costs $d_{ij}(\beta) = c_{ij} - \beta t_{ij}$. With these relative arc costs, we look for the minimal-cost-flow circulation $C^*(\beta)$ in the network, each arc and node having unit capacity. The node capacities can be handled directly, analogously to Wollmer,[13] or indirectly using dummy arcs with unit capacity.

Since the optimal flows are integral, if $C^*(\beta)$ has multiple cycles (loops with unit flow), they must be disjoint in view of the node capacities. Hence, the cycles of $C^*(\beta)$ are easy to enumerate. Let $A_k{}^\circ$ satisfy

$$r(A_k{}^\circ)=\min_{A\in C^*(\beta_k)} r(A).$$

In the circulation $C^*(\beta_k)$, let x_{ij}^k be the flow in arc (i, j). The value of this circulation is

$$w(\beta_k)=\sum_{i,j} x_{ij}^k(c_{ij}-\beta_k t_{ij}).$$

One easily verifies that $z(\beta_k)\geqq w(\beta_k)$ and $z(\beta_k)=0 \Leftrightarrow w(\beta_k)=0$, provided that $\beta^*\leqq\beta_k$. In the algorithm, the condition is clearly satisfied. Since at each nonterminal step $w(\beta_k)<0$, the number

$$\alpha_k=\sum_{i,j} x_{ij}^k c_{ij}/\sum_{i,j} x_{ij}^k t_{ij}$$

is less than β_k.

Obviously, $r(A_k{}^\circ)\leqq\alpha_k$. Collecting our observations, we have the following rule:

If $w(\beta_k)=0$, $A_k{}^\circ$ is optimal and $\beta^*=\beta_k$. If $w(\beta_k)<0$, set $A_{k+1}=A_k{}^\circ$ in step 6 of the main routine.

When $\beta=\beta^*$, the null circulation also solves the subproblem. However, the miniminal cycle from the preceding iteration is optimal.

We remark that α_k corresponds to the simplex multiplier that would be obtained by decomposing the linear program (I):

$$\min\sum_{i,j} x_{ij}c_{ij},\ 0\leqq x_{ij} \text{ for all } i, j,\ \sum_i x_{ij}=\sum_l x_{jl} \text{ for all } j,\ \sum_{i,j} x_{ij}t_{ij}=1,$$

that determines the minimal cost-to-time cycle, using a minimal-cost-circulation subproblem. We can impose unit capacities on the flow, since α_k is independent of any uniform positive flow amplification. (Geometrically, this corresponds to the fact that the solutions to the flow-balance equations lie in a convex polyhedral cone with vertex at the origin. The optimal solution will lie on a ray extending from the origin and passing through a vertex of the unit hypercube.) Program (I) corresponds to a deterministic Markov renewal program in which there is an optimal policy with a single recurrent chain (*see* JEWELL,[6] FOX,[5] and especially DENARDO AND FOX[3]).

UPDATING

NOTICE THAT β decreases from one iteration to the next and that this decrease is strict if β β^*. Thus, the subproblem's relative costs, which are of the form $c_{ij}-\beta t_{ij}+\theta_i-\theta_j-\sigma_{ij}$, are increased. Here θ_i and θ_j are the node potentials for nodes i and j respectively, i.e., they are the simplex multipliers for the flow-balance equations for nodes i and j. Of course, σ_{ij} is the multiplier for the capacity constraint on arc (i, j). Analysis of the complementary slackness conditions used in the out-of-kilter algorithm reveals that all arcs carrying no flow following the preceding iteration are still in kilter. The only arcs (if any) that go out of kilter are those carrying unit flow and whose relative costs change from negative or zero to positive. During each iteration, the in-kilter arcs remain in kilter. Each breakthrough puts all formerly out-of-kilter arcs in the chain in kilter.

ACCELERATION

STEP 6 of the main routine served to find a β, namely β_{k+1}, such that $\beta^* \leqq \beta_{k+1} < \beta_k$. If we had at hand a β, say β'_{k+1}, such that $\beta^* \leqq \beta'_{k+1} < \beta_{k+1}$, it would then be better to return to step 4 with β'_{k+1}. It is essential that $\beta^* \leqq \beta'_{k+1}$; otherwise, the subproblem gives the null circulation and no optimal cycle is necessarily at hand. However, β'_{k+1} need not correspond to a loss rate for any cycle.

We try to find a suitable β'_{k+1} by extrapolation, but we do not always succeed. As Eric Denardo has pointed out (in a personal communication) the β-intercept, say β°_{k+1}, of the line through the leftmost two points so far obtained on the graph of w versus β satisfies $\beta^* \leqq \beta^\circ_{k+1}$, since w is concave decreasing. We could thus modify the return to step 4 to use β°_{k+1} or β_{k+1}, whichever is smaller.

OTHER APPROACHES

Dantzig, Blattner, and Rao[2] use an approach that solves a subproblem parametrically by a shortest path approach that corresponds to the simplex method, except that steepest descent is not used. By contrast, we solve our subproblem by the out-of-kilter algorithm. In general, we expect that it requires more effort to solve our subproblem, but that the gain per iteration is greater. It appears that we take greater advantage of the previous solution (the section "Updating"). However, we do not know which algorithm is more efficient.

The work of RAO AND ZIONTS[10] appeared after our first draft was completed. Their publication contains related ideas, but does not preempt the present paper.

LAWLER[8] has proposed an approximate procedure based on binary search (bisection) of an interval $[a,b]$ for a zero of z (cf. the introduction), where $z(a)>0$ and $z(b)<0$. He takes advantage of the fact the search requires knowing only the sign of z at the points checked.

The discrete-time case (all $t_{ij}=1$) has been solved using a 'shortest route' approach (LAWLER[8], SHAPIRO[12]), which one suspects is best for this case.

No empirical comparisons are available, since our algorithm has not yet been programmed. Of course, any such comparisons would be biased by differences in the skills of the respective programmers. Perhaps most revealing would be least-squares polynomial fits to the running times as a function of the number of nodes in the network.

RELATED PROBLEMS

IN THE section on "The Subproblem" we pointed out that finding a cycle in a network with minimal cost-to-time ratio corresponds to deterministic single chain Markov renewal programming. An extension of the network formulation to deterministic multichain programs can be given along similar lines. The staircase-structured linear programming derived by Denardo and Fox[3] can be broken apart by decomposition so that the subproblem is exactly the network flow problem already given, except for its objective function. The master program, however, is no longer trivial. Even so, in large problems the network approach is probably more efficient than a direct attack using the simplex method on the original program.

In applications, the general multichain case seldom arises. Routing problems in which the 'links' can carry two-way traffic are single-chain problems, assuming that the terminals are connected.

NORMAN AND WHITE[9] have indicated heuristically that certain nondeterministic Markov programs can be solved approximately as deterministic programs using expectations. Our algorithm can be applied to the resulting program.

Another application of our algorithm occurs in scheduling parallel computations (REITER[11]).

One might think there is an analogous network formulation for nondeterministic Markov renewal programs, perhaps using networks with gains (JEWELL[7]), but the author has been unable to find a canonical network representation for such programs.

ACKNOWLEDGMENT

DISCUSSIONS with Eric Denardo led to a cleaner exposition.

REFERENCES

1. R. J. CLASEN, *The Numerical Solution of Network Problems Using the Out-of-Kilter Algorithm*, The Rand Corporation, RM-5456-PR (1968).
2. G. B. DANTZIG, W. BLATTNER, AND M. R. RAO, "Finding a Cycle in a Graph with Minimum Cost to Time Ratio with Applications to a Ship Routing Problem," pp. 77–83, *Theory of Graphs International Symposium*, Dunod, Paris, and Gordon and Breach, New York, 1966.
3. E. V. DENARDO AND B. L. FOX, "Multichain Markov Renewal Programs," *Siam J. Appl. Math* **16,** 468–487 (1968).
4. L. R. FORD AND D. R. FULKERSON, *Flows in Networks,* Princeton University Press, Princeton, New Jersey, 1966.
5. B. L. FOX, "Markov Renewal Programming by Linear Fractional Programming," *Siam J. Appl. Math.* **14,** 1418–1432 (1966).
6. W. S. JEWELL, "Markov-Renewal Programming, I and II," *Opns. Res.* **11,** 938–971 (1963).
7. ———, "Optimal Flow through Networks with Gains," *Opns. Res.* **10,** 476–499 (1962).
8. E. L. LAWLER, "Optimal Cycles in Doubly Weighted Directed Linear Graphs," pp. 209–213, *Theory of Graphs International Symposium*, Dunod, Paris, and Gordon and Breach, New York, 1966.
9. J. M. NORMAN AND D. J. WHITE, "A Method for Approximate Solutions to Stochastic Dynamic Programming Problems Using Expectations," *Opns. Res.* **16,** 296–306 (1968).
10. M. R. RAO AND S. ZIONTS, "Allocation of Transportation Units to Alternative Trips: A Column Generation Scheme with Out-of-Kilter Subproblems," *Opns. Res.* **16,** 52–63 (1968).
11. R. REITER, "Scheduling Parallel Computation," *J. ACM* **15,** 590–599 (1968).

12. J. F. Shapiro, "Shortest Route Methods for Finite State Space Dynamic Programming Problems," *SIAM J. Appl. Math* **16,** 1232–1250 (1968).
13. R. D. Wollmer, "Maximizing Flow Through a Network with Node and Arc Capacities," *Trans. Sci.* **2,** 213–232 (1968).

BINOMIAL DISTRIBUTION SAMPLE-SIZE NOMOGRAM

S. Matlin
General Electric Company, Valley Forge, Pennsylvania
(Received August 14, 1968)

This note gives a nomogram that provides a rapid approximation to the required sample size to be taken from a binomial distribution for a desired precision and confidence.

TO EMPLOY a Monte Carlo computer program, for example in calculating the kill probability (p) of an RV against a point target, the user is confronted with the problem of how many simulations or runs (n) to make. This sample size is a function of how accurately p is desired (within say $\pm\epsilon$) and with what confidence (γ) the conclusion can be drawn that the true lies p within $\pm\epsilon$ of the estimated $\hat{p}$. The attached sample-size nomogram provides a rapid approximation to the required n, given ϵ and γ.

USE OF THE NOMOGRAM

The nomogram is entered at the suspected value of p. One proceeds up to the desired value of ϵ, then across to the required confidence level γ, and finally down to the number of runs n. If p is expected to be less than 0.5, the same nomogram may be used by entering it at $(1-\hat{p})$ instead of at $\hat{p}$. If the user has no prior notion of how large p may be, then an upper limit to the sample size required may be found by assuming $\hat{p}=0.5$.

The interpretation of the confidence level is: if one takes samples of size n repeatedly from a binomially distributed population with parameter p, and computes the estimate $\hat{p}$ of p from each sample, then the intervals $\hat{p}_1\pm\epsilon$, $\hat{p}_2\pm\epsilon$, $\cdots$, $\hat{p}_k\pm\epsilon$ may be constructed, where $\hat{p}_k$ is computed from the kth repetition of taking a sample of size n; the fraction of these intervals that will contain the true p tends to γ as k increases.

DERIVATION

The nomogram was constructed by approximating the binomial distribution by a normal distribution having mean np and variance $p(1-p)/n$, which is a valid approximation when n is large ($np\geqq 9$). The relation employed is:

$$\text{Prob}\,\{\hat{p}-d_\gamma\sqrt{\hat{p}\,(1-\hat{p})/n}<p<\hat{p}+d_\gamma\sqrt{\hat{p}\,(1-\hat{p})/n}\}=\gamma,$$

Published in Operations Research on June 01, 1969 as DOI: 10.1287/opre.17.3.546.
This article has not been copyedited or formatted. The final version may differ from this version.