

SOME PROBLEMS IN DISCRETE OPTIMIZATION*,**

T.C. HU

University of Wisconsin, Madison, USA

Received 9 September 1970

Revised manuscript received 24 April 1971

The present paper concentrates on several problems of network flows and discrete optimization. Progress has been made on some of the problems while little is known about others. Some of the problems discussed are shortest paths, multi-commodity flows, traveling salesman problems, m -center problem, telepak problems and binary trees.

The present paper concentrates on several problems of network flows and discrete optimization. Progress has been made on some of the problems while little is known about others. Books and survey papers on networks and graphs are listed for general references. These books and papers form the first category of papers listed under [A1] to [A17].

The second category of papers is listed under B. *Minimum disconnecting set*. In proving the MAX-FLOW MIN-CUT theorem, Ford and Fulkerson [A5] developed a labeling method for finding a minimum cut separating the source from the sink in a network. The constructive proof picks out the minimum cut which is nearest to the source (i.e. the number of nodes connected to the source is a minimum when the arcs belonging to the minimum cut are deleted). If all the minimum cuts are needed, Plisch [B5] has a computer code to generate all the minimum cuts. The minimum cuts mentioned above were cuts separating the source and the sink. If we just want to disconnect the network by

* This paper was presented at the 7th Mathematical Programming Symposium 1970, The Hague, The Netherlands.

** Sponsored by the United States Army under Contract No.: DA-31-124-ARO-D-462 and the National Science Foundation, GJ-28339.

deleting the nodes and without first specifying the source and the sink, we can use the algorithm proposed by Kleitman [B4] for deciding if the minimum node disconnecting set is r or less. In other words, whether the deletion of r nodes can disconnect the network.

When there is more than one source-sink pair, each pair being for one special kind of flow, then we have multi-commodity flows in a network. A natural question is to ask for the minimum disconnecting set separating all the sources from their corresponding sinks in the network. The problem becomes a set-covering problem and has been treated by Bellmore and Ratliff [B1] [B2].

The labeling method [A5] finds the maximum flow in the network and picks out the minimum cut at the same time. The finiteness proof of the labeling method depends on the arc capacities being integers so that the flow value can be increased by at least one unit before the maximum flow value ν is reached. Thus, an upper bound on the number of labelings required would be ν , which is unknown at the beginning. When the arc capacities are real, the finiteness of the labelings has been obtained, by supplementing with some antizigzag rule (see for example [A10]), but no polynomial upper bounds on the number of labelings were obtained. Recently, Edmonds and Karp [B3] (reported in [A10]) proposed a slight modification of the labeling method and showed that the upper bound is $O(n^3)$ where n is the number of nodes in the network.

The third category of papers is listed under *C. Shortest paths and negative cycles*. This is an area in which people keep writing papers. According to the list compiled by Murchland [C8], there are more than 100 papers already, and the number is increasing rapidly. If the lengths of all arcs are positive, and the problem is to find the shortest paths from one node to all the other nodes, Dijkstra's method [C1] requires $O(n^2)$ operations, or $O(m)$ operations where n is the number of nodes and m is the number of arcs in the network. If shortest paths between all pairs of nodes are required (lengths can be negative as long as no negative cycle exists), the method based on Warshall [C9] and Floyd [C4] needs $2n^3$ operations. The method assumes no special structure of the network and works just as well for a large number of arcs as for a small number. A decomposition algorithm for larger networks with relatively small numbers of arcs was first proposed by Land and Stairs [C7]. The decomposition idea was successively modified by Hu [C5], Hu and Torres [C6], and Yen [C10]. Most papers on shortest paths or

negative cycles assume that the network is given as an $n \times n$ matrix initially and calculations are done in an "array" manner. If the network is given initially in a "list structure" then different types of algorithms would be more efficient.

Turning to the subject of *multi-commodity flows* (listed under D), we must first make the distinction between those problems which require "arc flows" to be integers and those problems which do not. Those problems without the integer requirement can be formulated as large linear programs. These large linear programs can be solved by "row generating" or "column generating" techniques similar to the decomposition principle of Dantzig and Wolfe. The first paper solving multi-commodity flows using a column generating technique is by Ford and Fulkerson [D1]. (In fact, the paper precedes the paper on the decomposition principle.) In that paper, a column generating subproblem becomes a shortest path problem. A more complicated problem in communication networks was solved by Gomory and Hu [D2]. Chapter 11 of Hu [A10] is devoted to multi-commodity flows using column or row generating techniques. Recently a game approach has been proposed by Grinold [D3].

When we require the arc flows to be integers, then intrinsically, these problems become integer programming problems. The unimodular property for single-commodity flows no longer holds. For two-commodity flows, Hu [D5] and Rothschild and Whinston [D9] have some results for undirected networks but no general results have been obtained on k -commodity flows for, say, $k > 4$.

The fifth category of papers is listed under *Traveling salesman problems*. This problem is usually cited as the problem of making a tour of n cities such that the total distance traveled is a minimum. The most common approaches to this problem are dynamic programming and branch and bound methods; see [E2]. Some of the algorithms are quite successful in solving 50-city problems. But even with computers of today's speed, no algorithm is currently able to solve a 200-city problem. The upper bound on the number of steps needed to solve an n -city problem is 2^n or some exponential function of n .

There have been some studies on the average number of steps needed to solve an n -city problem by the dynamic programming, cutting planes, branch and bound. The indication [E1] is that the average number of steps needed is $O(n^4)$. Although there is a very efficient computer code based on [E4], it is very unlikely that there will be an

algorithm with a polynomial upper bound. On the other hand, a traveling salesman's problem with special structure on the distance function may have a very efficient solution; see for example [E3]. When the cities are clustered into several groups, a decomposition approach may be used [E5].

The sixth problem may be called the *m-center problem*. One example of how this problem may occur is as follows. Assume that the population distribution of the State of Wisconsin is known. It is decided to build m theaters throughout the state such that the maximal number of people can enjoy the shows. A reasonable objective function would be to minimize the total distance travelled by all the people to the m theaters. Another objective function might be to minimize the maximal distance travelled by any one person. The plant-location or warehouse problem also belongs to this class. More abstractly, we will have a network with n nodes, each having a positive weight, and m centers are to be located in the network so as to minimize the total weighted distances between the centers and their assigned nodes. Various other choices of objective functions are possible, some choices making the problem more difficult than the others. The problem is usually trivial when $m = 1$. Many seemingly unrelated problems actually belong to this class. For example, let a continuous function of a single variable be defined on a closed interval. We may want to approximate this function by step functions. If we are allowed to use only five rectangles, what should be the heights and widths of each rectangle such that the maximum deviation between the function and the step functions is minimum?

Another variation of the problem may be to approximate a given step function (say n rectangles) by m rectangles ($m < n$). Very little is known about the *m-center problem*.

The seventh problem might be called the *telepak problem*. Let the distances between all the cities of U.S.A. be given, and suppose that the required numbers of telephone lines between every pair of cities are also known. To satisfy the requirement, we may build exactly the required number of telephone lines between the two cities, and do the same for all the requirements. However, the cost of building is not linear but concave. That is to say, a single telephone line costs one dollar per mile, but if we build one hundred lines between two cities, it costs 75 dollars per mile. If we build two hundred lines between two cities, it costs, say, 120 dollars per mile. Because the cost is concave, it

is cheaper to pack the lines into major routes than to build all telephone lines directly. To illustrate the idea of the problem, suppose that we have six cities A, B, C, D, E and F as shown in fig. 1. The numbers in fig. 1 are the mileages between the cities. Let there be 50 telephone lines required between city A and city C, and also between city B and city D. The total cost of building direct lines between A and C, B and D is

$$2 \times 20 \times 1 \times 50 = 2000 \text{ dollars.}$$

If we do not build direct lines between A and C but build the 50 lines by way of city B and city D, the total number of lines to be built between B and D is 100, and a cheaper rate is available. Thus the total cost is

$$2 \times 4 \times 1 \times 50 + 100 \times 0.75 \times 20 = 1900 \text{ dollars.}$$

If we do not build direct lines between A and C and between B and D, but build 50 lines between A and E, B and E, C and F, D and F, and also 100 lines between E and F, then the total cost is

$$4 \times 3 \times 1 \times 50 + 100 \times 0.75 \times 16 = 1800 \text{ dollars.}$$

The natural question is "what is the cheapest network configuration that will satisfy all the requirements??" This telepak problem has many practical applications and is also of interest from the theoretical point of view. Let the cost of building be c dollars per mile for a channel capable of handling any large number of telephone calls. Assume that we are not allowed to build through an intermediate city (such as E and F in fig. 1) unless there is a telephone line requirement starting or terminating in that city. Then the telepak problem

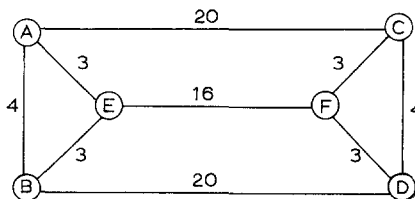


Fig. 1

becomes the standard problem of constructing a minimum spanning tree, for which a known efficient algorithm exists (see Kruskal [G5], Prim [G6]). If we are allowed to build through intermediate cities or even to create new junction points of telephone lines anywhere, then the telepak problem becomes the standard problem of Steiner's tree where no efficient general algorithm exists (the original Steiner's problem allows creation of junction points anywhere). The telepak problem has the flavor of the fixed-charge transportation problem but usually has thousands of cities involved. An algorithm which always gives the global optimum solution would necessarily have to examine all the data. This would be a formidable task. Here, I think the deterministic type of optimization method, such as the simplex method, may not be appropriate. Some approximate method which only examines local data should be applied (similar to the $Y-\Delta$ transformations [G1]). The challenge is to establish an upper bound on the gap between the optimum solution and the approximate solution.

The eighth problem might be called *erasing a graph*. Everyone knows that the PERT technique has been successfully applied to allocate a fixed budget among jobs of a project. The jobs are partially ordered due to technical restrictions (washing before drying, putting socks on before putting shoes on, etc.). More money allocated to a job gets that job done faster. Since the total budget of the project is fixed, the objective is to distribute the money optimally among the jobs such that the completion date of the project is earliest. The good thing about money is that if you do not use it now, you can use it later (no inflation, please!).

Let us now consider the optimum distribution of labor among jobs of a project. The jobs are partially ordered just as before. Now m workers are hired where any worker can do any of the jobs, every job needs a single worker, and every job takes one day for one worker. Under these extremely simplified assumptions, we can consider the jobs as nodes of an acyclic graph where the directed arcs indicate the order restrictions. Given the project represented by an acyclic graph, we can erase m nodes (jobs) at a time provided that we do not erase a node (job) before all its preceding nodes (jobs) are erased. What is the quickest way of erasing the graph?

The difference between labor and money can be seen in the following situation. At a certain stage of the project, there may be only s starting nodes in the graph ($m > s$). Then the unused labor ($m-s$) at

the time is essentially wasted (remember that a job can only take one man), while in the PERT model, an excess of money can always be applied to a single job or jobs to shorten their normal duration. There are some scattered results in this area (see [H1], [H2], [H3]), but no general results.

The ninth problem should be called *Binary trees*. Basically, we have a graph which is a tree. In fig. 2, we show a binary tree (called extended binary tree by Knuth [A12]). It is well known that in handling data in a computer, two kinds of information structures occur very frequently, namely, arrays and trees (or their generalized versions lists). Since all computers are intrinsically binary (including B.C.D. computers), it is no surprise that binary trees play an important role in handling information. Although there are many papers about trees in combinatorial mathematics, these papers are mostly about counting the number of trees in a class, not in searching for the optimum tree in a class. Since these problems usually come from the handling of information by a computer, the model is a very good representation of the real world. In operations research, binary trees give a new kind of optimum search problem unrelated to the search theory developed by B.O. Koopman and others.

For example, we may have information files about n people; these files are listed alphabetically according to the last names. What is the optimum way to search the files by the computer such that the average number of steps is a minimum? A particular search algorithm is equivalent to a binary tree where the circular node corresponds to a test and the square nodes correspond to the files. In general, we want an optimum binary tree under various definitions of optimality.

Several other topics related to networks and graphs are machine-job scheduling [J3], electrical networks, (see [J4], [J9]), telephone connecting networks (see [J2]), matching and related topics (see [J1], [J5]), or the construction of a graph with prescribed degrees of vertices

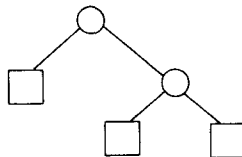


Fig. 2

(see [J6], [J7], [J8]). These are all listed under J and will not be reviewed here.

In recent years, a new branch of combinatorial mathematics has grown very rapidly. This new branch may be called discrete optimization, of which network flows and integer programming are subsets. Unlike the classical combinatorial mathematics which is concerned mostly with the existence or the number of configurations, discrete optimization searches for the optimal configuration among a finite number of configurations satisfying given constraints. Although the current development of discrete optimization is mostly towards algorithms, many deep and general theorems undoubtedly remain to be discovered in the future.

References

A. Books, bibliographies and survey papers

- [A1] C. Berge, *The theory of graphs*, Translated by A. Doig (Wiley, New York, 1962).
- [A2] C. Berge and A. Ghouila-Houri, *Programming, games and transportation networks*. translated by M. Merrington and C. Ramanujacharyula (Wiley, New York, 1965).
- [A3] R.G. Busacker and T.L. Saaty, *Finite graphs and networks* (McGraw-Hill, New York, 1965).
- [A4] N. Deo, *An extensive English language bibliography on graph theory and its applications*, Technical Report 32-1413, Jet Propulsion Laboratory (October 1969).
- [A5] L.R. Ford, Jr. and D.R. Fulkerson, *Flows in networks* (Princeton University Press, 1962).
- [A6] H. Frank and I.T. Frisch, *Communication, transmission and transportation networks* (Addison-Wesley, 1971).
- [A7] D.R. Fulkerson, "Flow networks and combinatorial operations research", *American Mathematical Monthly* 73 (2) (1966) 115-138.
- [A8] F. Harary, *Graph theory* (Addison-Wesley, New York, 1969).
- [A9] *Graph theory and its applications*, ed. B. Harris (Academic Press, New York, 1970).
- [A10] T.C. Hu, *Integer programming and network flows* (Addison-Wesley, New York, 1969).
- [A11] M. Iri, *Network flow, transportation and scheduling* (Academic Press, New York, 1969).
- [A12] D.E. Knuth, *The art of computer programming*, Vol. 1: *Fundamental algorithms* (Addison-Wesley, New York, 1968).
- [A13] J.W. Moon, *Topics on tournaments* (Holt, Rinehart and Winston, New York, 1968).
- [A14] O. Ore, *Theory of graphs* (American Math. Soc. Colloquium Publication, 1962).
- [A15] RAND "Research in combinatorics", a bibliography of selected RAND Publications, RAND Report Dept. (Jan. 1970).
- [A16] J. Turner and W. Kautz, "A survey of progress in graph theory in the Soviet Union", *SIAM Review* 12 (1970) (Supplement).
- [A17] J. Turner, "Key-word indexed bibliography of graph theory" In: *Proof techniques in graph theory*, ed. F. Harary (Academic Press, New York, 1969) pp. 189-330.

B. *Minimum disconnecting sets*

- [B1] M. Bellmore and H.D. Ratliff, "Set covering and involutory bases", The Johns Hopkins University, Report (1970).
- [B2] M. Bellmore and H.D. Ratliff, "Optimal defense of multi-commodity networks", to appear in *Management Science* (Applications).
- [B3] J. Edmonds and R.M. Karp, "A labeling method for maximal network flows which is bounded by a polynomial in the number of nodes", to appear. (Reported in T.C. Hu, *Integer programming and network flows*, pp. 161–170.)
- [B4] D.J. Kleitman, "Methods for investigating connectivity of larger graphs", *IEEE Transactions on Circuit Theory* (May, 1969).
- [B5] D.C. Plisch, "New results concerning separation theory of graphs", Ph.D. Thesis, E.E. Dept., University of Wisconsin, 1970 (under Professor D.P. Brown).

C. *Shortest paths and negative cycles*

- [C1] E.W. Dijkstra, "A note on two problems in connection with graphs", *Numerische Mathematik* 1 (1959) 265–271.
- [C2] S.E. Dreyfus, "An appraisal of some shortest path algorithms", *Operations Research* 17, No. 3 (1965).
- [C3] M. Florian and P. Robert, "A direct search method to locate negative cycles in a graph with application to minimum cost flows", Pub. # 14, Dept. of Information, University of Montreal, Canada.
- [C4] R.W. Floyd, "Algorithm 97: Shortest path", *Communication of ACM* 5 (6) (1962) 345.
- [C5] T.C. Hu, "A decomposition algorithm for shortest paths in a network". *Operations Research* 16 (1) (1968) 91–102.
- [C6] T.C. Hu and W.T. Torres, "Shortcut in the decomposition algorithm for shortest paths in a network", *IBM Journal of Research and Development* 13 (4) (1969) 387–390.
- [C7] A.H. Land and S.W. Stairs, "The extension of the cascade algorithm to larger graphs", *Management Science* 14 (L) (1967) pp. 29–33.
- [C8] J.D. Murchland, "Bibliography of the shortest route problem", LBS–TNT–62 (March, 1969); JDM–134 (July, 1970); Institut für Angewandte Reaktorphysik, Kernforschungszentrum Karlsruhe, Germany.
- [C9] S. Warshall, "A theorem on Boolean matrices", *Journal of ACM* 9 (11–12) (1962).
- [C10] J.Y. Yen, "A decomposition algorithm for finding all shortest distances in a series of linearly overlapping networks", to appear in *Operations Research*.

D. *Multi-commodity flows*

- [D1] L.R. Ford, Jr. and D.R. Fulkerson, "Suggested computation for maximal multi-commodity network flows", *Management Sciences* 5 (1) (1958) 97–101.
- [D2] R.E. Gomory and T.C. Hu, "Synthesis of a communication network", *Journal of SIAM* 12 (2) (1964) 348–369. (Also reported in T.C. Hu, *Integer programming and network flows*, pp. 197–213.)
- [D3] R.C. Grinold, "A multi-commodity max flow algorithm", *Operations Research* 16 (6) (1968) pp. 1235–1238.
- [D4] J.K. Hartman and L.S. Lasdon, "A generalized upper bounding algorithm for multi-commodity network flow problems", Tech. Memo No. 193, Case Western Reserve Univ. (June, 1970).
- [D5] T.C. Hu, "Multi-commodity network flows", *Operations Research* 11 (3) (1963) 344–360.
- [D6] W.S. Jewell, "Multi-commodity network solutions", ORC–66–23, Operation Res. Center, Univ. of California (Berkeley) (Sept., 1966).

- [D7] R. Saigal, "Multi-commodity flows in directed networks", ORC 67–38, Univ. of California (Berkeley) (Sept., 1967).
- [D8] M. Sakarovitch, "The multi-commodity maximal flow problem", ORC 66–25, Univ. of California (Berkeley) (1966).
- [D9] B. Rothschild and A. Whinston, "Feasibility of two commodity network flows" *Operations Research* 14 (6) (1966) 1121–1129.

E. Traveling salesman and traveling salesman problems

- [E1] M. Bellmore and J.C. Malone, "Pathology of traveling salesman algorithms", Technical Report, The Johns Hopkins University.
- [E2] M. Bellmore and G.L. Nemhauser, "The traveling salesman problem: A survey". *Operations Research* 16, No. 3 (1968) 538–558.
- [E3] P.C. Gilmore and R.E. Gomory, "Sequencing a one-state variable machine: A solvable case of the traveling salesman problem". *Operations Research* 12 (1964) 655–679.
- [E4] M. Held and R.M. Karp, "The traveling-salesman problem and minimum spanning trees", *Operations Research* to appear.
- [E5] T.C. Hu, "Decomposition of traveling salesman type problem", *Proc. IFORS*, Session A, Theory of Graphs (1966) pp. A32–A44.

F. *M*-center problem

- [F1] S.L. Hakimi, "Optimal distribution of switching centers and medians of a graph", *Operations Research* 12 (1964) 450–459.
- [F2] E. Minieka, "The *m*-center problem", *SIAM Review* 10, No. 1 (1970) 138–139.

G. Telepak problem

- [G1] J.B. Ackers, "The use of Wye-Delta transformation network simplification", *Operations Research* 8, No. 3 (1960) 311–323.
- [G2] A. Claus and D.J. Kleitman, "Heuristic methods for solving large scale network routing problems – The telepak problem" (Tufts Univ. and MIT).
- [G3] E.J. Cockayne, "On the efficiency of the algorithm for Steiner minimal trees", *SIAM Journal of Applied Mathematics* 18, No. 1 (1970) 150–159.
- [G4] E.N. Gilbert and H.O. Pollak, "Steiner minimal trees", *Journal SIAM* 16, No. 1 (1968) 1–29.
- [G5] J.B. Kruskal, Jr., "On the shortest spanning trees of a graph and the traveling salesman problem", *Proceedings American Mathematical Society* 7 (1956) 48–50.
- [G6] R.C. Prim, "Shortest connection networks and some generalizations", *Bell System Technical Journal* 36 (1957) 1389–1401.
- [G7] B. Rothfarb and M. Goldstein, "One terminal telepak problem", Office of Emergency Preparedness Report, *Operations Research* (to appear).
- [G8] T.C. Shah and R.D. Pederson, "Application of mathematical programming to optimal design of trunking networks", Collins Radio Co., Dallas, Texas.

H. Erasing a graph

- [H1] V.N. Burkov, "A problem in decomposing graphs" (see ref. B22 of [A15]).
- [H2] M. Fujii, T. Kasami and K. Ninomiya, "Optimal sequencing of two equivalent processors", *SIAM Journal of Applied Mathematics* 17 (4) (1969) 784–789.
- [H3] T.C. Hu, "Parallel sequencing and assembly line problems", *Operations Research* 9 (6) (1961) 841–848.

I. Binary trees

- [I1] M.R. Garey, "Optimal binary decision trees for diagnostic identification problems", Univ. of Wisconsin, Ph.D. Thesis in Computer Sciences, 1970.
- [I2] T.C. Hu and A.C. Tucker, "Optimal binary search trees", MRC TSR #1049, Math. Res. Center, Univ. of Wisconsin, Madison, 1970. *SIAM Journal of Applied Mathematics*, to appear.
- [I3] D.E. Knuth, "Optimal binary search trees", Computer Sciences Dept. Report 149, Stanford Univ., 1970.
- [I4] L.T. Reinwald and R.M. Soland, "Conversion of limited-entry decision tables to optimal computer programming" I: *Journal of ACM* 13 (1966) 339; II: 14 (1967) 742.

J. Scheduling, telephone connecting networks, electrical networks

Graphs with prescribed degrees

- [J1] M. Balinski, "Labelling to obtain a maximum matching", in: *Combinatorial mathematics and its applications*, eds. Bose and Dowling, (Univ. of North Carolina Press, 1970).
- [J2] V.E. Benes, *Mathematical theory of connecting networks and telephone traffic* (Academic Press, New York, 1965).
- [J3] R.W. Conway, W.L. Maxwell and L.W. Miller, *Theory of scheduling* (Addison-Wesley, New York, 1967). (See also *Naval Research Logistics Quarterly* 15, No. 2 (1968).)
- [J4] R.J. Duffin, "Network models", Dept. of Math., Carnegie-Mellon University Report 69-21, 1969. *Proceedings of the Symposium on Mathematical Aspects of Electrical Network Theory* (Am. Math. Soc., 1969).
- [J5] J. Edmonds, "Optimum branchings", *Mathematics of the Decision Sciences*, Part 1, Lectures in Applied Math., Vol. 11 eds. G.B. Dantzig and A.F. Veinott, Jr. (Am. Math. Society, Providence, R.I., 1968).
- [J6] S.L. Hakimi, "On the realization of a set of integers as degrees of the vertices of a linear graph", *SIAM Journal of Applied Mathematics* 11 (1963) 135-147.
- [J7] D.J. Kleitman, "Minimal number of multiple edges in realization of an incidence sequence without loops", *SIAM Journal of Applied Mathematics* 18, No. 1 (1970) 25-28.
- [J8] A.B. Owens, "On determining the minimum number of multiple edges for an incidence sequence", *SIAM Journal of Applied Mathematics* 18, No. 1 (1970) 238-240.
- [J9] P. Slepian, *Mathematical foundations of networks* (Springer-Verlag, Heidelberg, 1968).