International Journal of Pure & Applied Mathematical Research, Voleme1, issue1, Jan-Jun-2017, pp 36-41, ISSN: 2456-7493

HISTORY OF COMBINATORIAL OPTIMIZATION: STUDY OF AN APPLICATION BASED NETWORK FLOWS

Dr. Mohd. Rizwanullah[#1]

Associate Professor, Dept. of Mathematics and Statistics,
Manipal University Jaipur, Rajasthan, India.

***Abstract** - Combinatorics is a branch of pure mathematics concerning the study of discrete (and usually finite) objects. Combinatorial theory (or combinatorial analysis) is concerned with problems of enumeration and structure of mathematical objects. The objects may represent physical situation or things in applications or may be purely abstract and under study for theoretical reason. It is common practice to refer to the subject matter of combinatorial theory as combinatorics. The availability of reliable software, extremely fast and inexpensive hardware This paper highlights the historical development of combinatorial Optimization techniques specially focuses on network optimization techniques and then describes some very exciting future opportunities.*

Keywords: *Stochastic, Heuristics, Network flow, Max flow, Optimization, Combinatorics, Shortest Path, Extremal,*

Combinatorial Theory:

Combinatorial theory (or combinatorial analysis) is concerned with problems of enumeration and structure of mathematical objects. The objects may represent physical situation or things in applications or may be purely abstract and under study for theoretical reason. It is common practice to refer to the subject matter of combinatorial theory as combinatorics. Counting the number of objects of a certain type or the number of ways a particular operation can be carried out from the central problem of enumerative combinatorial theory. However, the structural features of the subject are of equal importance in theory and applications. These features are concerned with such things as how to specify when two objects (or process) are the same (i.e. definition of equivalence) and the existence of specific structure in a given class of objects (theory of structural synthesis).

Example Problems:

- Vehicle Routine Problem
- Traveling Salesman Problem (TSP)
- Minimum Spanning Tree Problem
- Integer Programming Problem
- Eight Queen Puzzle
- Knapsack Problem
- Cutting Stock Problem and so on.

Network Optimization:

The problems of network in which we investigate those values for the dimensions of the links and/or for the flows on the links which optimize the objective function (maximization or minimization), i.e. the technique which apply to the network problems to optimize the objective function under the given circumstances. For the solution of different situation/problems such as minimum cost, length, max. flow, profit etc. in networks and other like, the following family of network optimization techniques can be used:

1. Minimal Spanning Tree
2. Shortest route/path algorithm
3. Maximum flow algorithm
4. Minimum cost network flow algorithm
5. Critical Path algorithm (CPM)
6. Transportation & Assignment Problems
7. Dynamic Programming
8. Simulation Technique, etc.

1. Minimal Spanning Tree:

Spanning tree is a tree that includes every node of the graph. This technique can be used in the set of road, power/telephone supply cables, or gas pipe lines which connect a number of localities with the smallest total distance or cost.

2. Shortest Path Algorithm:

The shortest path algorithm is used to determine the shortest route between a source to destination in a network. This can be used in production planning, Knapsack problem, Businessman, Communication network, Replacement problem, Transportation problems, etc.

3. Maximum Flow Algorithm:

The idea of the maximum flow algorithm is to find a breakthrough path with net positive flow that links the source and sink nodes. This technique can be used in different problems such as: a water supply undertaking has a network of supply canals and pipes linking reservoirs, well and pumping stations to its customers, flow of traffic, of goods in a factory, of people and of telephone calls in a network of exchange etc.

4. Minimum Cost Network Flow Algorithm:

The technique can be used to determine the flows in the different arcs that minimize the total cost while satisfying the flows restriction on the arcs and the supply and demand amounts at the nodes. This technique can also be used in the set of nodes, power/telephone supply cables, or a gas pipeline, traveling salesman problems (Postman tour), etc.

5. Critical Path Algorithm:

Critical Path Method, abbreviated as CPM is also a network technique. CPM is not concerned with uncertain jobs as in PERT (Programme Evaluation and Review Technique). CPM is mostly used in construction projects, or in situation already handled, so that the details like the normal completion time, crash duration and cost of crashing are already known.

6. Assignment, Transportation, etc.

The Assignment problem is also an allocation problem. We have n jobs to perform with n persons and the problem is how to assign the jobs to the different persons involved.

7. Dynamic Programming:

The Dynamic programming is both a computer programming method and a mathematical optimization method, it refers to simplifying a complicated problem by breaking it down into simpler sub-problems in a *recursive* manner. While some decisions & problems cannot be taken apart this way, decisions that span several points of time do often break apart recursively.

8. Simulation Technique:

Simulation modelling can be used both as an analysis tool for predicting the effect of changes . Simulation is not an optimization procedure like linear programming. It allows us to make statement like, “your cost will be C if you take action X”, but it does not provide answer like, “Cost is minimized if you take action Y”.

Design theory:
Block designs are combinatorial designs of a special type. This area is one oldest parts of combinatorics, such as in Kirkman's schoolgirl problem proposed in 1850.

Order theory:

1. Shortest Path Problem:

The shortest path problems are the most fundamental and also the most commonly faced problems in the study of transportation and the communication networks. The shortest path problem arises when trying to determine the shortest, cheapest, or most reliable path between one or more than one pair of nodes in a network. The most importantly, algorithms for a wide variety of combinatorial optimization problem such as vehicle routine and network design often call for the solution of a large number of shortest path problem as subroutines consequently, designing and testing efficient algorithms for the shortest path problems has been a major area of research in network optimization. The shortest path problem and its generalizations have a voluminous research literature.

Label Setting Algorithm:

The first label setting algorithm was suggested by Dijkstra [1959], and independently by Dantzig [1960] and Whitney and Hiller [1960]. The original implementation of Dijkstra's algorithm runs in $O(n^2)$ time which is the optimal running time for fully dense networks (e.g. = Ω (n^2)), since any algorithm must examine every one. The following table summarizes various implementations of Dijkstra's algorithm designed to improve running on the worst-case or in practice. In the table, d = $[2 + \frac{m}{n}]$ represents the average degree of a node in the network plus 2.

#	Discoverers	Running Time
1.	Dijkstra [19591	$O(n^2)$
2.	Williams [1964]	O(m log n)
3.	Dial [1969]	(m + n C)
4.	Johnson [1977a]	$O(m \log_d^n)$
5.	Johnson [1977b]	O((m + n log C)log log C)
6.	Boas, Kaas and Zijlstra [1977]	O(nC + m log log nC)
7.	Denardo and Fox [1979]	O(mloglogC + nlogC)
8.	Johnson [1982]	O(m log log C)
9.	Fredman and Tarjan [1984]	O(m + n log n)
10.	Gabow [1985]	$O(m \log_d C)$
11.	Ahuja, Mehlhorn, Orlin (a) and Tarjan [19881	O(m + n log C) (b) $O(m + \frac{n \log C}{\log \log C})$ (c) $O(m + n\sqrt{\log C})$

Running times of the label setting algorithms

Glover, Klingman and Phillips [1985] proposed a new polynomially bounded label correcting algorithm, called the *partitioning shortest path (PSP) algorithm.* Ahuja and Orlin [1988] recently discovered a scaling variation of this approach that performs O(n21ogC) pivots and runs in O(nm.logC) time.

2. Maximum Flow Problem:

The maximum flow problem is distinguished by the long line of successive contributions the researchers have made in improving the worst-case complexity of algorithms; some, but not all, of these improvements have produced improvements in practice.

In the running times of the maximum flow algorithms which is shown below, n is the number of nodes, m is the number of arcs, and U is an upper bound on the integral arc capacities.

#	Discoverers	Running Time
1.	Edmonds and Karp 11972]	$O(nm^2)$
2.	Dinic [1970]	$O(n^2m)$
3.	Karzanov [1974]	$O(n^3)$
4.	Cherkasky [1977]	$O(n^2\sqrt{m})$
5.	Malhotra, Kumar and Maheshwari [1978]:	$O(n^3)$
6.	Tarjan [1984]	$O(n^3)$
7.	Bertsekas [1986]	$O(n^3)$
8.	Cheriyan and Maheshwari [1987]	$O(n^2\sqrt{m})$
9.	Ahuja and Orlin [1987]	$O(nm + n^2 \log U)$
10.	Ahuja, Orlin and Tarjan [1988]	(a) $O(nm + \frac{n^2 \log U}{\log \log U})$ (b) $O(nm + n^2\sqrt{\log U})$

(c) $O(nm \log(\frac{n\sqrt{\log U}}{m} + 2))$

Running times of the maximum flow algorithms.

3. Minimum Cost Flow Problem:

The minimum cost flow model is the most fundamental of all network flow problems. In this problem, we wish to determining a least cost shipment of a commodity through a network that will satisfy demands at certain nodes from available supplies at other nodes.

The minimum cost flow problem has a rich history. The classical transportation problem, a special case of the minimum cost flow problem . Dantzig [1951] developed the first complete solution procedure for the transportation problem by specializing his simplex algorithm for linear programming. Ford and Fulkerson [1956, 1957] suggested the first combinatorial algorithms for the un-capacitated and capacitated transportation problem; these algorithms are known as the primal-dual algorithms.

The Out-of-Kilter algorithm was independently discovered by Minty [1960] and Fulkerson [1961]. The negative cycle algorithm is credited to Klein [1967]. The recent *relaxation* algorithm by Bertsekas and Tseng (1988] is another interesting algorithm for solving the minimum cost flow problem. This algorithm maintains a pseudo flow satisfying the optimality conditions.

Polynomial Algorithms:

In the recent past, researchers have actively pursued the design of fast (weakly) polynomial and strongly polynomial algorithms for the minimum cost flow problem. An algorithm is strongly polynomial, if it's running time is polynomial in the number of nodes and arcs, and does not evolve terms containing logarithms of C or U. The table below reports running times for networks with n nodes and m arcs of which m' arcs are capacitated. The term S( ) is the running time for the shortest path problem to solve and the term M( ) has represented the corresponding running time to solve a maximum flow problem.

Polynomial Combinatorial Algorithms:

#	Discoverers	Running Time
1.	Edmonds and Karp [1972]	$O((n + m') \log U\, S(n, m, C))$
2.	Rock [1980]	$O((n + m') \log U\, S(n, m, C))$
3.	Rock [1980]	$O(n \log C\, M(n, m, U))$
4.	Bland and Jensen [1985]	$O(n \log C\, M(n, m, U))$
5.	Goldberg and Tarjan [1985]	$O(nm \log (n^2/m) \log nC)$
6.	Bertekas and Eckstein [1988]	$O(n^3 \log nC)$
7.	Gabow and Tarjan [1987]	$O(nm \log n \log U \log nC)$
8.	Goldberg and Tarjan [1988]	$O(nm \log n \log nC)$
9.	Ahuja, Goldberg, Orlin	$O(nm (\log U/\log \log U) \log nC)$
	and Tarjan [1988] and	$O(nm \log \log U \log nC)$

Strongly Polynomial Combinatorial Algorithms

#	Discoverers	Running Time
1.	Tardos [1985]	$O(m^4)$
2.	Orlin [1984]	$O((n + m')^2 S(n, m))$
3.	Fujishige [1986]	$O((n + m')^2 S(n, m))$
4.	Galil and Tardos [1986]	$O(n^2 \log n\, S(n, m))$
5.	Goldberg and Tarjan [1987]	$O(nm^2 \log n \log(n^2/m))$
6.	Goldberg and Tarjan [1988]	$O(nm^2 \log^2 n)$
7.	Orlin [1988]	$O((n + m') \log n\, S(n, m))$

Polynomial algorithms for the minimum cost flow problem

Assignment Problem:

The research community has developed several different algorithms for the assignment problem. Glover, Glover and Klingman [1986] is also a successive shortest path algorithm which integrates their threshold shortest path algorithm (see Glover, Sodini [1986] also suggested a similar threshold assignment algorithm. The relaxation approaches due to Dinic and Kronrod [1969], Hung and Rom [1980] and Engquist [1982] are also closely related to the successive shortest path approach.

Balinski [19851 developed the *signature method*, which is a dual simplex algorithm for the assignment problem. Balinski's algorithm performs $O(n^2)$ pivots and runs in $O(n^3)$ time. Goldfarb [1985] also describes some implementations of Balinski's algorithm that run in $O(n^3)$ time using simple data structures and in $O(nm + n^2 \log n)$ time using Fibonacci heaps.

Others Problems:

Several other problems related to the network optimization problem are of considerable theoretical and practical interests.

i) Multi-commodity flow problems:

For this class of problems, several commodities use the same underlying network, sharing common arc capacities. That is, the problem formulation contains "bundle constraints" that specify that the total flow on certain arcs cannot exceed the arc's capacity. The text by Kennington and Helgason [1980] describes the basic approaches to this problem, as do surveys by Assad [1978], Kennington [1978], and Ali et al. [1984].

(ii) Convex Cost Network Flow Problems:

One of the most natural extensions of the network flow models we have considered would be to replace the linear objective functions by more general convex cost functions. In some instances, for example when the cost function separates by arcs, that is $f(x) = \sum_{(i,j)\in A} f_{ij}(x_{ij})$ approximating each arc by a series of parallel arcs with linear costs, would permit us to use the techniques we have considered to solve these problems approximately to within any desired degree of accuracy. More elaborate algorithms are also possible. The general convex cost case requires solution techniques from nonlinear programming that are quire different that those I have described. As an overview to this literature, the reader might refer to the Kennington and Helgason [1980], Florian [1986] and the monograph by Rockafellar [1984].

iii) ***Network Design:***

The design problem in network itself is of considerable importance in practice and has generated an extensive literature of its own. Many design problems have been stated as fixed cost network flow problems: (some) arcs have an associated fixed cost which is incurred whenever the arc carries *any* flow. This class of problem requires solution techniques from integer programming and other type of solution methods from combinatorial optimization. Magnanti and Wong [1984].

References:

1. A. Campbell, *Aggregation for the probabilistic traveling salesman problem*, Comput Oper Res 33, 2703-2724, (2006).
2. Aarts, E. H. L. and Lenstra, J. K., Eds. 1997. *Local Search in Combinatorial Optimization*. Wiley, Chichester, UK
3. Bazaraa, M., and J.J. Jarvis. 1978. *Linear Programming and Network Flows*. John Wiley & Sons.
4. E.D. Anderson and Anderson, K.D. (1995) "*Pre-solving in linear programming*", Mathematical Programming 71 221-245.
5. F. Glover and M. Laguna (1992). "*Tabu Search," a chapter in Modern Heuristic Techniques for Combinatorial Optimization*.
6. Fleisher, L., Skutella, M., *The quickest multicommodity flow problem. Integer programming and combinatorial optimization*, Springer, Berlin, 2002, p. 36–53.
7. G.L. Nemhauser, L.A. Wolsey, *Integer and Combinatorial Optimization*, Wiley, New York, 1988.
8. Glover, F., and D. Klingman. 1976. *Network Applications in Industry and Government*. Technical Report, Center for Cybernetic Research, University of Texas, Austin, TX.
9. Golden, B., and T. L. Magnanti. 1977, *Deterministic Network Optimization*: A Bibliography. *Networks* 7, 149-183.
10. H. J. Greenberg 1998 "*An annotated bibiliography for post-solution analysis in mixed integer and*

combinatorial optimization" Advances in Computational and Stochastic Optimization, Logic Programming and Heuristic Search, ed. David L. Woodruff, Kluwer Academic Publishers, MA.

11. Lawler, E.L. 1976. *Combinatorial Optimization: Networks and Matroids.* Holt, Rinehart and Winston.
12. Lozovanu, D., Stratila, D., *The minimum-cost flow problem on dynamic networks and algorithm for its solving*. Bul. Acad. S¸ tiint¸ e Repub. Mold., Mat., vol. 3, 2001, p. 38–56.
13. Minieka, E. 1978. *Optimization Algorithms for Networks and Graphs.* Marcel Dekker, New York.
14. Papadimitriou, C.H., and K. Steiglitz. *1982. Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall.
15. R. Wong, *Vehicle routing for small package delivery and pickup services*, *The vehicle routing problem: Latest advances and challenges*, Bruce Golden, Raghu Raghavan, and Edward Wasil (Editors), Springer, August, (2007).
16. Smith, D. K. 1982. *Network Optimisation Practice: A Computational Guide.* John Wiley & Sons.
17. Zlochin, M., Birattari, M., Meuleau, N., and Dorigo, M. 2004. *Model-based search for combinatorial optimization: A critical survey*. *Ann. per. Res*. To appear.