БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ . BULGARIAN ACADEMY OF SCIENCES

ПРОБЛЕМИ НА ТЕХНИЧЕСКАТА КИБЕРНЕТИКА И РОБОТИКАТА, 50
PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS, 50

София . 2000 . Sofia

On the Complexity of an Algorithm for Optimization of the Network Flow with Inverse Linear Constraints

Atanas Atanassov

Institute of Information Technologies, 1113 Sofia

1. Introduction

Studying the complexity of an algorithm involves the answer of two basic computational questions – for the convergence and the amount of computing resources – time and/ or memory, needed for executing the algorithm. Three basic approaches are utilized for analyzing the performance of an algorithm - worst-case analysis, empirical analysis and average-case analysis [1, 2, 3]. The worst-case analysis provides a performance guarantee, while the empirical analysis gives an estimation of the algorithm behavior in practice. Optimization algorithms that exploit network structure are highly efficient. They can solve real-life network flow problems hundred times (by an order of magnitude) faster than the general algorithms of linear programming. In [5] a network flow, called a flow with inverse linear constraints (ILC-flow), has been introduced and investigated. The lower bounds of the arc flows are entirely replaced with linear inequalities on the arc flow functions. This flow owns reduced network properties. Searching for a feasible and optimal ILC-flows is not an easy task and such flows exist under certain conditions. An efficient approach for the solution of ILC-flow optimization problems is the approximation of these models by exploring the network characteristics. On this basis an iterative algorithm for an approximate solution of the ILC-flow optimization problems has been developed. Exact methods and algorithms for optimization of the standard network flow are applied. The paper focuses on the complexity of the approximate algorithm for optimization of the ILC-flow. The worst-case analysis is adopted.

2. Theoretical and algorithmic background

The optimization of ILC-flow includes solving two problems – Problem 1 for finding minimal ILC-flow and Problem 2 for finding ILC-flow with minimal cost

Problem 1 can be stated as:

$$\min v$$

subject to

(1) $$f(s, N) - f(N, t) = v;$$

(2) $$f(x, N) - f(N, x) = 0;$$

(3) $$\sum_{(x,y)\in U} b_i(x,y)\ f(x,y) \geq C_i,\ i\in I,$$

(4) $$f(x,y)\geq 0;\ (x,y)\in U,$$

where v is the value of the ILC-flow; N – the set of the nodes of the graph $G=\{N,U\}$, $|N|=n$; U – the set of its arcs, $|U|=m$; the nodes s and t – the source and the sink, the function f, $f{:}U\to R^+$ is the ILC-flow in G from s to t.

Problem 2 can be stated as:

$$\min \sum_{(x,y)\in U} a(x,y)\ f(x,y),$$

subject to (1)-(4), where $a(x,y)$, $(x,y)\in U$, are costs or objective arc estimations.

Consider a capacity constraint (3) of inequality type with an index i, $i\in I$, called an inverse linear constraint (ILC-constraint). Let us denote by D_i an arbitrary subset of U, such that for each $j, k\in I$, $D_i \cap D_k = \varnothing$, $\bigcup_{i\in I} D_i = U$, where $\varnothing$ is an empty set. Then in each ILC-constraint, the values of the flow on the arcs of the set D_i are in linear dependence. The coefficients $b_i(x,y)$ belong to the set of real non-zero numbers if $i\in I$ and $(x,y)\in D_i$, and are zeros, otherwise. The right-hand sides C_i of the ILC-constraints are real nonnegative numbers. They can be interpreted as "collective capacities" with respect to the arcs of the set D_i. The effectiveness of the algorithm for solving the stated optimization problems depends on the values of the coefficients $b_i(x,y)$ and on the structure of the set D_i.

The algorithm for solving Problem 1 and Problem 2 incorporates an iterative procedure of three main steps that can be generalized as follows:

A. The algorithm for optimization of ILC-flow is based on a constructive approach. In this approach an approximation of the ILC-flow via the standard flows, introduced by Ford and Fulkerson proceeds. The standard flow is defined by lower bounds of arc capacities, determined by appropriate relations and the standard network flow constraints. The following theorem guarantees that the approximating flow is an ILC-flow.

Theorem 1. Every non-zero realization of the standard network flow satisfies simultaneously the requirements (1)-(4) and is an ILC-flow at the same time.

In the conversion from ILC-flow into the standard flow the coefficients $b_i(x,y)$ play an important role for determining the lower bounds of the arc capacities. The objective arc estimations are used when seeking an ILC-flow with minimum cost. At solving Problem 2 and minimal value of the ILC-flow both $b_i(x,y)$ and $a(x,y)$ are utilized. The main idea is to direct the flow towards arcs with smaller coefficients.

The solution of the standard optimization problems – finding the maximum standard flow and finding standard flow with minimum cost needs the determination of the upper bounds of the arc capacities. In the general case in order to obtain a feasible flow and after that to solve the above problems sufficiently large real numbers can be applied to the upper bounds of the arc capacities. The disadvantage is that their "profile" may appear to be too far removed from the "profile" of the lower bounds of the arc capacities. This leads to increasing the iteration number and as a whole – the complexity of the algorithm. On the other hand if the two "profiles" are too close each other it is not guaranteed the feasibility of the approximating flow realization and therefore and the feasibility of the ILC-flow. Finding initial feasible standard flow is carried out by an iterative procedure that incorporates six steps. In the first step the initial upper bounds of the arc capacities of the standard flow in the original network are determined. Second – an extended network with all zero lower bounds is constructed. Third - the maximal flow in the extended network is found. The last two steps are based on the following theorem [4]:

Theorem 2. In the original network there exists a feasible flow if the maximum value of the flow in the extended network is equal to the integral lower bounds of the arc capacities.

If the condition of theorem 2 is fulfilled then the upper bounds of the arc capacities are determined by the respective lower bounds and the optimal flow realization in the extended network. Otherwise, the "profile" of the upper bounds of the arc capacities is augmented till the fulfillment of theorem 2 condition.

B. The standard approximating network flow is optimized. Highly efficient network flow algorithms for solving maximum flow problem and standard flow with minimal cost flow are applied. The obtained optimal realization of the standard flow is at the same time a realization of the ILC-flow but not obligatory the optimum one. The current cut is determined by a recurrent procedure. An index is applied to the current set. A number of quantities are determined.

C. There are stored the initial values of the standard flow realization, of the capacities and of some other quantities. A pathological case - the existence of a path from the source to the sink that contains arcs with only negative constraint coefficients, is eliminated by an appropriate procedure. The difference between the left hand side and right hand side of each constraint is determined. The validity of the necessary condition for minimality of the ILC-flow is checked.

Theorem 3. The necessary but not a sufficient condition for minimality of the ILC-flow is the existence of a cut, which separates the source and the sink and in which each forward arc and each backward arc with positive coefficients enter at least one saturated constraint.

"Rough'" and "fine" set-up of the arc capacity profile is accomplished if the necessary condition of theorem 3 does not hold. A transfer to step B for new iteration is performed after that. The iterative procedure ends if the necessary condition is satisfied. The approximate solution cannot be improved. The suboptimal ILC-flow is obtained.

In each iteration the algorithm for solving Problem 1 and Problem 2 controls and guarantees the fulfillment of the necessary condition for minimality of ILC-flow. This improves the approximate solution. By iterating the solution tends from above to the sought suboptimal solution. At each iteration exact highly efficient methods and algorithms of the standard network flow are applied.

3. Complexity analysis

The first generalized step of the algorithm for optimization of the ILC-flow approximates this flow by the standard network flow. A conversion is accomplished at which the initial profile of the lower bounds of the arc capacities is determined. A full network-tracing is needed. At that each arc is analyzed no more than once or in the worst case

- many times but the number of which is bounded by a constant. In practice this does not complicate the full network-tracing. In the developed algorithm two efficient methods for network analysis are realized - depth-first search (1) and breadth-first search (2). The difference is in maintaining the node list as a stack (1) or as a queue (2). The disadvantage of the first method, as a whole, is that when finding a path between each two node this path is not the shortest one. In the second method this disadvantage is avoided. Both methods examined each node twice – once at including the node into the stack or the queue, and second time when excluding the node from them – altogether $2n$ times. These examinations incorporate double analysis of each arc incident to a given node – altogether $2m$ times. Therefore the complexity of both methods is $O(n+m)$, which is practically the complexity of the first generalized step of the algorithm.

At the second generalized step of the developed algorithm the standard network, obtained in the previous step is optimized. The cornerstone of complexity analysis is the rational integration of highly efficient methods and algorithms for finding maximum approximating network flow and such flow with minimum cost into the proposed algorithm. The obtained realization is an ICL-flow, too. Each ILC-flow realization is called a pseudooptimal one due to the adoption of exact and fast network flow optimization algorithms. Finally a suboptimal ILC-flow is obtained. Fundamental problem of network flow programming is finding a maximum standard flow. All known algorithms for constructing a maximum flow are based on the iterative increment by the application of the augmenting path methods. The efficiency of each algorithm is caused by the proper selection of an augmenting path and by the opportunity for incrementing the flow at each iteration with maximum number of units. In the basic algorithm of Ford an Fulkerson the augmenting path is arbitrary selected and the flow is incremented by a unit. For finding each augmenting path n^2 operations or $O(n^2)$ time is needed. At the beginning the maximum value of the flow is unknown and in the worst case this algorithm is not bounded in the terms of n and m. As a whole the algorithm has a complexity of $O(Cm)$, where C is the integral arc capacity. Therefore the algorithm is pseudopolynomial. In the worst case this estimation may not be attained. The number of iterations is unbounded and the optimum solution is not achieved. If the shortest path in the network is selected for the augmentation then the algorithm becomes strongly polynomial with complexity $O(nm(n+m))$. When the flow is incremented simultaneously in more than one path the estimation is $O(n^2m)$. Further, if a node is eliminated instead of an arc at finding a blocking flow, and after reducing the number of iterations the complexity becomes $O(n^3)$. At the moment this is the best estimation for dense networks, The respective best estimation for sparse network is $O(nm \lg n)$. It is archived by using specific data structures called dynamic trees.

The second standard optimization problem for flow in networks is the one for finding a flow with minimum cost - mincost problem. Nevertheless that this problem is a firsthand generalization of the maximum flow problem the complexity analysis has more sophisticated character. For a long time strongly polynomial algorithm had not been developed and published. On the other hand, besides that for some pathological cases a number of pseudopolynomial algorithms have exponential complexity, their improvement and application are of real interest. An appropriate example is the standard negative cycle algorithm. Series of improvements concerning the identification of the negative cycle lead to obtaining optimal solutions for $O(A^*Cm\lg m)$, where A^* is the upper bound of the integral cost. For solving the mincost problem there are applied subroutines for finding the shortest path and maximum flow with complexity respectively $O(m+n\lg n)$ and $O(nm\lg(n^2/m))$. Complex dynamic tree structures of data are used. The most efficient pseudopolynomial algorithm for all networks except the very dense ones has a complexity $O(nm\lg\lg A^*\lg nC)$. The first strongly polynomial algorithm needs $O(m^4)$ time. At the time being the fastest strongly polynomial algorithm has a complexity of $O(m(\lg n)(m+n\lg n))$. This algorithm solves the mincost problem by a consequence of mlogn shortest path problems. It should be noted that the above results are only theoretical ones that must be yet proved by computational experiment.

In the second generalized step under study a number of substeps are executed. They are related to determining and storing basic values and current variables and to determining cuts by respective recurrent procedures. Let the number of ILC-constraints is no more than $m-1$, and each arc is analyzed at least once or several times but the number of examinations is bounded by a constant k^*. Then the complexity of all these operations does not exceed $O(k^*(n+m))$ or finally $O(n+m)$. If only the number of nodes is used in the estimation then it is obligatory to take care about the network size and density. Obviously the estimation $O(m^2)$ is preferred to $O(n^3m)$ at sparse networks and vise versa – at dense networks. In the present complexity analysis both parameters are used. Here it is of interest not the exact complexity of the proposed algorithm but only its asymptotic complexity. That is to say the asymptotic speed rise of the steps under the condition that the problem size – in case the number of nodes and arcs grows boundlessly. Therefore besides the numerous and various operations that are executed in this step, its complexity is defined by the time needed for running the maximum and mincost standard flow algorithms.

The third generalized step is basic in finding the suboptimal solution of ILC-flow problem. Here the fulfillment of the necessary condition for optimality of ILC-flow, according to theorem 3 is checked. This check needs $O(m)$ time if k is the number of ILC-constraints and in the worst case $k=m-1$. Depending on the checking result two separate corrections or set-ups of the profile of the approximating standard arc capacities proceed – "rough" set-up and "fine" set-up. In both set-ups depth-first search or width-first search is realized. Each arc is examined at least once. Besides there are updated the flow realization, the value of the flow and the cost function. As well an examination of at least one constraint or in the worst case - all constraints, is accomplished. The complexity of all these operations is $O(n+m+k)$. The number of ILC-constraints k is no more than $m-1$. The estimation is an asymptotic correlation. Therefore the overall complexity is $O(n+m+m-1)$ or finally $O(n+m)$. Let the necessary condition for optimality of the ILC-flow be valid. The end values of the obtained solution are determined for $O(n+m)$ time. Otherwise a transfer to the second generalized step of the algorithm for new iteration is performed. Each iteration approaches the intermediate solution to a finite value of ILC-flow – minimal or mincost ILC-flow. The number of iterations is bounded by n. These two circumstances make the developed algorithm finite and prove its convergence. Two basic prerequisites determine the insignificant influence of both iteration cycles on the overall complexity. The first one is that the number cycles for each "rough" and "fine" correction is bounded by constants. For each separate problem they are selected due to the preferable precision re every iteration and the total number if iterations. The smaller step corresponds to the greater number of iterations and vise versa. An important fact is that the number analyzed arcs reduces rapidly together with increasing the bounded number of iterations. The second prerequisite is that the number of the different maximum cuts is limited unlike the number of different flows. Nevertheless the iteration cycles increase the execution time needed for obtaining the suboptimal solution. In the proposed methodology for ILC-flow optimization each iteration is a single execution of a nested cycle at which the maximum or mincost approximating standard flow is found. Further, in the worst case each nested cycle is executed n times. Therefore the index of a power in the complexity of the highly efficient standard network flow algorithms increases no more than 1. For example, the complexity of the algorithm for finding minimal ILC-flow is between $O(n^3)$ and $O(n^4)$. As an approximation on the base of numerous computational experiments it may be accepted that the power in the complexity increases by 0.8, namely for the minimal ILC-flow algorithm it is not worse than O($n^{3.8}$).

4. Conclusions

This paper studies the computational complexity of an algorithm for the network flow with inverse linear constraints optimization. Two problems are solved by the proposed algorithm - finding the minimal ILC-flow and finding ILC-flow with minimum cost. The algorithm is based on the deduced necessary conditions for ILC-flow optimality and other theoretical inferences. An appropriate approximation of this flow by the standard network flows is realized. Exact methods and algorithms for optimization of the approximating flow are embedded. These algorithms are time determinative ones re the complexity of the proposed approximate algorithm. The approach of worst-case analysis for examining the performance of the algorithm is accomplished. The following main results are obtained:

1. The algorithm is convergent. Each iteration approaches the intermediate solution from above to a finite value of ILC-flow - minimal or mincost ILC-flow. The number of the iterations is bounded by the number of nodes n.

2. Each iteration is a single execution of a nested cycle. The complexity of the proposed algorithm is commensurate with the complexity of the standard network flow algorithms – the index of a power in this complexity increases no more than a unit.

References

1. Garey, M., D. Johnson. Computers and Intractability. A Guide to the Theory of NP-Completeness. San Francisco, W. Freeman and Co., 1979.
2. Ruhe, G. Algorithmic Aspects of Flows in Networks. Kluwer Academic Publishers, Dordrecht, 1991.
3. Papadimitriou, C. H., K. Steiglitz. Combinatorial Optimization: Algorithms and Complexity. N. J., Prentice-Hall Inc., Englewood Cliffs, 1982.
4. Evans, J., E. Minieka. Optimization Algorithms for Networks and Graphs. N. Y., Marcel Dekker, Inc., 1992.
5. Sgurev, V., A.T. Atanassov. An approximate method for optimization of a network flow with inverse linear constraints. - YUJOR, Vol.8, 1998, No 2, 289-313.

О вычислительной сложности приближенного метода для оптимизации сетевого потока с обратными линейными ограничениями

Атанас Атанасов

Институт информационных технологий, 1113 София

(Р е з ю м е)

В работе исследована вычислительная сложность приближенного алгоритма для оптимизации сетевого потока с обратными линейными ограничениями (ОЛО-потока). Анализирован интерактивный алгоритм приближенного решения задач определения минимального ОЛО-потока и такого же потока минимальной стоимости. Он основается на эффективной аппроксимации ОЛО-потока при помощи классического сетевого потока.

В основе предложенного анализа применяется подход определения времевой сложности алгоритма для "худшего случая". Времяопределяющими частями анализированного алгоритма являются точные методы и алгоритмы оптимизации классического сетевого потока. Анализом показана сходимость приближенного алгоритма и соизмеримость между его вычислительной сложностью и времевой оценкой классического сетевого потока. Этот результат происходит от ограниченного числа итераций и оттого, что каждая итерация представляет вложенный цикл и приближает сверху междинное решение к финальной величине ОЛО-потока.