

УДК 519.2

On Accuracy of Approximation for the Resource Constrained Shortest Path Problem

Aleksandr A. Soldatenko*

Institute of Mathematics and Computer Science
Siberian Federal University
Svobodny, 79, Krasnoyarsk, 660041
Russia

Received 14.02.2019, received in revised form 20.05.2019, accepted 20.08.2019

The paper we considers the Resource Constrained Shortest Path problem (RCSP). This problem is NP-hard extension of a well-known shortest path problem in the directed graph $G = (V, E)$. In the RCSP problem each arc e from E has a cost $w(e)$ and additional weight functions $r_i(e), i = 1, \dots, k$, which specifying its requirements from a finite set of resource. A polynomial time ϵ -approximation algorithm RevTree based on node labeling method is presented in the paper. The main advantage of the RevTree algorithm over existing ones is its ability to produce ϵ approximation of the RCSP problem in $\mathcal{O}(|V|^2)$ time. The present paper provides a proof of complexity and approximation of RevTree algorithm.

Keywords: combinatorial optimization, resource constrained shortest path, graph-based algorithm, efficient approximation algorithm.

DOI: 10.17516/1997-1397-2019-12-5-621-627.

Most modern telecommunication networks are multi-service because they provide users with various services: line telephony, cellular communications, cable television and data transmission. Quality of service (QoS) in multi-service networks defined by such parameters as throughput, transmission delay, jitter, cost and packet loss [1]. Any user's query for services implies determination of some route (desirable cost-effective) between two nodes based on the resource capability of a certain network. The problem of optimal service delivery in multi-service network can be formulated in mathematical term as a problem of finding resource constrained shortest path in a graph. This problem is called the Resource Constrained Shortest Path problem (RCSP) [2, 3].

There are two formulations of the RCSP problem: graph-theoretic and integer linear programming formulations. In graph-theoretic formulation multiservice network is represented by a directed graph where vertices indicate nodes of networks and arcs indicate channels of connection. Each arc is supposed to have a main weight and some additional weights which specify its requirements to move from a finite set of resource. It is necessary to find a feasible shortest path between two vertices, i.e. path that satisfies given resource constraints on resources needed to go through a given path. Having even one resource constraint, the RCSP problem considered to be NP-hard [4].

Three classes of methods to solve the RCSP problem are being pointed out: path ranking methods [4], node labeling methods [1, 2, 5] and methods that use the Lagrangian relaxation [6–8]. The first two classes of methods are based on the graph-theoretic formulation of the problem. Last class uses integer linear programming formulation. However, most of these methods work

*glinckon@gmail.com

slowly on networks of large dimension. Therefore, it is of great relevance to search for new approximation algorithms that can solve the RCSP problem quick on big networks [1, 8].

1. Resource Constrained Shortest Path problem and algorithms for its solution

The RCSP problem in graph-theoretic formulation is considered in accordance with the terminology which are commonly accepted in graph theory [9]. Let multi-service network be described by weighted oriented graph $G = (V, E)$ (below graph) without dual and self-loop edges, in which each vertex $v \in V$ specifies node of network and each arc $e \in E$ defines communication channel between corresponding network nodes, also $n = |V|$ and $m = |E|$. We assume that on the set of arcs of the graph $G = (V, E)$ we have a function $w(e): E \rightarrow \mathbb{R}^+$ which is assigned to each arc $e \in E$ weight $w(e) > 0$. Let P be the path in graph $G = (V, E)$ from vertex s to vertex d , where $s, d \in V$. We consider that weight of this (s, d) -path P is calculated as a sum of all its arcs i.e.

$$w(P) = \sum_{e \in P} w(e). \quad (1)$$

It means, that function $w(e)$ is additive. If $w(e)$ is the cost of travel along the arc e then $w(P)$ might be interpreted as the cost of passing (s, d) -path P in graph $G = (V, E)$.

For all arcs in $G = (V, E)$ except $w(e)$ we also have functions $r_i(e): E \rightarrow \mathbb{R}^+$, $i = 1, \dots, k$, which defines resource requirements for movement along arc e . We assume that for any (s, d) -path P the following is true:

$$r_i(P) = \sum_{e \in P} r_i(e), \quad i = 1, \dots, k. \quad (2)$$

We consider that the quantities $R_i \in \mathbb{R}^+$, $i = 1, \dots, k$, are given and they determine resource constraints for the multi-service network. Any (s, d) -path P is feasible if it satisfies resource constraints i.e.

$$r_i(P) \leq R_i, \quad i = 1, \dots, k. \quad (3)$$

It is required to find in $G = (V, E)$ the shortest path P which minimizes $w(P)$ in (1) and satisfies (3). Such path defines optimal solution of the RCSP problem and it is called resource constrained shortest (s, d) -path in graph G . Weight of this path is denoted by OPT .

Any feasible (s, d) -path may be considered as approximation solution of this problem. Obviously, if a set of feasible paths is empty then the RCSP problem does not have any solution. It may occur in case of tight resource constraints, for example, with $\min_{e \in E} r_i(e) > R_i$. For large values of this constraints, for instance, with $\sum_{e \in E} r_i(e) \leq R_i$, the RCSP problem degenerates in well-known problem of the shortest path (SP). It is clear that the shortest (s, d) -path in the graph $G = (V, E)$ is not always feasible. It is important to note that for fixed count of resources and non-negative of all weight functions $w(e), r_i(e)$ the RCSP problem is weakly NP problem [2, 5]. If we consider only one resource with $r(e) = 1$ for all arcs, then the RCSP problem is reduced to find the shortest (s, d) -path with less than R arcs. In this case, problem can be solved in polynomial time [4].

Nowadays, many approximation algorithms are being suggested for the RCSP problem, most of these using ideas of one or another pseudo-polynomial algorithms. Most of popular pseudopolynomial algorithms are based on sequential solution of the RCSP problem for various values of

path cost and resource constrains [2]. Their running time is $\mathcal{O}(mR)$ and $\mathcal{O}(mOPT)$. Many Polynomial Time Approximation Schemes has been developed based on these algorithms, including the Fully Polynomial Time Approximation Schemes (FPTAS), which consistently improve upper and lower bounds of solution [5, 10, 11]. For instance, in the study [11] the FPTAS has time complexity $\mathcal{O}\left(mn \log \log \log n + \frac{mn}{\epsilon}\right)$, where ϵ is a given approximation parameter.

The FPTAS' feature is dependency between a running time of algorithm and a value of ϵ , at the same time, time increases with increasing ϵ . When there is no such dependency and ϵ does not depend on size and parameters of a problem, then such approximation algorithms are called algorithms with guarantees approximation. Starting here and further in this paper saying the size of the RCSP problem we mean values n and m , which characterize number of vertices and arcs of graph $G = (V, E)$, and under parameters is values of functions $w(e), r_i(e)$. For the RCSP problem, we do not know any of algorithms with guarantees approximation. However, in many applications, the RCSP problems must be solved quickly with some accuracy, possibly depending on the parameters of the problem, but not affecting the solution time. We propose polynomial time ϵ -approximation algorithm (further algorithm RevTree) for the RCSP problem, which satisfices described requirements for a case with one resource R_i and $w(e), r_i(e)$ is positive real-valued functions. Next we are going to describe this algorithm, evaluate running time relative to the problem size and value of ϵ that depends on parameters of the RCSP problem. For the sake of brevity, we will write $r(e)$ and R instead $r_i(e)$ and R_i .

2. RevTree algorithm

The RevTree algorithm is based on a well-known Dijkstra's algorithm, which is used to solve the shortest path problem [9]. Dijkstra's algorithm uses a definition of neighborhood of graph vertex. The neighborhood of vertex $v \in V$ is set $\Gamma(v)$ of ends of arcs, which starts from v and have temporal labels. Set $\Gamma(v)$ defines possible directions to move along from vertex v in graph $G = (V, E)$. The RevTree algorithm decreases cardinality of the set $\Gamma(v)$ with considering resource constrain R , that allows to find a solution of the RCSP problem.

The RevTree algorithm consists of two phases and on both phases the Dijkstra's algorithm is performed. On the first phase algorithm finds minimum weight tree for function $r(e), e \in E$. The root of this tree is a target vertex d . Such tree defines for all vertices $u \in V$, also for $\Gamma(v)$, (u, d) -path. The weight of this path equals to minimal resource consumption when we move from u to d . Denote (u, d) -path by P_2 . Let P_1 be a path from start vertex s to current vertex v that is some solution of the RSCP problem for vertices s and v . According to the equation (2) this (s, v) -path consumes resource equal to $r(P_1)$. Then to move from vertex v to vertex $u \in \Gamma(v)$ it is necessary to satisfy the conditions

$$r(P_1) + r(v, u) + \pi(u) \leq R, \quad (4)$$

where $\pi(u) = \sum_{e \in P_2} r(e)$. The condition (4) guarantees that path (P_1, e, P_2) is feasible solution for the RCSP problem, where $e = (v, u) \in E$.

On the second phase the Dijkstra's algorithm with decreased neighborhoods is performed. A decrease of set $\Gamma(v)$ for all vertex $v \in V$ performed by the condition (4). If vertex $u \in \Gamma(v)$ does not satisfy the condition (4), it deleted from $\Gamma(v)$. The RevTree algorithm always finds feasible solution of the RCSP problem if it exist.

3. Complexity and approximation of the RevTree algorithm

Theorem 3.1 *The RevTree algorithm takes $\mathcal{O}(n^2)$ time and $\mathcal{O}(n)$ memory to solve the RCSP problem.*

Proof. It is known that the Dijkstra's algorithm takes $\mathcal{O}(n^2)$ time and $\mathcal{O}(n)$ memory [9]. On the first phase using the Dijkstra's algorithm we find minimum weight tree with root at vertex d . Saving of this tree takes $\mathcal{O}(n)$ memory. On the second phase we add to the Dijkstra's algorithm a check condition (4), does not affect on time complexity. Because both phases of the RevTree algorithm are performed consequently, then solution of the RCSP problem takes $\mathcal{O}(n^2)$ time and $\mathcal{O}(n)$ memory. \square

Now we obtain an estimate of the approximate solution found with the RevTree algorithm. Let's define function for each arc $e \in E$

$$\lambda(e) = \frac{r(e)}{w(e)}.$$

Calculate the minimal and the maximal values of this function on the set E :

$$\lambda_{min} = \min_{e \in E} \lambda(e) > 0,$$

$$\lambda_{max} = \max_{e \in E} \lambda(e) > 0.$$

Notice that the function $\lambda(e)$ is defined for any $e \in E$ because in the RCSP problem formulation we assume that $w(e) > 0, r(e) > 0, e \in E$.

Theorem 3.2 *The RevTree algorithm finds ϵ -approximation of a solution of the RCSP problem, if it exists for given vertices s and d , where $\epsilon = \frac{\lambda_{max}}{\lambda_{min}} - 1$.*

Proof. Let the RevTree algorithm found path P_1 on some step, which move from the start vertex s to the current vertex v that is optimal solution of the RSCP problem for vertices s and v . If vertex v coincides with target vertex d then algorithm found optimal solution of given problem. If vertex v coincides with s then path P_1 includes empty set of arcs and for it $w(P_1) = r(P_1) = 0$. In general case vertex v defines beginning of a part of (s, d) -path which should be found. Consider full neighborhood $\Gamma(v)$ of current vertex v as a set of ends of arcs which starts from v and has temporal labels. Let $x \in \Gamma(v)$ be a vertex for that $w(v, x) = \min_{u \in \Gamma(v)} w(v, u)$ but condition (4) is violated, i.e.

$$r(P_1) + r(v, x) + \pi(x) > R. \quad (5)$$

The Dijkstra's algorithm chooses that very vertex if neighborhood $\Gamma(v)$ is not decreased. The RevTree algorithm chooses vertex $u \in \Gamma(v)$ which satisfies the condition (4) and deviates from the shortest path in sense of (1) without resource constraints (3). Denote (v, d) -path found with full neighborhoods as P_{rest}^* and found with decreased neighborhoods as P_{rest} . For paths (P_1, P_{rest}) and (P_1, P_{rest}^*) follow inequalities are true

$$w(P_1) + w(P_{rest}^*) \leq OPT \leq w(P_1) + w(P_{rest}). \quad (6)$$

In (6) value $w(P_1) + w(P_{rest}^*)$ defines value of objective function for the shortest path without resource constrains and value $w(P_1) + w(P_{rest})$ is value of objective function for the path which has been found with the RevTree algorithm. By definition of the Theorem 2 approximated

solution for the RSCP problem are exist. Thus, we can find at least one feasible (s, d) -path which means always exist P_{rest} . Resource to move along any (v, d) -path is available and equal

$$R_{rest} = R - r(P_1).$$

Inequality $R_{rest} \geq \pi(v)$ is true consider (4). Using definition of function $\lambda(e)$ for all arcs $e \in E$ introduce following inequalities

$$0 < \lambda_{min}w(e) \leq r(e) \leq \lambda_{max}w(e).$$

Because functions $w(e)$ and $r(e)$ are additive then we get similar inequalities for any (v, d) -path include P_{rest}^* and P_{rest}

$$0 < \lambda_{min}w(P_{rest}^*) \leq r(P_{rest}^*) \leq \lambda_{max}w(P_{rest}^*), \quad (7)$$

$$0 < \lambda_{min}w(P_{rest}) \leq r(P_{rest}) \leq \lambda_{max}w(P_{rest}). \quad (8)$$

Path P_{rest}^* include vertex x and use (5) then following is true

$$r(P_{rest}^*) \geq r(v, x) + \pi(x) > R - r(P_1) = R_{rest}.$$

Here $\pi(x)$ is minimal required resource to move from vertex x to vertex d . Thus $r(P_{rest}^*) > R_{rest}$. Hence, with (7) we get

$$\frac{1}{\lambda_{max}}R_{rest} < w(P_{rest}^*). \quad (9)$$

From inequalities (8) follows the following

$$w(P_{rest}) \leq \frac{1}{\lambda_{min}}r(P_{rest}).$$

Because $w(P_{rest}^*) \leq w(P_{rest})$ and $r(P_{rest}) \leq R_{rest}$ then it is always true that

$$w(P_{rest}^*) \leq \frac{1}{\lambda_{min}}r(P_{rest}) \leq \frac{1}{\lambda_{min}}R_{rest}. \quad (10)$$

From (9) and (10) we get

$$\frac{1}{\lambda_{max}}R_{rest} < w(P_{rest}^*) \leq \frac{1}{\lambda_{min}}R_{rest}. \quad (11)$$

Now we perform an estimation of solution produced with the RevTree algorithm using inequalities (6). As a result

$$\begin{aligned} \frac{w(P_1) + w(P_{rest}) - OPT}{OPT} &\leq \frac{w(P_1) + w(P_{rest}) - (w(P_1) + w(P_{rest}^*))}{w(P_1) + w(P_{rest}^*)}, \\ \frac{w(P_1) + w(P_{rest}) - (w(P_1) + w(P_{rest}^*))}{w(P_1) + w(P_{rest}^*)} &= \frac{w(P_{rest}) - w(P_{rest}^*)}{w(P_1) + w(P_{rest}^*)}. \end{aligned}$$

Hence

$$\frac{w(P_1) + w(P_{rest}) - OPT}{OPT} \leq \frac{w(P_{rest}) - w(P_{rest}^*)}{w(P_1) + w(P_{rest}^*)}. \quad (12)$$

The greatest error between found and optimal solutions happens then $v = s$ and $w(P_{rest}) = \frac{1}{\lambda_{min}}R_{rest}$. Then inequality (12) takes form

$$\frac{\frac{1}{\lambda_{min}}R_{rest} - OPT}{OPT} \leq \frac{\frac{1}{\lambda_{min}}R_{rest} - w(P_{rest}^*)}{w(P_{rest}^*)}.$$

Use (11) get inequality $\frac{1}{\lambda_{max}}R_{rest} < w(P_{rest}^*)$. Hence we finally get

$$\frac{\frac{1}{\lambda_{min}}R_{rest} - OPT}{OPT} \leq \frac{\frac{1}{\lambda_{min}}R_{rest} - \frac{1}{\lambda_{max}}R_{rest}}{\frac{1}{\lambda_{max}}R_{rest}} = \frac{\lambda_{max}}{\lambda_{min}} - 1 = \epsilon.$$

Thus, feasible solution has been produced using the RevTree algorithm deviates from optimal solution of the RCSP problem not more then on value ϵ where $\epsilon = \frac{\lambda_{max}}{\lambda_{min}} - 1$. \square

Conditions $w(e) > 0, r(e) > 0, e \in E$ used in the RCSP problem formulation is needed only for proof of the Theorem 3.2.

Conclusion

In this paper we propose the polynomial time ϵ -approximation algorithm RevTree for RCSP problem for a case when there is only one resource. The algorithm always finds a feasible solution of problem if it exists. The algorithm takes $\mathcal{O}(n^2)$ time and produces a solution deviating from optimal not exceed $\epsilon = \frac{\lambda_{max}}{\lambda_{min}} - 1$. It is possible to extend the RevTree algorithm on cases with many resources. This extension has polynomial complexity, but additional research of solution approximation is needed.

References

- [1] P.Van Mieghem and et al., Quality of Service Routing, In Quality of Future Internet Services, LNCS, Vol. 2856, 2003, 80–117.
- [2] H.C.Joksch, The Shortest Route Problem with Constraints, *Mathematical Analysis and Application*, **14**(1966), 191–197.
- [3] M.Dror, Note on the complexity of the shortest path models for column generation in VRPTW, *Operation Research*, **42**(1994), 977–978.
- [4] M.R.Garey, D.S.Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, CA: W.H. Freeman, San Francisco, 1979.
- [5] I.Dumitrescu, N.Boland, Improved preprocessing, labeling and scaling algorithms for the weight-constrained shortest path problem, *Networks*, **42**(2003), 135–153.
- [6] K.Mehlhorn, M.Ziegelmann, Resource constrained shortest paths, In Algorithms ESA 2000, LNCS, Vol. 1879, 2000, 326–337.
- [7] M.Jepsen, B.Petersen, S.Spoorendonk, D.Pisinger, A branch-and-cut algorithm for the capacitated profitable tour problem, *Discrete Optimization*, **14**(2014), 78–96.
- [8] M.Horvath, T.Kis, Solving resource constrained shortest path problems with LP-based methods, *Computers*, **73**(2016), 150–164.
- [9] T.H.Cormen, C.E.Leiserson, R.L.Rivest, C.Stein, Introduction to Algorithms, 3rd edn. MIT Press, Boston, 2009.

- [10] R.Hassin. Approximation Schemes for the Restricted Shortest Path Problem, *Mathematics Oper. Res.*, **17**(1992), 36–42.
- [11] G.Xue, W.Zhang, J.Tang, K.Thulasiraman, Polynomial Time Approximation Algorithms for Multi-Constrained QoS Routing, *IEEE/ACM Transactions on Networking*, **16**(2008), no. 3, 656–669.

О точности аппроксимации для задачи о ресурсоограниченном кратчайшем пути

Александр А. Солдатенко

Институт математики и фундаментальной информатики

Сибирский федеральный университет

Свободный, 79, Красноярск, 660041

Россия

Рассматривается NP-трудная задача поиска ресурсоограниченного кратчайшего пути, известная под названием *Resource Constrained Shortest Path (RCSP)*. Задача RCSP является расширением задачи о кратчайшем пути в ориентированном графе $G = (V, E)$, когда для каждой дуги $e \in E$ кроме основной весовой функции $w(e)$ дополнительно задаются функции $r_i(e), i = 1, \dots, k$, численно отражающие потребности в ресурсах, которые необходимы для передвижения по этой дуге. Предлагается полиномиальный ϵ -приближенный алгоритм *RevTree*, использующий специальную маркировку вершин. Основное преимущество алгоритма *RevTree* по сравнению с существующими алгоритмами: он быстро находит допустимое решение задачи за $\mathcal{O}(|V|^2)$ время, отклоняясь от оптимального решения на величину, не превышающую ϵ . В работе приведены доказательства оценок сложности и точности алгоритма *RevTree*.

Ключевые слова: комбинаторная оптимизация, ресурсоограниченный кратчайший путь, алгоритмы на графах, эффективный алгоритм приближения.