ELSEVIER

Operations Research Letters 28 (2001) 213–219

operations research letters

www.elsevier.com/locate/dsw

A simple efficient approximation scheme for the restricted shortest path problem

Dean H. Lorenz[a,1], Danny Raz[b,*]

[a] *Department of Electrical Engineering, Technion, Israel Institute of Technology, Technion City, 3200 Haifa, Israel*
[b] *Bell Laboratories, Lucent Technologies, USA*

Received 16 December 1999; received in revised form 1 March 2001

Abstract

In this short paper we give a very simple fully polynomial approximation scheme for the restricted shortest path problem. The complexity of this ε-approximation scheme is $O(|E|n(\log\log n + 1/\varepsilon))$, which improves Hassin's original result (Math. Oper. Res. 17 (1) (1992) 36) by a factor of n. Furthermore, this complexity bound is valid for any graph, regardless of the cost values. This generalizes Hassin's results which apply only to acyclic graphs.

Our algorithm is based on Hassin's original result with two improvements. First we modify Hassin's result and achieve time complexity of $O(|E|n(\log\log(UB/LB) + 1/\varepsilon))$, where UB and LB are upper and lower bounds for the problem. This modified version can be applied to general graphs with any cost values. Then we combine it with our second contribution, which shows how to find an upper and a lower bound such that $UB/LB \leqslant n$, to obtain the claimed result.

1. Introduction

The restricted shortest path problem is a basic optimization problem that is both theoretically interesting and has practical applications (see for example [2]). In this problem the goal is to find the shortest path from a source node s to a target node t, among all paths that satisfy a certain criterion (a bounded delay for example, e.g. finding the minimal-cost T-path[2]). The problem is known to be $\mathcal{NP}$-complete and has a fully polynomial approximation scheme, FPAS [1].

One of the main difficulties in obtaining a fully polynomial approximation scheme for this problem is the lack of easily computable lower and upper bounds. For this reason some of the approximation schemes for this problem have time complexities which depend on the *values* of the cost (or length) of the edges.

* Corresponding author. Computer Science Department, Technion, Israel Institute of Technology, Technion City, 3200 Haifa, Israel. Tel.: +972-4-829-4938; fax: +972-4-822-1128.

E-mail addresses: deanh@tx.technion.ac.il (D.H. Lorenz), danny@cs.technion.ac.il, raz@research.bell-labs.com (D. Raz).

[1] Work done while visiting Bell-Labs, Lucent Technologies, and supported in part by DIMACS.

[2] A T-path is a path with delay smaller than or equal to T.

0167-6377/01/$ - see front matter
PII: S0167-6377(01)00069-4

Warburton [6] was the first to derive a polynomial approximation scheme for this problem on acyclic graphs. His result was improved by Hassin who gave a fully polynomial approximation scheme with time complexity of $O(|E|(n^2/\varepsilon)\log(n/\varepsilon))$ [1].

The main contribution of this paper is identifying a fairly simple and easy way to compute upper and lower bounds for this problem. The idea is to find a value c_j such that if we restrict our graph G to have only edges with cost c_j or smaller, it will still have a T-path, but if we delete all the edges with cost c_j then the resulting graph will no longer have a T-path. We then use c_j and nc_j as the lower and upper bounds on the optimal solution, and derive a FPAS with time complexity of $O(|E|(n/\varepsilon)\log\log n))$. We further improve this complexity by observing that all but the last iteration of this algorithm can work with $\varepsilon = 1$, thus the overall complexity is reduced to $O(|E|n(\log\log n + 1/\varepsilon))$. In fact we present a new test procedure which modifies the one used by Hassin. This procedure can be applied without any additional complexity increase to general graphs, even if some of the cost values associated with the edges equal zero.

Our techniques also apply to other approximation schemes for this problem. For example, Cynthia Phillips [4] provides a Polynomial Approximation Scheme,[3] which applies to general graphs, with time complexity of $O((|E|n/\varepsilon + (n^2/\varepsilon)\log(n^2/\varepsilon))\log\log UB/LB)$. Using our methods, it can be converted into a FPAS with time complexity of $O((|E|n + n^2\log n)\log\log n + |E|n/\varepsilon + (n^2/\varepsilon)\log(n^2/\varepsilon))$. This is slightly worse than what we achieve using the Hassin based scheme. On the other hand, Phillips' scheme, which uses Dijkstra's algorithm in an expanded network, is a more intuitive approximation (especially for graphs which contain cycles).

In the next section, we formally define the problem and summarize Hassin's results [1]. Then in Section 3 we describe our new approximation scheme and prove its time complexity.

2. Problem definition and Hassin's results

We use the definition from Hassin's paper [1]. The restricted shortest path problem is defined as follows.

Definition 1. Let $G = (V, E)$ be a graph with $|V| = n$ and $|E| = m$, such that each edge $e \in E$ is associated with a length (sometimes called cost) c_e and a transition time (sometimes called delay) d_e. Let T be a positive integer, and $s, t \in V$ be the source and target nodes. The restricted shortest path problem is to find a path $\boldsymbol{p}$ in G from s to t such that the transition time (delay) along this path is no greater than T, and the length (cost) of $\boldsymbol{p}$ is minimal.

The costs c_e and the delays d_e are assumed to be non-negative for all links. We denote the optimal cost by c^* and an optimal path by $\boldsymbol{p}^*$. As mentioned before this problem is $\mathcal{NP}$-complete but has a FPAS. Next we state the two main results from [1], which assume that the graph G is acyclic.

Theorem 1 (Hassin [1, Section 4])**.** *Given an instance of the restricted shortest path problem, upper and lower bounds on its optimal solution* (UB, and LB), *and an approximation factor* ε, *there exists an algorithm that finds an* ε*-approximated solution*[4] *with time complexity of*

$$O((mn/\varepsilon)\log\log(UB/LB)).$$

Using the sum of the $n-1$ longest edges as a trivial upper bound (denoted by B) and 1 as the lower bound, one gets an approximation scheme for the restricted shortest path problem with time complexity of $O((mn/\varepsilon)\log\log B)$. Trying to get an approximation scheme whose time complexity depends only on the graph

[3] Phillips' original paper, claims that it is a FPAS, but has an error that she acknowledges.

[4] A path with cost no greater than $(1+\varepsilon)c^*$.

size (m, and n) Hassin derived his second approximation scheme, that can be summarized by the following theorem.

Theorem 2 (Hassin [1, Section 5]). *Given an instance of the restricted shortest path problem, and an approximation factor ε, there exist an algorithm that finds an ε-approximated solution with time complexity of*

$$O(m(n^2/\varepsilon)\log(n/\varepsilon)).$$

Note that Hassin's algorithm is based on a pseudo-polynomial algorithm for this problem that has time complexity $O(mOPT)$, where OPT is the value of the optimal solution. The complexity bound of this algorithm is valid only for acyclic graphs. For general graphs that may have cost value of zero, the trivial bound is $O(nmOPT)$.

3. Main results

In this section we describe our new approximation scheme. First we observe that the time complexity of Theorem 1 can be improved to $O(mn(\log\log(UB/LB)+1/\varepsilon))$. Then we find an upper and a lower bound for the optimal solution such that the ratio between them is n. Combining these two results we get the claimed improved complexity. In order to explain how to improve Hassin's algorithm we have to get into more details.

Definition 2. Given an instance of the restricted shortest path problem, an approximation factor $0<\varepsilon$, and a value B, an *ε-test* is a procedure $T(\varepsilon,B)$ with the following properties:

- if $T(\varepsilon,B)$ answers YES then $OPT \geqslant B$, and
- if $T(\varepsilon,B)$ answers NO then $OPT \leqslant B(1+\varepsilon)$.

Warburton [6] proved that an ε-test procedure with time complexity of $O(mn/\varepsilon)$ exists for the restricted shortest path problem. Hassin then used this test procedure iteratively to get an upper and a lower bound with $UB/LB \leqslant 2$, using a binary search on $\log(UB/LB)$ [1]. Our observation is that this search can be carried out with a 1-test procedure, and will still generate in $O(\log\log(UB/LB))$ iterations an upper and a lower bound with $UB/LB \leqslant 2$. In order to make this paper self contained, and to prove that our complexity results hold for general graphs, we give a different proof based on a modified test procedure. The main idea is to scale the cost values, and then run the pseudo polynomial algorithm to find the smallest possible delay for each cost. The algorithm is presented in Fig. 1. We use $D(v,i)$ to indicate the minimum delay on a path from s to a node v, with cost bounded by the value i. The details of the Algorithm *SPPP* are important as they ensure that for all $l \in E$ we have $\tilde{c}_l \geqslant 1$, even if $c_l = 0$. We do require, however, that L and ε will be greater than 0.

Lemma 1. *Let $\boldsymbol{p}$ be any path, and $\tilde{c}$ defined in line 3 of Fig. 1, then the cost of $\boldsymbol{p}$ satisfies $c(\boldsymbol{p}) \leqslant \tilde{c}(\boldsymbol{p})S \leqslant c(\boldsymbol{p}) + nS$.*

Proof. For each $l \in E$ we have $c_l/S \leqslant \tilde{c}_l \leqslant c_l/S + 1$ hence $c_l \leqslant \tilde{c}_l S \leqslant c_l + S$ and

$$c(\boldsymbol{p}) = \sum_{l\in\boldsymbol{p}} c_l \leqslant S\sum_{l\in\boldsymbol{p}} \tilde{c}_l = \tilde{c}(\boldsymbol{p})S \leqslant c(\boldsymbol{p}) + nS. \qquad \square$$

Lemma 2. *Any path $\boldsymbol{p}$ returned by Algorithm SPPP satisfies*

$$c^* \leqslant c(\boldsymbol{p}) \leqslant U + (n+1)S = U + L\varepsilon.$$

Scaled Pseudo Polynomial Plus [SPPP] $(G(V,E),\{d_l,c_l\}_{l\in E},T,L,U,\epsilon)$

1. $S \leftarrow \frac{L\epsilon}{n+1}$
2. for each $l \in E$
3. define $\tilde{c}_l \equiv \lfloor c_l/S \rfloor + 1$
4. $\tilde{U} \leftarrow \lfloor U/S \rfloor + n + 1$
5. for all $v \neq s$
6. $D(v,0) \leftarrow \infty$
7. $D(s,0) \leftarrow 0$
8. for $i = 1,2,\ldots,\tilde{U}$
9. for $v \in V$
10. $D(v,i) \leftarrow D(v,i-1)$
11. for $l \in \{(u,v) \mid \tilde{c}_{(u,v)} \leq i\}$
12. $D(v,i) \leftarrow \min\{D(v,i), d_l + D(u,i-\tilde{c}_l)\}$
13. if $D(t,i) \leq T$
14. return the corresponding path and cost
15. return FAIL

Fig. 1. Algorithm Scaled Pseudo Polynomial Plus (SPPP).

Proof. By definition, $c^* \leqslant c(\boldsymbol{p})$. Since $\tilde{c}(\boldsymbol{p}) \leqslant \tilde{U}$ we have $\tilde{c}(\boldsymbol{p})S \leqslant \tilde{U}S \leqslant U + (n+1)S = U + L\varepsilon$. The result follows from Lemma 1. □

Lemma 3. *If $U \geqslant c^*$ then Algorithm SPPP returns a feasible path, $\boldsymbol{p}$, that satisfies $c(\boldsymbol{p}) \leqslant c^* + L\varepsilon$.*

Proof. For each $l \in \boldsymbol{p}^*$ we have $\tilde{c}_l \leqslant c_l/S + 1$. Thus,

$$\tilde{c}(\boldsymbol{p}^*) \equiv \sum_{l\in\boldsymbol{p}^*} \tilde{c}_l \leqslant c^*/S + |\boldsymbol{p}| \leqslant U/S + n \leqslant \tilde{U}$$

therefore $\boldsymbol{p}^*$ must be examined by the algorithm. By Lemma 1, $\tilde{c}(\boldsymbol{p}^*)S \leqslant c^* + nS \leqslant c^* + L\varepsilon$. Since $\boldsymbol{p}^*$ is examined by the algorithm, and since it finds an optimal solution, we must have $\tilde{c}(\boldsymbol{p}) \leqslant \tilde{c}(\boldsymbol{p}^*)$. Combining with Lemma 1 we get

$$c(\boldsymbol{p}) \leqslant \tilde{c}(\boldsymbol{p})S \leqslant \tilde{c}(\boldsymbol{p}^*)S \leqslant c^* + L\varepsilon. \qquad \square$$

Complexity. For each $1 \leqslant i \leqslant \tilde{U}$, each link is examined at most once, thus the overall complexity is

$$O(m\tilde{U}) = O\left(m\frac{n}{\varepsilon}\frac{U}{L} + nm\right).$$

If $U \geqslant L$, and $\varepsilon \leqslant 1$ then

$$O(m\tilde{U}) = O\left(m\frac{n}{\varepsilon}\frac{U}{L}\right).$$

We can now define an approximate test procedure.

Improved Hassin's algorithm [Hassin'] $(G(V,E),\{d_l,c_l\}_{l\in E},T,LB,UB,\epsilon)$

1. $B_L \leftarrow LB$
2. $B_U \leftarrow \lceil UB/2 \rceil$
3. while $(B_U/B_L > 2)$
4. $\quad B \leftarrow (B_L \cdot B_U)^{1/2}$;
5. $\quad$ if $T(1,B) =$ YES then $B_L \leftarrow B$
6. $\quad\quad$ else $(T(1,B) =$ NO$)$ $B_U \leftarrow B$;
7. return SPPP$(G(V,E),\{d_l,c_l\}_{l\in E},T,B_L,2B_U,\epsilon)$

Fig. 2. Improved Hassin's algorithm (Hassin').

Lemma 4. *The test*

$$T(1,B) = \begin{cases} \text{YES} & \textit{if SPPP}(G(V,E),\{d_l,c_l\}_{l\in E},T,B,B,1) \textit{ returns } \mathsf{FAIL} \\ \text{NO} & \textit{otherwise.} \end{cases}$$

is a 1*-test. (I.e., an* ε*-test with* $\varepsilon = 1$.)

Proof. We have to prove that if SPPP$(G(V,E),\{d_l,c_l\}_{l\in E},T,B,B,1)$ returns FAIL then $c^* > B$, otherwise $c^* \leqslant 2B$. The first part follows from Lemma 3 with $U = B$ and $\varepsilon = 1$, since if $c^* \leqslant B$, SPPP must return a feasible path. The second part follows from Lemma 2 since $L = U = B$ and any path returned by SPPP satisfies $c^* \leqslant c(\boldsymbol{p}) \leqslant U + (n+1)S = U + L\varepsilon = 2B$. □

Complexity. Calling SPPP with $U = L = B$ and $\varepsilon = 1$ requires $O(mn)$ steps.

We can now prove our first main result.

Theorem 3. *Given valid bounds* $0 < LB \leqslant c^* \leqslant UB$, *an* ε*-approximate solution can be found in* $O(mn/\varepsilon + mn \log\log \frac{UB}{LB})$.

Proof. Consider Algorithm Hassin' from Fig. 2. We first look for a lower bound B_L, and an *approximate* upper bound B_U such that $B_U/B_L \leqslant 2$ (the bounds are tight) and $2B_U$ is a valid upper bound (B_U approximates an upper bound within a factor of 2). We then use the bounds we found with Algorithm SPPP.

The bounds are found using a binary search on $\log B_L, \log B_U$ in the range $\log LB \ldots \log(UB/2)$, which terminates when $B_U/B_L \leqslant 2$. After each iteration the new value of $\log(B_U/B_L)$ is either $\log(B_L B_U)^{1/2}/B_L$, or $\log B_U/(B_L B_U)^{1/2}$. Since $\log(B_L B_U)^{1/2}/B_L = \log B_U/(B_L B_U)^{1/2} = \frac{1}{2}\log(B_U/B_L)$, the new value of $\log(B_U/B_L)$ is half the old value. Therefore, the binary search requires $O(\log\log(UB/LB))$ tests. Each test requires $O(mn)$ steps, thus, we can find the final values of B_L and B_U in $O(mn \log\log(UB/LB))$ steps.

By Lemma 4, $T(1,B)$ is a valid test, thus lines 5 and 6 in Fig. 2 ensure that at each iteration B_L is a valid lower bound, and $2B_U$ is a valid upper bound. We call SPPP (line 7) with valid bounds (i.e., $B_L \leqslant c^* \leqslant 2B_U$), therefore, by Lemma 3, $c(p) \leqslant c^* + B_L\varepsilon \leqslant c^*(1+\varepsilon)$. The call to SPPP requires $O(mn/\varepsilon)$ steps, since $U/L = 2B_U/B_L = O(1)$ in this case. □

Next we show how to find an upper and a lower bound for the optimal solution such that the ratio between them is n. We start by ordering the distinct values c_e. Let $c_1 < c_2 < \cdots < c_l$ be all the distinct length values of the edges. Clearly $l \leqslant |E| = m$. Define E_i to be the set of edges with length not greater than c_i, that is $E_i = \{e \in E \mid c_e \leqslant c_i\}$, for $1 \leqslant i \leqslant l$, and $E_0 = \emptyset$. Let $G_i = (V, E_i)$, then $G_l = G$, and $G_i \subset G_{i+1}$ for $0 \leqslant i \leqslant l-1$. Moreover, since G_l must have a T-path, (otherwise the instance has no solution) there must exist a unique

Simple Efficient Approximation [SEA] (G, T, ϵ)

1. $low \leftarrow 0; high \leftarrow l;$
2. while $low < high - 1$
3. $\quad j \leftarrow \lfloor (high + low)/2 \rfloor$
4. $\quad$ if $shortest_{s,t}path(G_j) < T$ then high$\leftarrow j;$
5. $\quad\quad$ else low$\leftarrow j;$
6. UB $= nc_{high}$; LB $\leftarrow c_{high};$
7. return (Hassin'(G,$\{d_l, c_l\}_{l \in E}$, T, LB, UB, ϵ))

Fig. 3. Algorithm simple efficient approximation.

index j $(1 \leqslant j \leqslant l)$ such that G_j has a T-path, and in G_{j-1} all paths from s to t have transition times larger than T.

Claim 1. *Let j be an index such that G_j has a T-path, and in G_{j-1} all paths from s to t have transition times larger than T, then*

$$c_j \leqslant c^* = OPT \leqslant nc_j.$$

Proof. Since G_j has a T-path, G must have a T path of length not greater than nc_j. On the other hand, since G_{j-1} has no T-path, all T-paths in G, including the path chosen by the optimal solution, must use at least one edge with length greater or equal to c_j. □

Thus the algorithm is: first find the "right" j, using binary search on the values $c_1, \ldots, c_l$, and then run the modified Hassin's algorithm with c_j, and nc_j as lower and upper bounds (see Fig. 3). Note that if $c_j = 0$ then any T-path in G_j is optimal, otherwise $L = c_j > 0$.

Theorem 4. *Algorithm SEA is a fully polynomial approximation scheme for the restricted shortest path problem with time complexity of*

$$O(mn(\log\log n + 1/\varepsilon)).$$

Proof. The fact that the algorithm is an approximation scheme follows from the observation that in line 6 we actually find upper and lower bounds for c^* (as proved by Claim 1) and from Theorem 3. The complexity of the second part of the algorithm (line 7 in Fig. 3) is $O(mn(\log\log n + 1/\varepsilon))$. For the first part, apart from having to sort the edges, we run the shortest path algorithm $O(\log m)$ times. Therefore, the complexity of the first part is $O(\log m)$ times the complexity of the shortest path algorithm which is $O(m \log n)$ (actually one can do it in $O(n \log n + m)$), but the dominant part remains Hassin's algorithm. □

4. Discussion

The main contribution of this paper is the introduction of new techniques that both improve the complexity, and enlarge the scope of the approximation scheme for the restricted shortest path problem. These techniques can be applied to other problems with similar characterizations. For example, as was already noted in [3], Hassin's techniques could be used to improve the approximation scheme for the capacitated plant allocation

problem. Another area for which these results can be applied is QoS partition and routing [2,5]. We are currently working on generalizations of our techniques which could be used to derive faster approximation algorithms for these problems.

References

[1] R. Hassin, Approximation schemes for the restricted shortest path problem, Math. Oper. Res. 17 (1) (1992) 36–42.
[2] D.H. Lorenz, A. Orda, QoS routing on networks with uncertain parameters, IEEE/ACM Trans. Networking 6 (6) (1998) 768–778.
[3] M. Labbè, E.F. Schmeichel, S.L. Hakimi, Errata and comments on "approximation algorithms for the capacitated plant allocation problem", Oper. Res. Lett. 18 (1996).
[4] C. Phillips, The network inhibition problem, Proceedings of the 25th Annual ACM Symposium on the Theory of Computing, May 1993, pp. 776–785.
[5] D. Raz, Y. Shavitt, Optimal partition of QoS requirements with discrete cost functions, IEEE J. Selected Areas Commun. 18 (12) (2000) 2593–2602.
[6] A. Warburton, Approximation of pareto optima in multiple-objective shortest path problems, Oper. Res. 35 (1987) 70–79.