Restricted Shortest Path Routing with Concave Costs

Pejman Khadivi, Shadrokh Samavi and Hossein Saidi
Department of Electrical and Computer Engineering
Isfahan University of Technology
Isfahan, IRAN
e-mail: khadivi@imap.cc

Abstract

Multi-constraint quality-of-service routing has become increasingly important as the Internet evolves to support real-time services. Restricted shortest path (RSP) is one of the important problems studied in the field of QoS routing. Various solutions have been proposed for RSP and other problems related to the QoS routing, in practical situations. Restricted shortest path problem with concave route costs are studied in this paper. This is a special version of the traditional RSP problem and is widely applicable in wireless and mobile ad hoc networks. In this paper, we propose new algorithms for this kind of routing. The effectiveness of our proposed solutions is shown through analytical reasoning and simulations.

1. Introduction

Routing is one of the most basic and widely studied problems in computer networking. While efficient algorithms have been proposed in the literature for *best effort routing*, *Quality of Service* (QoS) routing is an open problem. Selecting feasible paths that satisfy various requirements of the applications running over a network, is known as QoS routing [1][2][3].

Multi-Constraint Path (MCP) and *Restricted Shortest Path* (RSP) are the well-known problems, studied in the field of QoS routing. The MCP problem can be defined as a routing problem, trying to find a path that satisfies a number of constraints. Routes that satisfy the required constraints are named as *feasible* routes. In the RSP problem, it is required to find a feasible path that has a minimum cost among other feasible ones between a source node and a destination [2][3].

QoS routing is complex, and dealing with multiple QoS requirements makes this problem NP-Complete [4]. In the multi-constraint case, each network link has multiple weights which can be classified as additive, multiplicative or concave. For additive weights, the end-to-end weight of the path is the sum of the individual link values. Delay is an example of additive weights. A multiplicative path weight, such as path reliability, is the product of the link values along the path. Bandwidth belongs to the class of concave weights. The overall bandwidth of a path is equal to the minimum bandwidth of its links. It can be proven that optimum QoS routing with more than one constraint involving additive and/or multiplicative weights is an NP-Complete problem [4].

There are various methods for solving the MCP problem. *Depth-First Search* is an approach which is able to find a feasible path if one exists. However, its worst-case time complexity is exponential. A modified version of this approach is proposed in [5]. Even though the time complexity is reduced it is possible that the algorithm will not find a feasible path even if one exists [5].

In order to improve the performance and time complexity of the routing process, different single mixed metrics have been presented in the literature. Routing based on a mixed metric is not as effective as it may appear. When a single mixed metric is used for routing, part of the useful information is lost [3][6-8]. The TAMCRA algorithm presented in [7] uses a single metric and a *k-shortest path* algorithm in order to solve an MCP problem. This method reduces the performance shortcomings of using a mixed metric. The H_MCOP algorithm presented in [8] is used for solving a *Multi-Constraint Optimal Path* (MCOP) problem. The MCOP problem is a type of MCP (or RSP), which tries to find a feasible optimal path based on a cost associated with each link and path. In order to solve MCP problems, Khadivi et al. have proposed in [3] to take into account variations among different path weights in the routing procedure. Also, there are other methods, presented in the literature, which are based on the *Lagrange Relaxation* strategy [9][10]. An example is LARAC which is proposed in [10] for solving the RSP problems.

Yuan and Liu use a different definition of an optimal QoS path [11]. They present an extended version of the Bellman-Ford Algorithm to find all of the optimal QoS paths between a source and a destination. Then a feasible

1-4244-0212-3/06/$20.00/©2006 IEEE

path is selected if one exists. In [2] an efficient algorithm is proposed for solving the RSP problem. This algorithm is based on an iterative version of a strategy used for *all-hops k-shortest path* selection. In [12] all the metrics except one are changed to quantized integer values and then a polynomial time solution is presented for these new metrics. There are also methods that are based on *distributed routing* and *flooding* [13][14].

In this paper, a special version of the well-known RSP problem is considered which we name it as *RSP with concave route cost* (RSP-CC). In this problem, the cost of a path is the maximum of the cost of the links on that path. While it is not an NP-Complete problem, the RSP-CC is widely applicable in wireless ad hoc networks. For these networks, there could be a constraint on routes delay. Also, minimizing the distance between the consequent nodes on a single route is important in order to have long-life routes. Furthermore, distance minimization between the wireless transmitter and receiver, is important from the interference and power consumption point of views. Therefore, the distance between two neighboring nodes on a route can be considered as the link cost [15].

We propose two algorithms for solving the RSP-CC problems. The first one is an optimum polynomial-time solution which is based on the well-known Dijkstra shortest path algorithm. The optimality of this algorithm is proved analytically. The second solution is a modified version of the first one. This modified strategy results in sub-optimal routes with a lower time-complexity.

The remainder of the paper is organized as follows. In Section 2, a detailed definition of the problem is presented. In Section 3 our new solutions are proposed and evaluated analytically. Simulation results are presented in Section 4. Concluding remarks are given in Section 5.

2. Network Model and Problem Formulation

A network can be represented by a directed or undirected graph *G=(V, E)*, where *V* is the set of the nodes and *E* is the set of the links. In our routing problem, a path must be constructed between a specific *source* node and a *destination* node.

Definition 1: For a given network, *G=(V, E)*, and a certain source, *S*, and destination, *D*, $S, D \in V$, set π is the set of all the routes between *S* and *D*. Each route $p \in \pi$ is defined in the following manner:

$$p = (S = V_0, V_1, \ldots, V_n, V_{n+1} = D) \tag{1}$$

where, $V_i \in V$ is a neighbor of V_{i+1}, for $i = 0,1,\ldots,n$. Each node on the path should be traversed only once.

Definition 2: Consider a network topology, *G=(V, E)*. Also, assume that each link $(i, j) \in E$ has an additive non-negative cost, $c(i, j)$. Then, for a pair of source and destination nodes, S and D, the additive cost of a path $p \in \pi$, is $C_A(p)$ and is defined as follows:

$$C_A(p) = \sum_{(i,j)\in p} c(i, j) \tag{2}$$

The *Restricted Shortest Path* (RSP) problem is a routing problem and can be stated as follows.

Definition 3: Consider a network topology, *G=(V, E)*. Also, assume that each link $(i, j) \in E$ has an additive non-negative QoS weight, $w(i, j)$ and an additive non-negative cost, $c(i, j)$. Given a constraint, L, the RSP problem is to find a path $p \in \pi$ from a source node, *S*, to a destination node, *D*, with the minimum additive cost, such that,

$$w(p) \triangleq \sum_{(i,j)\in p} w(i, j) \le L \tag{3}$$

With additive path weights and costs, the RSP problem is an NP-Complete one. Therefore, different heuristics are proposed in the literature for this problem. While, additive path costs are conventional in routing problems, *concave path costs* are important in some cases such as routing in mobile ad-hoc networks.

Definition 4: Consider a network topology, *G=(V, E)*. Also, assume that each link $(i, j) \in E$ has a concave non-negative cost, $c(i, j)$. Then, for a pair of source and destination nodes, S and D, the concave cost of a path $p \in \pi$, is $C_C(p)$ and is defined as follows:

$$C_C(p) = \max_{(i,j)\in p} c(i, j) \tag{4}$$

The *Restricted Shortest Path with Concave Costs* (RSP-CC) problem is a routing problem and can be stated as follows:

Definition 5: Consider a network topology, *G=(V, E)*. Also, assume that each link $(i, j) \in E$ has an additive non-negative QoS weight, $w(i, j)$ and a concave non-negative cost, $c(i, j)$. Given a constraint, L, the RSP-CC problem is to find a path $p \in \pi$ from a source node, *S*, to a destination node, *D*, with the minimum concave cost, such that,

$$w(p) \triangleq \sum_{(i,j)\in p} w(i, j) \le L \tag{5}$$

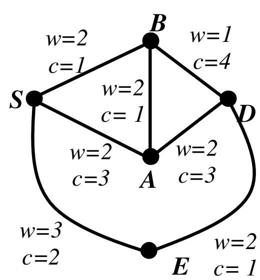
Definition 6: A path $p \in \pi$ in an RSP-CC problem is *optimally feasible* if and only if, it has the minimum concave cost among the paths which satisfy the constraint mentioned in (5).

While they have a high complexity, the RSP-CC problems are not NP-Complete. However, they are widely applicable in mobile ad hoc networks. These networks are constructed out of a number of *mobile stations* (MS's).

These stations are mobile and the single hop link between each pair of MS's, may be broken after a period of time. One of the main issues in ad hoc routing is to select long-life routes. Also, mobile nodes are battery powered and minimizing the power consumption, is another issue in ad hoc routing problems. Reducing the distance between consequent nodes on a route, results in long life routes which consumes less power [15][16].

An illustrative example of the RSP and RSP-CC problems is shown in Figure 1. Consider the network topology which is illustrated in Figure 1(a). Each link of this network has one additive weight and a cost which, in this example, could be interpreted as an additive or a concave cost. Five different paths exist between the source node, *S*, and the destination node, *D*. These paths are shown in Figure 1(b) with their corresponding weight and costs. The constraint is $L = 4.5$. Therefore, paths p_3, p_4 and p_5 are not the answer, because their weights are greater than *L*. The required constraint is satisfied by routing through the paths p_1 and p_2. However, the concave cost of p_1 is 3 which is less than the cost of p_2. Therefore, the answer of the RSP-CC problem is path p_1.

In the following section, routing algorithms are proposed which can find optimally feasible paths for the RSP-CC problems.



(a)

Path	$W(p)$	$C_A(p)$	$C_C(p)$
p_1:(S,A,D)	4	6	3
p_2:(S,B,D)	3	5	4
p_3:(S,A,B,D)	5	8	4
p_4:(S,B,A,D)	6	5	3
p_5:(S,E,D)	5	3	2

$L = 4.5$

(b)

RSP	RSP-CC
Result: p_2	Result: p_1
$W(p_2) = 3$	$W(p_1) = 4$
$C_A(p_2) = 5$	$C_C(p_1) = 3$

(c)

Figure 1. Example of RSP and RSP-CC problems. (a) A network topology (b) Existing paths and the required constraint (c) Optimally-feasible routes.

3. Pruning Based Routing

In this section, an optimum and *polynomial time* algorithm is proposed which may be used for solving the RSP-CC problem. This algorithm is based on the well-known Dijkstra shortest path and we name it as *Pruning Based Routing Algorithm*. Furthermore, a modified version of this algorithm is proposed which is a sub-optimal one with a lower time complexity.

3.1. Optimum Pruning Based Routing

In this section, an optimum routing algorithm is proposed in order to solve the RSP-CC problem. It is proved that the resulting paths are optimally-feasible.

The algorithm is illustrated in Figure 2. This is an iterative routing which works based on Dijkstra. At the beginning of each iteration, the Dijkstra algorithm is employed to find a shortest path between the source node, S, and the destination node, D. The shortest path is determined based on the additive weights. Therefore, if the result of Dijkstra is the path p^*, we have:

$$W(p^*) \leq W(p) \quad ; \quad \forall p \in \pi \tag{6}$$

where π is the set of the existing paths between the nodes S and D. If π is empty, it means that there is no path available between the nodes S and D and hence, $p^* = NULL$. If π is not empty and p^* satisfies the required constraint, L, the algorithm continues. In that case, the next step starts which is named as pruning.

In the pruning phase, the links of the network with the cost greater than or equal to the concave cost of the path, p^*, are removed. In other words, we have:

```
Algorithm Optimum-Pruning-Based-Routing;
  Input:G(V,E): network graph,
        S, D: source and destination nodes,
        L: constraint
  Output: Optimally-feasible path
Begin
SavedRoute ← NULL;
repeat
  Route ← Dijkstra (G,S,D);
  if (Route<>NULL) and
        (WEIGHT(Route)<=L) then
  begin
    SavedRoute ← Route;
    PathConcaveCost ← COST(Route);
    for all the links e∈ E do
      if COST(e)>=PathConcaveCost then
        G ← G - {e};
  end
until (Route=NULL)or(WEIGHT(Route)>L);
Return SavedRoute;
End.
```

Figure 2. Pseudo-code of the optimum-pruning-based routing algorithm.

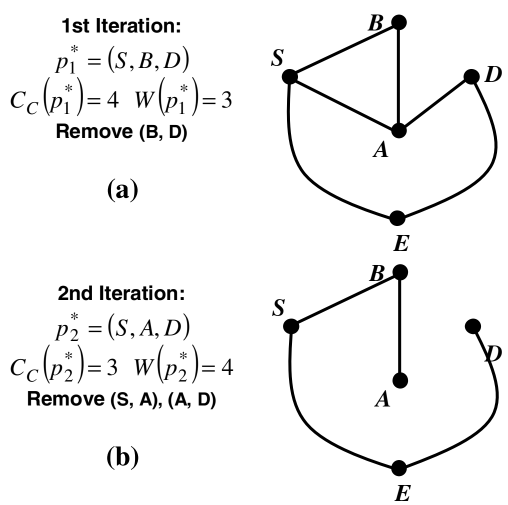


Figure 3. An illustrative example of using the algorithm of Figure 2 for solving the problem of Figure 1.

$$E = E - \{e\} \ \ iff \ \ c(e) \geq C_C\left(p^*\right) \tag{7}$$

where, E is set of the links in the network, e is a link, $c(e)$ is the cost of the link and $C_C\left(p^*\right)$ is the concave cost of p^* as defined by Definition 4. The routing process continues until there is no path available between the nodes S and D which satisfies the required constraint.

An illustrative example can be found in Figure 3. In this example, we are facing with an RSP-CC problem with $L = 4.5$. This problem is defined in Figure 1. In the first iteration, path $p_1^* = (S, B, D)$ is returned by the algorithm. The path weight is equal to 3, which satisfies the required constraint. Furthermore, the path's concave cost is equal to 4. Therefore, in the pruning phase, links with cost greater than 4 are removed from the network. This is shown in Figure 3(a). Figure 3(b) illustrates the second iteration of the algorithm. At the end of this iteration, links (S,A) and (A,D) are removed from the network. In the third iteration, the path which is returned by Dijkstra is $p_3^* = (S, E, D)$. This path is not a feasible path because its weight is 5, which is greater than the required constraint. Therefore, the algorithm stops and returns the last feasible path, $p_2^* = (S, A, D)$. In the following theorem, it is proved that this result is optimally-feasible.

Theorem 1: The path which is determined by the "*Optimum-Pruning-Based-Routing*" algorithm is optimally feasible.

Proof: Let us assume that the algorithm terminates after n iterations and the sequence of the paths which are generated by Dijkstra is p_1^*, p_2^*, … , p_n^*. Before iteration $j+1$ starts, all the links with the cost greater than or equal to $C_C\left(p_j^*\right)$ are removed from the network. Dijkstra returns the shortest path between the source and destination nodes. Also, if $j > i$, all the paths which exist in the j^{th} iteration have previously existed in the i^{th} iteration of the algorithm. Therefore, we have:

$$j > i \Rightarrow W\left(p_i^*\right) \leq W\left(p_j^*\right) \tag{8}$$

The algorithm terminates after a limited number of iterations, which is not greater than the number of the links. When $j > i$ then $C_C\left(p_j^*\right) < C_C\left(p_i^*\right)$. Otherwise, there must be a link $e \in p_j^*$ such that $c(e) \geq C_C\left(p_i^*\right)$. But, because $j > i$, all the links with the cost greater than or equal to $C_C\left(p_i^*\right)$ have been previously removed from the network. Hence, link $e \in p_j^*$, with the mentioned property, does not exist in the network when the j^{th} iteration is being started. Therefore, we have:

$$j > i \Rightarrow C_C\left(p_j^*\right) < C_C\left(p_i^*\right) \tag{9}$$

Hence, the following inequalities hold:

$$\begin{aligned} &W\left(p_1^*\right) \leq W\left(p_2^*\right) \leq \ldots \leq W\left(p_n^*\right) \\ &C_C\left(p_1^*\right) > C_C\left(p_2^*\right) > \ldots > C_C\left(p_n^*\right) \end{aligned} \tag{10}$$

The algorithm terminates either when $p_n^* = NULL$ or $W\left(p_n^*\right) > L$. Therefore, p_{n-1}^* is returned by the algorithm. Now, let us assume that there exists a path, such as q^*, with the following properties:

$$W\left(q^*\right) \leq L \ \ , \ \ C_C\left(q^*\right) < C_C\left(p_{n-1}^*\right) \tag{11}$$

During the routing process, path q^* was not generated by the algorithm. Therefore, q^* has a link, such as $e \in q^*$, which has been previously removed, let us say in the k^{th} iteration. Then, based on the algorithm's behavior $c(e) \geq C_C\left(p_k^*\right)$ and we have:

$$C_C\left(q^*\right) \geq C_C\left(p_k^*\right) \geq C_C\left(p_{n-1}^*\right) \tag{12}$$

Hence, the path q^* with the properties defined in (11) does not exist in the network and p_{n-1}^* is optimally feasible. ■

In the following sub-section a modified version of this algorithm is proposed which results in sub-optimal acceptable paths.

3.2. Modified Pruning Based Routing

While the *optimum-pruning-based-routing* algorithm, proposed in Section 3-1, returns optimally-feasible routes, it requires a long execution time to converge to the optimum result. In this section a modified *sub-optimal* routing strategy is proposed which may find an acceptable route in an acceptable execution time.

Let us consider the algorithm which is illustrated in Figure 2. The modified algorithm and its optimum version differ only in line "09". Let us assume that in a certain iteration of the algorithm, Dijkstra returns the path p^*. Then in the optimum algorithm, only the links with cost greater than $C_C\left(p^*\right)$ are removed during the pruning phase. In the modified version of the algorithm, links are pruned based on the following rule:

$$E = E - \{e\} \quad iff \quad c(e) \geq \varepsilon \times C_C\left(p^*\right) \tag{13}$$

where, ε is a constant coefficient, $0 \leq \varepsilon \leq 1$, and we name it as the *pruning factor*. The role of this coefficient is to reach a balance between the execution time of the routing algorithm and the resulting costs. It is clear that when $\varepsilon = 0.0$ the algorithm is equivalent with a single run of Dijkstra. Also, it is obvious that with $\varepsilon = 1.0$ the modified version of the algorithm, behaves exactly similar to the optimum one. The best value which may be used for the pruning factor depends on the network's topology as well as the links' weights and costs. In Section 4, the effect of this factor on the resulting routes is analyzed through simulations.

Let us assume that, for a certain routing problem, the modified version of the pruning based algorithm, terminates after *j+1* iterations and concave cost of the resulting path is equal to $C^{sub-opt}$. Also, assume that the concave cost of the path which is generated after the i^{th} iteration is $C_i^{sub-opt}$. After the i^{th} iteration, links with cost greater than $\varepsilon C_i^{sub-opt}$ are removed from the network. If instead of the modified version, the *optimum-pruning-based* algorithm was employed, the concave cost of the resulting route was equal to C^{opt}. Based on the behavior of these algorithms it is clear that:

$$\varepsilon C^{sub-opt} \leq C^{opt} \leq C^{sub-opt} \tag{14}$$

Employing the modified version of the algorithm, introduces an error in the cost of the resulting routes, which is defined by the following equation:

$$Sub-Optimality \quad Error \\ = \Delta_{sub-opt} = C^{sub-opt} - C^{opt} \tag{15}$$

An upper-bound may be easily found for $\Delta_{sub-opt}$ in the following manner. Based on (14) we have:

$$\Delta_{sub-opt} \leq C^{sub-opt} - \varepsilon C^{sub-opt} \tag{16}$$

Hence, the following inequality holds:

$$\Delta_{sub-opt} \leq (1-\varepsilon) C^{sub-opt} \tag{17}$$

Also, if (15) is multiplied by ε, we have:

$$\varepsilon \Delta_{sub-opt} = \varepsilon C^{sub-opt} - \varepsilon C^{opt} \tag{18}$$

Then, based on (14) the following inequality holds:

$$\Delta_{sub-opt} \leq \frac{(1-\varepsilon)}{\varepsilon} C^{opt} \tag{19}$$

In the following section, through simulations we analyze the behavior of the proposed algorithms.

4. Simulation Results

A large variety of simulation experiments have been performed using a wide range of different parameter settings. In this section some representative results are presented which illustrate the relative performance of the proposed algorithms. In the results to be presented, two performance parameters are measured for evaluation of the proposed strategies. Also, the performance of these solutions is compared with simple routing algorithms. The first parameter, to be measured, is the *Success Ratio* (SR), which is defined as the *percentage of time that the algorithm finds a feasible path*. The second measured parameter is the *average concave cost* of the generated routes.

It was mentioned in Sections 1 and 2 that the RSP-CC problem is important in the field of wireless and mobile communications. Therefore, simulations are performed in a way which is consistent with the behavior of wireless multi-hop networking. The simulation environment is constructed by a simulation area and a number of nodes. These nodes are randomly distributed over the simulation area based on a uniform distribution. In this paper, it is assumed that the simulations are performed over an 800×800 rectangular area and 420 nodes are employed for this purpose. Nodes *x* and *y* are neighbors if and only if, their distance is less than or equal to a *maximum distance*, $R_{\max}$. It is assumed that there is a link between nodes *x* and *y*, if and only if, they are neighbors. The concave cost of a link is equal to the distance between the corresponding nodes. Also, the additive weight of the links is equal to 1. Therefore, the paths' weights are measured in number of hops.

In each simulation case, the results are measured for 5000 random distributions of the nodes. For each distribution, 20 routing cases are examined. These routings are performed between 20 source nodes and a destination. The source and destination nodes are selected randomly based on uniform distribution. For each routing case, a constraint on the number of hops must be satisfied. This constraint is selected randomly from the range $[10, H_{\max}]$. Parameters $R_{\max}$ and $H_{\max}$ are simulation parameters.

For comparisons, three simple routing strategies are employed. In the following paragraphs these strategies are introduced. Suppose that paths p_1, p_2, …, p_k are available between the source node, S, and the destination node, D. Also assume that path p_i is formed in the following manner:

$$P_i = \left(S = V_{i,0}, V_{i,1}, \ldots, V_{i,m-1}, V_{i,m_i} = D\right) \tag{20}$$

where $V_{i,j}$ is the j^{th} node in p_i. Then, the following, metrics are defined:

Shortest-Longest-Link: In this case, we try to find a path from the source node to the destination with the shortest-longest-link. The length of the longest link in path p_i is named as D_i and we have:

$$D_i = \max_{j=1,\ldots m_i} length\left(V_{i,j-1}, V_{i,j}\right) \tag{21}$$

where, $length\left(V_{i,j-1}, V_{i,j}\right)$ is the distance between the nodes $V_{i,j-1}$ and $V_{i,j}$. Then, based on the shortest-longest-link strategy, the path p_S is selected when:

$$D_S = \min_{i=1,\ldots,k}\{D_i\} \tag{22}$$

Similar routing strategies have been previously used in [15] and [16].

Hop Count: With regard to (20), number of hops of the path p_i is equal to m_i. Therefore, in this strategy, the path p_S is selected when:

$$m_S = \min_{i=1,\ldots,k}\{m_i\} \tag{23}$$

Path Length: Assume that Γ_i is the length of the path p_i. Then, we have:

$$\Gamma_i = \sum_{j=1}^{m_i} length\left(V_{i,j-1}, V_{i,j}\right) \tag{24}$$

where, $length\left(V_{i,j-1}, V_{i,j}\right)$is the distance between the nodes $V_{i,j-1}$ and $V_{i,j}$. The path p_S is selected when:

$$\Gamma_S = \min_{i=1,\ldots,k}\{\Gamma_i\} \tag{25}$$

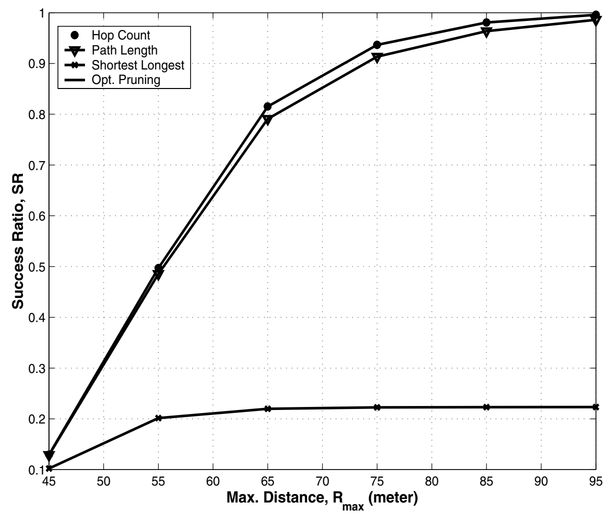


Figure 4. Success ratio when $R_{\max}$ is changing.

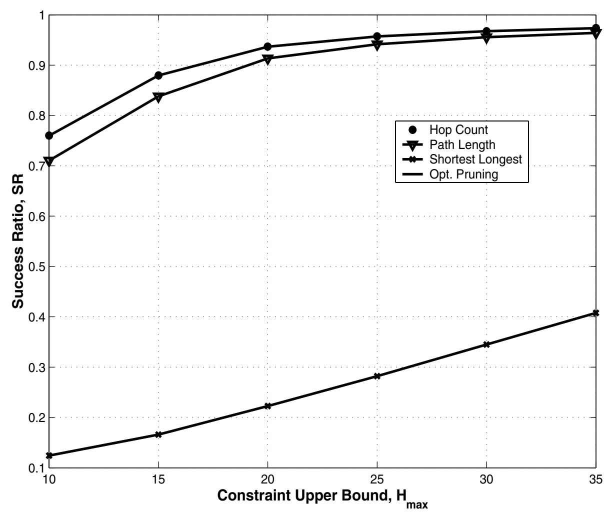


Figure 5. Success ratio when $H_{\max}$ is changing.

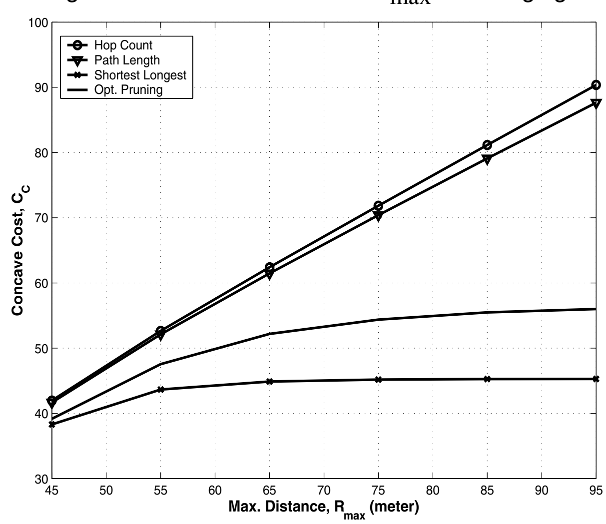


Figure 6. Average cost when $R_{\max}$ is changing.

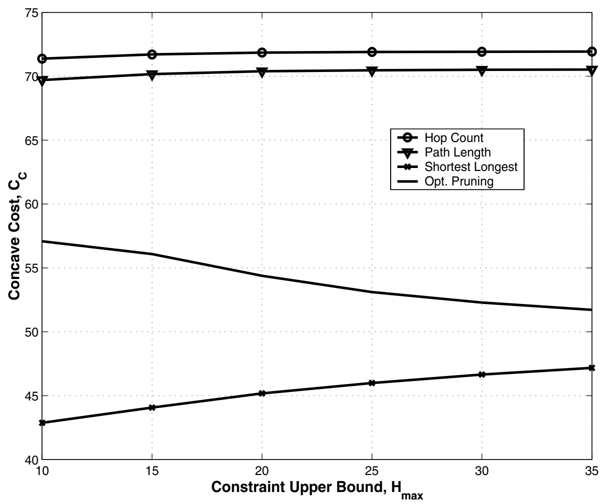


Figure 7. Average cost when $H_{\max}$ is changing.

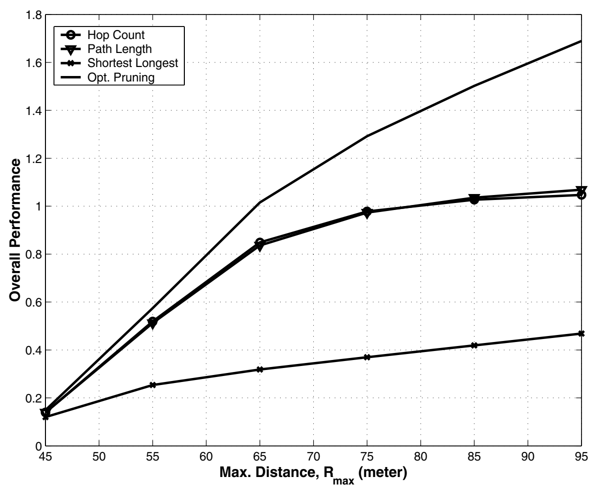


Figure 8. Overall performance when $R_{\max}$ is changing.

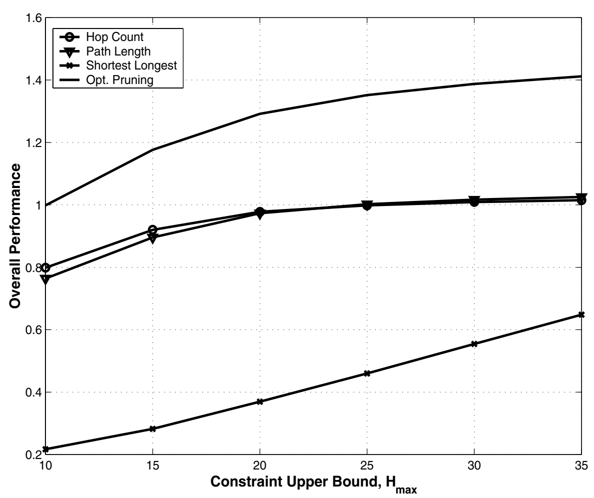


Figure 9. Overall performance when $H_{\max}$ is changing.

The routing algorithm is successful, if and only if, the required constraint is satisfied. In Figures 4 and 5 the *optimum-pruning-based* algorithm is compared with the routing strategies based on *hop-count*, *shortest-longest-link* and *path-length*, from the success ratio (SR) point of view. Figure 4, shows how the SR changes when $R_{\max}$ is changing. In Figure 5, it is illustrated that how the SR varies when $H_{\max}$ is changing. It is clear from these figures that the proposed solution has the same SR as the routing based on the hop-count metric. Also, it is obvious that routing based on the *Shortest-Longest-Link*, results in unacceptable success ratios. The reason is that, routing based on this metric, results in very long paths in terms of hop-count. Therefore, a large number of the generated routes are infeasible.

Different routing strategies are compared in Figures 6 and 7 with respect to the *average concave cost of the routes*. It is clear from these figures that routing based on the *shortest-longest-link* metric result in the lowest costs. However, the behavior of this strategy is not acceptable, because its SR is very low. The proposed *optimum-pruning-based* solution, results in acceptable SR and cost values.

In order to have a clear view of the performance of the proposed algorithms, let us define the following performance metric:

$$Overall \quad Performance = \left(\frac{Success \quad Ratio}{Average \quad Cost} \right) \times R_{\max} \qquad (26)$$

where, $R_{\max}$ is employed for normalizing the *average cost*. “Success Ratio” and “Average Concave Cost” are both taken into account when “*Overall Performance*” is used in the comparisons. A routing algorithm behaves well, when its success ratio is high and the average cost of the routes is low. Higher success ratios and lower average costs result in higher “overall performance”. The results are illustrated in Figures 8 and 9. It is clear from these figures that the proposed optimum-pruning-based strategy, results in the highest *overall-performances*.

Simulation results related to the modified version of the pruning-based algorithm are illustrated in Figures 10 and 11. Figure 10 shows how the average cost of the resulted routes change when the *pruning-factor*, ε, is varying between 0.5 and 1.0. Also the overall performance of the modified-pruning-based algorithm is compared with the performance of the optimum algorithm and the routing strategy based on the hop-count. It is clear that, as it is expected, when ε converges to 1.0, the modified-pruning algorithm converges to the optimum one. Also, it is obvious that when ε converges to 0.0, routes determined based on the modified algorithm converge to the routes generated based on the hop-count method.

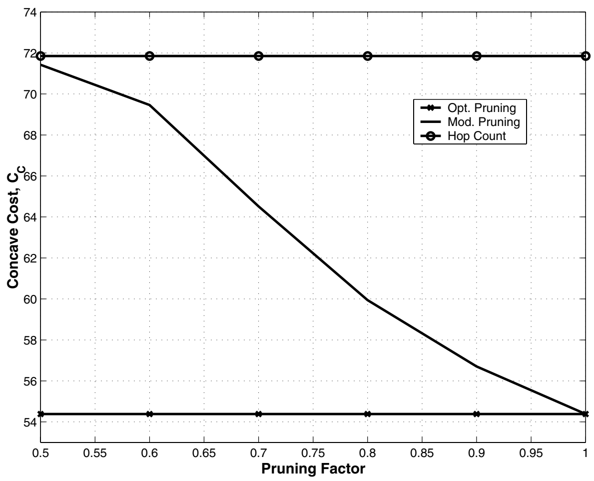


Figure 10. Average cost of the modified algorithm when $H_{\max}$ is changing.

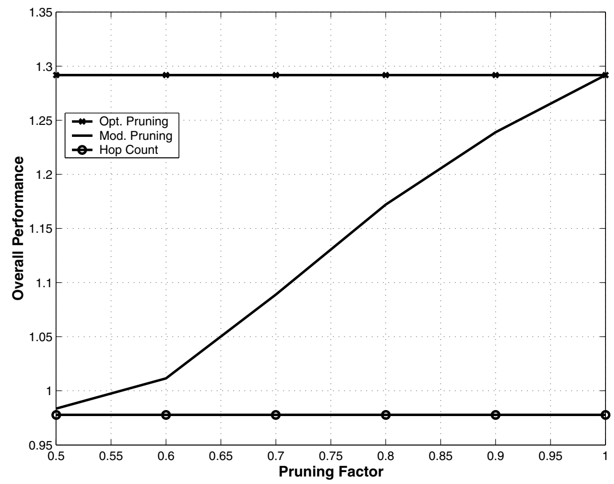


Figure 11. Overall performance of the modified algorithm when $H_{\max}$ is changing.

5. Conclusions

In this paper, a special version of the well-known RSP problem is considered which we name it as *RSP with concave route cost*. This problem is widely applicable in wireless ad hoc networks. We propose an optimum algorithm, named as *optimum-pruning-based-routing* for the RSP problems with concave route costs. We proved analytically that this algorithm results in optimally-feasible routes. Also, a modified version of the algorithm is proposed which results in sub-optimum results in shorter execution times. The effectiveness of the sub-optimal solution greatly depends on a constant coefficient which we named it as pruning factor. Choosing the best value for this coefficient depends on the network properties such as network topology and its weighting. Simulation results show the effectiveness of our proposed methods.

References

[1] Z. Wang, *Internet QoS: Architectures and Mechanisms for Quality of Service*, Morgan Kaufmann Publishers, 2001.

[2] G.Cheng, N.Ansari, "Achieving 100% Success Ratio in Finding the Delay Constraint Least Cost Path", *Proceedings of IEEE GlobeCom 2004*, Pp 1505-1509, 2004.

[3] P.Khadivi, S.Samavi, T.D.Todd, H.Saidi, "Multi-Constraint QoS Routing Using a New Single Mixed Metric", *Proc. of IEEE Int. Conf. on Communications (ICC 2004)*, Vol. 4 , France, Pp:2042 – 2046, 2004.

[4] Z. Wang, J. Crowcroft, "Quality-of-service routing for supporting multimedia applications", *IEEE Journal on Selected Areas in Communications*, Vol. 14, Pp. 1228 –1234, Sept. 1996.

[5] D.W. Shin, E.K.P.Chong, H.J.Siegel, "A multiconstraint QoS routing scheme using the depth-first search method with limited crankbacks", *Workshop on High Performance Switching and Routing*, Pp. 385 –389, 2001.

[6] J.M. Jaffe, "Algorithms for Finding Paths with Multiple Constraints", *Networks*, Vol. 14, Pp. 95-116, 1984.

[7] H. Neve, P. Mieghem, "A multiple quality of service routing algorithm for PNNI", *1998 IEEE ATM Workshop Proceedings*, Pp. 324 –328, 1998.

[8] T. Korkmaz, M. Krunz "Multi-constrained optimal path selection", *Proc. of INFOCOM 2001*, Vol.2, Pp. 834 –843, 2001.

[9] G. Feng, C.Douligeris, K. Makki, N. Pissinou, "Performance evaluation of delay-constrained least-cost QoS routing algorithms based on linear and nonlinear lagrange relaxation", *IEEE International Conference on Communications ICC 2002*, Vol.4, Pp. 2273 –2278, 2002.

[10] A. Juttner, B. Szviatovski, I. Mecs, Z. Rajko, "Lagrange relaxation based method for the QoS routing problem", *Proceedings INFOCOM 2001*, Vol. 2, Pp. 859 –868, 2001.

[11] X. Yuan, X. Liu, "Heuristic algorithms for multi-constrained quality of service routing", *Proceedings of INFOCOM 2001*, Vol. 2, Pp. 844 –853, 2001.

[12] S. Chen, K. Nahrstedt, "On finding multi-constrained paths", *IEEE International Conference on Communications ICC 98*, Vol. 2, Pp. 874 –879, 1998.

[13] J. Song, H. Pung, L. Jacob, "A multi-constrained distributed QoS routing algorithm", *IEEE International Conference on Networks ICON 2000*, Pp. 165 –171, 2000.

[14] A. Fei, M. Gerla, "Smart forwarding technique for routing with multiple QoS constraints", *IEEE Global Telecommunications Conference GLOBECOM '00*, Vol. 1, Pp. 599 –604, 2000.

[15] P.Khadivi, T.D.Todd, S.Samavi, H.Saidi, D.Zhao, "Mobile Ad Hoc Relaying in Hybrid WLAN/Cellular Systems for Dropping Probability Reduction", *Proceedings of the 9th CDMA International Conference (CiC 2004)*, Korea, October 25-28, 2004.

[16] V. Sreng, H. Yanikomeroglu, D.D. Falconer, "Relayer Selection Strategies in Cellular Networks with Peer-to-Peer Relaying", *Proceedings of the IEEE VTC'F03*, October 2003.