

# The Impact of Correlated Link Weights on QoS Routing

F.A. Kuipers and P. Van Mieghem

**Abstract**—Finding a path in a network based on multiple constraints (the MCP problem) is often referred to as QoS routing. QoS routing with constraints on multiple additive metrics has been proven to be NP-complete. This proof has dramatically influenced the research community, resulting into the common belief that exact QoS routing is intractable in practice. Hence, many heuristics for this problem were proposed, while hardly any exact algorithms. However, to our best knowledge, no one has ever examined which "worst-cases" cause NP-complete behavior. In fact, the MCP problem is not strong NP-complete, suggesting that in practice an exact QoS algorithm may work in polynomial time, making guaranteed QoS routing possible. The goal of this paper is to provide some properties and simulation results that indicate that NP-complete behavior hinges on a specific correlation structure between the link weights, which will be hardly ever encountered in practice.

## I. INTRODUCTION

There is an increasing demand for using real-time multimedia applications over the Internet. In order for these applications to work properly, Quality of Service (QoS) measures like bandwidth, delay, jitter and reliability, need to be controlled. Currently, the Internet cannot guarantee that the QoS requirements of applications will be satisfied. This has triggered the research community to (en masse) investigate the QoS problem, resulting in proposals for QoS-based frameworks (e.g. IntServ, DiffServ, constraint-based MPLS), QoS routing protocols (e.g. Q-OSPF, PNNI) and many QoS routing algorithms (mostly heuristics, see [13]).

Routing in general consists of two entities, namely the routing protocol and the routing algorithm. The routing protocol has the task of capturing the state of the network and its available network resources and disseminating this information throughout the network. The routing algorithm uses this information to compute shortest paths. Best-effort routing performs these tasks based on a single measure like hopcount or delay. QoS routing, however, must take into account multiple QoS requirements. Creating efficient QoS routing protocols is still an open issue that needs to be investigated further. In this paper, we assume that the network-state information is temporarily static, and that it has been distributed throughout the network and is accurately maintained at each node using QoS routing protocols. Once a node acquires the network-state information, it performs the second task in QoS routing, namely computing paths given multiple QoS constraints, also known as the multi-constrained path (MCP) problem. In this paper, we evaluate the complexity of exactly solving the MCP

problem. Before giving the formal definition of the MCP problem, let us first describe the notation that is used.

Let  $G(N, E)$  denote a network topology, where  $N$  is the set of nodes and  $E$  is the set of links. With a slight abuse of notation, we also use  $N$  and  $E$  to denote the number of nodes and the number of links, respectively. The number of QoS measures is denoted by  $m$ . Each link is characterized by a  $m$ -dimensional link weight vector, consisting of  $m$  non-negative QoS weights ( $w_i(u, v)$ ,  $i = 1, \dots, m$ ,  $(u, v) \in E$ ) as components. The QoS measure of a path can either be additive, multiplicative or min/max. In the case of additive measures (e.g. delay, jitter), the path weight of that measure equals the sum of the QoS weights of the links defining the path. Multiplicative measures (e.g. packet loss) can be transformed into additive weights by using the logarithm. Min(max) QoS measures of a path (e.g. available bandwidth and policy flags) refer to the minimum(maximum) of the QoS weights along the path. The QoS constraints of an application are expressed in the  $m$ -dimensional vector  $\vec{L}$ . Constraints on min(max) QoS measures can easily be treated by omitting all links (and possibly disconnected nodes) which do not satisfy the requested QoS constraint. We call this topology filtering. In contrast, constraints on additive QoS measures cause more difficulties. Without loss of generality, we assume all QoS measures to be additive.

*Definition 1: Multi-Constrained Path (MCP) problem.* Consider a network  $G(N, E)$ . Each link  $(u, v) \in E$  is specified by a link weight vector with as components  $m$  additive QoS weights  $w_i(u, v) \geq 0$ ,  $i = 1, \dots, m$ . Given  $m$  constraints  $L_i$ ,  $i = 1, \dots, m$ , the problem is to find a path  $P$  from a source node  $s$  to a destination node  $d$  such that

$$w_i(P) \stackrel{\text{def}}{=} \sum_{(u,v) \in P} w_i(u, v) \leq L_i \text{ for } i = 1, \dots, m$$

There may be multiple different paths in the graph  $G(N, E)$  that satisfy the constraints. Such paths are said to be feasible. According to Definition 1, any of these paths is a solution to the MCP problem. However, often it might be desirable to retrieve the optimal path, according to some criteria, within the constraints. This more difficult problem is known as the *Multi-Constrained Optimal Path* (MCOP) problem.

The rest of this paper is organized as follows. Section II presents an overview of related work. Section III will analyze the worst-case NP complexity of the MCP problem. The NP-completeness of the MCP problem strongly depends on the size of the link weights and the level of correlation between

those link weights. Section IV will analyze, mathematically and simulative, the impact of correlation on the complexity of solving the MCP problem. Section V will analyze the impact of the constraint values on the complexity. Finally, in Section VI, we will present our conclusions.

## II. RELATED WORK

The MCP problem is a NP-complete problem. Garey and Johnson [6] were the first to list the MCP problem with  $m = 2$  as being NP-complete, but they did not provide a proof. Wang and Crowcroft have provided this proof for  $m \geq 2$  in [23] and [24], which basically consisted of reducing the MCP problem for  $m = 2$  to an instance of the *partition* problem, a well-known NP-complete problem [6]. The effect of this proof has been tremendous, because it suggests that the MCP problem is intractable, in which case only heuristics can be used. Many simulations performed in [4], [5], [12], [20], [22] suggest that QoS routing may not be intractable in practice. There are certain NP-complete problems, such as *partition*, which are considered by many practitioners to be tractable. The reason for this is that, although no algorithms for solving them in time bounded by a polynomial in the length (e.g.  $N, E$ ) are known, there exist algorithms which solve those problems in time bounded by a polynomial in the input length (e.g. value of constraints) and the magnitude of the largest number (e.g. largest QoS weight) in the given problem instance [7]. Such algorithms are called pseudo-polynomial time algorithms. NP-complete problems for which no exact pseudo-polynomial time algorithm exists, are called NP-complete in the strong sense. In the case of the *partition* problem, the NP-completeness strongly depends on the fact that arbitrarily large numbers are allowed. If any upper bound were imposed on these numbers in advance, even a bound which is a polynomial function of the input length, there would exist a polynomial time algorithm for solving this (restricted) problem [7].

David Pisinger [19] has evaluated Knapsack problems, which are NP-complete problems (proved via reduction to the *partition* problem), and found that in practice these problems are tractable. For many more NP-complete problems, typical cases are easy to solve. A study of the phenomenon that typical cases are "easy", was performed by Cheeseman et al. [2], who introduced the concept of phase transitions in NP-complete problems. According to Cheeseman et al., NP-complete problems which are very under-constrained are soluble and it is usually easy to find one of the many solutions. NP-complete problems which are very over-constrained are insoluble. In the phase transition in between, problems are "critically constrained" and it is typically very hard to determine if they are soluble or insoluble [8]. Cheeseman et al. have conjectured that all NP-complete problems have at least one order parameter and that the hard to solve problems are around a critical value of this order parameter (phase transition). Although this conjecture does not hold for all NP-complete problems, there seems to be a connection between complexity and phase transitions. The lack of a phase transition seems to have significant computational implications: such problems

are either computationally tractable, or well-predicted by a single, trivial algorithm [10]. Note that the existence of a phase transition may also occur in problems that are not NP-complete. Monasson et al. [17], report an analytic solution and experimental investigation of the phase transition in K-satisfiability (the first problem shown to be NP-complete). Gent and Walsh [8] show that phase transitions occur in the *partition* problem.

Levin [14] advocated a different study of NP-complete problems by introducing the concept of average case complexity. He indicated that some NP-complete problems are "easy on average", while other (average case NP-complete) problems may not be.

There exists also some work in the literature revealing important properties of the MCP problem. We will mention three of those properties, that all strengthen our belief that in practice exact QoS routing is possible. First of all, the MCP problem is not strong NP-complete, because there exist pseudo-polynomial algorithms that exactly solve this problem (e.g., see [11], [15]). Secondly, if all, but one, measures take bounded integer values, then the MCP problem is solvable in polynomial time [3]. Finally, if some specific dependencies exist between QoS measures, exact QoS routing can be performed in polynomial time [16]. The goal of our work is to provide some more properties that suggest that exact QoS routing, in practice, is tractable.

## III. WORST-CASE COMPLEXITY ANALYSIS

In this section we will analyze the worst-case complexity of the MCP problem for  $m = 2$ . First, we will rewrite the proof that the MCP problem for  $m = 2$  is NP-complete [23], [24], and refer to it as the NP-proof:

*Proof:* Given a chain topology with  $n + 1$  nodes and  $2n$  links, each with a two-component weight vector  $\vec{w}$  as depicted in Figure 1 and a set of numbers  $a_i \in A$ ,  $0 \leq a_i \leq S$ , for  $i = 1, \dots, n$ , where  $S = \sum_{i=1}^n a_i$ . The constraints are chosen as follows:  $L_1 = nS - \frac{S}{2}$ , and  $L_2 = \frac{S}{2}$ . To solve the

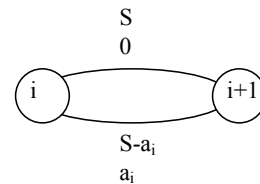


Fig. 1. The assignment of link weights to the links between nodes  $i$  and  $i + 1$ .

MCP problem, we need to find a path from node 1 to node  $n + 1$ , that obeys the constraints. Since, for all link weight vectors, the sum of the components equals  $S$ , we have that  $w_1(P) + w_2(P) = nS$ . Accordingly, a solution satisfying the constraints is only found if  $w_1(P) = nS - \frac{S}{2}$  and  $w_2(P) = \frac{S}{2}$ . The problem has now become an instance of the well-known NP-complete *partition* problem [6] and can only be solved by finding the set  $A' \subseteq A$ , for which  $\sum_{a_i \in A'} a_i = \frac{S}{2}$ . A feasible

path exists if the set  $A'$  exists, in which case it is retrieved by choosing the lower link if  $a_i \in A'$  and the upper link if  $a_i \notin A'$ . ■

”Realistic” networks often have a chain-like topology hidden in them to assure robustness/reliability (back-up paths). This is an important phenomenon, because it means that, in practice, networks have the potential to induce NP-complete behavior. The class of polynomial solvable graphs, i.e. the class of graphs in which the number of paths between two nodes increases as a polynomial function of  $N$ , is most likely very small (e.g. tree-, circle- and star-topologies). Fortunately, the underlying graph alone is not sufficient to lead to NP-complete behavior, we also need a specific link weight structure. Indeed, if all the components of the link weight vectors are the same, then irrespective of the underlying graph, the MCP problem becomes polynomially solvable. We will proceed by defining the link weight structure that leads to NP-complete behavior in the chain topology. We will use the chain topology as depicted in Figure 2 to aid in the search for a class of graphs, for which all paths from source  $s$  to destination  $d$  are non-dominated.

**Definition 2: Dominance.** Let  $\vec{a}$  and  $\vec{b}$  be two different path vectors, each consisting of  $m$  components.  $\vec{a}$  dominates  $\vec{b}$  if  $a_i \leq b_i$ , for  $i = 1, \dots, m$ , with at least one inequality. We will denote ” $\vec{a}$  dominates  $\vec{b}$ ” by ” $\vec{a} \leq \vec{b}$ ”. A vector  $\vec{a}$  is called non-dominated if there does not exist a vector  $\vec{b}$  that dominates  $\vec{a}$ . If  $\vec{a} = \vec{b}$ , we only consider one of these vectors to be non-dominated, because the other does not provide extra information for QoS routing.

In general, there are two important properties that can reduce the search-space when solving the MCP problem without loosing exactness, namely non-dominance and the constraints themselves. If a sub-path  $P$  from source  $s$  to node  $i$  exceeds one or more constraints, it can never become a feasible path<sup>1</sup>, because the path weight vector from  $i$  to destination node  $d$  consists of non-negative weights. Similar, if for two paths  $P_1, P_2$  from  $s$  to  $i$  holds that  $P_1 \leq P_2$ , then all weights of  $P_1$  are smaller (or equal) than those of  $P_2$  and hence we can omit  $P_2$  from our search-space and continue with  $P_1$  [5].

Without loss of generality, we assume that the link weights are chosen such that  $a_i > c_i$  and  $b_i < d_i$ , for  $i = 1, \dots, N$  ( $c_i > a_i$  and  $d_i < b_i$  would also have been possible). It can be verified that if  $a_i \geq c_i$  and  $b_i \geq d_i$  or  $c_i \geq a_i$  and  $d_i \geq b_i$  were allowed, this would lead to dominance.

**Property 1:** If, in a chain topology, there holds that

$$\begin{cases} a_i - c_i > \sum_{j=0}^{i-1} (a_j - c_j) \\ b_i - d_i < \sum_{j=0}^{i-1} (b_j - d_j) \end{cases} \quad (1)$$

for  $i = 1, \dots, N-1$ , where  $a_0 = b_0 = c_0 = d_0 = 0$ , then all  $2^{N-1}$  paths from node 1 to node  $N$  are non-dominated.

<sup>1</sup>This must also hold for the lower bound estimation of the end-to-end path weight vector  $\vec{w}(P) + \vec{b}$ , where  $\vec{b}$  denotes a lower bounds vector consisting of the one-dimensional shortest path weights from  $i$  to  $d$ .

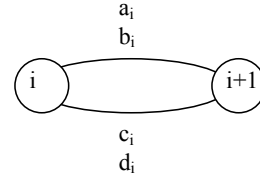


Fig. 2. Chain topology with two QoS weights per link.

*Proof:* We will give a proof by induction.

$i = 1$  : There are two paths from node 1 to node 2, namely  $P_1(1 \rightarrow 2) = \begin{pmatrix} a_1 \\ b_1 \end{pmatrix}$  and  $P_2(1 \rightarrow 2) = \begin{pmatrix} c_1 \\ d_1 \end{pmatrix}$ . According to formula (1):  $a_1 > c_1$  and  $b_1 < d_1$ , which shows that both paths from node 1 to node 2 are non-dominated.

The inductive step is to assume the correctness of formula (1) for a certain  $i$ . It remains to prove that it also holds for  $i+1$ : There are  $2^{i-1}$  paths from node 1 to  $i$ . From  $i$  there are two possible links to  $i+1$ , resulting in a total of  $2^i$  paths from node 1 to node  $i+1$ .  $2^{i-1}$  paths will follow the upper link from  $i$  to  $i+1$ , while the remaining  $2^{i-1}$  paths will follow the lower link. Since all paths at  $i$  are non-dominated (inductive assumption), the paths following the upper link are also non-dominated, because the same vector is added to each of the path vectors. The same property applies to the paths that follow the lower link. It remains to show that if (1) holds, then the paths following the upper link and the paths following the lower link do not dominate each other.

If (1) is obeyed, then all paths following the upper link possess a first path weight larger than the first weights of the paths following the lower link. Similar, the paths following the lower link have a second weight, which is larger than the second weights of the paths following the upper link. Hence the paths following different links are non-dominated. ■

**Corollary 1:** Property 1 is a sufficient but also necessary condition for all paths in the chain topology to be non-dominated.

*Proof:* We need to show that if formula (1) does not hold, then at least one path from node 1 to node  $i+1$  is dominated. If (1) does not hold, we have

$$\begin{cases} \sum_{j=0}^{i-1} c_j + a_i \leq \sum_{j=0}^{i-1} a_j + c_i \\ \sum_{j=0}^{i-1} d_j + b_i \geq \sum_{j=0}^{i-1} b_j + d_i \end{cases} \quad (2)$$

or

$$\begin{cases} \sum_{j=0}^{i-1} c_j + a_i > \sum_{j=0}^{i-1} a_j + c_i \\ \sum_{j=0}^{i-1} d_j + b_i \geq \sum_{j=0}^{i-1} b_j + d_i \end{cases} \quad (3)$$

or

$$\begin{cases} \sum_{j=0}^{i-1} c_j + a_i \leq \sum_{j=0}^{i-1} a_j + c_i \\ \sum_{j=0}^{i-1} d_j + b_i < \sum_{j=0}^{i-1} b_j + d_i \end{cases} \quad (4)$$

We have written these formulas in a slightly different form to illustrate that they correspond to two paths, namely the path that followed all the lower links up to node  $i$  and took the upper link from node  $i$  to node  $i+1$  and the path that took all the upper links towards node  $i$  and the lower link from

node  $i$  to node  $i + 1$ . Formula (2), without the equalities, is exactly the same as (1), but  $a$  is called  $c$  and  $b$  is called  $d$ . If the equality sign applies, then the path that followed all the lower links up to node  $i$  and took the upper link from node  $i$  to node  $i + 1$  is the same as the path that took all the upper links towards node  $i$  and the lower link from node  $i$  to node  $i + 1$ . According to Definition 1 one of these two paths is dominated. When formula (3) applies, the path that followed all the lower links up to node  $i$  and took the upper link from node  $i$  to node  $i + 1$  is dominated by (or dominates in the case of formula (4)) the path that took all the upper links towards node  $i$  and the lower link from node  $i$  to node  $i + 1$ . ■

At the beginning of this section we have mentioned that there are two important properties to reduce the search-space, namely non-dominance and the values of the constraints. If the constraint-values are chosen very large, then it will be easy to find a path subject to these constraints. On the other hand, if the constraint values are very strict, there may not be a path available that can obey these constraints. For the chain topology, besides formula (1), the constraints must lie in the range:

$$\begin{cases} \sum_{j=0}^{N-1} c_j \leq L_1 \leq \sum_{j=0}^{N-1} a_j \\ \sum_{j=0}^{N-1} d_j \geq L_2 \geq \sum_{j=0}^{N-1} b_j \end{cases}$$

for NP-complete behavior to occur (i.e. then the MCP problem reduces to the *partition* problem as illustrated in the NP-proof). Since  $c_i < a_i$ , the shortest path for measure 1 from node 1 to node  $N$ , equals  $\sum_{j=0}^{N-1} c_j$ . If  $L_1 < \sum_{j=0}^{N-1} c_j$ , then no feasible path exists. If  $L_1 > \sum_{j=0}^{N-1} a_j$ , then all possible (loop-free) paths can obey this constraint. The same reasoning applies to  $L_2$  and is motivated in section V. The fact that the *partition* problem is NP-complete, is because the values involved in an instance of the *partition* problem may be arbitrarily large. The same phenomenon is observed in formula (1), where the difference between  $a_i$  and  $c_i$  (and correspondingly  $d_i$  and  $b_i$ ) must grow exponentially:

$$\begin{aligned} a_{i+1} - c_{i+1} &> \sum_{j=0}^i (a_j - c_j) = (a_i - c_i) + \sum_{j=0}^{i-1} (a_j - c_j) \\ &> 2 \sum_{j=0}^{i-1} (a_j - c_j) > \dots > 2^{i-1} (a_1 - c_1) \end{aligned}$$

If  $a_i$  in the NP-proof are not chosen according to formula (1), but if they take bounded integer values, then the problem becomes polynomially solvable.

A second important phenomenon that we observe from formula (1) is that the link weights display a *negative correlation*. If the link weights would have had a positive correlation, then if  $a_i > c_i$  most likely also  $b_i > d_i$ , leading to dominance. In Section IV we will further evaluate the impact of correlation on the complexity of QoS routing.

**Corollary 2:** If there are more than two links (all with two weights) between two nodes in the chain topology, formula (1) should hold for all possible pairs of links, in order for all paths from node 1 to node  $N$  to be non-dominated.

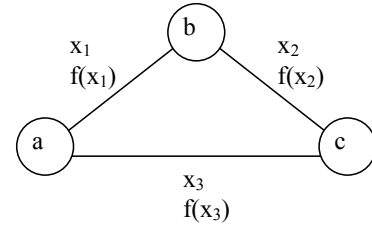


Fig. 3. Example topology to illustrate that not all functions can be mapped.

Links in the chain topology can be seen as sub-paths. In practice we do not expect links/sub-paths to obey formula (1). If formula (1) is not obeyed, corollary 2 suggests that when there are many sub-paths to a node, the probability that all these paths are non-dominated decreases.

#### IV. THE IMPACT OF LINK-CORRELATION ON COMPLEXITY

Section III indicated that correlation is necessary for NP-completeness. In this section we will discuss link-correlation by giving some properties and presenting simulation results.

##### A. Theory

Ma and Steenkiste [16] have shown that when specific dependencies (correlation) exists between QoS measures due to Weighted Fair Queueing scheduling, QoS routing can be performed in polynomial time. However, it is a misconception that if all QoS measures are a function of a common measure, then by just minimizing this common measure, we will have minimized all measures. We will illustrate that this is not always the case and provide some conditions for when this statement holds. We will denote by  $f(\cdot)$  a convex function, by  $\varphi(\cdot)$  a concave function, by  $\psi(\cdot)$  a linear function and by  $g(\cdot)$  a monotone increasing function.

Consider Figure 3. If  $f(x)$  is a convex function, then the shortest path based on  $x$  is not necessarily the shortest path for  $f(x)$ . For example, suppose that  $f(x) = e^x$  and  $x_1 = 2$ ,  $x_2 = 2$ ,  $x_3 = 3$ . Then the shortest path from  $a$  to  $c$  is  $a - c$  for  $x$ , but  $a - b - c$  for  $f(x)$ .

If  $\varphi(x)$  is a concave function, then the shortest path based on  $x$  is not necessarily the shortest path for  $\varphi(x)$ , e.g.  $\varphi(x) = \log(x)$  and  $x_1 = 1.2$ ,  $x_2 = 1.2$ ,  $x_3 = 2.2$ . Then the shortest path from  $a$  to  $c$  is  $a - c$  for  $x$ , but  $a - b - c$  for  $\varphi(x)$ .

In case of a linear function  $\psi(x)$ , then the shortest path based on  $x$  will also be the shortest path for  $\psi(x)$ .

**Property 2:** If all the components of a link weight vector are the same, the MCP problem is solvable in polynomial time with a standard one-dimensional shortest path algorithm.

**Property 3:** If  $w_i(u, v) = w_i(u', v')$ , for  $i = 1, \dots, m$ , for all  $(u, v), (u', v') \in E$ , then the MCP problem is solvable in polynomial time

Property 2 can be seen as an extreme case of positive link correlation: the  $m$  link weights on a link are correlated with coefficient  $\rho = 1$ . Property 3 can be seen as an extreme case

of positive path correlation: the  $i$ -th weights of all links are correlated to each other with coefficient  $\rho = 1$ . In practice we expect both positive link and path correlation to occur simultaneously.

In this section we consider graphs, for which all link weights are a function of a common link weight. Each link  $i$  has a weight vector  $\vec{w} = \begin{bmatrix} f_1(x_i) \\ \vdots \\ f_m(x_i) \end{bmatrix}$ , where  $x_i$  is the common link parameter (links may have different  $x_i$  and different  $f_j$ ). In the sequel we will refer to this graph as  $G_w$ . We also introduce the graph  $G_x$ , which is identical in structure to  $G_w$ , but for which the links only have weight  $x_i$ .

Let  $P_x$  be the shortest path from source  $s$  to destination  $d$  in  $G_x$ , then

$$w(P_x) = \sum_{i \in P_x} x_i \leq w(P) = \sum_{i \in P} x_i$$

where  $P$  is any other path ( $\neq P_x$ ) from  $s$  to  $d$  in  $G_x$ . Let  $\varphi(x)$  be a concave function, then

$$\varphi\left(\frac{1}{h} \sum_{i=1}^h x_i\right) \geq \frac{1}{h} \sum_{i=1}^h \varphi(x_i)$$

where  $h$  is the hopcount of a path  $P$ .

**Property 4:** If the weight vector of a link,  $\vec{w} = \begin{bmatrix} \varphi_1(x_i) \\ \vdots \\ \varphi_m(x_i) \end{bmatrix}$  with  $\varphi_j(x_i)$  concave functions, is a function of a single parameter  $x_i$  and if  $P$  is the shortest path from  $s$  to  $d$  in  $G_x$  with length  $X = \sum_{i=1}^h x_i$  and hopcount  $h$ , then  $P$  in  $G_w$  satisfies the constraint vector  $\vec{L}$  if

$$X \leq h\varphi_j^{-1}\left(\frac{L_j}{h}\right), \quad 1 \leq j \leq m \quad (5)$$

*Proof:* The constraints are obeyed if  $\sum_{i \in P} \varphi_j(x_i) \leq L_j$ . Since  $\varphi_j$  are concave functions:

$$\sum_{i=1}^h \varphi_j(x_i) \leq h\varphi_j\left(\frac{1}{h} \sum_{i=1}^h x_i\right) \leq L_j$$

or,

$$\varphi_j\left(\frac{1}{h} \sum_{i=1}^h x_i\right) \leq \frac{L_j}{h}$$

Hence,

$$X = \sum_{i=1}^h x_i \leq h\varphi_j^{-1}\left(\frac{L_j}{h}\right)$$

Note that although  $P$  is the shortest path in  $G_x$ , this does not mean that  $P$  is also the shortest path in  $G_w$  (there may be another path  $P'$  for which  $\sum_{i \in P'} \varphi(x_i) < \sum_{i \in P} \varphi(x_i)$ ). Equation (5) is a sufficient, but not a necessary condition, because there may be a path that does not obey (5) but still satisfies the constraints. ■

**Property 5:** If the weight vector of a link,  $\vec{w} = \begin{bmatrix} f_1(x_i) \\ \vdots \\ f_m(x_i) \end{bmatrix}$  with  $f_j(x)$  convex functions, is a function of a single parameter  $x_i$  and if  $P$  is the shortest path from  $s$  to  $d$  in  $G_x$  with length  $X = \sum_{i=1}^h x_i$  and hopcount  $h$ , then  $P$  (and therefore all paths) violates the constraints in  $G_w$  if

$$X > hf_j^{-1}\left(\frac{L_j}{h}\right) \quad (6)$$

for at least one  $j$ .

*Proof:* On the convexity,

$$hf_j\left(\frac{1}{h} \sum_{i=1}^h x_i\right) = hf_j\left(\frac{X}{h}\right) \leq \sum_{i=1}^h f_j(x_i)$$

The  $j$ -th constraint is violated if  $\sum_{i=1}^h f_j(x_i) > L_j$ , which is the case if  $hf_j\left(\frac{X}{h}\right) > L_j$ , which is equivalent to (6). ■

**Property 6:** If the weight vector of a link  $\vec{w} = \begin{bmatrix} g_1(x_i) \\ \vdots \\ g_m(x_i) \end{bmatrix}$

with  $g_j(x_i)$  monotone increasing and  $P$  is the shortest minimum hop path from  $s$  to  $d$  in  $G_x$  and  $x_i \leq x'_i$ , where  $x'_i$  is the  $i$ -th ordered common link weight of an other path  $P'$  from  $s$  to  $d$  in  $G_x$ , then  $P$  is also the shortest path in  $G_w$ .

*Proof:* The property is a corollary from Theorem 107 from [9]: Suppose that the sets  $(a)$  and  $(a')$  are arranged in descending order of magnitude. Then a necessary and sufficient condition that  $g(a'_1) + \dots + g(a'_n) \leq g(a_1) + \dots + g(a_n)$  should be true for all continuous and increasing  $g$  is that  $a'_v \leq a_v$  ( $v = 1, 2, \dots, n$ ). ■

## B. Simulation results

The number of distinct paths in a graph is upper bounded by the minimum cut, which is  $N - 1$  in a full mesh. In that same full mesh the number of different loop-free paths between two nodes equals  $\lfloor e(N - 2)! \rfloor$ . Consequently, in the worst-case,  $\lfloor e(N - 2)! \rfloor - (N - 1)$  paths are correlated (because they share one or multiple links with other paths). If many paths share links, as in dense graphs, they will have similar weight vectors, which increases the probability that paths will be dominated. In sparse graphs, paths will share fewer links, resulting in a smaller level of path correlation.

In this section we will evaluate the complexity of QoS routing through simulations. We will present the simulation results for the class of random graphs and the chain topology. The class of random graphs is of the type  $G_p(N)$  [1], where  $p$  is the expected link-density ( $p = 0.2$ ). The  $m = 2$  components of the link weight vector are correlated uniformly distributed random variables with correlation coefficient<sup>2</sup>  $\rho$  [18]. All simulations consisted of generating  $10^5$  different graphs and in each graph a path has been computed satisfying

<sup>2</sup>We have checked that the correlation coefficient  $\rho'$  of the generated random variables equals the desired  $\rho$ .

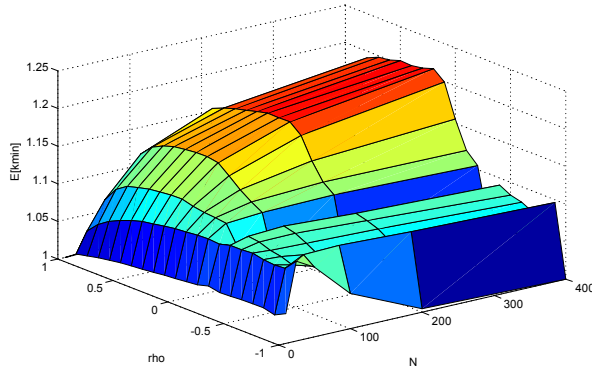


Fig. 4. Expected queue-size for the class  $G_p(N)$ , with uniformly distributed correlated link weights, as a function of the number of nodes  $N$  and the correlation coefficient  $\rho$ .

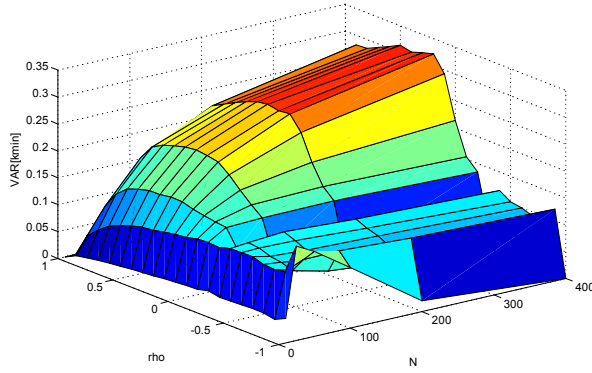


Fig. 5. Variance in queue-size for the class  $G_p(N)$ , with uniformly distributed correlated link weights, as a function of the number of nodes  $N$  and the correlation coefficient  $\rho$ .

the constraints via the SAMCRA algorithm [20]. SAMCRA does not only exactly solve the MCP problem, but also exactly solves the MCOP problem by finding the optimal path within the constraints. Since the MCOP problem is more difficult than the MCP problem, the simulation results presented here should be interpreted as an upper bound. We have simulated a worst-case scenario by choosing the constraints so large that all paths can satisfy the constraints. Therefore, SAMCRA must search in the largest search-space possible (all non-dominated paths between the source and destination), for the optimal path. If SAMCRA was only solving the MCP problem, choosing such large constraints would make the MCP problem "easy", because then any path is a solution to the MCP problem. During all simulations, we kept track of the minimum queue-size ( $k_{\min}$ ) needed to find a feasible path. This queue-size can grow as a factorial in the worst-case and presents our measure for NP-completeness in QoS routing. If TAMCRA [5], the polynomial-time predecessor of SAMCRA, had used this particular  $k_{\min}$  under the same conditions, it would have found the same optimal path as SAMCRA did. If a smaller queue-size had been used, TAMCRA would not have found the optimal path.

As illustrated in Figures 4-6, the results for the random

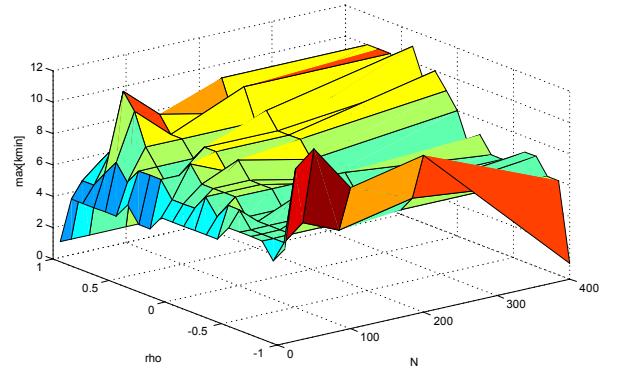


Fig. 6. The maximum observed queue-size in the class  $G_p(N)$ , with uniformly distributed correlated link weights, as a function of the number of nodes  $N$  and the correlation coefficient  $\rho$ .

graphs, do not display NP-complete behavior. We can see that a positive correlation leads to a slightly higher  $E[k_{\min}]$  than with a negative correlation. This peculiar phenomenon has only been observed in the class of random graphs, with uniformly distributed correlated link weights. If we use exponentially distributed correlated link weights, the first weight has a higher probability of being small, than with an uniform distribution. With an uniform distribution, each value for the first weight is equi-probable. Therefore, with exponentially distributed correlated link weights, there is a higher probability that the link weight vectors are similar. For uniformly distributed link weights there is a larger variability, leading to a somewhat worse performance than in the exponential case. However, in all cases the expected queue-size is close to one, leading to a complexity similar to that of Dijkstra's algorithm. The reason is most likely the randomness of the underlying graph, where the probability that a chain-like sub-graph occurs is very low. One of the measures for the "computational hardness" of a class of topologies is the average hopcount of an arbitrary path in that topology. The average hopcount (for  $m = 1$ ) scales as  $O(\log N)$  in a random graph, while as  $O(\sqrt{N})$  in a two-dimensional lattice and  $O(N)$  in the chain topology. These simulation results therefore suggest that, irrespective of the link weight structure, QoS routing in the class of random graphs (and according to [21] also Waxman graphs) is possible in polynomial time.

We have also evaluated the performance of SAMCRA in the chain topologies. The results are plotted in Figures 7 and 8.

Our simulation results<sup>3</sup> indicate that there is hardly any NP-behavior for the entire range of  $\rho$ , except for extreme negative values. We doubt that in practice link weights will display such a negative correlation, suggesting that exact QoS routing in practice, irrespective of the underlying topology, is possible in polynomial time.

<sup>3</sup>Recall that the simulation results reflect the complexity of the much more difficult MCOP problem.

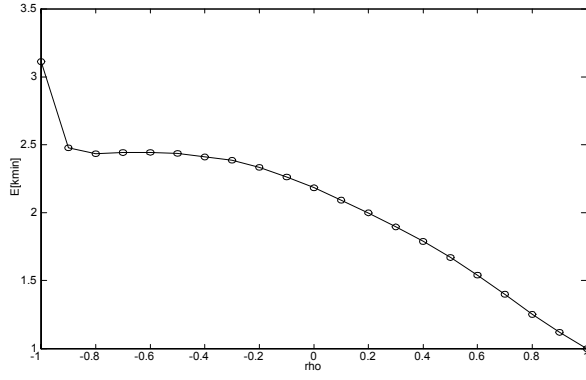


Fig. 7. The expected queue-size in the chain topology, with correlated uniformly distributed link weights for  $N = 50$ , as a function of the correlation coefficient  $\rho$ .

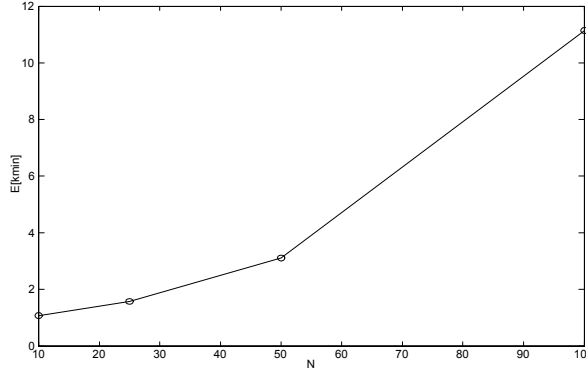


Fig. 8. The expected queue-size in the chain topology, with correlated uniformly distributed link weights for  $\rho = -1$ .

## V. THE IMPACT OF CONSTRAINTS ON COMPLEXITY

In this section we analyze the influence of the constraints on the complexity of the MCP problem. For this purpose, we will initiate an evaluation of the phase transition [2] in the MCP problem.

**Property 7:** Let  $P_{s-d;i}$  denote the path from source  $s$  to destination  $d$  for which  $w_i(P_{s-d;i}) \leq w_i(P^*)$ ,  $\forall P^*$ . Then, the MCP problem is only NP-complete when the constraints lie in the "NP-complete" range

$$w_i(P_{s-d;i}) < L_i < \max_{j=1,\dots,m} (w_i(P_{s-d;j})), \quad i = 1, \dots, m \quad (7)$$

*Proof:* We consider the two cases for which the constraints lie outside the range (7). If for any link measure  $i$  holds that  $L_i \leq w_i(P_{s-d;i})$ , then a feasible path  $P_{s-d}$  can only be present, when  $L_i = w_i(P_{s-d;i})$ , for  $i = 1, \dots, m$ . This path  $P_{s-d}$  can be found in polynomial time. Since  $w_i(P_{s-d;i})$  is the smallest attainable weight that a path can have for measure  $i$ , no feasible path can exist when  $L_i < w_i(P_{s-d;i})$ . If  $L_i \geq \max_{j=1,\dots,m} (w_i(P_{s-d;j}))$ , for  $i = 1, \dots, m$ , then all  $m$  one-dimensional shortest paths  $P_{s-d;i}$ , (for  $i = 1, \dots, m$ ) obey the constraints.

Since  $P_{s-d;i}$  can be found in polynomial time (e.g., via Dijkstra's algorithm), the MCP problem is solvable in polynomial

time if the constraints do not obey (7). ■

**Corollary 3:** The MCOP problem can only be NP-complete when  $L_i > w_i(P_{s-d;i})$ , for  $i = 1, \dots, m$ .

When the link weights are positively correlated, the NP-complete range (7) will be smaller than when the link weights are negatively correlated. This is illustrated in Figure 9 for  $m = 2$ .

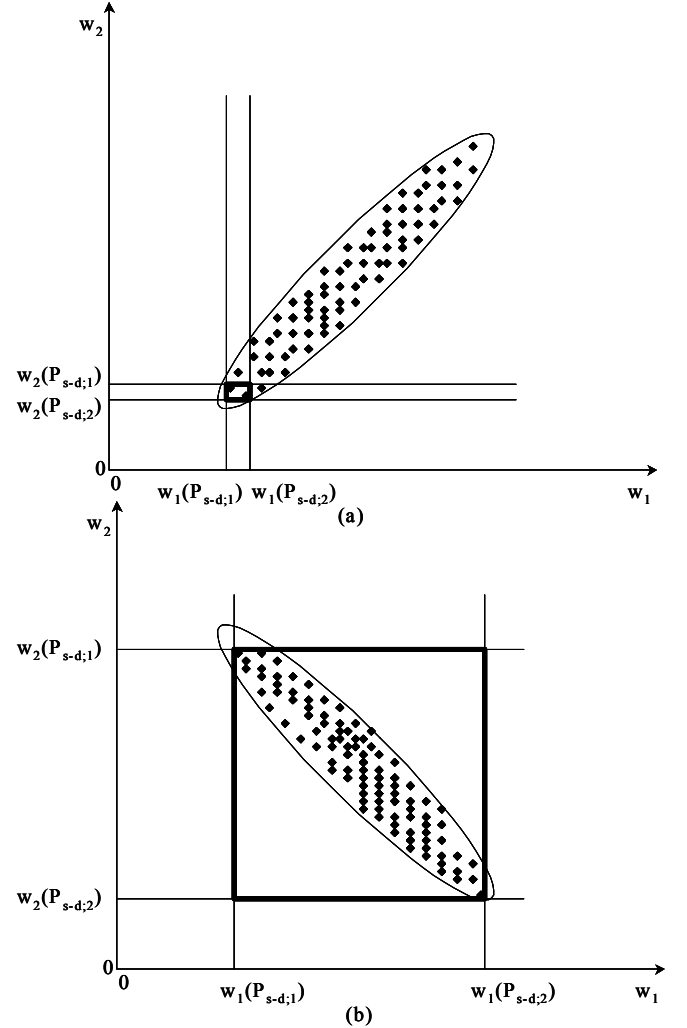


Fig. 9. The constraints range (bold square) for (a) positive correlation and (b) negative correlation. The dots in the figure denote paths in the two-dimensional space ( $m = 2$ ).

The work presented in section II suggests that there is a connection between worst-case complexity and phase transitions. Using the terminology of Gent and Walsh [8], if problems are very under-constrained, then it is usually easy to find one of the many solutions. When problems are very over-constrained, it is usually easy to determine that they are insoluble. In the phase transition in between, problems are "critically constrained" and it is typically very hard to determine if they are soluble or insoluble. Applied to the



MCP problem, we can distinct a phase transition based on the values of the constraints. If these values are smaller than the NP-complete range (7), the probability of finding a path obeying these constraints is zero. Moreover, it can be verified in polynomial time, that there exists no path in the graph that obeys the constraints (property 7). On the other hand, if the values of the constraints are very large (under-constrained), such that they exceed the NP-complete range (7), then a path satisfying these large constraints can be found in polynomial time. A phase transition is therefore expected to occur if the constraints obey the NP-complete range (7). For small values of  $L_i = w_i(P_{s-d,i}) + \epsilon$  (with  $\epsilon > 0$ ) in the NP-complete range (7), the MCP problem may still be insoluble, however the effort (complexity) needed to verify that indeed no feasible path is present in the graph has increased. In contrast to the case where the constraints  $L_i < w_i(P_{s-d,i})$ , only computing the  $m$  Dijkstra paths is not sufficient for determining that the problem is insoluble. The SAMCRA algorithm (or another exact MCP routing algorithm) must be invoked and will eventually observe that no path can obey the constraints. The larger the constraints are, the longer it will take SAMCRA to determine that no feasible path exists. Hence, increasing the constraints until a feasible path exists augments the complexity of its solution. On the other hand, when decreasing the constraints starting from the upper boundary of the NP-complete range (7), first many paths will obey these constraints  $L_i = \max_j(w_i(P_{s-d,j})) - \epsilon$  leading to a high probability that a feasible path will be found fast. If the values of the constraints decrease, the probability of finding a feasible path fast will also decrease. It is therefore expected that a phase transition occurs if there are only a few (if any) feasible paths present. In this case  $MCP \approx MCOP$ . The steepness of the phase transition depends on the NP-complete range (7), which is heavily influenced by the correlation coefficient  $\rho$  as illustrated in Figure 9. As discussed in the section IV, the correlation coefficient also impacts the level of complexity, which decreases if  $\rho$  increases. To be able to observe a phase transition, we must choose a configuration that induces NP-complete behavior: it should contain a chain structure and the link weights should have a negative correlation. We have chosen to simulate on the two-dimensional lattice in Figure 10 with  $N = 49$  nodes and correlated uniformly distributed link weights in the range  $[0,1]$ . Figure 10 also illustrates the presence of a chain topology in bold lines.

A worst-case scenario is obtained if the source node is positioned in the upper left corner and the destination node in the lower right corner, causing the largest minimum hopcount. For each constraint  $L_1$  and  $L_2$ , 100 different values were chosen in the NP-complete range (7) as discussed above, leading to a total of  $10^4$  iterations. Figure 11 displays the maximum queue-size<sup>4</sup>  $k$  used by SAMCRA, for  $N = 49$  and  $\rho = -1$ . The corresponding contour plot is given in Figure 12.

Different constraints can lead to different  $m$ -dimensional

<sup>4</sup> $k$  is different from the previously used  $k_{\min}$ , since  $k$  denotes the maximum queue-size in SAMCRA whereas  $k_{\min}$  is the queue-size that TAMCRA would have needed to attain the same solution as SAMCRA. We have used this larger value here, because  $k_{\min} = 0$  if there is no path present.

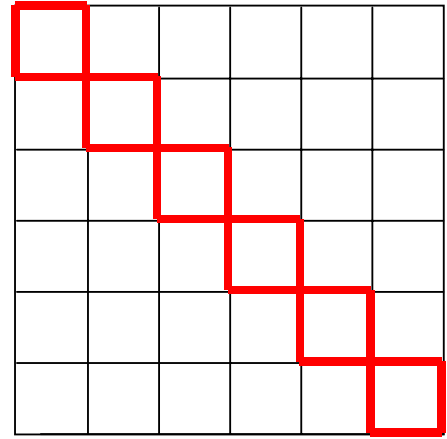


Fig. 10. Two-dimensional lattice with 49 nodes.

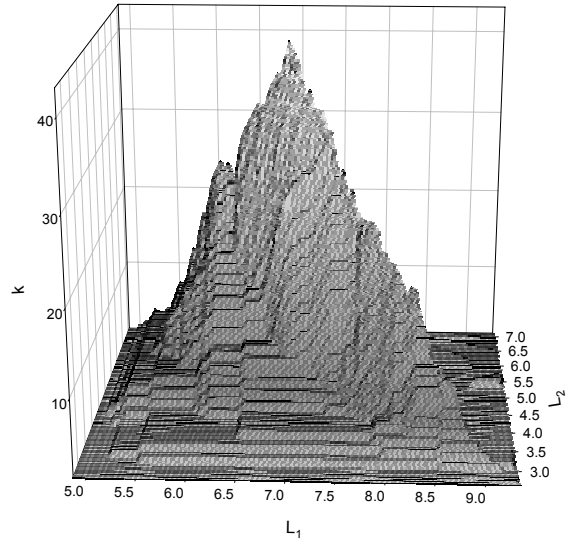


Fig. 11. The queue-size in a two-dimensional lattice, with correlated uniformly distributed link weights,  $N = 49$ ,  $\rho = -1$ , and  $10^4$  different constraint vectors.

shortest paths. For instance, if  $L_1$  is small (e.g. 5.0 in Figure 11) and  $L_2$  large (e.g. 7.0 in Figure 11), then a path  $P$  obeying these constraints must also have a small weight  $w_1(P) \leq L_1$  and the second weight may be large as long as  $w_2(P) \leq L_2$ . Since  $L_1$  is slightly larger than the weight  $w_1(P_{s-d,1})$  of the shortest Dijkstra path for measure 1, the path  $P$  may closely approximate  $P_{s-d,1}$  which may be easy to find as indicated by small  $k$  values in Figure 11. Similarly, if  $L_1$  is large (e.g. 9.0 in Figure 11) and  $L_2$  small (e.g. 3.0 in Figure 11), then a path  $P$  obeying these constraints may closely approximate the Dijkstra shortest path for measure 2  $P_{s-d,2}$  which may also be easy to find (as verified in Figure 11). We observe from Figure 11 that the complexity is largest when  $L_1 = 6.94$  and  $L_2 = 5.06$ . These values are situated near the center of the rectangle (Figure 9) spanned by the NP-complete range (7) at  $L_1^* = 7.09$  and  $L_2^* = 4.91$ . These observations seem



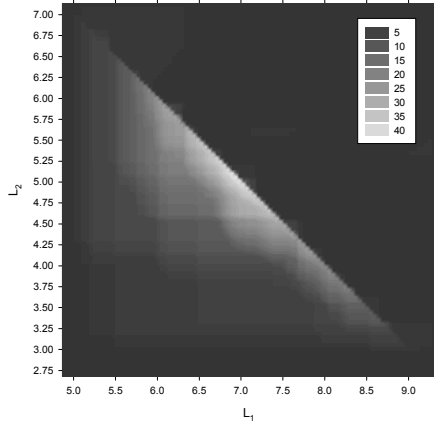


Fig. 12. Contour plot of the queue-size in a two-dimensional lattice, with correlated uniformly distributed link weights,  $N = 49$ ,  $\rho = -1$  and  $10^4$  different constraint vectors.

to suggest that the complexity is largest when the constraints closely approximate the weights of the  $m$ -dimensional shortest path  $P$ , for which  $\|\frac{\bar{w}(P)}{\bar{L}^*}\|_\infty$  is minimum and where  $\bar{L}^*$  equals the center of the NP-complete range (7).

The sharp edge/line in Figure 12, constituted by the different shortest paths, can be attributed to the extreme negative correlation ( $\rho = -1$ ) as explained in Figure 9b. Since the link weights are chosen in the range  $[0,1]$ , we have that for  $\rho = -1$ ,  $w_1(u,v) = 1 - w_2(u,v)$ ,  $\forall (u,v) \in E$ . Hence the path weights of any path  $P$  obey  $w_1(P) = h - w_2(P)$ , where  $w_i(P) = \sum_{(u,v) \in P} w_i(u,v)$  and  $h$  equals the hopcount of path  $P$ . If we again look at Figure 12, we may observe that the linear line, once continued, intersects both axes  $L_1$  and  $L_2$  at 12, which is precisely the minimum hopcount of the two-dimensional lattice with 49 nodes (as illustrated in Figure 10). Moreover, since  $w_1(P) = h - w_2(P)$ , we know that when  $L_1 + L_2 < h$ , then no feasible path exists. This means that for the class of two-dimensional lattices with correlated ( $\rho = -1$ ) uniformly distributed link weights, the constraints must obey  $L_1 + L_2 \geq h$ , for a feasible path to be possible. This condition for the constraints can be checked in polynomial time and it is therefore possible to obtain a much steeper phase transition than observed in Figures 11 and 12. Finally, we have also simulated with independent uniformly distributed link weights ( $\rho = 0$ ) in the range  $[0,1]$ . As discussed in section IV, the complexity of solving the MCP and MCOP problems is smaller than with a negative correlation. To observe a phase transition, we had to simulate with graphs larger than  $N = 49$ . Figure 13 gives the contour plot for  $N = 400$  and  $\rho = 0$ . The complexity is largest for  $L_1 = 12.58$  and  $L_2 = 15.11$ .

## VI. CONCLUSIONS

In this paper we have evaluated the complexity of Quality of Service (QoS) routing. Finding a path based on multiple QoS constraints is proven to be a NP-complete problem. However, this Multi-Constrained Path (MCP) selection problem is not NP-complete in the strong sense, meaning that a pseudo-

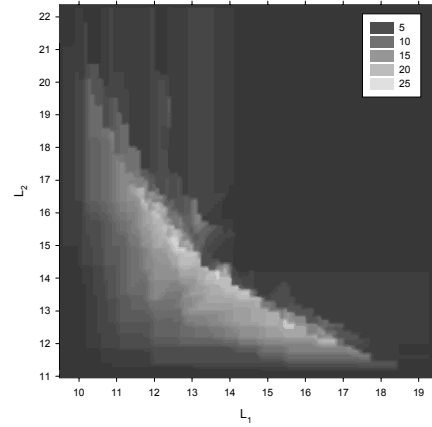


Fig. 13. Contour plot of the queue-size in a two-dimensional lattice, with uniformly distributed link weights,  $N = 400$ ,  $\rho = 0$  and  $10^4$  different constraint vectors.

polynomial algorithm can exactly solve the problem. The NP-completeness of the MCP problem hinges on four factors, namely 1) the underlying topology, 2) link weights that can grow arbitrarily large, 3) negative correlation among the link weights and 4) the values of the constraints. If the values of the constraints are very large then there is a very high probability of "easily" finding a path within the constraints. On the contrary, if the values of the constraints are very small, there is a very high probability that there is no path within the constraints. This indicates that there will be a phase transition if the constraints are around the weights of the  $m$ -dimensional shortest path in the network. In this case, it is expected to be difficult to establish whether a feasible path exists. If the four above mentioned factors are all necessary to induce NP-complete behavior, they will allow network and service providers to properly dimension their network and to avoid NP-complete behavior. Moreover, if the theory of phase transition holds for the MCP problem, then we know that QoS requirements close to the  $m$ -dimensional shortest path will, if admitted, provide the highest possible level of QoS, but also the highest computational cost. Such information is invaluable for pricing and billing mechanisms and admission control algorithms. Finally, a proper understanding and use of the four factors, will allow for efficient QoS routing at controlled computational costs.

Future study will focus on a more thorough evaluation and understanding of the expected NP-complete behavior at phase transitions. We will evaluate more distributions than solely the uniform distribution for generating link weights and use more topology generators (e.g., topologies generated using power laws that resemble the graph of the Internet). We also plan to analyze the influence of path correlation on the complexity of MCP.

## Acknowledgment

We would like to thank Selma Begtasevic for providing us with many simulation results.

## REFERENCES

- [1] B. Bollobas, *Random Graphs*, Cambridge University Press, second edition, 2001.
- [2] P. Cheeseman, B. Kanefsky and W.M. Taylor, "Where the Really Hard Problems Are", Proc. of IJCAI-91, pp. 331-337, San Mateo, CA, 1991.
- [3] S. Chen and K. Nahrstedt, "On finding multi-constrained paths", proc. of the ICC'98 conference, vol. 2, pp. 874-879, 1998.
- [4] H. De Neve and P. Van Mieghem, "A multiple quality of service routing algorithm for PNNI", IEEE ATM Workshop, pp. 324-328, Fairfax, May 26-29, 1998.
- [5] H. De Neve and P. Van Mieghem, "TAMCRA: A Tunable Accuracy Multiple Constraints Routing Algorithm", Computer Communications, 2000, vol. 23, pp. 667-679.
- [6] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, Freeman, San Francisco, 1979.
- [7] M.R. Garey and D.S. Johnson, "Strong" NP-completeness Results: Motivation, Examples, and Implications", Journal of the ACM, vol. 25, no. 3, July 1978, pp. 499-508.
- [8] I.P. Gent and T. Walsh, "Analysis of heuristic for number partitioning", Computational Intelligence, vol. 14, no. 3, pp. 430-452, 1998.
- [9] G.H. Hardy, J.E. Littlewood and G. Polya, *Inequalities*, Cambridge University Press, 2nd edition, 1973.
- [10] G. Istrate, "Computational Complexity and Phase Transitions", Proceedings of the 15th annual Conference on Computational Complexity, Florence, Italy, pp. 104-115, July 4-7, 2000.
- [11] J.M. Jaffe, "Algorithms for finding paths with multiple constraints", Networks, vol. 14, pp. 95-116, 1984.
- [12] F.A. Kuipers and P. Van Mieghem, "QoS routing: Average Complexity and Hopcount in m Dimensions", Proc. of Second COST 263 International Workshop, QoS 2001, Coimbra, Portugal, pp. 110-126, September 24-26, 2001.
- [13] F.A. Kuipers, T. Korkmaz, M. Krunz and P. Van Mieghem, "Overview of Constraint-Based Path Selection Algorithms for QoS Routing", IEEE Communications Magazine, vol. 40, no. 12, December 2002.
- [14] L.A. Levin, "Average Case Complete Problems", SIAM J. Comput., 15(1):285-286, 1986.
- [15] D.H. Lorenz and A. Orda, "QoS Routing in Networks with Uncertain Parameters", IEEE/ACM Transactions on Networking, vol. 6, no. 6, December 1998.
- [16] Q. Ma and P. Steenkiste, "Quality-of-Service Routing with Performance Guarantees", Proc. of IWQoS, New York, May 1997.
- [17] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman and L. Troyansky, "Determining computational complexity from characteristic 'phase transitions'", Nature, vol. 400, pp. 133-137, July 8, 1999.
- [18] E.F. Mykytka and C-Y Cheng, "Generating Correlated Random Variates Based on an Analogy Between Correlation and Force", Proc. of the 1994 Winter Simulation Conference, pp. 1413-1416, 1994.
- [19] D. Pisinger, *Algorithms for Knapsack Problems*, Ph.D. thesis, Dept. of Computer Science, University of Copenhagen, Denmark, February 1995.
- [20] P. Van Mieghem, H. De Neve and F.A. Kuipers, "Hop-by-hop Quality of Service Routing", Computer Networks, vol. 37/3-4, pp. 407-423, 2001.
- [21] P. Van Mieghem, "Paths in the simple Random Graph and the Waxman Graph", Probability in the Engineering and Informational Sciences (PEIS), vol. 15, pp. 535-555, 2001.
- [22] P. Van Mieghem and F.A. Kuipers, "On the Complexity of QoS Routing", to appear in Computer Communications, QoS 2001 special issue.
- [23] Z. Wang and J. Crowcroft, "Quality-of-Service Routing for Supporting Multimedia Applications", IEEE JSAC, vol. 14, no. 7, pp. 1228-1234, September, 1996.
- [24] Z. Wang, "On the complexity of quality of service routing", Information Processing Letters, vol. 69, pp. 111-114, 1999.