CCIT Report #427
June 2003

Path Protection and Blocking Probability Minimization in Optical Networks

David Lee
Bell Labs, Lucent Technologies
600 Mountain Ave.
Murray Hill, NJ 07974
lee@research.bell-labs.com

Lavy Libman*,† Ariel Orda†
Department of Electrical Engineering
Technion – Israel Institute of Technology
Haifa, Israel 32000
{libman@tx,ariel@ee}.technion.ac.il

Abstract

We study the problem of shared path protection in optical networks from the viewpoint of blocking probability minimization. Unlike the worst-case approach common in other studies, which typically deals with one failure at a time and requires recomputation of the protection paths after every failure, our framework assumes a fixed path protection strategy for every connection that does not change in response to failures in other connections, which is more realistic for networks with high failure rates. We study backup-path strategy configurations that minimize the blocking probability and derive various properties. We show that configurations which do not have partial overlaps among connections' backup paths are optimal in a rather general sense. We discuss the min-max optimal configuration properties and present an efficient algorithm for finding it in a case of special interest, while showing it to be NP-hard in general. In addition, we discuss the properties of the game resulting if each connection chooses its backup path selfishly; we show it to belong to the class of potential games, well-studied in the game theory literature, and derive several further properties resulting from its specific structure.

*Supported in part by the Israeli Ministry of Science L. Eshkol grant for Ph.D. students.

†Part of the research was performed while L. Libman and A. Orda were visiting Bell Labs, Lucent Technologies, Murray Hill, NJ.

I. Introduction

A well-recognized problem in high-speed all-optical networks is that fibers and switches frequently fail. When a failure occurs, the affected traffic must be rerouted over a backup path. There are various models for the computation and establishment of the backup path. For example, in the *link protection* model, when a link (i.e., a fiber) fails, the traffic is rerouted between the endpoints of that link, without altering other portions of the paths that have used it. In the *path protection* model, for any lightpath established for a connection[†], there is a corresponding *protection* path that is used whenever any link or node on the primary path fails. Both models, and their pros and cons, have been extensively described in the literature.

When failure recovery must be instantaneous, it is adequate to use *1+1 protection*, namely, transmit the same information on the protection path along with the primary path. Such a scheme, however, is normally required only for the most important traffic streams. Otherwise, if a small recovery delay is acceptable, the protection path need only be used as a backup after a failure on the primary path is detected. In that case, the backup path can be *shared*, fully or partially, among several connections. For the most part, the recent research on shared protection paths (e.g. [1], [2]) has taken the worst-case approach, where the basic requirement is that any two connections that traverse a common fiber or node (albeit on different wavelengths) on their primary paths must have completely disjoint backup paths, so that no single failure can cause blocking on the backup path. The problem of finding such paths is, therefore, strictly combinatorial on the network topology graph, without any relation to the *probability* of failures.

The worst-case approach can cope with one failure at a time, and thus requires that all the previous failures (as well as repairs) have been taken care of and reported to the entity that computes the backup paths.[‡] This requires extensive signaling [4], and may be unrealistic, especially if the rate of failures and repairs is high. Our study, on the other hand, considers the issue of protection paths in an alternative framework, in which dynamic information about the network state is not made available to the network nodes and connections; this framework is akin to the "failure-independent protection" concept of [5]. Thus, when a connection detects a signal loss (without detecting the particular link on its path that failed), it switches to its backup path while continuing to sample the primary path so as to detect when it becomes operational again; then, it frees the backup path and switches back to the primary path again. Neither the transitions between the primary and the backup paths, nor the failure state of particular links (which is not detected at all), need to be reported to other nodes or connections. Only the static information, i.e. the primary and backup paths of all connections, is assumed to be known.

As mentioned above, the protection paths may be shared, fully or partially, among connections. Consequently, a connection detecting a loss of signal may find itself blocked if any link on its backup path at that moment is being used by another connection (or failed on its own merit). We are concerned with finding path configurations that minimize the blocking probabilities. However, the objectives of the different connections are in conflict with each other; there may be many configurations in which reducing the blocking probability of one connection is only possible at the expense of increasing it for another. Therefore, in the first part of this paper, we study the problem of finding a *fair* path assignment, in the commonly used min-max sense (i.e. one that minimizes the maximum blocking probability of any connection). We explore the properties of such assignments; in particular, we show that the problem is NP-hard in the general case, derive a limit to its approximability in polynomial time, and present an efficient and exact solution algorithm for a case of special interest.

Subsequently, we turn our attention to noncooperative aspects of path protection, and study the properties of the game that arises if the connections choose their protection paths selfishly

[†] We use the terms 'lightpath', 'connection', and, sometimes, 'user', interchangeably.

[‡] An extension that allows for up to *two* simultaneous failures is studied in [3].

and independently of each other. Noncooperative routing and path assignment, where each user selfishly strives to optimize its own performance objective, has been studied in the past, using game-theoretic tools and concepts; in particular, [6], [7], [8], [9] describe the reasons and motivation for studying routing problems in this light, such as the inherent unwillingness or even inability of the users in large-scale networks to cooperate for a 'socially' optimal solution. Our study, though superficially similar in its model assumptions, differs from these works in a very important respect, namely, the structure of the users' cost functions: in all the works cited above, the user's objective is defined as a sum of individual link cost (or utility) functions, while the blocking probability objective cannot be expressed in this way. We describe this important difference in greater detail after defining the model formally. As a consequence, our results cannot be derived from those of previous studies on noncooperative routing; indeed, some properties of the corresponding game, such as the NP-hardness of finding the individual user's best-reply strategy, are quite different.

As is common in other studies on protection paths, we assume that the primary paths of all connections are set in advance, perhaps to satisfy other criteria such as quality-of-service (QoS) requirements; thus, the degree of freedom ("strategy space") is only in the selection of the backup paths. We assume a network with a single wavelength; thus, each link (fiber) can be used by one connection at most. Consequently, assuming that fibers fail and recover independently of each other, so do the connections traversing them. We point out that the single-wavelength assumption is still adequate even if multiple wavelengths are present in the network, as long as the primary and backup paths of a connection are required to use the same wavelength; then, essentially, the analysis holds for all wavelengths independently of each other.

The rest of this paper is structured as follows. Section II defines the formal model and notation. Section III introduces the class of partial overlap free path configurations and shows that, in a rather general sense, optimal configurations belong to this class. Section IV discusses the properties of optimal strategy configurations in the min-max sense, i.e. those that minimize the maximum blocking probability among all connections. Section V is devoted to the issues arising when the strategies are chosen by the individual connections in a noncooperative manner; in particular, it proves the existence of Nash equilibria and discusses related properties. Finally, section VI concludes our results and discusses future research directions.

II. MODEL FORMULATION

Consider an optical network whose underlying topology is described by a graph $G = (V, E)$. Every link $l \in E$ (fiber) has a probability p_l to fail (be 'down') at any given time, independently of other links; this probability need not be the same for different fibers.[†] The network is used by a set of connections $1, \dots, N$, with common source and destination nodes. A connection i is described by its pre-selected active path, from which the failure probability of the connection p_i can be drawn. As discussed in the Introduction, we assume a single wavelength in the network; hence, there is at most one connection traversing any given link, and the connections' failure probabilities are therefore independent. Furthermore, the active links can be effectively ignored for any further analysis, as they cannot be used in backup paths.[‡] We denote E' to be the set of links *not* used as active links.

The cost of a connection is defined as the probability of its path to be blocked, which happens when either another connection with an overlapping backup path fails, or a link

[†] For example, if the fiber operational status can be modeled as a Markov chain with two states ('up' and 'down') with exponentially-distributed transition times between them, then, in the steady state, there is no correlation among the down times of different fibers and a simple probability of being 'down' at an arbitrary moment is adequate.

[‡] Strictly speaking, if an active link is used in a backup path of another connection, there is a positive probability that the link will be available, if another link on the original connection's path fails and causes it to switch to its own backup path. However, in any realistic setting, the link failure probabilities are expected to be very low, hence the blocking probability of any connection using an active link on its backup path will be high.

on the connection's own backup path fails on its own merit. To write this formally (and conveniently), we define the following binary operator:

$$p_i \oplus p_j = p_i + p_j - p_i p_j,$$

and, accordingly,

$$\bigoplus_{i=1}^{N} p_i = p_1 \oplus p_2 \oplus \ldots \oplus p_N,$$

i.e. the probability of at least one occurence among $p_1, \ldots, p_N$. Note that the definition makes sense since the $\oplus$ operator is commutative and associative. We also define the *overlap indicator* function $\mathcal{O}(P_1, P_2)$ for any two paths P_1, P_2[†] to be 1 if and only if P_1 and P_2 overlap (in one link at least), and 0 otherwise. Similarly, for convenience, we define $\mathcal{O}(P, l)$ for a path P and a link $l \in E'$ to be 1 if and only if P traverses l, and 0 otherwise. Finally, we denote the backup path of connection i (also referred to as connection i's *strategy*) by S_i.[‡] With this notation, the cost functions are defined as follows:

$$J_i^\circ(S_1, \ldots, S_N) \triangleq \bigoplus_{\substack{j=1\\ j\neq i}}^{N} p_j \mathcal{O}(S_i, S_j) \oplus \bigoplus_{l\in E'} p_l \mathcal{O}(S_i, l). \tag{1}$$

Observe that the cost J_i°, as defined by (1), is the probability of connection i's path to be blocked, regardless of whether i's primary path has itself failed. Thus, it can be regarded as the blocking probability of connection i, conditioned on its own primary path failing. In addition, we shall also be interested in the unconditional blocking probability of connection i, which is

$$B_i^\circ(S_1, \ldots, S_N) \triangleq p_i \cdot J_i^\circ(S_1, \ldots, S_N). \tag{2}$$

It turns out that there exists a one-to-one monotonous mapping of the range $[0, 1]$ to the positive real-axis, under which the $\oplus$ operator becomes a simple sum. Indeed, define

$$\pi(p) \triangleq -\log(1-p); \tag{3}$$

then, $p_x = p_i \oplus p_j$ if and only if $\pi(p_x) = \pi(p_i) + \pi(p_j)$. For convenience, we henceforth abbereviate $\pi_i = \pi(p_i)$ for any $1 \leq i \leq N$ and $\pi_l = \pi(p_l)$ for any $l \in E'$. With this notation, we define

$$J_i(S_1, \ldots, S_N) \triangleq \sum_{\substack{j=1\\ j\neq i}}^{N} \pi_j \mathcal{O}(S_i, S_j) + \sum_{l\in E'} \pi_l \mathcal{O}(S_i, l). \tag{4}$$

Obviously, because of the monotonicity of the mapping, the relative ranking of the strategies under J_i is the same as with J_i°; however, working with J_i as the cost function turns out to be more convenient, and we will use it henceforth.

To complete this section, we introduce a few more definitions and notational conventions that will be handy in the subsequent discussion. First, we shall henceforth use the letters i,j,k exclusively to denote connections; thus, unless specifically stated otherwise, they are always assumed to be within $\{1, \ldots, N\}$. The letter l is used to denote links in E'. Paths are denoted by the letter P, except when referring specifically to the connections' strategies; in that case, S is used with a corresponding subscript (e.g., S_i, S_j, S_k, as well as $S_1, \ldots, S_N$).

For a path P, we denote by $\mathcal{U}(P) \triangleq \{i \,|S_i = P\}$ the set of connections whose strategy is P, and by $\mathcal{PO}(P) \triangleq \{i \,|\mathcal{O}(S_i, P) = 1\}$ the set of connections whose strategies overlap P, at least partially. We define $\pi_P \triangleq \sum_l \pi_l \mathcal{O}(P, l)$ to be the failure probability of the path

[†]Throughout this paper, unless stated otherwise, all paths mentioned are assumed by default to be from the common source to the common destination.

[‡]Throughout this paper, we generally use the letter P to denote paths, except for the specific purpose of referring to connections' strategies, for which S is used.

(more precisely, the corresponding value after the π-mapping), which is the probability of at least one link on the path to fail. The *nominal cost* of a path P is defined to be $C(P) \triangleq \pi_P + \sum_{i \in \mathcal{PO}(P)} \pi_i$; this is the cost that would be associated with a hypothetical new connection assigned to that path.

III. Optimality of Partial Overlap Free Configurations

Our first major result is that the subset of configurations with no partial overlaps, i.e., in which any two paths are either identical or completely link-disjoint, always contains an optimal configuration if the optimization criterion can be expressed via the strategy costs or the overlaps among connections.

A. Cost-related criteria

We begin by introducing the class of configurations with no partial overlaps.

Definition. A strategy configuration $\langle S_1, \ldots, S_N \rangle$ is termed *partial-overlap-free (POF)* if, for all i, $\mathcal{PO}(S_i) = \mathcal{U}(S_i)$.

In other words, in a POF configuration, if a pair of strategies (backup paths) is not link-disjoint, the paths must be identical; no partial overlaps among paths are allowed. The next fundamental lemma states that costs are minimized, in a certain sense, in POF configurations.

Lemma 1. *For any strategy configuration $\langle S_1, \ldots, S_N \rangle$ (not necessarily POF), there exists a POF configuration $\langle \tilde{S}_1, \ldots, \tilde{S}_N \rangle$ such that $J_i(\tilde{S}_1, \ldots, \tilde{S}_N) \leq J_i(S_1, \ldots, S_N)$ for all i.*

Proof. Define the set $\mathcal{R} \triangleq \bigcup_i \left[\mathcal{PO}(S_i) \setminus \mathcal{U}(S_i)\right]$, i.e., the set of all connections whose paths have partial overlaps with other connections. If $\mathcal{R}$ is empty, we are done; the configuration is already POF. Otherwise, denote

$$k^* = \arg\min_{k \in \mathcal{R}} \left[J_k(S_1, \ldots, S_N) + \pi_k\right], \tag{5}$$

and consider the alternative configuration $\langle S'_1, \ldots, S'_N \rangle$, defined as follows:

$$S'_i = \begin{cases} S_{k^*} & \text{if } \mathcal{O}(S_i, S_{k^*}) = 1, \\ S_i & \text{if } \mathcal{O}(S_i, S_{k^*}) = 0. \end{cases}$$

In other words, all the connections overlapping k^* are moved to k^*'s own path. In this alternative configuration,

- the cost of connection k^* has not changed, since it stayed in its original path and with the same overlapping other connections;
- for any $i \notin \mathcal{PO}(S_{k^*})$, the set of connections overlapping i is a subset of those in the original configuration, therefore $J_i(S'_1, \ldots, S'_N) \leq J_i(S_1, \ldots, S_N)$;
- for any $i \in \mathcal{PO}(S_{k^*})$, $i \neq k^*$,

$$J_i(S'_1, \ldots, S'_N) = J_{k^*}(S'_1, \ldots, S'_N) + \pi_{k^*} - \pi_i = J_{k^*}(S_1, \ldots, S_N) + \pi_{k^*} - \pi_i \leq J_i(S_1, \ldots, S_N) + \pi_i - \pi_i = J_i(S_1, \ldots, S_N),$$

  where the inequality is due to the choice of k^* (5).

Hence, in the new configuration, the cost of no connection has increased, yet, compared to the original configuration, the set $\mathcal{R}$ of connections with partial overlaps has been reduced by at least one element (namely, k^*, which no longer has a partial overlap with any other connection). Repeating this step as necessary, it follows by induction that eventually a POF configuration is reached, where the costs of all connections are no higher than originally. □

The following theorem follows as an immediate corollary of the above result.

Theorem 1. *For any network instance, any optimization criterion whose dependence on the strategies is only through their costs, and is monotonous in the costs, is optimized by a POF strategy configuration.*

Examples of optimization criteria that conform to the condition of Theorem 1 include minimization of maximum cost, minimization of maximum blocking probability, and minimization of the sum or average of the costs (weighted in any way that does not depend directly on the strategies).

B. Overlap-related criteria

This subsection is concerned with a different class of optimization criteria, involving directly the overlaps among connections rather than the path costs. These are addressed by the following lemma.

Lemma 2. *For any strategy configuration $\langle S_1, \ldots, S_N \rangle$ (not necessarily POF), there exists a POF configuration $\langle \tilde{S}_1, \ldots, \tilde{S}_N \rangle$ such that, for any i, j, $\mathcal{O}(\tilde{S}_i, \tilde{S}_j) = 1$ only if $\mathcal{O}(S_i, S_j) = 1$.*

Proof. Consider the graph $G' = (V, E')$ in terms of a flow network. Obviously, the maximal number of disjoint paths from the common source to the common destination is exactly the maximum flow that can be accomodated between the source and destination if all the links have a unit capacity. By the well-known max-flow min-cut theorem [10], it is also the capacity of the minimum cut, i.e., the minimum among all cuts of the number of links traversing the cut in the general direction from the source side to the destination side (not counting the links, if any, traversing the cut in the opposite direction). Each of the disjoint paths then traverses exactly one of the links in the minimum cut.

Denote the capacity of the minimum cut by M, the links in the cut by $l_1^c, \ldots, l_M^c$, and the corresponding disjoint paths between the source and destination by $P_1^c, \ldots, P_M^c$. Given a strategy configuration $\langle S_1, \ldots, S_N \rangle$, define the configuration $\langle \tilde{S}_1, \ldots, \tilde{S}_N \rangle$ as follows: $\tilde{S}_i = P_r^c$ only if S_i traverses the link l_r^c (if S_i traverses several links in the cut, choose one of them arbitrarily).

Since the new configuration is POF, it follows that, for any i, j, $\mathcal{O}(\tilde{S}_i, \tilde{S}_j) = 1$ holds if and only if $\tilde{S}_i = \tilde{S}_j$, which, by construction, requires that both S_i and S_j traversed the same link in the cut originally, hence $\mathcal{O}(S_i, S_j) = 1$. □

It follows that, given any strategy configuration S, it is always possible to rearrange the connections in a new configuration which is POF, without introducing any overlaps that did not exist originally. A direct consequence of this result is the following theorem.

Theorem 2. *For any network instance, any optimization criterion whose dependence on the strategies is only through their mutual overlap indicators, and is monotonous in them, is optimized by a POF strategy configuration.*

We emphasize that this result concerns a different class of optimization criteria than that of Theorem 1. Unlike the cost expression (4), which depends both on overlaps with other connections and on the links traversed, a criterion satisfying the condition of Theorem 2 is not allowed to depend directly on the path properties, such as, e.g., the path failure probability, hop count, etc; on the other hand, it can involve overlaps in any (monotonous) way, not just in the specific weighted sum that appears in (4).

IV. Min-Max Cost and Blocking Probability Optimization

In this section, we discuss in detail the problem of finding optimal configurations specifically in one of the following two senses: minimizing the maximum cost, or minimizing the maximum blocking probability. While these are different criteria, i.e. a configuration that optimizes one may not be optimizing the other for the same instance, they are similar enough for most of the discussion to be common for both; differences in the details will be emphasized where necessary.

Procedure MMC-Assign
Input: Disjoint paths $P_1, \ldots, P_M$.
Initialization: $N_{P_1} \leftarrow 0, \ldots, N_{P_M} \leftarrow 0$, $i \leftarrow 0$.
1. Set $i \leftarrow i+1$.
2. Set $r^* \leftarrow \arg\min_{1 \le r \le M}\{N_{P_r} \cdot \pi_i + \pi_{P_r}\}$.
3. Set $S_i \leftarrow P_{r^*}$ (i.e. assign connection i to path r^*) and $N_{P_{r^*}} \leftarrow N_{P_{r^*}} + 1$.
4. If $\sum_{r=1}^{M} N_{P_r} = N$, stop; else return to step 1.[†]

Fig. 1. Procedure MMC-Assign (Min-Max-Cost-Assign).

We begin by observing that, despite Theorem 1 that guarantees an optimum to be found among partial-overlap-free configurations, finding it remains a hard problem; it requires finding the optimal set of disjoint paths, as well as the assignment of the connections among those paths. In fact, as we shall see below, even if the disjoint paths are given, the sub-problem of optimal assignment of the connections among them is already NP-hard. Therefore, we attempt to gain insight to the properties of the optimal solution by studying special cases. Our contribution in this respect is twofold. First, we present an efficient solution algorithm in the special case of identical connections ($\pi_i = \pi_j$ for all $1 \le i, j \le N$). Subsequently, we study the general case of arbitrary connections and prove a certain property that holds when the number of connections is large. While short of solving the general case, our findings provide a significant insight to the properties of the optimal solution and can be used for its approximation under general realistic assumptions.

A. Special case: identical connections

Suppose that the connections have identical failure probabilities on their primary paths, i.e. $\pi_i = \pi_j$ for all i, j. We observe immediately that, in this case, the problems of minimizing the maximum cost or the maximum blocking probability coincide, since the factor of p_i that distinguishes between the cost (1) and the blocking probability (2) is identical for all i.

Our first goal is to show that, if the optimal set of disjoint paths were given, the remaining task of assigning the connections among the paths would be trivial, and could be accomplished simply by a greedy procedure that placed each connection in turn in the path with the least nominal cost so far. More formally, Figure 1 defines Procedure MMC-Assign, which finds the optimal assignment (i.e. one that minimizes the maximum cost) in a given set of disjoint paths. The correctness of the procedure is proved next.

Theorem 3. *For a given set of disjoint paths $P_1, \ldots, P_M$, Procedure MMC-Assign produces an assignment that minimizes the maximum cost (among all assignments in which the connections are forced to use only these paths).*

Proof. In this proof, for convenience and without loss of generality, we shall assume that the paths are originally given in a nonincreasing order of failure probabilities, i.e. $\pi_{P_1} \ge \ldots \pi_{P_M}$, and that, whenever the minimum in step 2 is attained by more than one path, the one with the lowest index is chosen to be r^*. We denote the cost of connection i in the assignment reached after Procedure MMC-Assign by $J_i^{(M)}$, and the nominal cost of path P_r in that assignment by $C^{(M)}(P_r)$. We define k to be the highest index among the connections that 'suffer' the maximum cost, i.e. $k \triangleq \max\left\{\arg\max_i J_i^{(M)}\right\}$.

Suppose that, upon completion of MMC-Assign, m paths (from among the M paths of the input) are actively used, i.e. assigned with at least one connection. Then these are the

[†]As described, this procedure takes N steps, each assigning one connection to a path. It is straightforward to see that, essentially, only M steps of "group assignments" take place in fact; e.g., if the three lowest-failure paths are P_1,P_2,P_3, then the algorithm will commence by assigning $(\pi_{P_2} - \pi_{P_1})/\pi_i$ connections to the best path (P_1), followed by $(\pi_{P_3} - \pi_{P_2})/\pi_i$ more connections to each of P_1 and P_2, and so on. This observation allows a more efficient implementation of the procedure, which is not elaborated in Figure 1 for the sake of clarity.

paths $P_1, \ldots, P_m$: indeed, whenever the path r^* chosen in step 2 is new (previously unused, i.e. $N_{P_{r^*}} = 0$), it is necessarily the one with the lowest π_{P_r} (and, consequently, the lowest index) among all the paths yet unused. Consequently, showing the resulting assignment to be optimal requires two parts: first, we show it to be no worse than any assignment that also actively uses paths $P_1, \ldots, P_m$; then, we show that no better assignment can be found that uses a different set of paths.

Observe that the sum of nominal costs of paths $P_1, \ldots, P_m$ is constant for all assignments that actively use them (and only them): $\sum_{r=1}^{m} C(P_r) = \sum_{r=1}^{m} \pi_{P_r} + N \cdot \pi_i$; alternatively, one can consider the average nominal cost, which is constant at

$$\bar{C} \triangleq \frac{1}{m} \left(\sum_{r=1}^{m} \pi_{P_r} + N \cdot \pi_i \right). \tag{6}$$

Hence, if it were possible for the nominal costs to take arbitrary continuous values, the maximum nominal cost would be minimized if all the nominal costs were simply equal to $\bar{C}$. In reality, for every r, the nominal cost of P_r is constrained to be π_{P_r} plus some integer multiple of π_i; it is straightforward to see that, in this case, a sufficient condition for an assignment to minimize the maximum nominal cost is that no two nominal costs differ by more than π_i. Alternatively, since a path's nominal cost is higher by a constant amount (specifically, π_i) than the cost of any connection assigned to it, this condition can be stated in terms of the connection costs: the maximum cost of a connection is minimized if no two costs differ by more than π_i.

On the other hand, it can be seen that, throughout the execution of MMC-Assign, the costs of no two connections indeed differ by more than π_i. This property is shown by induction on the iteration number. After the first iteration, it holds trivially. In any subsequent iteration, after steps 2–3 are performed, the cost of the connection just assigned (and all the others in its path) is at most π_i above what was the minimum cost before the iteration; consequently, the induction hypothesis is maintained. We conclude that MMC-Assign indeed reaches an assignment with the lowest maximum cost among all assignments that use the same set of paths ($P_1, \ldots, P_m$).

It remains to show that no better assignment can exist that actively uses m' paths, where $m' \neq m$. Incidentally, note that there cannot be a better assignment with m paths other than $P_1, \ldots, P_m$; choosing the paths with the lowest failure probabilities is obviously best (lower failure probabilities cannot increase the maximum cost in the optimal assignment). By the same token, we can subsequently assume that the m' paths in the allegedly better assignment are $P_1, \ldots, P_{m'}$.

First, consider any assignment that uses only the paths $P_1, \ldots, P_{m'}$, with $m' < m$. There must exist at least one path $P_{r'}$, $1 \leq r' \leq m'$, that has more connections in this assignment than in MMC-Assign. Therefore, its nominal cost in this assignment is

$$C(P_{r'}) \geq C^{(M)}(P_{r'}) + \pi_i \geq \max_r C^{(M)}(P_r), \tag{7}$$

where the rightmost inequality in (7) is due to the property of MMC-Assign already established above, namely, that the nominal costs of any pair of paths differ by no more than π_i. Consequently, an assignment using $m' < m$ paths cannot achieve a lower maximum nominal cost (and, therefore, a lower maximum connection cost) than MMC-Assign.

Conversely, suppose that a better assignment than the one found by MMC-Assign exists that uses $m' > m$ paths; thus, the path P_{m+1} is assigned at least one connection, whose cost we denote by $\tilde{J}$ (observe that the costs of all the connections in the path are identical, so it does not matter which connection is chosen if there is more than one). Then, in particular,

$$\pi_{P_{m+1}} \leq \tilde{J} < \max_i J_i^{(M)} = J_k^{(M)}. \tag{8}$$

Algorithm MMC-I
Initialization: $M \leftarrow 0$, $Min\_Cost \leftarrow \infty$.
1. Set $M \leftarrow M+1$.
2. Perform another iteration of the successive shortest path (SSP) algorithm, using π_l as the link lengths, to find M disjoint paths with minimum total length. If the iteration fails, go to step 6; else, denote the resulting paths by $P_1, \ldots, P_M$.
3. Set $Cost \leftarrow \frac{1}{M}\left(N \cdot \pi_i + \sum_{r=1}^{M} \pi_{P_r}\right)$.
4. If $Cost < Min\_Cost$, set $Min\_Cost \leftarrow Cost$ and record the paths.
5. Return to step 1.
6. Restore $P_1, \ldots, P_M$ to be the paths recorded in the last execution of step 4.
7. Execute Procedure MMC-Assign on the paths $P_1, \ldots, P_M$.

Fig. 2. Algorithm MMC-I (Min-Max-Cost-Identical).

Since, by definition, connection k is the last to have been assigned to its path during Procedure MMC-Assign, it follows that, in iteration k, the nominal cost of path P_{m+1} was lower than the one actually chosen, which contradicts step 2. $\square$

We now turn our attention to the problem of finding the optimal set of paths to run Procedure MMC-Assign on. Without loss of generality, we can assume that, if the optimal set has M^* paths, then they are all actively used in the assignment resulting after MMC-Assign; otherwise, the unused paths can be removed from the optimal set. As established during the proof of Theorem 3, the maximum nominal cost of a path in the resulting assignment is close to the average value given by (6) (more precisely, it is at most π_i above (6), and the maximum cost of a connection is, accordingly, at most π_i below (6)). Therefore, as a first approximation, the problem of optimizing the min-max cost reduces to finding the set of disjoint paths that minimizes (6), with $m = M^*$. If the optimal set size M^* were known in advance, this would simply mean finding a set of M^* disjoint paths with a minimal sum of failure probabilities. Fortunately, if the link failure probabilities π_l are treated as lengths, efficient algorithms for finding M disjoint paths with a minimal sum of lengths are known, e.g., the successive shortest path (SSP) algorithm [10]. The min-max cost configuration can therefore be found by iterating over the possible values of M, using the SSP algorithm as a subroutine to find the corresponding set of paths, and eventually keeping the paths that minimize (6). This algorithm, which we term MMC-I (for "Min-Max-Cost-Identical"), is described formally in Figure 2.

Theorem 4. *If π_l is an integer multiple of π_i for all $l \in E'$, then Algorithm MMC-I finds the min-max cost configuration precisely.*

Proof. If all the link failure probabilities are integer multiples of π_i, then so is π_P for any possible path P in the network. In that case, the maximum nominal cost resulting by the assignment in m paths can be rewritten precisely as

$$\left\lceil \frac{1}{m}\left(N + \frac{\sum_{j=1}^{m} \pi_{P_j}}{\pi_i}\right) \right\rceil \cdot \pi_i, \tag{9}$$

where $\lceil \cdot \rceil$ denotes the operation of rounding up to the nearest integer. Indeed, if $N + \frac{\sum_{j=1}^{m} \pi_{P_j}}{\pi_i}$ is an exact multiple of m, then the nominal costs are exactly balanced and given by (6); otherwise, those paths where the remainder of the connections are assigned will have a cost higher by one π_i than the rest.

We now show, by induction, that when iteration M is completed, the set of paths recorded in the last execution of step 4 corresponds to the lowest maximum cost that can be attained by M disjoint paths or less. First, this claim is obvious for the first iteration: the condition at step 4 is satisfied trivially, and the single shortest (i.e. most reliable) path is indeed saved. Now, consider iteration M. If the lowest max-cost configuration with M disjoint paths or less is indeed attained by M paths, with at least one connection assigned to each path, then its cost is given by (9) (with $m = M$), and, by virtue of the correctness of the SSP sub-algorithm, the sum of path lengths is the lowest possible; in that case, the condition in step 4 will be satisfied and the paths saved. Otherwise, i.e. if the lowest max-cost configuration with M paths or less requires only $m' < M$ paths, then, by the induction hypothesis, it has already been found at iteration m'. (In fact, due to the operation of rounding-up in (9), it is possible that the condition at step 4 will be satisfied again at iteration M without gaining further improvement, and the m'-path configuration will be replaced by an equivalent M-path one.)

Therefore, we conclude that algorithm MMC-I finds the optimal set of paths when it completes the maximum available number of disjoint paths in the graph. After that, the optimal assignment is found by executing Procedure MMC-Assign, the correctness of which was established by Theorem 3. □

We note that if the link failure probability parameters are not integer multiples of π_i, then, for a given M, the optimal collection of M disjoint paths is not necessary the one with the lowest sum of "path lengths". Indeed, suppose, for example, that there are two connections with $\pi_1 = \pi_2 = 1$. Then, a pair of paths with $\pi_{P_1} = \pi_{P_2} = 5.5$ is better than another pair with $\pi_{P_1} = 5, \pi_{P_2} = 5.7$, even though the sum of path lengths is higher; the maximum cost attained in the first pair is 5.5, while in the second pair it is 5.7. Thus, in the general (non-integer) case, algorithm MMC-I is only sufficient to find a configuration whose maximum cost is worse by up to a single π_i than the optimum. The next theorem shows that, in a certain sense, this is about the best that can be hoped for; specifically, any algorithm that reaches a better approximation than $0.5\pi_i$ must be NP-hard.

Theorem 5. *The problem of finding the min-max cost configuration with identical connections and arbitrary link failure probabilities, or even approximating it by less than $0.5\pi_i$, is NP-hard.*

Proof. The proof is by reduction from the 2-disjoint-paths problem, known to be NP-hard. An instance of the 2-disjoint-paths problem specifies a (directed) graph with two pairs of nodes (s_1, t_1 and s_2, t_2) and asks whether the graph contains two link-disjoint paths from s_1 to t_1 and from s_2 to t_2, respectively.[†]

Given a 2-disjoint-paths instance, augment the graph with a new source node s and destination node t, and links $s \to s_1$, $s \to s_2$, $t_1 \to t$, $t_2 \to t$. Set $\pi_{s\to s_1} = 0.25\pi_i$, $\pi_{s\to s_2} = 0.75\pi_i$, $\pi_{t_1\to t} = 0.5\pi_i$, and $\pi_{t_2\to t} = 0$, as well as $\pi_l = 0$ for all the links in the original graph itself (see Figure 3). We claim that, in the constructed network, the min-max cost that can be attained by two connections is $0.75\pi_i$ if and only if the 2-disjoint-paths problem has a positive solution; otherwise, the min-max cost is $1.25\pi_i$ or higher.

Indeed, if there exist two disjoint paths $s \to s_1 \to \cdots \to t_1 \to t$ and $s \to s_2 \to \cdots \to t_2 \to t$, then assigning one connection to each of these paths attains a cost of $0.75\pi_i$ for both (that is $0.25\pi_i + 0.5\pi_i$ in the first path and $0.75\pi_i + 0$ in the second). Otherwise, the best that can happen is for a path $s \to s_1 \to \cdots \to t_2 \to t$ to exist; then, however, no matter if another link-disjoint path $s \to s_2 \to \cdots \to t_1 \to t$ exists (in which case one of the connections can be assigned to it) or not (in which case both connections must be assigned to the same path), it can be seen that at least one of the connections will have a cost of $1.25\pi_i$. Therefore, any algorithm that approximates the min-max cost with identical connections by less than $0.5\pi_i$ can be used to detect the existence of disjoint paths between s_1 and t_1 and between s_2 and t_2. □

[†]The 2-disjoint-paths problem is actually a special case of the 2-commodity maximum flow problem, for unit link capacities [11].

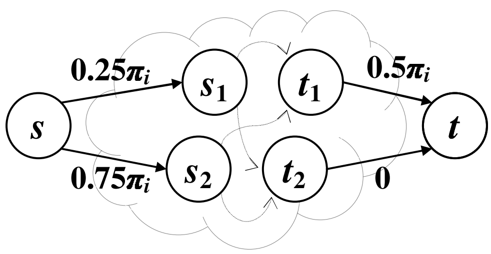

Fig. 3. The construction used in the proof of Theorem 5.

B. Arbitrary connections and the case of high sharing

We now turn back to discuss the general case, where the connections are not required to have an identical π_i for all i. In this case, even if a collection of link-disjoint paths is given, it is no longer true in general that the assignment process described above (placing each connection in turn to the path with the lowest cost so far) minimizes the maximum cost; this depends on the order of assignment. To state this formally, for a given collection of disjoint paths, minimizing the maximum cost within those paths reduces to the following grouping problem.

Problem **MIN-MAX-COST-GROUPING**

Instance: A set of connections $\{1, \dots, N\}$ with a failure probability parameter π_i for every $i = 1, \dots, N$, and a set of link-disjoint paths $\{P_1, \dots, P_M\}$ with corresponding failure probabilities π_{P_r} for every $r = 1, \dots, M$.

Question: Find an assignment of the N connections into M paths, i.e., a configuration in which $S_i \in \{P_1, \dots, P_M\}$ for all i, that minimizes $\max_r \left[\pi_{P_r} + \sum_{i \in \mathcal{U}(P_r)} \pi_i - \min_{i \in \mathcal{U}(P_r)} \pi_i\right]$.

Note that, for a group of connections sharing a common path, the cost of each connection (as defined by (4)) is the nominal cost of that path, minus the failure parameter π_i of that connection itself. Therefore, it is obvious that the maximum cost among all connections in one group is attained by the one whose own π_i is lowest. This explains the above subtraction in brackets of the smallest π_i in path r; the costs of all other connections using that path are known to be lower.

Note that if $\pi_{P_r} = 0$ for all r, then problem **MIN-MAX-COST-GROUPING** becomes a minor variation of the well-known NP-hard problem of equal-size bin packing; the only difference is the omission of the smallest element from the sum in every group. Indeed, our next theorem shows this problem to be NP-hard as well.

Theorem 6. *Problem **MIN-MAX-COST-GROUPING** is NP-hard, even for two paths with zero failure probability.*

Proof. The proof is by reduction from the NP-hard PARTITION problem [11]. An instance of that problem consists of a set (more precisely, a multiset) of numbers, and asks whether it can be partitioned into two subsets of equal sum.

Given an instance of PARTITION with K numbers, $a_1, \dots, a_K$, choose an arbitrary number A such that $A > \sum_{i=1}^{K} a_i$, and construct the following K connections: $\pi_i = A + a_i$ for all $i = 1, \dots, K$, as well as the following $K + 2$ 'auxiliary' connections: $\pi_i = A$ for all $i = K + 1, \dots, 2K + 2$. We claim that the solution of **MIN-MAX-COST-GROUPING** for these $N = 2K + 2$ connections and two paths with $\pi_{P_1} = \pi_{P_2} = 0$ achieves a maximum cost of $(K + 1)A + \frac{1}{2} \sum_{i=1}^{K} a_i$ if and only if the PARTITION instance has a solution.

Indeed, consider the solution of **MIN-MAX-COST-GROUPING**. First, note that it must assign $K + 1$ connections to each path; this way, the maximum cost will be at most $K \cdot A + \sum_{i=1}^{K} a_i$, while any other assignment puts at least $K + 2$ connections to one of the paths and thus has a maximum cost of no less than $(K + 1) \cdot A$, which is clearly higher (by the definition of A). Since there are only K connections with failure probabilities higher than A, it follows that each path must contain at least one of the 'auxiliary' connections; thus, the minimum π_i in each of the paths is equal to A. Consequently, in the optimal assignment, the sum of maximum costs in the paths, $\sum_{r=1}^{2}\left[\sum_{i\in\mathcal{U}(P_r)}\pi_i - \min_{i\in\mathcal{U}(P_r)}\pi_i\right]$, is equal to $2K\cdot A+\sum_{i=1}^{K}a_i$ in any case; accordingly, the maximum cost is minimized if the costs in both paths reach exactly half of this total, which can be attained if and only if the PARTITION instance has a solution. $\square$

Incidentally, to prevent possible confusion, we wish to emphasize that the equivalence to PARTITION in terms of hardness does not necessarily imply equivalence in the solution itself; in particular, unlike the case of identical connections, the optimal solution to **MIN-MAX-COST-GROUPING** is *not* necessarily the configuration that balances the nominal costs in all paths, even if one exists. The next example illustrates this point.

Example 1. Consider an instance with two 'heavy' connections, $\pi_1 = \pi_2 = 8$, and four 'light' ones, $\pi_3 = \cdots = \pi_6 = 1$, to be assigned to $M = 2$ paths whose own failure probabilities are zero. In this example, it can be easily verified that the optimal assignment is of both 'heavy' connections to one path (with a cost of 8 each) and the 'light' ones to the other (with a cost of 3 each), even though these costs are unbalanced and a different assignment exists which would balance the costs of both paths (namely, assign one 'heavy' and two 'light' connections to each path, which would make the maximum cost in both paths equal to 9). $\square$

Example 2. Consider the same network as in example 1 but with twenty 'light' connections instead of four, i.e., $\pi_1 = \pi_2 = 8$ and $\pi_3 = \cdots = \pi_{22} = 1$. In this case, the 'balanced' assignment of one 'heavy' and ten 'light' connections in each path is also the one that minimizes the maximum cost. However, it does not minimize the maximum blocking probability: indeed, the corresponding individual failure probabilities are $p_1 = p_2 = 1 - e^{-8}$ and $p_3 = \cdots = p_{22} = 1 - e^{-1}$, and it is easily verified (using (2)) that the maximum blocking probability is $\max_i B_i^\circ = \left(1 - e^{-8}\right)\left(1 - e^{-10}\right)$, while assigning the 'heavy' connections to one path and the 'light' ones to the other would achieve a maximum blocking probability of only $\left(1 - e^{-8}\right)\left(1 - e^{-8}\right)$. (In both cases, the maximum blocking probability is that of the 'heavy' connections.) Conversely, note that in the case of example 1, the maximum blocking probability would be minimized in the 'balanced' assignment. $\square$

Since the solution of **MIN-MAX-COST-GROUPING** cannot be easily expressed by a closed expression (such as (6) for identical connections), and, in particular, it does not have to be monotonous in the sum of the path 'lengths', it follows that the algorithm of Figure 2, based on successive shortest path computation, is no longer valid in general. Still, there are certain cases in which the optimal collection of disjoint paths can be identified, despite the fact that the subsequent solution of **MIN-MAX-COST-GROUPING** for those paths remains hard. Proposition 1 presents such a case, which we call the *high sharing* case; specifically, it states that, for a large enough number of connections (more precisely, when the sum of the connections' failure probabilities exceeds a certain threshold), it is best to use as many disjoint paths as are available. Before the proposition can be introduced, however, we need to prove the following auxiliary lemma.

Lemma 3. *Given a collection of M disjoint paths $P_1, \ldots, P_M$, it is always possible to find an assignment $\langle S_1, \ldots, S_N\rangle$ (with $S_j \in \{P_1, \ldots, P_M\}$ for all j) in which, for any connection i,*

$$J_i(S_1, \ldots, S_N) \leq \min_{1\leq r\leq M} C(P_r). \tag{10}$$

Proof. Assume, without loss of generality, that $\pi_1 \geq \cdots \geq \pi_N$, and consider the assignment reached by the following process: in each turn, connection i (starting from $i = 1$ to $i = N$) is assigned to the path r in which the nominal cost is the smallest so far. It can be shown, by induction, that (10) holds for the partial assignment resulting after each step. Indeed, at the beginning, (10) holds trivially; now, suppose that at step i^* connection i^* is assigned to path r^* and (10) held before the step. Then:

- for all connections in all paths except r^*, (10) continues to hold since the left-hand side remains unchanged while the right-hand side can only have increased;
- for connection i^* itself, the cost is precisely its path's nominal cost *before* the assignment step, which, by construction, was the minimum among all the nominal path costs;
- for any other connection $i < i^*$ that had already been assigned to path r^* previously, (10) must hold as well, since $J_i(S_1, \ldots, S_N) = J_{i^*}(S_1, \ldots, S_N) + \pi_{i^*} - \pi_i \leq J_{i^*}(S_1, \ldots, S_N)$.

Consequently, by induction, (10) holds for all connections after this assignment process is completed. □

Corollary. *Given a collection of M disjoint paths $P_1, \ldots, P_M$, it is always possible to find an assignment $\langle S_1, \ldots, S_N \rangle$ (with $S_j \in \{P_1, \ldots, P_M\}$ for all j) in which, for all $1 \leq r \leq M$,*

$$C(P_r) \leq \frac{1}{M} \left(\sum_{r'=1}^{M} \pi_{P_{r'}} + \sum_{j=1}^{N} \pi_j \right) + \frac{M-1}{M} \min_{i \in \mathcal{U}(P_r)} \pi_i, \tag{11}$$

and, accordingly, for all i,

$$J_i(S_1, \ldots, S_N) \leq \frac{1}{M} \left(\sum_{r'=1}^{M} \pi_{P_{r'}} + \sum_{j=1}^{N} \pi_j \right) - \frac{1}{M} \pi_i. \tag{12}$$

Proof. Since $\sum_{r'=1}^{M} C(P_{r'}) = \sum_{r'=1}^{M} \pi_{P_{r'}} + \sum_{j=1}^{N} \pi_j$ is constant, regardless of the assignment, it is obvious that any assignment in which connection i is in path r and

$$C(P_r) > \frac{1}{M} \left(\sum_{r'=1}^{M} \pi_{P_{r'}} + \sum_{j=1}^{N} \pi_j \right) + \frac{M-1}{M} \pi_i,$$

must also have at least one other path q in which

$$C(P_q) \leq \frac{\sum_{r'=1}^{M} C(P_{r'}) - C(P_r)}{M-1} < \frac{1}{M} \left(\sum_{r'=1}^{M} \pi_{P_{r'}} + \sum_{j=1}^{N} \pi_j \right) - \frac{1}{M} \pi_i,$$

while, on the other hand,

$$J_i(S_1, \ldots, S_N) = C(P_r) - \pi_i > \frac{1}{M} \left(\sum_{r'=1}^{M} \pi_{P_{r'}} + \sum_{j=1}^{N} \pi_j \right) - \frac{1}{M} \pi_i;$$

consequently, this assignment cannot satisfy condition (10). □

Proposition 1. *Suppose that:*

1) *for all i, $\pi_{min} \leq \pi_i \leq \pi_{max}$;*
2) *the 'shortest' (most reliable) path in the backup network has a failure probability of $\pi_{P_{min}}$;*
3) *the maximum number of disjoint paths in the backup network is M, and the lowest sum of lengths of M paths (e.g., found by the SSP algorithm) is Σ_{SSP}.*

Then, if

$$\sum_{i=1}^{N} \pi_i > (M-1) \left[\Sigma_{SSP} - M \cdot \pi_{P_{min}} + M \cdot \pi_{max} - \pi_{min} \right], \tag{13}$$

then any optimal collection of disjoint paths (in the min-max cost sense) must have M paths.

Proof. First, consider the lowest maximum cost that is conceivable in $M - 1$ disjoint paths (or less). In at least one path, the total failure probability of the connections assigned to that path must be no less than $\frac{1}{M-1}\sum_{i=1}^{N}\pi_i$. Consequently, even if all the $M-1$ paths are as reliable as the 'shortest' one, the maximum cost cannot be less than

$$\frac{1}{M-1}\sum_{i=1}^{N}\pi_i + \pi_{P_{min}} - \pi_{max}. \tag{14}$$

On the other hand, consider the collection of M paths found by SSP. By the corollary to lemma 3, it is possible to find an assignment in which the maximum cost is no higher than

$$\frac{1}{M}\left(\sum_{i=1}^{N}\pi_i + \Sigma_{SSP} - \pi_{min}\right). \tag{15}$$

Therefore, if expression (14) is higher than (15), then the configuration minimizing the maximum cost cannot be found in less than M paths. Multiplying both (14) and (15) by a common factor of $M(M-1)$ and extracting $\sum_{i=1}^{N}\pi_i$, the proposition follows. □

V. Noncooperative Path Protection

In this section, we are interested specifically in the scenario in which the backup path decision is performed by each connection for itself, without any coordination with other connections. This gives rise to a noncooperative game model, in which the connections are the "players" and their backup path selections are the "strategies". Each player strives to minimize its cost, given by (4) (or its blocking probability given by (2); from the viewpoint of any single connection, both expressions are equivalent in terms of the ranking among its strategies). More formally, the best strategy for connection i, given the strategies selected by other connections, is the one that minimizes $J_i(S_i, S_{-i})$, where S_{-i} (following standard game-theoretic notation) denotes the vector of strategies of all players except i. A configuration of strategies is called a *Nash equilibrium* if every connection's strategy is a best reply to the strategies of the other connections; that is, $\langle S_1^*, \ldots, S_N^*\rangle$ is a Nash equilibrium if, for every $1 \le i \le N$,

$$S_i^* \in \arg\min_{P} J_i(S_1^*, \ldots, S_{i-1}^*, P, S_{i+1}^*, \ldots, S_N^*). \tag{16}$$

At this point, it is helpful to emphasize the crucial difference from other routing game models studied in the literature. Routing games are similar to our setting, in the sense that path selections are made by noncooperative selfish players seeking to minimize their strategy costs. While various flavors of routing games, with some differing aspects, have been analysed in the past (see, e.g., [6], [7], [8], [9] and references therein), they all have one thing in common: the cost of a path is composed of the costs of individual links on that path; normally, a link cost is a function of only the total flow on the link (regardless of the 'identities' comprising that flow), and the path cost is defined as the sum (in some cases, the maximum) of those link costs. In contrast, no individual link cost functions can be defined in our model: the cost of using a path by a connection depends on the failure probabilities of other connections sharing that path, regardless of whether the overlap happens on just a single link or on many links along the path. Consequently, our current model requires special analysis, and its results cannot be derived from previous work on routing games, despite the superficial similarity.

A. The game potential and existence of Nash equilibria

We begin by introducing an important definition.

Definition. The *potential* of a strategy configuration $\langle S_1, \ldots, S_N\rangle$ is

$$\mathcal{L}(S_1, \ldots, S_N) \triangleq \sum_{j=1}^{N}\sum_{k=j+1}^{N}\pi_j\pi_k\mathcal{O}(S_j, S_k) + \sum_{j=1}^{N}\sum_{l\in E'}\pi_j\pi_l\mathcal{O}(S_j, l). \tag{17}$$

Lemma 4. *For any connection i, given the strategies of other connections S_{-i} and any S_i, S_i',*

$$\pi_i\left[J_i(S_i, S_{-i}) - J_i(S_i', S_{-i})\right] = \mathcal{L}(S_i, S_{-i}) - \mathcal{L}(S_i', S_{-i}). \tag{18}$$

Proof. Since the difference $\mathcal{L}(S_i, S_{-i}) - \mathcal{L}(S_i', S_{-i})$ is only due to a different strategy of i, all the terms except those including π_i vanish, i.e.,

$$\mathcal{L}(S_i, S_{-i}) - \mathcal{L}(S_i', S_{-i}) =$$
$$\left[\sum_{\substack{j=1\\ j\neq i}}^{N} \pi_i\pi_j\mathcal{O}(S_i, S_j) + \sum_{l\in E'} \pi_i\pi_l\mathcal{O}(S_i, l)\right] - \left[\sum_{\substack{j=1\\ j\neq i}}^{N} \pi_i\pi_j\mathcal{O}(S_i', S_j) + \sum_{l\in E'} \pi_i\pi_l\mathcal{O}(S_i', l)\right] =$$
$$\pi_i\left[J_i(S_i, S_{-i}) - J_i(S_i', S_{-i})\right].$$

□

We note that games possessing a potential function (i.e. a function that satisfies lemma 4) have been extensively studied in the game-theory literature; for an overview, see, e.g., [12] and references therein. Thus, by establishing the existence of a potential in our setting, all the general results from the theory of potential games can be applied; one such result, of fundamental importance, is the following.

Theorem 7. *The game possesses a Nash equilibrium.*

Proof. Since the number of strategy configurations is finite, one of them (at least) must minimize the potential; in that configuration, no connection can decrease its cost any further, and it is thus a Nash equilibrium. □

However, due to the special structure of our model, and, in particular, the fact that the strategy space of a connection is limited to the paths between its endpoints in a given network graph, the general properties of potential games provide only a limited interest. Therefore, in the next subsections we proceed to analyze our specific model more deeply.

We conclude this subsection with an example showing that a Nash equilibrium configuration is not generally unique, and that different Nash equilibria can vary widely in their performance (e.g. in the min-max sense). In this example, a network is constructed in which a configuration exists with a cost of 0 to all connections, yet has another equilibrium in which all connections overlap with each other.

Example 3. Construct a graph with the following nodes: s, t, and, for every $1 \leq i \leq N$, x_i and y_i; and the following links: for every $1 \leq i \leq N$, $(s \to x_i)$, $(x_i \to y_i)$, $(y_i \to t)$, and, except for $i = N$, $(y_i \to x_{i+1})$ (see Figure 4). Assign a failure probability $\pi_i = 1$ to all connections and $\pi_l = 0$ to all links. In this graph, there are N link-disjoint paths ($s \to x_i \to y_i \to t$ for every i); assigning one connection to each path attains an equilibrium with a cost of 0 to all connections (obviously, this is the equilibrium corresponding to the minimum potential). Yet, if the connections form a configuration in which all of them use the path $s \to x_1 \to y_1 \to x_2 \to y_2 \to x_3 \to \cdots \to y_{N-1} \to x_N \to y_N \to t$, then this configuration is an equilibrium; all the paths from s to t overlap at least one link with all the connections, so no improvement can be achieved unilaterally by any connection. In this equilibrium, obviously, the cost of each connection is $(N-1)$. □

B. Best Reply Computation is Hard

We now show that the problem faced by an individual connection to compute its best strategy, given a strategy configuration of other connections, is in general NP-hard. First, we define a simpler version of the problem that is a special case, and later prove that even this case is NP-hard.

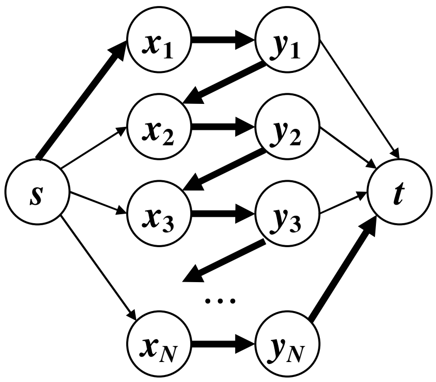

Fig. 4. The graph of Example 3. The **bold** path corresponds to the high-cost Nash equilibrium.

Problem **MIN-OVERLAP**

Instance: Graph $G = (V, E)$; collection of paths $S_1, \ldots, S_{N-1}$ with common source and destination nodes.

Question: Find a path S_N from the common source to the common destination that overlaps with as few of $S_1, \ldots, S_{N-1}$ as possible (equivalently, is link-disjoint with as many of $S_1, \ldots, S_{N-1}$ as possible).

It is obvious that **MIN-OVERLAP** is equivalent to a connection's best-reply problem in the special case of $\pi_l = 0$ for all l and $\pi_i = 1$ for all i; then, the cost of each connection is precisely the number of other connections it overlaps.

Theorem 8. *Problem MIN-OVERLAP is NP-hard.*

Proof. The proof is by reduction from the Minimum Satisfiability problem, which is known to be NP-hard [13]. An instance of that problem consists of a set of *clauses*, where each clause is a boolean 'or' of one or more *literals*, each literal being either a boolean variable or its negation; for example, $\{x_1 + \bar{x_2} + \bar{x_3}, x_2 + \bar{x_3}, x_3\}$. A solution of the instance is a truth assignment for the boolean variables that satisfies as *few* clauses as possible.

We now show a transformation from Minimum Satisfiability to **MIN-OVERLAP**. Given an instance of the former with K clauses and L variables, construct a graph with $L + 1$ nodes labeled $0, \ldots, L$, and, for every $1 \leq v \leq L$, three parallel links between nodes $v - 1$ and v, labeled "$up(v)$", "$middle(v)$", and "$down(v)$" (see Figure 5). For every clause k, construct a path S_k between nodes 0 and L as follows: for every $1 \leq v \leq L$, the path includes the link "$up(v)$" if the clause is satisfied by x_v, or the link "$down(v)$" if the clause is satisfied by $\bar{x_v}$, or the link "$middle(v)$" if the clause does not contain the variable x_v at all. In addition, construct $K + 1$ 'dummy' (and identical) paths between nodes 0 and L that use all the "$middle$" links throughout. Thus, there is a total of $N - 1 = 2K + 1$ paths in the **MIN-OVERLAP** instance.

Consider a solution to the constructed **MIN-OVERLAP** problem. Obviously, the solution path can only include "up" and/or "$down$" links; it then overlaps K paths at most, while including even one "$middle$" link would overlap $K + 1$ paths at least. However, any path that does not contain "$middle$" links corresponds to a boolean assignment of the variables ('true' for any x_v such that "$up(v)$" is on the path, 'false' otherwise), and overlaps exactly those paths that correspond to the clauses satisfied by this assignment. Hence, the solution of **MIN-OVERLAP** corresponds to the assignment that satisfies the minimum number of clauses. □

It should be realized that not only the best-reply problem is NP-hard, but the "better-reply" problem as well; that is, given the current strategy of a connection and the strategies of the others, it is intractable to decide whether a better strategy for this connection exists. If it were possible in polynomial time, then by starting with the current strategy and repeatedly looking

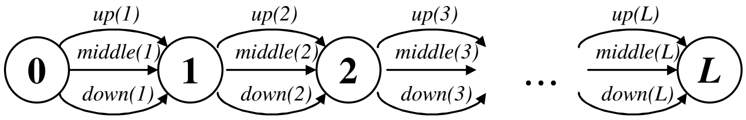

Fig. 5. The construction used in the proof of Theorem 8.

for a better one till no further improvement was possible, the best-reply strategy would be found. Even for the case of identical connections and zero failure probability of the links, the number of steps thus required to find the best-reply strategy would, obviously, be no greater than the number of connections.

However, the fact that best-reply strategies (and, consequently, Nash equilibria) are hard to detect does not reduce our interest in their properties. Even if the connections do not put a full effort to find their best-reply strategies but only check a few paths at random from time to time to see if an improvement is available, they are bound to converge to an equilibrium eventually, since the number of strategy configurations is finite and every strategy change causes a decrease in the potential; although, neither the connections themselves nor an outside observer may easily detect when the equilibrium has been reached.

C. Properties of minimum-potential configurations

As was demonstrated by Example 3, not all Nash equilibria in a given network are equivalent. If the network is to be operated at a Nash equilibrium, then, generally, the minimum-potential configuration should be preferred, as it roughly corresponds to the least inter-connection overlaps and least failure-prone paths; furthermore, it can be said (somewhat informally) that the minimum-potential equilibrium is the most stable, in the sense that small changes to the failure probabilities of connections and links, as well as adding or removing connections, are least likely to hurt the equilibrium property.

In this section, we study the properties of the minimum-potential configuration. We first prove that it can be found among partial-overlap-free configurations; note that this requires a special proof and cannot be derived from Theorem 1 or 2, because the potential expression (17) does not satisfy their requirements.[†] We then discuss the properties of the optimal collection of disjoint paths and the assignment of connections to them.

Theorem 9. *For any network instance, there exists a partial overlap free (POF) strategy configuration that minimizes the potential.*

Proof. Consider a minimum-potential strategy configuration $\langle S_1, \ldots, S_N \rangle$, and assume that it is not POF, i.e., there exists a pair of connections j and k, such that $\mathcal{O}(S_j, S_k) = 1$ and $S_j \neq S_k$. By appropriately grouping the elements in the potential expression, it can be seen that if all the connections in $\mathcal{U}(S_j)$ are moved to the path S_k, the potential increases by

$$\left(\sum_{i\in\mathcal{U}(S_j)} \pi_i\right) \cdot \left[\sum_{i\notin\mathcal{U}(S_j)\cup\mathcal{U}(S_k)}^{N} \pi_i \left[\mathcal{O}(S_k, S_i) - \mathcal{O}(S_j, S_i)\right] + \sum_{l\in E'} \pi_l \left[\mathcal{O}(S_k, l) - \mathcal{O}(S_j, l)\right]\right], \tag{19}$$

while if all the connections in $\mathcal{U}(S_k)$ are moved to the path S_j, the potential increases by

$$\left(\sum_{i\in\mathcal{U}(S_k)} \pi_i\right) \cdot \left[\sum_{i\notin\mathcal{U}(S_j)\cup\mathcal{U}(S_k)}^{N} \pi_i \left[\mathcal{O}(S_j, S_i) - \mathcal{O}(S_k, S_i)\right] + \sum_{l\in E'} \pi_l \left[\mathcal{O}(S_j, l) - \mathcal{O}(S_k, l)\right]\right]. \tag{20}$$

[†]It may seem that (17) is equivalent to the following combination: $\sum_{i=1}^{N} \pi_i J_i(S_1, \ldots, S_N)$, which would make it satisfy the requirement of theorem 1; however, this is not true: for every pair of overlapping connections i, j, it counts $\pi_i \pi_j$ twice, instead of once as in (17), while the elements due to overlaps between connections and links are counted correctly.

The expressions in brackets in (19) and (20) are seen to be exact opposites of each other. Consequently, at least one of the two expressions must be non-positive, i.e. the move in one of the directions does not increase the potential. However, such a move removes one of the partially overlapping path pairs, without introducing any new ones; thus, by repeating similar moves, a POF configuration is eventually reached which, by induction, has a potential no higher than the original. □

Consequently, finding the minimum-potential configuration, just like the min-max cost configuration (see section IV), is composed of selecting the optimal collection of link-disjoint paths, followed by optimally assigning the connections to these paths. The next theorem describes the optimal assignment once a collection of paths is given.

Lemma 5. *For a given collection of disjoint paths $P_1, \ldots, P_M$, the assignment that minimizes the potential is the one that minimizes $\sum_{r=1}^{M}[C(P_r)]^2$.*

Proof. For a configuration where the strategies are limited to the given collection of disjoint paths, the potential expression can be written as

$$\mathcal{L} = \sum_{r=1}^{M}\left[\frac{1}{2}\sum_{i\in\mathcal{U}(P_r)}\sum_{\substack{j\in\mathcal{U}(P_r)\\ j\neq i}}\pi_i\pi_j + \sum_{i\in\mathcal{U}(P_r)}\pi_i\pi_{P_r}\right] =$$

$$\sum_{r=1}^{M}\left[\frac{1}{2}\sum_{i\in\mathcal{U}(P_r)}\pi_i[C(P_r)-\pi_i-\pi_{P_r}] + [C(P_r)-\pi_{P_r}]\pi_{P_r}\right] =$$

$$\sum_{r=1}^{M}\left[\frac{1}{2}[C(P_r)-\pi_{P_r}]\sum_{i\in\mathcal{U}(P_r)}\pi_i - \frac{1}{2}\sum_{i\in\mathcal{U}(P_r)}(\pi_i)^2 + [C(P_r)-\pi_{P_r}]\pi_{P_r}\right] =$$

$$\sum_{r=1}^{M}\left[\frac{1}{2}[C(P_r)-\pi_{P_r}]^2 + [C(P_r)-\pi_{P_r}]\pi_{P_r}\right] - \frac{1}{2}\sum_{i=1}^{N}(\pi_i)^2 = \frac{1}{2}\sum_{r=1}^{M}[C(P_r)]^2 - \frac{1}{2}\sum_{r=1}^{M}(\pi_{P_r})^2 - \frac{1}{2}\sum_{i=1}^{N}(\pi_i)^2 . \tag{21}$$

Since the last two sums in (21) are constant and only $\sum_{r=1}^{M}[C(P_r)]^2$ depends on the configuration, the lemma follows. □

Corollary. *For a given collection of disjoint paths $P_1, \ldots, P_M$, if an assignment balances the nominal costs in all the paths, then it is optimal (i.e. attains the minimum potential).*

Proof. Note that $\sum_{r=1}^{M} C(P_r)$ is constant, independently of the configuration. As can be easily confirmed, for a group of elements with a constant sum, the minimum sum of squares is attained if all the elements are equal. Hence, such a configuration (if it exists) is optimal. □

It is insightful to compare this result with the min-max cost criterion, studied in section IV. As shown in example 1, the configuration minimizing the maximum cost is not necessarily the one that equalizes the nominal path costs. Indeed, consider again the parameters of example 1; namely, $\pi_1 = \pi_2 = 100$, $\pi_3 = \cdots = \pi_6 = 10$, and a network of two disjoint paths whose own failure probabilities are zero. It can be seen that the assignment of the two 'heavy' connections to one path and the four 'light' ones to the other, which was shown to minimize the maximum cost, is not a Nash equilibrium; indeed, each of the 'heavy' connections would unilaterally prefer to join the 'light' group. It is easily verified that the minimum-potential (and Nash equilibrium) configuration is the one that assigns one 'heavy' and two 'light' connections to each path; furthermore, in this case, this Nash equilibrium is unique (up to the specific connection identities in each path).

We complete this subsection by showing that in the high-sharing case, the potential is minimized in a configuration that uses the maximum number of disjoint paths available in the graph. This result is akin to Proposition 1, albeit the exact threshold is slightly different.

Proposition 2. *Under the same provisions as in Proposition 1, if*

$$\sum_{i=1}^{N} \pi_i > (M-1)\left[\Sigma_{SSP} - M \cdot \pi_{P_{min}} + \sqrt{\left(\Sigma_{SSP} - M \cdot \pi_{P_{min}}\right)^2 + \frac{M^2}{4(M-1)}\left(\pi_{max}\right)^2}\right], \tag{22}$$

then any optimal collection of disjoint paths (i.e. in which the minimum potential is attainable) must have M paths.

Proof. First, consider the lowest potential that is conceivable in $M-1$ disjoint paths (or less), attained if every path has a failure probability of $\pi_{P_{min}}$ and the nominal costs in all paths are perfectly equalized. Then, the potential value (using (21)) is

$$\frac{1}{2}\left\{(M-1)\left(\pi_{P_{min}} + \frac{1}{M-1}\sum_{i=1}^{N}\pi_i\right)^2 - (M-1)\left(\pi_{P_{min}}\right)^2 - \sum_{i=1}^{N}\left(\pi_i\right)^2\right\} =$$

$$\frac{1}{2(M-1)}\left(\sum_{i=1}^{N}\pi_i\right)^2 + \pi_{P_{min}} \cdot \sum_{i=1}^{N}\pi_i - \frac{1}{2}\sum_{i=1}^{N}\left(\pi_i\right)^2. \tag{23}$$

On the other hand, consider the collection of M paths found by SSP, and limit the attention to configurations where all the paths are actually used (i.e. assigned with one connection at least). Since the minimum-potential configuration is known to be a Nash equilibrium, it must, in particular, satisfy $C(P_r) - C(P_{r'}) \le \pi_{max}$ for all r, r'; therefore, consider the highest conceivable potential among such assignments, which is attained, as is obvious from (21), if $\sum_{r=1}^{M}\left[C(P_r)\right]^2$ is maximized and $\sum_{r=1}^{M}\left(\pi_{P_r}\right)^2$ is minimized.

Without going into the straightforward details, it can be verified that the solution of the corresponding optimization problem (maximize $\sum_{r=1}^{M}\left[C(P_r)\right]^2$, subject to $\sum_{r=1}^{M} C(P_r)$ being constant and $\max_r C(P_r) - \min_r C(P_r) \le \pi_{max}$), for an even M, is attained when $\frac{M}{2}$ paths have a nominal cost of $\frac{1}{M}\left(\sum_{i=1}^{N}\pi_i + \Sigma_{SSP}\right) + \frac{\pi_{max}}{2}$, and the other $\frac{M}{2}$ paths have a nominal cost of $\frac{1}{M}\left(\sum_{i=1}^{N}\pi_i + \Sigma_{SSP}\right) - \frac{\pi_{max}}{2}$. The value of the target expression, $\sum_{r=1}^{M}\left[C(P_r)\right]^2$, is then

$$\frac{1}{M}\left(\Sigma_{SSP} + \sum_{i=1}^{N}\pi_i\right)^2 + \frac{M}{4}\left(\pi_{max}\right)^2.^{\dagger} \tag{24}$$

In addition, $\sum_{r=1}^{M}\left(\pi_{P_r}\right)^2$ is minimized if the failure probabilities of all the paths are equal to $\frac{\Sigma_{SSP}}{M}$. Therefore, the minimum potential attainable in these paths cannot be higher than

$$\frac{1}{2}\left\{\frac{1}{M}\left(\Sigma_{SSP} + \sum_{i=1}^{N}\pi_i\right)^2 + \frac{M}{4}\left(\pi_{max}\right)^2 - M \cdot \left(\frac{\Sigma_{SSP}}{M}\right)^2 - \sum_{i=1}^{N}\left(\pi_i\right)^2\right\}, \tag{25}$$

Therefore, if expression (23) is higher than (25), then the potential-minimizing configuration cannot be found in less than M paths. After some straightforward algebraic simplification, this leads to the quadratic inequality

$$\frac{1}{2M(M-1)} \cdot \left(\sum_{i=1}^{N}\pi_i\right)^2 - \left(\frac{\Sigma_{SSP}}{M} - \cdot \pi_{P_{min}}\right) \cdot \sum_{i=1}^{N}\pi_i - \frac{M}{8}\left(\pi_{max}\right)^2 > 0, \tag{26}$$

[†] As stated, the above holds for even M. If M is odd, the solution is attained when $\frac{M-1}{2}$ paths have a cost of $\frac{1}{M}\left(\sum_{i=1}^{N}\pi_i + \Sigma_{SSP}\right) + \frac{M+1}{2M}\pi_{max}$, and the other $\frac{M+1}{2}$ paths have a cost of $\frac{1}{M}\left(\sum_{i=1}^{N}\pi_i + \Sigma_{SSP}\right) - \frac{M-1}{2M}\pi_{max}$. It can be verified that the value of $\sum_{r=1}^{M}\left[C(P_r)\right]^2$ in this case is less than given by expression (24); therefore, using (24) in the subsequent derivation yields a correct bound both for even and for odd M.

from which the proposition follows by solving for $\sum_{i=1}^{N} \pi_i$ as the inequality variable (and retaining just the positive range). □

VI. Conclusion

We have studied a network path assignment problem that was motivated by the minimization of blocking probability of backup paths in optical networks. This problem is characterized by a cost structure in which, unlike the more common bottleneck- or additive-type costs, the cost of a connection's path is determined by the other connections overlapping it, regardless of whether the overlap is on a single link or encompasses many links; thus, it cannot be defined in terms of a link cost function.

We considered both the optimal path assignment (in the sense of minimizing the maximum connection cost) and the noncooperative path assignment (where each connection selfishly chooses its own path), and established several fundamental properties for both. We proved that an optimal path assignment must exist within the set of partial-overlap-free configurations; after showing the problem to be NP-hard in general, we identified special cases where it can be solved efficiently and presented a limit to its approximability in polynomial time. For the noncooperative context, we showed the resuting scenario to belong to the class of potential games, and presented several properties of the minimum-potential configuration (which is also a Nash equilibrium); in particular, we demonstrated that, for a given set of disjoint paths, an assignment that balances the cost of all paths is also the one that minimizes the potential, but not necessarily achieves the minimum max-cost.

The model studied in this paper has a generic fundamental significance, beyond the specific context of path protection in optical networks. Indeed, the factor that distinguishes our work from various other studies of path assignment problems — namely, the path cost that includes a component dependent on other overlapping paths, regardless of the number of overlapping links — is characteristic in any other context where the path cost has an interpretation of blocking probability, e.g. whenever *overbooking* of network paths is allowed (i.e., in general, reserving more resources than the available capacity, in the anticipation that some of the reservations are not eventually used). Due to this generality, any further results that can be derived, such as efficient approximation algorithms for the optimal path assignment, have a potentially significant importance.

Our study was limited to the case in which all connections share the same source and destination, and extending the results for arbitrary source-destination pairs is a major task for further study. Since even the decision of existence of link-disjoint paths between two source-destination pairs is a well-known NP-hard problem (for directed graphs), it seems that further exploration in this direction would begin by identifying particular classes of network topologies for which the problem of minimum-cost path assignment could be solved efficiently.

In this study, we focused on networks with a single wavelength. As explained in section I, the single-wavelength assumption is adequate even for a multiple-wavelength network, as long as the backup paths are required to use the same wavelengths as the corresponding primary paths. The analysis becomes more complicated if the primary and backup paths are free to use different wavelengths, or if wavelength converters are present in the network. In these cases, the blocking probability expressions among the wavelengths become interdependent, and, in particular, the basic assumption of independence among the failure probabilities on the primary paths of different connections no longer holds; we therefore leave these cases for future study.

References

[1] K. Kar, M. Kodialam, and T. V. Lakshman. Routing restorable bandwidth guaranteed connections using maximum 2-route flows. In *Proc. IEEE Infocom*, New York, NY, June 2002.

[2] G. Li, D. Wang, C. Kalmanek, and R. Doverspike. Efficient distributed path selection for shared restoration connections. In *Proc. IEEE Infocom*, New York, NY, June 2002.

[3] H. Choi, S. Subramaniam, and H.-A. Choi. On double-link failure recovery in WDM optical networks. In *Proc. IEEE Infocom*, New York, NY, June 2002.

[4] C.-X. Chi, D.-W. Huang, D. Lee, and X.-R. Sun. Lazy flooding: a new technique for signaling in all-optical networks. In *Proc. Optical Fiber Communication (OFC)*, Anaheim, CA, March 2002.

[5] E. Bouillet, J.-F. Labourdette, G. Ellinas, R. Ramamurthy, and S. Chaudhuri. Stochastic approaches to compute shared mesh restored lightpaths in optical network architectures. In *Proc. IEEE Infocom*, New York, NY, June 2002.

[6] A. Orda, R. Rom, and N. Shimkin. Competitive routing in multiuser communication networks. *IEEE/ACM Transactions on Networking*, 1(5):510–521, October 1993.

[7] L. Libman and A. Orda. The designer's perspective to atomic noncooperative networks. *IEEE/ACM Transactions on Networking*, 7(6):875–884, December 1999.

[8] L. Libman and A. Orda. Atomic resource sharing in noncooperative networks. *Telecommunication Systems*, 17(4), August 2001.

[9] T. Roughgarden and E. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236–259, March 2002.

[10] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Upper Saddle River, NJ, 1993.

[11] M.R. Garey and D.S. Johnson. *Computers and Intractability: a Guide to the Theory of NP-Completeness*. W.H. Freeman & Company, New York, NY, 1979.

[12] M. Voorneveld, P. Borm, F. van Megen, S. Tijs, and G. Facchini. Congestion games and potentials reconsidered. *International Game Theory Review*, 1:283–299, 1999.

[13] R. Kohli, R. Krishnamurti, and P. Mirchandani. The minimum satisfiability problem. *SIAM Journal on Discrete Mathematics*, 7(2):275–283, 1994.