Int. J. Appl. Math. Comput. Sci., 2002, Vol.12, No.2, 257–267

THE BRANCH AND BOUND ALGORITHM FOR A BACKUP VIRTUAL PATH ASSIGNMENT IN SURVIVABLE ATM NETWORKS

KRZYSZTOF WALKOWIAK*

* Division of Systems and Computer Networks, Faculty of Electronics, Wrocław University of Technology
Wybrzeże Wyspiańskiego 27, 50–370 Wrocław, Poland
e-mail: walkowiak@zssk.pwr.wroc.pl

Issues of network survivability are important, since users of computer networks should be provided with some guarantees of data delivery. A large amount of data may be lost in high-speed Asynchronous Transfer Mode (ATM) due to a network failure and cause significant economic loses. This paper addresses problems of network survivability. The characteristics of virtual paths and their influence on network restoration are examined. A new problem of Backup Virtual Path Routing is presented for the local-destination rerouting strategy. The function of the flow lost due to a failure of a single link is chosen as the performance index. The problem of finding the optimal virtual path assignment is NP-complete. Therefore we develop an exact algorithm based on the branch and bound approach. Moreover, two heuristic algorithms are proposed. Numerical results are presented.

Keywords: survivable networks, ATM, branch and bound algorithm

1. Introduction

In recent years we have observed an increasing role of computer networks in the modern world. Computer networks offer nowadays a great number of opportunities to develop and conduct business electronically. Many companies, organizations and institutions rely on computer networks, using networks as a basic medium for transmitting various kinds of information. A network failure, even a short one, may cause a lot of damage and consequences, including economic losses, a significant revenue loss, political conflicts. Therefore we notice a rapid development of restoration methods that can be applied to provide network survivability.

Reducing the network construction cost and providing a sufficient level of survivability is a major problem for network planners and engineers. The increase in the bandwidth of optical transmission means that even a failure of a single link will impact many services (Kawamura and Tokizawa, 1995). Restoration methods providing survivability need to be self-healing. Self-healing means that the network has the ability to reconfigure itself around failures so that as little traffic as possible is lost (Van Landegem *et al.*, 1994).

A network failure may take place due to a wide variety of reasons, causing service disruption ranging in length from seconds to weeks. Representative events that cause failures are accidental cable cuts, hardware malfunctions, software errors, natural disasters (fires, floods, etc.) and human errors. Many reasons for failures are outside the control of network providers. A reliable network should provide a high level of availability. No system or component is failure-proof, so networks must have protection mechanisms against inevitable failures. On the one hand, traffic interruption is avoided as much as possible through secure fibre routings, cable protection, high design and equipment standards and security. On the other hand, network protection techniques are used only to limit the impact on services when inevitable outrages occur.

When a network failure occurs, a restoration mechanism is required. The process of network restoration after a failure includes: failure detection, propagation of information about the failure, spare capacity allocation, rerouting strategies (how traffic is distributed) and network control. In this paper we address only the problems of rerouting strategies and traffic distribution in ATM networks.

ATM is one of the most promising networking technologies. ATM offers: high performance, the ability to carry many types of services (data, voice, video), the ability to carry traffic over various kinds of physical networks, and Quality of Service (QoS) guarantees, which facilitate new classes of applications such as multimedia.

This paper concentrates on the issues of survivability mechanisms used in ATM networks. We present basic restoration methods, focusing on the virtual path protection method, for which three rerouting restoration strategies exist. We consider one of them, i.e. local-destination rerouting, and formulate the combinatorial optimization problem called the Backup Virtual Path Routing (BVPR) problem for local-destination rerouting. It is an NP-complete optimization problem. The objective function is the lost flow function. To our knowledge, this problem has not received much attention in the literature. There are only some simple heuristic algorithms solving the BVPR problem. Hence we construct a new exact algorithm based on the branch and bound method. Numerical experiments are included.

2. Virtual Path Concept in Survivable ATM Networks

In ATM information is transported in short fixed-length packets known as cells. ATM is a connection-oriented technology. It means that information between end systems is carried along an established virtual circuit. Routing is performed at the connection setup by making appropriate entries in routing look-up tables at every switch. ATM circuits are of two types: virtual paths (VP's), identified by virtual path identifiers (VPI's), and virtual channels (VC's), identified by VCI's. VP's are collections of VC's, and are multiplexed in a physical link. The virtual path concept has many advantages (Kawamura *et al.*, 1994; Kawamura and Tokizawa, 1995): simplifying the network structure since virtual paths are non-hierarchical, the independence of the path route establishment and the bandwidth assignment (a route is defined by a VPI), a zero bandwidth VP without an assigned bandwidth, simplifying the network control and management, grouping in one path similar traffic (traffic with the same QoS parameters). For more information on virtual paths see (Burgina and Dorman, 1991; Chlamatac *et al.*, 1994; Friesen *et al.*, 1996; Gerstel *et al.*, 1996; Sato *et al.*, 1990).

Van Landegem *et al.* (1994) presented four self-healing restoration methods:

- Automatic Protection Switching (Ayanoglu and Gitlin, 1996; Veitch and Johnson, 1997);
- Virtual Path Protection Switching (Anderson *et al.*, 1994; Ayanoglu and Gitlin, 1996; Kawamura *et al.*, 1994; Kawamura and Tokizawa, 1995; Murakami and Kim, 1996; Veitch and Johnson, 1997);
- Self-Healing Rings (Kajiyama *et al.*, 1994; May *et al.*, 1995);
- Flooding Algorithms (Ayanoglu and Gitlin, 1996; Kawamura and Tokizawa, 1995).

All schemes use some redundant resources in order to provide survivability. Note that all the four methods can use a virtual path as the basic protected element. In this paper we concentrate on the virtual path protection switching method. In this method the basic protected element is a virtual path. Each virtual path in the network has a backup virtual path. After a failure, the failed path is switched to the backup route. The process of switching is easy and includes changing VPI numbers in ATM switches. The configuration of backup virtual paths could be found by special algorithms and loaded to network nodes. At the beginning all backup virtual paths have zero bandwidth, and after activation they are assigned the necessary bandwidth.

The virtual path protection switching method includes two phases. The first phase consists in selecting a rerouting strategy and designing a backup path configuration in order to optimize survivability criteria. In (Nederlof *et al.*, 1995) the most important criteria are listed: the restoration time, lost (unrestored) traffic, the amount of spare capacity, the number of messages generated and the restoration cost. Secondly, calculation of the spare capacity is necessary to provide 100% restoration in the case of any failure. According to (Murakami and Kim, 1996), in fibre networks a single-link failure is the most common and frequently reported failure event. Therefore, in most optimization models a single-link failure is taken as the whole state space of failures and the spare capacity is computed to provide full restoration in the case of a failure of any single link. However, the method of virtual path protection switching could be also applied in networks with limited resources (the capacity of links). In such networks, 100% restoration is not always possible and routes are designed to minimize the effects of the failure. In this paper we assume that the network considered has fixed link capacities that cannot be changed.

In our approach we assume a statistical multiplexing of traffic with similar QoS parameters over virtual circuits within virtual paths, but for traffic of various virtual paths in one link we use deterministic multiplexing. Therefore the bandwidth of virtual paths can be simply summed to check capacity constraints. The notion of the equivalent capacity proposed in (Guerin *et al.*, 1991; Hui *et al.*, 1991) provides a unified metric representing the load of the virtual path and can be applied to determine the bandwidth requirement for virtual paths. This approach simplifies the analysis.

Three rerouting strategies are proposed in the literature (Anderson *et al.*, 1994; Ayanoglu and Gitlin, 1996):

- Source-Based Rerouting (Fig. 1(a)). Each virtual path has one backup copy, link-disjoint with the normal route of a virtual path. The affected virtual path is traced back to its source node, which reroutes the virtual path on a precomputed backup path, or the source node finds the backup route based on all information that the node has. The primary advantage of source-based rerouting is that it covers the whole network, so the spare capacity is used efficiently and the virtual path activation is distributed among many nodes. Nevertheless, a large number of restoration messages is generated in the network, the time of restoration is large and the process of determining backup routes is complicated.

- Local Rerouting (Fig. 1(b)). The local rerouting is the opposite of the foregoing technique. The backup route is found only around the failed link. Each virtual path has as many backup routes as links. Each backup route covers a normal route except the failed link. The nodes adjacent to the failed link are responsible for the process of rerouting. All the affected virtual paths are processed locally, and therefore the restoration time is small. Drawbacks of local rerouting are the following: the restoration process is performed by the same set of nodes, and only resources of a spare capacity near to the failure are applied.
- Local-Destination Rerouting (Fig. 1(c)). This strategy is a compromise between local and source-based reroutings. In this scheme the backup route is disjoint with the normal route starting from a beginning node of the failed link. Therefore, each virtual path has as many backup routes as links. The beginning node of the failed link is responsible for the process of rerouting. This scheme results in intermediate requirements of a spare capacity and the restoration time.

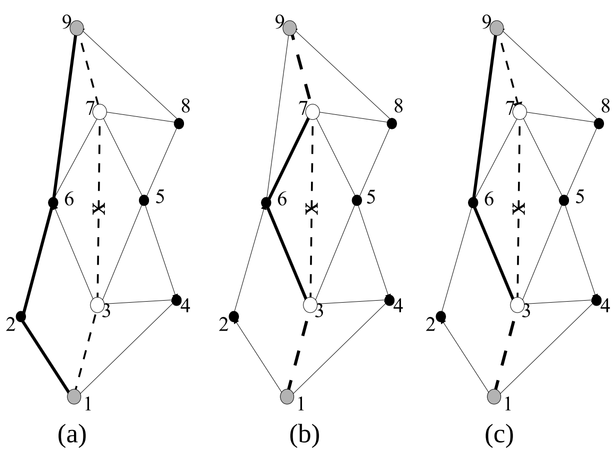

Fig. 1. Rerouting strategies: (a) source-based rerouting, (b) local rerouting, (c) local-destination rerouting.

Figure 1 illustrates the presented rerouting strategies. The virtual path with the working route 1-3-7-9 is broken when the link 3-7 fails. For source-based rerouting Node 1 is informed about the failure and is responsible for switching to the backup path 1-2-6-9 (Fig. 1(a)). In the local rerouting scheme, Node 3 switches to the backup path 3-6-7 to omit the failed link (Fig. 1(b)). It results in the backup route 1-3-6-7-9 for the demand pair 1-9. Finally, for local-destination rerouting, Node 3 switches the path to the 3-6-9 route, since the path is not changed from the source node to the beginning node of the failed link, it yields the 1-3-6-9 backup route (Fig. 1(c)). If we choose for the local rerouting strategy the 3-6-9-7 path to omit the 3-7 link, the considered virtual path suffers from backhauling, as two extra hopes are traversed (from Node 9 to Node 7, and back from Node 7 to Node 9) (Anderson *et al.*, 1994; Iraschko *et al.*, 1998).

3. Backup Virtual Path Routing Problem

In this section we formulate the Backup Virtual Path Routing (BVPR) problem. We apply the local-destination strategy for flow rerouting after a failure of a single link. The local-destination rerouting strategy has not been addressed broadly in the literature and, to our knowledge, the combinatorial optimization problems presented below have not received much attention in the literature. Consider an ATM network modelled as a directed graph $G = (N, L, C)$, where N is a set consisting of n nodes representing ATM switches, L is a set of l links and C is a vector of link capacities. Let $o(m)$ denote the origin node of link m and $d(m)$ denote the destination node of link m. We assume that the estimated bandwidth requirements and working routes of virtual paths are given. We consider a failure of a single link m. We must select backup routes in order to minimize the objective function of the flow lost after restoration of the failed link m. A set of backup route proposals that conform to the rerouting strategy for every virtual path are given. Such sets can be generated using the hop-limit approach (Herzberg *et al.*, 1995). This means that we do not process all possible routes, but only a subset of them. Accordingly, a significant reduction in the optimization problem size can be obtained. Moreover, in this approach we can eliminate backup routes that cause backhauling in the local rerouting (Anderson *et al.*, *1994*; Ayanoglu and Gitlin, 1996).

To mathematically represent the problem, we introduce the following notation:

c_i capacity of link i,

P set of p virtual paths in the network,

P_m set of p_m virtual paths which use link m,

Q_i estimated bandwidth requirement for VP i,

Π_{im} set of backup routes for VP i after a failure of link m, $\Pi_{im} = \{\pi^k_{im} : k = 1, \dots, l^m_i\}$,

$$a_{ij} = \begin{cases} 1 & \text{if the working route for VP } i \text{ uses link } j \in L, \\ 0 & \text{otherwise,} \end{cases}$$

$$y^k_{im} = \begin{cases} 1 & \text{if } \pi^k_{im} \text{ is the backup route for VP } i, \\ & \text{after a failure of link } m, \\ 0 & \text{otherwise,} \end{cases}$$

$$y^0_{im} = \begin{cases} 1 & \text{if the VP } i \text{ is not restored after a failure} \\ & \text{of link } m, \\ 0 & \text{otherwise,} \end{cases}$$

$$b_{im}^{kj} = \begin{cases} 1 & \text{if the route } \pi_{im}^k \text{ uses link } j, \\ 0 & \text{otherwise.} \end{cases}$$

Since the virtual path can use only one route, we have

$$\sum_{k=0}^{l_i^m} y_{im}^k = 1 \ \text{ for } m \in L, \ \ i \in P_m. \tag{1}$$

Let $\hat{Y}_r^m$ be the permutation of the values of all variables $y_{im}^k,\ k = 0, 1, \dots, l_i^m$, for which the condition (1) is satisfied, and let Y_r^m be the set of variables which are equal to one in the permutation $\hat{Y}_r^m$. The set Y_r^m is called a *selection*. Each selection determines a unique set of backup routes used for restoration of a survivable ATM network.

Let f_{jr}^m be the total flow on the j-th link after a restoration of the failed link m. We have

$$f_{jr}^m = \sum_{i=1}^{p} a_{ij} Q_i - \sum_{i=1}^{p} a_{ij} a_{im} Q_i + \sum_{i=1}^{p} \sum_{k=1}^{l_i^m} b_{im}^{kj} y_{im}^k Q_i. \tag{2}$$

The residual capacity on the j-th link after a restoration of the failed link m can be calculated as $e_{jr}^m = c_j - f_{jr}^m$.

Let R_Y^m denote the family of all selections Y_r^m for which the following condition is satisfied:

$$f_{ir}^m \le c_i \ \text{ for } \ i = 1, \dots, l. \tag{3}$$

The objective function is formulated as follows:

$$LFB_m(Y_r^m) = \sum_{i=1}^{p} a_{im} y_{im}^0 Q_i. \tag{4}$$

The function $LFB_m(Y_r^m)$ is the sum over all virtual paths that use link m (variable a_{im} is equal to 1) and are not restored after a failure of this link (variable y_{im}^0 is equal to 1). Therefore $LFB_m(Y_r^m)$ represents the flow that is lost after a failure of link m. The problem of Backup Virtual Path Routing in a survivable ATM network can be formulated as follows:

$$\min_{Y_r^m} LFB_m(Y_r^m) \tag{5}$$

subject to

$$Y_r^m \in R_Y^m. \tag{6}$$

In the BVPR problem, the decision variables are y_{im}^k. The selection Y_r^m is the set of all variables y_{im}^k which are equal to one. The function $LFB_m(Y_r^m)$ is an objective function that is to be minimized. The constraints of the BVPR problem are: condition (1), since a virtual path may use only one route; condition (3), since the flow of each link cannot exceed the capacity of that link; and condition (6), which states that only a feasible selection Y_r^m belonging to the set R_Y^m can be a solution to the BVPR problem. Note that the constraint (6) includes the constraints (1) and (3), therefore the BVPR problem can be defined as (5) and (6).

4. Branch and Bound Algorithm

The problem (5), (6) is NP-complete because it is equivalent to the non-bifurcated flow problem, which is NP-complete (Fratta *et al.*, 1973). Therefore we applied the branch and bound method to construct the exact algorithm. The branch and bound method is an intelligently structured search over the space of all feasible solutions. The solution space is repeatedly partitioned into smaller and smaller subsets, and a lower bound of the objective function is calculated within each subset. Those subsets with the bound that exceeds the best solution are excluded from further partitioning.

Generally, there are two methods that can be applied to find an optimal solution to network optimization problems related to the BVPR problem: linear (dynamic) programming and branch and bound algorithms. For complex problems like BVRP, the branch and bound method is more effective, since the number of variables is large. Moreover, the branch and bound algorithms have already been to solve network optimization problems related to the BVPR problem (Kasprzak 1985; Kasprzak 1989; Wang *et al.*, 1997; Kasprzak and Walkowiak, 2000; Walkowiak, 2000a; Walkowiak, 2000b).

4.1. Calculation Scheme

In our branch and bound algorithm, we start with the selection Y_1^m and generate a sequence of selections Y_r^m. In order to obtain the initial selection Y_1^m, we must solve the BVPR problem using heuristic algorithms. More details about these algorithms can be found in the next section. Each selection Y_r^m is obtained from a certain selection Y_p^m of the sequence by complementing one variable y_{im}^k by another variable y_{im}^h. Both the variables must be associated with the same virtual path i. For every selection Y_r^m we constantly fix a subset $U_r^m \in Y_r^m$ and momentarily fix a set U_r^{mt}. The variables in U_r^m are constantly fixed and denote the path from the initial selection Y_1^m to the current selection Y_r^m. Each momentarily fixed variable in U_r^{mt} is the variable abandoned during the backtracking process. There are two important elements in the branch and bound algorithm that are calculated for each selection Y_r^m: a lower bound of the criterion function and branching rules. The lower bound is calculated to check if a "better" solution can be found. If the testing is negative, we abandon the selection Y_r^m under consideration and backtrack to the selection Y_p^m, from which the selection Y_r^m was generated. The basic task of the branching rules is to find variables for complementing to generate a new selection with the least possible value of the criterion function. A more detailed description of the calculation scheme can be found in (Walkowiak, 2000a).

4.2. Heuristic Algorithms

In order to obtain an initial solution for the branch and bound algorithm, we propose two heuristic algorithms. One of them constitutes a modification of the flow deviation algorithm for non-bifurcated flows proposed in (Fratta *et al.*, 1973). We introduced a new link metric that is used to calculate the route's length. We implemented this algorithm and compared the results with the optimal results given by the branch and bound algorithm. More details can be found in Section 5.2.

The second proposition was an algorithm applying the genetic approach. We modified the well-known genetic algorithm proposed in (Goldberg, 1989). We tested the genetic algorithm for an optimization problem related to the BVPR problem. The obtained results were very close to optimal ones. More details on this subject are included in (Walkowiak, 2000a).

4.3. Branching Rules

The elementary task of the operation is to choose a normal variable and to reverse variables for complementing and generating a successor Y_s^m of the current selection Y_r^m with the least possible value of the criterion function. If the capacity constraint (3) for the current selection Y_r^m is satisfied ($Y_r^m \in R_Y^m$), we apply the choice operation. Otherwise we use the regulation operation.

4.3.1. Choice Operation

The purpose of the choice operation is to find variables $y_{im}^k = 0$, $y_{im}^h = 1$, where $y_{im}^k \in Y_r^m$, and to generate the new successor $Y_s^m = (Y_r^m - \{y_{im}^k\}) \cup \{y_{im}^h\}$. The choice operation consists of two phases. In the first one, we try to select variables for complementing for which the value of the objective function $LFB_m(Y_s^m)$ for a new selection will decrease. To obtain this, we restore a virtual path i ($y_{im}^0 = 1$, $y_{im}^0 \in Y_r^m$) which has not been restored yet.

If we cannot restore any virtual path i with $y_{im}^0 = 1$, then in the second phase we change the backup route of one of the virtual paths with $y_{im}^0 = 1$. As a result, this complementing should produce a new selection Y_s^m with more potentiality to restore a virtual path which has not been restored yet.

The set $E_r^m = Y_r^m - U_r^m$ includes variables y_{im}^k not constantly fixed, which can be complemented in the branching rule. Let E_r^{m0} denote all variables y_{im}^0 not constantly fixed:

$$E_r^{m0} = \bigcup_{i:y_{im}^0 \in E_r^m} \{y_{im}^0\}.$$

Let M_r^m be a set of variables y_{im}^k that are equal to zero and let variables y_{im}^h associated with them belong to E_r^m, i.e.

$$M_r^m = \left(\bigcup_{i:y_{im}^h \in E_r^m} \left\{ \bigcup_{k=0}^{l_i^m} \{y_{im}^k\} \right\} \right) - E_r^m.$$

We introduce a metric representing the length of the route π_{im}^k:

$$l_r^{FB}(\pi_{im}^k) = -\left(a_{im} \min_{j:b_{im}^{kj}=1} (e_{jr}^m) \right). \tag{7}$$

Notice that if we look for the shortest route according to metric $l_r^{FB}(\pi_{im}^k)$, we select a route for which the minimal value of the residual capacity e_{jr}^m of all the links belonging to π_{im}^k is the largest:

$$\left(-l_r^{FB}(\pi_{im}^k) \right) \geq Q_i. \tag{8}$$

We can use the metric (7) to check whether or not the activation of the route π_{im}^k satisfies the capacity constraint (3).

Let O_r^m denote the set of variables y_{im}^k equal to zero for which variables y_{im}^0 associated with them belong to E_r^{m0} and the condition (8) is satisfied, i.e.

$$O_r^m = \bigcup_{i:y_{im}^0 \in E_r^{m0}} \Big\{ y_{im}^k : k = 1, \dots, l_i^m \text{ and } \left(-l_r^{FB}(\pi_{im}^k) \right) \geq Q_i \Big\}.$$

Theorem 1. *Let $Y_r^m \in R_Y^m$. If $O_r^m \neq \emptyset$ and Y_s^m was generated from Y_r^m by complementing $y_{im}^0 = 0$, $y_{im}^k = 0$, where $y_{im}^0 \in E_r^{m0}$, $y_{im}^k \in O_r^m$, then*

$$LFB_m(Y_s^m) = LFB_m(Y_r^m) - Q_i$$

and

$$Y_s^m \in R_Y^m.$$

Proof. Notice that the structure of the sets O_r^m and E_r^{m0} implies the theorem. We assume that O_r^m is not empty. Hence there exist variables $y_{im}^0 \in E_r^{m0}$ and $y_{im}^k \in O_r^m$ such that selecting the backup route π_{im}^k for restoration of the virtual path does not violate the capacity constraint (3). Consequently, it implies $Y_s^m \in R_Y^m$. From the definition of the function $LFB_m(Y_r^m)$, we know that generating a new selection $Y_s^m = (Y_r^m - \{y_{im}^0\}) \cup \{y_{im}^k\}$ produces a reduction of Q_i in the value LFB_m, where Q_i is the bandwidth of virtual path i. Therefore

$$LFB_m(Y_s^m) = LFB_m(Y_r^m) - Q_i.$$

■

Let Q_{ri}^m denote the maximal value of the bandwidth of all the virtual paths i for which $y_{im}^k \in O_r^m$, i.e.

$$Q_{ri}^m = \max_{j:y_{jm}^k \in O_r^m} Q_j. \tag{9}$$

If the set O_r^m is not empty, then in the first phase of the choice operation we select for complementing variables $y_{im}^0 \in E_r^{m0}$ and $y_{im}^k \in O_r^m$ associated with Q_{ri}^m. We thus obtain a new selection $Y_s^m = (Y_r^m - \{y_{im}^0\}) \cup \{y_{im}^k\}$. According to Theorem 1, the value of the function LFB_m decreases as Q_i increases and the selection Y_s^m is a feasible solution ($Y_s^m \in R_Y^m$). As has been mentioned above, we select variables satisfying the condition (8). Owing to (9), the reduction in the value of LFB_m is bigger than for any other feasible complementing.

If the set O_r^m is empty, we perform the second phase of the choice operation.

Let D_r^m denote the set including all the links j with residual capacity e_{jr}^m too small to restore virtual paths i ($y_{im}^0 \in E_r^{m0}$) using the backup route π_{jr}^k such that $y_{im}^k \in M_r^m$:

$$D_r^m = \bigcup_{j\in L} \Big\{ j : e_{jr}^m < b_{im}^{kj} Q_i \quad \text{for} \quad y_{im}^0 \in E_r^{m0},\ \ y_{im}^k \in M_r^m \Big\}.$$

Note that D_r^m consists of links that block restoration of virtual paths which have not been restored yet ($y_{im}^0 = 1$). The residual capacity of these links is smaller than the virtual path bandwidth, and for that reason restoration causes violation of the capacity constraint (4) for links belonging to D_r^m. Let

$$\Lambda_{imr}^{kh} = a_{im} \sum_{j\in D_r^m} (b_{im}^{kj} - b_{im}^{hj}). \tag{10}$$

In the second phase of the choice operation we select for complementing variables $y_{im}^k \in E_r^m$ and $y_{im}^h \in M_r^m$ for which Λ_{imr}^{kh} has the maximum value. In this way, we obtain a new selection $Y_s^m = (Y_r^m - \{y_{im}^h\}) \cup \{y_{im}^k\}$. The value of Λ_{imr}^{kh} is the difference of the lengths of the routes of π_{im}^k and π_{im}^h, where the length of a route is defined as the number of links belonging to that route and the set D_r^m. Hence in the new selection Y_s^m the possibility of restoration blocking will be smaller and it will be easier to perform the first phase of the choice operation.

4.3.2. Regulation Operation

The regulation operation is executed if for the current selection Y_r^m the capacity constraint (3) is not satisfied. Let K_r^m be the set including all the links j with the violated capacity constraint

$$K_r^m = \bigcup_{j\in L} \{ j : f_{jr}^m > c_j \}.$$

Let

$$\Psi_{im}^k = a_{im} \sum_{j\in K_r^m} (b_{im}^{kj} - b_{im}^{hj}). \tag{11}$$

The purpose of the regulation operation is to reduce the flows in the links belonging to the set K_r^m. We select variables $y_{im}^k \in E_r^m$ and $y_{im}^h \in M_r^m$ for which Ψ_{imr}^{kh} in (11) has the maximal value. We obtain a new selection $Y_s^m = (Y_r^m - \{y_{im}^h\}) \cup \{y_{im}^k\}$. Note that the value of Ψ_{imr}^{kh} is the difference of the lengths of the routes π_{im}^k and π_{im}^h, where the length of a route is defined as the number of links belonging to that route and the set K_r^m. This means that for the new selection Y_s^m the flows of the links belonging to the set K_r^m will decrease.

4.4. Lower Bounds

In this section we present and compare three lower bounds of the function $LFB_m(Y_r^m)$ for the selection Y_r^m and any successor that can be generated from the selection Y_r^m.

Let f_{jr}^{mu} denote the overall flow of link j for the virtual paths with fixed variables y_{im}^k:

$$f_{jr}^{mu} = \sum_{i=1}^{p} a_{ij} Q_i - \sum_{i=1}^{p} a_{ij} a_{im} Q_i + \sum_{i,k:y_{im}^k \in U_r^m} b_{im}^{kj} Q_i \ \text{ for } j = 1,\dots,l;\ j \neq m.$$

Note that f_{jr}^{mu} consists of two terms. The first one ($\sum_{i=1}^{p} a_{ij} Q_i - \sum_{i=1}^{p} a_{ij} a_{im} Q_i$) is the flow of link j after releasing all virtual paths including the failed link m. The second term ($\sum_{i,k:y_{im}^k \in U_r^m} b_{im}^{kj} Q_i$) is the flow of fixed virtual paths including link j.

4.4.1. Lower Bound LB_{rm}^{LFB1}

In order to calculate the first lower bound LB_{rm}^{LFB1}, we relax some constraints of the problem (5), (6). Let $L(m)$ denote the set including the links leaving the beginning node of link m except the link m:

$$L(m) = \bigcup_{\substack{i\in N \\ i\neq d(m)}} \big\{ \langle o(m), i\rangle : \langle o(m), i\rangle \in L \big\}.$$

We assume that all links in the network, except the links from the set $L(m)$, have infinite capacity. This means that only the links from the set $L(m)$ may block the restored flow after a failure of link m.

Furthermore, we assume that all failed virtual paths may use any possible backup route for restoration. This means that all the links belonging to the set $L(m)$ are the bottleneck of the restoration process. After these relaxations, we can model the problem (5), (6) as a 0-1 knapsack problem. Weights and profits of objects in the knapsack problem are bandwidths of virtual paths. To calculate the lower bound LB_{rm}^{LFB1}, we must find a feasible assignment of virtual path bandwidths that maximizes the total value of the restored flow. The feasible assignment means that the flows of the links belonging to the set $L(m)$ cannot exceed the residual capacity of these links. Having the upper bound of the restored flow, we can calculate the lower bound of the lost flow.

Let P_r^m denote the set containing all the virtual paths for which variables y_{im}^k are not a constantly fixed set in the algorithm, i.e.

$$P_r^m = \bigcup_{i \in P_m} \{i : y_{im}^k \in E_r^m\}.$$

Let $KP(m,r)$ be the optimal solution to the knapsack problem. The lower bound LB_{rm}^{LFB1} can be calculated as follows:

$$LB_{rm}^{LFB1} = \sum_{i:y_{im}^0 \in U_r^m} (y_{im}^0 Q_i) + \sum_{i \in P_r^m} (Q_i) - KP(m,r). \tag{12}$$

The LB_{rm}^{LFB1} is the sum of two terms. The first one is the flow lost after a failure of link m for virtual paths with variables $y_{im}^0 \in U_r^m$ already fixed. In the second term ($\sum_{i \in P_r^m}(Q_i) - KP(m,r)$) we consider only virtual paths belonging to the set P_r^m, and it is a lower bound of the flow, which cannot be restored due to limited resources of the residual capacity.

4.4.2. Lower Bound LB_{rm}^{LFB2}

The knapsack problem is an NP-complete optimization problem (Sysło *et al.*, 1993). Therefore the second lower bound LB_{rm}^{LFB2} is a relaxation of the first lower bound LB_{rm}^{LFB1}. We assume that virtual paths can be restored using the bifurcated multicommodity flow. This means that in the relaxed knapsack problem variables associated with objects can have real values belonging to the interval $[0,1]$. Recall that in the knapsack problem those variables can have only integer values of 0 or 1.

The lower bound LB_{rm}^{LFB2} can be calculated as follows:

$$LB_{rm}^{LFB2} = \sum_{i:y_{im}^0 \in U_r^m} (y_{im}^0 Q_i) + \sum_{i \in P_r^m} (Q_i) - \sum_{i \in L(m)} (c_i - f_{ir}^{mu}). \tag{13}$$

Note that $\sum_{j \in L(m)} (c_j - f_{jr}^{mu})$ is an upper bound of the optimal solution to the knapsack problem.

4.4.3. Lower Bound LB_{rm}^{LFB3}

Let $\hat{\pi}_{im}^k$ denote the part of the route π_{im}^k originating from the node $o(m)$ (the beginning node of link m). We construct a network modelled as a graph $G_r^m = \langle N, L_r^m \rangle$ with link capacities c_r^m, where

$$L_r^m = \bigcup_{j \in L} \{j : j \in \hat{\pi}_{im}^k, y_{im}^k \in M_r^m\},$$

$$c_r^m(j) = \min\left\{ (c_j - f_{jr}^{mu}), \sum_{i \in P_{rj}^m} Q_i \right\}.$$

The set L_r^m includes all the links belonging to the routes $\hat{\pi}_{im}^k$ with variables $y_{im}^k \in M_r^m$. Only links $m \in L_r^m$ can be used for restoration of virtual paths with variables y_{im}^k which have not been fixed. The capacity $c_r^m(j)$ is the minimum of two values. The former is the residual capacity of link j. The latter is the capacity of all paths i for which any backup route $\hat{\pi}_{im}^k$ includes link j. The value of $c_r^m(j)$ limits the capacity of link j to a maximal value that can be used in the restoration process.

Let N_r^m be a set of destination nodes d_i of all the virtual paths belonging to P_r^m:

$$N_r^m = \bigcup_{i \in P_r^m} \{d_i\}.$$

We add to the graph G_r^m a new node u and links to construct a new network modelled as a graph $\hat{G}_r^m = \langle \hat{N}, \hat{L}_r^m \rangle$ with link capacities $\hat{C}_r^m$, where

$$\hat{N} = N \cup \{u\}, \quad \hat{L}_r^m = L_r^m \cup \left(\bigcup_{i \in N_r^m} \{\langle i, u \rangle\} \right),$$

$$\hat{c}_r^m(j) = \begin{cases} c_r^m(j) & \text{for links } j \in L_r^m, \\ \infty & \text{for other links } j. \end{cases}$$

Using the concept of virtual paths, the flow of every path is transmitted along one route. Let us relax this constraint for the graph $\hat{G}_r^m$ and use the bifurcated multicommodity flow. Moreover, we assume that the destination node of every virtual path belonging to the set P_r^m is the node u. Hence the flow of all virtual paths $i \in P_r^m$ can be modelled as one commodity transmitted from the node $o(m)$ (the begging node of the failed link m) to the node u. The value of this commodity is $\sum_{i\in P_r^m}(Q_i)$. Let $MF(m,r)$ denote the maximal flow from the node $o(m)$ to the node u in the graph $\hat{G}_r^m$ with link capacities defined by the set $\hat{C}_r^m$. According to (Ford and Fulkerson, 1969), $MF(m,r)$ is the upper bound of the flow that can be restored. Hence ($\sum_{i\in P_r^m}(Q_i) - MF(m,r)$) is a lower bound of the unrestored flow. The lower bound LB_{rm}^{LFB3} can be calculated as follows:

$$LB_{rm}^{LFB3} = \sum_{i:y_{im}^0\in U_r^m}(y_{im}^0 Q_i) + \sum_{i\in P_r^m}(Q_i) - MF(m,r). \tag{14}$$

Maximal flow algorithms can be found in (Ford and Fulkerson, 1969; Sysło *et al.*, 1993).

4.5. Algorithm

Let $Y_1^m \in R_Y^m$ be the initial solution. We assume that $U_1^m = \emptyset$, $U_1^{mt} = \emptyset$ and $LFB^* = \infty$. Let LB_{rm}^{LFB} denote the lower bound of the current selection Y_r^m. Depending on to which lower bound is applied, we select $LB_{rm}^{LFB} = LB_{rm}^{LFB1}$, $LB_{rm}^{LFB} = LB_{rm}^{LFB2}$ or $LB_{rm}^{LFB} = LB_{rm}^{LFB3}$.

Step 1. (Test step) If there exists a link i such that $f_{ir}^{mu} > c_i$, then go to Step 5. Otherwise, compute LB_{rm}^{LFB}. If $LB_{rm}^{LFB} \geq LFB^*$, then go to Step 5. Otherwise, go to Step 2.

Step 2. (Evaluation step) Compute $LFB(Y_r^m)$ and set M_r^m. If for all links i the condition $f_{ir}^m \leq c_i$ is satisfied, then $M_r^m := M_r^m - U_r^{mt}$. If $LFB(Y_r^m) < LFB^*$, then $LFB^* := LFB(Y_r^m)$. If $LFB^* = 0$, then terminate the algorithm. The set Y^* assigned to the current value of LFB^* is the optimal solution. If $LFB^* > 0$, go to Step 3. If the condition $f_{ir}^m > c_i$ is not satisfied for all links i, then identify a new set K_r^m including all such links. Go to Step 4.

Step 3. (Choice operation) If $M_r^m = \emptyset$, then go to Step 5. Otherwise, find the set O_r^m. Write $O_r^m := O_r^m - U_r^{mt}$. If $O_r^m \neq \emptyset$, then select variables $y_{im}^0 \in E_r^{m0}$, $y_{im}^k \in O_r^m$ with the satisfied condition (9). Generate a new selection Y_s^m (the successor of the current selection Y_r^m) in the following way:

$$Y_s^m := \left(Y_r^m - \left\{y_{im}^0\right\}\right) \cup \left\{y_{im}^k\right\},$$

$$U_s^m := U_r^m \cup \left\{y_{im}^k\right\}, \quad U_s^{mt} := U_r^{mt}.$$

Go to Step 1. If $O_r^m = \emptyset$, then select variables $y_{im}^k \in E_r^m$, $y_{im}^h \in M_r^m$ for which Λ_{imr}^{kh} in (10) has the maximal value. Generate a new selection Y_s^m (the successor of the current selection Y_r^m) in the following way:

$$Y_s^m := \left(Y_r^m - \left\{y_{im}^k\right\}\right) \cup \left\{y_{im}^h\right\},$$

$$U_s^m := U_r^m \cup \left\{y_{im}^h\right\}, \quad U_s^{mt} := U_r^{mt}.$$

Go to Step 1.

Step 4. (Regulation operation) If $M_r^m = \emptyset$, then go to Step 5. Otherwise, select variables $y_{im}^k \in E_r^m$, $y_{im}^h \in M_r^m$ for which Ψ_{imr}^{kh} in (11) has the maximal value. Generate a new selection Y_s^m (the successor of the current selection Y_r^m) in the following way:

$$Y_s^m := \left(Y_r^m - \left\{y_{im}^k\right\}\right) \cup \left\{y_{im}^h\right\},$$

$$U_s^m := U_r^m \cup \left\{y_{im}^h\right\}, \quad U_s^{mt} := U_r^{mt}.$$

Go to Step 1.

Step 5. (Backtracking step) Backtrack to the predecessor Y_p^m of the selection Y_r^m. If the selection Y_r^m has no predecessor, then the algorithm terminates. The selection Y* associated with the current LFB^* is optimal. Otherwise, drop the data for Y_r^m and update the data for Y_p^m as follows: If Y_r^m has been generated from Y_p^m by complementing $y_{im}^h := 1$, $y_{im}^k := 0$, then $U_p^{mt} := U_p^{mt} \cup \left\{y_{im}^h\right\}$. If the backtracking is performed for the (l_i^m-1)-th time by reverse variables of the normal variable y_{im}^k, then

$$U_p^m := U_p^m \cup \left\{y_{im}^k\right\}, \quad U_p^{mt} := U_p^{mt} - \left(\bigcup_{a=1}^{l_i^m}\left\{x_i^a\right\}\right).$$

Go to Step 1.

5. Numerical Examples

We implemented the branch and bound algorithm and the heuristic algorithm based on the FD method in C++ and performed a number of numerical experiments on a wide range of networks. We performed 302 tests over 20 links belonging to 3 different networks.

All tested networks have 10 nodes. They have different numbers of links: 46 (*NET*46), 42 (*NET*42) and 36 (*NET*36). One of the networks (*NET*42) is presented in Fig. 2. The tested links are denoted by thick lines. Singular tests differ with the number of virtual paths using the tested links, virtual paths bandwidths, sets of backup routes and link capacities.

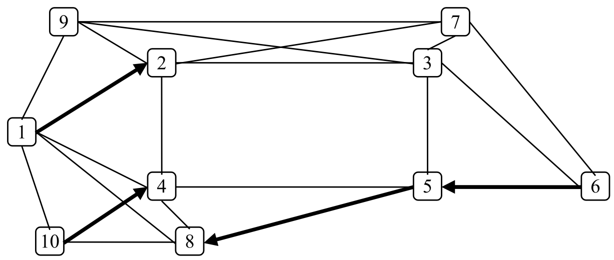

Fig. 2. A graph representing the tested *NET*42.

In order to evaluate the obtained results, we define some parameters. Let $avlu^h$ denote the average link utilization for all links leaving the node h. In order to calculate $avlu^h$, we must sum the flows of all the links leaving the node h and divide the sum by the capacity of all the links leaving the node h. Therefore the $avlu^h$ parameter can be interpreted as a saturation of links leaving the node h. The parameter nd_m is the number of links that leave the node $o(m)$. The tested links have the values of nd_m ranging from 3 to 6.

5.1. Lower Bounds Comparison

We compared the presented lower bounds. Figure 3 shows a comparison of three lower bounds for the arc $\langle 1,2\rangle$ of *NET*42 with parameter nd_m equal to 5. We noticed that for small values of nd_m all lower bounds give comparable results. Calculation of LB_{rm}^{LFB2} is the simplest. In order to obtain LB_{rm}^{LFB1} we must solve an NP-complete knapsack problem. Calculation of the LB_{rm}^{LFB3} requires calculation of a maximal flow.

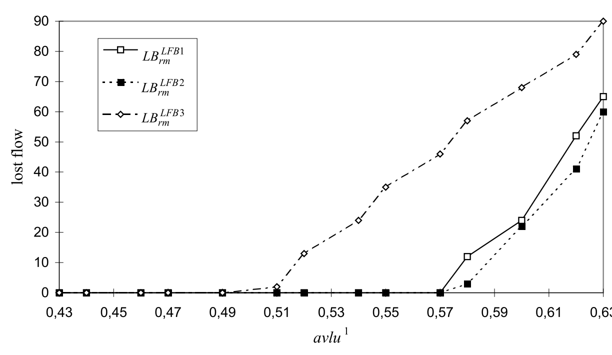

Fig. 3. Comparison of lower bounds for link $\langle 1,2\rangle$ of *NET*42 ($nd_m = 5$).

5.2. Evaluation of Results for the Heuristic Algorithm

In order to evaluate the results of the heuristic algorithm based on the flow deviation method, we compared these results with optimal results given by the branch and bound algorithm.

We performed 302 tests for 20 links of 3 tested networks. Summarizing the results of all tests, the heuristic algorithm gives results worse by 9.4% than the optimal results obtained from the branch and bound algorithm.

Figure 4 shows the percentage difference between the results of the heuristic algorithm and the optimal results plotted against the average utilization of the links leaving the beginning node of the failed link. Six value ranges of the parameter $avlu^h$ are presented.

Note that for increasing values of the parameter $avlu^h$ the results of the heuristic algorithm are getting closer to the optimal solution. For the range of $avlu^h$ from 0.9 to 0.1 the difference is only 0.3%. This is due to the fact that for greater values of $avlu^h$ the absolute value of the function *LFB* increases. Consequently, the relative value of the percentage difference is smaller.

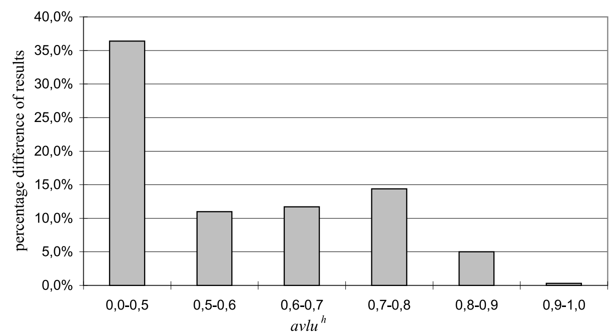

Fig. 4. The percentage difference the between results of the heuristic and the branch and bound algorithm.

Figure 5 shows the percentage difference between the results of the heuristic and the branch and bound algorithms plotted against the number of links leaving the beginning node of the failed link. We can see that the heuris-

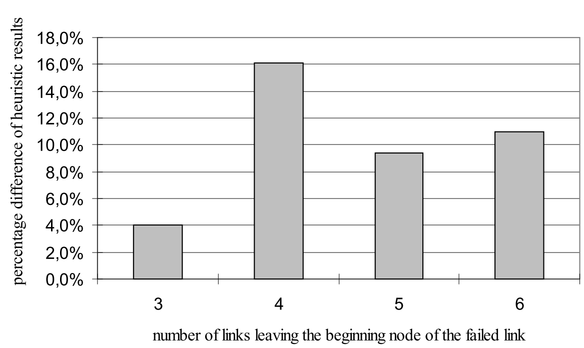

Fig. 5. The percentage difference between the results of the heuristic and the branch and bound algorithm plotted against the number of links leaving the beginning node of the failed link.

tic algorithm gives best results for a case in which the beginning node of the failed link leaves only three links. This means that in this case the flow of the failed link can be restored using only two remaining links. Therefore the number of possible backup routes is relatively small and the size of the solution space is not large. Consequently, the heuristic algorithm can find a solution close to the optimal one.

5.3. The Influence of the Number of the Virtual Paths Using the Failed Link

We also examined the influence of the number of the virtual paths using the failed link on the optimal value of the *LFB* function. We performed numerical experiments for link $\langle 1, 2 \rangle$ of *NET*42. Using the branch and bound algorithm $LFB_m^{\rm opt}$, we computed the optimal value of the *LFB* function for various numbers of virtual paths using the failed link. We assumed that the sum of all virtual path bandwidths is constant, independently of the virtual path number.

Figure 6 presents the corresponding results. Note that more virtual paths can utilize spare resources of the network better, since the path bandwidths are smaller. Therefore the optimal value of the *LFB* function increases with the number of virtual paths. The function *LFN* plotted in the figure represents the lower bound of the *LFB* function. Functions $LFB_m^{\rm opt}$ are piecewise linear due to discrete values of link capacities and virtual path bandwidths. More interesting results can be found in (Walkowiak, 2000a).

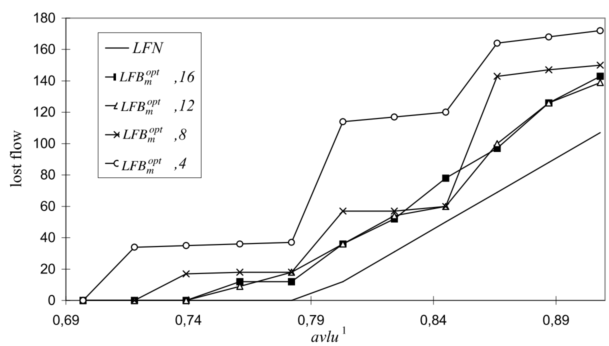

Fig. 6. Optimal value of the *LFB* function for various numbers of virtual paths crossing the link $\langle 1, 2 \rangle$ of *NET*42 plotted against the parameter $avlu^1$.

6. Conclusions

This paper concerns survivability mechanisms used in ATM networks. We introduced basic restoration methods and concentrated on the virtual path protection method using local-destination rerouting. We formulated the combinatorial optimization problem called the Backup Virtual Path Routing (BVPR). This problem is NP-complete. The objective function is a function of the flow lost due to a failure of a single link. To our knowledge this problem has not received much attention in the literature. We developed a new exact algorithm based on the branch and bound method. Numerical experiments showing the most interesting results are included.

Branch and bound algorithms are broadly used to obtain optimal solutions to network optimization problems related to the BVPR problem. Having an optimal solution given by an exact algorithm, we can evaluate the efficiency of heuristic algorithms by comparing their results with optimal ones. Moreover, exact algorithms can be applied in the phase of designing network flows. ATM networks are widely used as backbone networks carrying traffic between many widespread locations. Having estimated bandwidth requirements, we can assign optimal routes to virtual paths and spare significant amounts of money. Last but not least, the developing of exact algorithms helps, in our opinion, to understand the optimization problems under consideration better, and can be much more useful in the construction of effective heuristic algorithms than in the case of the exact ones.

The method proposed in this paper can be applied to provide survivability to an existing ATM network. ATM switches due to special software enable rerouting of virtual paths that are affected by the network failure. This software must be supplied with algorithms calculating routes of backup paths. Algorithms discussed in this work can be applied for that purpose.

In the future, we plan to develop exact and heuristic algorithms for two other rerouting strategies, which have not been considered in this paper, i.e. source-based rerouting and local rerouting. We also want to examine dynamic virtual path assignments using computer simulation methods.

References

Anderson J., Doshi B., Dravida S. and Harshavardhana P. (1994): *Fast restoration of ATM networks*. — IEEE JSAC, Vol. 12, No. 1, pp. 128–138.

Ayanoglu E. and Gitlin R. (1996): *Broadband network restoration*. — IEEE Comm. Mag., Vol. 34, No. 7, pp. 110–119.

Burgina J. and Dorman D. (1991): *Broadband ISDN resource management: The role of virtual paths*. — IEEE Comm. Mag., Vol. 29, No. 9, pp. 44–48.

Chlamatac I., Fargo A. and Zhang T. (1994): *Optimizing the system of virtual paths*. — IEEE/ACM Trans. Networking, Vol. 2, No. 12, pp. 581–587.

Ford L. and Fulkerson D. (1969): *Flows in Networks*. — Warsaw: Polish Scientific Publishers (in Polish).

Fratta L., Gerla M. and Kleinrock L. (1973): *The flow deviation method: An approach to store-and-forward communication network design*. — Networks, Vol. 3, No. 2, pp. 97–133.

Friesen V., Harms J. and Wong J. (1996): *Resource management with virtual paths in ATM networks*. — IEEE Network, Vol. 10, No. 5, pp. 10–20.

Gerstel O., Cidon I. and Zaks S. (1996): *The layout of virtual paths in ATM networks*. — IEEE/ACM Trans. Networking, Vol. 4, No. 12, pp. 873–884.

Goldberg D. (1989): *Genetic Algorithms in Search, Optimization, and Machine Learning*. — Reading: Addison-Wesley.

Guerin R., Ahmadi H. and Nagshineh M. (1991): *Equivalent capacity and its application to bandwidth allocation in high-speed networks*. — IEEE JSAC, Vol. 9, No. 9, pp. 968–981.

Herzberg M., Bye S. and Utano A. (1995): *The hop-limit approach for spare-capacity assignment in survivable networks*. — IEEE/ACM Trans. Networking, Vol. 3, No. 12, pp. 775–784.

Hui J., Gursoy M., Moayeri N. and Yates R. (1991): *A layered broadband switching architecture with physical and virtual path configurations*. — IEEE JSAC, Vol. 9, No. 12, pp. 1416–1426.

Iraschko R., MacGregor M. and Grover W. (1998): *Optimal capacity placement for path restoration in STM or ATM mesh-survivable networks*. — IEEE/ACM Trans. Networking, Vol. 6, No. 6, pp. 325–335.

Kajiyama Y., Tokura N. and Kikuchi K. (1994): *An ATM VP-based self-healing ring*. — IEEE JSAC, Vol. 12, No. 1, pp. 171–178.

Kasprzak A. (1985): *Optimal capacity and non-bifurcated flow assignment in a computer communication network*. — Syst. Sci., Vol. 11, No. 1, pp. 47–58.

Kasprzak A. (1989): *Optimization Algorithms of Flows, Channel Capacity and Topology in Computer Communication Networks*. — Wrocław: Publishing House of the Wrocław Univ. Techn.

Kasprzak A. and Walkowiak K. (2000): *Designing of virtual private networks in the public ATM network environment*. — Proc. Conf. 34th *Modelling and Simulation of Systems*, *MOSIS*, Ostrava, Czech Republic, pp. 111–116.

Kawamura R., Sato K. and Tokizawa I. (1994): *Self-healing ATM networks based on virtual path concept*. — IEEE JSAC, Vol. 12, No. 1, pp. 120–127.

Kawamura R. and Tokizawa I. (1995): *Self-healing virtual path architecture in ATM networks*. — IEEE Comm. Mag., Vol. 33, No. 9, pp. 72–79.

May K., Semal P., Du Y. and Hermann C. (1995): *A fast restoration system for ATM-ring-based LANs*. — IEEE Comm. Mag., Vol. 33, No. 9, pp. 90–98.

Murakami K. and Kim H. (1996): *Virtual path routing for survivable ATM networks*. — IEEE/ACM Trans. Networking, Vol. 4, No. 2, pp. 22–39.

Nederlof L., Struyve K., O'Shea C., Misser H. and Tamayo B. (1995): *End-to-end survivable broadband networks*. — IEEE Comm. Mag., Vol. 33, No. 9, pp. 63–70.

Sato K., Ohta S. and Tokizawa I. (1990): *Broad-band ATM network architecture based on virtual paths*. — IEEE Trans. Comm., Vol. 38, No. 1, pp. 1212–1222.

Sysło M., Deo N. and Kowalik J. (1993): *Discrete Optimization Algorithms*. — Warsaw: Polish Scientific Publishers (in Polish).

Van Landegem T., Vankwikelberge P. and Vanderstraeten H. (1994): *A self-healing ATM network based on multilink principles*. — IEEE JSAC, Vol. 12, No. 1, pp. 139–148.

Veitch P. and Hawker I. (1996): *Administration of restorable virtual path mesh networks*. — IEEE Comm. Mag., Vol. 34, No. 12, pp. 96–101.

Veitch P. and Johnson D. (1997): *ATM network resilience*. — IEEE Network, Vol. 11, No. 5, pp. 26–33.

Walkowiak K. (2000a): *Algorithms for assignment of virtual paths in survivable ATM networks*. — Ph.D. Thesis, Sci. Papers, Division of Systems and Computer Networks of Wrocław Univ. Technol., Series: Preprints 2/00 (in Polish).

Walkowiak K. (2000b): *An exact algorithm for designing backup virtual paths in survivable ATM networks*. — Proc. Conf. 22nd *International Scientific School*, *ISAT*, Szklarska Poręba, pp. 263–271.

Wang W., Tipper D., Jager B. and Mehdi D. (1997): *Fault recover routing in wide area networks*. — Proc. 15th Int. Teletraffic Congress, Washington, USA, pp. 1077–1086.

Received: 19 January 2001
Revised: 14 January 2002
Re-revised: 10 March 2002