Tradeoffs in routing reconfiguration problems

Nathann Cohen, David Coudert, Dorian Mazauric, Napoleão Nepomuceno, Nicolas Nisse

Tradeoffs in routing reconfiguration problems[†]

N. Cohen[1], D. Coudert[1], D. Mazauric[1], N. Nepomuceno[1,2], and N. Nisse[1]

[1]*MASCOTTE, INRIA, I3S, CNRS, Univ. Nice Sophia, France.*
[2]*Universidade Federal do Ceará, Fortaleza, CE, Brazil*

Nous étudions le problème du reroutage d'un ensemble de connexion dans un réseau. Il consiste à passer d'un routage initial (ensemble de chemins reliant des paires de nœuds) à un autre, en traitant séquentiellement chaque connexion. Il est parfois indispensable d'en interrompre temporairement certaines au cours du processus de reconfiguration, ce qui nous amène à étudier les compromis possibles entre deux mesures d'efficacité : le nombre total de connexions interrompues et le nombre maximum de connexions interrompues simultanément. Nous prouvons qu'établir de tels compromis mène à des problèmes NP-complets et difficiles à approcher (APX-difficiles voir non APX). Nous montrons ensuite que de bons compromis sont impossibles en général. Enfin, nous exhibons une classe d'instances de reroutage pour laquelle il est possible de minimiser le nombre de requêtes interrompues simultanément sans "trop" augmenter le nombre total de connexions interrompues. Ces résultats sont obtenus en modélisant ce problème par un jeu à l'aide d'agents mobiles.

Keywords: Graph searching games, process number, routing reconfiguration problem

1 Motivations and Modeling

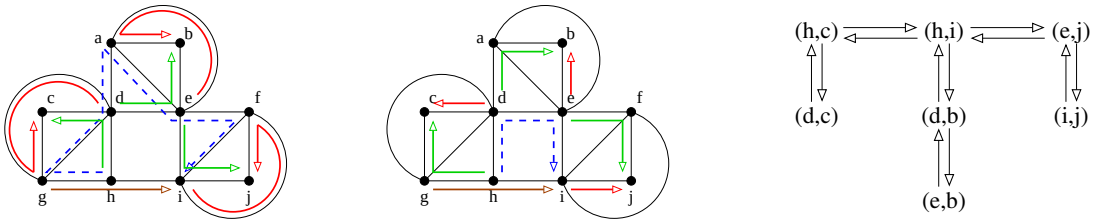
The *routing reconfiguration problem* occurs in connection-oriented networks such as telephone, MPLS, or WDM [3, 4, 12]. In such networks, a connection corresponds to the transmission of a data flow from a source to a destination, and it is usually associated with a capacitated path[‡]. Since links of the network could have to be repaired (and so made unavailable for some time), it might be necessary to change the routing of the connections using them, and so possibly the routing of other connections to free strategic resources. Once a new routing not using the unavailable links is computed, it is not acceptable to stop at once all the connections going on in order to change the routing, since it would result in a very poor quality of service. Instead, it is preferred that each connection first establishes (*sets up*) the new path on which it transmits data, and then stops (*tears down*) the former one. This requires a proper scheduling to avoid conflicts in accessing resources. The *routing reconfiguration problems* consists in finding a scheduling for rerouting each connection minimizing the total number of interruptions, resp., the maximum number of simultaneous interruptions (as cyclic dependencies might force to interrupt connections).

Fig. 1 represents an instance of the routing reconfiguration problem for a network composed of 10 nodes with symmetric arcs. Fig 1(a) corresponds to the initial routing. For example if links (d,g), (a,e), and (f,i) need to be repaired, then we must change the set of routes. Fig 1(b) represents a new routing not using unavailable links. A way to reconfigure the instance depicted in Fig. 1 is to interrupt connections $(h,c),(d,b),(e,j)$, then set up the new paths of all other connections, tear down their old routes, and finally, set up the new paths of connections $(h,c),(d,b),(e,j)$. Another strategy may consist in interrupting the connection (h,i), then sequentially : interrupt connection (h,c), reconfigure (d,c) without interrupting it, set up the new route of (h,c), then reconfigure in the same way requests (d,b) and (e,b), and then connections (e,j) and (i,j). Finally, set up the new route of (h,i). The first (resp. second) strategy interrupts a total of 3 (resp. 4) connections, and at most 3 (resp. 2) requests are simultaneously interrupted.

Following [4, 8], the routing reconfiguration problem can be expressed as a theoretical game on the *dependency digraph* [8]. Given an initial routing of all requests and a new one, the dependency digraph

[†]This work was partially funded by région PACA, ANR AGAPE, and ANR DIMAGREEN.

[‡] In this paper, each arc in the network has a capacity of one, and each connection requires one unit of capacity. Consequently, two different paths cannot share the same arc (a valid assumption in WDM networks).



(a) Old set of routes S_1. (b) New set of routes S_2. (c) Dependency digraph D from S_1 to S_2.

FIG. 1: Instance of the reconfiguration problem consisting of a network with 10 nodes, symmetric arcs, and 8 connections $(h,i),(h,c),(d,c),(d,b),(e,b),(e,j),(i,j),(g,i)$ to be reestablished.

contains one node per connection that must be switched. There is an arc from node u to node v if the initial route of connection v uses resources that are needed by the new route of connection u. Fig. 1(c) represents the dependency digraph of the instance depicted in Fig. 1. Indeed, there is an arc from vertex (d,c) to vertex (h,c), because the new route used by connection (d,c) (Fig. 1(b)) uses resources seized by connection (h,c) in the initial configuration (Fig. 1(a)). Other arcs are built in the same way.

In this paper, we study the routing reconfiguration problem through the *digraph processing* game [4], analogous to the *graph searching* game [7]. This game is defined by operations (or rules), which are very similar to the ones defining the *node search number* [7, 9] of a graph. The goal is to *clear*, or to *process*, all the vertices of a contaminated digraph D through a sequence of the following three operations :

- R_1 Place an agent on a vertex v of D (*interrupt the initial route of the corresponding connection*) ;
- R_2 Remove an agent from a vertex v of D if all its outneighbors are either processed or occupied by an agent, and process v (*establish the new route of the corresponding connection*) ;
- R_3 Process an unoccupied vertex v of D if all its outneighbors are either processed or occupied by an agent (*establish the new route, and then interrupt the old one, of the corresponding connection*).

A sequence of such operations that processes all vertices of D is called a *process strategy* of D. During this process, we use mobile agents that are sequentially placed at and removed from the vertices of D (note that an agent which has been removed from a vertex can be reused later). The number of agents used by a process strategy for D is the maximum number of agents present at the vertices of D during its execution. A vertex is *covered* during a strategy if it is occupied by an agent at some step of the process strategy.

We study this game according to two measures : **(1)** the total number of agents used, and **(2)** the total number of vertices covered by an agent during the processing of D. The reconfiguration problem is equivalent to the clearing of its dependency digraph. In this context, the two measures respectively correspond to the maximum number of simultaneous disruptions during the rerouting of the connections and to the total number of requests disrupted during the whole process. Previous works have studied the problem of minimizing each of these parameters independently [8, 3]. For the instance of Fig 1, the first strategy described before minimizes metric **(2)** and the second achieves the minimum for metric **(1)**. It is easy to show that for this instance, there does not exist a strategy such that 2 agents are used covering at most 3 nodes.

In this paper, we study the tradeoff between both these conflicting objectives, proving that there exists some instances for which minimizing one of these objectives arbitrarily impairs the quality of the solution for the other one. Since we prove that any digraph is the dependency digraph of some instance of the routing reconfiguration problem, it is sufficient to focus our study on the digraph processing game.

2 Definitions and Previous Work

Let D be a n-node directed graph. In the following, a (p,q)-*process strategy* denotes a process strategy for D using at most p agents and covering at most q vertices. When the number of covered vertices is not constrained, we note p-*process strategy* instead of (p,n)-process strategy. Similarly, when the number of agents is not constrained, we note q-*process strategy* instead of (n,q)-process strategy.

The **process number** of a digraph D, $pn(D)$, is the smallest p such that there exists a p-process strategy for D. The digraph D of Fig. 1(c) has $pn(D) = 2$. While digraphs with process number 0, 1, and 2 can

be recognized in polynomial time [5], computing the process number in general digraphs is not in APX and is NP-hard [4]. A distributed polynomial-time algorithm to compute the process number of trees with symmetric arcs has been proposed in [2]. Furthermore, the first heuristic computing the process number of any digraph is described in [3]. The *node search number* and the *pathwidth* (introduced by Robertson and Seymour in [11]) are graph invariants closely related to the notion of process number for undirected graph. The node search number of a graph G, denoted by $sn(G)$, is the smallest p such that rules R_1 and R_2 (R_3 is omitted) are sufficient to process G using at most p agents (see [7, 9]). It has been proved in [6] by Ellis *et al.* that for any graph G, $pw(G) = sn(G) - 1$, and in [4] that $pw(G) \leq pn(G) \leq pw(G) + 1$, where the graph G is considered as a symmetric digraph. Since the problem of determining the pathwidth of a graph is not in APX [10], it can be proved that the computation of all these parameters also is.

The **cover number** of a digraph D, $mfvs(D)$, is the smallest q such that there exists a q-process strategy for D. The digraph of Fig. 1(c) has $mfvs(D) = 3$. Determining the cover number of a digraph D entails the computation of the size of a *minimum feedback vertex set* (MFVS[§]), which is well known to be NP-complete and APX-hard.

In the following, we introduce new tradeoff metrics in order to study the loss one may expect on one parameter while constraining the other. In particular, what is the minimum number of vertices that must be covered by a process strategy using $pn(D)$ agents ? Similarly, what is the minimum number of agents that must be used to process D covering at most $mfvs(D)$ vertices ?

Definition 1 *Given an integer $q \geq mfvs(D)$, we denote by $pn_q(D)$ the minimum p such that a (p,q)-process strategy for D exists. We write $pn_{mfvs}(D)$ instead of $pn_{mfvs(D)}(D)$.*

Definition 2 *Given an integer $p \geq pn(D)$, we denote by $mfvs_p(D)$ the minimum q such that a (p,q)-process strategy for D exists. We write $mfvs_{pn}(D)$ instead of $mfvs_{pn(D)}(D)$.*

Note that $mfvs_{pn}(D)$ is the minimum number of vertices that must be covered by a $pn(D)$-process strategy, and that $pn_{mfvs}(D)$ is the minimum number of agents required by a $mfvs(D)$-process strategy. To illustrate the pertinence of these tradeoff metrics, consider the digraph D of Fig. 1(c). We can verify that there does not exist a $(2,3)$-process strategy for D (a process strategy minimizing both metrics). On the other hand, we have previously described a $(2,4)$-process strategy and a $(3,3)$-process strategy for D. We have : $pn_{mfvs}(D) = 3$ while $pn(D) = 2$, and $mfvs_{pn}(D) = 4$ while $mfvs(D) = 3$. Fig. 2 depicts the shape of the variation of the minimum number q of vertices covered by a p-process strategy for D ($p \geq pn(D)$), i.e. $mfvs_p(D)$ as a function of p. Note that this function is non-increasing and lower bounded by $mfvs(D)$.

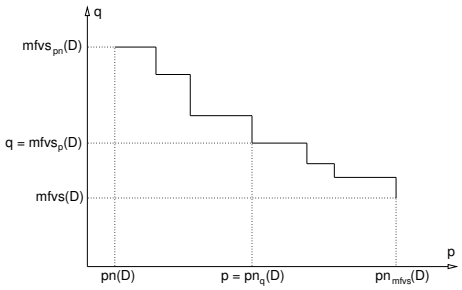


FIG. 2: $mfvs_p(D)$ as a function of p.

3 Our Results

We first prove that any digraph is the dependency digraph of an instance of the routing reconfiguration problem. Thus it shows the relevance of studying these problems through dependency digraph notion. We then prove that computing the process number of a digraph D is not in APX (and thus NP-hard), even if the subset of covered vertices is given, disproving a conjecture of Solano [12]. Other results consist of an analysis of the behaviour of the two measures $\frac{mfvs_{pn}}{mfvs}$ and $\frac{pn_{mfvs}}{pn}$, both in general digraphs and in symmetric digraphs. As mentioned above, in general, no process strategy minimizes both the number of agents and the number of covered vertices (see example of Fig. 1). Hence, we are interested in the loss on one measure when the other is constrained. Due to lack of space, all proofs are omitted and can be found in [1]. The following theorem shows the complexity of estimating the possible tradeoffs.

Theorem 1 *Let $\alpha \in [0,1]$. Given a digraph D, the problem of minimizing $\alpha.pn_{mfvs}(D) + (1-\alpha)pn(D)$ is not in APX, and the problem of minimizing $\alpha.mfvs_{pn}(D) + (1-\alpha)mfvs(D)$ is APX-hard.*

[§] A feedback vertex set of a digraph D is a set of vertices whose removal makes D acyclic

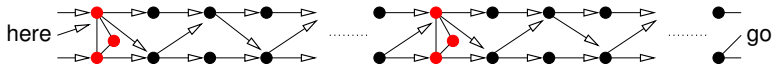


FIG. 3: Digraph D with arbitrarily large $mfvs_{pn}(D)/mfvs(D)$

We also show that computing these values replacing pn_{mfvs} by pn_{mfvs+q}, and $mfvs_{pn}$ by $mfvs_{pn+p}$, $p,q \geq 0$, are also not in APX and APX-hard. Then, we show that no good tradeoffs exist in general.

Theorem 2 *$\frac{pn_{mfvs}}{pn}$ and $\frac{mfvs_{pn}}{mfvs}$ are not bounded, even in the class of bounded process number digraphs.*

Sketch of proof : To show that $pn_{mfvs}(D)/pn(D)$ is not bounded in general, we consider a symmetric star D with n branches of length 2 (see Fig 1(c) for $n = 3$). We have $pn(D) = 2$ and $pn_{mfvs}(D) = n$. Indeed to process D with 2 agents, the central node r and one node of each branch must be covered (a total of $n+1$ nodes). Moreover, a strategy covering at most n nodes needs to put simultaneously n agents at neighbors of r. We generalize this result showing that for any q, $pn_{mfvs+q}(D)/pn(D)$ is not bounded in general. This can be shown using a digraph composed of $q+1$ stars of n branches each. For the second ratio, consider the digraph D depicted in Fig. 3 with $pn(D) = 3$ and $mfvs_{pn}(D) = 4+k$ where k is the number of pairs of nodes between the two symmetric triangles (K_3). Furthermore $mfvs(D) \leq 4$ because removing two nodes per K_3 is sufficient to make the digraph acyclic. We generalize this result proving that, for any p, $mfvs_{pn+p}(D)/mfvs(D)$ is not bounded. The proof is similar, using K_{p+1} instead of K_3. □

However, good tradeoffs may exist in particular classes.

Theorem 3 *$\frac{mfvs_{pn}(D)}{mfvs(D)} \leq pn(D)$ for any symmetric digraph D.*

Furthermore we exhibit a class of symmetric digraphs such that $mfvs_{pn}/mfvs \geq 3-\varepsilon$, for any $\varepsilon > 0$. A symmetric digraph D in this class is such that $mfvs_{pn}(D)/mfvs(D) = (3n+2)/(n+4)$ where $n+1$ is the size of the central clique (see Fig 4). We conjecture that $mfvs_{pn}(D)/mfvs(D) \leq 3$ for any symmetric digraph D.

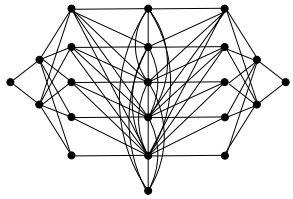


FIG. 4: Sym. digraph D for $n = 5$.

References

[1] N. Cohen, D. Coudert, D. Mazauric, N. Nepomuceno, and N. Nisse. Tradeoffs in process strategy games with application in the WDM reconfiguration problem. In *Fifth International conference on Fun with Algorithms (FUN)*, LNCS, 2010.

[2] D. Coudert, F. Huc, and D. Mazauric. A distributed algorithm for computing and updating the process number of a forest. In *22nd Int. Symposium on Distributed Computing (DISC)*, volume 5218 of *LNCS*, pages 500–501, 2008.

[3] D. Coudert, F. Huc, D. Mazauric, N. Nisse, and J-S. Sereni. Routing reconfiguration/process number : Coping with two classes of services. In *13th Conf. on Optical Network Design and Modeling (ONDM)*. IEEE, 2009.

[4] D. Coudert, S. Perennes, Q.-C. Pham, and J.-S. Sereni. Rerouting requests in WDM networks. In *7ème Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel)*, pages 17–20, May 2005.

[5] D. Coudert and J-S. Sereni. Characterization of graphs and digraphs with small process number. Research Report 6285, INRIA, September 2007. http://hal.inria.fr/inria-00171083/.

[6] J.A. Ellis, I.H. Sudborough, and J.S. Turner. The vertex separation and search number of a graph. *Information and Computation*, 113(1) :50–79, 1994.

[7] F. Fomin and D. Thilikos. An annotated bibliography on guaranteed graph searching. *Theor. Comp. Sci.*, 399(3) :236–245, 2008.

[8] N. Jose and A.K. Somani. Connection rerouting/network reconfiguration. In *4th International Workshop on Design of Reliable Communication Networks (DRCN)*, pages 23–30. IEEE, October 2003.

[9] M. Kirousis and C.H. Papadimitriou. Searching and pebbling. *Theor. Comp. Sci.*, 47(2) :205–218, 1986.

[10] N. Megiddo, S. L. Hakimi, M. R. Garey, D. S. Johnson, and C. H. Papadimitriou. The complexity of searching a graph. *J. Assoc. Comput. Mach.*, 35(1) :18–44, 1988.

[11] N. Robertson and P. D. Seymour. Graph minors. I. Excluding a forest. *J. Comb. Th. Ser. B*, 35(1) :39–61, 1983.

[12] F. Solano. Analyzing two different objectives of the WDM network reconfiguration problem. In *IEEE Global Communications Conference (Globecom)*, Honolulu, Hawai, USA, December 2009.