A Doctor Thesis

博士論文

Extended Multiroute Flow for Multilink Attack Networks

(拡張マルチルートフローとマルチリンク攻撃問題への応用)

by

Jean-François Alexis Baffier

バフィエール　ジャン　フランソワ　アレクシ

Submitted to

the Graduate School of the University of Tokyo

on December 12, 2014

in Partial Fulfillment of the Requirements

for the Degree of Doctor of Information Science and Technology

in Computer Science

Thesis Supervisor: Hiroshi Imai　今井浩

Professor of Computer Science

ABSTRACT

There are many polynomial-time algorithms that solve the maximum network flow problem (max-flow), including the recent results of Orlin (2013) where the max-flow of a network with n nodes and m links is solved in $O(mn)$. However, the amount of flow we get from a solution to those algorithms can drop significantly if some of the links in the network are attacked (or fail). Such a network flow that is robust against attacks is actively studied in various contexts.

This thesis covers a family of natural network flow problems against k attacks qualified as multilink-attack (MLA) Flow: MLA-Robust flow, MLA-Reliable flow, and MLA-Decomposition. Although the MLA-robust flow proves to be intractable, a maximum multiroute flow (MRF) of $k+1$ route provides a $(k+1)$-approximation to the three MLA-network flow variants. An extended version of the MRF is used to solve exactly in polynomial time both MLA-robust flow and MLA-reliable flow for a certain category of networks. Furthermore, it also leads to tighter bounds for the approximation algorithm.

An h-route flow, a MRF of h routes, is a non-negative linear combination of h link-disjoint paths. It was introduced by Kishimoto and Takeuchi (1993) where they extended the max-flow/min-cut duality property to the MRF context and provided an algorithm to compute a max-h-route flow in h iterations of a classical max-flow. Aggarwal and Orlin (2002) showed that a max-h-route flow can be computed in less than h steps for some graphs by using a parametric max-flow where the parameter is a restriction over the links capacities. Aneja, Chandrasekaran, and Nair (2003) considers a first variant of MRF where the number of route is a parameter. The MRF is extended in another direction by considering the number of route as any non-negative real number by Aneja, Chandrasekaran, Kabadi, and Nair (2007).

The extended multiroute flow (EMRF) in this work combines into a modified Eisner-Severance (ES) algorithm the parametric scheme of Aggarwal and Orlin for restricted max-flow and the two extensions concerning the route number by Aneja, Chandrasekaran, and Nair (2003) and Aneja, Chandrasekaran, Kabadai and Nair (2007). In $O(\lambda mn)$ –where λ is the connectivity of the network–, the EMRF computes the whole max-h-route flow function for any non-negative real number of route h. This method outputs more general results than the MRF with a similar complexity when the $h = \Omega(\lambda)$ and solve faster the problem of maximization of h for a given max-h-route flow value.

Robust network flows can be defined in many ways. MLA-network flows were designed by considering the simplest theoretical approach. The general problem was divide into three necessary variants, depending on the information of the attacker. The MLA-robust flow problem is to find the minimum max-flow value among $\binom{m}{k}$ networks obtained by deleting each set of k links. In this setup, we suppose that the attacker first destroy a set of k links, and then we can reroute the flow through the network. The MLA-reliable flow problem is to find a max-flow

of the network such that the flow value is maximum against any set of k link failures, when deleting the corresponding flow to those k links in the original flow. In this case, the attacker knows how much flow is routed on each link, and we cannot reroute the flow after the attack. Finally, the MLA-decomposition problem is quite similar to the reliable one, except that the attacker even knows the routing information at each node.

The multiroute flow is shown to provide a $(k+1)$-approximation algorithm to MLA-robust flow, MLA-reliable flow, and MLA-decomposition. The relation between EMRF and MLA-robust flow can be pushed further. The restricted max-flow described before is a piecewise linear function with at most $\lambda + 1$ parts . If a linear part with slope k exists, then there exists a well chosen h such that the max-h-route flow value is equal to the MLA-robust flow and to the MLA-reliable flow.

The networks can be classified into several categories if they are, for instance, EMRF-solvable or NP-hard. The NP-hard instances are inside a class of networks that possess a flying cutset, that is a MLA-robust cut that is not catch by the parametric scheme of Aggarwal and Orlin. The approximation ratio of MRF algorithm can be improved heuristically by the iterative iterative multiroute flow (IMRF) method, but also directly by the pseudo-tangent method or by application of MLA-robust flow and MLA-robust cut duality.

The success rate of EMRF, and the gap between the upper and lower bounds are evaluated on a set of instances: (almost-)random shape networks, 2D and 3D grids, complete graphs, and complex networks generated by the R-MAT method. Although the distribution of the capacities of the links alone seems to have almost no impact on the success rate, on the contrary the structure of the network has a huge impact as on R-MAT generated instances. Finally, real numbers capacities mapped into integral ones can be as efficient as the originals when the interval sized is big enough –m or m^2 in the experiments.

Acknowledgements

First and foremost, I would like to express my most sincere appreciation and gratitude to Professor Hiroshi Imai for his kind support and guidance all along this doctoral course. I would like to thank him for encouraging my research and for allowing me to grow as a researcher since my internship in his group four years ago. Those advices on both research as well as on my career have been, are, and will continue to be priceless.

My deepest gratitude to the chair of thesis committee, Professor Tetsuo Shibuya, for his patience, kindness and wiseness all along the evaluation process of my work. Also, I would like to sincerely thank the other committee members, Professor Naoki Kobayashi, Professor François Le Gall, Professor Kunihiko Sadakane, and Professor Rui Yamaguchi for their remarks and advices that helped in the improvement of this work.

I would like to express my appreciation to Professor Abdel Lisser for supporting and advising me for so long despite being at the other side of the world, in particular his encouragements and warm welcomes during my few visits to his group. I am grateful to Pr. Philippe Codognet and Pr. Antoine Deza for their wisdom and advices concerning my research and career.

I would like to express my appreciation to my co-authors, Dr. Vorapong Suppakitpaisarn, Mr. Wenkai Dai, and Mr. Hidefumi Hiraishi. A special thanks to Dr. Vorapong Suppakitpaisarn for sharing most of my deadlines during this project.

My deepest thanks to the following group of people I could shared research with, but also teaching or academic experiences related to this research: Pr. David Avis, Pr. Michel Minoux, Pr. François Le Gall, Dr. Hugo Duminil-Copin, Dr. Cédric Bentz, Dr. Florian Richoux, Dr. Hiroyuki Miyata, Dr. Norie Fu, Dr. Arnaud Knippel, Mr. Thibaut Lefebvre, Mr. Thomas Villeneuve, and Mr. Vincent Picard.

My thanks to Imai Laboratory members and alumni, Mami Takahashi, Takahiko Satoh, Takuya Akiba, Yoichi Iwata, Yuki Kawata, Chihiro Komaki, Yosuke Yano, Naoto Ohsaka, Takuto Ikuta, Shuichi Hirahara, Takanori Hayashi, Makoto Soejima, Alonso

Gragera ,Toshihiro Tanuma, Bingkai Lin, Simon Klein, Jérémy Cohen, Keigo Oka, Akira Motoyama, Yuto Hirakuri, Holger Thies, Kentaro Yamamoto, Kenta Takahashi, Hiroyuki Ohta, Junya Fukawa, Chitchanok Chuengsatiansup, Yoshikazu Aoshima, Akihiro Hashikura, Shunichi Matsuda, Ly Nguyen, Mathys Philipon, Frédéric Maillasson, Amaury Josse and others. My thanks to JFLI members, former members and visitors, in particular Pr. David Nowak, Dr. Yves Caniou, Dr. Thomas Silverston, and Dr. Romain Fontugne.

I am grateful to the staff of our department office, and other institution inside the University of Tokyo that supported me in all my procedures, particularly to Kaori Sato for her kindness an her help since my first contact with the Graduate School of Information Science and Technology.

Finally, I would like to thank the Ministry of Education, Culture, Sports, Science and Technology of Japan as the main financial support to this research.

Contents

1 Introduction **1**

1.1 From Graph Theory to Flows in a Network Under Attack 1

1.2 Motivation to Study Multilink-Attack Network Flows 3

1.3 Contribution of This Work 4

1.4 Broadening the Results and Applications 7

1.5 Organization of This Thesis 8

2 Preliminaries **10**

2.1 Fundamentals on Graph Theory, Duality and Complexity Theory 10

2.1.1 Graph Theory 10

2.1.2 Duality 12

2.1.3 Complexity 12

2.2 Flow in Network 13

2.2.1 Maximum Flow 14

2.2.2 Multiroute Flow 18

2.2.3 Other Robust Network Flows 24

2.3 Conclusion 24

3 Multilink-Attack Network Flow **25**

3.1 Statement of MLA-network flow problems . . . 26

3.1.1 MLA-robust Flow . . . 27

3.1.2 MLA-reliable Flow . . . 29

3.1.3 MLA-decomposition . . . 30

3.2 Complexity of the MLA Flows . . . 31

3.2.1 Complexity of MLA-robust Flow . . . 31

3.2.2 Complexity of Related Problems . . . 34

3.3 A first approximation . . . 39

3.3.1 Multiroute flow as an approximation algorithm . . . 39

3.3.2 Approximation Ratio and Edge-Connectivity . . . 44

3.4 Exact Algorithm . . . 46

3.4.1 Restricted Maximum Flow . . . 47

3.4.2 Route-Parametric Multiroute Flow . . . 48

3.4.3 Link-Parametric Multiroute Flow . . . 50

3.4.4 Application of Parametric Multiroute Flows to Solve MLA Problems 52

3.5 Conclusion . . . 54

4 Analysis and Improvement of Algorithms for MLA Flows **55**

4.1 Exact Algorithm . . . 57

4.1.1 Capacity differentiation . . . 57

4.1.2 Skipping steps . . . 58

4.2 Bounds and Flying cut . . . 59

4.3 Completed Multiroute Flow . . . 62

4.3.1 Definitions and bounds 62
4.3.2 Iterative Multiroute Flow 63
4.4 Classification of MLA instances 64
4.4.1 Optimal version using restricted max-flow 66
4.5 Conclusion 66

5 Applications and experiments 67
5.1 Generation of the network instances 67
5.1.1 R-MAT generation method for complex network 69
5.2 Mixed Algo 70
5.3 Comparison between MLA-reliable flow and MLA-decomposition 71
5.4 Effect of Greedy attacks on max-flow and completed-MRF 71
5.5 Success rate of the EMRF algorithm 72
5.5.1 Complete Graph 73
5.5.2 Grid 75
5.5.3 R-MAT 79
5.5.4 Randomly generated network 81
5.5.5 Capacity differentiation and mapping to integer 84
5.6 Bounds and iterative multiroute flow comparison 87
5.7 Conclusion 89

6 Conclusion 90

References 93

List of Figures 100

List of Tables **104**

List of Linear Programs **105**

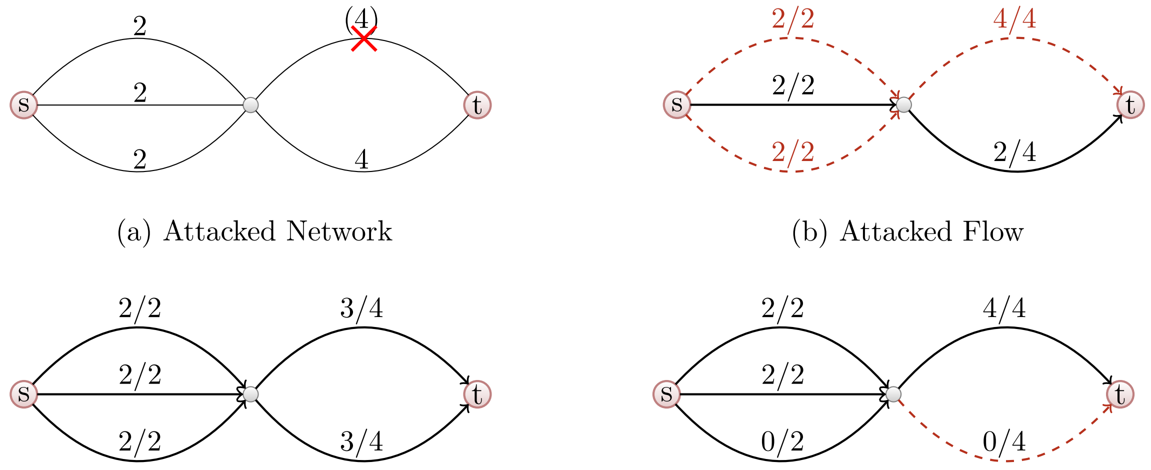
Chapter 1

Introduction

1.1 From Graph Theory to Flows in a Network Under Attack

A graph is a simple structure: a set of nodes and a set of links between the nodes. Despite this simplicity, it can model various abstract problems and real-world situations. A non-exhaustive list includes genealogical tree, matching, scheduling, map coloring, networks. The links of a graph can be weighted to represent diverse properties as capacity, cost, length, duration. A railway network can be represented as a set of links weighted by the length of each railway in between two neighboring stations – the nodes. This last problem, restricted to the soviet railway network, is the origin of the maximum flow notion introduced by Harris and Ross [58]. The first algorithm to solve any maximum flow problem is due to Ford and Fulkerson in [47] and is based on an augmenting path approach. There are many polynomial-time algorithms that solve the maximum network flow problem (max-flow) or its dual counterpart the minimum cut problem (min-cut), including the recent improvement of the push-relabel method by Orlin [81].

However, the amount of flow obtained from a solution computed by those algorithms can drop significantly if some of the links in the network are attacked (or fail). A straightforward approach, that consists in computing all the maximum flows of the network for any set of attacks, is expensive for instances that are not significantly small. Consequently, other kind of approaches to robust network flows are necessary and actively studied in various contexts. Some are quite wide and general, often leading to complex and costly algorithms, or to approximation algorithms with large bounds. Others are very specialized to specific instances or networks structures and cannot be applied to global cases. One tractable method is to construct a flow as a combination of resilient flows. For example, a multiroute flow is a linear combination of tuples of unitary edge disjoint



(a) Attacked Network

(b) Attacked Flow

(c) 1-MLA-reliable flow

(d) 1-MLA-robust flow

Figure 1.1: A network under attack and associated MLA-reliable flow and MLA-robust flow. The dashed lines are a possible impact of the attack on the flow. The attack is signaled by a cross.

paths from the source to the sink and thus are resilient to failures. It was introduced by Kishimoto, Takeuchi, and Kishi [70], where it is shown that a maximum multiroute can be obtained in polynomial time by using classical maximum flow algorithm as a subroutine. They also extend the duality property of max-flow/min-cut to multiroute flow/multiroute cut. In [1], using a parametric max-flow, Aggarwal and Orlin showed that a multiroute flow can be computed in less steps for some graphs by a binary search strategy.

This thesis covers a family of natural network flow problems against k attacks qualified as Multilink-Attack (MLA) Flows: MLA-Robust flow, MLA-Reliable flow, and MLA-Decomposition. These basic problems are defined in the simplest way to cover a wide range of situations. The differences between multilink attack (MLA) flow problems are minimal and based on the information the attacker possesses as described in figure 1.1. The MLA-robust flow problem is to find the minimum max-flow value among $\binom{m}{k}$ networks obtained by deleting each set of k links. In this setup, the attacker first destroy a set of k edges, then the networks can reroute the flow from the source. The MLA-reliable flow problem is to find a max-flow of the network such that the flow value is maximum against any set of k link attacks, when deleting the corresponding flow to those k links in the original flow. In this case, the attacker knows the flow on each link, and it cannot be rerouted after the attack. Finally, the MLA-decomposition problem is quite similar to the reliable one, except that the attacker even knows the routing information at each node. Some of those problems are proved intractable, however they can be approximated by another kind of flow: the multiroute flow.

The analysis of the extended multiroute flow (EMRF) method leads to a characterization necessary to NP-hard instances. Furthermore, the EMRF algorithm is combined with other lower and upper bounds methods, leading to a mixed-MLA flow algorithm running also in polynomial time. Both EMRF and mixed-MLA are then evaluated through various experiments measuring, among others, their speed, efficiency, and bounds tightness. The set of instances generated concerns different kind of instances as randomly generated shape with several capacity distribution, but also 2D and 3D grids, complete networks and complex networks generated by the Recursive MATrix (R-MAT) method.

Unless specified otherwise, the following notations are consistent through the whole thesis:

- A network $G = (V, E, c)$ where V is the set of nodes, E the set of links, and c the capacity function of the links;
- The number of nodes $n = |V|$ and the number of links $m = |E|$;
- A singular pair of nodes on G referred as the source s and the sink t;
- The source-sink link-connectivity λ;
- A number of attack k and a number of routes h.

Also, an instance will generally refers to a network associated with a given number of attacks k.

1.2 Motivation to Study Multilink-Attack Network Flows

Obviously, a solution – or its value – to MLA-robust flow, MLA-reliable flow, or MLA-decomposition has direct application to network flow problems. This strong property is further reinforced by the nature of MLA-robust flow. At the opposite of a multiroute flow, a MLA-robust flow solution is also maximum flow on the attacked network. As such, it is subject to the same various and wide area of application than the classical max-flow. As stated by Goldberg and Tarjan [52]:

> $[\cdots]$, algorithm design techniques and data structures developed to compute maximum flows are useful for other problems as well. Although a special

case of linear programming, the maximum flow problem is general enough so several important problems (such as the maximum bipartite matching problem) reduce to it.

Those MLA flow problems can significantly improve not only the management of existing network structures, but might be used to synthesis robust network. For example, if solutions of both MLA-robust flow and MLA-reliable flow have the same value, then the amount of information an attacker possesses on the network capacities is insignificant.

1.3 Contribution of This Work

The contributions to this work are numerous, and can be considered as an aggregation of small and large results around the field of robust network flows. Many of them have been proposed and accepted in the literature, and will be referred as appropriate. The results genuine to this thesis are presented chapter 3 onwards, and the few shared contributions are signaled and referred to the rightful authors.

Proposed by Baffier and Suppakitpaisarn [9], the base of this work is the formulation of the Multilink-Attack (MLA) problems: MLA-robust flow, MLA-reliable flow. The third kind, MLA-decomposition was introduced later in [10]. All along this research, the analysis of MLA flows led to a classification of the instances in the MLA context. This classification is represented by figure 4.5, page 65.

Although its formulation is simple, MLA-robust flow is proved to be NP-hard. As discussed in section 3.2, other MLA problems are also NP-hard – the proofs come from a joint work with Wenkai Dai and Doctor Vorapong Suppakitpaisarn. The corner stone of this project is the establishment of a correlation between a polynomial time algorithm called Extended Multiroute Flow (EMRF) and MLA-robust flow. The EMRF is a parametric version of the multiroute flow for which the number of route can be non-integer and varies between 1 and λ the source-sink link-connectivity of the network. Both notions of non-integrity and route-parametric were introduced prior to this work by Aneja, Chandrasekaran, and Nair [3], Aneja, Chandrasekaran, Kabadi, and Nair [4] where a method based on the restricted max-flow algorithm in [1] is also provided. In this thesis, the complexity of the algorithm has been improved –in the sense that this results is larger– from $n-1$ to λ where n is the number of nodes. This new result covers also multigraph where the connectivity might be higher than the number of nodes, but it is also more precise when the connectivity λ is much lower than n. If necessary, to remove any uncertainty when the number of attacks is a parameter, a MLA problem can be referred as k-MLA. For example, a 2-MLA-robust flow is an instance of MLA-robust flow when the attackers target two edges. As such, to solve a MLA problem on a network means to solve it for all possible values of k. Usually, an instance refers to a k-MLA instance –a network and a given number of attack.

The major analytical contribution to the MLA flow problems, presented in [10], is the relation between MLA-robust flow and multiroute flow. It consists in the existence of a sufficient condition on an instance for the MLA-robust flow to be tractable. More precisely, if this condition is met, EMRF is an exact polynomial-time algorithm to solve both MLA-robust flow and MLA-reliable flow, and their solutions have the same value. The corresponding class of instances is referred as EMRF-solvable in figure 4.5. At the opposite, the instances –or network instances– that cannot be solved by EMRF are characterized as possessing a *flying cut*. All NP-hard instances possess at least one flying cut.

As shown by Baffier and Suppakitpaisarn [9] and Baffier, Dai, and Suppakitpaisarn [12], when k is the number of attacks, the multiroute flow provides a $(k+1)$-approximation algorithm to MLA-robust flow, MLA-reliable flow, and MLA-decomposition. Nonetheless, the approximation ratio between MLA-robust flow and MLA-reliable flow can be up to $k+1$. Despite the multiroute flow providing a $(k+1)$-approximation of MLA problems, the gap between the value of the solution and the approximation can be relatively low. To compete with a classical max-flow, a multiroute flow needs to be completed to obtain a flow that is maximum under the constraint of the original multiroute flow component. Examples of good (equality) and bad completed multiroute flows are given.

The analysis of the robustness of multiroute flow shows that its resilience to attacks is due to its edge-disjoint components nature. The Iterative Multiroute Flow (IMRF) takes advantage of this property by iteratively completing a multiroute flow by other multiroute flow with a lower number of routes computed on the residual network of the previous multiroute flow. Unfortunately, some instances are a worst solution to attack than a classical max-flow. However, this idea leads to a class of instances of network instances where one maximum-flow can be solution to both MLA-robust flow and MLA-reliable flow for all the value of k, the number of attacks. That class of networks is called best-MLA flow.

The EMRF is improved in several ways. First point is its coverage: the EMRF-solvable class is widened by including a method called capacity-differentiation of the links. It consists in adding a set of relatively small values to the capacity of the links such that all the breaking points hidden by the equality of some links appear. Second improvement concerns the average complexity of the EMRF that can be improved by skipping the computation of some breaking points. A breaking point is actually the intersection of two lines and as such in the event a step of EMRF computes a breaking points, two slopes are obtained at once. At step i, the idea is then to analyze the min-cut corresponding to that step and compute the breaking of degree $i-2$ of this min-cut. Finally, compute a flow restricted by this value. If the results are as expected, then the breakpoint of degree $i-1$ is skipped. A heuristic version can be made by skipping as much steps as possible.

The mixed-MLA flow algorithm combines the EMRF and the analyses of upper and lower bounds of MLA-flow values. The upper bounds are simply based on the duality property of MLA-robust cut/MLA-robust cut, that is that the capacity of any cut is an upper bound to the MLA-robust cut value. Those cut can be chosen in different ways, among others: random search, previously computed MLA-robust cut for different number of attacks, or with some heuristic function. The lower bound is based on the pseudo-tangent property, that is that the value of MLA-reliable flow solution cannot be lower than the projection along the line of slope i passing by any point member of a line of slope bigger than i.

Along the analysis of the MLA flows, the EMRF also improved some application of the multiroute flow itself. For instance, a direct application to the problem of Maximization of Barrier Coverage in Sensor Network presented by DeWitt, Patt, and Shi [34]. This problem version consists in finding the best number of routes h that can guarantee a given max-h-route flow value in the network. The EMRF can output such result efficiently since it computes all the possible h values. Compared to the literature, the algorithm by Suppakitpaisarn and Baffier [86] outputs faster and more general results (non-integrity).

Even if the mixed-MLA algorithm outputs wider results in comparison with the EMRF, no theoretical bound, yet proved to be better than the multiroute flow approximation ratio, have been formulated. Various experiments are done to evaluate the performance on different kind of instances of most of the theoretical contributions listed before. Those experiments have been done using a push-relabel max-flow algorithm –evaluated by Boykov and Kolmogorov [25] to be the fastest in several practical setups– as a subroutine for EMRF method and comparison with classical multiroute flow algorithms. The success of the EMRF algorithm is strangely almost unaffected by the distribution of the capacity of the links alone. However, the shape, associated with links and their capacity distributions seems to have an impact, for instance a bad one, on the R-MAT generated matrices.

1.4 Broadening the Results and Applications

This thesis provides analysis of MLA-network flow problems and some algorithms to solve them or approach their solutions. However, deeper analyses could help in understanding the flying-cutset class of networks. Since the shape of the network seems deeply involved, analytic tools from the spectral theory field might hold some answers.

Nowadays, even by use the best theoretical or practical exact combinatorial algorithm for max-flow as a subroutine, the EMRF and mixed-MLA algorithms are not fast enough to solve large instances that are common. Such instances solutions needs to be approached through smart algorithms, and several methods are possible candidates. The most direct solution is to consider an approximation version of EMRF algorithm based on approximated max-flow subroutine. Considering application of successive approximated max-flow algorithm to restricted networks where the restriction value varies in heuristic manner might help in determine the existence of the different slopes of EMRF. The extension to an approximated mixed-MLA algorithm would then be straightforward.

The problem of large instances can be approached through methods that drop the global complexity inherent to the network size. For instance, a locally distributed version of MLA algorithms restricted to neighboring nodes –neighbours would be nodes at a certain maximum distance– or to clusters would reduce the overall complexity of the algorithm. Moreover, such a solution might apply to dynamic neworks such as mobile communication ones.

Massively parallel architecture is the upcoming future of computer hardware. As shown in the literature as in Caniou, Codognet, Richoux, Diaz, and Abreu [27], Moisan, Gaudreault, and Quimper [79], Xie and Davenport [90], simple or straightforward algorithms often do well on those structures. Also, such methods might allow a formulation of problem a bit deeper and realistic than the classical max-flow. For example, in van Hoeve and Régin [87], Van Hoeve, Pesant, and Rousseau [88] the authors present an adaption of the max-flow problem into a Constraint Satisfaction Problem (CSP).

All of MLA-robust flow, MLA-reliable flow, and MLA-decomposition have direct and not direct application. However, as for original max-flow problem, the field of application is still wide and many application can be formulated. This work introduces the notion of Iterative Multiroute Flow (IMRF) that is one of the candidate to synthesize a network under a set of constraints that resists to attacks in a predictable behavior while restricting the capacities of the link as much as possible.

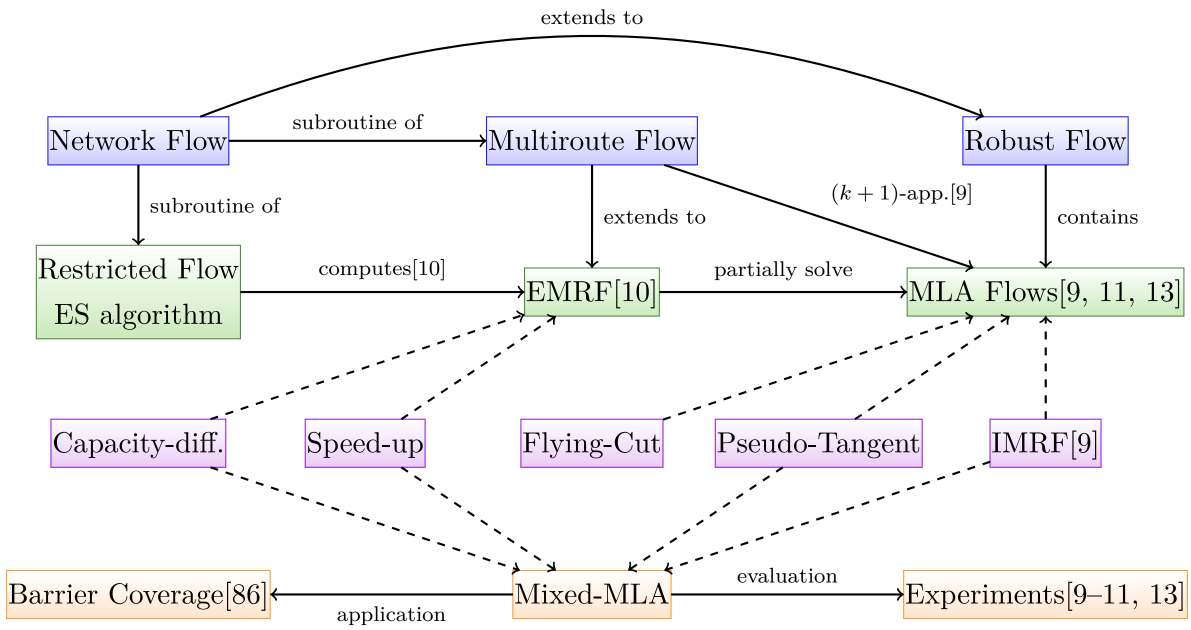


Figure 1.2: Organization of the thesis. Each layer represents a chapter: first layer is chapter 2 in blue, second layer is chapter 3 in green, third layer is chapter 4 in purple, and fourth layer is chapter 5 in orange. The dashed arrows represents improvement or analysis of an item.

1.5 Organization of This Thesis

The content presented in this thesis is composed as described below. Parts of it were published in the literature –or submitted– by the author and co-authors in Baffier and Suppakitpaisarn [9], Baffier, Suppakitpaisarn, Hiraishi, and Imai [10], Baffier, Dai, and Suppakitpaisarn [11, 12], Baffier, Suppakitpaisarn, Hiraishi, and Imai [13], Suppakitpaisarn and Baffier [86]. The flow chart of this thesis is detailed in figure 1.2.

Chapter 2 covers the fundamentals prior to this thesis together with the essential terminology: graph theory, network flow, time complexity, duality in optimization problems, multiroute flow. This part also surveys in a synthetic way studies related to the problem of attacks or failures in network.

The Multilink-Attack (MLA) network flow problems –MLA-robust flow, MLA-reliable flow, and MLA-decomposition– are formally defined in chapter 3. The complexity of MLA-flow problems is discussed along with a first approximation algorithm and an exact polynomial time algorithm for a class of networks or instances.

Various improvement in time and approximation values are designed in chapter 4. A heuristic approximation algorithm referred as Iterative Multiroute Flow (IMRF) is given, including its advantages and shortcomings. The class of graph solved by the EMRF is enlarged and leads to the characterization of a necessary condition for the NP-hardness of any MLA-network flow instance. Additionally, a set of lower and upper bounds are given to further improve the approximation algorithm. Finally, two methods to reduce the number of max-flow iterations on the average are provided, one being an exact algorithm, the other a heuristic one.

In chapter 5, both algorithms along with the lower and upper bounds analysis and the improvement of time computation are joint into the mixed-MLA algorithm. Experiments reviewing the different aspects of this algorithm – speed, approximation ratio, ratio of instances solved– are detailed and analyzed.

Chapter 2

Preliminaries

This chapter gives a basic introduction to most of the fundamental results prior to this thesis together with the essential terminology. Section 2.1 covers the foundations of graph theory along the basic principles of both duality in optimization problems and computational complexity theory. Those fields are vast, therefore the reader will only access here to necessary material in a clear and synthetic way. Section 2.2 covers flows in networks: the classical maximum flow problem and its dual counterpart the minimum cut are studied in section 2.2.1; then in section 2.2.2 the multiroute flow variants with natural robustness extends several of max-flow properties; last, section 2.2.3 explains several other robust network flows alternatives.

2.1 Fundamentals on Graph Theory, Duality and Complexity Theory

2.1.1 Graph Theory

Although the formal notion of graph, as defined in definition 2.1, appeared in the 20th century, it was preceded by the study of several questions which include the Seven Bridges of Königsberg and the Four Color problem. The first reference to the Seven Bridges of Königsberg comes from Euler, in 1736. The city of Königsberg[1] possessed 7 bridges crossing a river, as in section 2.1.1. The inhabitants of the city wondered if it was possible to walk around the city and cross each bridge only once. The answer is “no”, as Euler proved mathematically. The generalization of this property to any

[1]Renamed Kaliningrad in 1946.

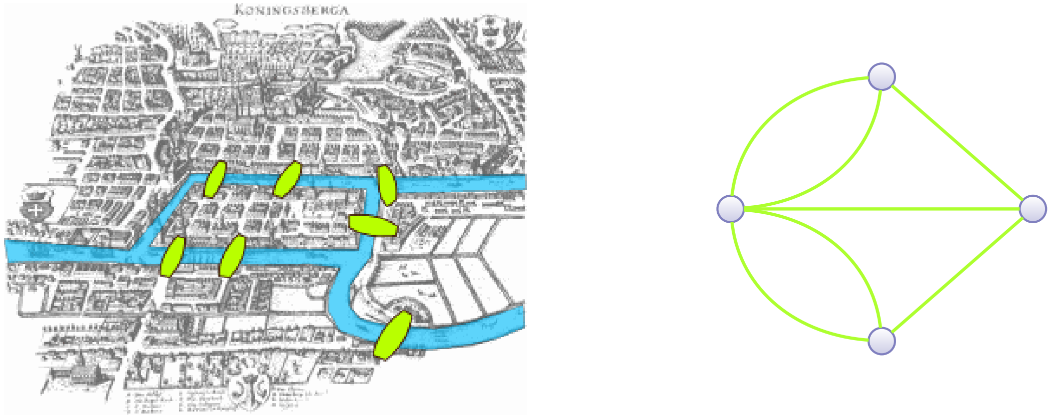


Figure 2.1: The Seven bridge of Königsberg[2] and associated graph

graph is called Eulerian cycle. In the middle of the 19th century, an English student in mathematics, named Francis Guthrie, submitted a conjecture to his mathematics professor Augustus De Morgan that stayed unsolved for over a century. The conjecture consisted in that one could always color an arbitrary map with at most four color such that any neighboring country would have a different color. This conjecture is now called the Four-Color Theorem and was proved in Appel and Haken [6], Appel, Haken, and Koch [7]. This proof also happens to be the first using a computer and was simplified by Robertson, Sanders, Seymour, and Thomas [84]. A proof not relying on the computer has yet to be known. Many other issues can be modeled as graphs, inter alia, data organization, road map, molecule, family tree.

Definition 2.1
A *graph* G is defined by two sets: a non-empty set V of elements called *vertices*, and a set E of *links* between the elements of V. An *undirected graph* refers to a graph with non-oriented links. A link is then also referred as an *edge*. A *directed graph*, or *digraph*, is a graph with oriented links – often called *arcs*.

Unless specified otherwise, the results in this work apply to both orientations. Such a graph structure might not be enough and is extended in several ways in the literature: automaton, Turing machine, network and others. Networks are weighted graphs, often oriented, with associated sink and source nodes as formally defined below.

Definition 2.2
A graph $G = (V, E, c)$ is a *weighted graph* if c is a function that assigned a value to each link. A *network* $G = (V, E, c, s, t)$ is a weighted graph with two singular vertices: a *source* s and a *sink* t. The weight function c is traditionally referred as the *capacity*. A vertex is also called a *node*.

[2]Source: Bogdan Giuşcă - Public domain (PD), based on the image – CC BY-SA 3.0

The capacity in this work is either a real-valued or integer-valued function, always nonnegative. Unless specified otherwise, the results applied to both cases. A network with several sinks and sources, referred as *multi-commodity* case, is outside the scope of this thesis. However, please note that many tractable[3] problems in a single-commodity case become intractable, as described in [33].

2.1.2 Duality and Optimization

The *duality* of an optimization problem implies that this problem may be considered from 2 different perspectives. If the first perspective refers to a maximization problem called *primal*, then the *dual* refers to a minimization problem. The value of a solution to the primal is then always equal or smaller than the one of the dual.

Definition 2.3
The *weak duality* of problem implies that the value of any feasible solution to the primal is then always equal or smaller than the value of any feasible solution to the dual. If this relation is an equality, then *strong duality* holds.

Strong duality holds for any of the dual problems covered in this thesis. Hence, the term *duality* will refer to strong duality. Several example will arise in this work, the first one being the max-flow/min-cut duality stated in theorem 2.1.

An important distinction to make in optimization concerns a solution and its value. Formally, the set of solutions to an optimization problem is referred as the *argument* of the function to optimize. Definition 2.4 is the formal definition of the argument of the minimum: $\arg\min$. The argument of the maximum ($\arg\max$) can be defined in a similar manner.

Definition 2.4
Given any function f defined on $\mathcal{D}$, the argument of the minimum of f can be defined as follows:

$$\arg\min_{x\in\mathcal{D}} f(x) = \{f(x)|\forall y \in \mathcal{D}, f(y) \geqslant f(x)\}.$$

2.1.3 Complexity

In computational complexity theory, algorithms possess different *time-complexity* – as a function of the input length. A *polynomial-time* algorithm is admitted to be good, or

[3]Definitions of tractability is given in section 2.1.3

efficient. In contrast, an *exponential-time* algorithm is considered as slow or inefficient. This distinction was first discussed by Cobham [32] and by Edmonds [41]. A problem is said to be *intractable* if there is no possible polynomial-time algorithm to solve it.

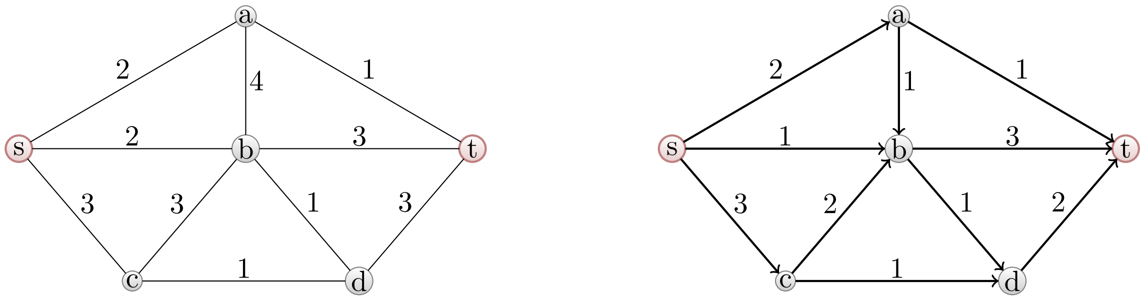
Definition 2.5
The class of problem P regroups all the problems that can be solved in polynomial time in the length of the input by a *deterministic Turing machine*.

Any *decision problem* with positive answers verifiable in polynomial time by a deterministic Turing machine is in the class of problem NP. Alternatively, NP is also the set of decision problems that can be positively answered in polynomial time by a *non-deterministic* Turing machine.
EXP is the class of problem solvable in exponential-time.
The following relation holds: $\mathsf{P} \subseteq \mathsf{NP} \subseteq \mathsf{EXP}$.

Inside complexity classes, some problems are considered harder than others. In NP, those problems are called NP-complete. Any problem in any NP can be reduced in polynomial time to any NP-complete problem. Any problem at least as hard as an NP-complete one is said to be NP-hard. The relation between P and NP might be the most famous open question in complexity theory and is often used has an argument to prove the NP-hardness of some problems. Those problems are intractable, unless the hierarchy collapses – here unless $\mathsf{P} = \mathsf{NP}$. Despite this uncertainty, classifying a problem to be in P or NP gives a theoretical insight on the difficulty that is often relevant for the practical cases. Once a problem is proved to be intractable (unless the hierarchy collapse), a common approach is to look for which classes of instances make the problem tractable. As an example: The 4-coloring of a graph is NP-complete, but the 4-coloring of a planar graph is always true due to the Four-Color Theorem, thus tractable.

2.2 Flow in Network

As formally described in definition 2.2, the notion of (weighted) graphs may extend to networks. A classical network question involves the computation of a maximum flow from a source node to a sink node while respecting the capacities of the links and the flow conservation constraints – only the source node produces flow and only the sink node absorbs it. This section covers the classical maximum flow and minimum cut notions in section 2.2.1, the multiroute flow – a variant of flow with natural robust properties – in section 2.2.2, and other approaches to robust networks in section 2.2.3.



(a) Original network (b) Extracted max-flow

Figure 2.2: An undirected network and its max-flow of value 6.
The following list is a valid decomposition of this max-flow: $s \to a \to t$ with coefficient 1, $s \to a \to b \to t$ with coefficient 1, $s \to b \to d \to t$ with coefficient 1, $s \to c \to b \to t$ with coefficient 2, $s \to c \to d \to t$ with coefficient 1.

2.2.1 Maximum Flow

The maximum flow problem was first formulated by Harris and Ross [58] as a simplified model of the flow in the Soviet railway traffic. Furthermore, it may model any traffic in a distribution system, for example water pipes, but also helps in solving many other problems as task scheduling, stock management, communication network. A flow in a network is subject to two sets of constraints: the capacities of the links and the conservation of the flow.

Definition 2.6
A feasible *flow* of a network $G = (V, E, c)$ is a flow such that the flow entering and the flow leaving any intermediate node are equals. Additionally, the flow on each link is lower than the capacity of this link.
A feasible flow that is bigger than or equal to any other feasible flow is a *maximum flow*. To find such a flow is called the maximum flow problem (max-flow).

The feasibility constraints of a flow are formally written in linear program 2.1. The set $\mathscr{F}$ refers to the set of all s-t flows of G. Figure 2.2 is a max-flow example. Figure 2.2a pictures the original undirected network, and figure 2.2b an extracted max-flow from s to t.

Definition 2.7
Given a network $G = (V, E, c)$ and a feasible flow f on G, the *residual network* is defined as $G' = (V, E, c')$ where $c'(e) = c(e) - f(e)$ for any element of E. The function c' is referred as the *residual flow*.

$$
\begin{aligned}
\max \quad & f && (2.1)\\
\text{s.t.} \quad & 0 \leqslant x_{ij} \leqslant c_{ij}, \quad \text{for } (i,j) \in E, && (2.2)\\
& \sum_{i:(i,k)\in E} x_{ik} - \sum_{j:(k,j)\in E} x_{kj} = \begin{cases} -f & \text{if } k = s \\ f & \text{if } k = t \\ 0 & \text{otherwise.} \end{cases} && (2.3)
\end{aligned}
$$

Linear Program 2.1: Maximum flow

Algorithm	Time complexity
Ford-Fulkerson	$O(mf)$
Edmonds-Karp	$O(nm^2)$
Dinic blocking flow	$O(n^2m)$
Blocking flow with dynamic tree	$O(nm\log(n^2/m))$
Binary blocking flow	$O(m\min(n^{\frac{2}{3}}, \sqrt{m})\log(n^2/m)\log U)$
Push-relabel	$O(n^2m)$
Push-relabel: FIFO vertex selection	$O(n^3)$
Malhotra, Pramodh-Kumar, Maheshwari	$O(n^3)$
Orlin, and King-Rao-Tarjan	$O(mn)$

Table 2.1: Complexity of max-flow algorithms where f is the max-flow value, $n = |V|$ the number of nodes, $m = |E|$ the number of links, and U is the largest capacity.

The max-flow problem can be written as linear program 2.1. If the capacities are integral, then the resulting maximum flow is also integral.

The first algorithm to compute a max-flow was proposed by Ford and Fulkerson [47]. It involves the use of augmenting paths as defined in definition 2.8. Other approaches have been designed over the years, notably the blocking flow algorithm, the push-relabel algorithm, and their extensions.

A detailed survey about the max-flow history is done by Goldberg and Tarjan [52]. The comparison of the computation time of some of the max-flow algorithms is presented in table 2.1.

Definition 2.8
A *path* from a vertex u to a vertex v is an oriented sequence of links such that: if a and

b are two consecutive links, then the end vertex of a is also the origin vertex of b; the first element of the sequence as u as origin; and v is the end vertex of the last link.
In a flow context, an *augmenting path* is path composed of links where the flow value is strictly lower than the capacity. Additionally, $\mathcal{P}$ refers to the set of all paths in the network.

The Ford-Fulkerson algorithm starts with a feasible flow of value 0. As each step, a new feasible flow is obtained by sending a unit of flow through one of the augmenting paths in the network. The algorithm stops when no augmenting path is left and ouputs a maximum flow. This method is improved by Edmonds and Karp [42] and referred as Edmonds-Karp's max-flow algorithm. This shortest augmenting path approach was independently designed by Dinic [37] in the Dinic[4] blocking flow algorithm.

Many algorithms to compute a maximum were designed up to now, often with a better time complexity. The following ones are well-known as deterministic methods:

- the shortest augmenting path method of Edmonds and Karp,
- Dinic blocking flow algorithm,
- the push-relabel method of Goldberg and Tarjan.

Along with the value (on each link) of a flow, one can be interested in the decomposition of this flow along all the paths in $\mathcal{P}$. This decomposition is particularly important in case of attacks, since such a knowledge provides the exact information about which data is lost. Formally, such a decomposition can be defined using elementary flows as in definitions 2.9 and 2.10. An example is given in figure 2.2.

Definition 2.9
An *elementary flow* –or *unitary* flow– is flow of value 1 along one path from the source to the sink.
A flow is then also a non-negative linear combination of such elementary flows, in which the value of the flow on each link does not exceed the associated capacity.

Definition 2.10
Let $G = (V, E, c)$ be a network and f a flow in G. Let H be a function from $\mathcal{P}$ to $\mathbb{R}_{\geqslant 0}$. If $\sum\limits_{P \in \mathcal{P} : e \in P} H(P) \leq c(e)$ for all $e \in E$, H is called a valid *decomposition* of G.

[4] The right spelling is Dinitz, however due to a mistake while translating [37], Dinic was used. In this work, the spelling Dinic will be used to keep coherence.

$$
\begin{aligned}
\max \quad & \sum_{j=1}^{|\mathcal{P}|} p_j && && (2.4)\\
\text{s.t.} \quad & \sum_{j:e_i \in P_j} p_j \leqslant c_i, && \text{for } 1 \leqslant i \leqslant |E|, && (2.5)\\
& p_j \geqslant 0, && \text{for } 1 \leqslant j \leqslant |\mathcal{P}|. && (2.6)
\end{aligned}
$$

Linear Program 2.2: Maximum flow decomposition

Note that, as for any solution to the max-flow problem, the flow on the edges is integral if the capacities are integral. Also, a decomposition, as formulated in linear program 2.2, with integral flow on each path is reachable.

Definition 2.11

A *cut* is a separation of the nodes of G into two subsets. Although being equivalent in practice, a *cutset* design the set of links crossing the cut. The *value* of a cut is the sum of the capacities of the cutset. A cut separating the source s from the sink t is an $s-t$ cut. In the case of a directed network, the value of the cut is the sum of the links going from the side of s to the side of t minus the sum of the links going from the side of t to the side of s.

The set $\mathscr{C}$ refers to the set of all s-t cuts of G. A *minimum-cut* (min-cut) is a cut with the lowest value over all the cut in $\mathscr{C}$. A *minimal-cutset* is a cutset with the minimum size over all the cutset in $\mathscr{C}$.

Follows the max-flow/min-cut duality theorem proved by Ford and Fulkerson [47].

Theorem 2.1

The value of a maximum flow from s to t is equal to the value of a minimum cut separating s from t. If the capacities are integral, the max-flow value is integral.

Unfortunately, the value of a classical max-flow is unreliable in the case of failures or attack in a network, as figure 1.1 depicts. Computing all the possible failures requires an exponential number of max-flow iterations. A simple way to offer a bigger amount of flow in the network is to oversize the capacity of each link in the network as explained by Bley, Grötschel, and Wessäly [23]. However in Schwartz [85], the authors show the cost of maintenance of larger-capacity links can be significantly higher .

For each network, there is a minimum number of attacks such that the source is disconnected from the sink, i.e. there is no s-t path left. This number is called the edge-connectivity of the network. Kawarabayashi and Thorup [63] provide a deterministic global minimum cut of undirected unweighted graph – that is also the connectivity – in near-linear time.

Definition 2.12
The *edge-connectivity* – respectively *node-connectivity* – of a network is the maximum number of edge-disjoint – respectively node-disjoint – paths from the source to sink.

Since this thesis focuses on attacks on the links, from now on, the *connectivity* refers to the edge-connectivity. Note however that, as shown in [64], the attack problem can be extended to nodes by a simple transformation of the network that increases both V and E by n components. Other transformations to fit other networks setup can often be realized in a polynomial increase in the number of components in the original network.

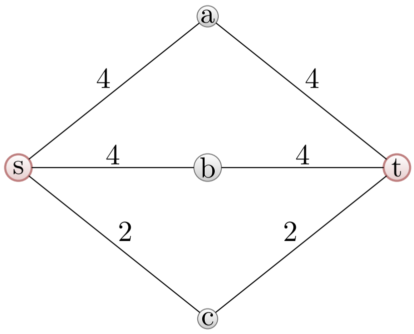
2.2.2 Multiroute Flow

In [70], Kishimoto, Takeuchi, and Kishi introduce a variant of flow with natural robustness properties. This flow is defined using the following idea: a pair of edge-disjoint paths –the paths do not share a link– with elementary flow value guarantee that at least one unit of flow is still carried after a single attack on any edge in the network. This flow variant is called two-route flow and an instance is given in figure 2.3. This concept is then extended to any number of routes in Kishimoto [64], Kishimoto and Takeuchi [68] and referred as multiroute flow. The authors also show that the max-flow/min-cut duality holds for the multiroute flow context and gives a first polynomial time algorithm.

The notion of elementary flow defined in definition 2.9 can be extended to the multiroute flow context and used in definition 2.14 to define the multiroute flow.

Definition 2.13 (elementary h-route flow)
An *elementary h-route flow* is a tuple of h edge-disjoint elementary flows. Each of those elementary flow is composed of path, and those paths do not share any link. Additionally, the flow value on each path is 1.

Definition 2.14 (h-route flow)
A h-route flow –a multiroute flow of h routes– is a non-negative linear combination of elementary h-route flow, in which the flow value on any edge does not exceed this edge capacity. The value of the h-route flow is the sum of the coefficient of the linear combination multiply by h.
A max-h-route flow is a h-route flow such that its value is at least as large as the value of any other h-route flow.



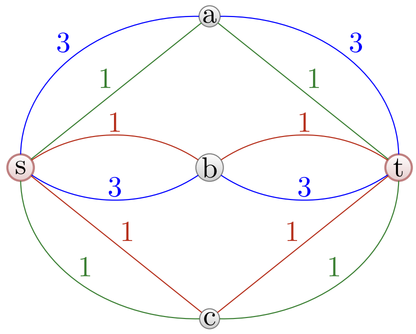


Figure 2.3: Example of a max-2-route flow. This flow is the linear combination of three elementary 2-route flows with coefficient 3 (blue), 1 (green), and 1 (red).

Please note that, sometimes in the literature, the value of the h-route flow is only the sum of the coefficient in the linear combination. All the properties presented or proved in this work are not affected beside an h factor in some formulas.

A max-h-route flow can be characterized as feasible flow with a restriction on the capacity edges as formulated by Aggarwal and Orlin [1]. It can be expressed as the following lemma.

Lemma 2.1 (h-route flow characterization [1])
Suppose that $f \in \mathscr{F}$ is a feasible flow in the network $G = (V, E)$ from node s to node t, and the flow into the sink node t is $h \times f$ for some integer h. Suppose further that $0 \leqslant x_{ij} \leqslant v$ for all arcs $(i, j) \in E$. Then, f is a h-route flow.

The concept of h-elementary flow and h-route can be explained through an example on figure 2.3. Consider the pair of elementary flows $f_a : s \to a \to t$ and $f_b : s \to b \to t$. This is an elementary 2-route from s to t, and thus a feasible 2-route flow. If the elementary flow $f_c : s \to c \to t$ is added to f_a and f_b, this combination becomes an elementary 3-route flow but is not an elementary 2-route flow anymore. However, an equivalent 2-route flow can be obtained by the following linear combination: $\frac{1}{3}(f_a + f_b) + \frac{1}{3}(f_a + f_c) + \frac{1}{3}(f_b + f_c)$. Of course, in this example non of the previous flow is a max-2-route flow. As shown in figure 2.3, the following linear combination is a max-2-route flow: $3 \times (f_a + f_b) + 1 \times (f_a + f_c) + 1 \times (f_b + f_c)$.

In general, a max-h-route flow is not a solution to the max-flow problem, though it is a feasible flow. However, a truncated maximum flow can always be extracted from the residual graph of a maximum multiroute flow. A classical max-flow with resilient properties is then constructed by adding the maximum multiroute flow and the truncated max-flow. This property is proved and further discussed in section 4.3.

By construction, those flows bring a natural robustness to attacks despite not always being optimal. Figure 2.3 is also an example of non-optimality of a max-2-route flow.

The value of the multiroute flow is $10 = 2 \times (3 + 1 + 1)$, by combining the two pairs of edge-disjoint paths of value 1 and the pair of edge-disjoint paths of value 3. Then, the value guaranteed by this max-2-route after one attack is 5. However, the best attack is to target and edge with capacity value 4, and the flow available is then 6. Beside being attacks-resilient, multiroute flow also can help to restore network services after failures or attacks as described by Kar, Kodialam, and Lakshman [61]. In that work, the restoration technique makes use of 2-route flows where one route is the active path and the other the backup path.

A concept dual to the multiroute flow exists and necessitates the definition of the partial sum of ordered capacities of a cutset. However, due to its nature, this concept refers in this thesis to the acronym MLA (Multilink-Attack) discussed from chapter 3 onwards.

Definition 2.15 (MLA-robust capacity)
Given a cut $X \in \mathscr{C}$, let $\{e_0, e_1, \ldots, e_p\}$ be the cut-set of X, where $c(e_i) \geqslant c(e_{i+1})$ for any $0 \leqslant i < p$. For $j > p$, $\alpha_j(X) = 0$. For $0 \leqslant j \leqslant p$, the MLA-robust capacity of X, $\alpha_j(X)$ is defined as

$$\alpha_j(X) = \sum_{i=j}^{p} c(e_i).$$

Definition 2.16 (h-route capacity [64, 68])
The h-route capacity of a cut X is given by

$$\beta_h(X) = \min_{0 \leqslant i \leqslant h-1} \left(\frac{h}{h-i} \cdot \alpha_i(X) \right).$$

Theorem 2.2 (h-route duality)
The value of a max-h-route flow is equal to the h-capacity of a min-h-route cut, $\min\limits_{X \in \mathscr{C}} \beta_h(X)$.

A simpler duality proof than the one in [64, 68] is exposed by Bagchi, Chaudhary, Kolman, and Sgall [16] and a simpler proof of correctness and complexity of multiroute flow algorithm by Du and Chandrasekaran [39].

Beside linear program 2.3, several algorithms were designed to compute the maximum multiroute flow. The single-commodity case can be resolved by three distinct polynomial time algorithms using a max-flow as a subroutine: Kishimoto's, binary-search, and primal-dual algorithms. Any of those solutions applies a max-flow algorithm to a restricted graph defined as follows.

Definition 2.17 (restricted max-flow)
The *restriction* of a network $G = (V, E, c)$ to a non-negative real x is a network $G^x = (V, E, c')$ where $c'(e) = \min(c(e), x)$. The max-flow value on G^x is referred as $\mathcal{F}(x)$.

$$\begin{aligned}
\max \quad & f && (2.7)\\
\text{s.t.} \quad & 0 \leqslant x_{ij} \leqslant c_{ij}, \quad \text{for } (i,j) \in E && (2.8)\\
& \sum_{i:(i,k)\in E} x_{ik} - \sum_{j:(k,j)\in E} x_{kj} = \begin{cases} -f & \text{if } k = s \\ f & \text{if } k = t \\ 0 & \text{otherwise} \end{cases} && (2.9)\\
& x_{ij} \leqslant \frac{f}{h}, \quad \text{for } (i,j) \in E && (2.10)
\end{aligned}$$

Linear Program 2.3: Maximum h-route flow

input : A network graph $G = (V, E, c)$, where $c : E \to \mathbb{R}_{\geqslant 0}$ is the capacity of each edge. An integral route number h. A max-flow subroutine f.
output : The flow on each edge $\mathcal{F} : E \to \mathbb{R}_{\geqslant 0}$, that is updated at each iteration of f.

```
Assign i ← 0; v ← f(G); p ← v/h.
while v < h · p do
    p = (v − i·p)/(h − i);
    v = f(G^p).
end
```

Algorithm 2: General Scheme of Kishimoto's multiroute flow algorithm

The first approach was designed in [64, 68]. The idea behind the algorithm is to determine the optimal solution by solving at most h successive max-flow algorithms where the capacity of the original network is restricted in an inductive way. A general scheme is given in algorithm 2, and the proof of correctness in [64].

The two other methods come from Aggarwal and Orlin [1] where direct applications are given to a generalization of Birkhoff's theorem and to a scheduling problem. The binary-search algorithm for multiroute flow is based on the following properties.

Proposition 2.1 (Multiroute Flow properties [1])

- *The value of a max-h-route flow, as defined in definition 2.14, is*

$$\mathcal{F}(x^*) = \max_{x \geqslant 0}(\mathcal{F}(x) | \mathcal{F}(x) - h \cdot x \geqslant 0),$$

- *this optimal value x^* of x belongs to the interval $[0, U]$ –where U is the highest capacity– and must be an integral multiple of $\frac{1}{i}$ where i lies in $[1..h]$,*

- *the function $\mathcal{F}$ is concave and piecewise linear.*

Those properties enable a binary search for x^* from the interval $[0, U]$ to an interval $[a, b]$ such that $h^2 \cdot (b - a) \leqslant 1$. Thus this algorithm requires at most $O(\log(hU))$ max-flow iterations.

The third and last method is a classical primal-dual approach: the idea is to start with a pair of primal and dual solutions –flow and cut– and browse the search-space till a feasible pair with equal values is discovered. This method finds an optimal pair in at most h step. It is worth noticing that this approach may lead to usable results when the search space is too large to systematically obtain an optimal solution.

Finally, an experimental comparison of the speed of the three different approaches is done by Guerriero and Mancini [54], along with an overview of each algorithm. The experiments are done on a large set of randomly generated oriented instances with several topologies and various distribution of the arc capacities. Kishimoto's original algorithm is faster on the average on fully random networks and 3D grid random networks; the binary-search multiroute flow has the best performances on grid random networks; the primal-dual approach outperforms the two others in fully random networks and 3D grid random networks with capacity range of [1..1000] and certain values of the number of routes h.

As discussed by Aggarwal and Orlin [1], Birkhoff's theorem [22] may be generalized to matrices that are not doubly stochastic by using multiroute flow. The same work shows how to apply this flow to a machine scheduling problem.

Kishimoto [64] also studies the decomposition of a multiroute flow into a linear combination of h-edge-disjoint paths.

Theorem 2.3
There exists a polynomial-time algorithm that can decompose a max-h-route flow solution into a linear combination of $O(m^2)$ h-edge-disjoint paths.

This result was further improved by Aggarwal and Orlin [1] to a combination of $O(m)$ h-edge-disjoint paths.

The multiroute flow problem can be extended to a wider range by considering either a non-integral or parametric number of route h. First approach was discussed by Aneja, Chandrasekaran, Kabadi, and Nair [4] and is formalized in definition 2.18 and lemma 2.2. The constraint (2.10) in linear program 2.3 is preserved when h is a strictly non-negative real number as formalized in definition 2.18. The parametric approach is schemed by Aneja, Chandrasekaran, and Nair [3] as a possible application to a method to find the all pairs parametric minimum cuts as an extension of Gomory-Hu cut-trees [53]. However, the details of this parametric approach will be presented in chapter 3 in a method combining both a non-integral and parametric number of routes. The piecewise-linearity of the restricted flow $\mathcal{F}$ is first proved by Aggarwal and Orlin [1] where the authors give an upper-bound of m^2U line segments. This bound is further improved to $n-1$ in [3]. However, this bound is based on the fact that the source-edge connectivity of a network is at most $(n-1)$. As mentioned in chapter 3, the number of breakpoints is exactly the source-edge connectivity λ, even in the case of multigraph where λ can be bigger than $n-1$. It also provide a more accurate complexity when $\lambda \ll n$.

The formulation by Aneja, Chandrasekaran, Kabadi, and Nair [4] directly extends lemma 2.1 originally formulated by Aggarwal and Orlin [1].

Definition 2.18
For $h \in \mathbb{R}^+$, a flow F with value v is an h-route flow if $F(e) \leqslant \frac{v}{h}$ for all $e \in E$.

Lemma 2.2
For real number $h \leqslant \lambda$, a max-h-route flow value is equal to v^* if $v^* > 0$ and the max-flow value of $G^{\frac{v^*}{h}}$ is equal to v^*.

Another parametric extension is characterized by the variation of an edge capacity. In Diallo, Gueye, and Berthomé [35, 36] the authors study the impact of this variable edge on the computation of the all pair max-2-route flow through the use of 2-cut-tree, the extension of Gomory-Hu trees to the 2-route flow context. The study of Baffier [8] provides a polynomial time algorithm to compute a link-parametric multiroute flow for any number of route and analyses on the multilink-parametric case. An exponential time algorithm can be derived from this multiparametric case analysis.

A multi-commodity version of the multiroute flow is studied by Chandrasekaran, Nair, Aneja, and Kabadi [29], Chuzhoy, Makarychev, Vijayaraghavan, and Zhou [31], and Kolman and Scheideler [72]. The multi-commodity case for classical max-flow is an intractable problem, and this still true in the multiroute flow context for any number of routes.

The dual problem of multiroute flow, the multiroute cut, is generally formulated as finding the minimum multiroute cut between all pairs of nodes of the network. As mentioned by Chekuri and Khanna [30], 2-route cuts are of importance when it comes to feedback problems in undirected networks. The authors considers the following problems under the 2-route cut context: single-source multiple-sink, multiway-cut, multicut, sparsest cut.

2.2.3 Other Robust Network Flow

Since the introduction of network problems, various robust network flow problem have been formulated. These include several extensions of the multiroute flow introduced in section 2.2.2. In Kishimoto [65], Kishimoto and Takeuchi [69] the δ-reliable flow is defined as a combination of δ-reliable channels. If such a channel with flow f has a link or node failure, the flow left cannot be less than $\delta \times f$. The multiroute flow is extended in another way by Kishimoto [66, 67] where the flow has to be balanced in a parametric way on the node and the edges. A balanced flow, as defined by Zimmermann [91], has natural robust properties.

In Bagchi, Chaudhary, Scheideler, and Kolman [14], Bagchi, Chaudhary, Goodrich, and Xu [15], Bagchi, Chaudhary, and Kolman [17] the usage of k edge disjoint paths against a specific type of attacks is shown by the k-Edge-Disjoint Path problem, k-Disjoint-Flow problem and k-Edge-Disjoint Path Coloring problem. One might considers a robust optimization approach that leads to a general conception of the attacked link problem as the robust maximum flow with polyhedral uncertainty sets by Minoux [78].

The question of the flow in a network concerned by random failures –or blinded attacks– is also relevant, as introduced by Aneja and Nair [5]. It has also been studied, for instance, by Lin [76, 77] as a flow with stochastic link failures. Further random or distributed scenarios are studied in various context by Bettstetter and Hartmann [21] and Kobayashi and Otsuki [71].

2.3 Conclusion

Fault-tolerance is one of the most important factors in designing networks, therefore many theoretical and practical approaches to failures and attacks in a network exist. However, some are either too general and complex, either too specific. Chapter 3 introduces a simple, yet meaningful formulation of flow problems in a network under attacks.

Chapter 3

Multilink-Attack Network Flow

A necessary clarification to a meaningful approach of the problem leans on the differentiation between two point-of-views of the network. *Attacker* refers to an entity with the goal to minimize the remaining amount of flow in the attacked network. The goal of the *defender* lies in maximizing this remaining flow. The most efficient attacks are equivalent to the worst failures, and blinded attacks are equivalent to random failures. As observed by Albert, Jeong, and Barabási [2], the latter case is not a major drawback in complex networks that are common nowadays. Some insights on the random failures scenario have been discussed in section 2.2.3. This research focuses on unblinded attacks.

The formulation of the flow in a network under attacks is strongly dependent on the amount of information available to the attacker. This work considers a family of fundamental flow problems against k attacks qualified as Multilink-Attack (MLA) Flows and formally defined in section 3.1. When the attacker only knows the structure of the network – nodes, links, and capacities – the problem is named MLA-robust flow. In this setup, the attacker first destroys a set of k links, and then the defender can reroute the flow through the network. The max-flow/min-cut duality is also extended to MLA-robust context. The MLA-reliable flow problem is to find a max-flow of the network such that the flow value is maximum against any set of k link failures, when deleting the corresponding flow to those k links in the original flow. In this case, the attacker knows how much is routed on each link, and the defender cannot reroute the flow after the attacks. The maximum amount of flow left after k attacks is called MLA-effectiveness. A MLA-reliable flow solution is a flow with the maximum MLA-effectiveness. Finally, the MLA-decomposition problem is similar to the reliable one, except the fact that the attacker also knows the routing information at each node. As such, an MLA-effectiveness of a decomposition can be derived.

Despite being simple to define, MLA-robust flow, MLA-reliable flow, and MLA-decomposition are difficult to compute as discussed in section 3.2. MLA-robust flow optimization problem, for instance, is NP-hard.

The value of the different flows are compared in section 3.3. The value of a max-$(k+1)$-route flow is lower than its MLA-effectiveness but also the values of solutions to MLA-decomposition, MLA-reliable flow, and MLA-robust flow. Furthermore, the MLA-robust flow value is lower than $(k+1)$ times the max-$(k+1)$-route flow value. Thus, a solution to the max-$(k+1)$-route flow is a $(k+1)$-approximation to the three MLA flow problems, and this value is reached for some instances. However, this bound can be improved for non-extreme cases.

The dualities of MLA-robust flow/MLA-robust cut and maximum multiroute flow/minimum multiroute cut lead to a possible relation between those two problems. Indeed, as shown in section 3.4, a condition on the shape of a parametric multiroute is sufficient to guarantee the computation of MLA-robust flow in polynomial time. The analysis is as follows. A route-parametric multiroute-flow with non-integer route values –called extended multiroute flow (EMRF) in this work– can be extracted from a parametric flow in a way similar as by Aggarwal and Orlin [1]. The parametric max-flow function is piecewise linear with at most $\lambda + 1$ parts – where λ is the source-sink connectivity of the network. If a linear part with slope k exists, then there exists a well chosen h such that the h-route flow value is equal to the MLA-robust flow value and the MLA-reliable flow value. This result can be extended to the link-parametric multiroute flow where the parameter is the capacity of a virtual link added between the source and the sink. The function can be computed in polynomial time as shown by Baffier [8].

3.1 Statement of MLA-network Flow Problems

The three MLA-network flows discussed above cover cases where the attacker knows some information on the network. This section gives fundamental definitions and properties of MLA-robust flow, MLA-reliable flow and MLA-decomposition. A distinction about the type of instances that are considered is necessary. Two types of instances can be defined. In the first case, a network coupled with a given integer k forms a k-MLA instance. In the second case, the number of attacks is considered as a variable, and the network itself is the only input to the MLA problems. From now on, an instance will generally design a couple (G, k) and the second case will be referred as a network instance. The definition in this section supposes that an instance $(G = (V, E, c, s, t), k)$ is given.

Both MLA-robust flow and MLA-reliable flow are introduced by Baffier, Suppakitpaisarn, Hiraishi, and Imai [10] and MLA-decomposition by Baffier, Dai, and Suppakitpaisarn [12].

3.1.1 MLA-robust Flow

The MLA-robust flow variant corresponds to the case of an attacker knowing exactly the shape of the network, i.e. the nodes, the links and the capacities. Unlike the two other kinds of MLA-network flows, the attacker has no knowledge of the flow values on the links, even less its decomposition. The MLA-robust flow is a maximum flow that can be routed from the source to the sink after any set of k attacks, as formally defined below. The problem can be considered as a special case of robust maximum flow with polyhedral uncertainty (R-MAX-FLOW-KCU), which is proposed and shown to be NP-hard by Minoux [78].

Definition 3.1 (MLA-robust flow[10])
Given a set of links $S \subset E$, the value ϕ_S is defined as the value of the max-flow of the network $(V, E \setminus S)$. Let $S^* = \underset{S \subseteq E : |S| = k}{\arg\min}\ \phi_S$. The MLA-robust flow is the max-flow of the network $(V, E \setminus S^*)$.

This definition leads to an exponential time exact algorithm through a direct enumeration of all the different cut S of size $m - k$. This algorithm is used on small instances in chapter 5.

The MLA-robust capacity, introduced in definition 2.15, is a partial sum of the ordered capacities of a cutset. By construction, it gives direct information on the robustness of a cut against attacks. Naturally, a minimum cut problem can be associated to it, and leads to the duality of MLA-robust flow and MLA-robust cut.

Definition 3.2 (MLA-robust cut[10])
The MLA-robust cut problem is to find a s-t cut with the minimum MLA-robust capacity. The set of solutions to this problem is formally expressed by $\underset{X \in \mathscr{C}}{\arg\min}\, \alpha_k(X)$. Denote the value of any of the solution as $\alpha_k^* = \min_{X \in \mathscr{C}} \alpha_k(X)$.

Lemma 3.1 (Duality[10])
The MLA-robust capacity of a MLA-robust cut is equal to the value of a MLA-robust flow.

Proof. For any $S \subseteq E$, let μ_S be the capacity of a minimum cut of $(V, E \setminus S)$, and $\mu_S(X)$ the capacity in $(V, E \setminus S)$ of any cut X.
Let f be the value of the MLA-robust flow.

By duality of max-flow and min-cut, it follows that

$$f = \min_{S \subseteq E : |S| = k} \phi_S = \min_{S \subseteq E : |S| = k} \mu_S = \min_{S \subseteq E : |S| = k} \left(\min_{X \in \mathscr{C}} \mu_S(X) \right).$$

Let consider a set of links S^* and a cut X^* such that

$$S^* \in \arg\min_{S \subseteq E : |S| = k} \left(\min_{X \in \mathscr{C}} \mu_S(X) \right) \text{ and } X^* \in \arg\min_{X \in \mathscr{C}} \left(\min_{S \subseteq E : |S| = k} \mu_S(X) \right).$$

Let the cut-set of X^* be $E^* = \{e_0, \ldots, e_p\}$, where $c(e_0) \geq c(e_1) \geq \cdots \geq c(e_p)$. Hence $\mu_{S^*}(X^*) = \sum\limits_{e \in E^* \setminus S^*} c(e)$. Two cases follow, $E^* \subseteq S^*$ and $E^* \nsubseteq S^*$.

Case 1
When $E^* \subseteq S^*$,

$$\mu_{S^*}(X^*) = \alpha_k(X^*) = 0 \Rightarrow f = \min_{X^* \in \mathscr{C}} \alpha_k(X^*) = 0.$$

Case 2
Next, consider the case when $E^* \nsubseteq S^*$.
Assume $S^* \nsubseteq E^*$. Let e, e' be links such that $e \in S^* \setminus E^*$ and $e' \in E^* \setminus S^*$.
If $S' := S^* \cup \{e'\} - \{e\}$, then

$$\mu_{S'}(X^*) = \mu_{S^*}(X^*) - c(e') < \mu_{S^*}(X^*).$$

This contradicts the assumption that $\mu_{S^*}(X^*)$ is the minimum value among $X \in \mathscr{C}$ and $S \subseteq E$ for $|S| = k$.
Hence, $S^* \subseteq E^*$.

To minimize $\mu_{S^*}(X^*) = \sum_{e \in E^* \setminus S^*} c(e)$, it is obvious that S^* must be a set of links in E^* with the largest capacity, $\{e_0, \ldots, e_{k-1}\}$. Thus,

$$\mu_{S^*}(X^*) = \alpha_k(X^*) \Rightarrow f = \min_{X \in \mathscr{C}} \alpha_k(X).$$

□

$$
\begin{aligned}
\max \quad & h && (3.1)\\
\text{s.t.} \quad & 0 \leqslant x_{ij} \leqslant c_{ij}, \quad \text{for } (i,j) \in E && (3.2)\\
& \sum_{i:(i,k)\in E} x_{ik} - \sum_{j:(k,j)\in E} x_{kj} = \begin{cases} -f & \text{if } k = s \\ f & \text{if } k = t \\ 0 & \text{otherwise} \end{cases} && (3.3)\\
& \sum_{(i,j)\in C} x_{ij} - \sum_{(i,j)\in S} x_{ij} \geqslant h, \quad \text{for } C \in \mathcal{C} \text{ and } S \in \mathcal{S}(C) && (3.4)
\end{aligned}
$$

Linear Program 3.1: MLA-reliable

3.1.2 MLA-reliable Flow

Additionally to the knowledge of the shape of the network, suppose that the attacker also knows the amount of flow going through each link and that the defender cannot reroute the flow after the attack. Or, at least, there is a certain duration for which the defender cannot reroute the flow. For instance, such a setup might correspond to the amount of information that can be expected to be routed before the recovery of the network. This value is called MLA-effectiveness and formalized in definition 3.3.

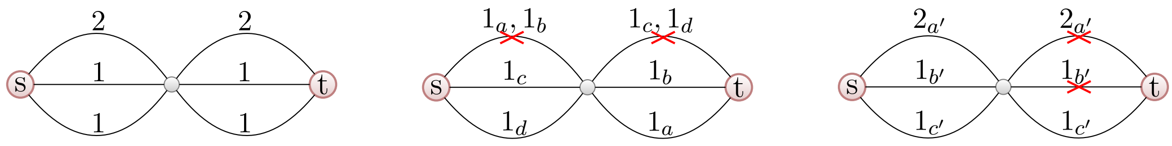
The MLA-reliable flow problem is to find a max-flow of the network such that the flow value is maximum against any set of k link failures, when deleting the corresponding flow to those k links in the original flow. Thus, a MLA-reliable flow solution is a flow with the maximum MLA-effectiveness. The problem can be considered as a special case of the minimax problem to combat link attacks, which is proposed and solved heuristically by Lee, Misra, and Rubenstein [75]. The case when $k = 1$ can be solved by the method for δ-reliable flow proposed by Kishimoto and Takeuchi [69].

Definition 3.3 (MLA-effectiveness[10])
Let F be a valid flow, and let ϕ_S^F be a max-flow of a network $G' = (V, E \setminus S, f)$, where $f(e)$ is a value of the flow F on link e. The MLA-effectiveness of F against k attacks is defined as

$$\gamma_k(F) = \min_{S \subseteq E : |S| = k} \phi_S^F.$$

Definition 3.4 (MLA-reliable flow[10])
The MLA-reliable flow can be defined as

$$F^* = \arg\max_{F \in \mathscr{F}} \gamma_k(F).$$



(a) Attacked Flow (b) Worst Decomposition (c) MLA-decomposition

Figure 3.1: A flow (a) under attack and two possible decomposition of the flow. The attacks are signaled by a cross. (b) is the worst solution, attacker can drop the flow to 0 by attacking the two top edges. (c) is optimal with value 1, and thus a solution to MLA-decomposition.

$$\max \quad \sum_{j=1}^{|\mathcal{P}|} p_j - f' \tag{3.5}$$

$$\text{s.t.} \quad \sum_{j:e_i \in P_j} p_j \leqslant c_i, \qquad \text{for } 1 \leqslant i \leqslant |E|, \tag{3.6}$$

$$\sum_{j:P_j \cap S} p_j \leqslant f', \qquad \text{for } S \in \mathcal{S}, \tag{3.7}$$

$$p_j \geqslant 0, \qquad \text{for } 1 \leqslant j \leqslant |\mathcal{P}|. \tag{3.8}$$

Linear Program 3.2: MLA-decomposition

Let $\gamma_k^* = \max_{F \in \mathscr{F}} \gamma_k(F)$ denotes the value of F^*.

The MLA-reliable flow problem can be expressed as linear program 3.1.

3.1.3 MLA-decomposition

Denote $\mathcal{P}$ the family of all simple s-t paths, i.e. paths without cycle. The MLA-decomposition problem can be defined as follows.

Definition 3.5 (MLA-effectiveness of a decomposition[12])
Let $\mathcal{S} := \{S : S \subseteq E \text{ and } |S| = k\}$, and H be a valid decomposition of G, as in definition 2.10. The MLA-effectiveness of H, $\psi_k(H)$, can be defined as

$$\psi_k(H) := \min_{S \in \mathcal{S}} \sum_{P \in \mathcal{P}: P \cap S = \emptyset} H(P).$$

Definition 3.6 (MLA-decomposition[12])
Let $\mathcal{H}$ be the family of all valid decompositions. The set of solutions D^* to MLA-decomposition are defined as $D^* = \arg\max_{H \in \mathcal{H}} \psi_k(H)$. The value of a solution of D^* is noted $\psi_k^* = \max_{H \in \mathcal{H}} \psi_k(H)$.

The MLA-decomposition problems can be represented as the linear program 3.2. Figure 3.1c is a 2-MLA-decomposition.

3.2 Complexity of the Multilink-Attack Network Flow Problems

Although the complexity of some MLA problems are yet to be established, several cases are analyzed or presented here. The results presented in section 3.2.2 are joint work of Baffier, Dai, and Suppakitpaisarn [11, 12], and can be considered outside the scope of contribution of this thesis. More precisely those proofs are the work of Wenkai Dai and Vorapong Suppakitpaisarn. However, those properties are helpful to interpret experimental results about MLA-network flows in section 5.3. On the opposite, the results concerning the complexity of MLA-robust flow in section 3.2.1 are original to this work.

3.2.1 Complexity of MLA-robust Flow

The proof of NP-hardness of the MLA-robust flow is based on the proof of NP-hardness of R-MAX-FLOW-KCU by [78] where the author reduces minimum cut into bound sets (MIN-CUT-INTO BOUNDED SETS) into R-MAX-FLOW-KCU. Moreover, as mentioned in section 3.1.1, MLA-robust flow is a subproblem of R-MAX-FLOW-KCU, and thus a good approximation algorithm to NP-hard instances of MLA-robust flow might be extended to an approximation algorithm to R-MAX-FLOW-KCU.

Definition 3.7 (MIN-CUT-INTO BOUNDED SETS)
Given $G = (V, E)$ with $|V| = 2n$, MIN-CUT-INTO BOUNDED SETS is the problem to find a cut $(S, V \setminus S)$ in which $|S| = n$, and the capacity of the cut is not larger than any other cut $(S', V \setminus S')$ with $|S'| = n$. In the unweighted case, it means that $(S, V \setminus S)$ is minimal –the cardinality of $(S, V \setminus S)$ is not larger than the cardinality of any other cut– among the cuts separating V into two equally sized subset.

Proposition 3.1 (Garey and Johnson [49])
MIN-CUT-INTO BOUNDED SETS is NP*-hard, even for the unweighted case.*

Theorem 3.1
MLA-robust flow is NP*-hard.*

Proof. Given an unweighted graph instance of MIN-CUT-INTO BOUNDED SETS $G = (V, E)$ such that $|V| = 2n$, consider an instance of MLA-robust flow with $k{=}n$ and $G' = (V', E')$. V' and E' can be defined in the following way.

- Let α, β be positive integers, and ε be a real number such that $\frac{1}{\varepsilon} \in \mathbb{Z}^+$. It is also required that α, β, and $\frac{1}{\varepsilon}$ grow in polynomial rate of n.
- $V' = V \cup \{s, t\}$.
- $E' = E_s \cup E_t \cup E_{copy}$.
  - E_s are edges from s to $i \in V$. For each node $i \in V$, there are $1 + \frac{1}{\varepsilon}$ edges connecting s and i. One of them has a capacity equal to α, and the others have capacities equal to ε.
  - E_t are edges from $i \in V$ to t. For each node $i \in V$, there are $\frac{\beta}{\varepsilon}$ edges connecting i and t. All of them has a capacity equal to ε.
  - Lastly, the set E_{copy} contains $\frac{1}{\varepsilon}$ copies of all edges in E. Similar to the edge in E_t, all edges has a capacity equal to ε.

First argument is that there exists a bijection from an arbitrary cut of $G, (A, V \setminus A)$, to an s-t cut of G', $(A', V' \setminus A')$, such that $A' \supset \{s\}$ and $V' \setminus A' \supset \{t\}$. That is because $(A \cup \{s\}, V \setminus (A \cup \{t\}))$ is clearly the s-t cut of G', and $(A' \setminus \{s\}, (V' \setminus A' \cup \{t\}))$ is a cut in G. The remaining part of the proof will aim to show that the values α, β, ε can be selected such that the MLA-robust cut of graph G' is corresponding to the MIN-CUT-INTO BOUNDED SETS in that bijection. Hence, MIN-CUT-INTO BOUNDED SETS can be solved in polynomial time, if MLA-robust flow can be solved in polynomial time.

Let consider a cut $(A', V' \setminus A')$ of G' such that $A' \supset \{s\}$ and $V' \setminus A' \supset \{t\}$.
Denote $n_1(A') := |A' \setminus \{s\}|$ and $n_2(A') := 2n - n_1(A')$. Let $A := A' \setminus \{s\}$.
The cutset of $(A', V' \setminus A')$ in G' contains $\frac{n_1(A') \cdot \beta}{\varepsilon}$ edges in E_t, $n_2(A') \cdot (\frac{1}{\varepsilon} + 1)$ edges in E_s, and $\frac{1}{\varepsilon}$ copies of cutset of $(A, V \setminus A)$ in G.

Let $c(A')$ be the capacity of the cut $(A, V \setminus A)$. If $n \leqslant n_2(A')$, it follows that the capacity remains after attacks is

$$n_1 \cdot \beta + n_2(A') + (n_2(A') - n)\alpha + c(A').$$

Otherwise, the capacity remaining after attacks is

$$n_1 \cdot \beta + n_2(A') + c(A') - (n - n_2(A'))\varepsilon.$$

Let this remaining capacity be referred as $w(A')$.

Let $\alpha = n^3$, $\beta = 2n^2$, and $\varepsilon = \frac{1}{n^3}$.
It will be shown that any cut $(B, V' \setminus B)$ such that $n_2(B) \neq n$, implies $w(B) > w(A')$ for any A' such that $n_2(A') = n$.

Case 1
If $n_2(B) > n$,

$$\begin{aligned}
w(B) = && n_1(B) \cdot \beta + n_2(B) + (n_2(B) - n)\alpha + c(B) \\
= && (2n - n_2(B)) \cdot 2n^2 + n_2(B) + (n_2(B) - n)n^3 + c(B) \\
= && (n^3 - 2n^2 + 1) \cdot n_2(B) + 4n^3 - n^4 + c(B)
\end{aligned}$$

Assume $n_2(A') = n$, it follows that

$$w(A') = (n^3 - 2n^2 + 1) \cdot n + 4n^3 - n^4 + c(A')$$

Since $(n^3 - 2n^2 + 1) \geqslant 0$ for any $n \in \mathbb{N}$ and $0 \leqslant c(B), c(A') \leqslant n$

$$\begin{aligned}
w(B) - w(A') = && (n^3 - 2n^2 + 1) \cdot n_2(B) + 4n^3 - n^4 + c(B) \\
&& - \left[(n^3 - 2n^2 + 1) \cdot n + 4n^3 - n^4 + c(A')\right] \\
\geqslant && (n^3 - 2n^2 + 1) + c(B) - c(A') \\
\geqslant && (n^3 - 2n^2 + 1) - n
\end{aligned}$$

For all $n \geqslant 3$, it follows that $w(B) - w(A') > 0$.

Case 2
Next, consider the case when $n_2(B) < n$.

$$\begin{aligned}
w(B) = && n_1(B)\beta + n_2(B) + c(B) - (n - n_2(B)) \cdot \varepsilon \\
= && (2n - n^2(B))2n^2 + n_2(B) + c(B) - \frac{n - n_2(B)}{n_3} \\
= && (\frac{1}{n^3} - 2n^2 + 1) \cdot n_2(B) + 4n^3 - \frac{1}{n^2} + c(B)
\end{aligned}$$

Since $(\frac{1}{n^3} - 2n^2 + 1) \leqslant 0$ for all $n \in \mathbb{N}$ and $0 \leqslant c(B), c(A') \leqslant n$,

$$\begin{aligned} w(B) - w(A') = & \quad (\frac{1}{n^3} - 2n^2 + 1) \cdot n_2(B) + 4n^3 - \frac{1}{n^2} + c(B) - 2n^3 - n - c(A') \\ \geqslant & \quad (\frac{1}{n^3} - 2n^2 + 1)(n-1) + 2n^3 - n - \frac{1}{n^2} + c(B) - c(A') \\ \geqslant & \quad 2n^2 - 1 - \frac{1}{n^3} + c(B) - c(A') \\ \geqslant & \quad 2n^2 - n - 1 - \frac{1}{n^3} \end{aligned}$$

Hence $w(B) - w(A') > 0$, for all $n \geqslant 2$.

Thus it can be shown that $w(B) - w(A') > 0$ if $n_2(B) \neq n$ and $n_2(A') = n$. Hence, the cut that minimize the function w is one of the cut in which $n_2(A') = n$. Note that those cuts are corresponding to the cut $(A, V \setminus A)$ in graph G such that $|A| = n$ as required for the MIN-CUT-INTO BOUNDED SETS.
Let consider the value of $w(A')$ for such A'. The previous discussion implies that $w(A') = (n^3 - 2n^2 + 1) \cdot n + 4n^3 - n^4 + c(A') = 2n^3 + n + c(A')$. Hence, the cut that minimize $w(A')$ also minimized $c(A')$, the capacity of the corresponding cut in G.
It follows from there that the solution of MLA-robust flow for $n \geqslant 3$ is also the solution of MIN-CUT-INTO BOUNDED SETS.

Thus, for $n \geqslant 3$, MLA-robust flow is NP-hard. □

Corollary 3.1
By duality, MLA-robust cut is also NP*-hard.*

3.2.2 Complexity of Related Problems

The results of theorem 3.2 are based on a reduction of k-independent set to MLA-decomposition-attack, as detailed by Baffier, Dai, and Suppakitpaisarn [11].

Definition 3.8 (k-independent set)
Given a graph $G = (V, E)$. Decide if there is a set $S \subseteq V$ such that $|S| = k$ and there is no link between any two nodes in S.

Definition 3.9 (MLA-decomposition-attack[12])
Given a valid decomposition $F = (V, E, H)$ and a real number η. Decide if there is a set $S \subseteq E$ such that $|S| = k$ and the MLA-effectiveness of H, $\psi_k(H) := \sum_{P \in \mathcal{P} : P \cap S = \emptyset} H(P)$, is less than or equal to η.

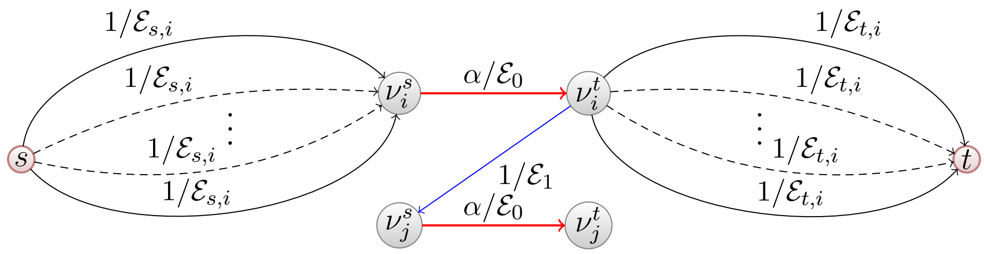


Figure 3.2: Proof of the NP-hardness of the MLA-decomposition-attack. Illustration, from Baffier et al. [12], of the transformation from given edge (v_i, v_j) with $v_i, v_j \in V$ into the decomposition instance. The label on arc shows the flow value and edge type, which are seperated by "/". Some arcs of (ν_j^s, ν_j^t) are omitted.

Theorem 3.2 (By Wenkai Dai and Vorapong Suppakitpaisarn in [12])
MLA-decomposition-attack is NP*-hard.*

Proof by Wenkai Dai and Vorapong Suppakitpaisarn in [12]. We give a reduction from the k-independent set problem on directed graph, which remains in NP-complete as well. As the given flow is a directed graph, we need the direction of arcs to imply the directions of flows. Given an instance of k-independent set $G = (V, E), k$, we can construct an instance of our problem $(V', E', H), k', \eta$ as follows (illustrated in figure 3.2). Firstly, we set a constant α which has $\alpha > |V'|^2$. Let $V = \{v_1, \ldots, v_{|V|}\}$, and for each $v_i \in V$ we use $d_{\text{out}}(v_i)$ and $d_{\text{in}}(v_i)$ to indicate the out-degree and in-degree of v_i in G respectively. We construct a set of nodes $\mathcal{V} \subset V'$, where for each $1 \le i \le |V|$ there are two nodes ν_i^s and ν_i^t in $\mathcal{V}$. Then, the set of nodes V' in the constructed instance is given as $V' := \mathcal{V} \cup \{s, t\}$, where s and t are the source and sink in V'.

To define a set of directed edges, we denote

$$\mathcal{E}_0 := \left\{(\nu_i^s, \nu_i^t) : 1 \le i \le |V|\right\},$$

$$\mathcal{E}_1 := \left\{(\nu_i^t, \nu_j^s) : (v_i, v_j) \in E\right\},$$

for each $v_i \in V$, we set two sets of directed edges

$$\mathcal{E}_{s,i} = \{e_j : e_j := (s, \nu_i^s) \text{ for } 1 \le j \le \alpha - d_{\text{in}}(v_i)\},$$

$$\mathcal{E}_{t,i} = \left\{e_j : e_j := (\nu_i^t, t) \text{ for } 1 \le j \le \alpha - d_{\text{out}}(v_i)\right\}.$$

The set of edges of our construction, E', is $\mathcal{E}_{\text{temp}} \,\dot\cup\, \mathcal{E}_0 \,\dot\cup\, \mathcal{E}_1$ with $\mathcal{E}_{\text{temp}} := \dot{\bigcup}_{1 \le i \le |V|} (\mathcal{E}_{s,i} \dot{\bigcup} \mathcal{E}_{t,i})$. Here, we simply use multiple edges in our construction, but the corresponding simple graph can be constructed by adding node for each paralleling edge.

Let E_i and E'_i be a set of incoming and outgoing edges of a node v_i respectively. To construct our flow decomposition, let $I_i : E_i \rightarrow \{1, \ldots, |E_i|\}$ be a function such that $I_i(e) \neq I_i(e')$ if $e \neq e'$. Similarly, let $O_i : E'_i \rightarrow \{1, \ldots, |E'_i|\}$ be a function such that $O_i(e) \neq O_i(e')$ if $e \neq e'$. Also, we denote $\alpha - d_{\text{in}}(v_i) - d_{\text{out}}(v_i)$ as α_i.

For any $1 \leq i \leq |V|$, we can construct a set of paths $\mathcal{P}_i$ as follows. For $1 \leq j \leq \alpha_i$, we denote a path $\{s, \nu_i^s, \nu_i^t, t\}$ which uses edges e_j in $\mathcal{E}_{s,i}$ and $\mathcal{E}_{t,i}$ as $P_{i,j}$, and $\mathcal{P}_i := \{P_{i,j} : 1 \leq j \leq \alpha_i\}$.

For all $e := (v_i, v_j)$, we denote a path $\{s, \nu_i^s, \nu_i^t, \nu_j^s, \nu_j^t, t\}$ which uses an edge $e_{\alpha_i + O_i(e)}$ in $\mathcal{E}_{s,i}$ and an edge $e_{\alpha_i + I_i(e)}$ in $\mathcal{E}_{t,i}$ as P_e. We refer to $\{P_e : e \in E\}$ as $\mathcal{P}_0$.

Our given decomposition H can be defined as follows. $H(P) := 1$ if $P \in \bigcup_{i=0}^{|V|} \mathcal{P}_i$, and $H(P) := 0$ otherwise. We note that all paths with $H(P) = 1$ do not share any edge in $\mathcal{E}_{s,i}$ and $\mathcal{E}_{t,i}$.

Finally, we set $k' := k$, $\beta := \sum_{1 \leq i \leq |V|} (\alpha - d_{\text{out}}(v_i))$ and $\eta := \beta - (k' * \alpha)$. It is trivial to notice our construction can be finished in polynomial time.

By this construction, we can easily confirm that the flow H is a valid decomposition with a flow value equals to β. It can be easily seen that all optimal choices for that edge attack is a subset of $\mathcal{E}_0$ as $\alpha \gg k'$. Suppose that the attackers choose to attack the edge set $S' := \{(\nu_i^s, \nu_i^t) : i \in J\}$ when $J_{S'} := \{i_1, \ldots, i_k\}$. We know that the remaining flow on the decomposition H is $\sum\limits_{1 \leq i \leq |V| : i \notin J} \alpha_i + |E_{S'}|$, when $E_{S'} := \{(v_i, v_j) \in E : i, j \notin J\}$. That is $\beta - k\alpha + |E'_{S'}|$, when $E'_{S'} := \{(v_i, v_j) \in E : i, j \in J\}$.

By that, the smallest remaining flow is no more than $\eta = \beta - k\alpha$, if and only if there exists a set S' with $|E'_{S'}| = 0$. That set is k-independent set. The effectiveness of the decomposition H is no more than η, if and only if there exists a k-independent set in G. □

Theorem 3.3 (Proof by Wenkai Dai and Vorapong Suppakitpaisarn in [11])
Given a flow network F and integer k. Calculating k-MLA-Effectiveness of F is NP*-hard.*

Proof by Wenkai Dai and Vorapong Suppakitpaisarn in [11]. We give a reduction from the k-clique set problem. Given an instance of k-clique set $G = (V, E), k$, we can construct an instance of our problem $(V', E', F), k', \eta$ as follows (illustrated in figure 3.3). First, we set one constant α to a constant much larger than $2(|E| + |V|)$. Let $V = \{v_1, \ldots, v_{|V|}\}$. We construct a set of nodes $\mathcal{V} \subset V'$, where for each vertex $v_i \in V$ there are two nodes

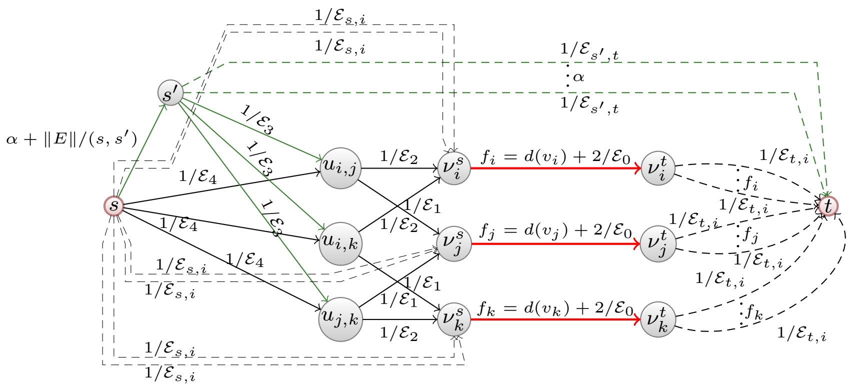


Figure 3.3: Proof of the NP-hardness of the MLA-effectiveness. Illustration of the transformation from a 3-clique. The label on arc shows the flow value and edge type, which are separated by "/". The edge type is defined in Appendix, $d(v)$ is the degree of vertex v in the clique, and $\alpha \gg 2|V| + 2|E|$.

ν_i^s and ν_i^t in $\mathcal{V}$. And for each $e = (v_i, v_j) \in E$, there is a node $u_{i,j}$ in a set of nodes $\mathcal{U}$. Then, the set of nodes V' in the constructed instance is given as $V' := \mathcal{V} \uplus \mathcal{U} \uplus \{s, t, s'\}$, where s and t are the source and sink in V'. Hereafter, we use $d(v)$ to denote the degree of node $v \in V$ in the given graph G. For the simplicity of this proof, we use multiple edges in our construction. The simple graph can be constructed by adding one node for each paralleling edge.

We define the sets of directed edges as followings:

$$\mathcal{E}_0 := \left\{(\nu_i^s, \nu_i^t) : 1 \le i \le |V|\right\},$$

$$\mathcal{E}_1 := \left\{(u_{i,j}, \nu_j^s) : (v_i, v_j) \in E\right\},$$

$$\mathcal{E}_2 := \left\{(u_{i,j}, \nu_i^s) : (v_i, v_j) \in E\right\},$$

$$\mathcal{E}_3 := \left\{(s', u_{i,j}) : (v_i, v_j) \in E\right\},$$

$$\mathcal{E}_4 := \left\{(s, u_{i,j}) : (v_i, v_j) \in E\right\}.$$

For each $v_i \in V$, we set two sets of directed edges

$$\mathcal{E}_{s,i} := \{e_j : e_j := (s, \nu_i^s) \text{ for } 1 \le j \le 2\},$$

$$\mathcal{E}_{t,i} := \left\{e_j : e_j := (\nu_i^t, t) \text{ for } 1 \le j \le d(v_i) + 2\right\}.$$

In addition, there are one edge $e_{s,s'} = (s, s')$, and a set of α multiple edges between nodes s' and t,

$$\mathcal{E}_{s',t} := \{e_j : e_j := (s', t) \text{ for } 1 \le j \le \alpha\}.$$

The set of all edges in our construction is

$$E' := \mathcal{E}_{\text{temp}} \biguplus_{j\in[0,4]} \mathcal{E}_j \biguplus \mathcal{E}_{s',t} \biguplus \{e_{s,s'}\},$$

where $\mathcal{E}_{\text{temp}} := \biguplus_{i\in[1,|V|]} (\mathcal{E}_{s,i} \biguplus \mathcal{E}_{t,i})$. For each edge $e \in \mathcal{E}_0$, we give the flow value as $F(e) = d(v_i) + 2$, where v_i is the node in V corresponding to $e = (\nu_i^s, \nu_i^t)$. Moreover, we set $F(e_{s,s'}) := \alpha + |E|$. For all other edges, we set $F(e) = 1$. Finally, we set $k' := k + 1$ and

$$\eta := \mathcal{F} - \alpha - |E| - k(k-1)/2 - 2k,$$

where $\mathcal{F} := \alpha + 2\,|V| + 2\,|E|$ is the original flow value before the attack. It is trivial to notice that our construction can be finished in polynomial time.

By comparing the flow values of this construction, it is trivial to know the edge (s, s') will always be attacked because of its large flow value $\alpha + |E|$. All remaining attacks should only be included in the set $\mathcal{E}_0$. Suppose that there is a k-clique $G_c \subseteq G$. After attackers remove (s, s') and the edge $(\nu_i^s, \nu_i^t) \in \mathcal{E}_0$ corresponding to all $v_i \in G_c$, only $|V| - k$ edges are remained in $\mathcal{E}_0$. Each of them can have flow 2 supplied through two multiple edges (s, ν_i^s). We can have a remaining flow $2\,|V| - 2k$ by this supply.

Beside that supply, each $u_{i,j}$ also has flow 1 supplied through $(s, u_{i,j})$. It can be easily seen that the supply at the node $u_{i,j}$ can be delivered to the sink node t, unless both (ν_i^s, ν_i^t) and (ν_i^s, ν_j^t) are attacked. By the attack mentioned in the previous paragraph, we know that there are $k(k-1)/2$ edges in which both nodes incident to them are attack. Hence, the flow we get by this type of supply is $|E| - k(k-1)/2$.

We can conclude that the max-flow on the remained network is $|E| - k(k-1)/2 + 2\,|V| - 2k$, which is the given value of η.

Conversely, if we have the max-flow on the remained flow as η, we know $|V| - k$ edges in $\mathcal{E}_0$ are remained. Each of them will carry 2 flows through the edges $\mathcal{E}_{s,i}$.

With the given value η, we know that at least $k(k-1)/2$ nodes in $\mathcal{U}$ are blocked, and there is no flow through each of these nodes. To block a node $u_{i,j} \in \mathcal{U}$, the corresponding two edges (ν_i^s, ν_i^t) and (ν_j^s, ν_j^t) have to be attacked simultaneously. Since at most k edges in $\mathcal{E}_0$ can be attacked, this implies every two attacked edges (ν_i^s, ν_i^t) and (ν_j^s, ν_j^t) must have a blocked node $u_{i,j} \in \mathcal{U}$ connecting both of them. By our construction, each edge (ν_i^s, ν_i^t) in $\mathcal{E}_0$ is corresponding vertex $v_i \in G$. The set of k attacked edges can be mapped to a set of k nodes in G. Every pairs of two nodes in that node set has an edge in the given set of edges E, since each node $u \in \mathcal{U}$ has a corresponding edge in E. We can conclude from there that the set of k attacked edges in $\mathcal{E}_0$ implies a k-clique in G. $\square$

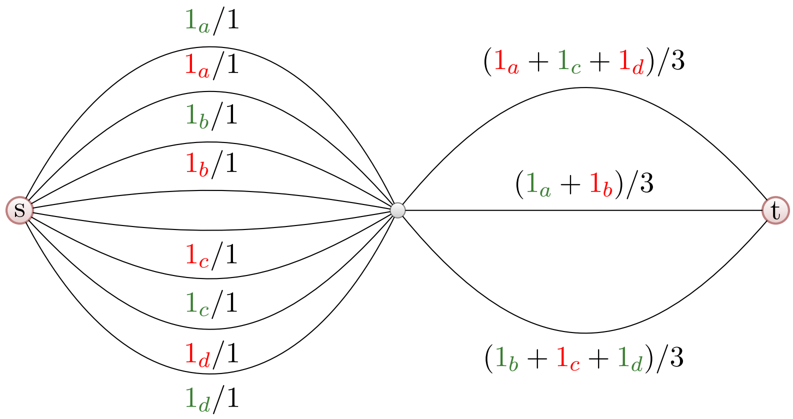


Figure 3.4: Network with distinct flow values when the number of attack is $k = 1$. A max-2-route flow solution is shown as a combination of four unitary 2-route flows: a, b, c, and d. $\mathscr{M}_k = 8$, $\mathscr{M}'_k = 5$, $\gamma^*_k = \frac{16}{3}$, and $\alpha^*_k = 6$. The value guaranteed by this solution is $\frac{\mathscr{M}_k}{2} = 4$.

3.3 A First Approximation

In this section, the multiroute flow is shown to be an approximation algorithm for any MLA flow. The results concerning only MLA-robust flow and MLA-reliable flow were introduced by Baffier and Suppakitpaisarn [9]. The rest concerns MLA-decomposition and were introduced by Baffier, Dai, and Suppakitpaisarn [12].

Recall the notations defined in section 3.1: γ^*_k represents the value of a MLA-reliable flow, and α^*_k represents the value of the MLA-robust flow.
Let $\mathscr{M}_k$ be the value of a max-$(k+1)$-route flow $\mathscr{M}$. Let $\mathscr{M}'_k$ be the MLA-effectiveness of $\mathscr{M}$. The network of figure 3.4 is an example with different value for γ^*_k, α^*_k, $\mathscr{M}_k$, and $\mathscr{M}'_k$.

3.3.1 Multiroute Flow as a $(k+1)$-Approximation Algorithm

The following proposition holds for any network and any number of attacks k.

Proposition 3.2 ([9])

$$\frac{\mathscr{M}_k}{k+1} \leqslant \mathscr{M}'_k \leqslant \gamma^*_k \leqslant \alpha^*_k$$

Proof. As discussed in [1], the MLA-effectiveness $\mathscr{M}'_k$ of the max-$(k+1)$-route flow is always greater than or equal to $\frac{\mathscr{M}_k}{k+1}$, the guaranteed value of a max-$(k+1)$-route flow

after k attacks.
It is direct from definition 3.4 that the value γ_k^* of the MLA-reliable flow is the maximum effectiveness among all the flow in the network. Thus, the effectiveness $\mathscr{M}_k'$ of the maximum multiroute flow cannot be larger than γ_k^*.
Last, in the MLA-robust context, the attacker action precedes the routing of the flow, at the opposite of the MLA-reliable flow variant. Hence, $\gamma_k^* \leqslant \alpha_k^*$. □

Corollary 3.2 is the cornerstone that gives the approximation ratio of the multiroute flow algorithm to the MLA-network flow problems. Its proof derived from the MLA-robust flow/MLA-robust cut duality and from lemma 3.2. This lemma says that, given a cut, its MLA-robust capacity cannot be $k+1$ times bigger than its $(k+1)$-route capacity. The result is then extended in corollary 3.2 to the MLA-robust capacity of a MLA-robust cut and to the $(k+1)$-route capacity of min-$(k+1)$-route cut.

Lemma 3.2 ([9])
For a given cut X, $\alpha_k(X) \leqslant \beta_{k+1}(X)$.

Proof. By definition 2.16,

$$\beta_{k+1}(X) = \min_{0 \leqslant i \leqslant k} \left(\frac{k+1}{k+1-i} \cdot \alpha_i(X) \right).$$

Hence,

$$\begin{aligned} \frac{\alpha_k(X)}{\beta_{k+1}(X)} &= \frac{\alpha_k(X)}{\min\limits_{0 \leqslant i \leqslant k} \left(\frac{k+1}{k+1-i} \cdot \alpha_i(X) \right)} \\ &= \alpha_k(X) \cdot \max_{0 \leqslant i \leqslant k} \left(\frac{k+1-i}{(k+1)\alpha_i(X)} \right) \\ &\leqslant \alpha_k(X) \cdot \max_{0 \leqslant i \leqslant k} \left(\frac{1}{\alpha_i(X)} \right) &= 1. \end{aligned}$$

□

Corollary 3.2 ([9])

$$\alpha_k^* \leqslant \mathscr{M}_k$$

Proof. Let X be a cut for a MLA-robust flow, and Y be a cut for a max-$(k+1)$-route flow.
By the duality property given in lemma 3.1,

$$\alpha_k^* = \alpha_k(X).$$

Similarly, from theorem 2.2, the value of a max-$(k+1)$-route flow is $\beta_{k+1}(Y)$:

$$\mathscr{M}_k = \beta_{k+1}(Y).$$

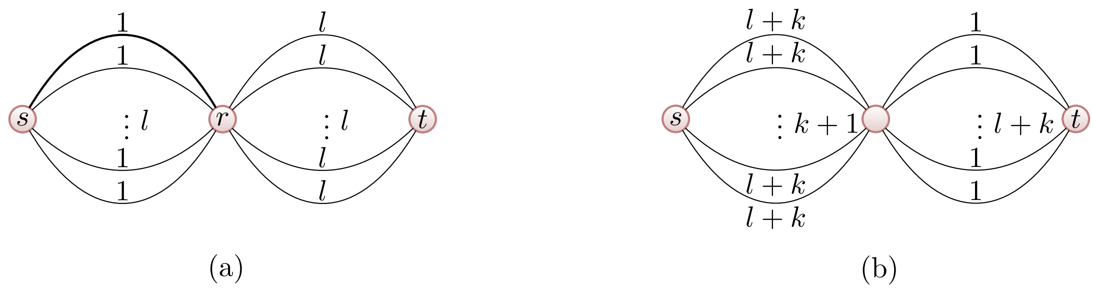


Figure 3.5: The networks used in the proofs of propositions 3.3 to 3.5.

From the previous lemma,

$$\frac{\alpha_k(X)}{\beta_{k+1}(X)} \leqslant 1$$

and

$$\frac{\alpha_k(Y)}{\beta_{k+1}(Y)} \leqslant 1.$$

Also, the nature of X and Y implies $\beta_{k+1}(Y) \leqslant \beta_{k+1}(X)$ and $\alpha_k(X) \leqslant \alpha_k(Y)$.

This corollary is proved by contradiction. Assume that

$$\frac{\alpha_k^*}{\mathscr{M}_k} = \frac{\alpha_k(X)}{\beta_{k+1}(Y)} > 1.$$

Since

$$\frac{\alpha_k(Y)}{\beta_{k+1}(Y)} \leqslant 1,$$

then

$$\frac{\alpha_k(X)}{\beta_{k+1}(Y)} > \frac{\alpha_k(Y)}{\beta_{k+1}(Y)}.$$

Multiplying by $\beta_{k+1}(Y)$ both sides of the inequality brings $\alpha_k(X) > \alpha_k(Y)$. That contradicts the starting assumption that X is a MLA-robust cut. □

This shows that the approximation ratio provided by the max-$(k+1)$-route flow is at most $k+1$ for both MLA-robust flow and MLA-reliable flow. This ratio is reached on an infinite graph for any value of k as shown below in propositions 3.3 to 3.5.

Proposition 3.3 (Baffier and Suppakitpaisarn [9])
There exists an infinite graph such that $\mathscr{M}'_k = (k+1)\cdot\frac{\mathscr{M}_k}{k+1} = \mathscr{M}_k$.

Proof. Let M be a max-$(k+1)$-route flow solution between s and r in the network of figure 3.5a. On that network,

$$\frac{\mathscr{M}_k}{k+1} = \frac{l}{k+1} \text{ and } \mathscr{M}'_k = \alpha_k^* = l - k.$$

The required equality is then true when l approaches infinity. □

Proposition 3.4 (Baffier and Suppakitpaisarn [9])
There exists an infinite graph such that $\gamma_k^* = (k+1) \cdot \mathscr{M}_k'$.

Proof. Consider the graph from figure 3.5a, this time with s and t as the source and sink, respectively. Based on theorem 2.2, for any flow solution M of the max-$(k+1)$-route flow, $\mathscr{M}_k = l$. Let M be such that on the right cut, the flow of M goes through only $k+1$ edges.
Hence $\mathscr{M}_k' = \frac{l}{k+1}$. Obviously $\gamma_k^* = l - k$.
The required equality is then true when l approaches infinity. □

Proposition 3.5 (Baffier and Suppakitpaisarn [9])
There exists an infinite graph such that $\alpha_k^* = (k+1) \cdot \gamma_k^*$.

Proof. This time consider the network from figure 3.5b. The max-flow value is $l + k$.
By nature,

$$\gamma_k^* = \frac{l+k}{k+1} \text{ and } \alpha_k^* = l.$$

The required equality is true when l approaches infinity. □

The last problem, MLA-decomposition, is slightly different from the two others. It consists in finding a decomposition of the flow, not only the flow itself. For this reason, the MLA-decomposition is compared to the decomposition of a max-$(k+1)$-route flow. Despite that difference, the results of corollary 3.2 will be extended to MLA-decomposition in corollary 3.3. Denote ψ_k^* the value of a MLA-decomposition.
For each $P_h \in \mathcal{P}_h$, let W_{P_h} be a function from $\mathcal{P}$ to $\mathbb{R}_{\geqslant 0}$, where $W_{P_h}(P) = 1$ if P is one of the h edge-disjoint paths in P_h, and $W_{P_h}(P) = 0$ otherwise.

Lemma 3.3 (Baffier, Dai, and Suppakitpaisarn [12])

$$\frac{\mathscr{M}_k}{k+1} \leqslant \psi_k^*$$

Proof. Let $f : E \to \mathbb{R}_{\geq 0}$ be a max-$(k+1)$-route flow of the graph G.
By nature of multiroute flows, there exists a valid decomposition H_{k+1} such that,

$$\text{for all } e, \sum_{P \in \mathcal{P} : e \in P} H_{k+1}(P) = f(e)$$

and

$$H_{k+1} = \sum_{P_{k+1} \in \mathcal{P}_{k+1}} b(P_{k+1}) W_{P_{k+1}}$$

for some function $b : \mathcal{P}_{k+1} \to \mathbb{R}_{\geq 0}$.
Since each $P_{k+1} \in \mathcal{P}_{k+1}$ is a $(k+1)$-edge-disjoint path, any

$$S' \in \mathcal{S} := \{S : S \subseteq E \text{ and } |S| = k\}$$

can attack at most k members of the path contained in P_{k+1}. By that, there is at least one path in each P_{k+1} survive after the attack.
The remaining value is at least

$$\sum_{P_{k+1} \in \mathcal{P}_{k+1}} b(P_{k+1}) = \frac{\mathscr{M}_k}{k+1}.$$

Hence,

$$\psi_k(H_{k+1}) \geq \frac{\mathscr{M}_k}{k+1}.$$

Finally,

$$\psi_k^* = \max_{H \in \mathcal{H}} \psi_k(H) \geq \psi_k(H_{k+1})$$

implies

$$\psi_k^* \geq \frac{\mathscr{M}_k}{k+1}.$$

□

Lemma 3.4 (Baffier, Dai, and Suppakitpaisarn [12])

$$\psi_k^* \leq \gamma_k^*$$

Proof. Let H_D be a solution of MLA-decomposition, S_D be a set of edges that the attacker choose when he knows the decomposition, and $F_D := (V, E, f_D)$ be a flow value on each edge, i.e. $f_D(e) := \sum_{P \in \mathcal{P} : e \in P} H_D(P)$.
Similarly, let $F_B := (V, E, f_B)$ be a solution of max-MLA-reliable, and S_B be a set of edges that the attacker choose to attack in that problem.
For $S \subseteq E$ and a flow $F := (V, E, f)$, let $\psi(H, S)$ be $\sum_{P \in \mathcal{P} : P \cap S = \emptyset} H(P)$ and $\beta(F, S)$ be a max-flow value of $(V, E \setminus S, f)$.
The definition of MLA-decomposition discussed in section 3.1.3, lead to

$$\psi_k^* = \psi(H_D, S_D) \leq \psi(H_D, S_B).$$

It can be easily seen that by the definition of MLA-reliable flow that

$$\psi(H_D, S_B) \leq \beta(F_D, S_B) \text{ and } \beta(F_D, S_B) \leq \beta(F_B, S_B) = \gamma_k^*.$$

□

Corollary 3.3 (Baffier, Dai, and Suppakitpaisarn [12])

$$\mathscr{M}_k \geq \gamma_k^* \geq \psi_k^* \geq \frac{\mathscr{M}_k}{k+1}$$

The combination of corollaries 3.2 and 3.3 and lemma 3.3 directly imply the existence of a $(k+1)$-approximation algorithm for the MLA-robust flow, the MLA-reliable flow, and the MLA-decomposition.

Theorem 3.4 (Baffier and Suppakitpaisarn [9], Baffier, Dai, and Suppakitpaisarn [12]) *A solution to the max-$(k+1)$-route flow problem provides a $(k+1)$-approximation to the MLA-robust flow, the MLA-reliable flow, and the MLA-decomposition problems.*

3.3.2 Approximation Ratio and Edge-Connectivity

The upper bound $k+1$ can be further improved for a graph with small source-sink edge connectivity λ. The theorem 3.5 shows that the approximation ratio provided by the maximum multiroute flow can be significantly smaller than $k+1$.

Lemma 3.5 ([9])
Let X be a MLA-robust cut, and let Y be a minimum $(k+1)$-route cut. The ratio between the MLA-robust capacity of X and $(k+1)$-route capacity of Y cannot be larger than $\frac{p-k}{p}$ when $p = \min(p_X, p_Y)$, p_X is the cardinality of the cut-set of X, and p_Y is the cardinality of the cut-set of Y.

Proof. By definitions 2.15 and 3.1,

$$\frac{\alpha_k^*}{\mathscr{M}_k} = \frac{\alpha_k(X)}{(k+1)\min\limits_{0\leqslant i\leqslant k}\left[\frac{\alpha_i(Y)}{k+1-i}\right]} = \frac{\alpha_k(X)}{k+1}\cdot\max_{0\leqslant i\leqslant k}\frac{k+1-i}{\alpha_i(Y)} \leqslant \frac{\alpha_k(Y)}{k+1}\cdot\max_{0\leqslant i\leqslant k}\frac{k+1-i}{\alpha_i(Y)}.$$

Furthermore, the average value of the capacity $\frac{\alpha_i(Y)}{p_X-i}$ should be at least $\frac{\alpha_j(Y)}{p_X-j}$ if $i \leqslant j$. Hence,

$$\frac{\alpha_k(Y)}{p_X-k} \leqslant \frac{\alpha_i(Y)}{p_X-i}.$$

Then,

$$\begin{aligned}
\tfrac{\alpha_k(Y)}{k+1}\cdot\max_{0\leqslant i\leqslant k}\tfrac{k+1-i}{\alpha_i(Y)} &\leqslant \tfrac{\alpha_k(Y)}{k+1}\cdot\max_{0\leqslant i\leqslant k}\tfrac{k+1-i}{\frac{p_X-i}{p_X-k}\alpha_k(Y)}\\
&\leqslant \tfrac{1}{k+1}\max_{0\leqslant i\leqslant k}\left[(p_X-k)\tfrac{k+1-i}{p_X-i}\right]\\
&\leqslant \tfrac{p_X-k}{p_X}
\end{aligned}$$

Similarly,

$$\frac{\alpha_k(X)}{(k+1)\min\limits_{0\leqslant i\leqslant k}\left[\frac{\alpha_i(Y)}{k+1-i}\right]} \leqslant \frac{p_Y - k}{p_Y}.$$

Hence,

$$\frac{\alpha_k(X)}{(k+1)\min\limits_{0\leqslant i\leqslant k}\left[\frac{\alpha_i(Y)}{k+1-i}\right]} \leqslant \frac{p - k}{p}.$$

□

Theorem 3.5 ([9])
Let U be the ratio between the largest and smallest capacities of the edges in the network. The maximum multiroute flow algorithm provides a $\frac{U(\lambda-k)}{U(\lambda-k)+k}(k+1)$-approximation algorithm for MLA-robust flow and the MLA-reliable flow problems.

Proof. Let X be a MLA-robust cut, and let Z be a minimal cut –as defined in section 2.2.1. Let $\alpha_k(Z)$ and $\alpha_k(X)$ be the MLA-robust capacities of Z and X. Note that if p is the cardinality of X, then $\alpha_k(X)$ is the sum of the $p-k$ lowest capacities of the edges in X. Let U be the ratio between the largest and smallest capacities, hence

$$\frac{\alpha_k(Z)}{\alpha_k(X)} \leqslant U \cdot \frac{\lambda - k}{p - k}.$$

Since $\alpha_k(Z) \geqslant \alpha_k(X)$ by the definition of the MLA-robust cut,

$$1 \leqslant \frac{\alpha_k(Z)}{\alpha_k(X)} \leqslant U \cdot \frac{\lambda - k}{p - k}.$$

Then,

$$p - k \leqslant U \cdot (\lambda - k) \text{ implies } p \leqslant U \cdot (\lambda - k) + k.$$

Thus the upper bound is updated to

$$\frac{p-k}{p} \cdot (k+1) \leqslant \frac{U(\lambda - k)}{U(\lambda - k) + k}(k+1).$$

□

Corollary 3.4 ([9])
If the cut-set of min-$(k+1)$-route cut has cardinality $k+1$, $\alpha_k^* = \gamma_k^* = \frac{\mathscr{M}_k}{k+1}$.

Proof. Lemma 3.5 says that $\frac{\alpha_k^*}{\mathscr{M}_k} \leqslant \frac{p-k}{p}$. For this corollary, $p = k+1$. Hence,

$$\frac{\alpha_k^*}{\mathscr{M}_k} \leqslant \frac{k+1-k}{k+1} = \frac{1}{k+1}.$$

□

3.4 Polynomial Time Exact Algorithms using Parametric Flows

This section covers two variant of the ES method, that has been modified through the analysis in sections 3.4.2 and 3.4.3 of two parametric variants of the multiroute flow: route-parametric and link-parametric methods. This work combines the first method with the non-integer number route variant of the multiroute flow as in [4] and described in definition 2.18 and lemma 2.2. The EMRF, i.e. the ES method applied to route-parametric multiroute flow with a non-integer number of route, can compute in $O(\lambda T)$ a MLA-robust flow for a certain category of networks. Its efficiency is improved in chapter 4 and evaluated through experiments in chapter 5. The analysis of the second method, is partially based on the work by Baffier [8], and leads to a $O(\lambda^2 T)$ exact algorithm to compute a MLA-robust flow for another category of network.

As stated in theorem 2.2, maximum multiroute flow/minimum multiroute cut are dual problems. Same holds from lemma 3.1 for MLA-robust flow and MLA-robust cut. Additionally, as seen in definition 2.16, for any number of route h, the h-route capacity of a cut is defined using MLA-robust capacities of the same cut. This indicates a possible relation between multiroute flow and MLA-robust flow.

The parametric optimization scheme is introduced to several problems in network design, such as in Karp and Orlin [62] where the authors consider the parametric shortest path problem . As discussed by Gusfield [55, 56], the goal of this scheme is to compute the solution of all the parametric instances efficiently. Then, when the parameter is fixed, a solution to the instance is retrieved instead of computed. For instance, the parametric multiroute flow algorithm proposed in [8] outputs a function taking a capacity of a parametric link e to a maximum multiroute flow value. It outputs a vector of breakpoints of this continuous function, from which the value of the multiroute flow can be retrieved in logarithmic time once the value of the parametric link is fixed. Several works have introduced the variants of that scheme to solve problems in database by Eisner and Severance [43] and in computer vision by Kolmogorov, Boykov, and Rother [73]. For those variants, the output function is shown to be linear piecewise. In [43] the authors introduce a method referred here as ES algorithm to find that linear piecewise function in $O(pT)$ when p is the number of line segments in that function and T is the running time to get an optimization result for a specific parameter value.

As mentioned by Kolmogorov, Boykov, and Rother [73], the ES algorithm makes at most two calls to the max-flow algorithm per breakpoint (on average). Here, due to a specific setup the EMRF makes at most 1 call to max-flow per breakpoint in the worst case. A method to reduce the average number of those calls is detailed in chapter 4.

Section 3.4.1 describes the restricted maximum flow introduced by Aggarwal and Orlin [1] and the time complexity to compute it through the ES algorithm. The analysis and the complexity of the route-parametric multiroute flow is covered in section 3.4.2, and the one of link-parametric multiroute flow in section 3.4.3. Finally, the link between the route-parametric (and the EMRF, its variants with a noninteger number of routes) and MLA-robust flow is shown in section 3.4.4.

3.4.1 Restricted Maximum Flow

As defined in definition 2.17, for any nonnegative real number x, the value $\mathcal{F}(x)$ denote the maximum flow of the restricted network G^x. The work by Aneja, Chandrasekaran, and Nair [3] already mentions such an algorithm to compute the restricted flow function $\mathcal{F}$ in at most $(n-1)$ steps. This complexity is actually based on the maximum size of a a minimal cut on the restricted network, that is the source-edge connectivity. However, the connectivity can be quite different from the number of nodes. For instance, in grid networks, the connectivity is related to the number of dimension in the grid, not the size of the network that can be arbitrary high. On the other hand, many practical networks are not representable by simple graphs, and multiple links between nodes are sometimes allowed. In that case the connectivity can be arbitrary high compared to the number of nodes. Thus, the results were reformulated in Baffier, Suppakitpaisarn, Hiraishi, and Imai [10] as two corollaries with the goal to be used in the computation of MLA flows.

Corollary 3.5 ([10])
The parametric function $\mathcal{F}$ is linear piecewise, continuous, and derivative non-increasing. The function contains at most $\lambda + 1$ line segments, and there exists x^* such that $\mathcal{F}$ is constant on $[x^*, \infty[$.

Corollary 3.6 ([10])
Using ES algorithm, the function $\mathcal{F}$ is computed in $O(\lambda T)$.

For the remaining part of this work, $\mathcal{S} = \{(\eta_0, \mathcal{F}(\eta_0)), \ldots, (\eta_q, \mathcal{F}(\eta_q))\}$ is the output of ES algorithm, when $0 = \eta_0 < \eta_1 < \cdots < \eta_q$, and the derivation of $\mathcal{F}$ at η_i^- is not equal to the derivation at η_i^+ for $1 \leqslant i \leqslant q$. The pair $(\eta_i, \mathcal{F}(\eta_i))$ is called a breakpoint.

Those breakpoints fully describe function $\mathcal{F}$, since for any nonnegative real number x,

$$\mathcal{F}(x) = \begin{cases} \mathcal{F}(\eta_i) + \mathcal{F}'(\eta_i^+)(x - \eta_i) & \text{for } \eta_i \leqslant x \leqslant \eta_{i+1}, \\ \mathcal{F}(\eta_q) & \text{for } x \geqslant \eta_q, \end{cases}$$

where $\mathcal{F}'(\eta_i^+) := \frac{\mathcal{F}(\eta_{i+1})-\mathcal{F}(\eta_i)}{\eta_{i+1}-\eta_i}$. From corollary 3.5 $q \leqslant \lambda + 1$, hence the description of the breakpoints can be stored in $O(\lambda)$ memory.

3.4.2 Route-Parametric Multiroute Flow

Recall definition 2.18 and lemma 2.2. The definition of function $\mathcal{F}$ implies that a max-h-route flow value is equal to v^* if $v^* = \mathcal{F}(\frac{v^*}{h})$ and $v^* > 0$ for $h \leqslant \lambda$. The value is $\mathcal{F}(\eta_1)$ when $h = \lambda$, and 0 for any $h > \lambda$.

Let $\mathcal{R}$ be the function taking $h \in \mathbb{R}_+$ to the max-h-route flow value. Then, the following results derive from lemma 2.2 and theorem 2.2 and also formally define the function $\mathcal{R}$.

Theorem 3.6 ([10])
For $1 \leqslant i \leqslant q$, let $h_i := \frac{\mathcal{F}(\eta_i)}{\eta_i}$. Also, for $0 \leqslant i \leqslant q$, let $\mu_i := \mathcal{F}'(\eta_i^+) := \frac{\mathcal{F}(\eta_{i+1})-\mathcal{F}(\eta_i)}{\eta_{i+1}-\eta_i}$ be a derivation of $\mathcal{F}$ at η_i^+, and $\gamma_i := \mathcal{F}(\eta_i) - \mathcal{F}'(\eta_i^+)\eta_i$, then

$$\mathcal{R}(h) = \begin{cases} \frac{h\gamma_i}{h-\mu_i} & \text{for } h_{i+1} < h \leqslant h_i, \\ \gamma_q & \text{for } 0 < h \leqslant h_q, \\ 0 & \text{for } h > h_1. \end{cases}$$

Proof. Recall from theorem 2.2 that $\mathcal{F}(x) = \min_{0 \leqslant i \leqslant \lambda} (ix + P_i)$. Since $P_i > 0$ for $i < \lambda$ and $P_\lambda = 0$, there exists $x^* > 0$ such that $\mathcal{F}(x) = \lambda x + P_\lambda = \lambda x$ for $x \leqslant x^*$. Thus, $(\eta_1, \mathcal{F}(\eta_1)) = (\eta_1, \lambda\eta_1)$, and $h_1 = \lambda$. As stated above, $\mathcal{F}(h) = 0$ for $h > h_1$.

Lemma 2.2 states that the max-h-route flow value is equal to hx if $hx = \mathcal{F}(x)$. The goal is to find the point (x, hx) where the line hx cut the function $\mathcal{F}(x)$. That is equivalent to the task of finding a value x such that $\mathcal{L}(x) := \mathcal{F}(x) - hx$ is equal to 0. When $h < \lambda$, the function is increasing for a small x and decreasing after the derivation of $\mathcal{F}$ becomes less than h. When h becomes larger, the point x such that $\mathcal{F}$ begin to decrease comes faster, and the decrement are faster. Thus, the value x such that $\mathcal{L}(x) = 0$ becomes smaller in that case.

Consider the case when $h = h_i$ for some $1 \leqslant i \leqslant q$. Since $h_i := \frac{\mathcal{F}(\eta_i)}{\eta_i}$, then $h_i\eta_i = \mathcal{F}(\eta_i)$. Hence, the max-$h$-route flow is $h_i\eta_i$ and the cut point is $(\eta_i, h_i\eta_i)$.

When $h = h_q$, the value x such that $\mathcal{L}(x) = 0$ is η_q. Previous paragraph induces that x will become larger for $h < h_q$. The definition of breakpoints and corollary 3.6 implies

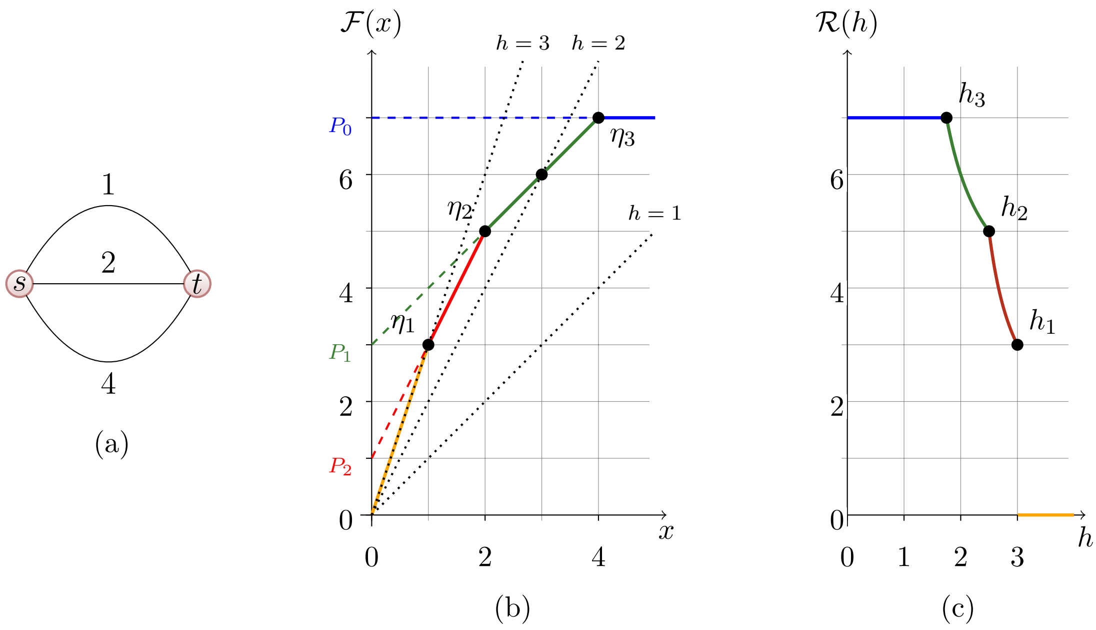


Figure 3.6: Example of a route-parametric multiroute flow:(a) the original network (b) The parametric function $\mathcal{F}$ of this network (c) the function $\mathcal{R}$ taking the value h to the max-h-route flow value

$\mathcal{F}(x) = \mathcal{F}(\eta_q) = \gamma_q$ for all $x \geqslant \eta_q$. Hence, if $h < h_q$, the point that the line hx cut $\mathcal{F}(x)$ is (x^*, γ_q) for some x^*. The max-h-route flow is γ_q.

Finally when $h_{i+1} < h \leqslant h_i$, let x be such that $\mathcal{L}(x) = 0$, then x is in between η_i and η_{i+1}. Because of that, the line hx cut the function $\mathcal{F}(x)$ at the line segment linking $(\eta_i, \mathcal{F}(\eta_i))$ and $(\eta_{i+1}, \mathcal{F}(\eta_{i+1}))$. Thus, algebraically, the line segment is $\mu_i x + \gamma_i$, and the cut point is $(\frac{\gamma_i}{h-\mu_i}, \frac{h\gamma_i}{h-\mu_i})$. Hence the max-$h$-route flow is $\frac{h\gamma_i}{h-\mu_i}$. □

By theorem 3.6, a corollary follows.

Corollary 3.7 ([10])
The function $\mathcal{R}$ taking the value h to max-h-route flow is hyperbolic piecewise with at most $\lambda + 1$ hyperbolic segments. The function can be computed in $O(\lambda T)$.

Proof. Corollary 3.6 states that the set of breakpoints of $\mathcal{R}$ can be computed in $O(\lambda T)$. Then, $\mathcal{R}$ is computed from those breakpoints in $O(\lambda)$ as shown in theorem 3.6. □

The function that results from corollary 3.7 can compute the max h-route flow value for any h in $O(\log \lambda)$. Those include the case when $h = \lambda$. In [1], Aggarwal and Orlin improve the max-h-route flow algorithm, but the time complexity still is $O(\lambda mn)$ only to compute the max-λ-route flow in the worst case. Because of this, the EMRF algorithm outputs more general results without increasing the time complexity –in particular when k is close to λ.

3.4.3 Link-Parametric Multiroute Flow

The network considered in this section possesses a different parametric component. That is, the capacity of a given link e can be any real number, while the capacities of the other links are fixed. For this section, h is a fixed integer. The link e is called a *parametric edge*, and the variable z denotes the capacity of e. This work slightly improved the one by Baffier [8] by simplifying the proofs and making the relation with the MLA-robust flow problem possible, as discussed in section 3.4.4.

Let $\alpha_i^z(X)$, $\beta_i^z(X)$ refer to the values $\alpha_i(X)$ and $\beta_i(X)$ when the capacity of that parametric edge is equal to z. Also, let $\mathscr{C}_e$ be the set of cuts containing the parametric edge e, and $\overline{\mathscr{C}_e} := \mathscr{C} \setminus \mathscr{C}_e$.

Lemma 3.6 ([8, 10])
For any cut $X \in \mathscr{C}_e$, $\alpha_i^z(X) = \min(\alpha_{i-1}^0(X), \alpha_i^0(X) + z)$ for any $i \geqslant 1$ and $\alpha_0^z(X) = \alpha_0^0(X) + z$.

Proof. The second statement is obvious. To prove the first one, let the cut-set of X be $\{e_0, e_1, \dots, e_p\} \cup \{e\}$ where $c(e_0) \geqslant c(e_1) \geqslant \cdots \geqslant c(e_p)$ and $e_i \neq e$ for all i.
Remark that $\alpha_i^0(X) = \sum_{j=i}^{p} c(e_j)$.

If $z \geqslant c(e_{i-1})$, then $\{e_0, \dots, e_{i-2}, e\}$ will be the set of edges with the i largest capacities. Hence,

$$\alpha_i^z(X) = \sum_{j=i-1}^{p} c(e_j) = \alpha_{i-1}^0(X).$$

If $z < c(e_{i-1})$, then $\{e_0, \dots, e_{i-1}\}$ will be the set of edges of size i with the i largest capacities. Hence,

$$\alpha_i^z(X) = \sum_{j=i}^{p} c(e_j) + z = \alpha_i^0(X) + z.$$

□

Theorem 3.7 ([8, 10])
Let $\mathscr{M}_h(z)$ be a value of a max-h-flow when the capacity of the parametric edge is z. The function $\mathscr{M}_h$ is $(h+1)$-piecewise linear, continuous, and derivative non-increasing. Also, there exists some z^* such that the derivative of $\mathscr{M}_h$ is 0 for all $z \geqslant z^*$.

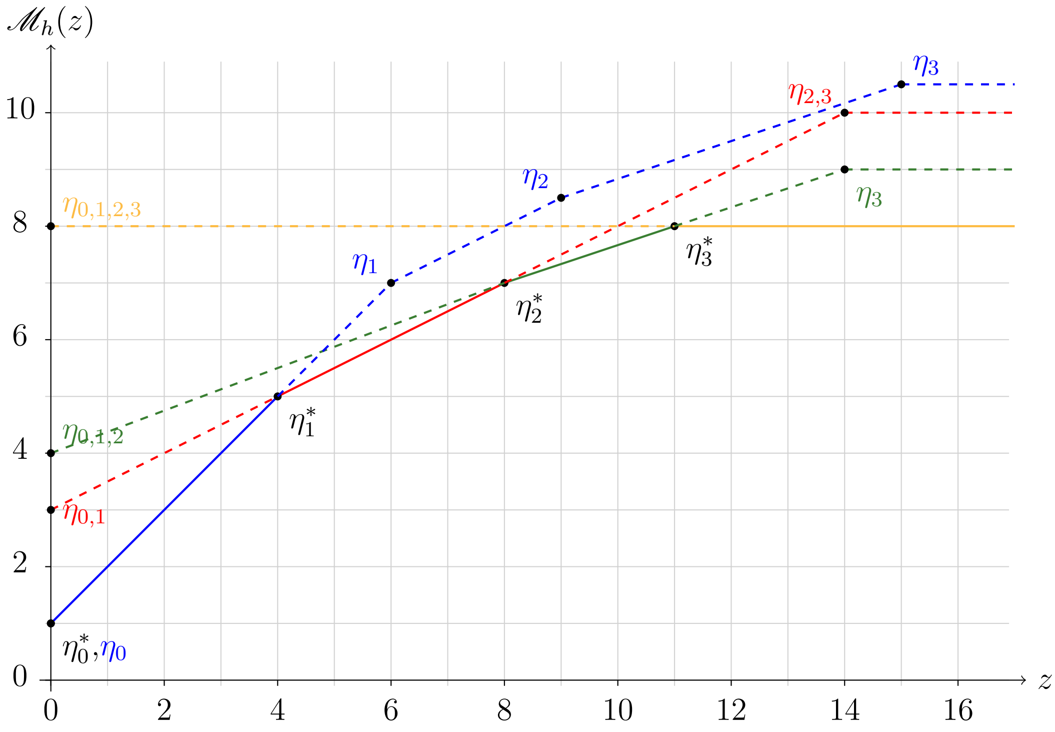


Figure 3.7: The link-parametric max-h-route flow function $\mathscr{M}_h$ is represented by the plain lines. The different cuts are represented by a color each, and the dashed parts correspond to value of the parameter when the concerned cut is not minimum. The breakpoints are indicating by a $*$ mark.

Proof. Let $P_i^z = \min_{X \in \mathscr{C}} \alpha_i^z(X)$. Lemma 3.6 implies that

$$P_i^z = \min\left(\min_{X \in \mathscr{C}_e}(\alpha_{i-1}^0(X), \alpha_i^0(X) + z), \min_{X \in \mathscr{C}} \alpha_i(X)\right).$$

Let $J_i := \min\left(\min_{X \in \mathscr{C}_e} \alpha_{i-1}^0(X), \min_{X \in \mathscr{C}} \alpha_i(X)\right)$, and $H_i := \min_{X \in \mathscr{C}_e} \alpha_i^0(X)$.
Hence $P_i^z = \min(J_i, H_i + z)$.
By definition 2.16, $\mathscr{M}_h(z) = \min_{0 \leqslant i < h}\left(\frac{hP_i^z}{h-i}\right)$. Thus, $\mathscr{M}_h(z) = \min\left(J, \min_{0 \leqslant i < h}\left[\frac{h(H_i+z)}{h-i}\right]\right)$, where $J := \min_{0 \leqslant i < h}\left(\frac{hJ_i}{h-i}\right)$.

It is easy to see that $\mu_i(z) := \frac{h(H_i+z)}{h-i}$ is a linear function of z. Since the function $\mathscr{M}_h$ is the minimum of those linear functions, the function is $(h+1)$-piecewise linear, continuous, and derivative non-increasing.
Moreover the functions μ_i are increasing, there exists some z^* such that $\mu_i(z) \geqslant J$ for all i and $z \geqslant z^*$.
Hence $\mathscr{M}_h(z) = J$ for those z. $\square$

Corollary 3.8 ([10])

Using ES algorithm, the breakpoints of function $\mathscr{M}_h$ can be found in $O(h^2T)$.

Proof. As shown in theorem 3.7, the function $\mathscr{M}_h$ at most $h+1$ line segments. The calculation of $\mathscr{M}_h(z)$ at a specific z is the computation of one max-h-route flow value. By any multiroute flow algorithm, the flow value can be computed in $O(hT)$.
Hence, it takes $O(h^2T)$ to find all breakpoints of $\mathscr{M}_h$. □

Similarly to section 3.4.2, $\mathscr{M}_h(z)$ can be calculated for a specific z from the set of breakpoints obtained from ES algorithm. The calculation can be done as follows, assuming that the set of breakpoints is $\{(\eta_0, \mathscr{M}_h(\eta_0)), \dots, (\eta_q, \mathscr{M}_h(\eta_q))\}$.

$$\mathscr{M}_h(z) = \begin{cases} (\mathscr{M}_h(\eta_{i+1}) - \mathscr{M}_h(\eta_i))\frac{z-\eta_i}{\eta_{i+1}-\eta_i}, & \text{for } \eta_i \leqslant z \leqslant \eta_{i+1} \\ \mathscr{M}_h(\eta_q), & \text{for } z \geqslant \eta_q \end{cases}.$$

3.4.4 Application of Parametric Multiroute Flows to Solve MLA Problems

The results of sections 3.4.2 and 3.4.3 shows a close relation between both route-parametric and link-parametric multiroute flow. As the first is computed faster, this work will mainly concern the correlation between MLA-robust flow and EMRF –that is the route-parametric multiroute flow extended to a non-integer number of routes.
However, remark that by adding a parametric link between the source and the sink of the network, if the link-parametric multiroute flow function has a line of slope k, then the k-MLA-robust flow is solved. The proof is not detailed here but comes from the formulation of the link-parametric multiroute flow function.

Theorem 3.8 ([10])

$$\mathcal{F}(x) = \min_{0 \leqslant i \leqslant \lambda} (ix + \alpha_i^*)$$

Proof. Consider a cut $X \in \mathscr{C}$ with a cutset $E = \{e_0, \dots, e_p\}$ such that

$$c(e_0) \geqslant c(e_1) \geqslant \cdots \geqslant c(e_p).$$

Denote the capacity of X in the graph G^x by

$$\mathcal{C}_X(x) = \sum_{i:c(e_i)<x} c(e_i) + \sum_{i:c(e_i)\geqslant x} x.$$

Recall definition 2.15 and denote $\sum_{i=\ell}^{p} c(e_i)$ as $\alpha_\ell(X)$. Then,

$$\mathcal{C}_X(x) = \begin{cases} (p+1)x = \alpha_{p+1}(X) + (p+1)x \text{ if } x \leqslant c(e_p) \\ \mathcal{C}_X(x) = \alpha_\ell(X) + \ell x \text{ if } c(e_\ell) < x \leqslant c(e_{\ell-1}) \\ \mathcal{C}_X(x) = \alpha_0(X) \text{ otherwise} \end{cases}.$$

Now, the function shown above can be simplified to $\mathcal{C}_X(x) = \min_{0 \leqslant j \leqslant p+1} (\alpha_j(X) + jx)$. Let $c_\ell(X) < x \leqslant c_{\ell-1}(X)$. From the previous paragraph, $\mathcal{C}_X(x) = \alpha_\ell(X) + \ell x$. To prove that

$$\alpha_\ell(X) + \ell x = \min_{0 \leqslant j \leqslant p+1} (\alpha_j(X) + jx),$$

it is sufficient to show that,

$$\text{for any } j \neq \ell,\ \alpha_\ell(X) + \ell x \leqslant \alpha_j(X) + jx.$$

$$\begin{aligned} \text{For } j < \ell, \qquad \alpha_j(X) + jx = & \sum_{i=j}^{p} c(e_i) + jx = \sum_{i=\ell}^{p} c(e_i) + \sum_{i=j}^{\ell-1} c(e_i) + jx \\ \geqslant & \sum_{i=\ell}^{p} c(e_i) + (\ell - j)x + jx = \alpha_\ell(X) + \ell x. \end{aligned}$$

$$\begin{aligned} \text{For } j > \ell, \qquad \alpha_j(X) + \ell x = & \sum_{i=j}^{p} c(e_i) + jx = \sum_{i=j}^{p} c(e_i) + (j - \ell)x + \ell x \\ \geqslant & \sum_{i=j}^{p} c(e_i) + \sum_{i=\ell}^{j-1} c(e_i) + jx = \alpha_\ell(X) + \ell x. \end{aligned}$$

A similar argument is sufficient to show the case when $x > c(e_0)$ and $x \leqslant c(e_p)$. Since $\alpha_i(X) = 0$ for $i \geqslant p$, the formula can be further simplify to $\mathcal{C}_X(x) = \min_{j \geqslant 0} (\alpha_j(X) + jx)$.

It is easy to see that $\mathcal{F}(x) = \min_{X \in \mathscr{C}} \mathcal{C}_X(x)$. Hence,

$$\mathcal{F}(x) = \min_{X \in \mathscr{C}} \min_{i \geqslant 0} (ix + \alpha_i(X)) = \min_{i \geqslant 0} \left(ix + \min_{X \in \mathscr{C}} \alpha_i(X) \right) = \min_{i \geqslant 0} (ix + \alpha_i^*).$$

Since $\alpha_i^* = 0$ for $i \geqslant \lambda$, it implies

$$\mathcal{F}(x) = \min_{0 \leqslant i \leqslant \lambda} (ix + \alpha_i^*).$$

□

Corollary 3.9 ([10])
If there exists a real number x such that $kx + \alpha_k^* = \mathcal{F}(x)$, then the max-$\left(\frac{\mathcal{F}(x)}{x}\right)$-route flow value is equal to the k-MLA-robust flow and k-MLA-reliable flow values.

Proof. Let $h := \frac{\mathcal{F}(x)}{x}$.
Then, theorem 3.8 implies that the cut point of lines $y = hx$ and $y = \mathcal{F}(x)$ is $(x, kx+\alpha_k^*)$.
From lemma 2.2, the max-h-route flow value is $kx + \alpha_k^*$.
By definition of max-h-route flow, the amount of flow remaining after k edge attacks is at least $\frac{h-k}{h}(kx + \alpha_k^*)$. Then,

$$\frac{h-k}{h}(kx + \alpha_k^*) = \frac{\frac{\mathcal{F}(x)}{x} - k}{\frac{\mathcal{F}(x)}{x}}(kx + \alpha_k^*) = \frac{\frac{kx+\alpha_k^*}{x} - k}{\frac{kx+\alpha_k^*}{x}}(kx + \alpha_k^*) = \alpha_k^*.$$

Hence, the max-h-route value is equal to MLA-robust flow value. Since the MLA-reliable flow is in between MLA-robust flow and max-h-route flow value, the MLA-reliable flow is also equal to that max-h-route flow value. □

3.5 Conclusion

The MLA-robust flow, MLA-reliable flow, and MLA-decomposition problems cover several variants of an optimal flow in a network under attack. The max-$(k+1)$-route flow is a $(k+1)$-approximation algorithm to any of the MLA flows. Other approximation algorithm, including a heuristic one, will be presented in chapter 4. In the same chapter, several techniques to increase the category of network instances covered and the speed of the exact algorithm will be introduced.

Chapter 4

Analysis and Improvement of Algorithms for MLA Flows

The extended multiroute flow (EMRF) method designed in chapter 3 provides an exact polynomial time algorithm to compute MLA-robust flow and MLA-reliable flow in polynomial time for a class of instances. Along with it, a max-$(k+1)$-route flow is a $(k+1)$ approximation to both k-MLA-robust flow and k-MLA-reliable flow. In this chapter, the category of instances solvable with the exact algorithm is enlarged and the quality of the approximation is improved. In both cases, the average speed of the algorithms is enhanced. The analysis of both the exact and approximated methods and their improvements leads to a better characterization of MLA instances and networks.

The original EMRF method suffers one obvious default: it solves only a part of the possible instances. As the MLA-robust flow problem is NP-hard, as shown in section 3.2, this lack cannot be fully solved. However, some improvement to enlarge that class are done in sections 4.1.1 and 4.1.2, and those instances are said to be EMRF-solvable.

Section 4.1.2 considers the possibility to skip some of the steps in the construction of EMRF function. The breakpoints of the restricted max-flow are at the intersection of lines with different slopes. Thus, the computation of a restricted max-flow in such a point brings information about two slopes in one step. Hence, by analyzing the composition of a cut, the position of the breakpoints related to that cut can be predicted, and possibly skipped. However, since a point is only the intersection of two lines, skipping heuristically more than a point might lead to a lost of information. Consider skipping more. In the ES algorithm, instead of looking at the current slope intersection, compute what would be the intersection between two cutset parametric flow, and, if the value computed is as expected, then stop. The new computation steps number is at most the number of switching cutset.

The relation between the MLA-robust flow and the max-flow is such that, if the piece of the parametric function with slope k exists, then the k-MLA-robust flow is solved in polynomial time. In this part, a necessary condition to the intractability of an instance is detailed and referred as flying cut. Consider the case where two breakpoints coincide, with indexes k and $k+1$. The intersection of the ordinate axis with the line of slope k passing by the breakpoint will give the exact value of the k-MLA-robust flow. In section 4.2, the matter of improving the bounds of MLA-robust flow is discussed. There, it is proved that the line with slope $k+1$ passing by the breakpoint of degree k provide a lower bound of the $(k+1)$-MLA-robust flow value. This method, referred as pseudo-tangent directly improves the previous lower bound given by the max-$(k+1)$-route flow. This pseudo-tangent method also solve the problem of link capacities with the same values. However, the information that the problem is solved is not detected by this method, making the capacity-differentiation approach more suitable depending of the objective. The upper bound of the MLA-robust flow can be heuristically searched by using the duality of MLA-robust flow and MLA-robust cut, and the natural property of any MLA-robust cut.

The current approximation and exact algorithm compute the function value of the solution. However, is it possible to construct a flow that is the best against any number of attacks? A partial answer is called IMRF and was introduced by Baffier and Suppakitpaisarn [9]. This flow is built on combining iteratively multiroute flow with different route values. At each iteration, the multiroute flow is computed on the residual network of the previous step. If the k-MLA-efficiency of the IMRF is equal to the k-MLA-robust flow, then it is the best resilient flow against k attacks. If this property holds for any number of attacks, then the iterative flow is a solution to the best-MLA flow problem. This IMRF is based on the analysis of completed-MRF described in section 4.3.

The characterization –complexity, properties, type– of the MLA instances can be detailed further than just EMRF-solvable or not, as shown in section 4.4 and figure 4.5. The overview of the MLA instances classification considers the previously mentioned: EMRF-solvable (including capacity differentiation and pseudo-tangent methods), flying-cut, best-MLA, IMRF-solvable.

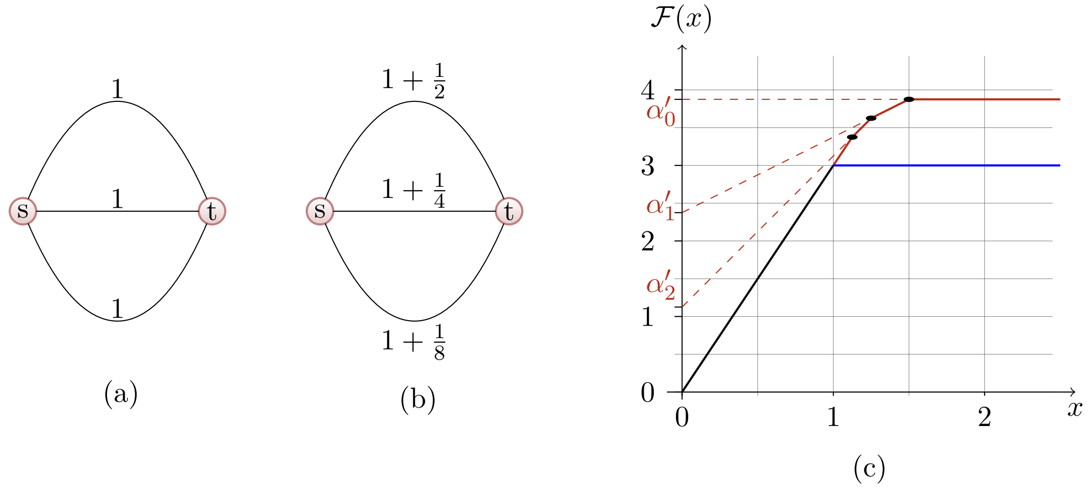


Figure 4.1: Example of the capacity differentiation algorithm. The plain line (in black) from $(0,0)$ to $(1,3)$ are common to the plot of figures 4.1a and 4.1b. The straight line (in blue) from $(1,3)$ with constant value represent the figure 4.1a. The piecewise line (in red) from $(1,3)$ that takes successive slope values represents figure 4.1b. The dashed line (in red) represents the projections of the of the max-flow of figure 4.1b. The MLA-robust flow values are obtained by rounding down to the integer value $\alpha'_0, \alpha'_1, \alpha'_2$: $\alpha^*_0 = \lfloor \alpha'_0 \rfloor = 3$; $\alpha^*_1 = \lfloor \alpha'_1 \rfloor = 2$; $\alpha^*_2 = \lfloor \alpha'_2 \rfloor = 1$.

4.1 Exact Algorithm Improvements

4.1.1 Capacity differentiation

The exact algorithm suffers one default. The MLA-robust flow values deduction depends on the existence of part of the parametric function of slope k. One case of a missing slope is easy to explain. It is directly due to the equality of some capacities of the links as shown in figure 4.1. This can be solved by forcing a differentiation in the capacity of the edges. However, this differentiation should not modify the result in an inconvenient way, that is not being able to output the correct value. It can be done by adding $\frac{1}{2^i}$ to each edge of index i. In the case of integer capacities, because the sum of the $\frac{1}{2^i}$ series converges to 1 by inferior value, it can be considered that the floored value of the flow is the correct one. It can be extended to the general case by considering the smallest interval between the capacities of any pair of links. The efficiency of this method is evaluated on different kind of networks with integral values in sections 5.5.3 and 5.5.5.

From this point onwards, all the capacities will be differentiated if necessary, without any loss in the analysis. For instance, if the restricted max-flow $\mathcal{F}$ of a network is missing a slope, consider it is not due to a tuple of links with the same capacity value.

4.1.2 Skipping steps

Given a network, suppose there exists a cut X that is a MLA-robust cut for several successive values of $k \in [i..j]$. Knowing that property, computing all the k-MLA-robust cut values can be done by just sorting and adding the capacity of the links in X such that all the MLA-robust capacity values of X are computed. By the nature of X, those MLA-robust capacity values are also the MLA-robust cut values –and the MLA-robust flow values by duality. This hypothesis is used to try to lower the average time complexity of the EMRF algorithm.

Definition 4.1
A breakpoint of degree $i \in [0..(\lambda - 1)]$ of the restricted flow function $\mathcal{F}$ is said to be a *switching point* if, and only if, there exist no cut X such that X is a solution to both i-MLA-robust cut and $(i+1)$-MLA-robust cut.
The breakpoint of degree λ is always a *switching point.*
The *switching degree* of a network is then defined as the number of *switching points* of the restricted flow function.

The method discussed above definition 4.1 cannot improve the number of max-flow iteration step below the switching degree of the network. Although, in the worst case, the switching degree of a network is equal to λ that implies $\lambda + 1$ different max-flow computation. There exists instances with any value of the switching degree, that is between 1 and $\lambda + 1$. However, for networks with a low enough switching degree, the restricted flow algorithm can be computed faster. Thus the EMRF method is faster on the average.

For any cut X, define $\mathcal{F}_X$ as the restricted capacity of the cut X.

Proposition 4.1
Let a cut X be a solution to $(i+1)$-MLA-robust cut where $i \in [1..(\lambda - 2)]$. Let x_{i-1} be the abscissa of the breakpoint of degree $i - 1$ of $\mathcal{F}_X$. If $\mathcal{F}_X(x_{i-1}) = \mathcal{F}(x_{i-1})$, then X is also a solution to $(i-1)$-MLA-robust cut and a solution to i-MLA-robust cut.

Proof. Since $\mathcal{F}_X(x_{i-1}) = \mathcal{F}(x_{i-1}$, the point of abscissa x_{i-1} is also a breakpoint of $\mathcal{F}$. Suppose it is not the case. The function $\mathcal{F}_X$ has its $(i-1)$ breakpoint with abscissa x_{i-1} then there exists a non-negative real number ε such that $\mathcal{F}_X(x_{i-1} - \varepsilon) < \mathcal{F}(x_{i-1} - \varepsilon)$. That is not possible by definition of $\mathcal{F}$. Hence X is a $(i-1)$-MLA-robust cut.

Then, X is the solution of $(i+1)$-MLA-robust cut and the continuity of $\mathcal{F}$ implies $\mathcal{F}_X(x_i) = \mathcal{F}(x_i)$ where x_i is the abscissa of the breakpoint of degree i of $\mathcal{F}_X$.

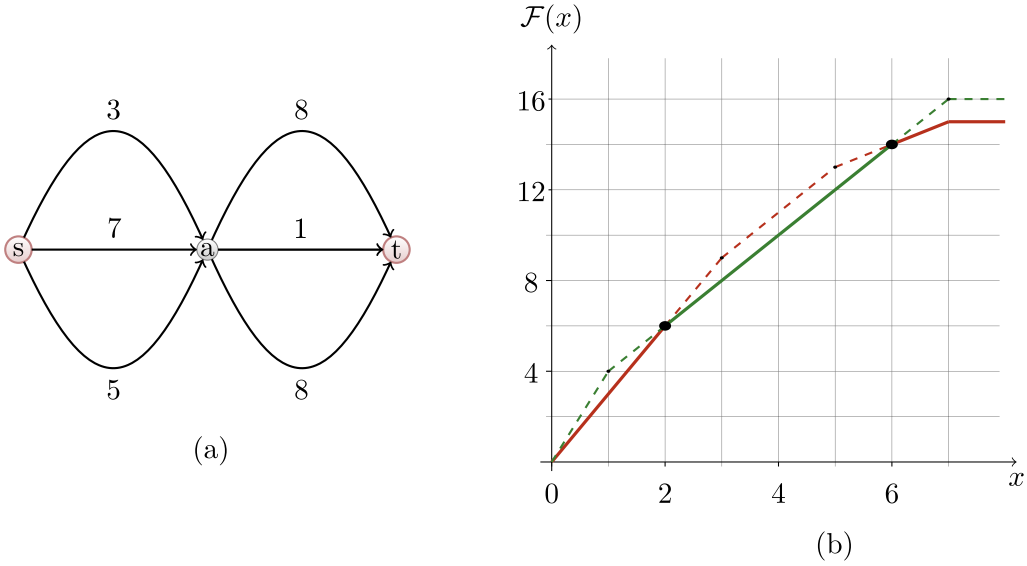


Figure 4.2: Skipping method is not an exact algorithm in the general case when the number of break points skipped is at least 2. For instance, the part of the $\mathcal{F}$ with slope 2 (in green) between the breakpoints $(2, 6)$ and $(6, 14)$ is associated with the cutset separating s from a. If those two breakpoints are skipped, the use of the left cut will not be detected, and $\mathcal{F}$ will be wrongly plotted between $(2, 6)$ and $(6, 14)$ (red dashed part).

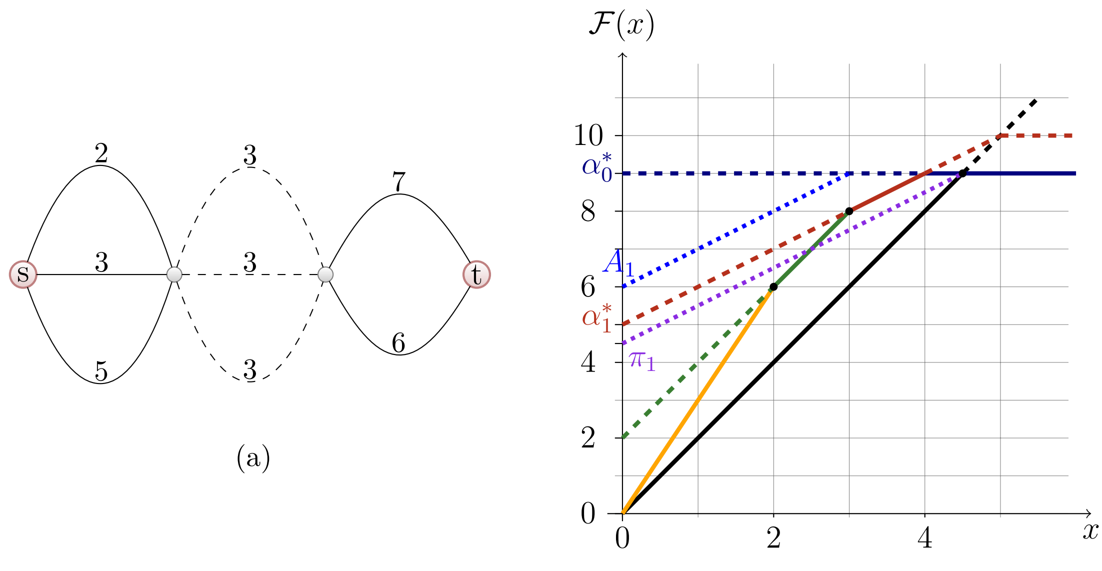
Finally, the two breakpoints of degree $i-1$ and i of $\mathcal{F}_X$ being also the one of $\mathcal{F}$ implies that X is a solution to i-MLA-robust cut. □

The idea consists, at each step, in skipping the computation of some breakpoints by computing the list of breakpoints of a current minimum restricted cut. Let r be the number of breakpoints skipped at each step. As seen in proposition 4.1, this method always gives an exact result when $r = 1$. However, figure 4.2 is a counter-example when $r = 2$, thus for any $r \geqslant 2$.

4.2 Bounds and Flying cut

Instances or networks that are EMRF-solvable can be solved in polynomial time. The other instances or networks, that also include the NP-hard cases, are qualified to possess a flying-cut, as defined below.

Definition 4.2
If an instance of i-MLA-robust flow is not EMRF-solvable, then the associated network is not EMRF-solvable either. This instance is said to have a *flying-cut* (of degree i) as a solution. The network is said to possess a *flying-cut* (of degree i).



(b)

Figure 4.3: Network with a flying cutset of degree 1 and associated bounds. The lower bound is the pseudo-tangent π_1 and the upper bound is A_1 –from the cutset with the dashed links of value 3 in figure 4.3a. The plain line (in black) of slope starting from the origin represents the value of the right with 2 edges. The other plain lines with slope 3 (in orange), slope 2 (in green), slope 1 (in red) represent the value of the of the left cut when it is minimum.

For a given interval corresponding to the slope of degree i in $\mathcal{F}$, the restricted capacity of such a cut is always strictly larger than the capacity of a min-cut on the restricted network. The cut restricted capacity seems to *fly over* the restricted max-flow value. An example is given in figure 4.3 where the left cut of the network flies over the max-flow function on $[3, 4]$, making the 1-MLA-robust flow not EMRF-solvable.

In chapter 3, the max-$(k+1)$-route flow is shown to be a $(k+1)$-approximation to k-MLA-robust flow, giving both a lower and an upper bound. This section introduces a deterministic lower bound referred as pseudo-tangent method. Additionally, the upper bound can be simply searched, randomly or heuristically, using the duality of MLA-robust flow and MLA-robust cut.

The pseudo-tangent idea is the following. If the solution of an i-MLA-robust flow instance is a flying cut, then the line of slope i do not exist in the restricted max-flow function $\mathcal{F}$. Thus the breakpoints of degree i and $i-1$ are coincident. The pseudo-tangent projection is the value that the solution to i-MLA-robust flow is an arbitrary small line of slope i would exist between the two breakpoints.

Definition 4.3

Let G be a graph with connectivity λ and a restricted flow function $\mathcal{F}$. For any integer i, denote π_i the *pseudo-tangent projection* of degree i is formally defined as:

- If $\exists x_i \in \mathbb{R}$ such that $\mathcal{F}(x_i) - i \times x_i = \alpha_i^*$, then $\pi_i = \alpha_i^*$;
- Otherwise there exist unique $j, k \in \{0..\lambda\}$ such that

$$\begin{cases} \mathcal{F}(x_j) - j \times x_j = \alpha_j^*, \\ \mathcal{F}(x_k) - k \times x_k = \alpha_k^*, \\ \mathcal{F}(x_j) - k \times x_j = \alpha_k^*, \end{cases}$$

  then $\pi_i = \mathcal{F}(x_j) - i \times x_j$.

Lemma 4.1

$$\pi_i \leqslant \gamma_i^*$$

where γ_i^* is the i-MLA-reliable flow value.

Proof. Suppose there exists a cut X such that $\pi_i > \gamma_i(X)$. Let $\mathcal{F}_X$ be the restricted capacity function over the cut X. By max-flow/min-cut duality, for any x, $\mathcal{F}(x)$ cannot be bigger than $\gamma_i(X) + i \times x$. Hence,

$$\mathcal{F}_X(x_j) \leqslant \gamma_i(X) + ix_j < \pi_i + ix_j < \mathcal{F}(x_j).$$

So $\mathcal{F}_X(x_j) < F(x_j)$, that is impossible. Thus, for any cut $X \in \mathscr{C}$,

$$\pi_i \leqslant \gamma_i(X).$$

□

The breakpoints of degree 0 corresponds to the classical max-flow and thus 0-MLA-robust flow is always EMRF-solvable. Hence, for any nonnegative number of attacks, lemma 4.1, conjugated with the MLA-robust flow/MLA-robust cut duality leads to the following theorem on upper and lower bounds to both MLA-reliable flow and MLA-robust flow problems.

Theorem 4.1

$$\textit{For any } i \in [0..(\lambda-1)], \quad \pi_i \leqslant \gamma_i^* \leqslant \alpha_i^* \leqslant \min\left(\min_{X \in \mathscr{C}} (\alpha_i(X)), \alpha_{i-1}^*\right).$$

Proof. The left relation, $\pi_i \leqslant \gamma_i^*$, is exactly lemma 4.1. In chapter 3, the central relation, $\gamma_i^* \leqslant \alpha_i^*$ was proved as proposition 3.2.

The nature of MLA-robust flow implies that

$$\forall i \in [0..(\lambda - 1)], \quad \alpha_i^* \leqslant \alpha_{i-1}^*.$$

Finally, the duality of MLA-robust flow/MLA-robust cut from lemma 3.1 implies that

$$\alpha_i^* \leqslant \min_{X \in \mathscr{C}} \left(\alpha_i(X)\right).$$

□

Figure 4.3 is an example of a flying cut, lower bound by the pseudo-tangent method, and upper bound by the duality of MLA-robust flow and MLA-robust cut.

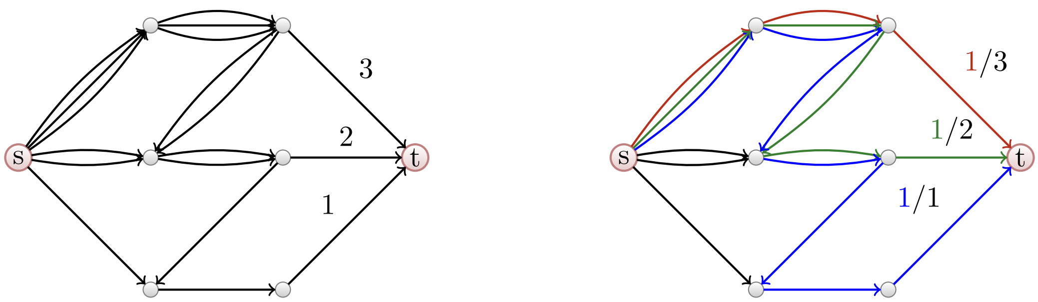
4.3 Completed Multiroute Flow

In section 2.2.2, the structure of multiroute flow (multiroute flow) shows a natural resilience to attacks. Then, in section 3.3, an exact approximation ratio between MLA-network flows and multiroute flow is detailed. However, a solution to the multiroute flow is not a solution to max-flow in the general case. Once a multiroute flow, $\mathscr{M}$, is computed, the residual graph might still be connected. In that case, another max-flow or set of flows can be computed and added to $\mathscr{M}$ to form a completed multiroute flow (completed-MRF).

However, the solution found to the multiroute flow might hinder its own completion, making its value lower than a classical max-flow. Section 4.3.1 covers an example that gives a first, relatively bad, lower bound to the maximum ratio between a completed-MRF and a max-flow values. Despite this bound, a completed-MRF based on an iterative completion by successive multiroute flow –with a decreasing number of routes– is designed in section 4.3.2.
The practical efficiency of completed-MRF is evaluated in section 5.4.

4.3.1 Definitions and bounds

Definition 4.4
For any nonnegative integer p, let $H_p = (V, E)$ be a network where $V = \{a_1, \cdots, a_p\} \cup \{b_1, \cdots, b_p\} \cup \{s, t\}$ and E is constructed as follows. For any integer $i \in \{1..p\}$:



(a) (b)

Figure 4.4: Network with a ratio of $\frac{\lambda-1}{2}$ between completed max-λ-route flow and classical max-flow.

- i arcs of capacity 1 from s to a_i,
- i arcs of capacity 1 from a_i to b_i,
- $i-1$ arcs of capacity 1 from b_i to b_{i-1} – except when $i=0$,
- 1 arc of capacity i from b_i to t.

Figure 4.4a represents the graph H_3. For any integer λ, the network H_λ is such that a max-λ-route flow that saturates a whole cut can be computed (as in figure 4.4b). In that case the ratio between classical max-flow and max-λ-route flow values can be more than $\frac{\lambda-1}{2}$.

4.3.2 Iterative Multiroute Flow

The previous section showed that there can be a significantly large gap between the values of the solutions to the max-$(k+1)$-route flow and to the MLA-reliable flow. This gap is reduced in this section by proposing a tighter heuristic algorithm called the IMRF (iterative multiroute flow).

The basic idea behind this new method is due to the ratio between multiroute flow and MLA-network flows detailed in section 3.3.2. For a given number of attacks k, the more h, the number of route is closer to k, the more the multiroute flow value tends to be close to its effectiveness. However, for a given number of route, several multiroute flows with different effectiveness can exist. The value of the multiroute flow can be small, relatively to its effectiveness and MLA-network flows values as seen in section 3.3. The IMRF is constructed to make use of the value h being close to the connectivity λ of the network.

input : A network graph $G = (V, E, c)$, where $c : E \to \mathbb{R}^+$ is the capacity of each edge

output : The flow on each edge $F : E \to \mathbb{R}^+$

Let λ be the source-sink edge connectivity of graph G.
Assign $F(e) \leftarrow 0$ for all $e \in E$.
for $h \leftarrow \lambda$ **to** 1 **do**
    Find the h-route flow of graph G.
    Let $F_h : E \to \mathbb{R}^+$ be the flow on each edge in that h-route flow.
    $F(e) = F(e) + F_h(e)$.
    $c(e) = c(e) - F_h(e)$.
end

Algorithm 3: iterative multiroute flow (IMRF) algorithm [10]

This new method consists in computing iterative max-h-route flow for h decreasing by one from λ to 1. The first step where $h = \lambda$, computes a max-h-route on the original network. At each step $h < \lambda$, the next max-h-route flow will be computed on the residual network from step $h + 1$. The IMRF is then the linear combination of each of those max-h-route flow. It can be seen as an iterative method to construct completed multiroute flows. However, as discussed in section 4.3.1, this approximation can be at least as bad as $\frac{\lambda-1}{2}$. This method is presented as algorithm 3 and evaluated in section 5.6.

Theorem 4.2 ([10])
The worst case time complexity of IMRF algorithm is $O(\lambda^2 T)$, where λ is the source-sink edge connectivity the network, and T is the computation-time complexity of the max-flow algorithm.

Proof. The method iterates over h from λ down to 1, the number of executions of the max-flow subroutine is

$$\sum_{h=1}^{\lambda}(h+1) = \frac{(\lambda+1)(\lambda+2)}{2} = O(\lambda^2).$$

□

4.4 Classification of MLA instances

Through chapters 3 and 4, various properties of the MLA instances have been analyzed and resumed, among others, in figure 4.5. EMRF-solvable defines a sufficient condition for

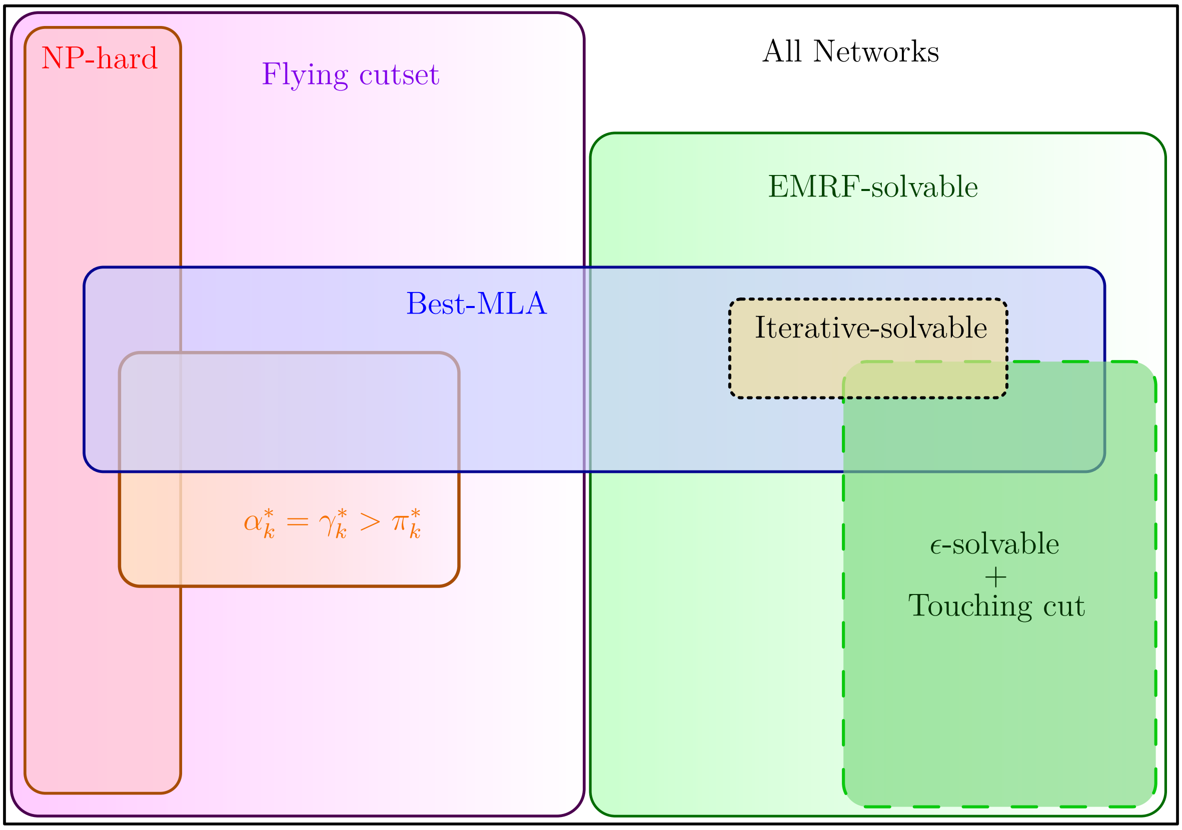
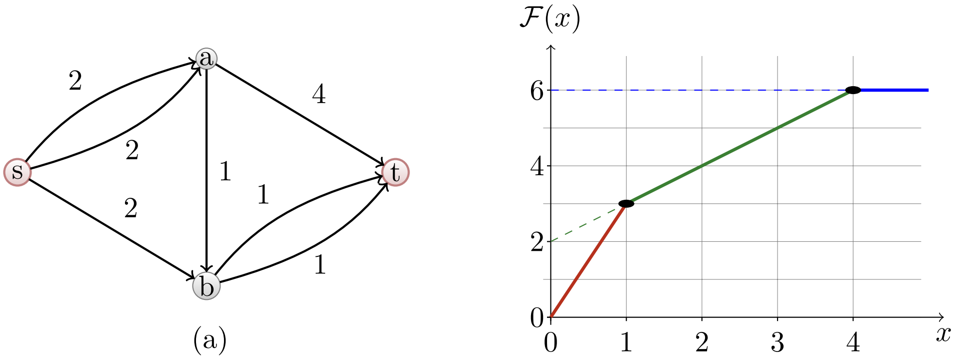


Figure 4.5: Class diagram

an instance to be solvable in polynomial time. Flying-cut, its counterpart, characterizes a necessary condition for an instance to be NP-hard. The EMRF-solvable class has been enhanced by two other methods. First is the capacity-differentiation that prevents when the EMRF algorithm fails due to the equality between two capacities of the links. In that case, the fact that the instance (or network) is solved is detected. The second method is the pseudo-tangent approach that also solves the problem of capacities equality, but without being detected as solved –except if an upper bound equal to the pseudo-tangent is found. However, this method also gives a deterministic lower bound in logarithmic time in the connectivity once the EMRF is computed.

Other cases have been represented in figure 4.5, as the existence of a best-MLA flow described in section 4.4.1. Another class that might be analyzed in future works is when both MLA-robust flow and MLA-reliable flow values are equals but strictly larger than the pseudo-tangent –if not, it would be the EMRF-solvable class.



(a)

(b)

Figure 4.6: A counter-example to the existence of best MLA-flow: the MLA-reliable flow when $k = 0$ force the arc (a, t) to have a flow of value 4; when $k = 1$, the arc (a, b) must have a flow of value 1, forcing the arc (a, t) to have a flow value of at most 3.

4.4.1 Optimal version using restricted max-flow

An ideal flow in this attack context is a flow that would be solution to both MLA-robust flow and MLA-reliable flow on the network for any value of k. This flow would be a kind of best-MLA flow, and the class of networks possessing one are referred as such in figure 4.5. However, this is impossible in the general case as shown by the counter-example of figure 4.6. The class can be furthered reduced to the case of a best-MLA that would also be an IMRF –referred as IMRF-solvable in figure 4.5. In many cases, if proving a network is IMRF-solvable might not be more relevant than proving it to have a best-MLA flow, it might be used to synthesize a best-MLA flow from a set of constraint that are progressively met during the completion of each step of the IMRF algorithm.

4.5 Conclusion

In this chapter, various improvement concerning the EMRF have been provided along with the algorithm to compute some lower and upper bounds for MLA flows. Also, a classification of different types of MLA instances is given. Both the exact polynomial time algorithms and the lower and upper bounds methods will be combined into the mixed-MLA flow algorithm in chapter 5. This new method will be evaluated on various criteria through experimental results.

Chapter 5

Applications and experiments

The MLA-robust flow and MLA-reliable flow can be solved in polynomial time through the extended multiroute flow algorithm (EMRF). However, as the NP-hardness of MLA-robust flow shows it, this algorithm does not success in the general case. In chapter 4, this algorithm is widened to a larger class of instances. More precisely, the NP-hard instances are shown to belong to the *flying-cut* class. This part covers various experiment about the extended multiroute flow algorithm, brute force MLA-flow algorithm, and the more general mixed-MLA algorithm. The latter is described in section 5.2 and mixed the EMRF –possibly including capacity differentiation– with the different upper and lower bounds method described in section 4.2.

Those experiments are conducted from section 5.3 onwards on randomly generated networks as described in section 5.1. They mainly follow two axes. First comes the evaluation of both EMRF and mixed-MLA algorithms through various kind of networks structures: random shape with different capacity distribution, complete networks, 2D and 3D grids, and complex networks generated through the R-MAT method. The second axis corresponds to the evaluation and comparison of different methods proposed though this thesis on a smaller set of network structures: random shape and R-MAT networks. Additionally, the average robustness gain of a completed-MRF over classical max-flow is estimated through a greedy attacks algorithm in section 5.4.

5.1 Generation of the network instances

In all the experiments, any generated network will have its number of nodes referred as n, its number of links as m, and its link-connectivity as λ. The number of networks generated

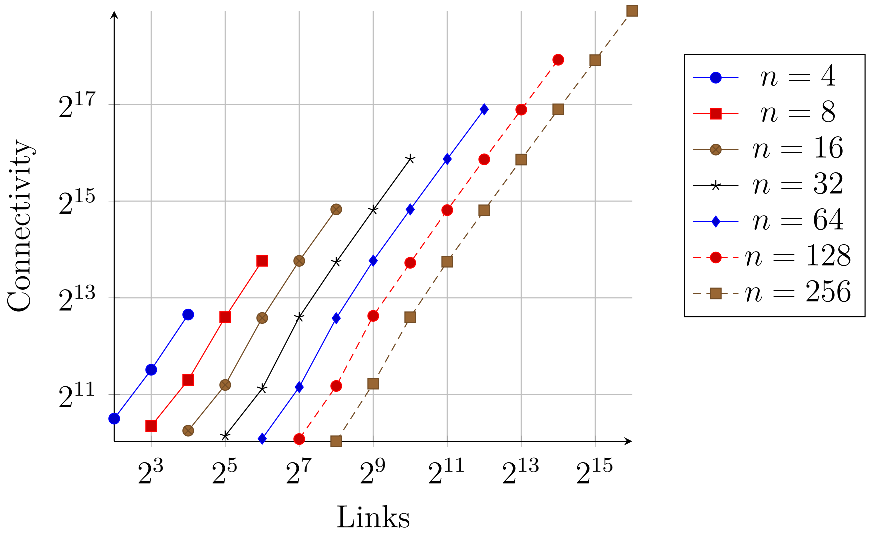


Figure 5.1: Average connectivity on almost-randomly generated networks.

for each setup of the benchmark (nodes, links, structure, links capacity distribution, $\cdots$) is always 1000. The number of nodes and links are chosen such that $m \times n \leqslant 2^{24}$.

A *complete network* is such that each pair of nodes is connected exactly twice (once for each direction). The number of node n varies in $\{2^p | p \in [2..8]\}$, and $m = 2(n-1)$. The connectivity λ is always equal to $n - 1$. The capacities of each link is generated randomly in $]0, 1]$.

A *grid* is a network such that each node is connected to two neighbors along each dimension of the grid. The connectivity λ is equal to the double of the dimension (here, 4 or 6). For 2D grids, the number of node n_0 on each dimension varies in $\{2^p | p \in [2..6]\}$, then $n = n_0^2$ and $m = 2n$. For 3D grids, the number of node n_0 on each dimension varies in $\{2^p | p \in [2, 3]\}$, then $n = n_0^2$ and $m = 2n$.

The *randomly generated shape network* –referred for simplicity here as random shape network– are all generated following the same rule, independently of the capacity distribution of the links. The number of node n varies in $\{2^p | p \in [2..8]\}$, then the number of links m varies in $\{2^q | q \in [p..p^2]\}$. To have a network with higher source-sink connectivity, each node is chosen with probability $\frac{1}{n}$ to be the tail endpoint of an edge. Exceptions to this are the source node and sink node, where the probabilities are $\frac{2}{n}$ and 0, respectively. Similarly, the probabilities that the source node, the sink node, and the other nodes are chosen to be a head endpoint are 0, $\frac{2}{n}$, and $\frac{1}{n}$, respectively. Figure 5.1 shows the average connectivity of the generated random shape network, that is above the value expected in the uniformly generated shape case. The capacities are then generated in $]0, 1]$ following five different kind of distributions: uniform; gauss with

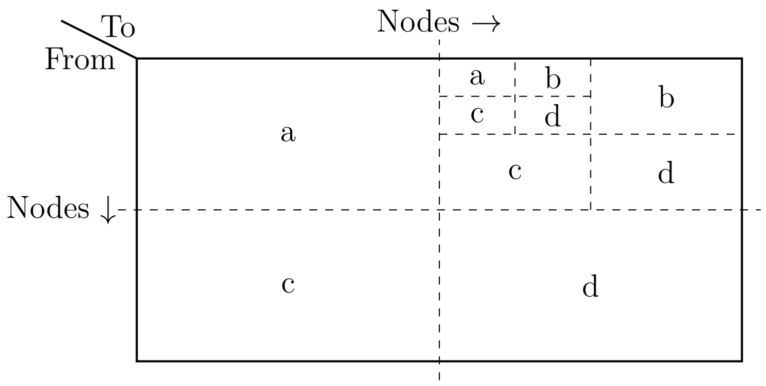


Figure 5.2: The R-MAT model as described by Chakrabarti, Faloutsos, and Zhan [28].

value 0.5 and variance 0.1 – similar to the Dirac impulse in Dirac [38]; gauss with value 0.5 and variance 0.25; power law as in [46] (exponential with density equal to 2); 1− power law (exponential with density equal to 2).

5.1.1 R-MAT generation method for complex networks

The R-MAT generator can successfully describes many pattern of *complex network* as described by Chakrabarti, Faloutsos, and Zhan [28]. The idea is as follows, given a number of nodes n, generate a $n \times n$ matrix M of integers with 0 as initial value. Given four real numbers a, b, c, and d such that, typically,

- $a + b + c + d = 1$,
- $a \geqslant b$, $a \geqslant c$, $a \geqslant d$,
- $b \geqslant d$, $c \geqslant d$,

the matrix M is then divided recursively into equal sized squares as in figure 5.2. The four values a, b, c, and d represents the probability to choose an area. Then, for a given link, an area is chosen randomly following the four probabilities, and recursively, in the same way, till the area chosen correspond a unique pair of line-column in the matrix. Increase by 1 the value on this element of M. The procedure is then repeated as many time as needed.

In the experiments, instead of generating a instances with a specific number of links, the R-MAT instances are generated with a specific total sum of the capacities of the links. The number of links cannot be higher than n^2, and will be much lower on the

input : A network graph $G = (V, E, c)$, where $c : E \to \mathbb{R}_{\geqslant 0}$ is the capacity of each edge.

output : A vector of pairs $(a_k, A_k)_{k \in 0..\lambda}$ where a_k is a lower bound of γ_k^* and A_k is an upper bound of α_k^*.

1 Vector $v \leftarrow extendedMultirouteFlow$; $C \leftarrow MLA - robustCut$.
2 **for** $k \in [1..(\lambda - 1)]$ **do**
3 $\quad v[k].a \leftarrow pseudoTangent(G, k)$;
4 $\quad v[k].A \leftarrow \min(v[k-1].A, \min_{X \in C}(\alpha_k(X))$;
5 **end**

Algorithm 5: General Scheme of mixed-MLA flow algorithm

average. The source and sink are chosen are nodes corresponding to 1 and 2 in the sense of the generating matrix M. Thus, the connectivity between both is the highest in the graph on the average. Described as a typical setting in [28], the values of the parameters of the R-MAT method here are the following: $a = 0.5$, $b = 0.2$, $c = .2$, and $d = 0.1$.

5.2 Mixed Algorithm for MLA-Flow

Following the results of chapters 3 and 4, an algorithm mixing the different results and optimizations is designed. This algorithm is called here mixed algorithm. Its features includes a set of upper bounds, a lower bounds based on the pseudotangent projection property, and uses the extended multiroute flow algorithm as a base.
As described in algorithm 5 –line 1–, the mixed-MLA algorithm starts by computing the set of breakpoint of the network by using the extended multiroute flow algorithm (EMRF). More precisely, a variation called extended multiroute cut algorithm is used to memorize a set of associated k-MLA-robust cut. The breakpoints of any degree of those cuts are always successfully computed.
In a second time, – line 3 – the algorithm provides the pseudotangent π_k as a lower bound of the MLA-reliable flow value γ_k^* – thus lower bound of α_k^*– for each $k \in [1..(\lambda - 1)]$. Finally, for each value of $k \in [1..(\lambda - 1)]$, an upper bound is chosen as the minimum between all the $\alpha_k(X)$ and α_{k-1}^* for any X in the previously computed set of k-MLA-robust cuts.
The results of the algorithm is then a vector of pairs with upper and lower bounds for each value of k. If the upper and lower bounds are equal for a given k, then k-MLA-robust flow is polynomial time solvable for this instance. If the upper and lower bounds are equal for all the value of k, then MLA-robust flow is polynomial solvable for this network.

This algorithm should be improved in the future by the incorporation of deeper analyses, such as the use of the link-parametric multiroute flow algorithm.

5.3 Comparison between MLA-reliable flow and MLA-decomposition

This section covers a comparison between MLA-reliable flow and MLA-decomposition on small instances and presented by Baffier, Dai, and Suppakitpaisarn [12]. The complexity discussion in section 3.2 suggests that both problem might be difficult. For that reason, the comparison between both is evaluated on small instances using linear programs 3.1 and 3.2.

The experiments setup is as follows: 3000 random-shape networks with $n = 4$. Among them, 1000 networks have $m = 4$, 1000 networks have $m = 8$, and the other 1000 networks have $m = 16$. Although the instances are quite small, MLA-reliable flow and MLA-decomposition values are equals in all of those 3000 networks. This indicates that cases when two problems are different are unlikely to happen, when networks are small and randomly generated. Attackers with the knowledge of flow decomposition might not be able to find a better attack, if the flow decomposition is optimally selected.

5.4 Effect of Greedy attacks on max-flow and completed-MRF

As discussed in section 4.3, completed-MRF can be a relatively bad approximation. However, the natural robustness properties of the multiroute flow might help into finding a more resilient flow for the average case. This experiment aims to compare the k-effectiveness obtained from a completed max-$(k+1)$-route flow with the k-effectiveness of a classical max-flow algorithm. when the number of attacks is $k = \lambda - 1$. Although, a maximum multiroute flow and a max-flow of each network can be computed efficiently, finding the optimal attack for those flows is shown to be NP-hard by Baffier, Dai, and Suppakitpaisarn [12]. However, the problem can be considered as maximizing a monotone submodular function subjects to the cardinality constraint k. Thus, that optimal attacks can be approximated with solutions from an $(e-1)/e \approx 0.63$-approximation algorithm proposed by Nemhauser, Wolsey, and Fisher [80].

The improvement of the completed-MRF over the classical max-flow is shown in figure 5.3b. The improvement is at least 15%. When the number of edges is small

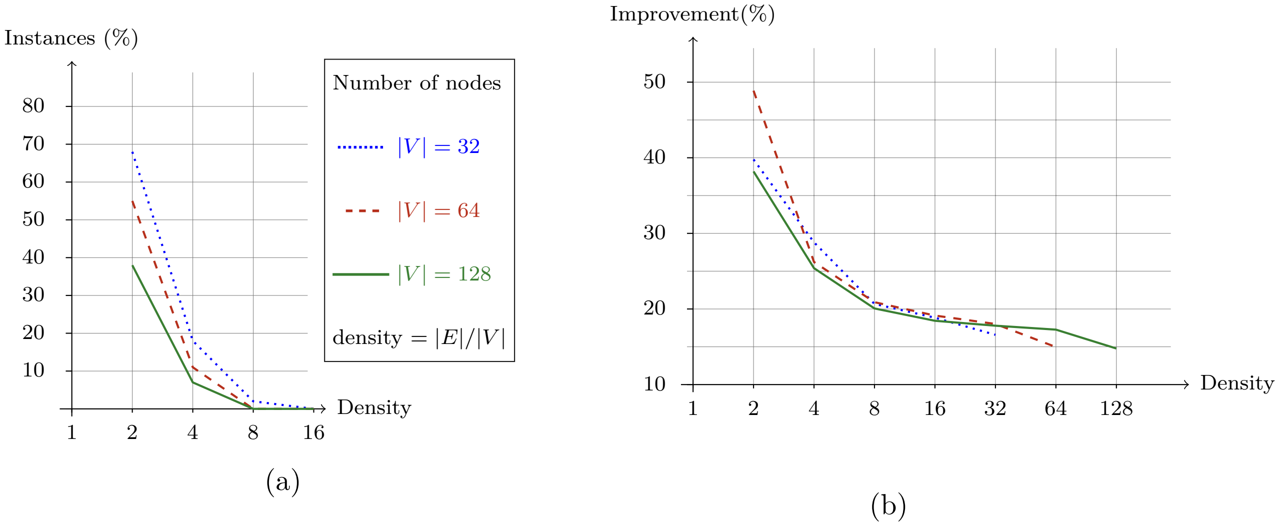


Figure 5.3: Evaluation of completed-MRF solutions with classical max-flow solutions. The attacks are computed greedily and number $k = \lambda - 1$. By construction, a completed-MRF solution cannot be completely disconnected by those k attacks. Those experiments results are classified by the number of nodes of the network (different plots) and by the number of edges: density $= |E|/|V|$.
(a) Percentage of classical max-flow instances completely cut by greedy attacks.
(b) Improvement of the remaining flow by the max-λ-route flow compared to the classical max-flow.

compared to the number of nodes, that improvement tends to be larger. The improvement is at least 37% when $m = 2n$.

Beside this average result, the completed-MRF method outperforms the classical max-flow algorithm in all $18,000$ networks and all possible value of k. Moreover, the attacker can completely cut some of the networks, when information is carried using classical max-flow. The percentage of those instances is shown in figure 5.3a. When $|V| = 128$ and $|E| = 256$, the max-flow instances completely cut is as large as 38%.

5.5 Success rate of the EMRF algorithm

This section covers the success rate and distribution of failure through the networks and instances. The success of EMRF algorithm is evaluated for complete networks (uniform capacity distribution) in section 5.5.1, R-MAT complex networks (integral capacities) in section 5.5.3, $2d$ and $3d$ grids (uniform capacity distribution) in section 5.5.2. The case of randomly generated shape (almost-uniform link distribution) is provided in section 5.5.4.

It covers different distribution of the capacities (as described above) and comparison in function of the distribution and in function of the number of nodes. All the networks are generated following the protocol in section 5.1.
The term *instance* refers to a couple network and number of attacks k and corresponds to the solving of k-MLA-robust flow and k-MLA-reliable flow. In the other, the term *network* refers to an instantiated network without any number of attacks associated. Thus it correspond to the solving the MLA flow problems for all the possible number of attacks.

5.5.1 Complete Graph

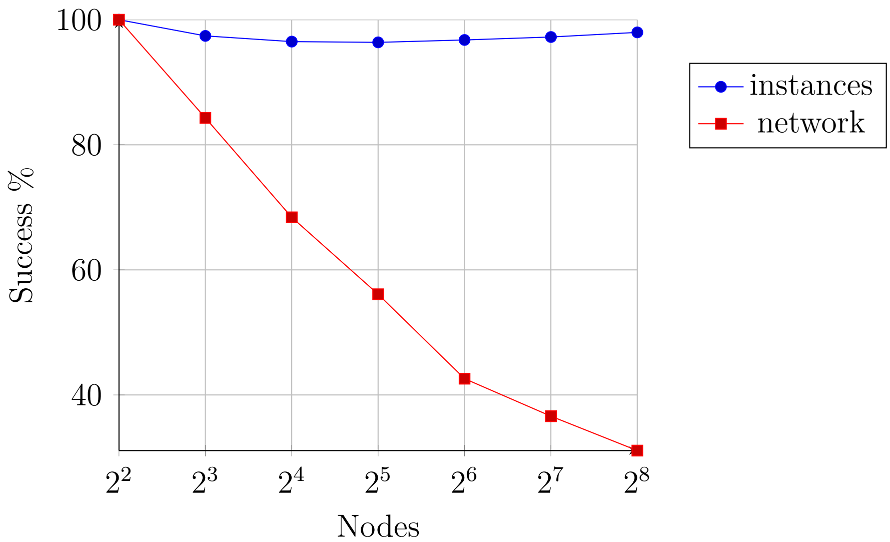


Figure 5.4: Success rate of EMRF algorithm on complete networks.
The number of links in a complete network is about the square of the number of nodes. The latter one takes the values of the power of 2 between $2^2 = 4$ and $2^8 = 256$ and is represented by a logarithmic scaled axis.

As it shown in figure 5.4, the success rate of EMRF algorithm on instances is never below 96% and can be considered as almost constant. Remark that the connectivity is linearly growing in function of the number of nodes ($\lambda = n - 1$), and at the same time the success rate of networks solved decreases exponentially. Figure 5.5 shows that failures tend to be concentrated on a number of attacks close to the connectivity. As seen in section 3.3.2, the approximation ratio of multiroute flow tends to be better when k is near λ. Most of those failures occurs a small amount of times in each network with a flying cutset as shown by figure 5.6.

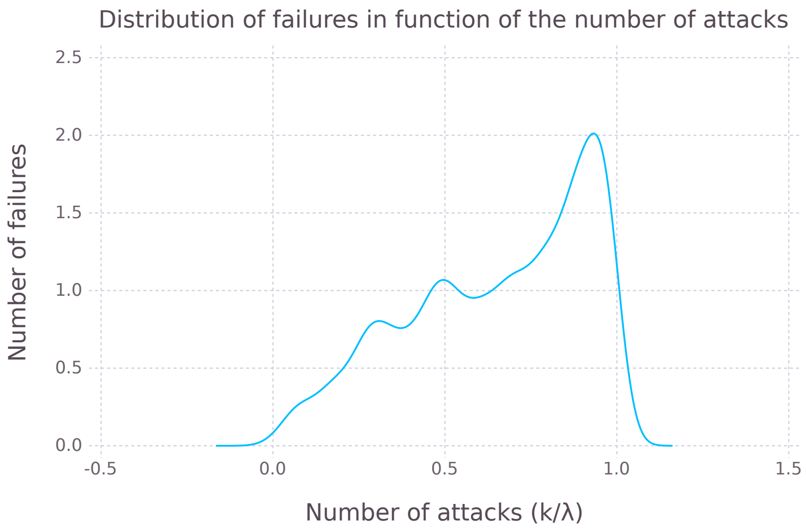


Figure 5.5: Distribution of failures of the EMRF algorithm applied to complete networks. The failures are distributed relatively to the number of attacks $k \in [0..\lambda - 1]$ where $\lambda = n - 1$ is the source-sink edge-connectivity. The plot is a density function estimated through the data.

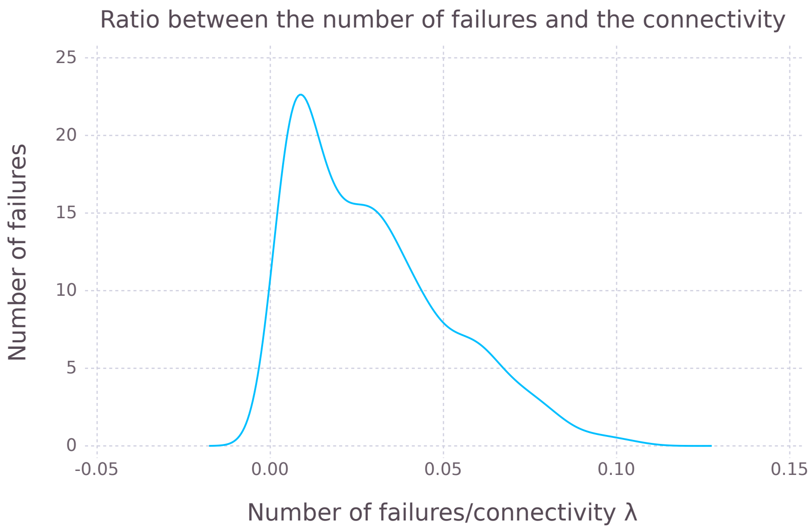


Figure 5.6: Distribution of the ratio between number of failures of the EMRF algorithm for complete networks and their connectivities.
The plot is a density function estimated through the data.

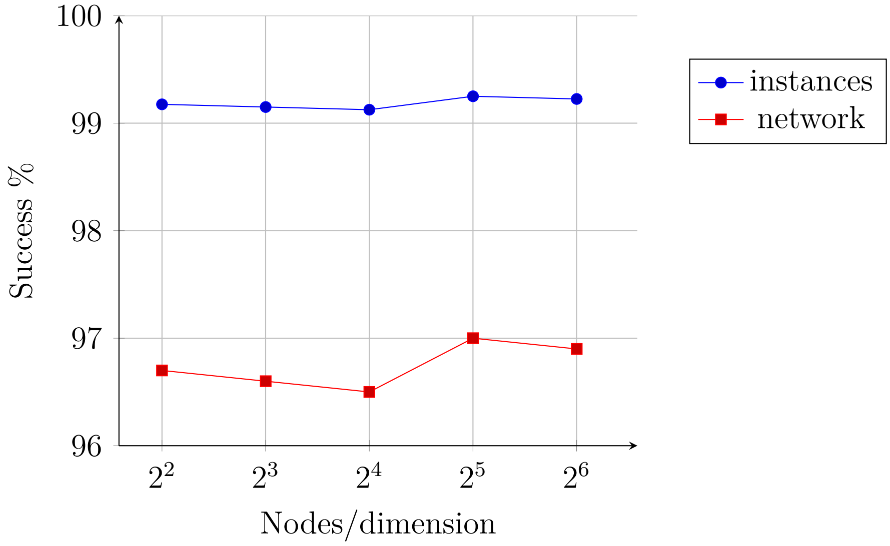


Figure 5.7: Success rate of EMRF algorithm on 2D grids.
The number of nodes by dimension takes the values the values of the power of 2 between $2^2 = 4$ and $2^6 = 64$ and is represented by a logarithmic scaled axis. The connectivity of a 2D grid is always 4.

5.5.2 Grid

The connectivity in networks grids is constant for a given number of dimensions. For 2D grids, $\lambda_{2D} = 4$; and for 3D grids, $\lambda_{3D} = 6$. The success rate of EMRF on both 2D and 3D grids seems independent of the size of the grid, whether it be for instances or full networks. Figure 5.7 shows that the number of instances –respectively networks– solved is always more than 99% –respectively 96%. Concerning the distribution of failure over the number of attacks, figures 5.8 and 5.11 show that, for 2D grids, failures occurs mainly when the number of attack is either 1 or 2. For 3D grids, the failures repartition is consistent whatever the number of nodes, $\frac{1}{3}$ around 0.5, $\frac{4}{5}$ lower or equal 0.5. Finally, figures 5.9 and 5.12 show that failures occur at most once in 2D grids. For 3D grids, 90% of the failing networks have one failure and 10% have two failures.

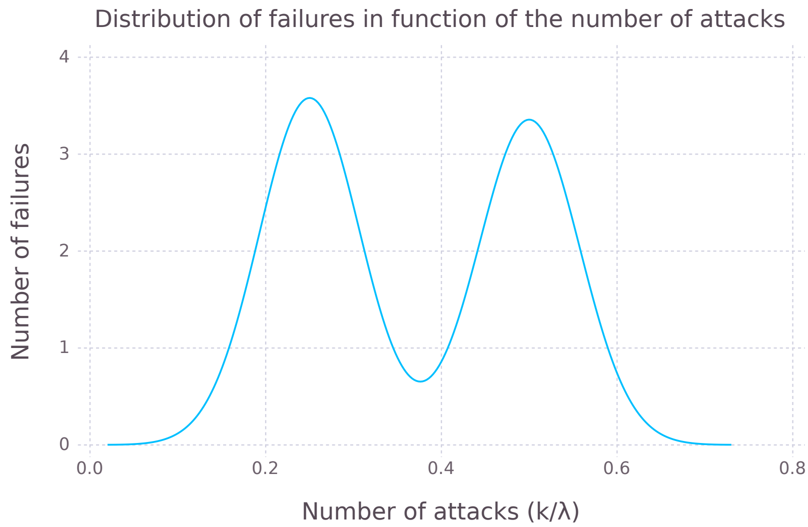


Figure 5.8: Distribution of failures of the EMRF algorithm applied to 2D grids.
The failures are distributed relatively to the number of attacks $k \in [0..\lambda - 1]$ where $\lambda = 4$ is the source-sink edge-connectivity. The plot is a density function estimated through the data. Because of the relatively small value of λ, the density can be considered in a discrete way: when $\frac{k}{\lambda} = \{0.25, 0.5, 0.75\}$.

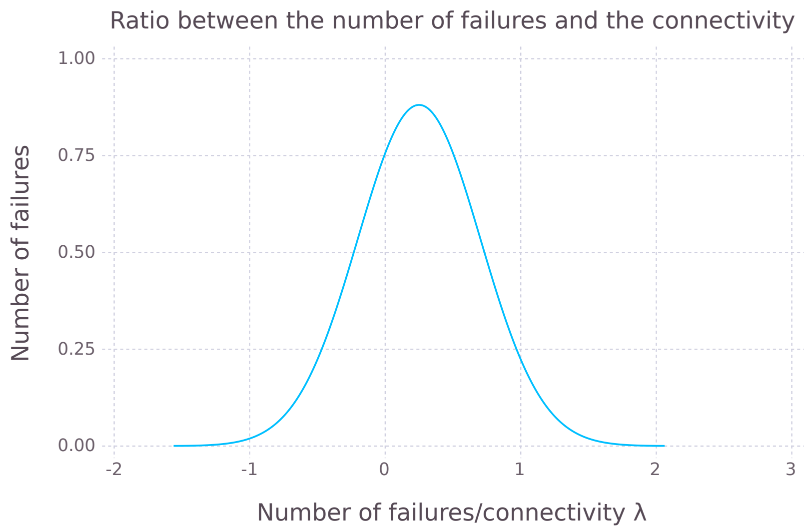


Figure 5.9: Distribution of the ratio between number of failures of the EMRF algorithm for 2D grids and their connectivities.
The plot is a density function estimated through the data. Because of the relatively small value of λ, the density can be considered in a discrete way: when $\frac{k}{\lambda} = \{0.25, 0.5, 0.75\}$. In those experiments the EMRF algorithm only fails for one value of k for the 2D grids with a flying cut.

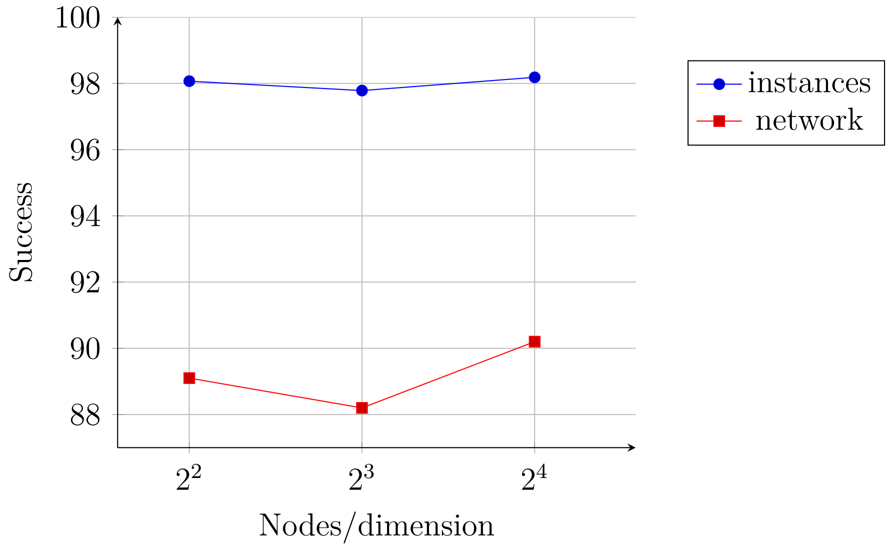


Figure 5.10: Success rate of EMRF algorithm on 3D grids.
The number of nodes by dimension takes the values the values of the power of 2 between $2^2 = 4$ and $2^4 = 16$ and is represented by a logarithmic scaled axis. The connectivity of a 3D grid is always 6.

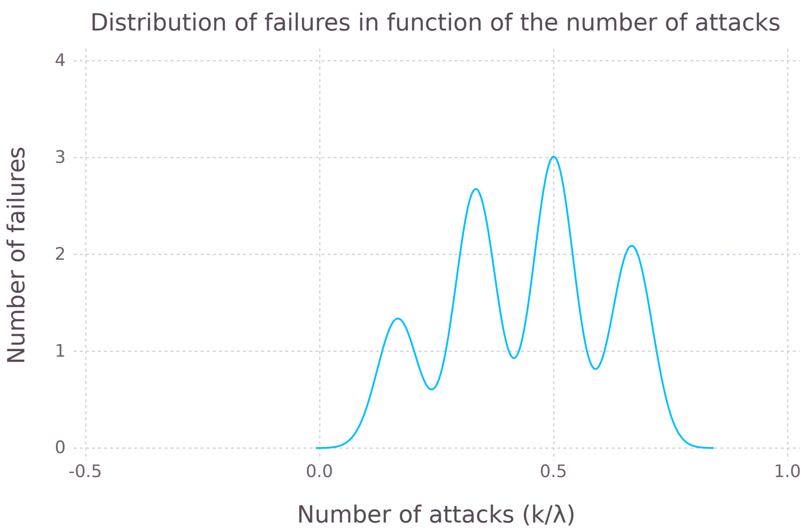


Figure 5.11: Distribution of failures of the EMRF algorithm applied to 3D grids.
The failures are distributed relatively to the number of attacks $k \in [0..\lambda - 1]$ where $\lambda = 6$ is the source-sink edge-connectivity. The plot is a density function estimated through the data. Because of the relatively small value of λ, the density can be considered in a discrete way: when $\frac{k}{\lambda} = \{\frac{1}{6}, \frac{2}{6}, \cdots, \frac{5}{6}\}$.

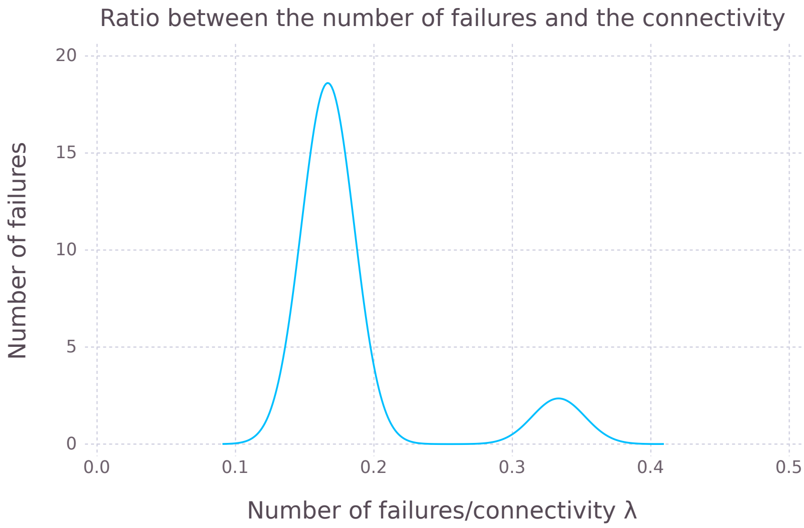


Figure 5.12: Distribution of the ratio between number of failures of the EMRF algorithm for 3D grids and their connectivities.
The plot is a density function estimated through the data. Because of the relatively small value of λ, the density can be considered in a discrete way: when $\frac{k}{\lambda} = \{\frac{1}{6}, \frac{2}{6}, \cdots, \frac{5}{6}\}$.

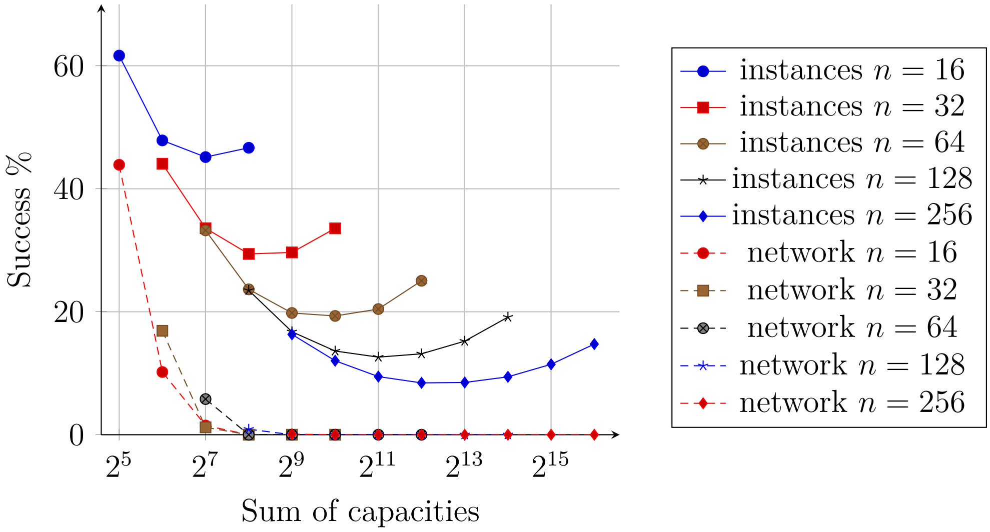


Figure 5.13: Success rate of EMRF algorithm on R-MAT generated networks.
The number of nodes takes the values the values of the power of 2 between $2^4 = 16$ and $2^8 = 256$. The total sum the links capacities –which are integral– takes the values of the power of 2 between $2^5 = 32$ and $2^16 = 65536$. The plain lines represents the success rate on instances and the dotted lines on networks.

5.5.3 R-MAT

The instances and networks generated by the R-MAT method are relevant for their representation of complex networks. However, as shown by figure 5.13, the success rate of EMRF on instances is relatively low: between 20% and 60% for networks with a number of nodes lower than 64, between 15% and 20% when $n = 128$, and can be lower than 10% when $n = 128$. For a total sum of the capacities big enough, there are no networks that can be solved entirely by EMRF algorithm. As discussed in section 4.1.1, some of the failures might come from links with equal capacities. Figure 5.14 shows the improvement of the capacity differentiation method on the EMRF. Those improvement are significant for small networks and show almost no difference when $n \geqslant 128$.

Finally, figure 5.15 shows that the repartition of failures in the networks is uniform. Thus, the multiroute flow is likely not a good approximation for MLA-robust flow on R-MAT networks, at the opposite of experiements in section 5.4.

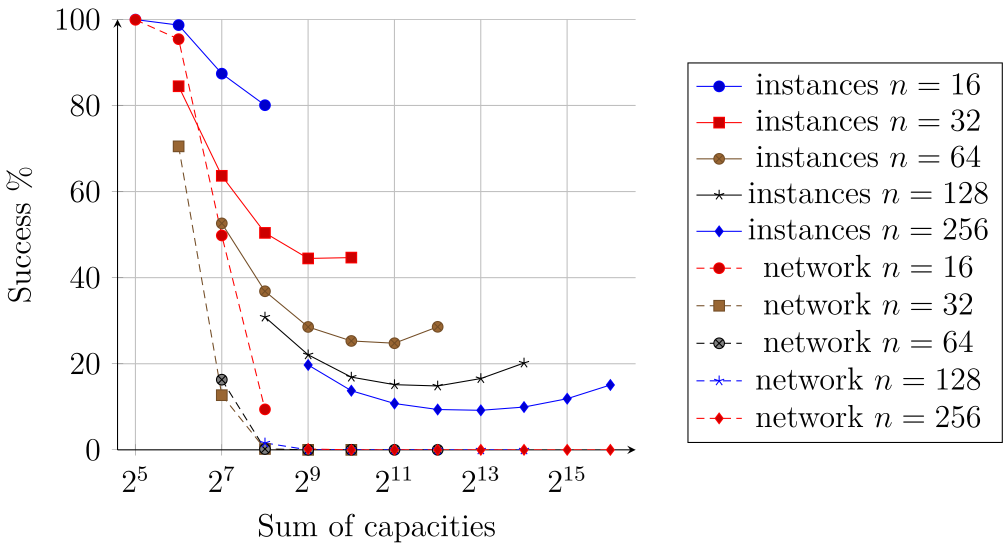


Figure 5.14: Success rate of EMRF algorithm with capacity differentiation on R-MAT generated networks.
The number of nodes takes the values the values of the power of 2 between $2^4 = 16$ and $2^8 = 256$. The total sum the links capacities –which are integral– takes the values of the power of 2 between $2^5 = 32$ and $2^16 = 65536$. The plain lines represents the success rate on instances and the dotted lines on networks.

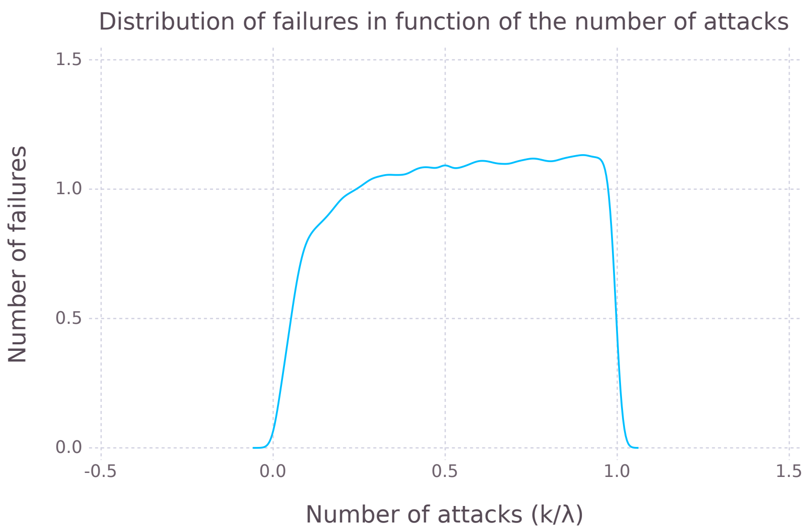


Figure 5.15: Distribution of failures of the EMRF algorithm applied to R-MAT generated networks.
The failures are distributed relatively to the number of attacks $k \in [0..\lambda - 1]$ where $\lambda = n - 1$ is the source-sink edge-connectivity. The plot is a density function estimated through the data.

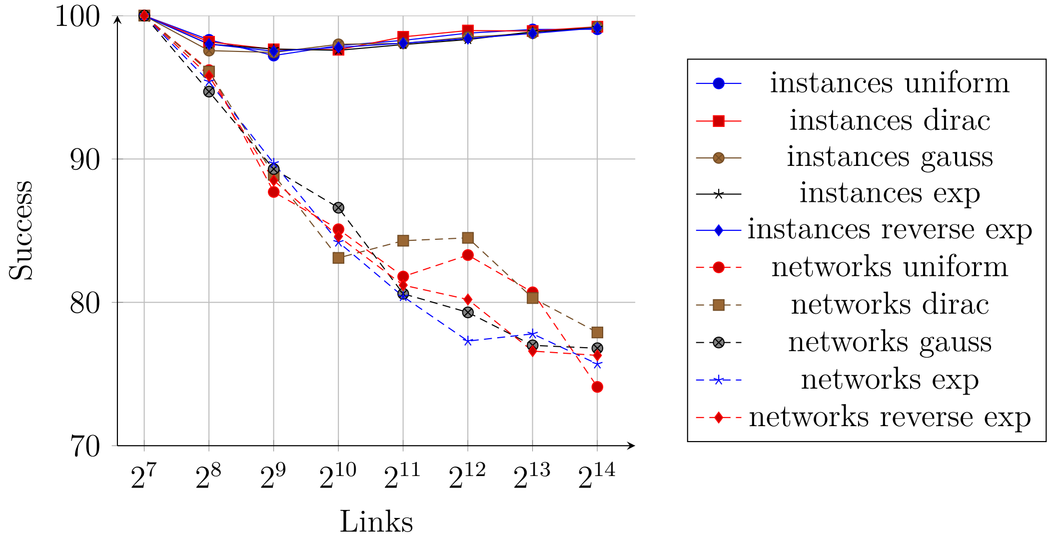


Figure 5.16: Success of EMRF algorithm on almost-random shape networks with 128 nodes.

5.5.4 Randomly generated network

Since the layout of the capacity in the network has a huge impact on the cutsets, the experiments in this section compare the behavior of the EMRF algorithm on random-shape networks with different distributions of the capacities. For any number of nodes tested in those experiments, the behavior of the algorithm is similar for any distribution. In the case of the instances, the success rate of the EMRF, shown for $n = 128$ and $n = 256$ in figures 5.16 and 5.17, never goes below 97% and is close to 99% when $m = n^2$. The success rate on networks is more variable in function of the distribution. Since no distribution is better or worst than the other for the setups with a different number of nodes, it is believed that the impact of the distribution of the capacities is negligible.

Figure 5.18 shows the behavior of the EMRF algorithm on random-shape networks with a uniform distribution of the capacities in function of the number of nodes. As mentioned before, the shape of those results is similar for the other distributions. The plot of the success rate for networks is pseudo-linear on a logarithmic scale, and can be considered as decreasing exponentially fast in function of the number links.

The repartition of the failures between all the possible values of k is almost uniform for all the distributions, as shown in figure 5.19. In that sense, it is different from the complete networks case where the failures where concentrated on high values of k. The multiroute flow approximation ratio cannot be predicted from those experiments. The ratio of failures by networks presented in figure 5.20 tends to confirm that the different

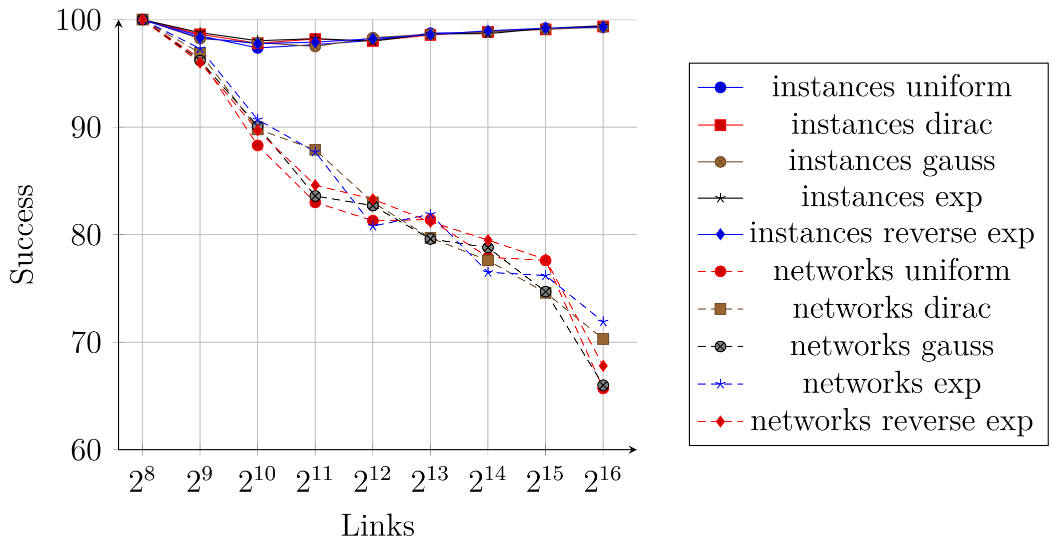


Figure 5.17: Success of EMRF algorithm on almost-random shape networks with 256 nodes.

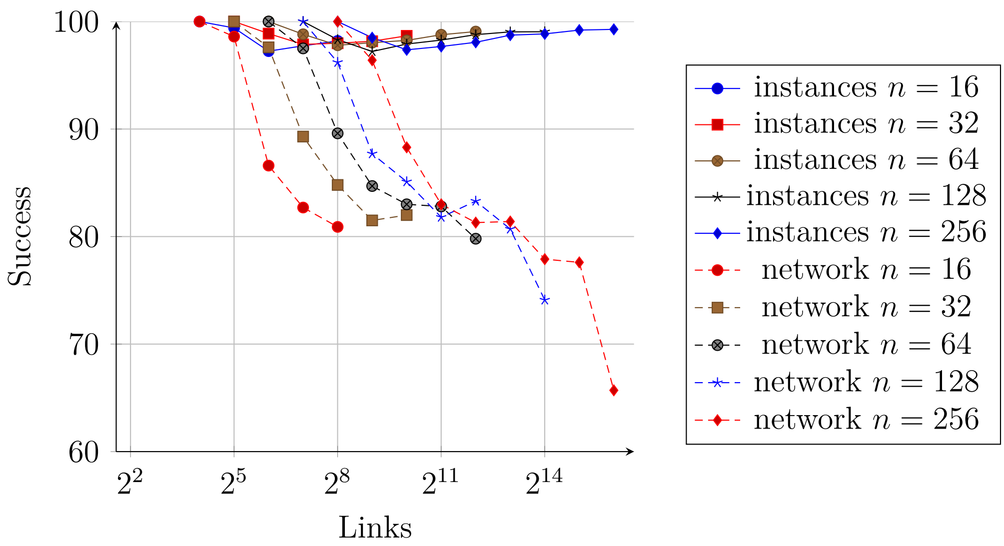


Figure 5.18: Success of EMRF algorithm on almost-random shape networks with uniform capacity distribution in function of the number of nodes.

distribution of the capacity of the links has a very low impact on the success of EMRF.

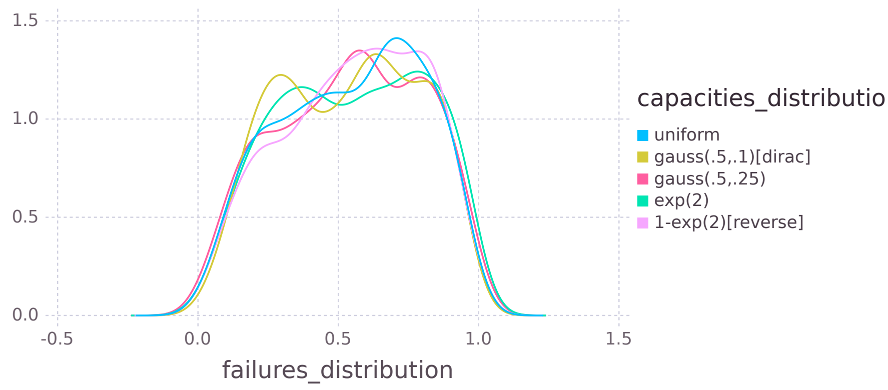


Figure 5.19: Distribution of failures of the EMRF algorithm applied to (almost-) random-shape networks with $n = 256$ and $m = 65536$.
The failures are distributed relatively to the number of attacks $k \in [0..\lambda - 1]$ where λ is the source-sink edge-connectivity –evaluated in figure 5.1. The plot is a density function estimated through the data. The distributions of those failures are almost independent of the distributions of the links capacities in those experiments.

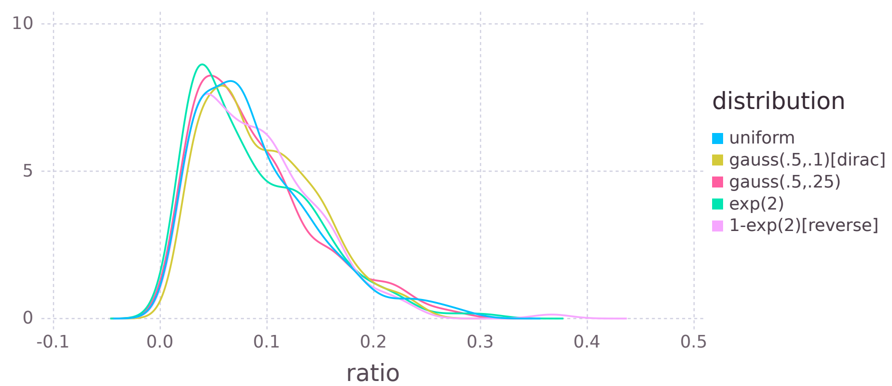


Figure 5.20: Distribution of the ratio between number of failures of the EMRF algorithm for (almost-) random-shape networks and their connectivities with $n = 256$ and $m = 65536$.
The plot is a density function estimated through the data. The distributions of those ratios are almost independent of the distributions of the links capacities in those experiments.

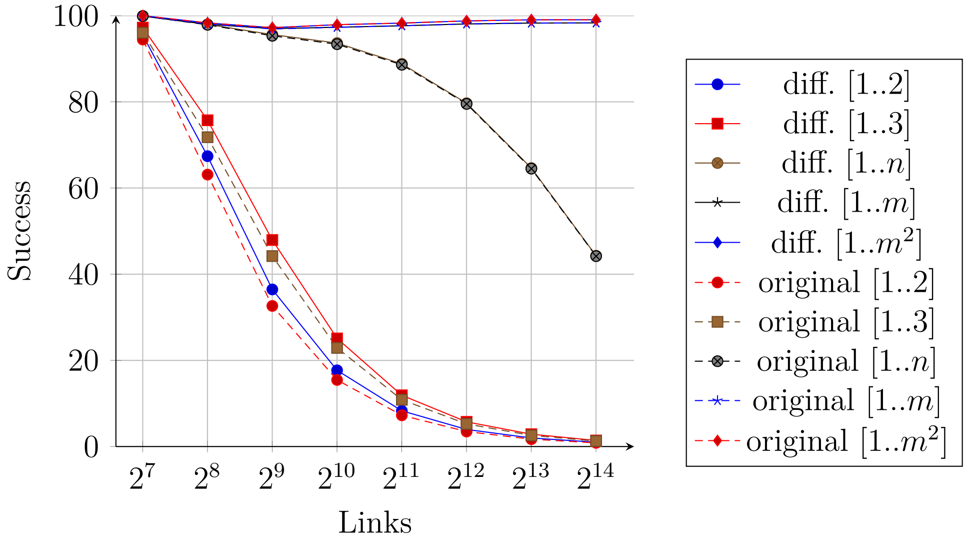


Figure 5.21: Comparison of the success rate of EMRF algorithm with or without capacity differentiation applied to instances when $n = 2^7 = 128$.

5.5.5 Capacity differentiation and mapping to integer

For practical reasons, as simplicity, storage of values, or necessity, the capacities of real networks are sometimes represented by integers. In that end, the capacity-differentiation of the EMRF algorithm might help to solve more instances and networks. A first evaluation of this extended method is given for R-MAT generated network in section 5.5.3, that is relatively efficient for small networks. The results of this experiments tends to be similar for random-shape instances networks as seen in figures 5.21 and 5.22. For smaller networks, the efficiency of the capacity-differentiation method is more visible.

The second question targeted by those experiments on networks with integer capacities is the efficiency of mapping real number into integer intervals. The map chosen are $[1..2], [1..3], [1..n], [1..m], [1..m^2]$. Figure 5.23 shows that both mapping on $[1..m]$ and $[1..m^2]$ for solving instances is almost as efficient as using float values. For the map $[1..m]$, the number of instances not solved is about 1% lower than the original. However, for mapping networks, it is as efficient as with using floats only for the map $[1..m^2]$ as it can be see in figure 5.24. This difference is due to the huge impact of 1% less of instances solved by EMRF on the map $[1..m]$. This impact is higher when the connectivity of the network increases.

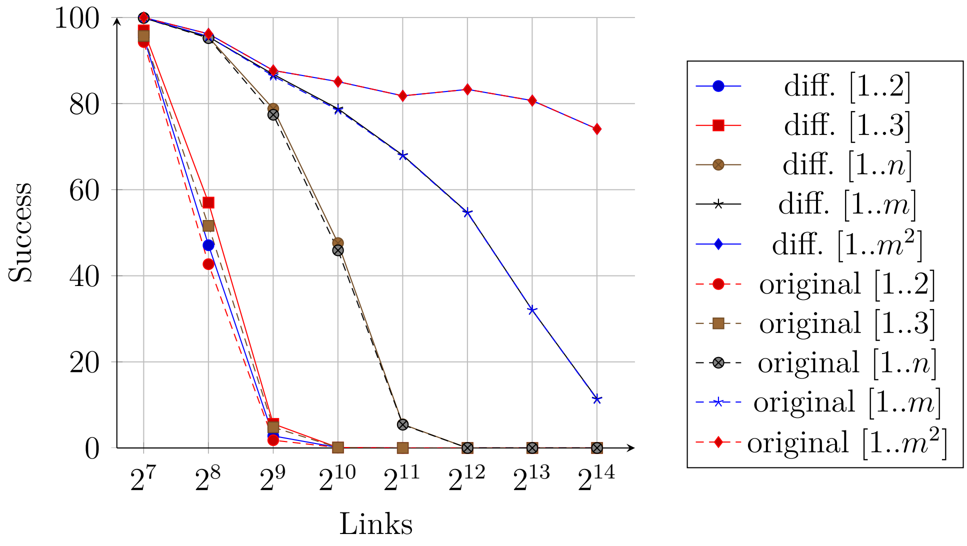


Figure 5.22: Comparison of the success rate of EMRF algorithm with or without capacity differentiation applied to networks when $n = 2^7 = 128$.

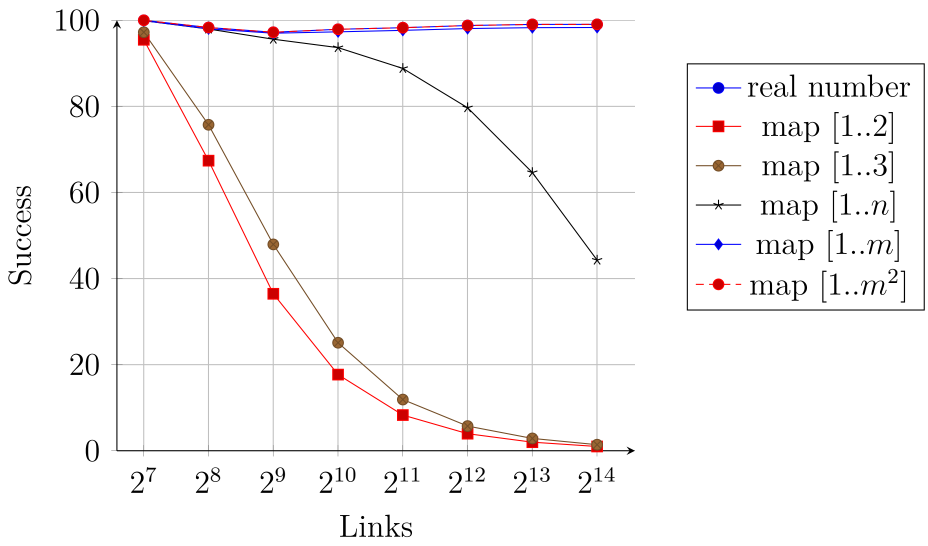


Figure 5.23: Comparison of the efficiency of mapping to the success rate of EMRF algorithm with or without capacity differentiation applied to instances when $n = 2^7 = 128$.

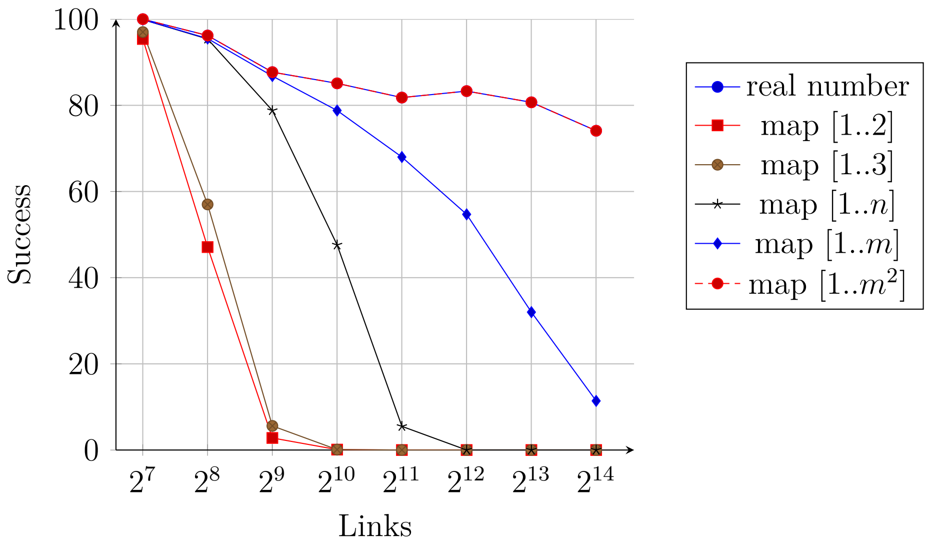


Figure 5.24: Comparison of the efficiency of mapping to the success rate of EMRF algorithm with or without capacity differentiation applied to networks when $n = 2^7 = 128$.

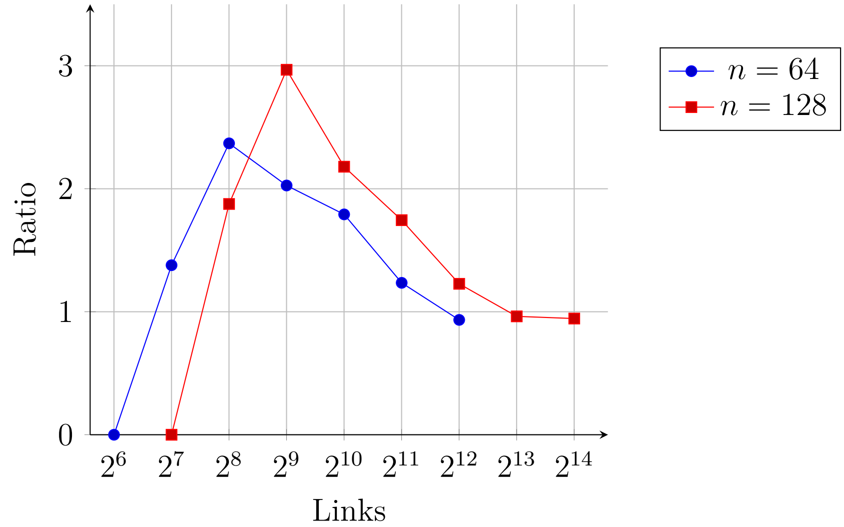


Figure 5.25: Comparison of upper and lower bounds of the mixed-EMRF algorithm on almost-random shape instances when $n = 2^7 = 128$.

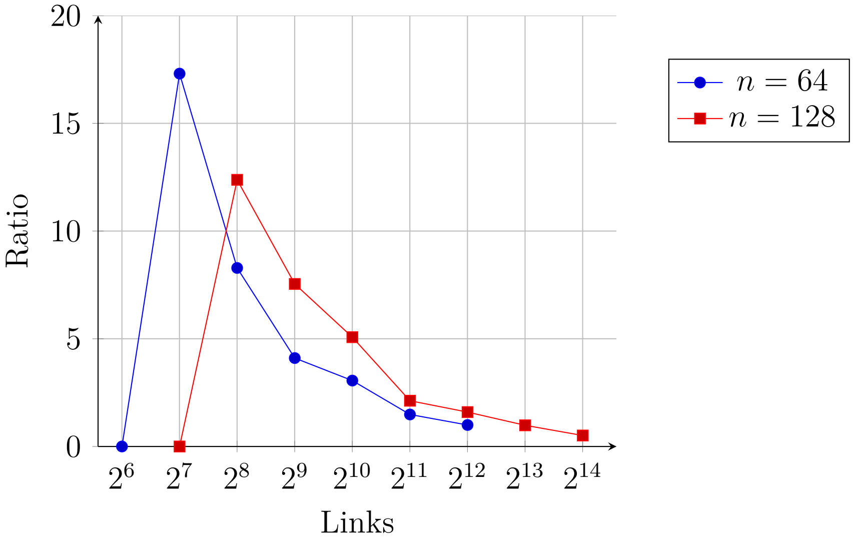


Figure 5.26: Comparison of upper and lower bounds of the mixed-EMRF algorithm on almost-random shape instances with a flying cut when $n = 2^7 = 128$.

5.6 Bounds and iterative multiroute flow comparison

The mixed-MLA algorithm introduced in section 5.2 uses a combination of EMRF and upper and lower bounds methods. This section covers experiments to evaluate those bounds values on random-shape networks and R-MAT generated networks. In figure 5.25, the gap between upper and lower bounds in mixed-MLA algorithm is always less than 3% on random-shape instances over all all the instances. This value increase on instances with a flying cutset –figure 5.26– up to 17% when $n = 64$ and up to 12% when

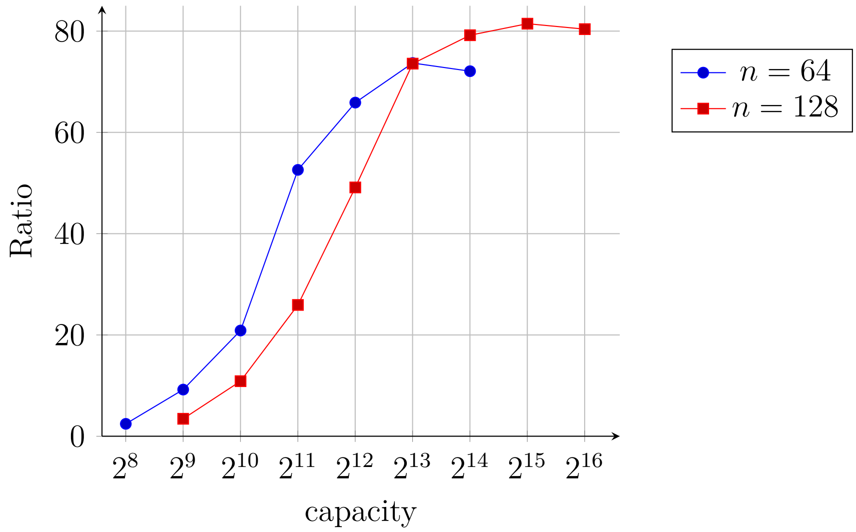


Figure 5.27: Comparison of upper and lower bounds of the mixed-EMRF algorithm on R-MAT instances.

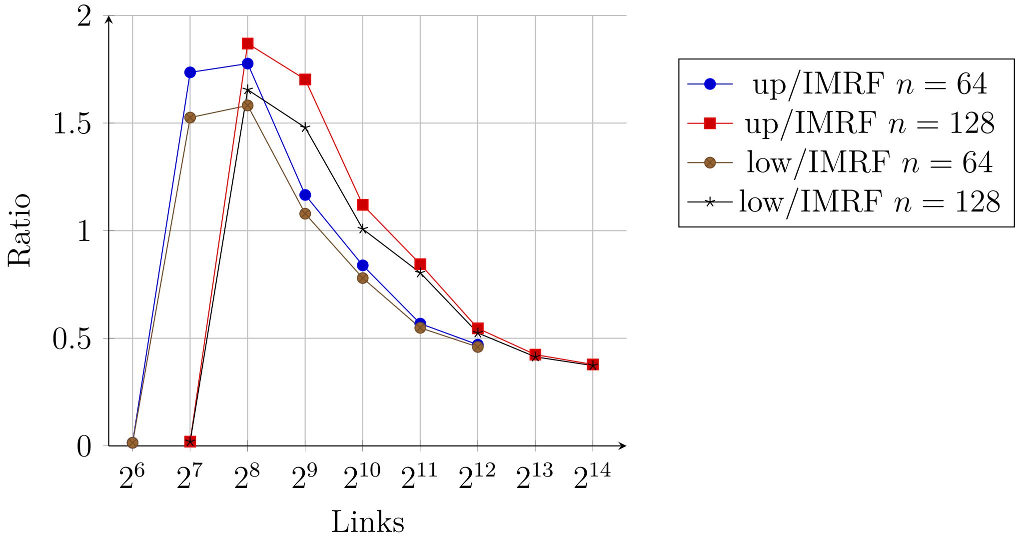


Figure 5.28: Comparison of upper and lower bounds with IMRF on almost-random shape instances.

$n = 128$. Unfortunately, the gap increase up to 80% on R-MAT generated networks as shown in figure 5.27.

The IMRF algorithm is a fast algorithm that might be a good approximation algorithm, as suggested by the experiments in figure 5.27. In figure 5.28, the average gap between IMRF and the upper bound of mixed-MLA algorithm is less than 2%.

5.7 Conclusion

Various experiments have been realized to size the efficiency of the different robust flow methods. The major experimental interpretation coming from those results is that the distribution of the capacities alone is not sufficient to impact strongly the success rate of EMRF algorithm. However, networks generated following a specific complex network model have a higher chance to possess flying cuts. The reason of this lower rate of success is currently unknown but motivate deeper analyses of the structures of the instances. Spectral graph theory is a candidate tool.

The bounds based on the analyses of the MLA-network flow problems in previous chapter gives reasonable results, even on the R-MAT complex network model. Designing a smart way to search for a better upper bound could reduce this gap, and be powerful tool to size the problem on large networks where the max-flow subroutine cannot be deterministic.

Chapter 6

Conclusion

In this thesis, an extended version of the multiroute flow is used to approximate, analyze and partially solve a new family of network flow problems: MLA-robust flow, MLA-reliable flow, and MLA-decomposition. Those three problems naturally widen the classical maximum flow notion, its dual counterpart the minimum cut and the decomposition of a flow into the context of a network under attacks. Yet their simple formulation covers a wide range of problems. They only differ in the amount of information detained by the attacker and the actions available to the defender. In the MLA-robust flow situation, the attacker strikes first while possessing all the network structure information, that are the nodes, the links, and the capacities. The defender is in a reactive position and try reroute a max-flow from the source to the sink after the attacks. On the contrary, in the MLA-reliable flow context, the defender makes a preemptive move by computing a flow that is maximum against any set of attacks. The information of the attacker is widened to the flow value on the links. This situation is typical of a network infrastructure preemptive measure before the application of new maximum-flow represented by the MLA-robust flow. Finally, in the MLA-decomposition case, the attacker even knows the flow decomposition and the defender try to maximize the number of active paths in the decomposition of the flow after any set of attacks.

This family of problems is at least partially intractable, for instance MLA-robust flow is NP-hard. The work of Wenkai Dai and Vorapong Suppakitpaisarn deepens the analyses of the complexity by proving the NP-hardness of MLA-decomposition-attack and MLA-effectiveness.

A first polynomial-time approximation algorithm is found in the notion of multiroute flow. When k is the number of attacks, a max-$(k+1)$-route flow is a $(k+1)$-approximation algorithm to MLA-robust flow, MLA-reliable flow, and MLA-decomposition. The gap can be tighter, for instance when the number of routes is near to the source-sink edge-connectivity of the network. The lower bound is later improved by using the pseudo-tangent method and the upper bound by using the duality between MLA-robust flow and MLA-robust cut.

The study of the multiroute flow, extended to a parametric non-integer number of routes version called EMRF, provides an exact solution to both MLA-robust flow and MLA-robust cut in polynomial for a certain category of instances called EMRF-solvable. This category is enlarged to include the instances solved by either the capacity-differentiation or the pseudo-tangent methods. At the opposite, the category of instances not EMRF-solvable is characterized by the presence of a specific kind of cut called flying-cut. Finally, another category of networks is introduced that consider the networks on which there is a flow, named best-MLA, that is the solution for both MLA-robust flow and MLA-reliable flow for each possible number of attacks. The iterative multiroute flow (IMRF) is another fast approximation algorithm for MLA-network flows, based on the iteration of successive multiroute flow with a decreasing number of routes and on the notion of completed-multiroute flow.

The EMRF algorithm combined with the lower and upper bounds methods –pseudo-tangent and MLA-robust flow/MLA-robust cut duality– makes the mixed-MLA algorithm. This algorithm computes successfully the MLA-robust flow and MLA-reliable flow values if the instance is EMRF-solvable. Otherwise it outputs a pair of lower and upper bounds. It is evaluated on various instances: random-shape networks with several capacity distribution, 2D and 3D grids, complete networks and complex networks generated by the R-MAT method. The success rate does not seem to be impacted by the distribution of the capacities alone. On networks generated by the R-MAT the success was much lower than for the other kind of instances while keeping a reasonable gap between the bounds.

In the future, several axes might pursued to solve the MLA-network flow problems. First comes the concern for the speed, since on nowadays large network, the speed of the best deterministic max-flow algorithm is not enough. Several solutions are possible, including: the design and use of fast max-flow approximation algorithms, the use of heuristic search on massively parallel architecture, or the improvement of the upper and lower bounds.

Despite its direct applications, both the EMRF and the MLA-network flow problems possess an optimization version of their number of routes or their number of attacks. Additionally, by their nature of network flows, both probably have various applications that are yet to be found.

The categorization of the instances into a class of networks that can be solved by a best-MLA flow might help into the synthesis of modern networks. The IMRF method is a strong candidate to a fast and efficient method to synthesis such networks.

References

[1] C. C. Aggarwal and J. B. Orlin. On multiroute maximum flows in networks. *Networks*, 39(1):43–52, 2002.

[2] R. Albert, H. Jeong, and A.-L. Barabási. Error and attack tolerance of complex networks. *Nature*, 406(6794):378–382, 2000.

[3] Y. Aneja, R. Chandrasekaran, and K. Nair. Parametric min-cuts analysis in a network. *Discrete Applied Mathematics*, 127(3):679–689, 2003.

[4] Y. Aneja, R. Chandrasekaran, S. Kabadi, and K. Nair. Flows over edge-disjoint mixed multipaths and applications. *Discrete Applied Mathematics*, 155(15):1979–2000, 2007.

[5] Y. P. Aneja and K. P. K. Nair. Maximal expected flow in a network subject to arc failures. *Networks*, 10(1):45–57, 1980.

[6] K. Appel and W. Haken. Every planar map is four colorable. part i: Discharging. *Illinois Journal of Mathematics*, 21(3):429–490, 1977.

[7] K. Appel, W. Haken, and J. Koch. Every planar map is four colorable. part ii: Reducibility. *Illinois Journal of Mathematics*, 21(3):491–567, 1977.

[8] J.-F. Baffier. Parametrized multiroute flow. Master's thesis, Université Paris-Sud, September 2011.

[9] J.-F. Baffier and V. Suppakitpaisarn. A $(k+1)$-approximation robust network flow algorithm and a tighter heuristic method using iterative multiroute flow. In S. Pal and K. Sadakane, editors, *Algorithms and Computation*, volume 8344 of *Lecture Notes in Computer Science*, pages 68–79. Springer International Publishing, 2014.

[10] J.-F. Baffier, V. Suppakitpaisarn, H. Hiraishi, and H. Imai. Parametric multiroute flow and its application to robust network with k edge failures. In P. Fouilhoux,

L. E. N. Gouveia, A. R. Mahjoub, and V. T. Paschos, editors, *Combinatorial Optimization*, Lecture Notes in Computer Science, pages 26–37. Springer International Publishing, 2014.

[11] J.-F. Baffier, W. Dai, and V. Suppakitpaisarn. Temporary title: Hardness and approximability results on multilink-attack flow problems. Submitted to COCOON 2015, 2015.

[12] J.-F. Baffier, W. Dai, and V. Suppakitpaisarn. Robust network flow against attackers with knowledge of flow decomposition. Submitted to HPSR 2015, 2015.

[13] J.-F. Baffier, V. Suppakitpaisarn, H. Hiraishi, and H. Imai. Parametric multiroute flow and its application to robust network with k edge failures. Submitted to special issue of Discrete Optimization on Combinatorial Optimization associated with ISCO2014, 2015.

[14] A. Bagchi, A. Chaudhary, C. Scheideler, and P. Kolman. Algorithms for fault-tolerant routing in circuit switched networks. In *Proceedings of the Fourteenth Annual ACM Symposium on Parallel Algorithms and Architectures*, SPAA '02, pages 265–274, New York, NY, USA, 2002. ACM.

[15] A. Bagchi, A. Chaudhary, M. Goodrich, and S. Xu. Constructing disjoint paths for secure communication. volume 2848 of *Lecture Notes in Computer Science*, pages 181–195. Springer Berlin Heidelberg, 2003.

[16] A. Bagchi, A. Chaudhary, P. Kolman, and J. Sgall. A simple combinatorial proof of duality of multiroute flows and cuts. Technical report, 2004.

[17] A. Bagchi, A. Chaudhary, and P. Kolman. Short length Menger's theorem and reliable optical routing. *Theoretical Computer Science*, 339(2 – 3):315–332, 2005.

[18] D. Barth, P. Berthomé, M. Diallo, and A. Ferreira. Revisiting parametric multi-terminal problems: Maximum flows, minimum cuts and cut-tree computations. *Discrete Optimization*, 3(3):195–205, 2006. Graphs and Combinatorial Optimization The Cologne/Twente Workshop on Graphs and Combinatorial Optimization.

[19] Y. Bejerano and P. Koppol. Resource reservation comparison of fault resilient routing schemes. In *Proc. LCN 2014*, 2014. (in press).

[20] P. Berthomé, M. Diallo, and A. Ferreira. Generalized parametric multi-terminal flows problem. In H. Bodlaender, editor, *Graph-Theoretic Concepts in Computer Science*, volume 2880 of *Lecture Notes in Computer Science*, pages 71–80. Springer Berlin Heidelberg, 2003.

[21] C. Bettstetter and C. Hartmann. Connectivity of wireless multihop networks in a shadow fading environment. *Wireless Networks*, 11(5):571–579, September 2005.

[22] G. Birkhoff. Three observations on linear algebra. *Revi. Univ. Nac. Tucuman, Ser A*, 5:147–151, 1946.

[23] A. Bley, M. Grötschel, and R. Wessäly. Design of broadband virtual private networks: Model and heuristics for the b-win. In *Robust communication networks: Interconnection and survivability, volume 53 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 1–16. American Mathematical Society, 1998.

[24] F. T. Boesch, F. Harary, and J. A. Kabell. Graphs as models of communication network vulnerability: Connectivity and persistence. *Networks*, 11(1):57–63, 1981.

[25] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, September 2004.

[26] A. Broder, D. Dolev, M. Fischer, and B. Simons. Efficient fault-tolerant routings in networks. *Information and Computation*, 75(1):52–64, 1987.

[27] Y. Caniou, P. Codognet, F. Richoux, D. Diaz, and S. Abreu. Large-scale parallelism for constraint-based local search: the costas array case study. *Constraints*, pages 1–27, 2014.

[28] D. Chakrabarti, C. Faloutsos, and Y. Zhan. Visualization of large networks with min-cut plots, a-plots and r-mat. *International Journal of Human-Computer Studies*, 65(5):434–445, 2007. Ambient intelligence: From interaction to insight.

[29] R. Chandrasekaran, K. Nair, Y. Aneja, and S. Kabadi. Multi-terminal multipath flows: synthesis. *Discrete Applied Mathematics*, 143($1-3$):182–193, 2004.

[30] C. Chekuri and S. Khanna. Algorithms for 2-route cut problems. In L. Aceto, I. Damgård, L. Goldberg, M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz, editors, *Automata, Languages and Programming*, volume 5125 of *Lecture Notes in Computer Science*, pages 472–484. Springer Berlin Heidelberg, 2008.

[31] J. Chuzhoy, Y. Makarychev, A. Vijayaraghavan, and Y. Zhou. Approximation algorithms and hardness of the k-route cut problem. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 780–799. SIAM, 2012.

[32] A. Cobham. The intrinsic computational difficulty of functions. In *Proceedings of the 1964 Congress for Logic, Methodology, and the Philosophy of Science*, pages 24–30, 1965.

[33] M.-C. Costa, L. Létocart, and F. Roupin. Minimal multicut and maximal integer multiflow: A survey. *European Journal of Operational Research*, 162(1):55–69, 2005. Logistics: From Theory to Application.

[34] J. DeWitt, S. Patt, and H. Shi. Maximizing continuous barrier coverage in energy harvesting sensor networks. *The 2014 IEEE International Conference on Communications*, 2014. in press.

[35] M. Diallo, S. Gueye, and P. Berthomé. Impact of a varying capacity on the all pairs 2-route network flows. *Electronic Notes in Discrete Mathematics*, 35(0):59–64, 2009. LAGOS'09 – V Latin-American Algorithms, Graphs and Optimization Symposium.

[36] M. Diallo, S. Gueye, and P. Berthomé. Sensitivity analysis on the all pairs q-route flows in a network. *International Transactions in Operational Research*, 17(1): 103–117, 2010.

[37] E. Dinic. Metod porazryadnogo sokrashcheniya nevyazok i transportnye zadachi. *Issledovaniya po Diskretnoi Matematike*, pages 46–57, 1973.

[38] P. A. M. Dirac. *The Principles of Quantum Mechanics*. International Series of Monographs on Physics. Clarendon, fourth edition, 1958.

[39] D. Du and R. Chandrasekaran. The multiroute maximum flow problem revisited. *Networks*, 47(2):81–92, 2006.

[40] D. Du and S. N. Kabadi. An improved algorithm for decomposing arc flows into multipath flows. *Operations Research Letters*, 34(1):53–57, 2006.

[41] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of mathematics*, 17(3): 449–467, 1965.

[42] J. Edmonds and R. M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264, April 1972.

[43] M. J. Eisner and D. G. Severance. Mathematical techniques for efficient record segmentation in large shared databases. *J. ACM*, 23(4):619–635, October 1976.

[44] S. Elloumi, E. Gourdin, and T. Lefebvre. Does network coding improve the throughput of a survivable multicast network? In *10th International Conference on the Design of Reliable Communication Networks (DRCN) 2014*, pages 1–8, April 2014.

[45] G. Exoo. On a measure of communication network vulnerability. *Networks*, 12(4): 405–409, 1982.

[46] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. volume 29, pages 251–262, New York, NY, USA, August 1999. ACM.

[47] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian journal of Mathematics*, 8(3):399–404, 1956.

[48] E. Fusy. Uniform random sampling of planar graphs in linear time. *Random Structures & Algorithms*, 35(4):464–522, 2009.

[49] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* WH Freeman & Co., 1979.

[50] N. Garg and J. Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. *SIAM Journal on Computing*, 37(2):630–652, 2007.

[51] A. V. Goldberg and R. E. Tarjan. Finding minimum-cost circulations by canceling negative cycles. *J. ACM*, 36(4):873–886, October 1989.

[52] A. V. Goldberg and R. E. Tarjan. Efficient maximum flow algorithms. *Commun. ACM*, 57(8):82–89, August 2014.

[53] R. Gomory and T. Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):551–570, 1961.

[54] F. Guerriero and M. Mancini. Experimental evaluation of solution approaches for the k-route maximum flow problem. *Computers & Operations Research*, 32(9): 2453–2478, 2005.

[55] D. M. Gusfield. *Sensitivity Analysis for Combinatorial Optimization.* PhD thesis, 1980.

[56] D. M. Gusfield. Parametric combinatorial computing and a problem of program module distribution. *J. ACM*, 30(3):551–563, July 1983.

[57] F. Halim, R. Yap, and Y. Wu. A mapreduce-based maximum-flow algorithm for large small-world network graphs. In *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, pages 192–202, June 2011.

[58] T. Harris and F. Ross. Fundamentals of a method for evaluating rail net capacities. Technical report, DTIC Document, 1955.

[59] T. Hartman, A. Hassidim, H. Kaplan, D. Raz, and M. Segalov. How to split a flow? In *INFOCOM, 2012 Proceedings IEEE*, pages 828–836. IEEE, March 2012.

[60] A. Hulgeri and S. Sudarshan. Parametric query optimization for linear and piecewise linear cost functions. In *Proceedings of the 28th International Conference on Very Large Data Bases*, VLDB '02, pages 167–178. VLDB Endowment, 2002.

[61] K. Kar, M. Kodialam, and T. Lakshman. Routing restorable bandwidth guaranteed connections using maximum 2-route flows. *Networking, IEEE/ACM Transactions on*, 11(5):772–781, October 2003.

[62] R. M. Karp and J. B. Orlin. Parametric shortest path algorithms with an application to cyclic staffing. *Discrete Applied Mathematics*, 3(1):37–45, 1981.

[63] K. Kawarabayashi and M. Thorup. Deterministic global minimum cut of a simple graph in near-linear time. *arXiv:1411.5123v3 [cs.DS]*, November 2014.

[64] W. Kishimoto. A method for obtaining the maximum multiroute flows in a network. *Networks*, 27(4):279–291, 1996.

[65] W. Kishimoto. Reliable flow with failures in a network. *Reliability, IEEE Transactions on*, 46(3):308–315, September 1997.

[66] W. Kishimoto. A method for obtaining the maximum (δ, η)-balanced flow in a network. In J. Pahl, T. Reiners, and S. Voß, editors, *Network Optimization*, volume 6701 of *Lecture Notes in Computer Science*, pages 230–242. Springer Berlin Heidelberg, 2011.

[67] W. Kishimoto. An extension of a balanced flow in a network. *Electronic Notes in Discrete Mathematics*, 41(0):109–116, 2013.

[68] W. Kishimoto and M. Takeuchi. m-route flows in a network. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 77(5):1–18, 1994.

[69] W. Kishimoto and M. Takeuchi. A method for obtaining the maximum δ-reliable flow in a network. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 81(5):776–783, May 1998.

[70] W. Kishimoto, M. Takeuchi, and G. Kishi. Two-route flows in an undirected flow network. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 76(6):38–59, 1993.

[71] Y. Kobayashi and K. Otsuki. Max-flow min-cut theorem and faster algorithms in a circular disk failure model.

[72] P. Kolman and C. Scheideler. Towards duality of multicommodity multiroute cuts and flows: Multilevel ball-growing. *Theory of Computing Systems*, 53(2):341–363, 2013.

[73] V. Kolmogorov, Y. Boykov, and C. Rother. Applications of parametric maxflow in computer vision. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, October 2007.

[74] S. Kumar, T. Lai, M. Posner, and P. Sinha. Maximizing the lifetime of a barrier of wireless sensors. *Mobile Computing, IEEE Transactions on*, 9(8):1161–1172, August 2010.

[75] P. Lee, V. Misra, and D. Rubenstein. Distributed algorithms for secure multipath routing. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 1952–1963 vol. 3, March 2005.

[76] Y.-K. Lin. A simple algorithm for reliability evaluation of a stochastic-flow network with node failure. *Computers & Operations Research*, 28(13):1277–1285, 2001.

[77] Y.-K. Lin. Using minimal cuts to evaluate the system reliability of a stochastic-flow network with failures at nodes and arcs. *Reliability Engineering & System Safety*, 75(1):41–46, 2002.

[78] M. Minoux. On robust maximum flow with polyhedral uncertainty sets. *Optimization Letters*, 3(3):367–376, 2009.

[79] T. Moisan, J. Gaudreault, and C.-G. Quimper. Parallel discrepancy-based search. In C. Schulte, editor, *Principles and Practice of Constraint Programming*, volume 8124 of *Lecture Notes in Computer Science*, pages 30–46. Springer Berlin Heidelberg, 2013.

[80] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978.

[81] J. B. Orlin. Max flows in $o(nm)$ time, or better. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 765–774, New York, NY, USA, 2013. ACM.

[82] M. Patra and B. Jayasingh. A software agent based approach for fraud detection in network crimes. In S. Manandhar, J. Austin, U. Desai, Y. Oyanagi, and A. Talukder, editors, *Applied Computing*, volume 3285 of *Lecture Notes in Computer Science*, pages 310–316. Springer Berlin Heidelberg, 2004.

[83] V. Popeskic. Cdp attacks: Cisco discovery protocol attack, December 2011. URL http://howdoesinternetwork.com/2011/cdp-attack.

[84] N. Robertson, D. Sanders, P. Seymour, and R. Thomas. The four-colour theorem. *Journal of Combinatorial Theory, Series B*, 70(1):2–44, 1997.

[85] M. Schwartz. *Telecommunication Networks: Protocols, Modeling and Analysis*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1986.

[86] V. Suppakitpaisarn and J.-F. Baffier. Speeding up algorithm for maximizing barrier coverage using parametric multiroute flow. In *Proc. TJIA 2014*, pages 80–84, 2014.

[87] W.-J. van Hoeve and J.-C. Régin. Open constraints in a closed world. In J. Beck and B. Smith, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, volume 3990 of *Lecture Notes in Computer Science*, pages 244–257. Springer Berlin Heidelberg, 2006.

[88] W.-J. Van Hoeve, G. Pesant, and L.-M. Rousseau. On global warming: Flow-based soft global constraints. *Journal of Heuristics*, 12(4-5):347–373, 2006.

[89] B. Vatinlen, F. Chauvet, P. Chrétienne, and P. Mahey. Simple bounds and greedy algorithms for decomposing a flow into a minimal set of paths. *European Journal of Operational Research*, 185(3):1390–1401, 2008.

[90] F. Xie and A. Davenport. Massively parallel constraint programming for supercomputers: Challenges and initial results. In A. Lodi, M. Milano, and P. Toth, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, volume 6140 of *Lecture Notes in Computer Science*, pages 334–338. Springer Berlin Heidelberg, 2010.

[91] U. Zimmermann. On the complexity of the dual method for maximum balanced flows. *Discrete Applied Mathematics*, 50(1):77–88, 1994.

List of Figures

1.1 Example of MLA-network flows 2

1.2 Organization of thesis. 8

2.1 The Seven Bridge of Königsberg 11

2.2 Example max-flow 14

2.3 Example of a max-2-route flow 19

3.1 Example of MLA-decompostion 30

3.2 Proof of the NP-hardness of the MLA-decomposition-attack 35

3.3 Proof of the NP-hardness of the MLA-effectiveness 37

3.4 Network with distinct flow values 39

3.5 Networks used to prove propositions 3.3 to 3.5 41

3.6 Example of a route-parametric multiroute flow 49

3.7 Link-parametric max-h-route flow 51

4.1 Capacity-differentiation method. 57

4.2 Skipping breakpoint example 59

4.3 Bounds and flying-cutset 60

4.4 Lower bound of the ratio of completed flow 63

4.5 Class diagram 65

4.6 A counter-example to the existence of best MLA-flow 66

5.1 Average connectivity on almost-randomly generated networks. 68

5.2 The R-MAT model as described by Chakrabarti, Faloutsos, and Zhan [28]. 69

5.3 Evaluation of completed-MRF 72

5.4 Success rate of EMRF algorithm on complete networks.
The number of links in a complete network is about the square of the number of nodes. The latter one takes the values of the power of 2 between $2^2 = 4$ and $2^8 = 256$ and is represented by a logarithmic scaled axis. . . . 73

5.5 Distribution of failures of the EMRF algorithm applied to complete networks.
The failures are distributed relatively to the number of attacks $k \in [0..\lambda-1]$ where $\lambda = n-1$ is the source-sink edge-connectivity. The plot is a density function estimated through the data. 74

5.6 Distribution of the ratio between number of failures of the EMRF algorithm for complete networks and their connectivities.
The plot is a density function estimated through the data. 74

5.7 Success rate of EMRF algorithm on 2D grids.
The number of nodes by dimension takes the values the values of the power of 2 between $2^2 = 4$ and $2^6 = 64$ and is represented by a logarithmic scaled axis. The connectivity of a 2D grid is always 4. 75

5.8 Distribution of failures of the EMRF algorithm applied to 2D grids.
The failures are distributed relatively to the number of attacks $k \in [0..\lambda - 1]$ where $\lambda = 4$ is the source-sink edge-connectivity. The plot is a density function estimated through the data. Because of the relatively small value of λ, the density can be considered in a discrete way: when $\frac{k}{\lambda} = \{0.25, 0.5, 0.75\}$. . . . 76

5.9 Distribution of the ratio between number of failures of the EMRF algorithm for 2D grids and their connectivities.
The plot is a density function estimated through the data. Because of the relatively small value of λ, the density can be considered in a discrete way: when $\frac{k}{\lambda} = \{0.25, 0.5, 0.75\}$. In those experiments the EMRF algorithm only fails for one value of k for the 2D grids with a flying cut. 76

5.10 Success rate of EMRF algorithm on 3D grids.
The number of nodes by dimension takes the values the values of the power of 2 between $2^2 = 4$ and $2^4 = 16$ and is represented by a logarithmic scaled axis. The connectivity of a 3D grid is always 6. 77

5.11 Distribution of failures of the EMRF algorithm applied to 3D grids.
The failures are distributed relatively to the number of attacks $k \in [0..\lambda - 1]$ where $\lambda = 6$ is the source-sink edge-connectivity. The plot is a density function estimated through the data. Because of the relatively small value of λ, the density can be considered in a discrete way: when $\frac{k}{\lambda} = \{\frac{1}{6}, \frac{2}{6}, \cdots, \frac{5}{6}\}$. 77

5.12 Distribution of the ratio between number of failures of the EMRF algorithm for 3D grids and their connectivities.
The plot is a density function estimated through the data. Because of the relatively small value of λ, the density can be considered in a discrete way: when $\frac{k}{\lambda} = \{\frac{1}{6}, \frac{2}{6}, \cdots, \frac{5}{6}\}$. 78

5.13 Success rate of EMRF algorithm on R-MAT generated networks 79

5.14 Success rate of EMRF algorithm with capacity differentiation on R-MAT generated networks 80

5.15 Distribution of failures of the EMRF algorithm applied to R-MAT generated networks.
The failures are distributed relatively to the number of attacks $k \in [0..\lambda-1]$ where $\lambda = n - 1$ is the source-sink edge-connectivity. The plot is a density function estimated through the data. 80

5.16 Success of EMRF algorithm on almost-random shape networks with 128 nodes. 81

5.17 Success of EMRF algorithm on almost-random shape networks with 256 nodes. 82

5.18 Success of EMRF algorithm on almost-random shape networks with uniform capacity distribution in function of the number of nodes. 82

5.19 Distribution of failures of the EMRF algorithm applied to (almost-) random-shape networks with $n = 256$ and $m = 65536$.
The failures are distributed relatively to the number of attacks $k \in [0..\lambda-1]$ where λ is the source-sink edge-connectivity –evaluated in figure 5.1. The plot is a density function estimated through the data. The distributions of those failures are almost independent of the distributions of the links capacities in those experiments. 83

5.20 Distribution of the ratio between number of failures of the EMRF algorithm for (almost-) random-shape networks and their connectivities with $n = 256$ and $m = 65536$.
The plot is a density function estimated through the data. The distributions of those ratios are almost independent of the distributions of the links capacities in those experiments. 83

5.21 Comparison of the success rate of EMRF algorithm with or without capacity differentiation applied to instances when $n = 2^7 = 128$. 84

5.22 Comparison of the success rate of EMRF algorithm with or without capacity differentiation applied to networks when $n = 2^7 = 128$. 85

5.23 Comparison of the efficiency of mapping to the success rate of EMRF algorithm with or without capacity differentiation applied to instances when $n = 2^7 = 128$. 85

5.24 Comparison of the efficiency of mapping to the success rate of EMRF algorithm with or without capacity differentiation applied to networks when $n = 2^7 = 128$. 86

5.25 Comparison of upper and lower bounds of the mixed-EMRF algorithm on almost-random shape instances when $n = 2^7 = 128$. 87

5.26 Comparison of upper and lower bounds of the mixed-EMRF algorithm on almost-random shape instances with a flying cut when $n = 2^7 = 128$. . 87

5.27 Comparison of upper and lower bounds of the mixed-EMRF algorithm on R-MAT instances. 88

5.28 Comparison between bounds and IMRF on random shape instances. . . . 88

List of Tables

2.1 Complexity of max-flow 15

List of Linear Programs

2.1 Maximum flow 15
2.2 Maximum flow decomposition 17
2.3 Maximum h-route flow 21
3.1 MLA-reliable 29
3.2 MLA-decomposition 30