

Title	ノードの高次元的な相関関係に基づいた頑健なネットワーク構造を構築する手法
Author(s)	KIM, JAEHO
Citation	
Issue Date	2024-03
Type	Thesis or Dissertation
Text version	ETD
URL	http://hdl.handle.net/10119/19077
Rights	
Description	Supervisor: 林 幸雄, 先端科学技術研究科, 博士

博士論文

ノードの高次元的な相関関係に基づいた 頑健なネットワーク構造を構築する手法

KIM JAEHO

主任研究指導教員 林 幸雄

北陸先端科学技術大学院大学 金沢大学
融合科学共同専攻

[融合科学]

令和6年3月

Abstract

Various infrastructure networks in modern society form the basis of many social and economic activities and are continually increasing in complexity and importance. Network science plays an important role in simplifying them as networks consisting of nodes and links and in analyzing information, resources, or their flow within networks. In fact, network science has revealed that real networks commonly have a scale-free (SF) structure. The most important feature of SF networks is that their degree distribution follow a power-law. This indicates that a few nodes (called hubs) have a lots of links, and most nodes have relatively few links. Since almost all nodes are connected to hubs, the network has a high efficiency in disseminating information because the hub can shorten the paths between nodes in the network. Although SF networks are robust against random failures, they are extremely vulnerable to malicious attacks targeting hubs. Therefore, hubs are advantageous for information dissemination, but they are also structural weakness. Moreover, many real networks are frequently threatened by natural and man-made disasters. Since infrastructure networks support our society, it is necessary to overcome their inherent weaknesses and construct robust structures.

Conventional studies for network robustness have mainly focused on degree correlations. For example, it is shown that onion-like structures with positive degree correlation have optimal robustness to malicious attacks. However, with too high degree correlation, networks have low robustness. Therefore, it is necessary to find new principles to construct a robust structure. Recently, it is shown that loop structures are more important than degree correlations to construct robust networks. The reason is that a network without loops become a weak tree structure. Thus, in order to construct robust structure, it is important to enhance loops so that prevent the network from becoming a tree structure. Several rewiring methods based on enhancing loops have been proposed. These methods construct more robust networks than the method increasing the degree correlation. Furthermore, it is commonly confirmed that the networks rewired by such methods have small width of degree distributions. Having small width of degree distributions corresponds to decreasing the gap between the maximum and the minimum degrees. Thus, as an extreme example, a random regular graph with the zero gap has optimal robustness. Such optimal robustness of a regular graph is verified by perturbation analysis.

Based on the above backgrounds, this paper has the following three objectives. First, we propose a self-healing method based on enhancing loops and reducing the width of degree distribution. Then, we clarify the effect of community structure on network robustness. Finally, we propose a method to make configurations of data-centers that has robust and even load distribution.

keywords: Robustness of connectivity, Enhancing loops, The minimum gap of degree distribution, Distributed self-healing, Community structure, Data-center configuration.

目次

第 1 章	緒言	14
1.1	研究の背景と目的	14
1.2	論文の構成	18
第 2 章	本研究に対する先行研究	20
2.1	ネットワークの特性	20
2.1.1	基本定義	20
2.1.2	次数相関	20
2.1.3	頑健性	21
2.1.4	効率性	22
2.2	スケールフリー (SF) ネットワーク	22
2.3	中央集権システムから分散システムへの転換	23
2.4	頑健な構造の先行研究	24
2.4.1	次数相関と頑健性	24
2.4.2	ループと頑健性	25
2.4.3	コミュニティと頑健性	27
2.5	ソーシャル・キャピタル	28
2.6	情報拡散の拠点	29
2.6.1	中心性	30
2.6.2	支配集合問題	31
2.6.3	頂点被覆問題	32
第 3 章	頑健なネットワーク構造へ再構築する自己修復法と、その分散アルゴリズム	33
3.1	構造再構築の必要性	33

3.2	従来の自己修復法の改良	34
3.3	改良修復法の提案	36
3.3.1	改良修復法の概要	36
3.3.2	分散処理を考慮した改良	37
3.4	先の修復法と改良修復法の修復結果比較	39
3.4.1	分析指標とサンプルネットワーク	39
3.4.2	改良修復法による頑健性、効率性の向上	40
3.4.3	改良修復法による構造改善	45
3.4.4	資源制御の比較	49
3.5	改良修復法の分散アルゴリズム記述	50
3.5.1	非同期処理ための前提	50
3.5.2	分散アルゴリズムの各フェーズ説明	51
3.5.3	まとめ	67
第 4 章	反モジュール化における頑健性と効率性の共存	69
4.1	コミュニティ構造と頑健性の関係	69
4.2	反モジュール化による性モジュール性制御	70
4.2.1	コミュニティ構造を持つレギュラグラフの設定	70
4.2.2	反モジュール化と従来のモジュール化	71
4.3	反モジュール化の効果確認	73
4.3.1	反モジュール化による頑健性の向上	73
4.3.2	反モジュール化による効率性の向上	86
4.3.3	従来のモジュール化の修正	89
4.4	まとめ	93
第 5 章	頑健性と負荷分散を考慮したデータセンター配置構成	94
5.1	データセンターと利用者両方に安全な配置	94
5.2	配置構成を求める近似法	96
5.2.1	貪欲ヒューリスティック法 (Greedy Heuristic、GH)	96
5.2.2	マルチホップカバー方法 (Multi-Hop、MH)	96
5.3	求められた配置構成の評価	97
5.3.1	評価のための初期設定と評価指標	97
5.3.2	攻撃以前の結果比較	99

5.3.3	攻撃後の結果比較	103
5.4	まとめ	107
第 6 章	結言	108
6.1	研究成果の要約	108
6.2	今後の課題	109
参考文献		112

図目次

3.1	通信範囲として3ホップが必要な理由の説明。(A) 攻撃直後のネットワーク。修復リンクは、被害を受けたノード間に追加される。(B) 2ホップだけ通信できる場合、(C) 3ホップまで通信できる場合。赤色ノードは故障ノード、青色ノードは故障ノードへのリンク(赤色破線)が切れたせいで被害を受けたノードである。緑色線は修復リンクを意味ある。	38
3.2	攻撃を受けたネットワーク上でのリング形成と、ループ強化の概要図。赤色ノードは故障ノード、青色ノードは故障ノードへのリンク(赤色破線)が切れたせいで被害を受けたノードである。緑色と黄色線は、それぞれリング形成とループ強化ために生成される修復リンクである。	38
3.3	OpenFlight、AS Oregon、PowerGrid 上での改良修復法(上向き三角)と既存法(下向き三角)の比較。各 r_h において、修復したネットワークの (A) 最大連結成分サイズ比率 $S(q)/(1 - q)N$, (B) 頑健性指標 $R(q)$ 、(C) 経路の効率 $E(q)$ 。赤、緑、青、黄、紫色線は、それぞれ $r_h = 0.05, 0.1, 0.2, 0.5, 1.0$ に対応する結果である。黒破線は攻撃以前のネットワークが持つ指標である。	41

3.4	再構築された OpenFlight が持つ頑健性。(A) HDA で攻撃されたネットワークを再構築し、それをまた HDA で攻撃した時の最大連結成分サイズ比率 $S(q)/(1-q)N$ 、(B) HDA で攻撃されたネットワークを再構築し、それを LA で攻撃した時の最大連結成分サイズ比率 $S(q)/(1-q)N$ 、(C) LA で攻撃されたネットワークを再構築し、それを HDA で攻撃した時の最大連結成分サイズ比率 $S(q)/(1-q)N$ 、(D) LA で攻撃されたネットワークを再構築し、それをまた LA で攻撃した時の最大連結成分サイズ比率 $S(q)/(1-q)N$ 。赤、緑、青、黄、紫色線は、それぞれ $r_h = 0.05, 0.1, 0.2, 0.5, 1.0$ に対応する結果である。黒破線は攻撃以前のネットワークが持つ指標である。	43
3.5	再構築された AS Oregon が持つ頑健性。(A) HDA で攻撃されたネットワークを再構築し、それをまた HDA で攻撃した時の最大連結成分サイズ比率 $S(q)/(1-q)N$ 、(B) HDA で攻撃されたネットワークを再構築し、それを LA で攻撃した時の最大連結成分サイズ比率 $S(q)/(1-q)N$ 、(C) LA で攻撃されたネットワークを再構築し、それを HDA で攻撃した時の最大連結成分サイズ比率 $S(q)/(1-q)N$ 、(D) LA で攻撃されたネットワークを再構築し、それをまた LA で攻撃した時の最大連結成分サイズ比率 $S(q)/(1-q)N$ 。赤、緑、青、黄、紫色線は、それぞれ $r_h = 0.05, 0.1, 0.2, 0.5, 1.0$ に対応する結果である。黒破線は攻撃以前のネットワークが持つ指標である。	44
3.6	攻撃以前のネットワークと、修復したネットワーク上で、ローカルマップサイズの比率。赤、緑、青、黄、紫色線は、それぞれ $r_h = 0.05, 0.1, 0.2, 0.5, 1.0$ に対応する結果である。	45
3.7	改良修復法で修復されたネットワークの次数分布。修復は qN 個のノードが壊された後に進められた：(A) $q = 0.1$ 、(B) $q = 0.5$ 、(C) $q = 0.9$ 。赤、緑色バーは、それぞれ $r_h = 0.5$ 、 $r_h = 1.0$ での次数分布である。黒垂直線は、攻撃以前のネットワークの平均次数である (表 1 参照)。 . . .	46
3.8	OpenFlight, AS Oregon、PowerGrid 上でのローカル資源制御 (プラス) とグローバル資源制御 (クロス) の結果比較。各 r_h において、修復したネットワークの (A) 最大連結成分サイズ比率 $S(q)/(1-q)N$ 、(B) 頑健性指標 $R(q)$ 、(C) 経路の効率 $E(q)$ 。赤、緑、青、黄、紫色線は、それぞれ $r_h = 0.05, 0.1, 0.2, 0.5, 1.0$ に対応する結果である。	50
3.9	提案した分散アルゴリズムにおける、同時処理の概念図。	52

3.10	5つのフェーズで構成される修復プロセスのフローチャート。	53
3.11	フェーズ2のプロセスの概要。円内の数字はノードIDを表す。(A) 根ノード(青色ノード)から葉ノード(緑色ノード)に <i>ModeChange</i> メッセージが伝播する。黒色矢印は <i>ModeChange</i> メッセージの方向を示す。赤色ノードと赤色線は故障ノードとそのリンクである。黒色線は配信木での親子関係を表す。黒色点線は、両ノードが親子関係を持たないことを意味する。(B) <i>Back</i> メッセージは葉ノードから送信される。”+ 数字”は <i>Back</i> メッセージ内である <i>TreeSize</i> を計算するための増分を示す。	59
3.12	進行中のフェーズ2の例。(A) 連結成分内にノード1、2、3の3つの根ノード(青色ノード)が存在する。この場合、ノード4と5は黒色矢印で示される複数の <i>ModeChange</i> メッセージを受け取れる。黒色線はノード間の直接的な連結を示す。(B) 結果として、連結成分内に3つの配信木(灰色三角形)が構築される。黒色点線は、2つのノードが直接繋がっているが、互いに異なる配信木に属することを表す。	60
3.13	フェーズ3のプロセス。このネットワークでは、ノード1が最も低いIDを持つリーダーノードで、ノード2、3、4、5が拡張された修復候補であることを知っている。ノード4と5の配信木が別の連結成分にあるため、これらの配信木のサイズはノード1、2、3の配信木とは別に蓄積する必要がある。したがって、フェーズ3では、ノード1は2つの連結成分、CC1とCC2を区別している。赤色ノードと赤色線は故障ノードとそのリンクを示す。青色ノードは被害を受けたノードである。灰色三角形は各根ノードの配信木を表す。灰色多角形は、連結成分(CC)である。	62
3.14	リング形成の概要図。(A) リーダーノード(ノード1)は拡張された修復候補にメッセージを送信する(黒色矢印)。(B) メッセージを受信すると、被害を受けたノード(青色ノード)は、各自新しいリンク(緑色矢印)を生成する。赤色ノードは故障ノードにである。	65
4.1	リワイヤリング前後のモジュラーネットワーク構成。(A) 高いモジュール性持つ初期構成。(B) リワイヤされたネットワークは頑健性が増加する。ノード色はモジュールを意味する。	71

- 4.2 MB 攻撃に対する最大連結成分のサイズ比率。ネットワークの次数 d は 9 で、モジュール数 m_o は、(A) $m_o = 1000$, (B) $m_o = 500$, (C) $m_o = 100$, (D) $m_o = 50$, (E) $m_o = 20$, and (F) $m_o = 5$ である。線の色は反モジュール化における、リワイヤするイントラリンク数の比率 w' に対応する。 75
- 4.3 MB 攻撃に対する最大連結成分のサイズ比率。ネットワークの次数 d は 4 で、モジュール数 m_o は、(A) $m_o = 2000$, (B) $m_o = 1000$, (C) $m_o = 500$, (D) $m_o = 100$, (E) $m_o = 20$, and (F) $m_o = 5$ である。線の色は反モジュール化における、リワイヤするイントラリンク数の比率 w' に対応する。 76
- 4.4 MB 攻撃に対する最大連結成分のサイズ比率。ネットワークの次数 d は 19 で、モジュール数 m_o は、(A) $m_o = 500$, (B) $m_o = 200$, (C) $m_o = 100$, (D) $m_o = 50$, (E) $m_o = 20$, and (F) $m_o = 5$ である。線の色は反モジュール化における、リワイヤするイントラリンク数の比率 w' に対応する。 77
- 4.5 3つの攻撃に対する最大連結成分のサイズ比率。 $d = 4$ で、(A) $m_o = 8$ 、MB 攻撃、(B) $m_o = 4$ 、MB 攻撃、(C) $m_o = 8$ 、IB 攻撃、(D) $m_o = 4$ 、IB 攻撃、(E) $m_o = 8$ 、RF and (F) $m_o = 4$ 、RF の場合、最大連結成分のサイズ比率である。線の色は反モジュール化における、リワイヤするイントラリンク数の比率 w' に対応する。 78
- 4.6 3つの攻撃に対する最大連結成分のサイズ比率。 $d = 9$ で、(A) $m_o = 8$ 、MB 攻撃、(B) $m_o = 4$ 、MB 攻撃、(C) $m_o = 8$ 、IB 攻撃、(D) $m_o = 4$ 、IB 攻撃、(E) $m_o = 8$ 、RF and (F) $m_o = 4$ 、RF の場合、最大連結成分のサイズ比率である。線の色は反モジュール化における、リワイヤするイントラリンク数の比率 w' に対応する。 79
- 4.7 3つの攻撃に対する最大連結成分のサイズ比率。 $d = 19$ で、(A) $m_o = 8$ 、MB 攻撃、(B) $m_o = 4$ 、MB 攻撃、(C) $m_o = 8$ 、IB 攻撃、(D) $m_o = 4$ 、IB 攻撃、(E) $m_o = 8$ 、RF and (F) $m_o = 4$ 、RF の場合、最大連結成分のサイズ比率である。線の色は反モジュール化における、リワイヤするイントラリンク数の比率 w' に対応する。 80

- 4.8 3種類の攻撃に対する頑健性指標 R とモジュール性 Q 間の関係。次数 d は 9 である。(A) MB 攻撃に対する頑健性 R_{MB} 、(B) IB 攻撃に対する頑健性 R_{IB} 、(C)RF に対する頑健性 R_{RF} 。線色はモジュール数 m_o に対応する。黒線、黒点線、シアン色線は、各自モジュール構造がない、ER ランダムグラフ、SF ネットワークモデル、ランダム 9-レギュラグラフをそれぞれ指す。 82
- 4.9 3種類の攻撃に対する頑健性指標 R とモジュール性 Q 間の関係。次数 d は 4 である。(A) MB 攻撃に対する頑健性 R_{MB} 、(B) IB 攻撃に対する頑健性 R_{IB} 、(C)RF に対する頑健性 R_{RF} 。線色はモジュール数 m_o に対応する。黒線、黒点線、シアン色線は、各自モジュール構造がない、ER ランダムグラフ、SF ネットワークモデル、ランダム 4-レギュラグラフをそれぞれ指す。 82
- 4.10 3種類の攻撃に対する頑健性指標 R とモジュール性 Q 間の関係。次数 d は 19 である。(A) MB 攻撃に対する頑健性 R_{MB} 、(B) IB 攻撃に対する頑健性 R_{IB} 、(C)RF に対する頑健性 R_{RF} 。線の色はモジュール数 m_o に対応する。黒線、黒点線、シアン色線は、各自モジュール構造がない、ER ランダムグラフ、SF ネットワークモデル、ランダム 19-レギュラグラフをそれぞれ指す。 83
- 4.11 平均最短経路長 L ネットワークサイズ N の関係。 $d = 4$ のネットワークで、モジュールサイズは (A) $N_m = 5$ 、(B) $N_m = 20$ 、(C) $N_m = 50$ 、(D) $N_m = 100$ である。線色は反モジュール化における、リワイヤするイントラリンク数の比率 w' に対応する。 87
- 4.12 平均最短経路長 L ネットワークサイズ N の関係。 $d = 9$ のネットワークで、モジュールサイズは (A) $N_m = 5$ 、(B) $N_m = 20$ 、(C) $N_m = 50$ 、(D) $N_m = 100$ である。線色は反モジュール化における、リワイヤするイントラリンク数の比率 w' に対応する。 87
- 4.13 平均最短経路長 L ネットワークサイズ N の関係。 $d = 19$ のネットワークで、モジュールサイズは (A) $N_m = 5$ 、(B) $N_m = 20$ 、(C) $N_m = 50$ 、(D) $N_m = 100$ である。線色は反モジュール化における、リワイヤするイントラリンク数の比率 w' に対応する。 88

4.14	従来のモジュール化法を適用したネットワーク ($d = 9$, $m_o = 100$) の次数分布。モジュール化においてリワイヤするインターリンク数の比率 w は (A) $w = 0.988$ 、(B) $w = 0.695$ 、(C) $w = 0.494$ である。各 w は式 4.1 により $w' = 0.01, 0.3, 0.5$ に対応する値である。	89
4.15	修正したモジュール化のプロセス。格好中の数字はプロセスの各ステップである。黄色の円はモジュールである。黒点線と青色線は、各自インターリンクとイントラリンクを指す。	90
4.16	修正したモジュール化と反モジュール化を適用したネットワークの、MB 攻撃に対する頑健性比較。MB 攻撃後の最大連結成分のサイズ比率は、 $d = 9$ でモジュール数 m_o が (A) $m_o = 100$ 、(B) $m_o = 20$ 、(C) $m_o = 5$ であるネットワークで測定した。点線と実線は、修正されたモジュール化における w でのリワイヤリングによる結果と反モジュール化における w' でのリワイヤリングによる結果を表す。	91
4.17	修正したモジュール化と反モジュール化を適用したネットワークの、IB 攻撃に対する頑健性比較。MB 攻撃後の最大連結成分のサイズ比率は、 $d = 9$ でモジュール数 m_o が (A) $m_o = 100$ 、(B) $m_o = 20$ 、(C) $m_o = 5$ であるネットワークで測定した。点線と実線は、修正されたモジュール化における w でのリワイヤリングによる結果と反モジュール化における w' でのリワイヤリングによる結果を表す。	91
4.18	修正したモジュール化と反モジュール化を適用したネットワークの、RF に対する頑健性比較。MB 攻撃後の最大連結成分のサイズ比率は、 $d = 9$ でモジュール数 m_o が (A) $m_o = 100$ 、(B) $m_o = 20$ 、(C) $m_o = 5$ であるネットワークで測定した。点線と実線は、修正されたモジュール化における w でのリワイヤリングによる結果と反モジュール化における w' でのリワイヤリングによる結果を表す。	92
5.1	Cost266 で各ノードがデータセンターとして選ばれる頻度。(A)GH 法による結果、(B)MH 法による結果	101
5.2	Cost266 に両方を用いて最小数のデータセンターが配置された場合のデータセンターが持つ負荷の分散値。	102
5.3	各配置法で、攻撃後のネットワークに代替を設置する。(a) 2 ホップを再カーバーする必要な代替数、(b) 3 ホップを再カーバーする必要な代替数	104

5.4	2 ホップカーバーするために、5 個のデータセンターが設置された場合の結果 (A) データセンターと利用者間の平均距離、(B) 負荷の分散、(C) 負荷の最大値。	105
5.5	3 ホップカーバーするために、3 個のデータセンターが設置された場合の結果 (A) データセンターと利用者間の平均距離、(B) 負荷の分散、(C) 負荷の最大値。	106

表目次

3.1	攻撃以前、インフラネットワークの基本性質。 N と M は、それぞれネットワークのノード数とリンク数である。 k_{avg} 、 k_{max} 、 k_{min} は、平均、最小、最大次数である。 D はネットワークの直径で、 r は次数相関を示す。	40
3.2	最大連結成分にあるリンク数。 M_{Heal} は修復されたネットワーク全体のリンク数である。 M_{Comp} はもし最大連結成分が完全グラフである場合のリンク数である。	47
3.3	qN 個のノードを破壊した後、修復されたネットワークの最大次数。	48
4.1	イントラリンクとインターリンクの最初値。	71
4.2	$d = 9$ と $m_o = 200$ ネットワーク上で、3種類の攻撃 (MB、IB、RF) に対する、攻撃率 q の臨界値 q_c 。	81
4.3	$d = 4$ と $m_o = 2000$ ネットワーク上で、 w' に対する Q 、IR(全体リンク数に対するインターリンク数の比率)、従来のモジュール化における w を示す。	83
4.4	$d = 4$ と $m_o = 100$ ネットワーク上で、 w' に対する Q 、IR(全体リンク数に対するインターリンク数の比率)、従来のモジュール化における w を示す。	83
4.5	$d = 4$ と $m_o = 5$ ネットワーク上で、 w' に対する Q 、IR(全体リンク数に対するインターリンク数の比率)、従来のモジュール化における w を示す。	84
4.6	$d = 9$ と $m_o = 1000$ ネットワーク上で、 w' に対する Q 、IR(全体リンク数に対するインターリンク数の比率)、従来のモジュール化における w を示す。	84
4.7	$d = 9$ と $m_o = 100$ ネットワーク上で、 w' に対する Q 、IR(全体リンク数に対するインターリンク数の比率)、従来のモジュール化における w を示す。	84

4.8	$d = 9$ と $m_o = 5$ ネットワーク上で、 w' に対する Q 、IR(全体リンク数に対するインターリンク数の比率)、従来のモジュール化における w を示す。	84
4.9	$d = 19$ と $m_o = 500$ ネットワーク上で、 w' に対する Q 、IR(全体リンク数に対するインターリンク数の比率)、従来のモジュール化における w を示す。	85
4.10	$d = 19$ と $m_o = 100$ ネットワーク上で、 w' に対する Q 、IR(全体リンク数に対するインターリンク数の比率)、従来のモジュール化における w を示す。	85
4.11	$d = 19$ と $m_o = 5$ ネットワーク上で、 w' に対する Q 、IR(全体リンク数に対するインターリンク数の比率)、従来のモジュール化における w を示す。	85
5.1	実ネットワークの基本特性	98
5.2	最短経路の平均値	100

第1章

緒言

1.1 研究の背景と目的

現代社会には交通網、電力網、水道網、通信網など、様々なインフラネットワークが存在する。これらは、都市計画、経済発展、公共安全、環境管理など多くの社会的あるいは経済的な活動の基盤となり、その複雑さと重要性が持続的に増加している。ネットワーク科学では、これらの構造を点（ノード）と線（リンク）で構成されたネットワークとして単純化し、ネットワーク内の情報、資源、またはそれらの流れを分析する。ネットワーク科学における重要な発見の1つとして、ほとんどの現実ネットワークが共通的に持つ構造特性を明らかにしている [1]。SF ネットワークの特徴は、次数 k （ノードに繋がっているリンク数）の分布 $P(k)$ が $P(k) \sim k^{-\gamma}$ のべき乗測に従うことである。 $P(k)$ は次数が k であるノードの割合を表し、現実ネットワーク上では主に $2 < \gamma < 3$ を持つ。言い換えると、この次数分布は、少数のノードが非常に多くのリンクを持っており（これをハブという）、ほとんどのノードは相対的に少ないリンクを持つことを示す。ほぼ全てのノードがハブに連結されているため、ハブによりネットワーク内のノード間の経路を短縮できるので、ネットワークは情報拡散において高い効率性を持つ。また、SF ネットワークは偶発的な故障にも全体の連結性を保持できるため、頑健だと言えるが、重要なハブを狙う攻撃に対してはネットワークの連結性がすぐ分断され、機能も麻痺する深刻な問題が存在する [2, 3]。ゆえに、ハブを持つネットワークは情報拡散には有利だが、構造的には脆弱な特性を持つ。

この構造的な弱点に加えて、多くのネットワークは自然災害やテロ攻撃に頻繁に脅かされている。例えば、イギリスの空港における未知の攻撃者によるたった数件のドローン事故が、ヨーロッパで年間1千に達する運航を妨げ [4]、悪天候や自然現象による多数の空

港閉鎖が起こった [5]。通信システムのバックボーンであるインターネットも、DDoS 攻撃によって絶えず脅かされている [6]。北米での大停電の主な原因はささやかな障害事故である [7]。さらに、2011 年に日本で発生した大地震と大津波によって、グローバルな供給網が損傷した [8]。インフラネットワークは現代社会を維持するための重要な機能を果たしているため、様々な危機に対する持続可能性を確保するためには、内在的な弱点を克服し、頑健な構造に再構築する必要がある。

従来の頑健性研究では、べき乗則の次数分布を持つ SF ネットワークと、ポアソン分布の Erdős-Rényi (ER) ランダムグラフをピンポイントに対象としてきた。ここで、SF ネットワークは、新しいノードがリンクを生成する時に既存のノードの次数に比例して連結するような、優先的選択の利己的規則によって生成される。また頑健性研究は次数分布 $P(k)$ に着目しているが、繋がっているノード同士の次数の関係 (次数相関 $P(k'|k)$) にも注目している。例えば、悪意のある攻撃に正の次数相関を持つ玉ねぎ状構造が最適な頑健性を持つことを確認されている [9,10]。しかしながら、あまりにも高い次数相関を持つと、ネットワークはむしろ低い頑健性を持つ [10–12]。ゆえに、頑健な構造を作るためには新しい原理に注目する必要性があると考えられる。

近年、頑健なネットワークの構築には、ループ構造が、従来の次数相関により重要であることが徐々に明らかになってきた。実際、頑健な玉ねぎ状構造を成長しながら生成する方法として、複製または仲介による手法が提案された [13–15]。このような逐次成長法は、従来のネットワーク全体のつながりを切り替える方法 [9,10] より現実的な方法である。また、頑健性に対するループ構造の重要性は、ネットワークを破壊する最悪の方法がループを削除することと漸近的に同等である事実から支持される [16]。すなわち、ネットワークからループを全部削除すると木構造になるが、木構造は少数の節ノードを削除することで連結成分がすぐにバラバラになる構造である。したがって、ループを強化してネットワークが木構造にならないようにすることが、頑健な構造を作るために非常に重要となる。そこで最近の研究として、ループ強化に基づいて頑健な構造を構築する種々のリワイヤリング法が提案された。このループ構成に必要な不可欠なノードを最大化するリワイヤリング法で、次数相関を増加することよりもさらに頑健なネットワークを作ることができる [17]。但し、最大化を測るためのループ構成に必要な不可欠なノード集合 (Feedback Vertex Set、FVS) を探すことすら NP 困難であり、研究 [17] ではループを強化するために近似法を使っている。この近似法は、統計物理学の Cavity 法に基づいたメッセージ伝搬式である [18]。また種々のリワイヤリング法 [17] による結果から、頑健なネットワークは次数分布が均質化することが共通に確認されている [17]。次数分布の均質化は、最大次数と最小次数の差が減少することに相当する。均質な次数分布を持つ頑健なネット

ワークは、SF ネットワークの生成原理である利己的な優先的選択ではなく、逆優先的選択 (Inverse Preferential Attachment, IPA) により生成できる [19]。なおかつ、極端的な例として、差が0であるランダム d -レギュラグラフが最適な頑健性を持ち、この構造にはハブのような構造的な弱点が存在しない [20]。レギュラグラフが持つ最適な頑健性は、二峰性の網羅的な離散摂動と多峰性のランダム摂動との比較で検証されている [20]。

以上の背景を踏まえて、本稿では次の3点を具体的な目的とする。各目標の概説は以下のようなものである。

目的1 ループ強化と次数分布の均等化に基づいて自己修復法を提案し、それをネットワーク上で分散的に処理できるアルゴリズムを記述する。

目的2 コミュニティ構造がネットワークの頑健性に与える影響を明確にする。

目的3 頑健で均等な負荷分散を持つ、情報拡散の拠点の配置法を提案する。

目的1 に対して、頑健な構造を作る新しい原理であるループ強化と次数分布の均質化に基づいて、レジリエンスの立場から、被害を受けたネットワークを再構築する自己修復法を提案する [KIM1]。レジリエンスは、特にシステム工学や環境科学から近年注目を集めている概念である。レジリエンスの定義は、環境変化にもかかわらず、システムが必要な機能を維持し、必ずしも元の構造に戻らなくても、変化する環境条件に適応能力を持ってさらに強くなるように再構築することも意味する [21, 22]。先に、ループ強化に基づいてネットワークを再構築する自己修復法が提案されている [23]。この自己修復法は、従来のヒューリスティック手法 [24] より、頑健な構造を再構築できる。また、従来の修復法としてコンピュータ科学における Forgiving-Tree アルゴリズム [25] が提案されたが、このアルゴリズムは短い経路長になることに着目はするが、脆弱な二分木を生成するので、頑健性は考慮されてない。一方、頑健な構造を再構築できる自己修復法 [23] はまだ改善の余地があると考えられる。なぜなら、先の自己修復法 [23] ではループを強化するため、FVS の計算に高い計算量を使うからである。したがって、本研究で提案する自己修復法 [KIM1] では、まず被害を受けたノード間を、1番簡単なループであるリング状に連結する。そして、次数分布の幅を狭くするために、低い次数を持つノードに修復リンクを追加する。これは、一般的に攻撃で高い次数のノードが取り除かれる傾向があるからである。さらに [KIM1] では、被害を受けたネットワーク上で、各ノードが再構築のために自ら修復先を判断し、お互い協力しなら再構築を行う、自律的なネットワークシステムための分散アルゴリズムを提案する。なぜなら、インフラネットワークは常に大規模で複雑であり、従来の中央制御では全てのデータ処理や資源管理がハブのような単一の中央司令塔で行われるから、中央司令塔が壊れるだけでシステム全体が麻痺する可能性があるので分散

制御が望ましい。この分散型のシステムの長所は第2章でさらに説明する。[KIM1]の結果から、提案する修復法が、従来の自己修復法 [23] より頑健で高い効率性を持つ構造へ再構築できることと、再構築されたネットワークの次数分布が均質化されることを示す。但し、効率性を元の構造より高くするためには、ほぼ全ての修復資源 (リンク) を使う必要があることも確認された。

目的2に対して、最適の頑健性を持つレギュラグラフでも、コミュニティ構造を考慮するとSFネットワークより脆弱になれることを示す [KIM2]。コミュニティ構造は、いろんな分野の現実ネットワークで発見され、共通の興味や背景を持つノードの集合を意味する。このコミュニティを識別することで、ネットワークを応用目的に対して効果的に分析できる。例えば、マーケティングにとって、似ている興味を持つグループを発見することは、ターゲット広告に重要になる。また詳細は第2章で述べるが、ネットワークを構成要素で持つソーシャル・キャピタルは、組織内外の繋がりにより、大きく結合型と橋渡し型に分類される。本稿では、[KIM2]の結果が持つ社会学的な意義についても議論する。最新の研究 [26] では、Erdős-Rényi (ER) ランダムグラフやSFネットワーク上で、同じコミュニティにあるノード同士だけ繋がっていると、以前にはなかった新しい弱点が発することを示されている。しかしながら、この研究 [26] には二つの問題がある。まず、コミュニティ構造と頑健性間の関係を示すために、異なるコミュニティ間を繋ぐリンクを、コミュニティ内を繋ぐように張り替えながら頑健性を測るが、この際次数分布が変化する。次に、その張り替えにより、孤立ノードが生じることがある。次数分布は頑健性に大きな影響を与えるので、研究 [26] に基づく実験ではコミュニティ構造と頑健性間の関係を明らかにできない。したがって、コミュニティ構造がネットワークの頑健性に与える影響を明確にするために、[KIM2] では次数分布の幅が0で最適の頑健性を持つ d -レギュラグラフを用いて、次数分布を保存する切り替え法を用いる。詳しくは切り替えの対象になるネットワークを初期構成して、多数の d -レギュラグラフを一つの輪に連結する。これでコミュニティ間を繋ぐリンク数は最小になる。そのネットワークで、コミュニティ内の連結をコミュニティ間の連結へ切り替える方法 (反モジュール化) を行い、切り替えの比率を変えながら、ネットワークが持つ攻撃に対する頑健性と最短経路の効率性を調べる。本研究では、例え最適の頑健性を持つレギュラグラフでも、反モジュール化により新しい弱点が生じることが示す。

目的3に対して、FVSを求める際に用いた、統計物理学的なメッセージ伝搬による近似解法を応用した手法が、均等な負荷を持つ局所的な情報拡散の拠点を設置することに役立つことを示す [KIM3]。[KIM3] では分散処理の目的の一つである負荷分散に注目して、均等な負荷を持つ局所的な情報拡散の拠点を設置する方法を考える。負荷分散に注目する

理由は、[KIM1] で修復法ための処理を全てのノードに分散させたが、各ノードの処理量が均等である保証はないからである。SF ネットワークにおいて、ハブは情報拡散の重要な役割を果たしたので構造的な弱点になったことと、またそのような中央制御は信頼性が低いことから、頑健な構造には処理先での負荷の均等さを考える必要があると考えられる。実際、Content Delivery Network(CDN) において、多数の局所的な拠点 (データセンター) を各地へ分散して配置することで、動画などコンテンツを提供している。従来、情報拡散の拠点として中心性を用いてハブのようなノードを選ぶことが多いが、利用者数とコンテンツ容量は過去より大きく増加しつつあるので、拠点にかかる負荷を分散させることが、利用者が高品質のサービスを体験するのに重要となる。ゆえに、従来の配置方法とは異なり、情報科学のマルチホップ被覆と、統計物理学的なメッセージ伝搬による近似解法を融合した方法で選定されたノードを情報拡散の拠点にすることで、より均等に負荷を分散させることができ、頑健な構造であることを示す [KIM3]。

これらの研究 [KIM1,KIM2,KIM3] では次数分布を含めてループとコミュニティ構造のような、ノード間のより高次元の相関関係に注目して、頑健なネットワーク構築を目指している。本稿で示す結果から、従来の次数相関より、悪意のある攻撃に対して頑健で、さらに情報拡散に対して高い効率性も持つネットワークの構築に効果的であることが分かる。また、研究 [KIM1,KIM2,KIM3] は分散システムの高い頑健性も示している。[KIM1] では分散処理ができるアルゴリズムを記述して、攻撃に脆弱な中央集権システムから、全てのノードが自律的に動作する分散システムへ転換した。[KIM2] ではハブのような一か所で多くの動作をするノードが生じないために、実ネットワーク上で、ノードが均等な作業負荷を持つ配置法を提案し、頑健な配置構成を示した。但し、ネットワーク上で、このような分散環境を構築する際、構造的な分離 (お互い稀に連結するコミュニティ) は、以前には存在しなかった新しい脆弱性をもたらすことも [KIM3] から明らかにした。したがって、頑健なシステムためには、機能的には従来の中央集権型から、分散型への履行を考慮する必要はあるが、構造的な分離を注意すべきであることと言える。

1.2 論文の構成

本論文では、全体を 6 章で構成している。

第 1 章の緒言に続いて、第 2 章では、本研究で扱うネットワークの基本定義と、次数、次数相関、頑健性指標、効率性指標、コミュニティに関する指標 (モジュール性) を説明する。また、次数相関、ループ強化に基づいた頑健性研究、ハブのような重要ノードを探索する研究、コミュニティ構造と頑健性関係に対する、これまでの研究成果について概説

する。

第3章では、攻撃を受けたネットワークを頑健な構造へ再構築する自己修復法と、それを分散アルゴリズムについて説明する。まず、再構築する理由を述べて、提案する自己修復法の説明する、なおかつ、次数分布を均質化すること、分散処理を前提に修復資源でリンクを隣からすぐに調達するような制御法を使うことが、提案法と従来の自己修復法とは違うことを説明する。提案法で再構築されたネットワークが、従来の修復法で再構築されたネットワークより高い連結性、頑健性、効率性を持つことを示す。また、提案する修復法を分散処理アルゴリズムとして記述する。

第4章では、コミュニティ構造と頑健性の関係について明らかにする。1番高いモジュール性を持つレギュラグラフの設定し、そのグラフ上でコミュニティ内のリンクを、コミュニティ間のリンクへ切り替えることでモジュール性を調整する方法(反モジュール化)について紹介する。反モジュール化により、モジュール性が減少することで、頑健性と効率性が両方高くなることを示す。また、従来のモジュール化が次数分布を保存しないため、それを修正する方法を説明する。

第5章では、情報拡散の拠点であるデータセンターを配置するために、ハブのような重要ノードを探索する方法を紹介する。そして、その方法で求められた配置構成の効率性、負荷分散を比較する。比較対象になる、従来の配置法と、統計物理的なアプローチとマルチホップが融合した新しい配置法について説明する。両配置法で、作られた配置構成の負荷の分散値、最短経路の平均値を示す。また、1番高い負荷を持つデータセンターを壊す攻撃に対して、代替サーバーを設置する方法によって、負荷の分散値と最大値、そして最短経路の平均値が低くなることを示す。

第6章では、結論として以上の成果をまとめる。

第2章

本研究に対する先行研究

2.1 ネットワークの特性

2.1.1 基本定義

本研究で用いるネットワークは、グラフ理論のグラフ $G = (V, E)$ と同じで、それは頂点(ノード)の集合 V と、辺(リンク)の集合 E で構成される。ネットワーク内の全てのノードは、 i と j のような識別子が付けており、例えばノード i と j 間にあるリンクは (i, j) と表記する。但し、リンクの繋ぎ方として、リンクの両端ノードが同一であること(自己ループ)、2つのノード間に複数のリンクがあること(重複リンク)は認めない。また、本研究ではリンクの方向性と、重みを考慮しない、無向・非重み付けグラフを扱う。このようなグラフで、ノード i の次数 k_i は以下の式で定義される。

$$k_i = \sum_{j=1}^N A_{ij}. \quad (2.1)$$

ここで、 $N = |V|$ はネットワーク全体のノード数、 A はグラフ G に対する隣接行列である。隣接行列の構成要素 A_{ij} の値は、1 (ノード i と j 間にリンク有り)、もしくは0 (ノード i と j 間にリンク無し) である。つまり、次数 k_i は、ノード i に繋がっているリンク数を指す。

2.1.2 次数相関

次数を用いてネットワークを分析する方法として、次数分布において、あるノードの次数が k である確率 $P(k)$ が議論される。後述するが、このによりネットワークが持つ性質

が異なる。また、分析のために、繋がっているノード同士の次数の関係（次数相関）も議論される。まずあるノード i において、次のような最近接ノードの平均次数 $k_{nn}(k_i)$ を求める。

$$k_{nn}(k_i) = \frac{1}{k_i} \sum_{j=1}^N A_{ij} k_j. \quad (2.2)$$

ここで、 nn は最近接の略語である。このように、全てのノードに対して、次の式で計算できる。

$$k_{nn}(k) = \sum_{k'} k' P(k'|k). \quad (2.3)$$

ここで、 $P(k'|k)$ は、ランダムに選んだリンクの両端が、次数 k と k' のノードである確率である。つまり、 $k_{nn}(k)$ は、次数が k である全てのノードにて、最近接ノードの平均次数を指す。次数相関は、次数 k と $k_{nn}(k)$ 関係にて定義される。もし k が増加するほど、 $k_{nn}(k)$ の増加すると、ネットワークは正の次数相関を持つと言われる。そのネットワークでのリンクは、次数がほぼ同じであるノード同士を繋ぐ。逆に、 $k_{nn}(k)$ が k により減少すると、ネットワークは負の次数相関を持ち、次数の差が大きいノード同士が繋がっていることを意味する。Pearson の相関係数を用いると、次の式のように、次数相関を一つの値 ($-1 < r < 1$) で表現できる。

$$r = \frac{\sum_{ij} (A_{ij} - k_i k_j / 2M) k_i k_j}{\sum_{ij} (k_i \delta_{ij} - k_i k_j / 2M) k_i k_j}. \quad (2.4)$$

ここで、 $M = |E|$ はネットワークにある全てのリンク数で、 δ_{ij} の値は 1 ($i = j$)、もしくは 0 ($i \neq j$) である。

2.1.3 頑健性

ネットワークの中にあるノードは、直接繋がっている隣のノードを通じて、より遠くにあるノードへ辿ることができる。連結成分は、任意の 2 つのノード間に必ず経路が存在するような部分グラフである。最大連結成分は、その連結成分の中でノード数 (サイズ) が 1 番多いものを指す。偶発的な故障や悪意のある攻撃に対して、ネットワークの頑健性を測る指標 R [11] が、次のように提案された。

$$R = \frac{1}{N} \sum_{Q=1}^N \frac{S(Q)}{N}. \quad (2.5)$$

ここで、 N はネットワークのノード数、 $S(Q)$ は故障や攻撃により Q 個のノードが削除された場合の最大連結成分のサイズである。つまり、 R はネットワークでノードを一つずつ

削除するたびに、どれぐらいのノードが他のノードとの連結性を維持しているかを観測するものである。 R 値の範囲は $\frac{1}{N} \leq R \leq 0.5$ であり、数値が大きいほどネットワークは頑健である。

2.1.4 効率性

本研究で高い効率性は、ノード同士の経路の長さが短いことを指す。なぜなら、ノード間の経路の長さが短いほど、コストや時間が掛からないよう、ノード同士の物や情報などの効率的な交換ができるからである。したがって、効率性指標 E を次の式で計算する。

$$E = \frac{1}{N(N-1)} \sum_{i,j=1, i \neq j}^N \frac{1}{d_{ij}}。 \quad (2.6)$$

ここで、 N はネットワークのノード数、 d_{ij} はノード i と j 間の最短経路の長さである。また本研究で、経路の長さはホップ数（リンク数）で数える。

2.2 スケールフリー (SF) ネットワーク

N 個のノードにおいて、あるノード同士間に確率 p でリンクを生成することで作られた Erdos-Renyi(ER) ランダムグラフは、ネットワーク科学が注目される以前から数学でよく研究されたグラフである。しかしながら、現実のネットワークはランダムに生成されていないし、ER ランダムグラフが持つポアソン次数分布を持っていないので、ER ランダムグラフでは現実のネットワークの構造や性質を十分に反映できない問題があった。これに対して、多くの現実ネットワークが、スケールフリー構造を持つことが今世紀初頭から明らかになった。スケールフリー構造では、次数分布がべき乗則に従う [1]。

$$P(k) \propto k^{-\gamma} \quad (2.7)$$

現実のネットワークは主に $2 < \gamma < 3$ のべき次数を持つ。SF ネットワークでは、ごく一部のノード（ハブ）が多くのリンクを持ち、ほとんどのノードは比較的少ないリンクを持つ。これらはソーシャルネットワーク、インターネット、タンパク質相互作用、学術共著など、様々な分野で見られる。SF ネットワークは 1999 年の Barabasi と Albert によって広められた優先的選択メカニズムにより生成できる [3]。このメカニズムは、ネットワーク生成の際に、新しいノードが既に多くのリンクを持つ既存のノードに繋がる可能性が高いことである。つまり、金持ちがより豊かになれる、ことを示唆する。この過程により、少数のハブが出現する。ハブを持つネットワークは、偶発的な故障にも全体の連結

性を保持できるため、頑健だと言える。しかしながら、ハブを狙って攻撃すると、ネットワークはすぐに連結もすぐ分断され、機能が麻痺する問題がある。

2.3 中央集権システムから分散システムへの転換

分散システムは、複数の独立した構成要素が互いに連結され、まるで一体のように機能するシステムであり、その発展過程はコンピューティング技術の進化と密接に関連している。初期には単純なネットワーク接続とデータ共有に重点を置いていたが、時間が経つにつれてより複雑な作業処理、データ保存、資源共有などができる技術が開発されてきた。分散システムは、独立した各構成要素が協力して効率的に作業を行う。これは中央集権的システムと対比される。中央集権的システムはすべてのデータ処理と資源管理が単一の中央司令塔によって行われる構造であり、これはボトルネック現象やシステム全体の失敗リスクを増加させる可能性がある。一方、分散システムはシステムの一部が故障しても全体のシステムが持続的に機能できる能力を提供する。分散システムはこのような長所から、大規模な計算が必要な科学研究、大規模ウェブサービス、クラウドコンピューティングなど様々な分野で活用されている。これらに応じて、分散システムの目的は次の6つである [27]。

負荷分散 大きな仕事を複数で分担させてシステム全体の性能向上を図る

処理分散 データの発生やサービスを必要とする場所で直接処理を行うことで輸送費用や応答時間の減少を図る

機能分散 個別の機能を持つ特定の処理装置を用意し各機能をシステム全体へ共有させることで応答時間や信頼性に役立つ

危険分散 一部が故障しても他の処理装置があるのでシステム全体がダウンされないよう信頼性を向上する

管理分散 異なる利用者や環境が集めて一つのシステムを構成しても個別の運用管理ができるようにする

拡張分散 システムの拡張要求に応じて機能の追加と技術の進歩の取り込みを容易にする

これらを目的とする分散システムは分離と透明性という特徴を持つ。分離はシステムの構成要素が物理的あるいは論理的に分離されていることを示す。本研究の5章で物理的な分離がもたらせる脆弱性について説明する。また、透明性は分散システムが多数のコンピューターやサーバーを、まるで単一システムのように見せることから生じられる特徴で

ある。透明性の種類は次の6つがある。

アクセス 利用者はデータ、資源、処理装置が、身近にあるいは遠くにあるかに関係なく同一な操作法でアクセスできる

並行アクセス 複数の利用者はお互い干渉なしで並行してアクセスできる

位置 利用者はデータ、資源、処理装置がどこにあるかを知らなくても、それらを利用できる

重複 信頼性や性能の向上ため、データや資源のコピーが存在してもそれを意識せずに利用できる

障害 処理装置や回線に部分的な障害が発生しても、利用者はそれを意識せずに利用できる

システムの変化 負荷分散ための仕事の移動、データや資源の位置変更、システム仕様の変更などに関係なく、利用者はそれを意識せずに利用できる

2.4 頑健な構造の先行研究

2.4.1 次数相関と頑健性

従来、頑健性指標を用いて頑健な構造を構築することが議論された。研究では、次数が高いノードから順番に破壊する攻撃(次数順攻撃)に対する頑健性指標が増加するように、次数分布を保存しながらリンクの繋がりを変える貪欲的なリワイヤリング法を提案した [11]。

1. 一様ランダムにリンク (i, j) 、 (k, l) を選ぶ。
2. それらのリンクを切って、新しくリンク (i, k) 、 (j, l) を作る。
3. このリワイヤリングにより、ネットワークの頑健性指標の値が増加すると、1を繰り返す。もし、増加しなかったら、リンクを元に戻して、1を繰り返す。

このリワイヤリング法を SF ネットワークに適用すると、玉ねぎ状構造が作られ、それは悪意のある攻撃に非常に強い頑健性を持つ構造であることが示された。玉ねぎ構造は中心部は次数が高いノードがお互い連結され、その中心を次数が低いノードが幾層にも囲むことで可視化される。各層は次数が同じノード同士が繋がっていて、正の次数相関を持つ。この構造が次数順攻撃を受けると、被害は主に中心部にあるノードが受け、周辺にある低次数のノードはそのままお互い繋がっているため、高い連結性を維持できる。なおかつ、

頑健性指標ではなく、次数相関 r が増加するように、上記の方法と同じリワイヤリングをしても、高い頑健性を持つ構造が作られることが示された。これらの結果から、頑健性と次数相関は正比例関係を持つように見えるかも知れないが、 $r > 0.4$ の場合にはむしろ頑健性が低くなる問題が示された [10–12]。ゆえに、次数相関による頑健性の極大化ができないため、頑健な構造を作られる新しい原理が必要になった。

2.4.2 ループと頑健性

近年の研究では、従来の次数相関ではなく、頑健な構造に対してループの重要性を強調している。ループは始点と終点が同一な経路である。つまり、あるノードから出発し、多数のノードを経て、元に戻る構造である。このようにネットワーク上で、特定のパターンが複数のノード間でしきりに現れると、それをモチーフだとする。モチーフ研究はネットワークの構造的、機能的な特性の理解に貢献している。例えば、生物学的なネットワーク、特に遺伝子やタンパク質の相互作用ネットワーク上で、3つ、あるいは4つのノードで構成されるモチーフは細胞内の複雑な信号伝達経路を表す。

上記の同じく頑健性に対する研究についても、従来の次数相関では2つのノード関係に注目したことから、ループのような3つ以上のノード関係に目を移している。実際、仲介と複製によりループ構造を作り出すことで、高い頑健性と次数相関を持つ玉ねぎ構造が生成された [13–15]。また、ループの重要性は、ネットワークを分断する Dismantling 問題と、ネットワークからループを消す Decycling 問題が、ネットワークサイズが無限大に発散する場合において漸近的に等価である事実から支持される [16]。ここで、Dismantling 問題はグラフ G での最大連結成分サイズを、定数 C より小さくするために必要な削除ノード数の最小比率 $\theta_{dis}(G, C)$ を、Decycling 問題はグラフ G から全てのループを消すために必要な削除ノード数の最小比率 $\theta_{dec}(G)$ を求めることである。研究では、各問題が次数分布 q_k のランダムグラフで、次のように定義されることを示した。

$$\theta_{dis}(q_k) = \lim_{C \rightarrow \inf} \lim_{N \rightarrow \inf} E[\theta_{dis}(G, C)], \theta_{dec}(q_k) = \lim_{N \rightarrow \inf} E[\theta_{dec}(G)]. \quad (2.8)$$

ここで、 $E[\cdot]$ は次数分布 q_k のランダムグラフアンサンブルにおける平均値である。研究では、任意の次数分布では $\theta_{dis}(q_k) \leq \theta_{dec}(q_k)$ であるが、 $\langle k^2 \rangle < \inf$ なら、 $\theta_{dis}(q_k) = \theta_{dec}(q_k)$ であることを示した。ネットワークからループを全部除去するために必要なノード集合は、コンピュータ科学では Feedback Vertex Set (FVS) だとも呼ばれている。しかしながら、最小 FVS の厳密解を求めることは組み合わせ最適化問題の NP 困難な組み合わせ問題に属するので、多項式時間で解くアルゴリズムが存在しない。したがって、統計物理

学の Cavity 法に基づいたアルゴリズム [18] により、近似解を求める方法が提案された。Cavity 法は、複雑なシステム上で、強く相互作用する多体システムの統計的な特性を理解するための方法である。システムから1つの構成要素を削除した後、残りがどのように作動するかを観察する。それで削除された構成要素が、システムに与える影響を評価できる。近接する構成要素同士は互いにメッセージをやり取りするので、Feedback が繰り返す可能性がある。したがって、平均場近似を適用して、各構成要素が平均的な効果によって独立で作動すると仮定する。この方法に基づいて、各ノードが FVS に属する確率を求める。アルゴリズム [18] では、FVS に属する確率が1番高いノードを選んで削除して、確率を再計算する。この作業をネットワークが木構造になるまで、繰り返す。そうすると、木構造になるまでに削除されたノード集合が FVS だとする。このように、ネットワークからループを全部削除すると木構造になるが、木構造は少数の節ノードを削除することで連結成分がすぐにバラバラになる構造である。したがって、ループを強化してネットワークが木構造にならずにすることが、頑健な構造を作ることに大事である。実際、ループ強化に基づく、FVS を増やす下のようなリワイヤリング法 [17] が提案された。

1. FVS に属する確率が最大な、お互い繋がっていないノード i と j を選ぶ。
2. 選んだノードの各最近接から、それぞれ FVS に属する確率が最小なノード k と l を選ぶ。
3. 既存のリンク $(i, k), (j, l)$ を切って、新しいリンク $(i, j), (k, l)$ を追加する。
4. 全てのノードに対し、FVS に属する確率を再計算する。

次数分布を保存する、ループを強化するこのリワイヤリング法では、従来の次数相関を高めるリワイヤリング法より高い頑健性を持つネットワークを作ることができる。また、下のような、次数分布を変化させる、ループを強化するこのリワイヤリング法 [17] では、ネットワークは負の次数相関であっても、高い頑健性を持つことを確認した。

1. FVS に属する確率が最大なノードを両端にするリンクを削除する。
2. FVS に属する確率が最小なノード間に新しいリンクを追加する。
3. 全てのノードに対し、FVS に属する確率を再計算する。

さらに、そのリワイヤリング法より、変化させる次数分布は、最大次数と最小次数間の幅が減少することも確認した。この次数分布の均質化現象が頑健性に与える影響について、あらゆる次数分布を持つランダムネットワークでの頑健性を調べた研究がある [20]。その結果では、次数分布の分散が小さくなるほど、頑健性と最小 FVS のサイズが増加することを示した。したがって、次数分布の幅が0である、ランダムレギュラグラフが最適の

耐性を持つことになる。上記の結果から、頑健な構造のためには、従来研究されてきたノードとノード間の関係 (次数相関) に注目することより、ループ構造や次数分布のような、多数ノードの関係に注目することが重要であることが明らかになった。したがって、私達はループ強化と次数分布の均質化に着目して、第 3 章でネットワークを頑健な構造へ再構築する分散アルゴリズムを提案する。

2.4.3 コミュニティと頑健性

多数ノードの相関関係について、ループだけではなく、コミュニティも頑健性について重要な役割をしている。コミュニティとは、ネットワーク内で、互いに強く連結されているノードの集合である。これらのノードは、ネットワークの他の部分にあるノードと比較して、互いにより多くの連結を持っている。このような密集した多数ノードの連結型は、ネットワークをより小さく管理しやすい部分に分割できる。また、コミュニティを分析することで、ネットワーク内で特定の機能をするノード集合を識別できる。例えば、生物学的ネットワークで、各コミュニティは特定の生物学的機能を遂行する。ソーシャルネットワークで、コミュニティは、人々がどのようにグループ化され、相互作用するかを示す。これは、ソーシャルメディア、コミュニケーションパターン、マーケティング戦略などの分析に重要な影響を与える。

モジュール性 Q [28] は、ネットワークがどれぐらいうまくコミュニティに分割されているかを測定する指標で、次の式で求められる。ここで、モジュールはコミュニティに対応する。

$$Q = \frac{1}{2M} \sum_{i,j} (A_{ij} - \frac{k_i k_j}{2M}) \delta_{i,j}, \quad (2.9)$$

ここで、 A は隣接行列、 k_i はノード i の次数、 $\delta_{i,j}$ はもしノード i と j が同じモジュールに属すると 1、そうではないなら 0 をとする。高い Q を持つネットワークは、コミュニティ内部の連結がコミュニティ間の連結より多いであることを意味する。ゆえに、異なるコミュニティ間を繋ぐノードが、構造的な弱点になりやすいことが示された [26, 29]。

研究 [29] ではモジュール内のリンク数とモジュール間のリンク数の比を一定にして、ネットワークが特定数値以下のモジュール数を持つと、モジュール間を連結するノードを攻撃することに脆弱だということを示した。但し、この研究ではネットワークモデルとして ER ランダムグラフと SF ネットワークだけを扱い、以降第 6 章で見せるだろうが、頑健性に対してモジュール数よりモジュール性が重要である。

また、研究 [26] では SF ネットワークを対象に、 Q と R の関係について述べた。ネッ

トワークの Q は次のようなリワイヤリング法で調節される。

1. 全てのノードはモジュール番号を一様ランダムに割り当てられる。この時、各モジュールに属するノード数はほぼ同一である。
2. モジュール間を繋ぐ各リンクにおいて、確率 w で以下の過程を実行する。
 - (a) リンク両端のノードのいずれかをランダムに選ぶ。
 - (b) 選んだノードと同じモジュールに属する、他のノードを選ぶ。但し、ノードを選ぶ確率はノードの次数に比例する。
 - (c) リンクを切り、選んだ2つのノード間に新しいリンクを作る。
3. もしこの過程で、他のノードが孤立されれば、そのノードは削除する。

結果として、リワイヤされた SF ネットワークに対して、 Q が高いほど、次数と媒介中心性に基づいた攻撃に対する頑健性 R が低くなることを示した。しかしながら、次数分布は頑健性に重要な影響を及ぼすにもかかわらず、リワイヤ法はネットワークの次数分布を大きく変化させる。ただでさえ SF ネットワークは、脆弱な次数分布を持つため、変化した次数分布が頑健性に与える影響については統制されていない。また、コミュニティ構造を持つネットワークに対してかなり有効な、モジュール間を繋ぐノードを狙う攻撃法を考慮しない。したがって、第5章ではモジュール性 Q と頑健性 R の純粋な関係を示す。

2.5 ソーシャル・キャピタル

ソーシャル・キャピタルという概念は、社会学、政治学、経済学など多くの分野で議論されている。さまざまな研究者により、ソーシャル・キャピタルの明確な定義が定められる中で、一般的に通用される定義として、アメリカの政治学者のロバート・パットナムが提唱した以下のものがある。

人々の協調行動を活発にすることによって、社会の効率性を高めることのできる、信頼、規範、ネットワークといった社会組織の特徴 [30]

この定義によると、ソーシャル・キャピタルの3要素は信頼、規範、ネットワークとなる。信頼があれば、組織は取引などをする際に、相手について事前に調査する必要がなくなり、社会の全体的な効率性が高まる。パットナムの定義では特に、いろんな規範の中で、互酬性を強調する。互酬性とは相互依存的な利益交換であり、これは現在だけではなく、未来を含めて同等な価値を交換することである。したがって、互酬性は現時代に蔓延する利己主義ではなく、相手を配慮する愛他主義に基づく。ネットワークには垂直・水平的なネットワークがあるが、パットナムは水平的なネットワークの役割を強調する。ここ

で垂直的ネットワークは上司と部下のような関係であり、水平的ネットワークはスポーツクラブや協同組合などを意味する。水平的ネットワークが構築されるほど、社会の人々はより積極的・自主的に協力すると考えられる。3つの構成要素の関係について、パットナムは互酬性と水平的ネットワークの構築から、社会的な信頼が生じる可能性を指摘した。このようなソーシャル・キャピタルは様々な社会問題に深くかかわり、健全な社会を実現するために非常に重要となる。

ソーシャル・キャピタルは組織の内外部の結び付きにより、結合型と橋渡し型に分類される。結合型のソーシャル・キャピタルは、高いレベルの同質性を特徴とするグループまたはコミュニティ内のつながりを表すものである。これは家族もしくは、同一な趣味や背景を共有する友達関係を意味する。結合型ではグループ内の社会的な弱者へ惜しみなく助けるが、言い換えると、強い結合型は排他性へ繋がる潜在的な危険性を持つ [31]。また、結合型は経済的な成果に対して、何の影響も及ぼさず、または否定的な影響を与える [32]。

橋渡し型のソーシャル・キャピタルは、分断されているグループまたはコミュニティ間のつながりを表すものである。橋渡し型は、結合型とは違って多数の長所がある。まず橋渡しは社会的境界を越えるため、他分野の人との連結を通じて異なる価値観、信念に対する寛容と受容を高める傾向がある。また、異なる組織が情報、アイデア、イノベーションを共有し交換することを可能にし、異なる価値観を持つ組織間の合意を導出できる。

第5章では、コミュニティ間の連結が稀なら、ネットワークに新しい構造的な弱点が生じることを見せる。そのため、レギュラグラフは1つのコミュニティだとし、コミュニティ間の連結とコミュニティ内の連結の比率を変えながら、攻撃に対する頑健性を測っている。ここで、コミュニティ間の連結は橋渡し型に、コミュニティ内の連結は結合型に対応できる。そうするとこの結果は、ソーシャル・キャピタルにおいて、橋渡し型は高い頑健性と関係あることを意味する。ソーシャル・キャピタルの分析では、橋渡し型の長所を既に強調していたため、第5章での結果では、橋渡し型が持つ高い頑健性について定量的に示している。

2.6 情報拡散の拠点

SF ネットワークにおいて、ネットワーク全体への情報拡散に影響を及ぼしているハブは重要ノードである。しかしながら、ハブだけではなく、ネットワークに存在する他の重要ノードを探すための様々な方法が提案されている。歴史的には、社会的関係とグループ内での個人の位置と影響力を理解しようにして、中心性という概念が社会学におけるネットワーク分析から考えられた [33]。中心性は、どのような特徴がノードの重要性を区分づ

けるかに対する質問に対して、各ノードが持つ数値を提供する。このような数値を通じて最も重要なノードを確認することができる。有名な中心性指標として Google 創立者が開発した PageRank がある [34]。PageRank はネットワーク内のノード (ウェブページ) が他のノードから受ける推薦 (ハイパーリンク) の数などに基づいて各ノードの重要性を計算し、ウェブページの重要性を順位化する。しかしながら、中心性に基づくと、ノードが持つ負荷を考慮しない問題がある。高い中心性を持つノードは、情報やトラフィックなどの処理において、高い負荷に耐えなければならないため、もしそのノードが故障した場合、ネットワーク全体の性能に非常に大きな問題が発生する。したがって、私達は頑健な構造のために、重要ノードに対する負荷を均等に分散させることが重要だと考えた。

2.6.1 中心性

ハブは高い次数を持っているため、情報の効率的な伝達と拡散に大事な役割をしている。このように、ネットワーク科学では、中心性という、ノードやリンクがどれだけ重要であるかを測定する概念について議論されてきた。本研究では、ノードの中心性だけを考慮する。次の中心性指標は、ノードの相対的な重要性を表す。

2.6.1.1 次数中心性

1 番簡単な指標として、次数で中心性を評価する。これは直感的にあるノードがいくつの他のノードと連結されているかを分かる。この観点からは、ハブが 1 番重要なノードである。

2.6.1.2 媒介中心性

媒介中心性の概念は 1970 年代に Freeman により紹介された [35]。それは、ネットワークにある全ての最短経路において、頻繁に出現するノードが重要だということである。Freeman は社会ネットワークで、媒介中心性を用いて、個人や組織の影響力を把握した。あるノード v の媒介中心性 $C_B(v)$ を求める式は次のようである。

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}. \quad (2.10)$$

ここで、 σ_{st} はノード s と t 間にある全ての最短経路数、 $\sigma_{st}(v)$ はそれらの最短経路の中で、ノード v を通す最短経路の数である。つまり、高い媒介中心性のノードがなくなると、多くのノードは常に情報伝達などのために使っていた経路を失うことである。但し、式 2.6 だけで媒介中心性はネットワークのサイズに左右される。ゆえに、媒介中心性は常

に最大値で正規化される。媒介中心性 $C_B(v)$ が最大なら、 $\frac{\sigma_{st}(v)}{\sigma_{st}}$ は 1 であるので、最大値はノード v を含まないノードのペアの数 $(N-1)(N-2)/2$ である。したがって、媒介中心性は次のように求められる。

$$C'_B(v) = \frac{2 \times C_B(v)}{(N-1)(N-2)} \quad (2.11)$$

2.6.1.3 近接中心性

媒介中心性では最短経路に出現する頻度を考慮したが、近接中心性では最短経路の長さを考えた。つまり、あるノードが高い近接中心性を持つこと、距離的に中心に位置することを意味する。この概念は 1950 年に Bavelas により、最短経路長さの逆数として定義された [33]。

$$C_C(v) = \frac{N-1}{\sum_u d(u,v)}. \quad (2.12)$$

$d(u,v)$ はノード u と v 間の最短経路の長さである。それにノード v が届けるノード数 $N-1$ をかけることで、他のネットワーク上での近接中心性も比較ができる。

2.6.2 支配集合問題

例え情報拡散の拠点として、あらゆる中心性が高いノードを複数選ぶ場合、各ノードの拡散領域が重なり、拡散領域のサイズも不均等な問題がある。したがって、本研究では、より均等な拡散領域サイズを得るために、支配集合問題の解法を考慮する。支配集合の定義は、ノード集合 V の部分集合 $S \subset V$ で、任意のノード $u \in V \setminus S$ が $s \in S$ の隣接ノードである集合である。 s が情報拡散の拠点なら、 u は拠点によりカバーされ、1 ホップ (リンク 1 つ) の最短経路が保証される。しかしながら、実ネットワークでは常に資源が限られているため、拠点の数はでき k だけ最小数で設置することが良いと共に、拠点ではないノードはできるだけ拠点と近くにあるので望ましい。したがって、支配集合を拡張した、 d ホップ支配集合を考える。その定義はノード集合 V の部分集合 $S \subset V$ で、任意のノード $u \in V \setminus S$ が $s \in S$ の d -ホップ隣接である集合である。但し、最小集合の d ホップ支配集合を探すことは NP 困難であるため、多くの近似法が提案された。例えば、 d ホップ内にあるノード数が一番多いノードを拠点として選ぶ方法がある [36, 37]。

2.6.3 頂点被覆問題

近年、情報拡散の拠点選定において、拡散領域の重なりを回避するために、頂点被覆問題を解く近似法 [38] とマルチホップカバーを融合した方法が提案された [39]。頂点被覆問題の定義は、グラフにおいてあるノードの部分集合が頂点被覆集合なら、全てのリンクは、その部分集合にあるノードによってカバーされている状態である。つまり、全てのリンクは、選ばれたノード集合に属する任意のノードと必ず繋がっている必要がある。Cavity 法に基づいた近似法 [38] では、各ノードはカバー、ノンカバー、カバーあるいはノンカバーのいずれかのジョーカー、この3つの状態のいずれかの1つを持つと設定し、その自由エネルギー最小化に基づいて支配集合を探す。ジョーカー状態を持つノードは、支配集合の探索中で自由にカバーやノンカバーになれるので、より幅広い探索ができる。これをマルチホップと融合した方法の詳細なアルゴリズムは、式とともに第4章で説明する。結果的に、この方法 [39] で得られた集合のサイズは、従来のコンピュータ科学からの2近似法で求めた集合のサイズの半分である。さらに、提案された方法で選択されたノードは、重要ノードを探すための従来の方法で選択されたノードより強い拡散力を持っていることが報告されている。

第3章

頑健なネットワーク構造へ再構築する自己修復法と、その分散アルゴリズム

3.1 構造再構築の必要性

多くの社会的、技術的、生物学的ネットワークは、一般的に次数分布がべき乗則に従うスケールフリー（SF）ネットワークで表されることが明らかになっている [1]。しかしながら、SF ネットワークは、高い次数を持つハブノードに対する標的攻撃に対して極めて脆弱である [2,3]。航空網、インターネット、電力網などのインフラネットワークでは、その脆弱な構造を持つため自然災害や人災に脅かされている。例えば、イギリスの空港における未知の攻撃者によるドローン事故が、ヨーロッパで年間1千に達する運航を妨げ [4]、悪天候や自然現象による多数の空港閉鎖が起こった [5]。通信システムのバックボーンであるインターネットも、DDoS 攻撃によって絶えず脅かされている [6]。北米での停電の主な原因は自然災害である [7]。さらに、2011年に日本で発生した大地震と大津波によって、グローバルな供給網が損傷した [8]。一方、こうしたインフラネットワークは持続可能な社会にとって必要不可欠であるため、連結性を維持するためには構造的な弱点を克服する必要がある。そこで、レジリエンスに基づいたシステム設計は、特にシステム工学や環境科学から注目を集めている。レジリエンスの概念には、障害から元の状態に回復するだけでなく、適応能力を持ってシステムを再構築することも含まれる [21,22]。レジリエンスの立場から、結合の頑健性を向上させるためには、被害を受けたネットワークを修復によって再構築することが求められる。なぜなら、もし被害を受けたネットワークが元の

構造に回復しても、極めて脆弱な SF 構造が依然として残るからである。

3.2 従来の自己修復法の改良

最先端の研究では、悪意のある攻撃に対する玉ねぎ状構造が最適な頑健性を持つことが確認されている [10, 11]。次数分布を保存しながら、次数相関を増加させるようにリワイヤリングすることで玉ねぎ状構造が形成される [9, 11]。しかしながら、正の次数相関を増加させることよりもループを強化することがより効果的であることが示されている [14–17]。ネットワークを分断する Dismantling 問題と、ネットワークからループを消す Decycling 問題が、ネットワークサイズが無限大に発散する場合において漸近的に等価である事実から、このループ強化の有効性は支持される [16]。なぜなら、ネットワークがループのない木になると、任意のノードの除去によって容易に断片化することから、ループの重要性は直感的に理解できる。ここで、Dismantling 問題は、グラフを与えられたサイズ以下に断片化させるノードの最小セットを見つける問題であり、Decycling 問題は、グラフを非循環的にするノードの最小セットを見つける問題です。サイクルを除去するために必要なノード集合は、コンピュータ科学では Feedback Vertex Set (FVS) とも呼ばれているが、FVS の見つけ方は組み合わせ最適化問題の NP 困難に属する。ループ強化に着目して、玉ねぎ状構造を生成するためには、成長するネットワークにおいて、複製 [14] や仲介による方法 [15] が効果的である。さらに、ループを強化するリワイヤリングは、次数分布を保存するか否かに関わらず、玉ねぎ状構造だけではなく、その反論的な負の次数相関を持つ頑健なネットワークも生成する [17]。したがって、ループを強化することで、ネットワークは木構造にならず、より頑健になる。言い換えれば、ループ強化は FVS のサイズを最大化することに対応する。

一方、ネットワーク再構築のため、ループ強化に基づいた新しい自己修復法が提案された [23]。この修復法は二つのメカニズムを用いている。まず、より頑健な連結性を得るために、新しい修復リンクを追加できる範囲を段々拡張して、広範囲にループ構造を生成する。この修復範囲の拡張は相互依存する 2 層正方格子に対する研究でのされたものである [40]。そして、ノードが FVS に属する確率を求める BP アルゴリズム [18] を用いて、FVS に属する確率が低いノード同士を修復リンクで繋げる。そうすることで、ループ構成に必要な不可欠なノード数を増やせる。実際、この自己修復法によって、再構築されたネットワークは、従来のヒューリスティック手法によるものよりも高い頑健性を持つことが示された。まず、簡単で局所的な修復法 [24] では、より被害を受けたノードを優先して修復リンクが追加する。そして、Forgiving-Tree アルゴリズム [25] では、コンピュータ

科学において、削除されたノードの最近隣ノード間の連結が木構造として置き換えることで、再構築されたネットワークの直径が減少するが、このアルゴリズムは木構造を生成するので、頑健性を考慮しなかった問題がある。

[23] の利点にもかかわらず、リワイヤされたネットワークがより狭い次数分布の幅を持つことで、より高い頑健性を持つ事実 [17] に注目することで、修復法はさらに改良できる。攻撃後の修復で幅を狭くするためには、低い次数を持つノードに修復リンクを追加することが効果的である。なぜなら、一般的に攻撃で高い次数のノードが取り除かれる傾向があるからである。これが後述するように低い次数を持つノード同士に新しいリンクを生成する理由である。

本章では、適応型無線通信 [41] や将来のマイクロ波電力伝送 [42] のような、インフラネットワークが修復対象だと想定する。デバイス技術の進歩に依存するため詳細な実現法には深入りしないが、攻撃や災害後の被害を受けたシステムで、再構築のためにシステムの構成要素が自ら判断し、お互い協力して作業をする、自律的なネットワークシステムための分散アルゴリズムを提案する。そのようなインフラは大規模で複雑であるため、中央制御よりも分散制御が望ましい。なぜなら、分散制御ネットワークシステムは、中央制御とは対照的に以下のようないくつかの利点を提供するからである [43–45]。

1. システムのサイズやトポロジー構造に関係なく、各ノードは同じアルゴリズムを持って動かすので、分散システムは汎用性を持つ。
2. 異なる所にあるノードが、同時に自分のタスクを処理するため、処理時間は中央制御による順次処理よりも短い。
3. 分散制御は、攻撃や災害後で生き残ったノードが協力するので、全体がシャットダウンすることを防止し、高い信頼性を持つが、中央制御は制御センターが機能不全になることだけで致命的な被害を受ける。
4. 分散アルゴリズムでは、ネットワークの迅速な再構築のために修復リンクをローカルに制御することができる。

上記のように、汎用性、同時性、信頼性、そしてローカルな資源制御の利点を備えた分散型修復アルゴリズムを記述した。ここで対象とするインフラネットワークは、航空網、インターネット、電力網だどとする。

3.3 改良修復法の提案

3.3.1 改良修復法の概要

通常、自然災害や人為的な災害によって攻撃されたノードはすぐには回復せず、長期にわたって機能不全が続き、隣接するノードはリンクの喪失により持続的に被害を受けていると考えられる。このような緊急事態で、改良した修復方法は連結性を以下のように保持できる。

まず、一般性を失わずに、同時に攻撃された qN 個のノードが持っていたリンクの一部を修復のために再利用できると仮定する。ここで q は同時に攻撃されたノードの割合を示し、 N は攻撃前の元のネットワークサイズである。この仮定は実現可能である。例えば、通信ネットワークでリンクに相当するケーブルや無線ビームが切れても一部は再利用可能であり、航空ネットワークでリンクに相当する飛行経路は方向を自由に変更可能である。一方、再利用可能なリンクの数は、ネットワークが受けた被害レベルに依存すると考えられる。再利用可能なリンクが足りない場合、投資などにより、使えるリンクを追加させることもあり得る。そこで、修復ためのリンク総数 M_h を以下のように定義する。

$$M_h \stackrel{\text{def}}{=} r_h \times \sum_{i \in D_q} k_i \quad (3.1)$$

ここで、 k_i はノード i の次数、 D_q は攻撃されたノードの集合である。 \sum は攻撃されたノードの次数和だが、そうすると攻撃ノード間にあるリンクは重複に数えられるので、それ排除した計算することを指す。 r_h は被害状況に応じて $0 < r_h \leq 1$ の範囲を持つ制御パラメータとする。 M_h 個の修復リンクを、被害を受けたノード間に追加することを繰り返す。改良修復法はこの各ステップの概要説明と、改良の理由を以下に述べる。

ステップ1 攻撃で故障したノードの最近接ノードは、故障ノードへのリンクが切断されることから、自分が受けた被害を検知し、最近接ノード同士に連結しようとする。検知後、被害を受けたノードは物資やパケットの転送のための**通常モード**を停止し、**修復モード**を開始する。修復モードで、被害を受けたノードは自分から3ホップ以内の他のノードへ制御信号を送信するが、これらの送信手段（例えば、無線デバイスなどの制御信号用）は光ファイバー通信や飛行輸送のような通常的手段とは異なるものとする。先の修復法とは違って、初期設定として、各ノードは自分から3ホップ以内のノードの識別子（ID）を保存

するためのデータ、いわゆるローカルマップを持っていると仮定する。これは修復法を分散アルゴリズムで記述するためである。修復をする前に、修復の候補者集合はそのローカルマップと同じである。これらの候補は、より大きな連結性を維持するために、初期の集合から段々拡張される。

ステップ2 被害を受けたノードを繋ぐようなリングを形成する。もし修復リンクがリングを完全に形成するのに十分でない場合、限られた M_h リンクでできるだけ最大の連結性を得るよう、繋ぐノードの順序を決める。この順序は、先の修復法 [23] ではランダム順である。

ステップ3 リングを形成した後も、修復リンクが余る場合、リング上にリンクを追加してリングを強化する。その際、重複連結や自己ループを禁止する。先の修復法 [23] では、残余の修復リンクが、FVS に属する確率が低いノード間に追加された。しかしながら、改良修復方法で修復リンクは、次数が低いノード間に追加される。

3.3.2 分散処理を考慮した改良

初期設定前に、攻撃されたノードの隣にある被害ノードは通常モードでローカルマップを既に持っている。ローカルマップは3ホップ以内のノードIDを記憶する。3ホップ以内である理由は以下の通りである。図3.1は攻撃後の被害ノード（青い円）間の追加できる修復リンクを示す。図3.1Bでは、ローカルマップの範囲が2ホップまでの場合、ノード1は修復リンクの候補としてノード4のみを見つける。ノード2も1つのノードのみを見つける。そのため、ノード1と2間に修復リンクを追加しても、ネットワークはまだ断片化されている。しかしながら、図3.1Cのように、範囲が3ホップの場合、各ノードはより多くの候補を見つけられる。しかも、ローカルマップは段々拡張できる。例えば、ノード1は初めに3ホップ以内の候補のセットを $CN_1 = 2, 4, 5$ として持っている。ノード2が CN_1 に含まれているため、 $CN_2 = 1, 3, 4, 6$ は $CN_1 \leftarrow CN_1 \cup CN_2$ として併合される。さらに、ノード3が CN_2 に含まれているため、 CN_3 も $CN_1 \leftarrow CN_1 \cup CN_2 \cup CN_3$ として併合される。最後に、ノード1の CN_1 は $1, 2, 3, 4, 5, 6, 7$ として拡張される。他のノード2,3,4,5,6, および7についても、 CN を持つ。したがって、制御信号を通じて徐々に修復の候補となるローカルマップを拡張することは、局所分散的な修復法において合理的である。

ローカルマップを拡張した後、修復リンク（図3.2の緑色の線）を追加して被害ノー

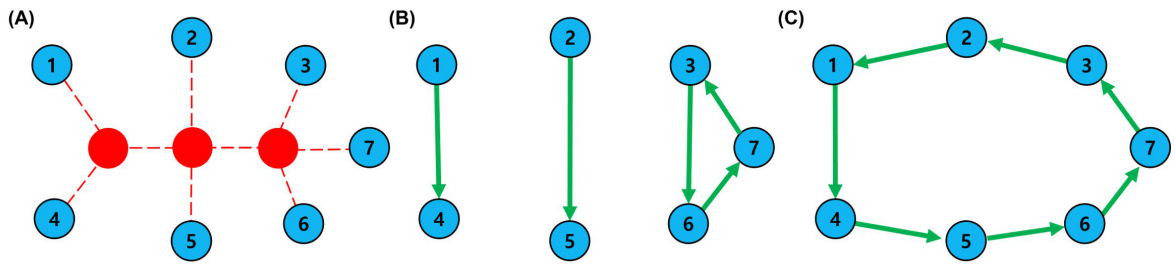


図 3.1: 通信範囲として3ホップが必要な理由の説明。(A) 攻撃直後のネットワーク。修復リンクは、被害を受けたノード間に追加される。(B) 2ホップだけ通信できる場合、(C) 3ホップまで通信できる場合。赤色ノードは故障ノード、青色ノードは故障ノードへのリンク (赤色破線) が切れたせいで被害を受けたノードである。緑色線は修復リンクを意味する。

ド間にリングが形成される。ある同じリング上で連結されたノードは、拡張された同じローカルマップを持つ。但し、ネットワークにいくつかのリングが形成される。先の修復法 [23] では、ランダム順にリングを形成した代わりに、本研究では、より大きな連結性を維持するために、属している連結成分のサイズにしたがってリングを形成する。同じサイズの連結成分に属するノードが多数ある場合、ランダムに選んでリングを形成する。その詳細なで説明する。

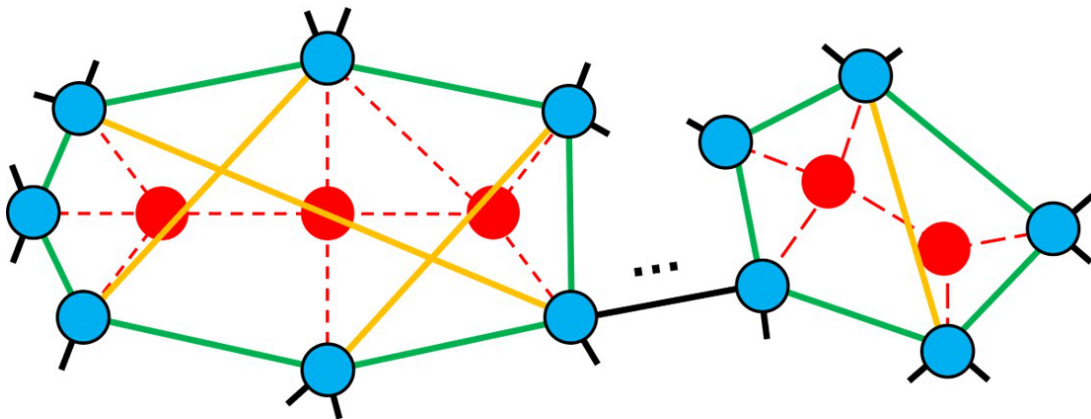


図 3.2: 攻撃を受けたネットワーク上でのリング形成と、ループ強化の概要図。赤色ノードは故障ノード、青色ノードは故障ノードへのリンク (赤色破線) が切れたせいで被害を受けたノードである。緑色と黄色線は、それぞれリング形成とループ強化のために生成される修復リンクである。

また、各リング上で最も次数が低いノードがループ強化のために選ばれる（図 3.2 の黄色の線）。最も次数が低いノード間を連結するというアイデアは、最近の研究 [19, 46] から着想を得た。このループ強化は、リング形成後の残余修復リンクの数まで繰り返される。一方、先の修復法 [23] では、ループ強化に基づいて、ループ形成ために必要不可欠なノードであるフィードバック頂点セット (FVS) を推定する信念伝搬 (BP) アルゴリズム [18] を用いた。3.3 節で、改良修復法が先の修復法 [23] より優れていることを示す。

分散プロセスために、通常の計算における順次プロセスのためのグローバルな資源制御ではなく、ローカルな資源制御を考慮する。図 3.2 のように制御パラメータ r_h が 1 である場合、左右それぞれのサブグラフにおける修復リンク（赤点線）の数は、ローカル制御によると、9 と 5 である。これらの数値は、式 (1) から、グローバルな D_q の代わりに、サブグラフ内の攻撃されたノード集合である $i \in D_q^{local}$ を使用するように修正して得られる。また、図 3.2 のように、左右のサブグラフのそれぞれでリングを形成するために、独立して 9 つまたは 5 つの修復リンクが追加される。リング形成後、左のサブグラフには 2 つのリンクが残る。しかしながら、右のサブグラフでは、5 個のノード間にリングを形成するために 5 本のリンクが追加されるため、残余リンクはない。この時、2 つの残余リンクは左サブグラフ内のみに使われる。

一方、グローバル制御においては、全体のグラフで $9 + 5 = 14$ の修復リンクが使える。したがって、右のサブグラフに 1 つまたは 2 つの残余リンクが追加される可能性がある。ローカル資源制御による修復が、グローバル資源制御による修復とほぼ同じ頑健性と効率性が得られることを 3.3 節で示す。

3.4 先の修復法と改良修復法の修復結果比較

3.4.1 分析指標とサンプルネットワーク

上記の手順で修復したネットワークに対して、先の修復法 [23] と改良修復法を比較する。悪意のある攻撃後、それぞれの修復法で再構築された典型的なインフラストラクチャネットワーク [47–49] に対して、以下の 3 つの評価指標を測る。ここで、悪意のある攻撃として、ネットワークから次数が 1 番高いノードを壊した後、次数を再計算して結果的に qN のノードを削除する攻撃 (HDA) を考える。また、地震や洪水のような災害からノード集合が一気に消されることを反映するために、局所的攻撃 (Localized Attack, LA) も考える。LA は、一様ランダムに選ばれたノードとその近接ノードと一緒にネットワークから削除する。

1. 残余内の最大連結成分のサイズの比率 (LCC 比率) $S^{1st}(q)/(1-q)N$ は、 qN ノードへの攻撃後に再構築されたネットワークの最大連結成分に属するノードの数を表す。
2. 残余内の頑健性指標 [11] $R(q) \stackrel{\text{def}}{=} \frac{1}{N} \sum_{q'=\frac{1}{N}}^1 \circ \frac{S(q')}{(1-q')N}$ は qN ノードへの攻撃後に再構築されたネットワークから、さらに $q'N$ ノードを削除した場合の LCC サイズで、 \sum は $q' = \frac{1}{N}, \frac{2}{N}, \dots, \frac{N-1}{N}, \frac{N}{N} = 1$ に対する和を表す。
3. 残余内の経路の効率 $E(q) \stackrel{\text{def}}{=} \frac{1}{N(N-1)} \sum_{i,j} \frac{1}{d_{ij}}$ は、ノード i と j 間のホップ数を数えた最短経路の長さを表す。

これらの指標の範囲は、 $0 < S(q)/(1-q)N \leq 1$ 、 $\frac{1}{N} < R \leq 0.5$ 、および $0 < E \leq 1$ である。表 3.1 は、無向リングで繋がっているグラフとして抽出された、元のネットワークの基本的な特性を示す。

ネットワーク	N	M	k_{avg}	k_{max}	k_{min}	D	r
Internet AS Oregon	6474	12572	3.9	1458	1	9	-0.181
Airline OpenFlight	2905	15645	10.8	242	1	14	0.048
PowerGrid	4941	6594	2.7	19	1	46	0.0034

表 3.1: 攻撃以前、インフラネットワークの基本性質。 N と M は、それぞれネットワークのノード数とリンク数である。 k_{avg} 、 k_{max} 、 k_{min} は、平均、最小、最大次数である。 D はネットワークの直径で、 r は次数相関を示す。

改良修復法と先の修復法 [23] により再構築されたネットワークに対して、上記の 3 つの評価指標を 100 回測り平均値を出す。攻撃を受けても連結性を維持できるネットワークを再構築するためには、高い LCC 比率を得ることが重要である。

3.4.2 改良修復法による頑健性、効率性の向上

図 3.3A は、両修復法により OpenFlight と AS Oregon に対して比較的高い LCC 比率を示す (各色線に対して上向き三角と下向き三角のマーク)。特に、LCC 比率は $r_h \geq 0.5$ の場合に最高値を示す。OpenFlight と AS Oregon に対する紫と黄色の線が重なっている。さらに、高い頑健性と任意のノード間が短い経路を持つことも、修復したネットワークがさらなる攻撃に対する耐久性を持ち、効果的な通信や輸送のために重要である。図

3.3B と 3.3C は、改良修復法が OpenFlight と AS Oregon に対して先の修復法よりも高い $R(q)$ と $E(q)$ の値を持っていることを示す ($r_h \geq 0.5$ の場合に上向き三角でマークされた黄色と紫色線)。さらに、 $R(q)$ の値は元のネットワークの値 (黒点線) よりも高い。

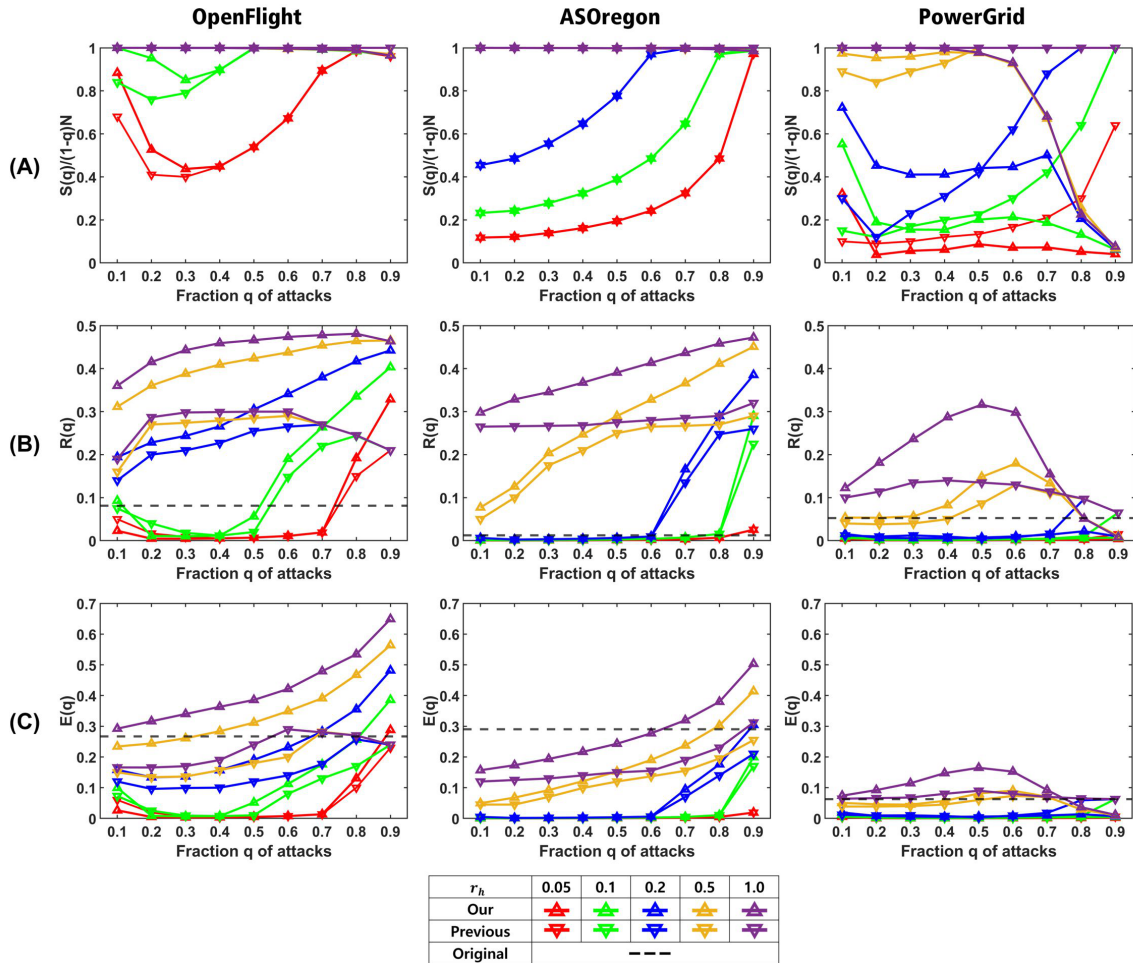


図 3.3: OpenFlight、AS Oregon、PowerGrid 上での改良修復法 (上向き三角) と既存法 (下向き三角) の比較。各 r_h において、修復したネットワークの (A) 最大連結成分サイズ比率 $S(q)/(1-q)N$, (B) 頑健性指標 $R(q)$ 、(C) 経路の効率 $E(q)$ 。赤、緑、青、黄、紫色線は、それぞれ $r_h = 0.05, 0.1, 0.2, 0.5, 1.0$ に対応する結果である。黒破線は攻撃以前のネットワークが持つ指標である。

また、図 3.3A では、リング形成のためのノード連結順に関係なく、両修復法が類似し

た比率を示す。しかしながら、図 3.3B および 3.3C のように、ループ強化するために最小次数のノードを選択することは、先の修復法 [23] で BP を適用したことより、 $R(q)$ と $E(q)$ の両方の値が増加した。これらの結果は、頑健性のためには、FVS を増やせることより、次数が低いノード同士を繋ぐことがより有効であることを意味する。例外として、表 3.1 の大きな直径を持つ PowerGrid に対しては、わずかに異なる結果が部分的に得られている。PowerGrid に関しては、図 3.3A の右側で、 $q = 0.5$ での LCC 比率が改良修復法（上向き三角でマークされた黄色と紫色線）で急激に減少する。さらに、図 3.3B および 3.3C の右側で示されるように、 $q \geq 0.7$ での改良修復法による $R(q)$ と $E(q)$ の値（上向き三角でマークされた）は、先の修復法によるもの（黄色と紫の線に下向き三角でマークされた）よりも低くなっている。大きな直径のネットワークで、効果が低い理由についてはまだ不明である。

図 3.4 は HDA と LA を同時に考慮する際の LCC 比率を示す。図 3.4A、B は HDA で攻撃された OpenFlight を再構築し、再構築したネットワークにおいてそれぞれ HDA と LA に対する LCC 比率を指す。また、図 3.4C、D は LA で攻撃された OpenFlight を再構築し、再構築したネットワークにおいてそれぞれ HDA と LA に対する LCC 比率を指す。図 3.4A と C を見ると、提案法で再構築されたネットワークは、一度被害を受けた経験がある攻撃に対しては高い頑健性を持つ。なおかつ、図 3.4B と D のように、それらのネットワークは、経験しなかった攻撃に対しても、高い頑健性を持つ。特に、 $r_h \leq 0.2$ の低い再利用率でも、LA に対する高い頑健性を得られる（赤、緑、青色線）。しかしながら、LA により大きく破壊された場合 ($q \geq 0.8$)、HDA に対する頑健性は急激に減少する（図 3.4C）。この現象の理由は次のように考えられる。最初に各ノードは自分から 3 ホップでノードと通信できると仮定した。したがって、ネットワーク内の LA によって巨大な穴が作られる場合、ノードはメッセージを遠くにあるノードへ転送できない。この問題は、提案法における連結性を低下に繋がる。つまり、通信範囲の制限と連結性の維持の間にはトレードオフが存在する。もし実のインフラネットワークが 80% 以上破壊されると、提案法はうまく機能しない可能性がある。但し、被害が大きい状況は稀な現象であり、そのような状況でも、ネットワークを修復するよりは、新しいインフラを構築することが増しである。ゆえに、このようなトレードオフは提案法の限界にはならず、提案法は局地的攻撃に対しても効果的であると結論づける。

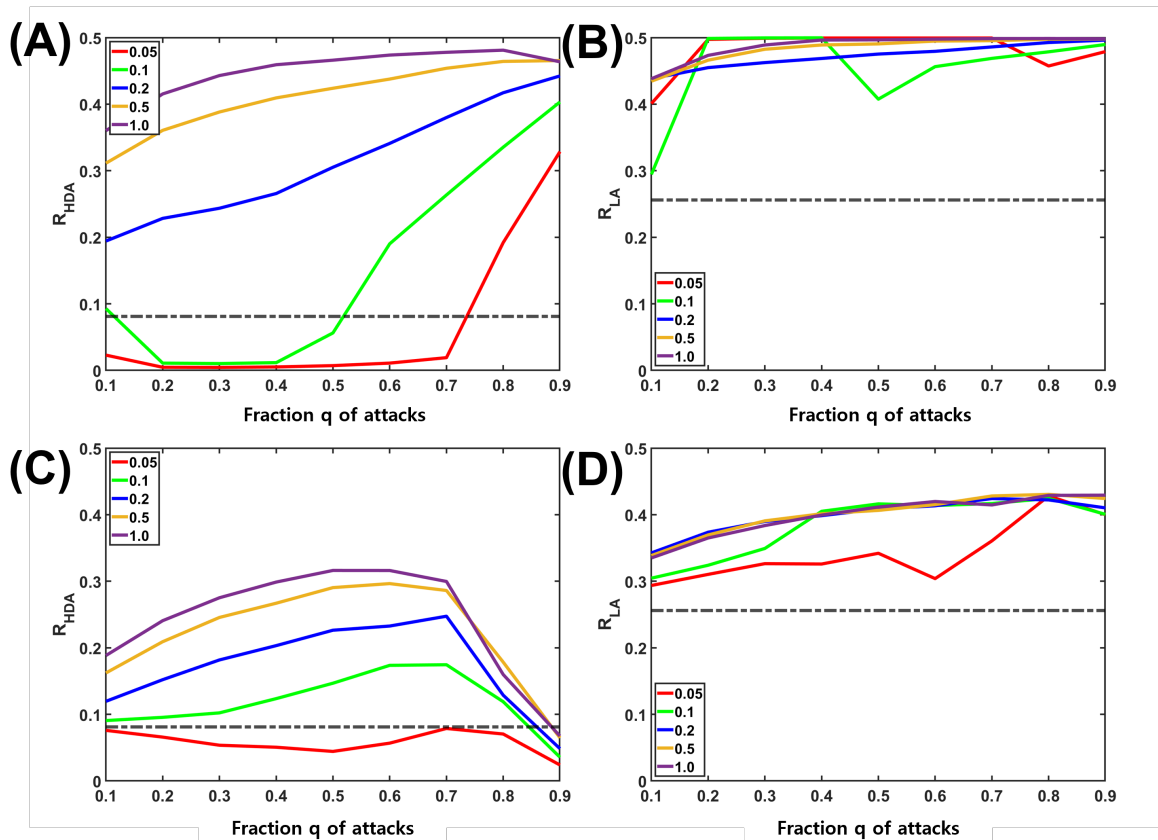


図 3.4: 再構築された OpenFlight が持つ頑健性。(A) HDA で攻撃されたネットワークを再構築し、それをまた HDA で攻撃した時の最大連結成分サイズ比率 $S(q)/(1-q)N$ 、(B) HDA で攻撃されたネットワークを再構築し、それを LA で攻撃した時の最大連結成分サイズ比率 $S(q)/(1-q)N$ 、(C) LA で攻撃されたネットワークを再構築し、それを HDA で攻撃した時の最大連結成分サイズ比率 $S(q)/(1-q)N$ 、(D) LA で攻撃されたネットワークを再構築し、それをまた LA で攻撃した時の最大連結成分サイズ比率 $S(q)/(1-q)N$ 。赤、緑、青、黄、紫色線は、それぞれ $r_h = 0.05, 0.1, 0.2, 0.5, 1.0$ に対応する結果である。黒破線は攻撃以前のネットワークが持つ指標である。

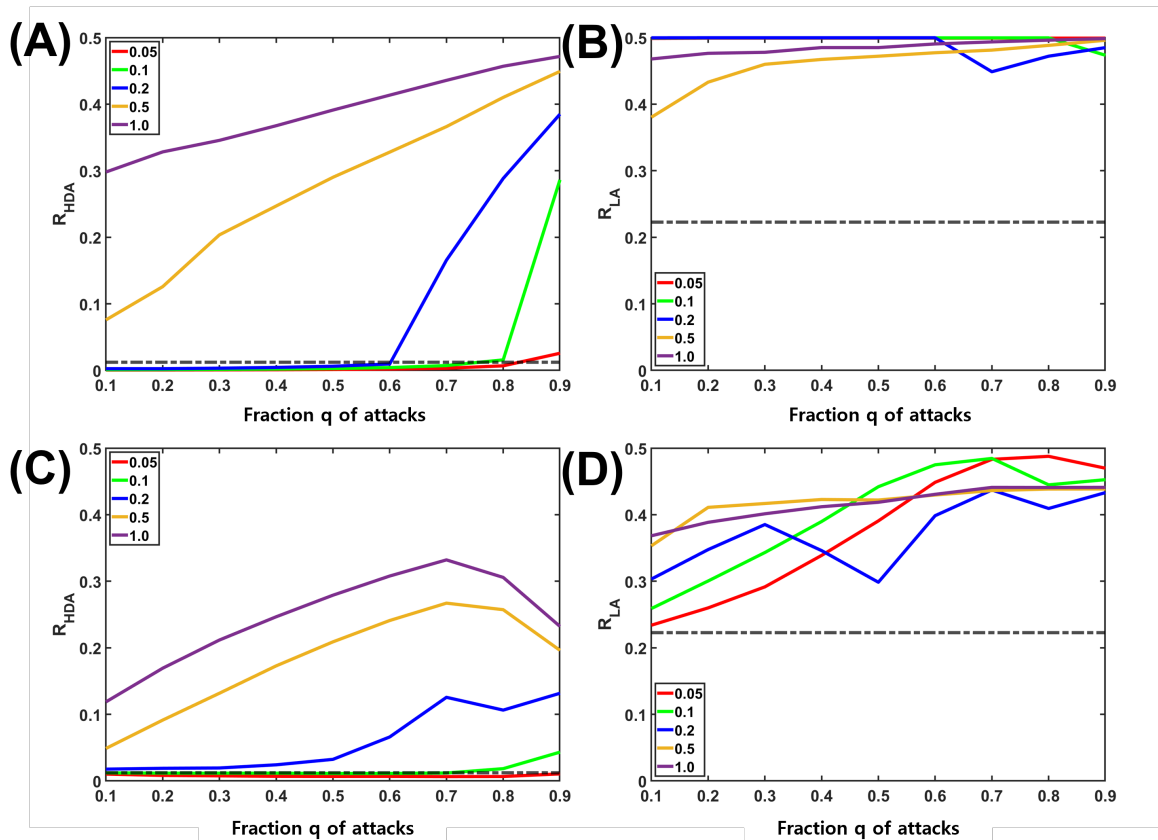


図 3.5: 再構築された AS Oregon が持つ頑健性。(A) HDA で攻撃されたネットワークを再構築し、それをまた HDA で攻撃した時の最大連結成分サイズ比率 $S(q)/(1-q)N$ 、(B) HDA で攻撃されたネットワークを再構築し、それを LA で攻撃した時の最大連結成分サイズ比率 $S(q)/(1-q)N$ 、(C) LA で攻撃されたネットワークを再構築し、それを HDA で攻撃した時の最大連結成分サイズ比率 $S(q)/(1-q)N$ 、(D) LA で攻撃されたネットワークを再構築し、それをまた LA で攻撃した時の最大連結成分サイズ比率 $S(q)/(1-q)N$ 。赤、緑、青、黄、紫色線は、それぞれ $r_h = 0.05, 0.1, 0.2, 0.5, 1.0$ に対応する結果である。黒破線は攻撃以前のネットワークが持つ指標である。

3.4.3 改良修復法による構造改善

次に、修復されたネットワークの構造変化を見るために、隣接ノードのサイズ (図 3.6) と次数分布 (3.7) を、元のネットワークのものと比較する。隣接ノードのサイズは、ローカルマップの最大比率 (Maximum Ratio、MR) として測定する。各ノードのローカルマップには、自分から 3 ホップ以内のノード ID がある。ローカルマップの MR は、ローカルマップに含まれるノード ID の数と、元のネットワークサイズ N の比率として定義する。この値が 1 に近い場合、ノードがほとんど全てのノード ID を持っていることを指す。図 3.6 は、改良修復法により修復されたネットワークの MR を示す。PowerGrid を除いて、改良修復法によって再構築されたネットワークは、OpenFlight の MR が 0.6 未満、AS Oregon の MR が 0.2 未満である。攻撃前の AS Oregon と OpenFlight では、巨大な次数のハブが存在するため、最大 MR は 1 に近い (黒点線)。一方、修復されたネットワークにはハブがないため、MR は小さくなる。しかしながら、PowerGrid の場合 (黄色と紫色線)、再構築されたネットワークは、元のネットワーク (図 3.6 の右側) よりも高い MR を持っている。したがって、MR の大小が意味することは、ネットワークでのハブ有無である。

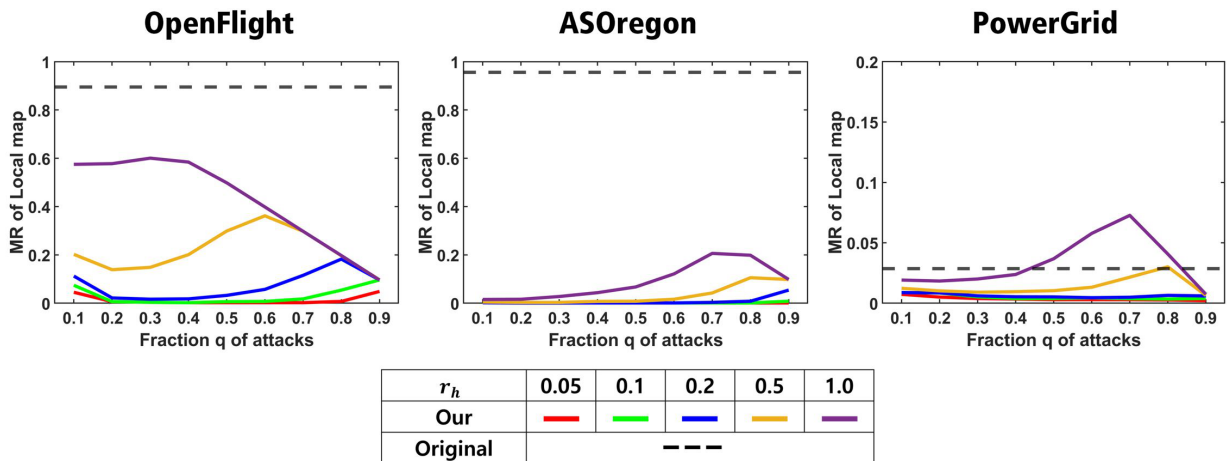


図 3.6: 攻撃以前のネットワークと、修復したネットワーク上で、ローカルマップサイズの比率。赤、緑、青、黄、紫色線は、それぞれ $r_h = 0.05, 0.1, 0.2, 0.5, 1.0$ に対応する結果である。

図 3.7 は、改良修復法により再構築されたネットワークの次数分布を示す。OpenFlight と AS Oregon の場合、再構築された次数分布の幅は狭い。言い換えると、元の SF ネットワークが持つ、べき乗次数分布における最大と最小の次数のギャップよりも、再構築されたネットワークにおけるそれらのギャップが小さくなる。但し、修復リンクが追加されたとしても、再構築されたネットワークが完全グラフにはならない。表 3.2 のように、 $r_h = 1.0$ でも、完全グラフを形成するには修復リンクが足りない。

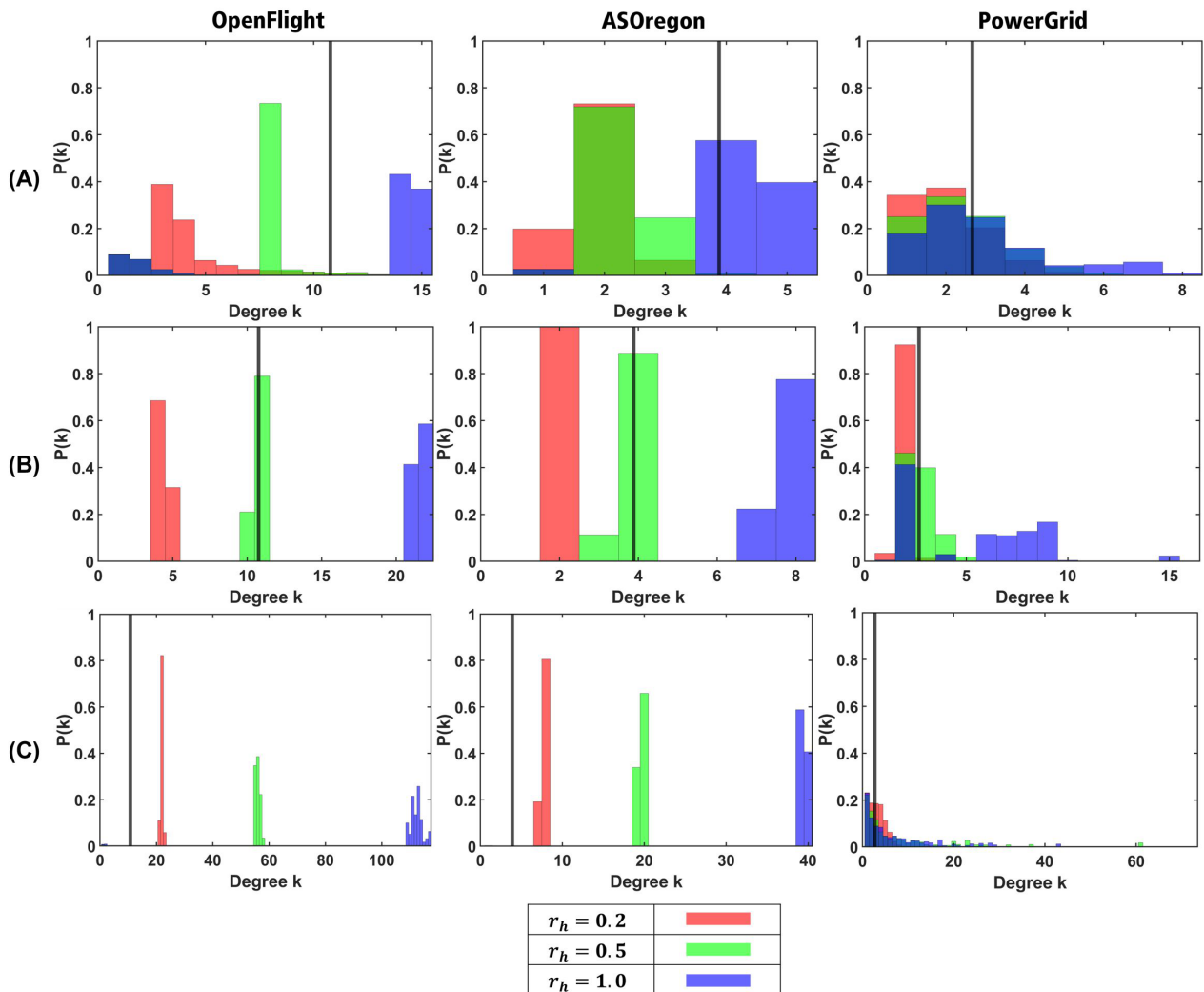


図 3.7: 改良修復法で修復されたネットワークの次数分布。修復は qN 個のノードが壊された後に進められた：(A) $q = 0.1$ 、(B) $q = 0.5$ 、(C) $q = 0.9$ 。赤、緑色バーは、それぞれ $r_h = 0.5$ 、 $r_h = 1.0$ での次数分布である。黒垂直線は、攻撃以前のネットワークの平均次数である (表 1 参照)。

ネットワーク	q	0.1	0.5	0.9
	M			
OpenFlight	M_{Heal}	15645.0	15638.0	15403.0
	M_{Comp}	3415191.0	1049076.0	39135.5
ASOregon	M_{Heal}	12572.0	12562.0	12457.0
	M_{Comp}	16974051.0	5221296.0	202846.2
PowerGrid	M_{Heal}	6594.0	6490.0	296.3
	M_{Comp}	9885681.0	2941525.0	552.7

表 3.2: 最大連結成分にあるリンク数。 M_{Heal} は修復されたネットワーク全体のリンク数である。 M_{Comp} はもし最大連結成分が完全グラフである場合のリンク数である。

先の修復法では、ノード i が元の次数 k_i だけのポートを再利用できると想定した [23]。ノード i に k_i を超える修復リンクが追加された場合、追加のポートが必要となる。ここで、ポートはノードでのチャンネルやプラグソケットに対応する。ポートは、空港の滑走路やプラグソケットのように、リンクに相当する飛行機やケーブルから独立している。この仮定で、少数のノードに修復リンクが集中することを避けるために、追加ポートの制限を持つ自己修復方法が検討されてきた [23]。しかしながら、そのような制限は、改良修復法では範囲外で、その理由は以下の通りである。先の修復法によって再構築されたネットワークの次数分布は指数分布に従い、追加ポートの最大数は、元のネットワークでの最大次数より低い [23]。対照的に、改良修復法により再構築ネットワークの次数分布は図 3.7 のように、狭い幅を持つ。さらに、最大次数は元のネットワークでの数値よりも低い（表 3.1 および 3.3）。

特に、改良修復法による最大次数は OpenFlight、AS Oregon、PowerGrid でそれぞれ 15、5、7 になるが、 $q = 0.1$ の場合、既存の方法では 65、19、4 である [23]。例え、次数 1 のノードが改良修復法で最大次数のノードになったとしても、改良修復法で要る追加ポートの数は非常に少ない。したがって、追加ポートの制限なしでポートの数を節約できるため、そのような制限をあまり考慮せず、修復ための追加ポートは事前に装備すべきだと想定した。

ネットワーク	q	0.1	0.5	0.9
	r_h			
OpenFlight	0.2	13.0	5.0	22.4
	0.5	13.0	11.0	56.5
	1.0	15.0	22.0	112.9
ASOregon	0.2	4.0	2.0	8.0
	0.5	4.0	4.0	20.0
	1.0	5.0	8.0	40.0
PowerGrid	0.2	7.0	4.6	8.0
	0.5	7.7	5.2	33.7
	1.0	8.0	9.2	32.4

表 3.3: qN 個のノードを破壊した後、修復されたネットワークの最大次数。

3.4.4 資源制御の比較

最後に、2つの資源制御を比較する。図 3.8 は、ローカルとグローバル資源制御をそれぞれ + 記号と × 記号でマークして、改良修復法により再構築されたネットワークに対する、3つの評価指標を示す。OpenFlight の図 3.8A のように、ローカル制御は $r_h \leq 0.1$ でグローバル制御よりもわずかに高い比率を持つ (+ 記号でマークされた赤と緑色線)。一方、 $r_h \geq 0.2$ では、ローカル制御とグローバル制御の LCC 比率はほぼ一致している(重なった紫、青、黄色線)。OpenFlight と AS Oregon の図 3.8B と 3.8C を見ると、ローカル制御とグローバル制御の $R(q)$ と $E(q)$ の値も同色線ごと一致している。例外として PowerGrid で、ローカル制御とグローバル制御は、それぞれ $r_h \leq 0.2$ で異なる LCC 比率を持つ(図 3.8A の赤、緑、青色線)。これらの結果から見ると、ローカルおよびグローバル資源制御による再構築がほぼ同じ評価指標を持つことを指すが、ローカル資源制御は 3.5 節で記述する分散アルゴリズムに適した長所がある。

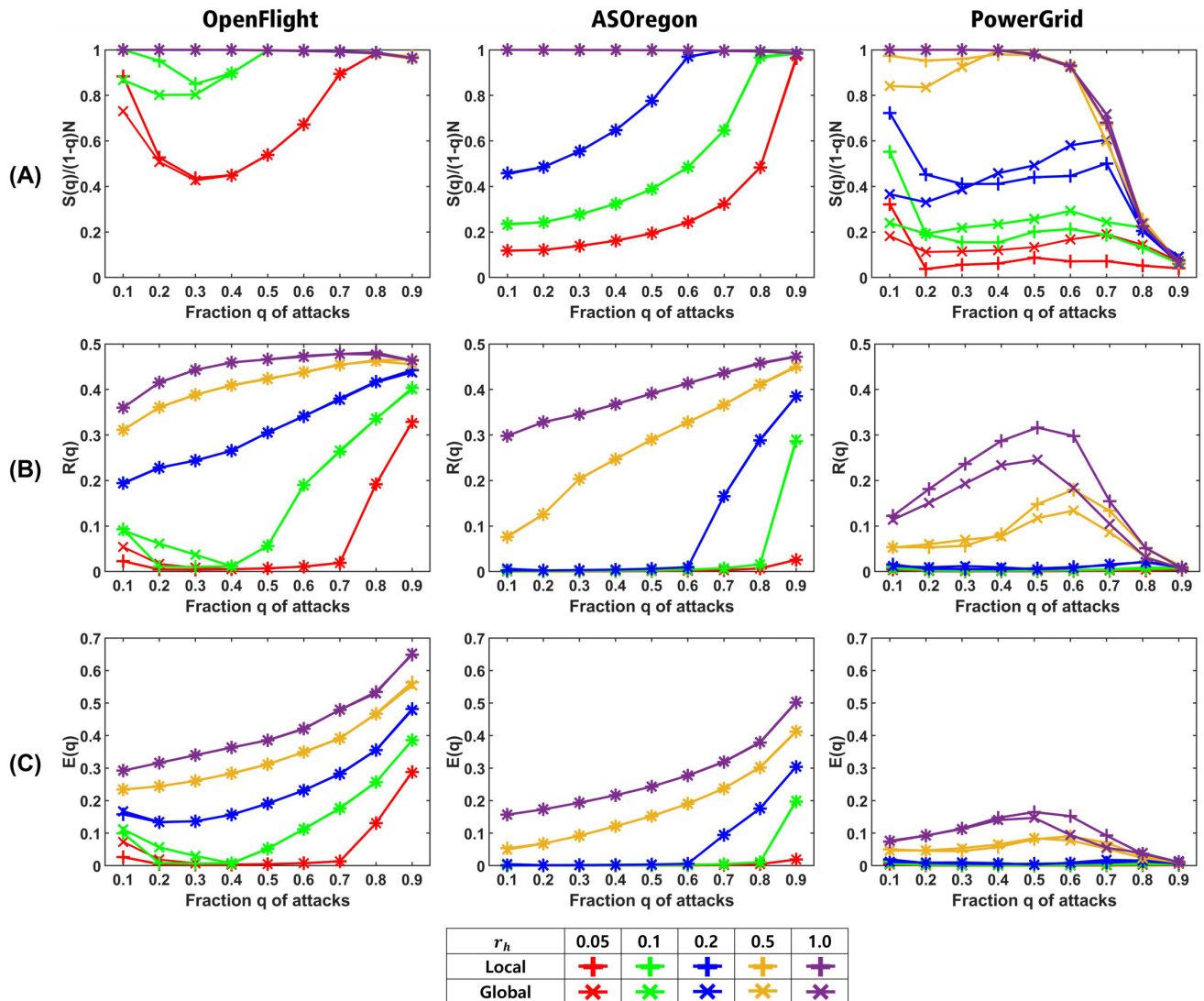


図 3.8: OpenFlight, AS Oregon、PowerGrid 上でのローカル資源制御 (プラス) とグローバル資源制御 (クロス) の結果比較。各 r_h において、修復したネットワークの (A) 最大連結成分サイズ比率 $S(q)/(1-q)N$, (B) 頑健性指標 $R(q)$, (C) 経路の効率 $E(q)$ 。赤、緑、青、黄、紫色線は、それぞれ $r_h = 0.05, 0.1, 0.2, 0.5, 1.0$ に対応する結果である。

3.5 改良修復法の分散アルゴリズム記述

3.5.1 非同期処理のための前提

本節では、一般的な非同期処理 [44, 45] における以下の仮定に基づいた、修復アルゴリズムの分散プロセスを記述した。

1. 各ノードは到着順に格納された FIFO キューにあるメッセージに即座に対応し、到着順が送信ノードの局所的な時計から影響を受けることはない。
2. ノード間で短いメッセージにある制御信号によって通信が行われるため、内部処理にかかる時間は、物理的要因による短い遅延と比べて、無視できる。
3. 内部処理はメッセージ受信直後にすぐ開始される。
4. 各ノードは固有の識別子 (ラベル) を持ち、これはアルゴリズムが実行されても更新や上書きされない。
5. 各ノードは直接的に最も近い隣接ノードの故障のみを検出できる。
6. 各ノードは、分散的な方法で修復ために、同じアルゴリズムを実行する。
7. 提案した修復アルゴリズムが実行されている間に、ネットワークに追加的な故障は発生しないものとする。もし新しい故障が発生した場合、プロセスのロールバックが必要とする。

仮定により、修復アルゴリズムは必ず停止性を持つ。なぜなら、ノードはタイミングは違うが各自受けたメッセージを到着順に即座に処理し、ノードのラベルはアルゴリズムの実行中には上書きされない。また、アルゴリズムの実行中にノードがメッセージを送る対象は、自分と同じ修復範囲に存在するノードである。ゆえに、メッセージの送り先は限定されている。そして、修復処理には反復や後戻りはないから結果的にアルゴリズムは停止性持ち、実際のシミュレーションで停止することを確認している。通常の PC を用いて分散アルゴリズムのエミュレートをするために、Python の AsyncIO モジュールを用いた。その際、ノードが処理すべきタスクは、拡張した修復候補とメッセージを送受信することで構成される。各ノードは AsyncIO を用いて同時にタスクを生成する。AsyncIO モジュールは全てのノードについて、タスクの同時実行をエミュレートできるが、内部的には順序が付けられる。偶然等の物理的要因による短い遅延に対応させて、全てのノードに対するタスク実行順序をランダム化する。但し、タスクを生成しないノードは順序から排除する。図 3.9 は、このようなタスク処理の概略図を示す。水平方向と垂直方向はそれぞれ、修復モードのノードと、修復プロセスのグローバルタイムラインを示す。

3.5.2 分散アルゴリズムの各フェーズ説明

非同期処理において、分散アルゴリズムは5つのフェーズで構成 (3.10) され、各フェーズで使用される変数とメッセージによって区別されている。全ての変数とその初期値は通常モードで既に設定されていると仮定する。以下、各フェーズの概要と、分散アルゴリズム

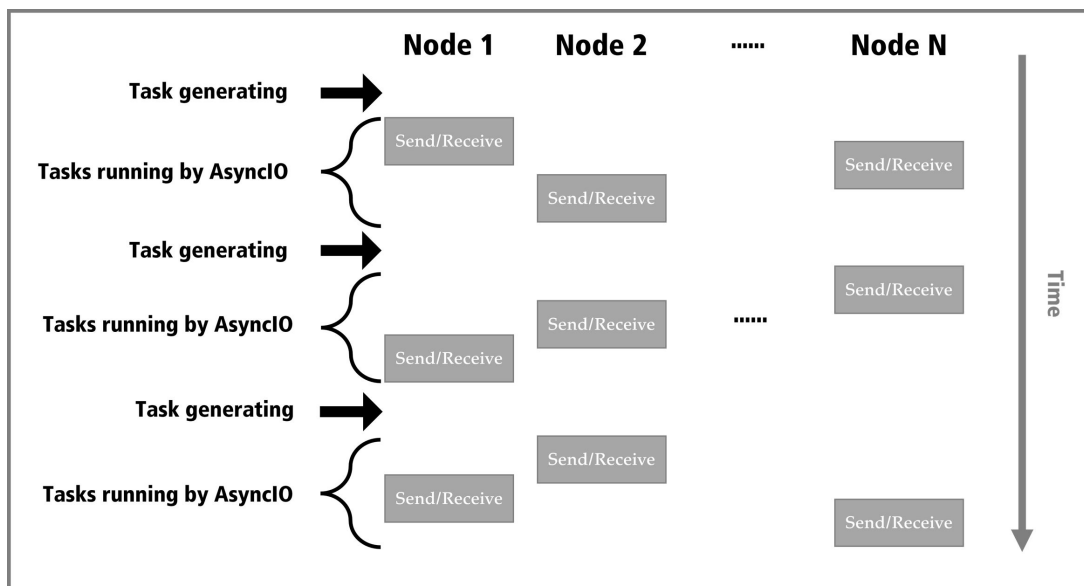


図 3.9: 提案した分散アルゴリズムにおける、同時処理の概念図。

ムとしての記述を示す。

3.5.2.1 アルゴリズム実行ための初期設定

攻撃後、隣接ノードは切断されたリンクの被害を検知し、通常モードから修復モードへ変換する。被害を受けたノードは、修復モードでローカルマップにあるノードと直接メッセージを交換できると仮定した。それぞれの被害を受けたノードは同時に初めてのフェーズを実行して、アルゴリズムの最初のフェーズを開始するためのタスクを生成する。

3.5.2.2 フェーズ 1： 修復候補を拡張する

フェーズ 1 は、改良修復法説明のステップ 1 に相当する。被害を受けた各ノードは修復先の候補ノードを以下のようにして拡大する。開始後、被害を受けたノードはローカルマップにあるノードにまず *Gathering* メッセージを送信する。候補ノードの ID は *Gathering* メッセージ内に入っている。したがって、候補ノードを拡大するために、ノード i は *Gathering* メッセージを受信するたびに自分の *ExtendedCandidates* を更新する (アルゴリズム 1: 2 行)。 *ExtendedCandidates* は候補ノードの ID 集合である。メッセージの交換を通じて、被害を受けたノードはローカルマップの初期集合から候補ノードを拡大する (アルゴリズム 1: 3 行)。但し、 *Gathering* メッセージの交換を不必要に繰り返さないように、フェーズ 1 で生成されるタスクの数を限定して *UpperLimitGathering*

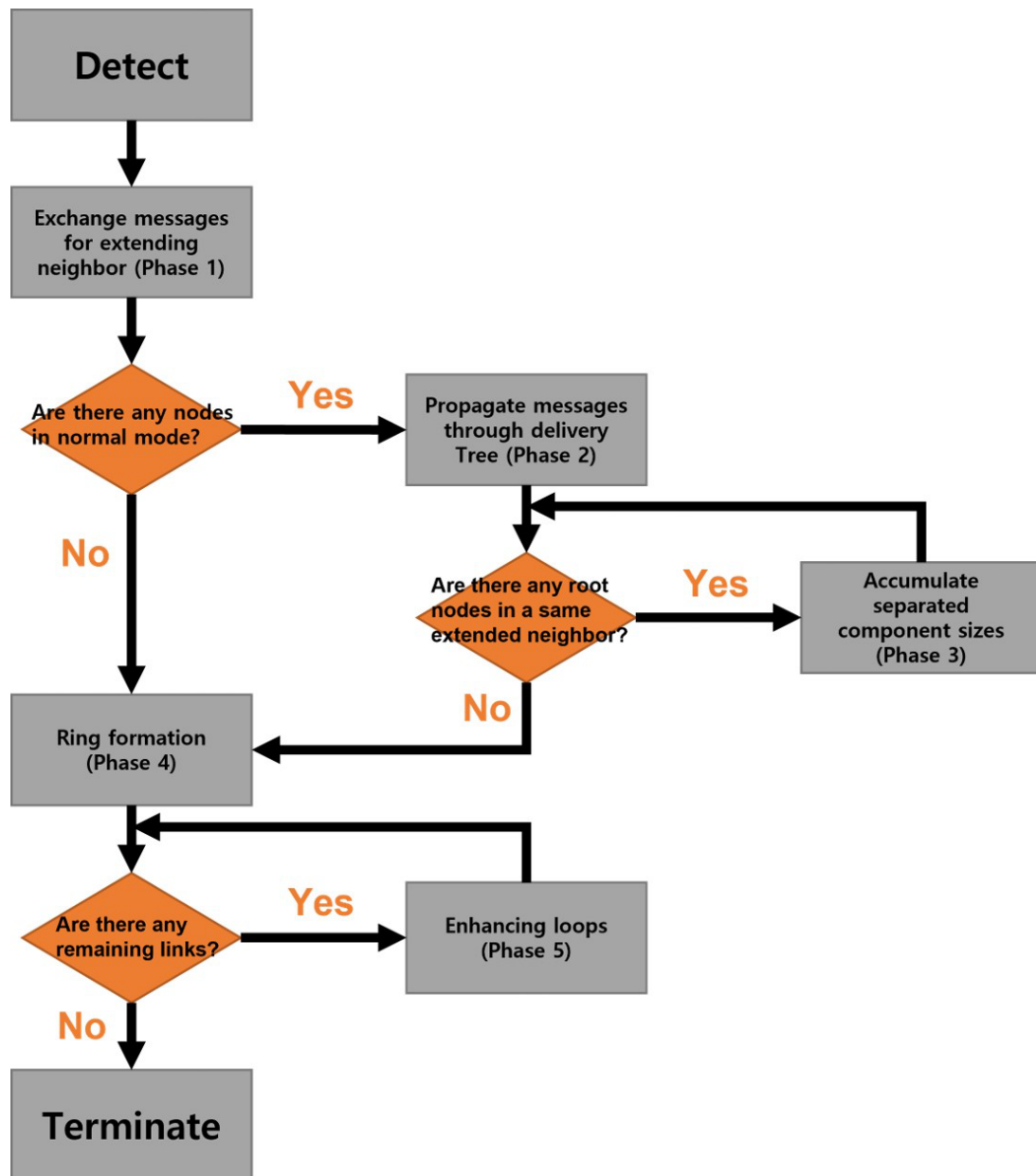


図 3.10: 5つのフェーズで構成される修復プロセスのフローチャート。

を定義する。候補はネットワーク全体を超えて拡大することはできないから、その上限値はネットワーク直径 D に等しい。 D タスクを生成した後、ノードはフェーズ 2 を開始する (アルゴリズム 1: 7 行)。フェーズ 2 を開始する前に、後続のフェーズのために *Parent* と *Leader* の 2 つの変数を補う (アルゴリズム 1: 5-6 行)。被害を受けた各ノードは根ノードになり、*Parent* は自分の ID に設定する。さらに、*ExtendedCandidates*

内で最も低い ID を持つノードをリーダーノードに設定する。

ノード i において、フェーズ 1 で用いる変数

ExtendedCandidates: 修復先の候補ノード集合。初期値は、初期設定で持ったローカルマップと同値である。

NumberOfTask: ノードがフェーズ 1 で生成したタスク数。タスクを生成する際に、変数は 1 ずつ増加する。

UpperLimitGathering: ノードがフェーズ 1 で生成できるタスク数の上限値

Algorithm 1 ノード i において、フェーズ 1 の疑似コード

- 1: ノード i の受信: ノード j からの *Gathering*(*ExtendedCandidates_j*)
 - 2: *ExtendedCandidates_i* \leftarrow *ExtendedCandidates_i* \cup *ExtendedCandidates_j*
 - 3: ノード j へ送信 *Gathering*(*ExtendedCandidates_i*)
 - 4: **if** *NumberOfTask_i* = *UpperLimitGathering_i* **then**
 - 5: *Parent_i* \leftarrow i
 - 6: *ExtendedCandidates_i* から、ID が 1 番低いノードをリーダーノードに設定する。
 - 7: フェーズ 1 を終わらせ、フェーズ 2 を始める。
 - 8: **end if**
-

3.5.2.3 フェーズ2: 配信木におけるメッセージ伝搬

フェーズ2では、お互い違う所にある被害を受けたノード間にリングを形成する。リングの連結順は、ノードが属する連結成分のサイズによって決められる。したがって、フェーズ2では無駄なフラッディングを避けるために配信木上でメッセージを伝搬する。メッセージの到着順に基づいて、ノード間の親子関係が次のように決定される。

図3.11はフェーズ2の概要を示す。*ModeChange*メッセージは、根ノードから葉ノードへの機能障害を通知する。図3.11Aに、根ノード(ノード1)はまず隣ノード(ノード2と3)に*ModeChange*メッセージを送信する。他にも被害を受けた各ノードは根ノードとして機能する。ノード2と3で*ModeChange*メッセージを受信すると、それらのモードは通常から修復に変更される。配信木を構築するために、ノード2と3はノード1を配信木内の親ノードとして定義される(アルゴリズム2:3行)。

図3.11では、ノード1が親になるので、ノード2と3の*NumberOfUnclearParentChild*は1減少する(アルゴリズム2:6行)。*NumberOfUnclearParentChild_i*の値は、親子関係を決定するために残っているノードの数を表す。隣接ノードが親または子として定義されるたびに、*NumberOfUnclearParentChild*は1減少する。次に、ノード2と3は親を除く最近接ノードに*ModeChange*メッセージを送信する(アルゴリズム2:10-12行)。図3.11Aのように、ノード6はノード2と3からほぼ同時に*ModeChange*メッセージを受け取る。

本アルゴリズムで、ノードはメッセージの到着順に応じて親を選択するものとする。この場合、ノード6はノード3を親にして、ノード2へ*Reject*メッセージを送信する(アルゴリズム2:16行)。その後、ノード2はノード6からの*Reject*メッセージを受け取り、ノード6が自分の子ノードではないことを判断できる。したがって、ノード2は自分の*NumberOfUnclearParentChild*を1減少させる(アルゴリズム2:19行)。この伝搬を通じて、配信木が構築される。

各葉ノードが親から*ModeChange*メッセージを受け取ると、自分のサイズを表す変数*TreeSize*^{def}1の*Back*メッセージを親に送る(アルゴリズム2:8行)。図3.11Bで示すように、*ModeChange*メッセージはノード4、5、6、7を経由して葉ノード(ノード8、9、10、11)に到着する。ノード4、5、6、7は子ノードからもらった*TreeSize*を蓄積し、合計サイズの*Back*メッセージを親に送る(アルゴリズム2:27行)。これらのプロセスを繰り返すことで、根ノードは自分が属する連結成分のサイズが分かる。

図3.12では、同じ連結成分内に複数の根ノードが存在することがある場合を示す。図3.12Aでは、ノード4と5が複数の根ノードから*ModeChange*メッセージを受信す

る。連結成分のサイズを計算するために、連結成分にある木のサイズを蓄積する必要がある。フェーズ3で木のサイズを計算するために、根ノードは木のサイズだけでなく、隣接根ノードの ID (*AdjacentRoots*) も取得する。図 3.12A で、ノード 4 はノード 1 と 2 から *ModeChange* メッセージを受信する。ノード 4 は、それらのルートの ID を自分の *AdjacentRoots* に保存する (アルゴリズム 2:4 行)。その後、ノード 4 からノード 1 に *Reject* メッセージを送り (アルゴリズム 2:16 行)、ノード 2 に自分の *AdjacentRoots* を含む *Back* メッセージを送った後、ノード 1 と 2 の *AdjacentRoots* はそれぞれ $AdjacentRoots_1 = 1, 2$ 、 $AdjacentRoots_2 = 1, 2, 3$ になる (アルゴリズム 2:20,26 行)。ノード 1、2、3 の ID が $AdjacentRoots_2$ に存在するので、ノード 2 は自分の木がノード 1 と 3 の木と直接繋がっていることが分かる。

但し、全ての根ノードにおいて、フェーズ2が終わる時間は各自異なる可能性がある。同時にフェーズ3を開始するために、まずリーダーノードのみが、自分の配信木サイズを取得した直後にフェーズ3を開始すると仮定する (アルゴリズム 2:39 行)。そして、リーダーではない他のノードは、連結成分にある多数の配信木サイズを蓄積するために、リーダーからの *RequiringInfo* メッセージを受信するまで待機する (アルゴリズム 2:35-38, 41 行)。既にフェーズ1で、被害を受けたノードは *ExtendedCandidates* 内で最も ID が低いノードをリーダーノードとして定義した。

ノード i において、フェーズ2で用いる変数

Parent: 親ノードの ID

AdjacentRoots: 根ノード ID の集合。この集合にある根ノードの配信木は、ノード i の配信木と直接繋がっている。

NumberOfUnclearParentChild: まだ親子関係が決定されていない最近接ノード数。最小値は次数と同じ。

TreeSize: 葉ノードから、ノード i までの木のサイズ。

Algorithm 2 ノード i において、フェーズ2の疑似コード

- 1: ノード i の受信: ノード j からの $ModeChange(j, AdjacentRoots_j)$
- 2: **if** $Parent_i = NULL$ **then**
- 3: $Parent_i \leftarrow j$
- 4: $AdjacentRoots_i \leftarrow AdjacentRoots_j$
- 5: $TreeSize_i \leftarrow 1$
- 6: $NumberOfUnclearParentChild_i \leftarrow \|Neighbors_i\| - 1$

```

7:   if  $NumberOfUnclearParentChild_i = 0$  then
8:       送信 ノード  $j \rightsquigarrow Back(i, AdjacentRoots_i, TreeSize_i)$ 
9:   else
10:      for all  $k \in Neighbors_i \setminus j$  do
11:          ノード  $k \rightsquigarrow ModeChange(i, AdjacentRoots_i)$ 
12:      end for
13:   end if
14: else
15:      $AdjacentRoots_i \leftarrow AdjacentRoots_i \cup AdjacentRoots_j$ 
16:     送信 ノード  $j \rightsquigarrow Reject(i, AdjacentRoots_i)$ 
17: end if

18: ノード  $i$  の受信 : ノード  $j$  からの  $Reject(j, AdjacentRoots_j)$ 
19:    $NumberOfUnclearParentChild_i \leftarrow NumberOfUnclearParentChild_i - 1$ 
20:    $AdjacentRoots_i \leftarrow AdjacentRoots_i \cup AdjacentRoots_j$ 
21:   if  $NumberOfUnclearParentChild_i = 0$  then
22:       送信  $Parent_i \rightsquigarrow Back(i, AdjacentRoots_i, TreeSize_i)$ 
23:   end if

24: ノード  $i$  の受信 : ノード  $j$  からの  $Back(j, AdjacentRoots_j, TreeSize_j)$ 
25:    $NumberOfUnclearParentChild_i \leftarrow NumberOfUnclearParentChild_i - 1$ 
26:    $AdjacentRoots_i \leftarrow AdjacentRoots_i \cup AdjacentRoots_j$ 
27:    $TreeSize_i \leftarrow TreeSize_i + TreeSize_j$ 
28:   if  $NumberOfUnclearParentChild_i = 0$  then
29:       if  $Parent_i \neq i$  then
30:           送信  $Parent_i \rightsquigarrow Back(i, AdjacentRoots_i, TreeSize_i)$ 
31:       end if
32:   else
33:        $SetOfTreeSize_i \leftarrow SetOfTreeSize_i \cup \{TreeSize_i\}$ 
34:        $SetOfAdjacentRoots_i \leftarrow SetOfAdjacentRoots_i \cup \{AdjacentRoots_i\}$ 
35:       if ノード  $i$  がリーダー then
36:           for all  $k \in ExtendedCandidates_i \setminus i$  do

```

```
37:         送信 ノード  $k$  へ RequiringInfo( $i$ )
38:     end for
39:     フェーズ 2 を終了、フェーズ 3 を開始
40: else
41:     フェーズ 2 を終了するが、リーダーから RequiringInfo を受信するまで待
    機する
42: end if
43: end if
```

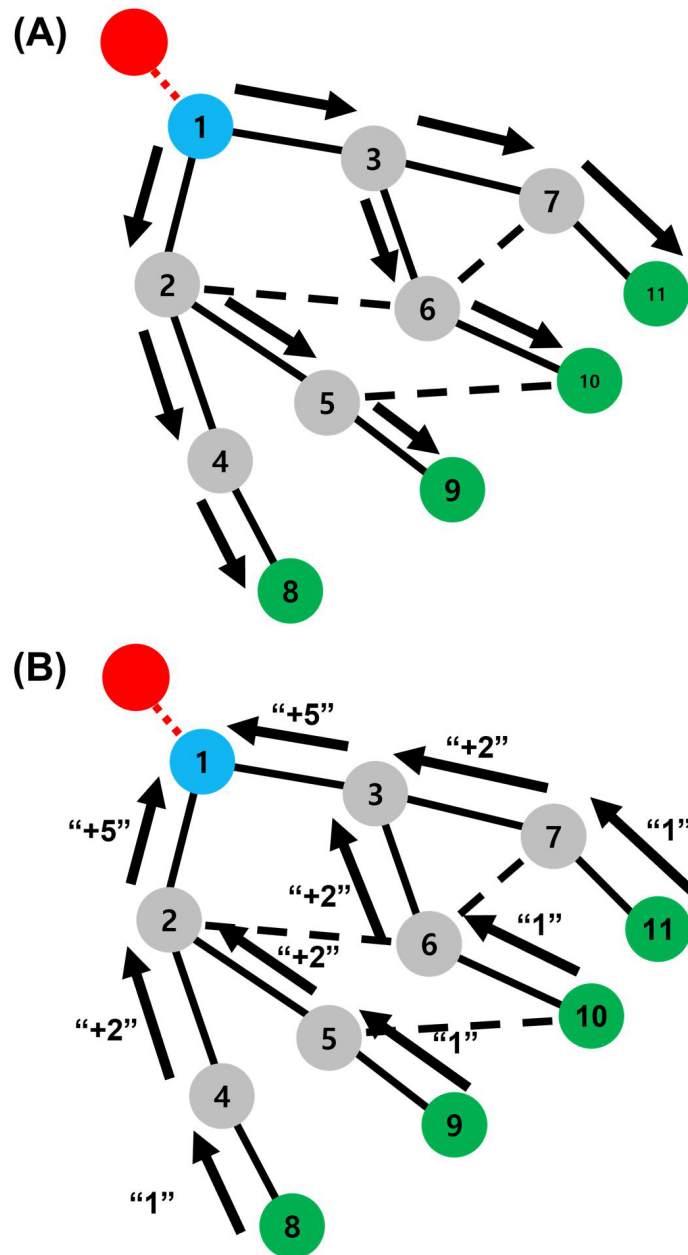


図 3.11: フェーズ 2 のプロセスの概要。円内の数字はノード ID を表す。(A) 根ノード (青色ノード) から葉ノード (緑色ノード) に *ModeChange* メッセージが伝播する。黒色矢印は *ModeChange* メッセージの方向を示す。赤色ノードと赤色線は故障ノードとそのリンクである。黒色線は配信木での親子関係を表す。黒色点線は、両ノードが親子関係を持たないことを意味する。(B) *Back* メッセージは葉ノードから送信される。"+ 数字" は *Back* メッセージ内である *TreeSize* を計算するための増分を示す。

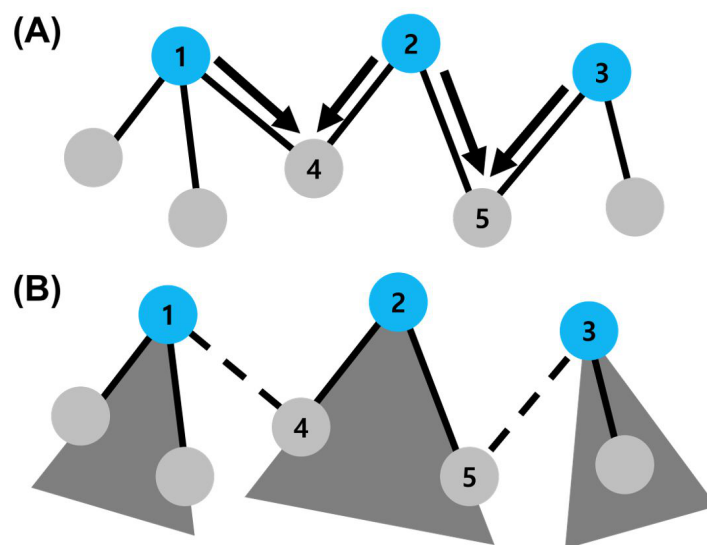


図 3.12: 進行中のフェーズ 2 の例。(A) 連結成分内にノード 1、2、3 の 3 つの根ノード（青色ノード）が存在する。この場合、ノード 4 と 5 は黒色矢印で示される複数の *ModeChange* メッセージを受け取れる。黒色線はノード間の直接的な連結を示す。(B) 結果として、連結成分内に 3 つの配信木（灰色三角形）が構築される。黒色点線は、2 つのノードが直接繋がっているが、互いに異なる配信木に属することを表す。

3.5.2.4 フェーズ3: 連結成分サイズの計算

分離された連結成分サイズを蓄積するために、根ノードはリーダーノードから *RequiringInfo* メッセージを受け取った後、*SendingInfo* メッセージを送信する (アルゴリズム 3:1-2 行)。 *SendingInfo* メッセージには次の3つの変数: (1) *TreeSize*、(2) *AdjacentRoots*、(3) *MalfuncNodes* が含まれる。 *TreeSize* と *AdjacentRoots* はフェーズ2で説明されている。 *MalfuncNodes_i* は、ノード *i* によって検知された故障ノードの ID とその次数を示す。この変数を設定するために、被害を受けたノードは既に通常モードで最近接ノードの次数を持っていると仮定した。リーダーが *SendingInfo* メッセージを受け取ると、メッセージ内の3つの変数それぞれはリーダーの *SetOfTreeSize*、*SetOfAdjacentRoots*、*SetOfAttackedNodes* へ併合される (アルゴリズム 3:4-6 行)。

図 3.13 で、ノード 1、2、3、4、5 はリーダーがノード 1 であり、同じ *ExtendedCandidates* を持っている。 *SendingInfo* メッセージを受け取ることにより、ノード 1 はノード 1、2、3、4、5 の木のサイズを知ることができる (図 3.13 の灰色の三角形)。しかしながら、それらの木が同じ連結成分内で繋がっているかどうかは分からない。したがって、それを区別するために *AdjacentRoots* を用いる。例えば、ノード 1 がノード 2、3、4、5 から *SendingInfo* メッセージを受け取るたびに、それ自分の3つの変数を更新する (アルゴリズム 3:4-6 行)。最後に、ノード 1 の *SetOfAdjacentRoots* は 1,2、1,2,3、2,3、4,5、4,5 となり、これは5つの部分集合を含む集合になる。 *SetOfAdjacentRoots₁* で、いくつかの部分集合の要素は重複する場合がある。

ExtendedCandidates での連結成分を区別するために、ノード 1 は *SetOfAdjacentRoots₁* の部分集合を重複がないまで、部分集合同士を併合する。そうすると *SetOfAdjacentRoots₁* は 1,2,3、4,5 になる。 *SetOfAdjacentRoots₁* の各部分集合は、ノード 1、2、3 が同じ連結成分に属し、ノード 4、5 がそれとは別の連結成分に属することを示す。結合された *SetOfAdjacentRoots₁* を用いて、ノード 1 は各連結成分のサイズを計算する (アルゴリズム 3:8 行)。分離された連結成分のサイズは、*SetOfTreeSize* に木のサイズとして存在する。このプロセスを通じて、リーダーノードは *SetOfAdjacentRoots* の各部分集合ごとに、連結成分のサイズが分かり、連結成分の数も分かる。リーダーノードはまた、*ExtendedCandidates* のノードが検知した故障ノードの ID と、その次数も取得する。

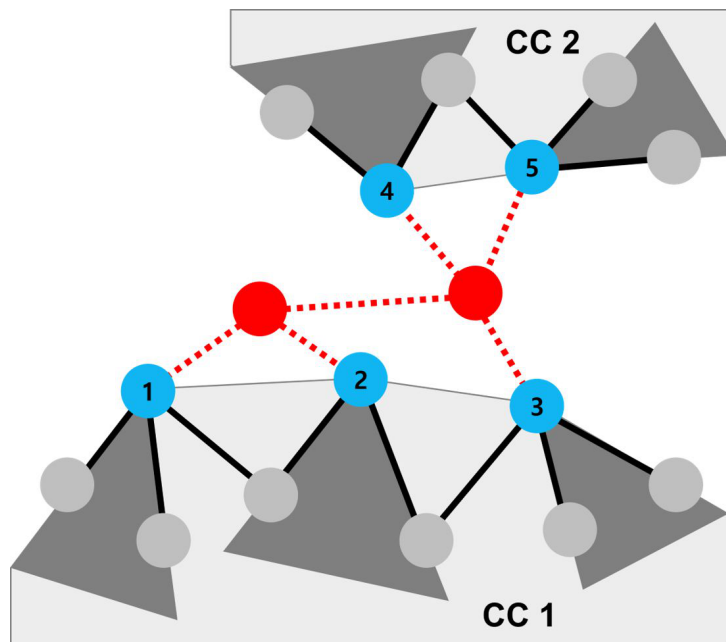


図 3.13: フェーズ 3 のプロセス。このネットワークでは、ノード 1 が最も低い ID を持つリーダーノードで、ノード 2、3、4、5 が拡張された修復候補であることを知っている。ノード 4 と 5 の配信木が別の連結成分にあるため、これらの配信木のサイズはノード 1、2、3 の配信木とは別に蓄積する必要がある。したがって、フェーズ 3 では、ノード 1 は 2 つの連結成分、CC1 と CC2 を区別している。赤色ノードと赤色線は故障ノードとそのリンクを示す。青色ノードは被害を受けたノードである。灰色三角形は各根ノードの配信木を表す。灰色多角形は、連結成分 (CC) である。

ノード i において、フェーズ 3 で用いる変数

MalfuncNodes: ノード i が検知した故障ノードの ID と次数。

SetOfTreeSize: リーダーの *ExtendedCandidates* から受信した *TreeSize* の集合。

SetOfAdjacentRoots: リーダーの *ExtendedCandidates* から受信した *AdjacentRoots* の集合。

SetOfAttackedNodes: リーダーの *ExtendedCandidates* から受信した *MalfuncNodes* の集合。

Algorithm 3 ノード i において、フェーズ 3 の疑似コード

- 1: ノード i の受信: ノード j からの $RequiringInfo(j)$
 - 2: 送信 ノード j へ $SendingInfo(TreeSize_i, AdjacentRoots_i, MalfuncNodes_i)$

 - 3: ノード i の受信: ノード j からの $SendingInfo(TreeSize_j, AdjacentRoots_j, MalfuncNodes_j)$
 - 4: $SetOfTreeSize_i \leftarrow SetOfTreeSize_i \cup \{TreeSize_i\}$
 - 5: $SetOfAdjacentRoots_i \leftarrow SetOfAdjacentRoots_i \cup \{AdjacentRoots_i\}$
 - 6: $SetOfAttackedNodes_i \leftarrow SetOfAttackedNodes_i \cup \{MalfuncNodes_i\}$
 - 7: **if** $SendingInfo$ を全ての $ExtendedCandidates$ から受信 **then**
 - 8: 各連結成分のサイズを計算する
 - 9: フェーズ 3 を終了し、フェーズ 4 を開始。
 - 10: **end if**
-

3.5.2.5 フェーズ4: リング形成

フェーズ4は、改良修復法のステップ2に対応するリング形成を行う。フェーズ3までに、被害を受けたノードは修復ための候補ノードを拡張し、それらが属する連結成分のサイズを計算した。このフェーズでは、リーダーノードと、拡張された候補ノード間でメッセージを交換することにより、リングを形成する。リングを形成する前に、リーダーノードはローカル資源制御のため、式(1)を変更して修復リンク数を計算する。リーダーノードはフェーズ3で故障ノードの次数を取得していたが、重複を避けて次数和を計算するには、故障ノード間のリンク有無を確認する必要がある。確認プロセスは次のとおりである。

リーダーノードはそれぞれ、自分の *ExtendedCandidates* 内のノードに *AllErrInfo* メッセージを送信する。故障ノードの ID が *AllErrInfo* メッセージに入っている。*AllErrInfo* メッセージを受け取ることで、他のノードも故障ノードの ID を取得できる。元々被害を受けたノードは隣接ノードの故障だけを検知でき、3ホップまでのノードの ID を保存するローカルマップを持つ。被害を受けたノードは、故障ノードの ID が自分のローカルマップに存在するかを確認できる。ローカルマップ内で、故障ノード間のリンクは *DupCountLinks* 変数として定義される(アルゴリズム4: 1行)。この変数は、故障ノードの ID ペアを表す。その後、ノードは *DupCountLinks* を *DupLinksInfo* メッセージに入れてリーダーノードに送信する。リーダーノードが自分の *ExtendedCandidates* 内の全てのノードから *DupLinksInfo* メッセージを受け取ると、式(1)の修正に従って修復リンク数を計算できる(アルゴリズム4: 6-7行)。修復リンク数を計算した後、リーダーノードは *CommandConnec* メッセージを自分の *ExtendedCandidates* 内のノードに返信する(アルゴリズム4: 8-10行)。例えば、図3.14Aで、リングの連結順を [1, 4, 5, 6, 3, 2, 1] だとする。この順番に従い、ノード5は、リーダーからノード6の ID が入っている *CommandConnec* メッセージを受け取り、ノード6と修復リンクを生成する。各ノードが独立して修復リンクを生成することにより(図3.14Bの緑色矢印)、拡張された候補ノード間に大きなリングが形成される。

ノード i において、フェーズ4で用いる変数

DupCountLinks: 故障ノード間にあるリンク ID。この ID は、例えばノード i と j 間にあると、 (i, j) と表記する。このリンクの数は、故障ノードの次数和から引かれる。

IdOfNewNeighbor: 修復リンクで連結すべきノード ID。

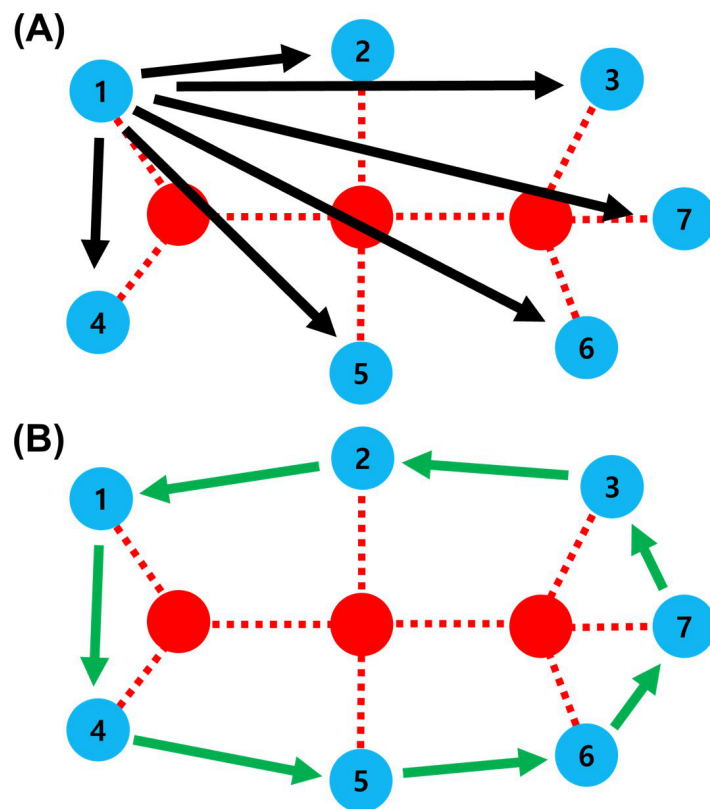


図 3.14: リング形成の概要図。(A) リーダーノード (ノード 1) は拡張された修復候補にメッセージを送信する (黒色矢印)。(B) メッセージを受信すると、被害を受けたノード (青色ノード) は、各自新しいリンク (緑色矢印) を生成する。赤色ノードは故障ノードにである。

Algorithm 4 ノード i において、フェーズ 4 の疑似コード

- 1: ノード i の受信: ノード j からの $AllMalfuncEvents(SetOfAttackedNodes_j)$
 - 2: 故障ノード間にあるリンクを確認して、そのリンクの ID を $DupCountLinks_i$ に入れる。
 - 3: 送信 リーダーへ $DupLinksInfo(DupCountLinks_i)$

 - 4: ノード i の受信: ノード j からの $DupLinksInfo(DupCountLinks_j)$
 - 5: $DupCountLinks_i \leftarrow DupCountLinks_i \cup DupCountLinks_j$
 - 6: **if** ノード i が自分のローカルマップにある全てのノードから $DupLinksInfo$ をもらう **then**
 - 7: **for all** $do k \in ExtendedCandidates_i \setminus i$
 - 8: 送信 ノード k へ $CommandConnec(IdOfNewNeighbor_i)$
 - 9: **end for**
 - 10: **end if**

 - 11: ノード i の受信: ノード j からの $CommandConnec(IdOfNewNeighbor_j)$
 - 12: $IdOfNewNeighbor_j$ にある ID と新しいリンクを生成する。
-

3.5.2.6 フェーズ5: 各リング上でループ強化

フェーズ5は、改良修復法のステップ3に対応して、各リーダーノードが、ループを強化するために、自分のリング上で最小次数のノードを2つ選ぶことを基本プロセスとする。リング形成後、残余リンク数だけ、以下のプロセスが繰り返される。リーダーノードは、選んだ2つのノードに *AddingLink* メッセージを送信する。この *AddingLink* メッセージには、連結すべきノード ID があり、修復リンクを生成する (アルゴリズム 5: 2行)。その間、リーダーノードは、ノードの次数を再計算した後、他のノードに *AddingLink* メッセージを送信する。リーダーノードは、残余リンクがなくなるまで、*AddingLink* メッセージの送信を繰り返す。最後に、各リーダーノードは、修復プロセスを終了するために、自分の *ExtendedCandidates* のノードに *Ter* メッセージを送信する。この *Ter* メッセージを受け取ったノードは、フェーズ2で構築した配信木を通じて *Ter* メッセージを伝搬させる。また、モードを修復から通常に変更する (アルゴリズム 5: 4行)。修復したネットワークで、通常機能を再開する前に、各ノードは最近接ノードが通常モードに変更されたかを確認する必要がある。但し、それはアルゴリズムを実行させるシステムによって、必要な準備が異なる。例えば、通信ネットワークでは新しいルーティングテーブルが必要である。

ノード i において、フェーズ5の疑似コード

IdOfNewNeighbor: 修復リンクを生成すべきノード ID。

Algorithm 5 ノード i において、フェーズ5の疑似コード

- 1: ノード i の受信: ノード j からの *AddingLink*(*IdOfNewNeighbor_j*)
 - 2: *IdOfNewNeighbor* と新しいリンクを生成する。
 - 3: ノード i の受信: ノード j からの *Ter*()
 - 4: 修復プロセスを終了し、通常モードへ転換する。
-

3.5.3 まとめ

先の修復法 [23] をさらに改善するため、分散処理に基づく自己修復によるネットワークを再構築する改良修復法を提案した。特に、リング上でループを強化するために、最大

次数と最小次数の差を縮小するような、低い次数ノード同士に修復リンクを追加した。コンピュータシミュレーションを通じて、改良修復法が先の修復法 [23] よりも高い頑健性と効率性を持つを示した。

改良修復法の主なアイデアは、より狭い次数分布の差が、より高い頑健性をもたらすこと [17, 19, 46] である。その結果、改良修復法によって再構築されたネットワークの次数分布は、元のネットワークの長い裾野を持つべき乗分布から、狭い範囲の次数分布を持つように変更される。さらに、偶然の物理的要因による遅延を想定して、タスクの実行順序をランダム化することによって、改良修復法を非同期処理でエミュレートをした。記述した分散アルゴリズムにより、修復リンクは迅速な再構築のために配分される。さらに、局所分散プロセスは逐次プロセスとほぼ同じ結果が得られることも確認している。これらの発見は、局所分散修復によって、頑健で効率的なネットワークを再構築する方法の基礎を提供する。将来の研究としては、より破壊的なループ攻撃に対する我々の方法の有効性を調査する。例えば、CoreHD [50]、集団影響 [51]、BP 攻撃 [52] などが考えられる。

第4章

反モジュール化における頑健性と効率性の共存

4.1 コミュニティ構造と頑健性の関係

エネルギー、交通、通信システムは、社会を支えるためのサービスを提供する。しかしながら、これらのネットワークシステムは、ハブに対する攻撃に極めて脆弱なスケールフリー (SF) 構造を持つ [2]。したがって、より頑健なネットワークを構築することは、複雑ネットワークにおける重要な課題の1つである。最近、ネットワークを分断するディスマントリング問題と、ネットワークからループを消すデシクリング問題が、ネットワークサイズ N が無限大に発散する場合において漸近的に等価であること、ループ構造を強化することが頑健なネットワークを構築するのに非常に重要であることが明らかになっている [16]。ここで、ディスマントリング問題は、ネットワークを指定されたサイズまで破壊するために必要な最小数の除去ノードを見つけることであり、デシクリング問題はループを形成するために必要な最小数のノードを見つけることである。全てのループがネットワークから取り除かれると、そのネットワークはどの節ノードを削除してもすぐに分断される木構造になる。したがって、ネットワークが木構造にならないためには、ループ強化が重要と言える。実際に、ループ強化に基づいたリワイヤリング法 [17] は、頑健なネットワークを生成しながら、最大次数と最小次数の差を減少させる共通の現象が観測された。言い換えれば、この差が小さいほどネットワークはより頑健になる。最も極端な場合は、差が0であるランダム d -レギュラーグラフが、最適な頑健性を持つ [20] [53]。

これまで、レギュラーグラフの研究は数多く存在するが [54] [55]、それらは頑健性の研究ではなく、主にスペクトル解析やグラフ理論のための研究であった。むしろ、頑健性研

究の対象では、Erdős-Rényi (ER) ランダムグラフや SF ネットワークが対象とされた。ゆえに、レギュラーグラフは頑健性研究の盲点となっている。

一方、多くの実ネットワークがモジュラー構造を持っているため、複雑ネットワークにおいてモジュラー構造もまた重要な課題となる。例えば、社会ネットワークにおいて、モジュールは共通の興味や背景を持つコミュニティやグループに相当する。ネットワークが高いモジュール性を持つ場合、同じモジュール内のノード同士は互いに密に連結されているが、異なるモジュールのノード同士は疎に連結されている。最近の研究によると [29] [26]、モジュラー構造を持つ ER ランダムグラフや SF ネットワークは、モジュラー構造がないものよりも攻撃に対して脆弱であることが報告されている。ここで、ネットワークのモジュール性はリンクのリワイヤリングによって制御される。さらに、モジュラーネットワークに対して最も破壊的攻撃である、高い媒介中心性を持ちながら、モジュール間を繋ぐノードを優先的に狙うモジュールベース (MB) 攻撃法が提案されている [56]。上記の研究から、例え最適の耐性をランダム d -レギュラーグラフ [20] [53] でもモジュラー構造を持つと脆弱になると予測される。モジュラー d -レギュラーグラフのモジュール性は、[26] から着想を得たリワイヤリング法によって制御する。ここで、さらに低いモジュール性の d -レギュラーグラフは、連結の頑健性と経路の効率性を同時に持つことを新たに示す。

4.2 反モジュール化による性モジュール性制御

4.2.1 コミュニティ構造を持つレギュラーグラフの設定

ランダム d -レギュラーグラフで構成されるモジュラーネットワークを考える。各ランダム d -レギュラーグラフはモジュールに相当する。図 4.1A のように、初期状態のネットワークは、サイズ N_m のモジュール m_o 個がリング状に繋がっている。ここで、ネットワークのノード総数とリンクの総数は $N = N_m \times m_o$ および $M = (d \times N)/2$ で一定とする。この初期構成は、リング構造が最小数のモジュール間リンク (インターリンク) と最大数のモジュール内リンク (イントラリンク) で、ネットワーク全体の連結性を維持するため、非常に高いモジュール性を持つ。しかしながら、このネットワークは、モジュール間を繋ぐノードを狙って削除する攻撃に極めて脆弱である。特別な例として、各モジュールが完全結合のクリークなら、そのネットワークは穴居人グラフ [57] と呼ばれる。穴居人グラフでは、少数のイントラリンクからインターリンクへリワイヤリングすることで、平均経路長が $O(\log N)$ の小世界 (SW) 特性が発現する。このように、私達は頑健性だけ

でなく、SW 特性で表す効率性についても議論する。

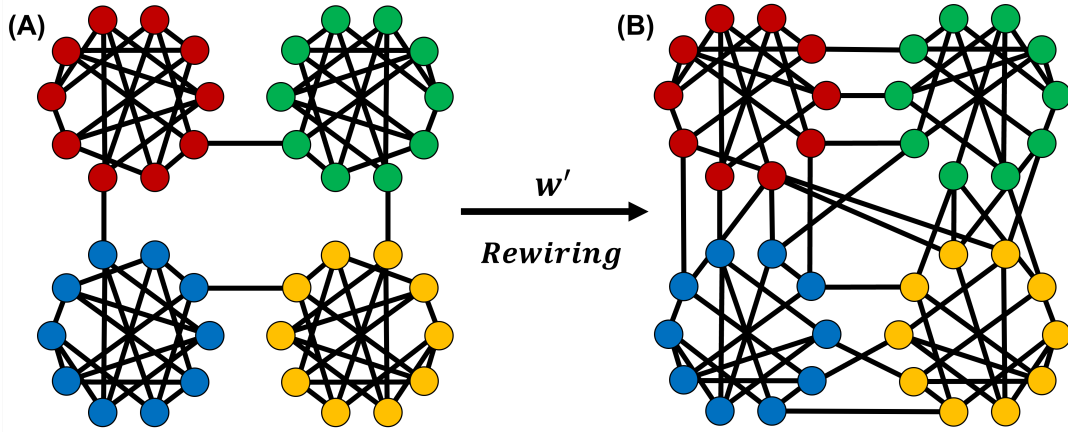


図 4.1: リワイヤリング前後のモジュラーネットワーク構成。(A) 高いモジュール性を持つ初期構成。(B) リワイヤされたネットワークは頑健性が増加する。ノード色はモジュールを意味する。

4.2.2 反モジュール化と従来のモジュール化

ネットワークのモジュール性を制御するため、図 4.1B に示すように、イントラリンクをインターリンクへ一様ランダムにリワイヤリングする。このリワイヤリング法は、従来のモジュール化 [26] の逆プロセスと見なされるが、ランダム結合のためリング構造をあまり作らない。表 4.1 は、初期のイントラリンクおよびインターリンクの数を示す。

表 4.1: イントラリンクとインターリンクの最初値。

	イントラリンク	リワイヤリング	インターリンク
反モジュール化	$M - m_o$	$\frac{w'}{w}$	m_o
モジュール化 [26]	M/m_o	$\frac{w}{w'}$	$M - M/m_o$

反モジュール化によって、イントラリンクの数は減少するのみである。なぜなら、イントラリンクとインターリンクの合計は定数 M だからである。その代わりに、インターリンクの数はモジュール化 [26] によってただ減少するのみである。リワイヤリングするリンク数の比率 w' を用いて、反モジュール化においてイントラリンクがランダムにインターリンクへリワイヤされる (図.4.1(B))。一方でリワイヤリング率 w を用いて、モジュール

化においてインターリンクがランダムにイントラリンクへリワイヤされる [26]。さらに、 w' と w の関係は以下の式から導かれる。

$$m_o + w'(M - m_o)w' = (1 - w)(M - M/m_o)。$$

上式の左辺と右辺はリワイヤリングした後のそれぞれ反モジュール化した後のインターリンク数と、モジュール化 [26] した後に残ったインターリンク数である。上式より、

$$w' = 1 - \frac{1 + w(m_o - 1)}{m_o(M - m_o)} M, \quad (4.1)$$

が得られる。反モジュール化の詳細なプロセスを以下にまとめる。このプロセスを通じて、リワイヤされたネットワークの次数分布と全体の連結性は維持できるが、従来のモジュール化 [26] ではそうではない。このような違いについては 5.3 節で議論する。

- ステップ 1 最初に、あるイントラリンクをランダムに削除する。削除されたリンクの末端ノード 1 つをランダムに選択する。
- ステップ 2 ステップ 1 で選択したノードとは異なるモジュールに属するイントラリンクをランダムに削除する。削除されたリンクの末端ノード 1 つをランダムに選択する。
- ステップ 3 選択した 2 つのノード間に新しいインターリンクを生成する。
- ステップ 4 ステップ 2 で、あるいは以前のステップ 4 で削除したリンクとは異なるイントラリンクを削除する。削除されたリンクの末端ノード 1 つをランダムに選択する。
- ステップ 5 次数を保存するために、ステップ 2 で、あるいは以前のステップ 4 で選択されなかった末端ノードを再び選択する。選択された 2 つのノード間に新しいインターリンクを生成する。
- ステップ 6 与えられた $0 < w' < 1$ に対して $w'(M - m_o)$ 個のイントラリンクが生成されるまで、ステップ 4 と 5 を繰り返す。リワイヤリングによってネットワークが分断される場合、他の末端ノードやイントラリンクを再び選択する。繰り返しの最後に、ステップ 1 で選択されなかったノードに最後のインターリンクが追加される。

頑健性を調べるために、以下の 3 種類の攻撃 RF、IB、MB を考える。RF(ランダム障害) は、一様ランダムに選ばれたノードが削除される。IB(初期の媒介中心性) 攻撃は、媒介中心性が高い順でノードが削除される。MB(モジュールベース) 攻撃は、基本的に媒介中心性が高い順でノードが削除されるが、最大連結成分に属するしながら、インターリ

リンクの末端ノードであるノードが標的として優先される。MB および IB 攻撃は、特にモジュラーネットワークにとって破壊的であるが、RF はよく知られた典型的な障害で、MB、IB 攻撃との頑健性を比較するために考慮する。

4.3 反モジュール化の効果確認

4.3.1 反モジュール化による頑健性の向上

反モジュール化したネットワークの頑健性とパスの効率を調べる。ここで、モジュラーネットワークの次数とサイズは、それぞれ $d = 4, 9, 19$ と $N = 10^4$ とする。サイズ $N = m_o \times N_m$ が一定値なら、与えられたモジュール数 m_o に対して、各モジュールのサイズ N_m も一定となる。 m_o が最大である場合、各モジュールがクリーク K_{d+1} の穴居人グラフになる [57]。以下の 4 つの分析指標を、割合 w' を変えながら 100 回測定して平均値を求める。

1. モジュール性 [28] $Q = \frac{1}{2M} \sum_{i,j} (A_{ij} - \frac{k_i k_j}{2M}) \delta_{i,j}$ 、 A は隣接行列、 k_i はノード i の次数、 $\delta_{i,j}$ はもしノード i と j が同じモジュールに属すると 1、そうではないなら 0 とする。
2. 最大連結成分のサイズの比率 $S^{1st}(q)/N$ 、 $S^{1st}(q)$ は qN 個のノードを壊して最大連結成分に属するノード数。 $0 < q \leq 1$ は攻撃で壊れたノード数の比率である。
3. 頑健性指標 [11] $R = \frac{1}{N} \sum_{q=\frac{1}{N}}^1 \frac{S(q)}{N}$ 。和は $q = \frac{1}{N}, \frac{2}{N}, \dots, \frac{N-1}{N}$ 、そして $\frac{N}{N} = 1$ を表す。
4. 平均最短経路長 $L = \frac{1}{N(N-1)} \sum_{i,j} d_{ij}$ 、 d_{ij} はノード i と j 間の最短経路の長さである。

反モジュール化されたネットワークにおいて、MB 攻撃に対する最大連結成分サイズの比率 $S^{1st}(q)/N$ を調べる。図 4.2A、B、C、D、E、および F は、それぞれ $m_o = 1000, 500, 100, 50, 20$ 、および 5 に対応する 4-レギュラーグラフに対する結果である。図 4.2A に示すように、 $w' = 0$ の灰色線は初期のリング構造なので、頑健性が急激に減少する。しかしながら、反モジュール化で比率 w' の増加につれて、色線は右にシフトしていく。 $w' = 0.9$ で、色線はモジュール構造を持っていないランダム d -レギュラーグラフのための紫色線に近づく。このような色線のシフトは、図 4.2B、C、D、E、および F でも観察できる。各図 4.2A、B、C、D、および E から、同じ色線を比較すると、共通的に m_o が小さいほど、色線はより右にシフトする。

一方、図 4.2C、D、E、および F では、 $w' \geq 0.5$ に相当する色線は (青色と紫色線) ほぼ重なっている。さらに、図 4.2F のように、 $w' = 0$ および 0.1 (灰色と赤色線) の場合、色線はまず $S^{1st}(q)/N \approx N_m/N = 0.2$ までに減少する。その後、色線は約 0.2 から徐々に減少する形を表す。これを理由は、MB 攻撃がネットワークを 2 段階で破壊するからである。まず、MB 攻撃でインターリンクの末端ノードを狙い、全てのモジュールを孤立させる。次に、孤立した各モジュールが徐々に崩壊する。これらの MB 攻撃に対する結果はネットワークの d が 4、19 でも、同じ結果となる (図 4.34.4)。IB 攻撃および RF の場合でも、MB 攻撃に対する結果と同じであるが、頑健性は RF、IB、MB 攻撃順に高くなる。詳しい数値は後の表 4.2 で示す。

しかしながら、いくつかのネットワークでは全体と異なる結果を出した。例えば、 $d = 4$ で $m_o = 2000$ の場合、 $w' = 0.01$ の頑健性が $w' = 0.1$ のより高い。また、 $m_o = 5$ の場合、 $w' = 0.1$ の頑健性が高い攻撃率 q では $w' = 0.9$ のものを逆転する。反モジュール化によりモジュール性が低くなるにもかかわらず、攻撃に対する頑健性が逆転する現象の理由はまだ明らかではない。したがって、 $m_o = 4、8$ の結果を出して、逆転現象の理由がモジュール数と関係あるかを調べる。図 4.5 は、 $d = 4、m_o = 4、8$ の場合を、3つの攻撃に対する $S^{1st}(q)/N$ を示す。逆転現象は MB 攻撃だけに起こり、 $S^{1st}(q)/N$ の差はモジュール数が小さいほど大きい。但し、モジュール数が小さいほどモジュールのサイズは大きいし、MB 攻撃は LCC をまずモジュール 1つのサイズまで減少させることを考えると、モジュール数が小さいほど、MB 攻撃に対する $S^{1st}(q)/N$ の差が大きくなるのは自然である。また、図 4.64.7 のように、より高い次数 $d = 9、19$ での結果を見ると、逆転現象は著しくない。つまり、反モジュール化において頑健性の逆転現象はスパースなネットワーク上で、MB 攻撃の性質により生じる現象である。

表 4.2 は、 $d = 9$ と $m_o = 200$ のネットワークにおいて、MB、IB 攻撃、そして RF に対する第 2 大連結成分のサイズが最大になる q の臨界値 $q_c(q_c^{MB}, q_c^{IB}, \text{そして } q_c^{RF})$ を示す。ここで、最大連結成分のサイズが攻撃により連続的に減少しつつ、ある臨界値でほぼ 0 になる形のような 2 次相転移を見せると、その臨界値では第 2 大連結成分のサイズが最大になり、臨界値は頑健性の指標にもなる [26]。 q_c^{MB} 、 q_c^{IB} 、そして q_c^{RF} の値は、 w' が大きくなるほど増加する。特に、 q_c^{MB} と q_c^{IB} は $w' = 0.1$ と $w' = 0.3$ の間で急激に増加する。 $w' \leq 0.3$ の場合、 q_c^{RF} は q_c^{MB} と q_c^{IB} より高い。しかしながら、 $w' \geq 0.5$ の場合、 q_c^{RF} は q_c^{MB} と q_c^{IB} よりも低くなる。特に $w' = 0.9$ の場合、 q_c^{IB} はランダム d -レギュラーグラフにおける浸透閾値 $1 - 1/(d - 1) = 1 - 0.1251 - 1/(d - 1) = 1 - 0.125(d = 9$ の場合) に近づく [20]。表 4.2 から、低いモジュール性を持つ ($w' \geq 0.5$) ランダム d -レギュラーグラフが、意図的な攻撃の IB に対して、偶発的な故障の RF より、強い頑健性

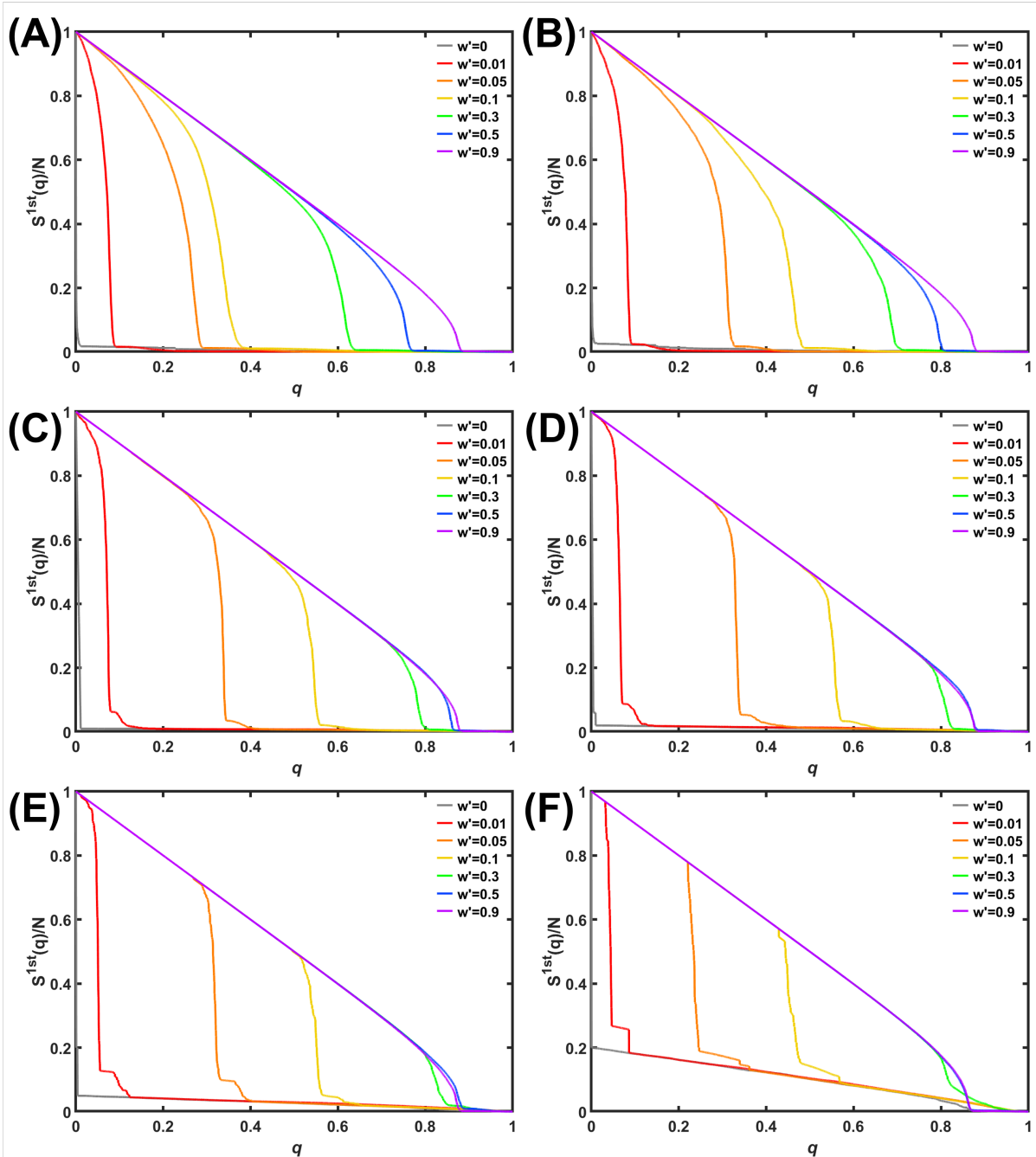


図 4.2: MB 攻撃に対する最大連結成分のサイズ比率。ネットワークの次数 d は 9 で、モジュール数 m_o は、(A) $m_o = 1000$, (B) $m_o = 500$, (C) $m_o = 100$, (D) $m_o = 50$, (E) $m_o = 20$, and (F) $m_o = 5$ である。線の色は反モジュール化における、リワイヤするイントラリンク数の比率 w' に対応する。

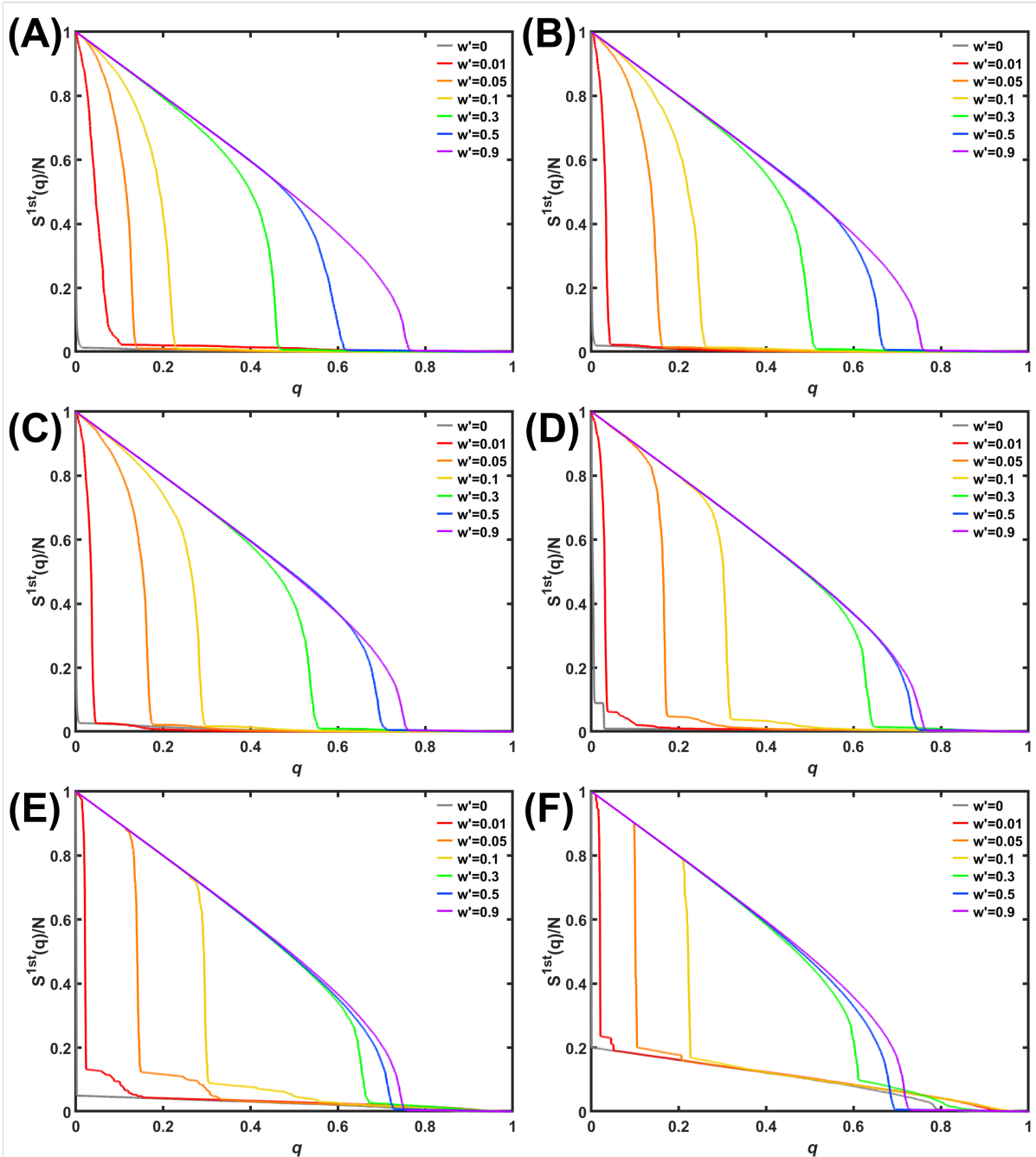


図 4.3: MB 攻撃に対する最大連結成分のサイズ比率。ネットワークの次数 d は 4 で、モジュール数 m_o は、(A) $m_o = 2000$, (B) $m_o = 1000$, (C) $m_o = 500$, (D) $m_o = 100$, (E) $m_o = 20$, and (F) $m_o = 5$ である。線の色は反モジュール化における、リワイヤするイントラリンク数の比率 w' に対応する。

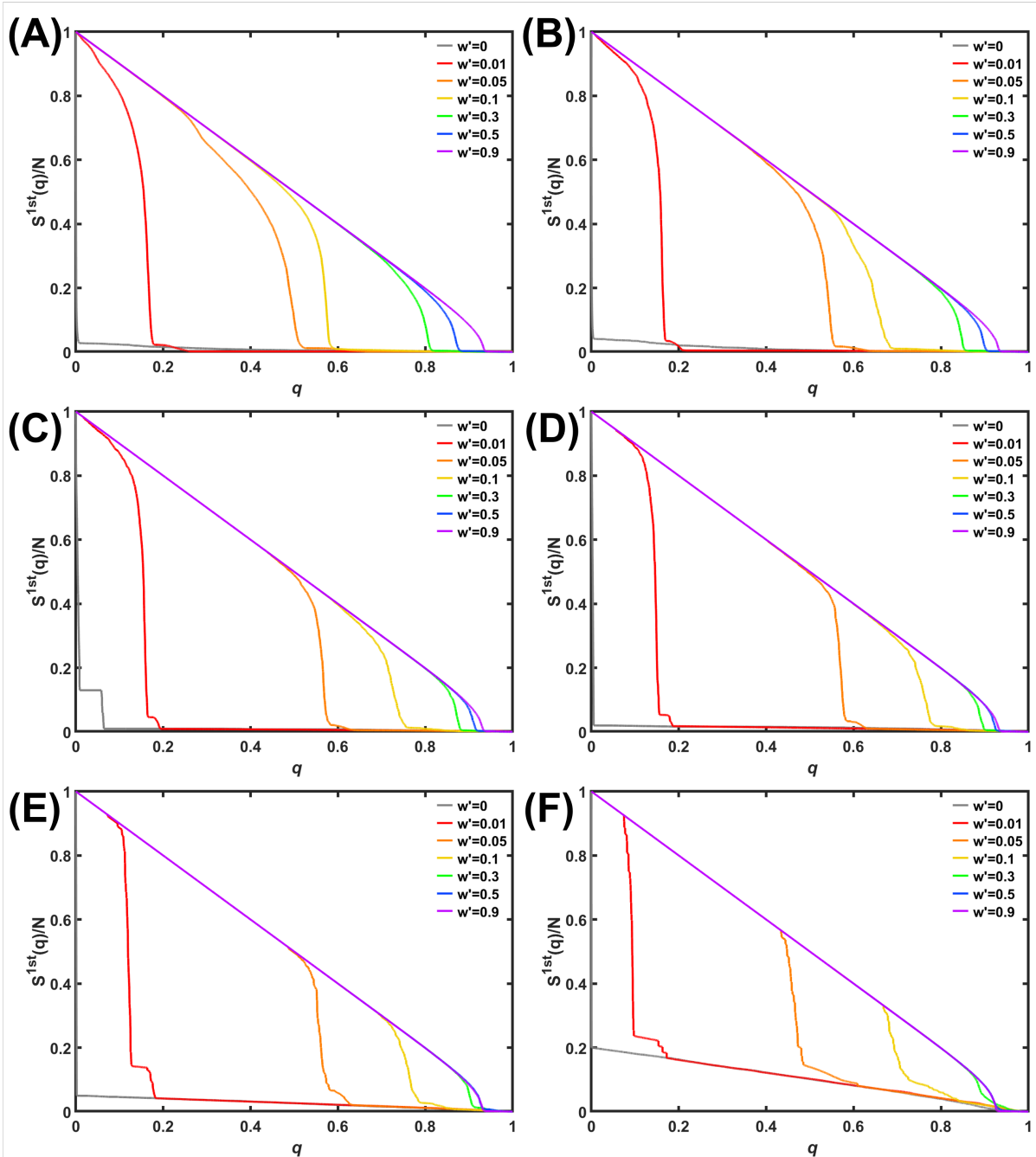


図 4.4: MB 攻撃に対する最大連結成分のサイズ比率。ネットワークの次数 d は 19 で、モジュール数 m_o は、(A) $m_o = 500$, (B) $m_o = 200$, (C) $m_o = 100$, (D) $m_o = 50$, (E) $m_o = 20$, and (F) $m_o = 5$ である。線の色は反モジュール化における、リワイヤするイントラリンク数の比率 w' に対応する。

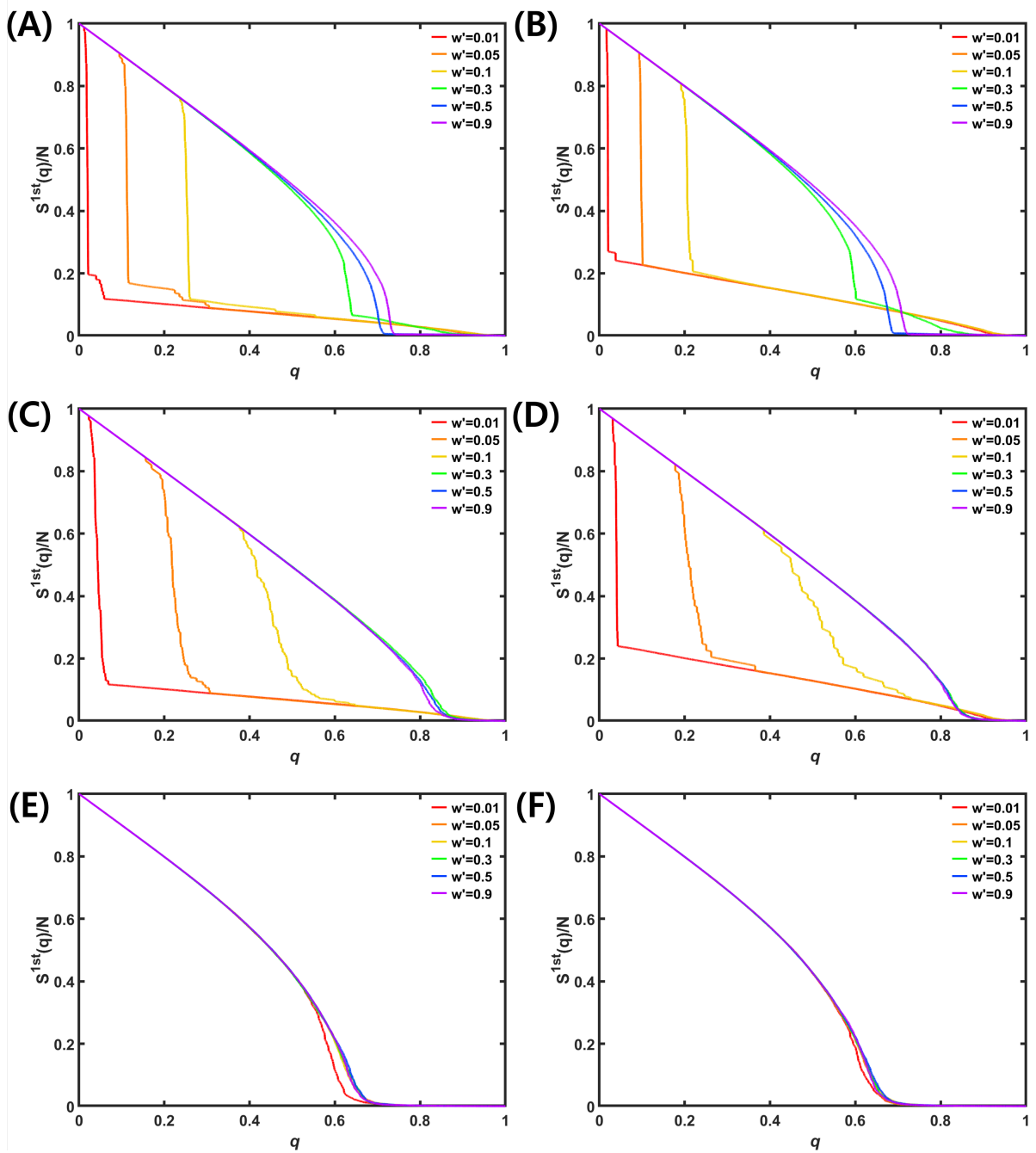


図 4.5: 3つの攻撃に対する最大連結成分のサイズ比率。 $d = 4$ で、(A) $m_o = 8$ 、MB 攻撃、(B) $m_o = 4$ 、MB 攻撃、(C) $m_o = 8$ 、IB 攻撃、(D) $m_o = 4$ 、IB 攻撃、(E) $m_o = 8$ 、RF and (F) $m_o = 4$ 、RF の場合、最大連結成分のサイズ比率である。線の色は反モジュール化における、リワイヤするイントラリンク数の比率 w' に対応する。

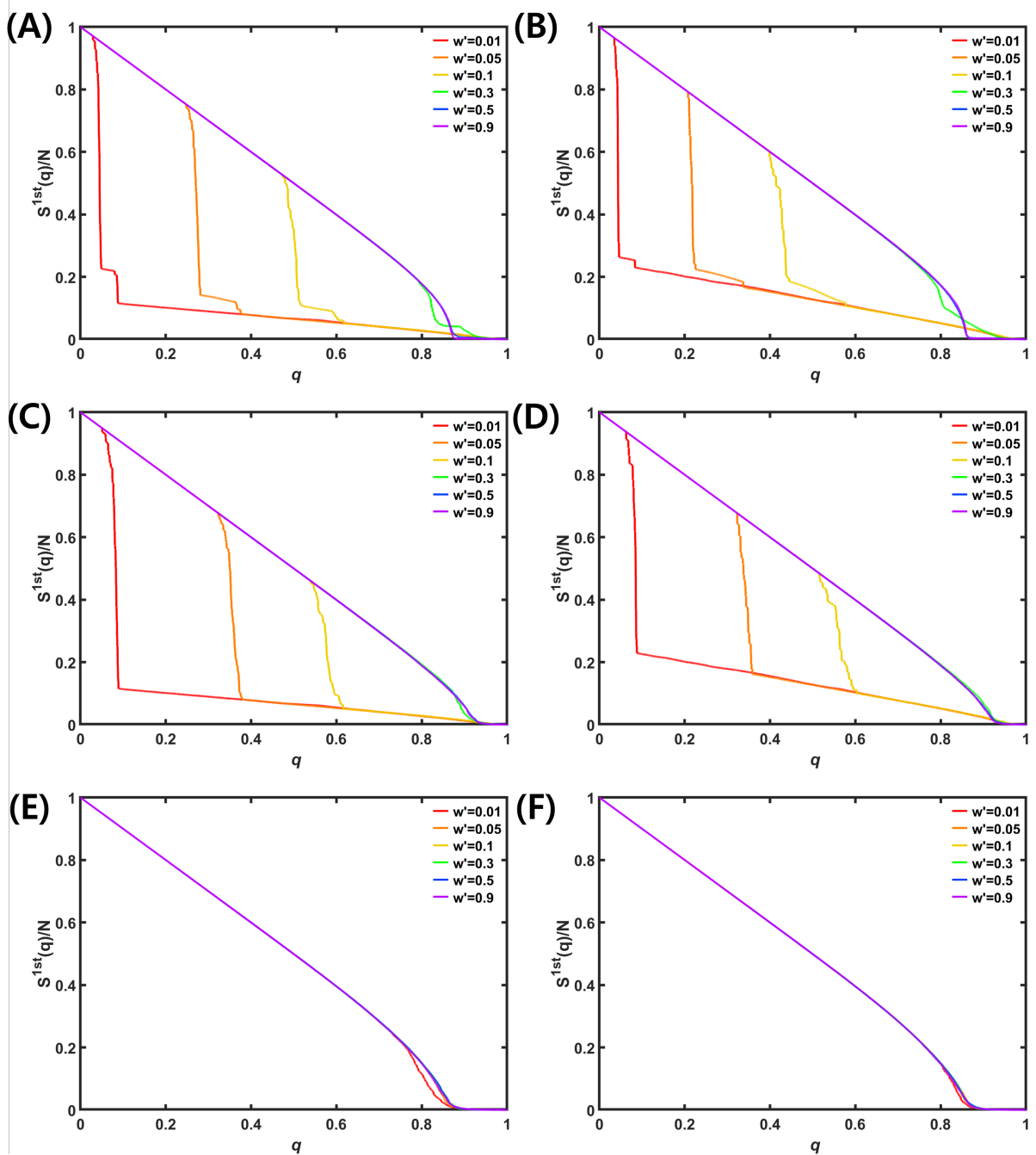


図 4.6: 3つの攻撃に対する最大連結成分のサイズ比率。 $d = 9$ で、(A) $m_o = 8$ 、MB 攻撃、(B) $m_o = 4$ 、MB 攻撃、(C) $m_o = 8$ 、IB 攻撃、(D) $m_o = 4$ 、IB 攻撃、(E) $m_o = 8$ 、RF and (F) $m_o = 4$ 、RF の場合、最大連結成分のサイズ比率である。線の色は反モジュール化における、リワイヤするイントラリンク数の比率 w' に対応する。

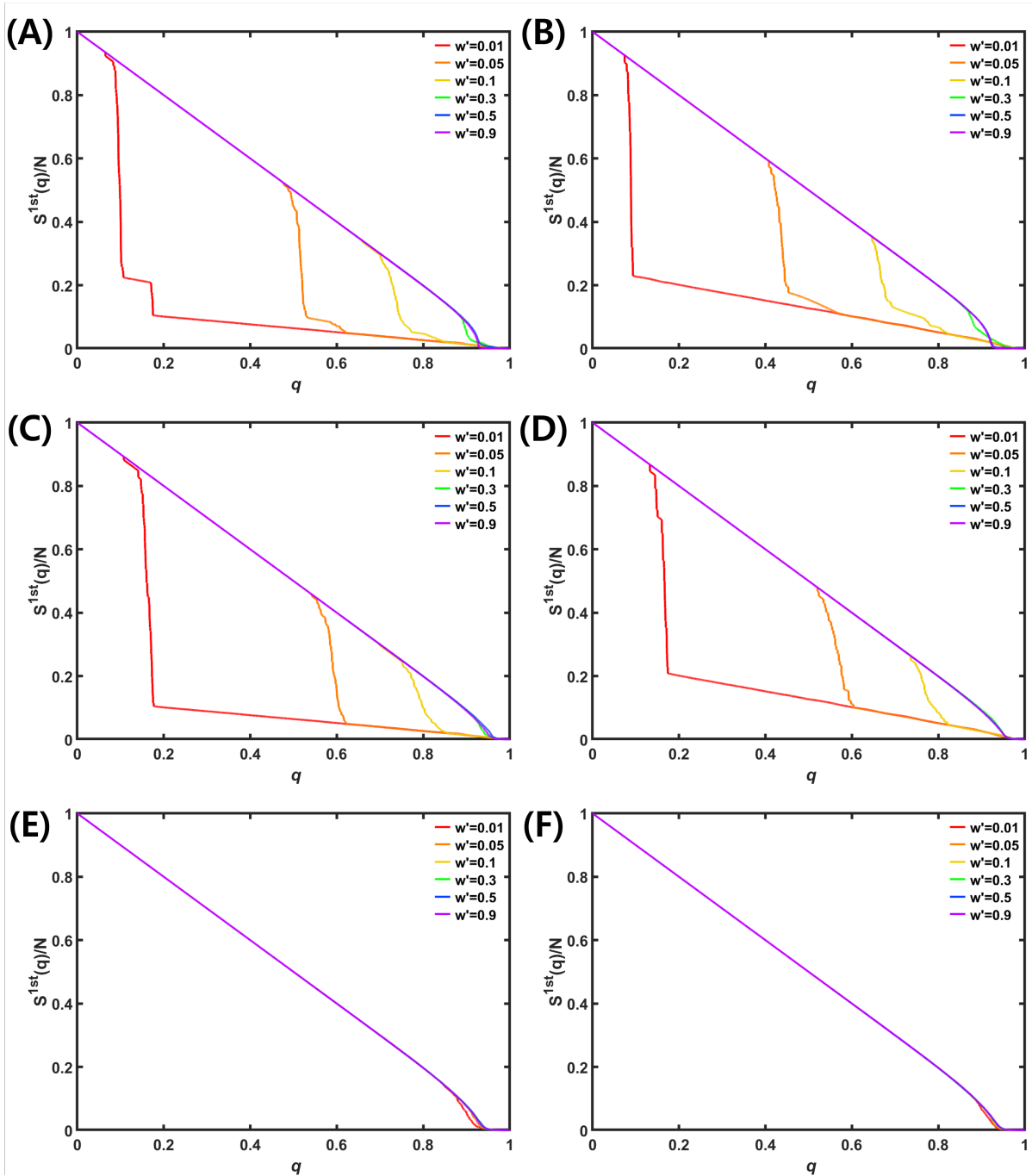


図 4.7: 3つの攻撃に対する最大連結成分のサイズ比率。 $d = 19$ で、(A) $m_o = 8$ 、MB 攻撃、(B) $m_o = 4$ 、MB 攻撃、(C) $m_o = 8$ 、IB 攻撃、(D) $m_o = 4$ 、IB 攻撃、(E) $m_o = 8$ 、RF and (F) $m_o = 4$ 、RF の場合、最大連結成分のサイズ比率である。線の色は反モジュール化における、リワイヤするイントラリンク数の比率 w' に対応する。

を持つことを発見した。この現象は、ER ランダムグラフや SF ネットワークでは見られない [58]。この現象が生じる理由としては、IB 攻撃は最短経路が頻繁に通過するコア部分にあるノードのみを削除する代わりに、周辺部にある残りのノードがネットワーク全体の連結性を維持できたと考えられる。

表 4.2: $d = 9$ と $m_o = 200$ ネットワーク上で、3種類の攻撃 (MB、IB、RF) に対する、攻撃率 q の臨界値 q_c 。

	$w'=0.01$	$w'=0.05$	$w'=0.1$	$w'=0.3$	$w'=0.5$	$w'=0.9$
MB	0.075	0.340	0.546	0.786	0.858	0.874
IB	0.092	0.353	0.559	0.803	0.886	0.911
RF	0.592	0.775	0.809	0.848	0.850	0.861

反モジュール化において、高い w' でネットワークは、低いモジュール性を持つことを示す。図 4.8 では、 m_o の変化にかかわらず、3種類の攻撃に対する頑健性指標 R とモジュール性 Q が単調減少関数である。図 4.8A と B では、 $Q < 0.2$ のとき、 w' に対応する全ての色線が、モジュール構造がないランダム 9-レギュラーグラフに対する頑健性を表すシアン色線に近づく。図 4.8C にある、全ての色線も、同じく $Q < 0.5$ のときにシアン色線に近づく。さらに、図 4.8A と B では、2つの現象が分かる。1つは、モジュール数が $m_o = 1000$ から $m_o = 50$ までに変化すると、赤色線から緑色線にシフトする。逆に、モジュール数が $m_o = 50$ から $m_o = 5$ までに変化すると、緑色線から紫色線にシフトする。しかしながら、このような現象は図 4.8C では存在しない。但し、これらの現象の理由は不明である。

また、高い Q を持つネットワークの R_{MB} 、 R_{IB} 、そして R_{RF} は、モジュール構造がない ER ランダムグラフと (黒線) や、非常に脆弱な構造だと知られている SF ネットワーク (黒点線) よりも低くなる。これは、モジュール構造がないランダム d -レギュラーグラフが悪意のある攻撃に対して最適な頑健性を持つにも関わらず [20] [53]、モジュール性を増加させることで SF ネットワークよりも脆弱になれることを意味する。上記の結果は、図 4.9、4.10 のように、次数 d を変えても同様に得られる。既に SF ネットワークモデルや実ネットワークでも、モジュール性の増加に伴って IB 攻撃および次数の初期値順で攻撃する方法に対して、頑健性が低下することが示されている [26]。なおかつ、 Q と R の関係ではなく、 w' に対する Q 、インターリンク数の比率 (IR)、 w を表 4.3-4.11 に見せる。

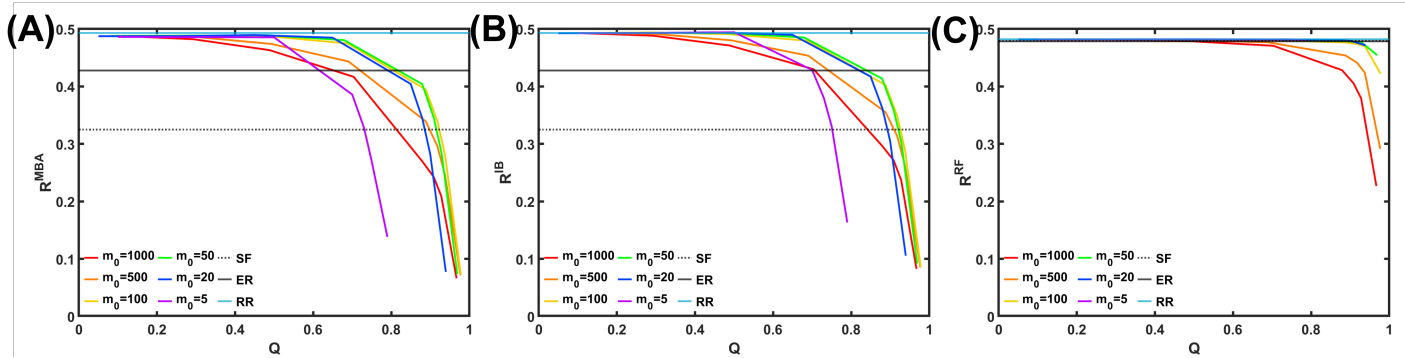


図 4.8: 3 種類の攻撃に対する頑健性指標 R とモジュール性 Q 間の関係。次数 d は 9 である。(A) MB 攻撃に対する頑健性 R_{MB} 、(B) IB 攻撃に対する頑健性 R_{IB} 、(C) RF に対する頑健性 R_{RF} 。線色はモジュール数 m_o に対応する。黒線、黒点線、シアン色線は、各自モジュール構造がない、ER ランダムグラフ、SF ネットワークモデル、ランダム 9-レギュラーグラフをそれぞれ指す。

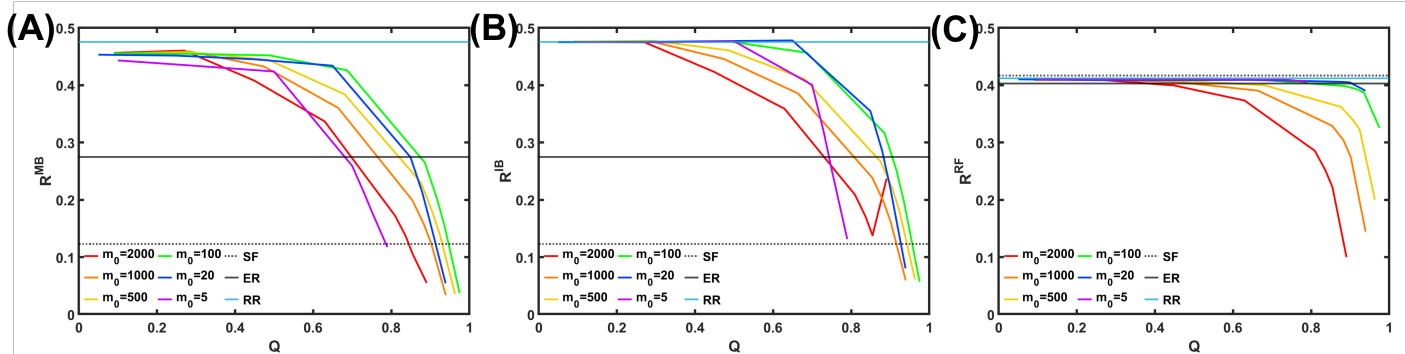


図 4.9: 3 種類の攻撃に対する頑健性指標 R とモジュール性 Q 間の関係。次数 d は 4 である。(A) MB 攻撃に対する頑健性 R_{MB} 、(B) IB 攻撃に対する頑健性 R_{IB} 、(C) RF に対する頑健性 R_{RF} 。線色はモジュール数 m_o に対応する。黒線、黒点線、シアン色線は、各自モジュール構造がない、ER ランダムグラフ、SF ネットワークモデル、ランダム 4-レギュラーグラフをそれぞれ指す。

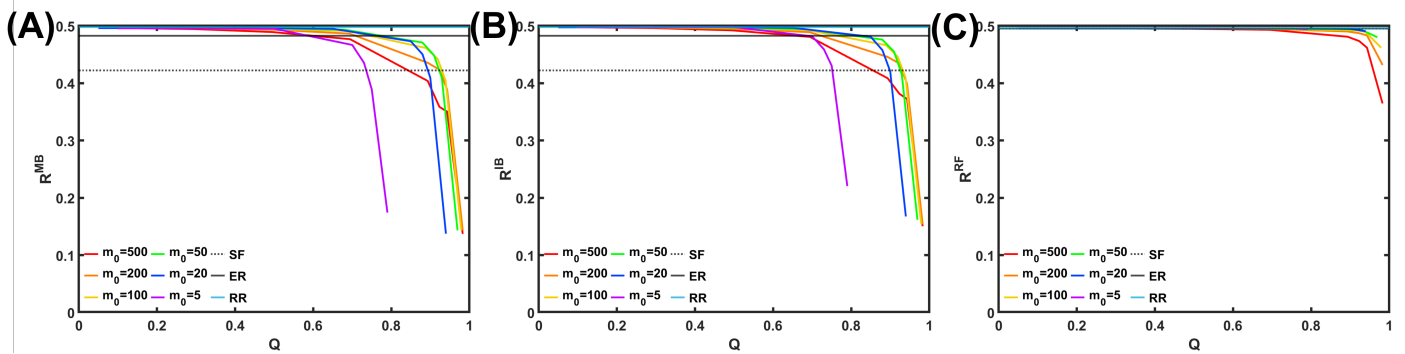


図 4.10: 3 種類の攻撃に対する頑健性指標 R とモジュール性 Q 間の関係。次数 d は 19 である。(A) MB 攻撃に対する頑健性 R_{MB} 、(B) IB 攻撃に対する頑健性 R_{IB} 、(C) RF に対する頑健性 R_{RF} 。線の色はモジュール数 m_o に対応する。黒線、黒点線、シアン色線は、各自モジュール構造がない、ER ランダムグラフ、SF ネットワークモデル、ランダム 19-レギュラーグラフをそれぞれ指す。

表 4.3: $d = 4$ と $m_o = 2000$ ネットワーク上で、 w' に対する Q 、IR(全体リンク数に対するインターリンク数の比率)、従来のモジュール化における w を示す。

	$w'=0.01$	$w'=0.05$	$w'=0.1$	$w'=0.3$	$w'=0.5$	$w'=0.7$
Q	0.891	0.855	0.810	0.627	0.450	0.270
IR	0.109	0.145	0.190	0.370	0.550	0.730
w	0.891	0.855	0.810	0.630	0.450	0.270

表 4.4: $d = 4$ と $m_o = 100$ ネットワーク上で、 w' に対する Q 、IR(全体リンク数に対するインターリンク数の比率)、従来のモジュール化における w を示す。

	$w'=0.01$	$w'=0.05$	$w'=0.1$	$w'=0.3$	$w'=0.5$	$w'=0.7$
Q	0.975	0.935	0.886	0.687	0.488	0.289
IR	0.015	0.055	0.105	0.304	0.503	0.702
w	0.985	0.945	0.894	0.693	0.492	0.291

表 4.5: $d = 4$ と $m_o = 5$ ネットワーク上で、 w' に対する Q 、IR(全体リンク数に対するインターリンク数の比率)、従来のモジュール化における w を示す。

	$w'=0.01$	$w'=0.05$	$w'=0.1$	$w'=0.3$	$w'=0.5$	$w'=0.7$
Q	0.790	0.750	0.700	0.500	0.300	0.100
IR	0.010	0.050	0.100	0.300	0.500	0.700
w	0.987	0.937	0.875	0.625	0.375	0.125

表 4.6: $d = 9$ と $m_o = 1000$ ネットワーク上で、 w' に対する Q 、IR(全体リンク数に対するインターリンク数の比率)、従来のモジュール化における w を示す。

	$w'=0.01$	$w'=0.05$	$w'=0.1$	$w'=0.3$	$w'=0.5$	$w'=0.7$
Q	0.967	0.928	0.879	0.704	0.488	0.292
IR	0.032	0.071	0.120	0.316	0.511	0.707
w	0.968	0.929	0.880	0.684	0.488	0.293

表 4.7: $d = 9$ と $m_o = 100$ ネットワーク上で、 w' に対する Q 、IR(全体リンク数に対するインターリンク数の比率)、従来のモジュール化における w を示す。

	$w'=0.01$	$w'=0.05$	$w'=0.1$	$w'=0.3$	$w'=0.5$	$w'=0.7$
Q	0.978	0.938	0.888	0.688	0.489	0.289
IR	0.012	0.052	0.102	0.302	0.501	0.701
w	0.988	0.947	0.897	0.695	0.494	0.292

表 4.8: $d = 9$ と $m_o = 5$ ネットワーク上で、 w' に対する Q 、IR(全体リンク数に対するインターリンク数の比率)、従来のモジュール化における w を示す。

	$w'=0.01$	$w'=0.05$	$w'=0.1$	$w'=0.3$	$w'=0.5$	$w'=0.7$
Q	0.790	0.750	0.700	0.500	0.300	0.100
IR	0.010	0.050	0.100	0.300	0.500	0.700
w	0.987	0.937	0.875	0.625	0.375	0.125

表 4.9: $d = 19$ と $m_o = 500$ ネットワーク上で、 w' に対する Q 、IR(全体リンク数に対するインターリンク数の比率)、従来のモジュール化における w を示す。

	$w'=0.01$	$w'=0.05$	$w'=0.1$	$w'=0.3$	$w'=0.5$	$w'=0.7$
Q	0.983	0.943	0.893	0.694	0.495	0.296
IR	0.015	0.055	0.105	0.304	0.503	0.702
w	0.985	0.945	0.895	0.696	0.496	0.297

表 4.10: $d = 19$ と $m_o = 100$ ネットワーク上で、 w' に対する Q 、IR(全体リンク数に対するインターリンク数の比率)、従来のモジュール化における w を示す。

	$w'=0.01$	$w'=0.05$	$w'=0.1$	$w'=0.3$	$w'=0.5$	$w'=0.7$
Q	0.979	0.939	0.889	0.689	0.489	0.290
IR	0.011	0.051	0.101	0.301	0.501	0.700
w	0.989	0.948	0.898	0.696	0.494	0.293

表 4.11: $d = 19$ と $m_o = 5$ ネットワーク上で、 w' に対する Q 、IR(全体リンク数に対するインターリンク数の比率)、従来のモジュール化における w を示す。

	$w'=0.01$	$w'=0.05$	$w'=0.1$	$w'=0.3$	$w'=0.5$	$w'=0.7$
Q	0.790	0.750	0.700	0.500	0.300	0.100
IR	0.010	0.050	0.100	0.300	0.500	0.700
w	0.987	0.937	0.875	0.625	0.375	0.125

4.3.2 反モジュール化による効率性の向上

次に、反モジュール化したネットワークは、高い頑健性だけでなく、高い効率性も持つことを示す。図 4.11A は、 $d = 4$ 、 $N_m = 5$ の場合 (穴居人グラフ [57])、ネットワークのサイズ N と平均最短経路長 L の関係を示す。リング構造 ($w' = 0$) では、 L は $O(N)$ である (黒線)。しかしながら、ごく一部のイントラリンクをリワイヤするだけで ($w' = 0.1$)、 L は $O(\log N)$ になる (黄色線)。図 4.11B、C、そして D を見ると、 N_m を変えても反モジュール化したネットワークは、SW 特性を持つことが分かる。同様の結果が、 $d = 9$ 、 19 の結果 (図 4.12、4.13) でも得られている。これらの結果は、図 4.11A の穴居人グラフの頑健性も含まれている。なぜなら、クリークはレギュラーグラフの最も密集したケースだからである。

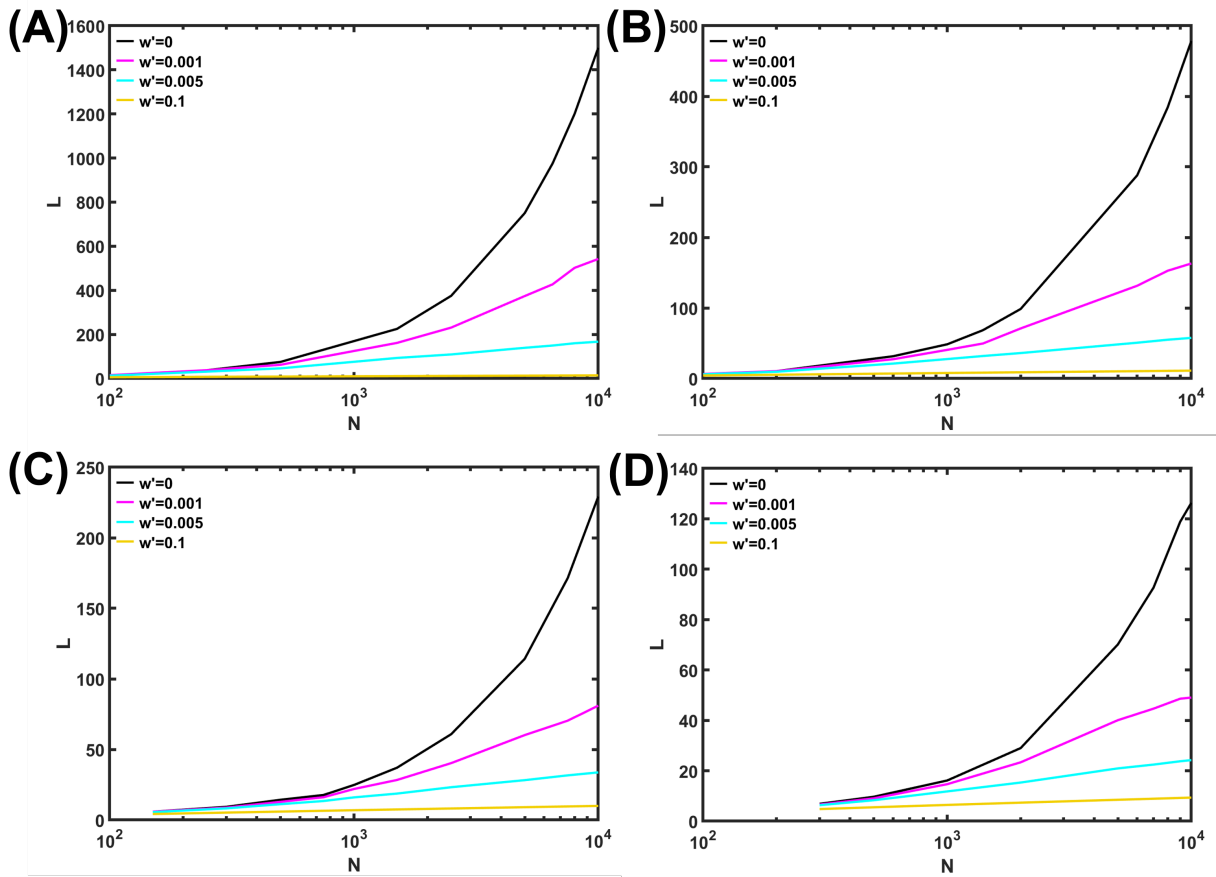


図 4.11: 平均最短経路長 L ネットワークサイズ N の関係。 $d = 4$ のネットワークで、モジュールサイズは (A) $N_m = 5$ 、(B) $N_m = 20$ 、(C) $N_m = 50$ 、(D) $N_m = 100$ である。線色は反モジュール化における、リワイヤするイントラリンク数の比率 w' に対応する。

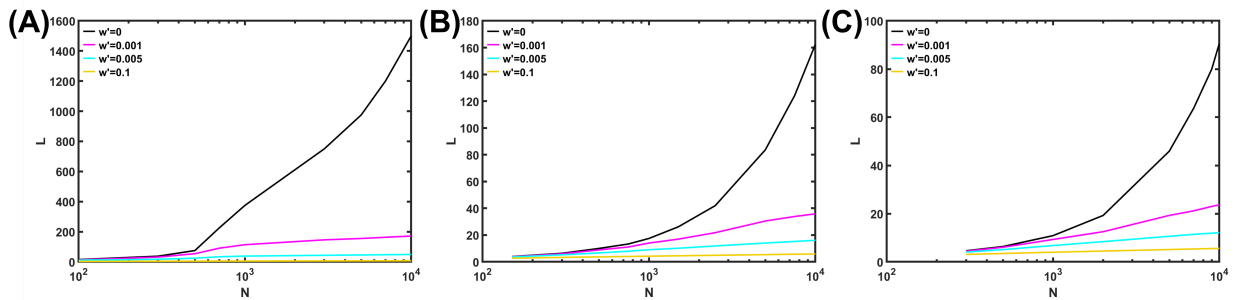


図 4.12: 平均最短経路長 L ネットワークサイズ N の関係。 $d = 9$ のネットワークで、モジュールサイズは (A) $N_m = 5$ 、(B) $N_m = 20$ 、(C) $N_m = 50$ 、(D) $N_m = 100$ である。線色は反モジュール化における、リワイヤするイントラリンク数の比率 w' に対応する。

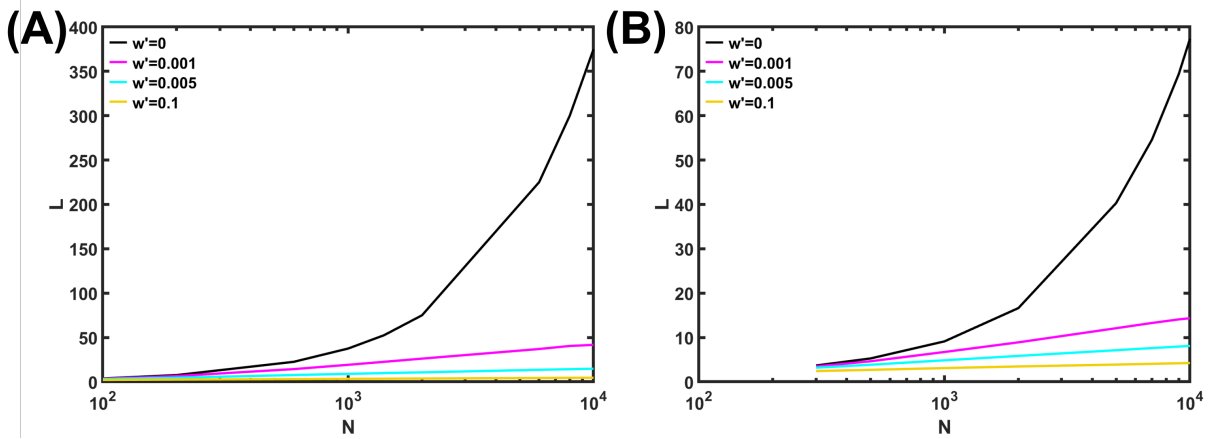


図 4.13: 平均最短経路長 L ネットワークサイズ N の関係。 $d = 19$ のネットワークで、モジュールサイズは (A) $N_m = 5$ 、(B) $N_m = 20$ 、(C) $N_m = 50$ 、(D) $N_m = 100$ である。線色は反モジュール化における、リワイヤするイントラリンク数の比率 w' に対応する。

4.3.3 従来のモジュール化の修正

従来のモジュール化 [26] は、次数分布と全体の連結性を維持しないため、その修正版を提案する。図 4.14A、B、および C は、従来のモジュール化 [26] を $d = 9$ 、 $m_o = 100$ のネットワークに適用した後の次数分布が幅を持つことを示している。異なる d および m_o のネットワークでも、次数分布に幅が生じることを確認している。これは、従来のモジュール化 [26] でリワイヤされたネットワークは、もはやランダム d -レギュラーグラフではないことを意味する。

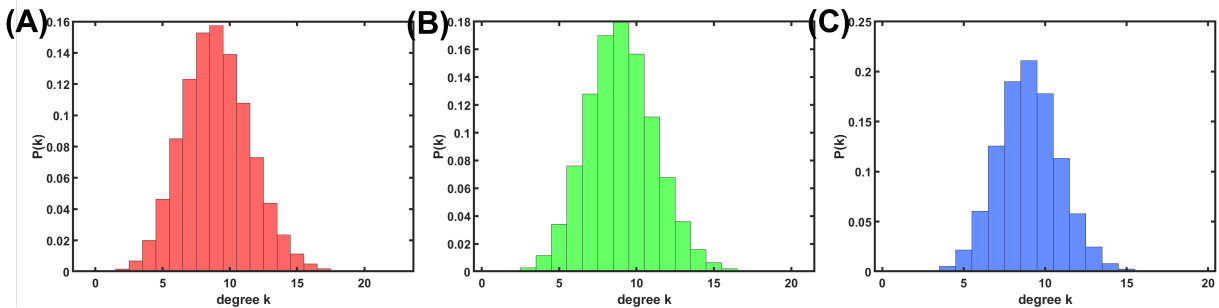


図 4.14: 従来のモジュール化法を適用したネットワーク ($d = 9$ 、 $m_o = 100$) の次数分布。モジュール化においてリワイヤするインターリンク数の比率 w は (A) $w = 0.988$ 、(B) $w = 0.695$ 、(C) $w = 0.494$ である。各 w は式 4.1 により $w' = 0.01, 0.3, 0.5$ に対応する値である。

したがって、 d -レギュラーグラフの構造を維持するために、モジュール化のプロセス [26] を修正する。基本的なアイデアは、先の反モジュール化と似ている。図 4.15 は修正したプロセスである。初めに、モジュール番号 $1, 2, \dots, m_o$ がランダムにノードに割り当てられ、各モジュールのサイズ N_m は一定である。(1) あるインターリンク (i, j) がランダムに削除される。そして、削除されたリンクの末端ノード j がランダムに選ばれる。(2) ノード j と同じモジュールに属し、ノード j と繋がっていないノード k がランダムに選択される。そして、ノード j と k の間に新しいイントラリンクを生成する。(3) ノードの次数を維持するために、インターリンク (k, l) をランダムに削除する。(4) ノード l と同じモジュールにあるノード m をランダムに選択する。そして、ノード l と m の間に新しいイントラリンクを生成する。(5) インターリンク (m, n) がランダムに削除される。(5) の後、ノード n は次のリワイヤリングの対象となる。このようなプロセスを、連結成分やノードが孤立することなく $w(M - M/m_o)$ 個のイントラリンクが生成されるまで繰り返

す。しかしながら、繰り返しのプロセスの終わりには、最初に選択されたノード i の次数は 1 が減少する。一方で、最後に選択されたノード p の次数は 1 が増加する。

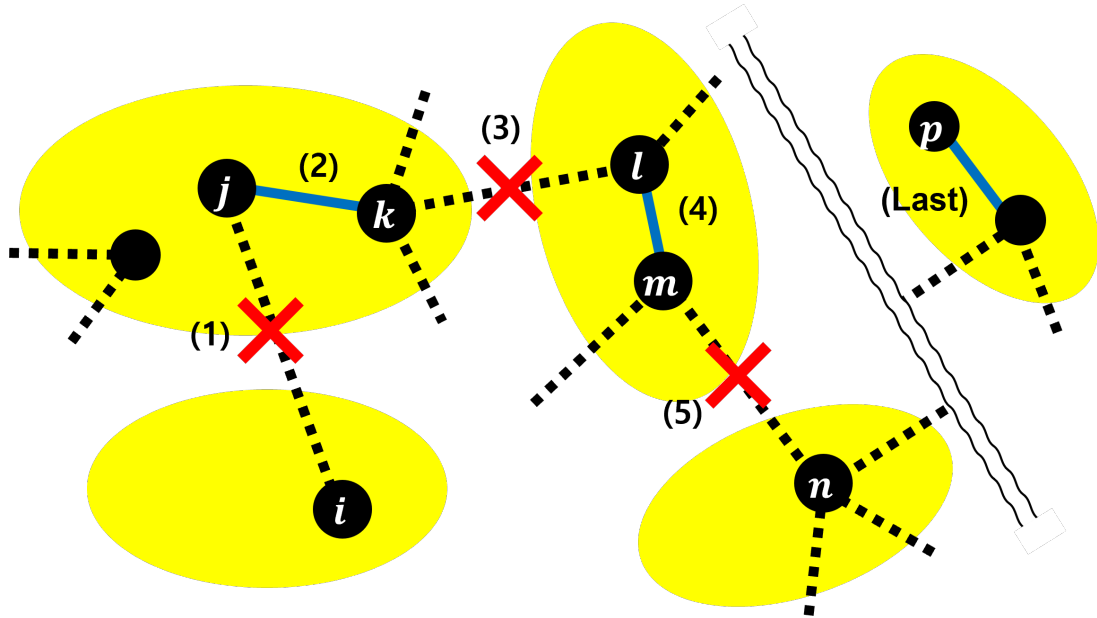


図 4.15: 修正したモジュール化のプロセス。格好中の数字はプロセスの各ステップである。黄色の円はモジュールである。黒点線と青色線は、各自インターリンクとイントラリンクを指す。

式 (1) で対応付けられたリワイヤ率 w と w' において、図 4.16A から、リワイヤされた $m_o = 100$ のネットワーク上で、修正したモジュール化 (点線) と反モジュール化 (実線) の結果がほぼ一致することが分かる。しかしながら、 m_o が小さいほど、実線と点線の間幅は大きくなる。特に $m_o = 5$ の図 4.16C のように、オレンジ色と黄色の線の幅が顕著である。一方、修正したモジュール化における高い w でリワイヤされたネットワークはリング構造になりにくい、図 4.16A、B、C の赤色線はほぼ同一となり、リング構造の影響は見られない。さらに、修正したモジュール化 (点線) と反モジュール化 (実線) の結果は、図 4.174.18 のように、IB 攻撃と RF の場合についても、上記と同様な結果が得られる。

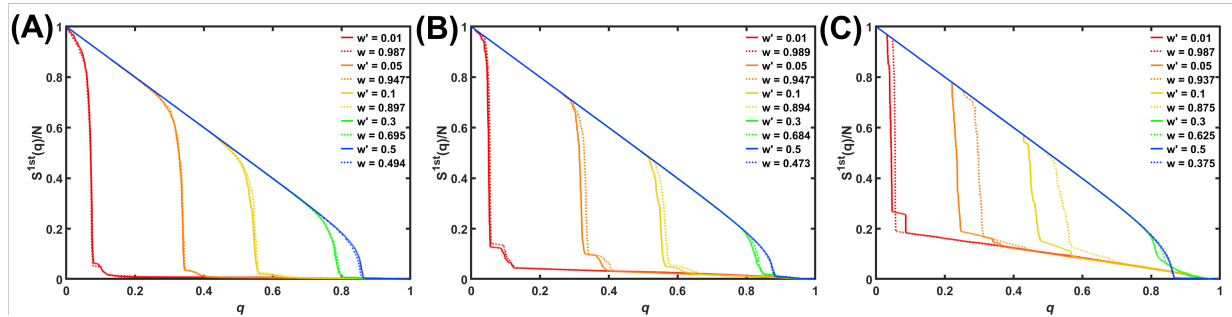


図 4.16: 修正したモジュール化と反モジュール化を適用したネットワークの、MB 攻撃に対する頑健性比較。MB 攻撃後の最大連結成分のサイズ比率は、 $d = 9$ でモジュール数 m_o が (A) $m_o = 100$ 、(B) $m_o = 20$ 、(C) $m_o = 5$ であるネットワークで測定した。点線と実線は、修正されたモジュール化における w でのリワイヤリングによる結果と反モジュール化における w' でのリワイヤリングによる結果を表す。

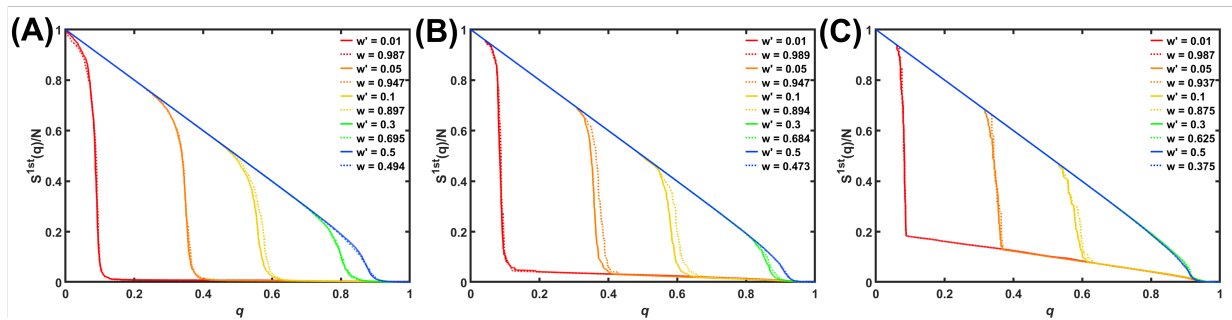


図 4.17: 修正したモジュール化と反モジュール化を適用したネットワークの、IB 攻撃に対する頑健性比較。MB 攻撃後の最大連結成分のサイズ比率は、 $d = 9$ でモジュール数 m_o が (A) $m_o = 100$ 、(B) $m_o = 20$ 、(C) $m_o = 5$ であるネットワークで測定した。点線と実線は、修正されたモジュール化における w でのリワイヤリングによる結果と反モジュール化における w' でのリワイヤリングによる結果を表す。

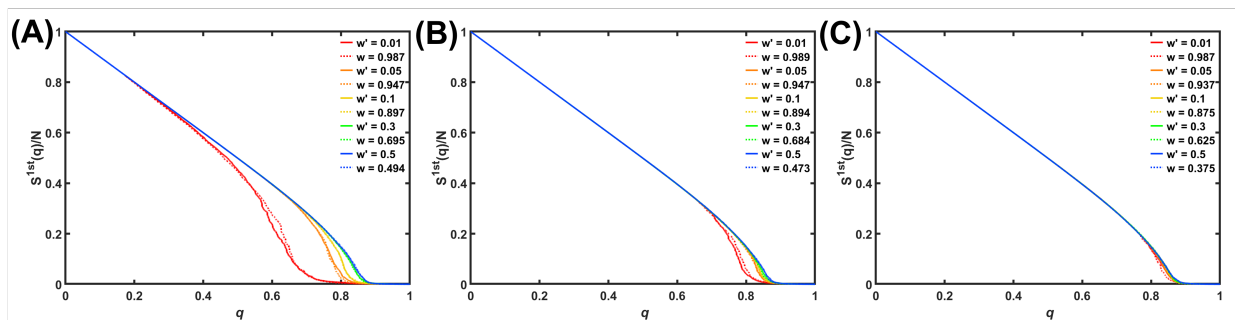


図 4.18: 修正したモジュール化と反モジュール化を適用したネットワークの、RF に対する頑健性比較。MB 攻撃後の最大連結成分のサイズ比率は、 $d = 9$ でモジュール数 m_o が (A) $m_o = 100$ 、(B) $m_o = 20$ 、(C) $m_o = 5$ であるネットワークで測定した。点線と実線は、修正されたモジュール化における w でのリワイヤリングによる結果と反モジュール化における w' でのリワイヤリングによる結果を表す。

4.4 まとめ

モジュラー d -レギュラーグラフの頑健性について調べた。予想通り、反モジュール化によりリワイヤされたイントラリンクの数が増えるほど、MB、IB 攻撃、そして RF に対する頑健性が向上する。言い換えれば、モジュール性が高くなるにつれて、MB 攻撃に対する頑健性が IB 攻撃に比べてより低く減少する。特に、モジュール性が 0.8 を超えると、モジュラー d -レギュラーグラフの頑健性は、非常に脆弱な SF ネットワークの頑健性よりも低くなる。さらに、反モジュール化でリワイヤされたネットワークでは、高い頑健性と経路効率が共存できる。実際、穴居人グラフ [57] のように、モジュラー d -レギュラーグラフは少数のイントラリンク ($w' = 0.1$) をリワイヤするだけで SW 特性を持つ。しかしながら、 $w' = 0.1$ では、モジュラー d -レギュラーグラフはまだ低い頑健性を持ったままである。したがって、高い頑健性と経路効率を同時に持つためには、少なくとも $0.1 < w' < 0.3$ の範囲でより多くのリワイヤリングが必要だと言える。

一方、低モジュール性を持つモジュラー d -レギュラーグラフは、偶発的な故障の RF よりも、選択的な IB 攻撃に対してより頑健となることを発見した。修正したモジュール化でリワイヤされたネットワークの頑健性は、その逆プロセスである反モジュール化によるものとほぼ同一であることも確認した。

しかしながら、私達の結果にはまだ明らかでない理由がいくつか残っている。特定のモジュール数を持つリワイヤされたネットワークの頑健性の結果は、一部が他の結果と異なる。例えば、 $d = 4$ と最大モジュール数での図 4.3A に示されたように、 $w' = 0.01$ のネットワークは $w' = 0.1$ のものよりも IB 攻撃に対する頑健性が高い。最大連結成分に関する図 4.3、4.2、4.4 の $m_o = 5$ では、ノード削除の攻撃率 $q \simeq 0.85, 0.7$ 、および 0.92 で $w' = 0.9$ と $w' = 0.1$ の線が交差する。このような現象についてはさらなる分析が必要である。また、他の攻撃、例えば、リンク攻撃 [59]、他のノード中心性に基づく攻撃 [60]、CoreHD [50]、集団影響 [51]、信念伝搬 [52]、または空間攻撃 [61] に対する頑健性についても今後の研究課題となる。

第5章

頑健性と負荷分散を考慮したデータセンター配置構成

5.1 データセンターと利用者両方に安全な配置

現在、インターネット上のクラウド環境を通じて、データセンターへの便利な遠隔接続が可能になっている。しかしながら、自然災害や人為的な災害の増加はデータセンターに脅威をもたらし、利用者はデータセンターのシャットダウン、もしくはより長い接続経路を迂回することにより、低い QoS に直面する可能性が日々高くなっている。ネットワークが、利用者への接続性を持続的に保持するために、レジリエンスが主に通信分野で多くの注目を集めている [62,63]。ここで、レジリエンスは「ネットワークが様々な障害や、通常の運用を続けることに対して、サービスを提供し続ける能力」と定義されている [62]。この定義に従い、本章では、データセンターと利用者で構成されるデータセンターネットワークにおいて、災害にも利用者ができるだけ短い接続経路を持つようなデータセンター配置法について議論する。

一方、データセンターが耐えるべき負荷を均等に分散させることは、運用コストを減らし、データへの効率的な接続妨害をするボトルネックを予防するために重要である。しかしながら、レジリエンスと負荷分散の重要性にもかかわらず、両方を同時に考慮するデータセンターの配置法は稀である。これに対する既存研究としては、グラフ理論の支配集合問題を用いて、影響力があるノードを見つけるものが少なからずある [64–67]。

ここで、支配集合の定義は、ノード集合 V の部分集合 $S \subset V$ で、任意のノード $u \in V \setminus S$ が $s \in S$ の隣接ノードである集合である。 s をデータセンター、 u を利用者だとすると、センターネットワークは利用者に対して、1 ホップ (リンク 1 つ) の接続経路を保証する。

しかしながら、実ネットワークでは常に資源が限られているため、最小限のデータセンターを設置しながら、利用者の接続経路ができるだけ短いようにする必要がある。したがって、支配集合を拡張した、 d ホップ支配集合を考える。その定義はノード集合 V の部分集合 $S \subset V$ で、任意のノード $u \in V \setminus S$ が $s \in S$ の d -ホップ隣接である集合である。[64]。但し、最小集合の d ホップ支配集合を探すことは NP 困難であるため [64]、多くのヒューリスティック手法 [64–67] が提案されている。しかしながら、これらの研究は攻撃に対するレジリエンスまたは負荷分散のいずれかだけを考慮している。例えば、度数に基づく方法 [64] が知られているが、特定のノードに、利用者の接続が集中されて、他のノードより高い負荷を持つ。

本章では、実ネットワークのトポロジーを使用して、災害にも強い接続性とデータセンターが持つ負荷のバランスの両方を考慮して、データセンターを配置する方法を頂点被覆問題として近似的に解く。支配集合問題は頂点被覆問題に還元できる。

データセンターの配置は、NP 困難であるため以下の 2 種類の近似法を考慮する。 d ホップ支配集合を用いた貪欲ヒューリスティック法 [36, 37]、統計物理アプローチ [38] とマルチホップカバーを組み合わせた方法 [39] である。

従来の 2 近似法 [68] では、あるリンクの末端ノードを、ネットワークをカバーするノードで選ぶことを繰り返して、支配集合を求める。求められた支配集合のサイズは最適集合サイズの精々 2 倍である。一方、[38] の方法では、統計物理学の Cavity 法に基づく。そこで、各ノードはカバー、ノンカバー、カバーあるいはノンカバーのいずれかのジョーカー、この 3 つの状態のいずれかの 1 つを持つと設定し、その自由エネルギー最小化に基づいて支配集合を探す。ジョーカー状態を持つノードは、支配集合の探索中で自由にカバーやノンカバーになれるので、より幅広い探索ができる。結果的に求められた支配集合は、2 近似法 [68] で求めた支配集合サイズの半分である。さらに、[38] で選択されたノードは、影響力のあるノード [39] を見つけるための従来の方法 [60, 69–73] で選択されたノードより強い拡散力を持っていることが報告されている。このような効果的な方法 [65] を、従来の貪欲ヒューリスティック方法 [36, 37] と比較する。両方の詳細なプロセスは 4.2 節で説明する。

5.2 配置構成を求める近似法

5.2.1 貪欲ヒューリスティック法 (Greedy Heuristic、GH)

Greedy Heuristic 方法 [36, 37](以下、GH 法) の基本的なメカニズムは、多くの d ホップ隣接ノードをカバーできるノードをデータセンターとしてヒューリスティックに選択することである。各ノードは、センター、カバー済み、ノンカバー、これら 3 つの状態のいずれかを持つものとする。初期状態では、全てのノードはノンカバーである。また、ノード i の T_i を、ノード i から d ホップ以内にあるノンカバーのノード数と定義する。GH 法は以下のようなものである。

- ステップ1 各ノード $j \in V$ の T_j を計算する。最も高い T_i を持つノード i をセンターとして選択する。
- ステップ2 選ばれたノードの状態をノンカバーからセンターに変更し、その d ホップ隣接ノードの状態もノンカバーからカバー済みに変更する。
- ステップ3 ステップ1と2を繰り返し、ネットワーク内にノンカバーのノードがなくなるまで実行する。

5.2.2 マルチホップカバー方法 (Multi-Hop、MH)

情報拡散にとって重要な種ノードを見つけるために、統計物理に基づいた方法 [38] から着想を得た効果的な方法 [39] が提案されている。マルチホップカバー方法 [39](以下、MH 法) は、お互い拡散する領域が重ならないように、マルチホップカバーを基本メカニズムとしている。方法 [38] で最小頂点被覆問題を解くために、ノードはカバー (0 で表記)、ノンカバー (1)、カバーあるいはノンカバーのいずれかのジョーカー (*) として、この 3 つの状態のいずれかの 1 つを持つものとする。各状態の確率 π^1, π^0, π^* は、次のメッセージ伝搬式 (5.1)~(5.3) を用いて計算する。

$$\pi_i^0 = C_i^{-1} \prod_{j \in \partial i} (1 - \pi_{j \rightarrow i}^0), \quad (5.1)$$

$$\pi_i^* = C_i^{-1} e^{-y} \sum_{j \in \partial i} \pi_{j \rightarrow i}^0 \prod_{j' \in \partial i \setminus j} (1 - \pi_{j' \rightarrow i}^0), \quad (5.2)$$

$$\pi_i^1 = C_i^{-1} e^{-y} \left[1 - \prod_{j \in \partial i} (1 - \pi_{j \rightarrow i}^0) - \sum_{j \in \partial i} \pi_{j \rightarrow i}^0 \prod_{j' \in \partial i \setminus j} (1 - \pi_{j' \rightarrow i}^0) \right], \quad (5.3)$$

ここで、 $\partial i \setminus j$ はノード j を除いたノード i の最近接ノード集合であり、 e^{-y} は種ノード数を最小化するためのペナルティで、 y は温度の逆パラメータである。 C_i は正規化 $\pi_i^0 + \pi_i^1 + \pi_i^* = 1$ のためのものです。

$$C_i = e^{-y} \left[1 - (1 - e^{-y} \prod_{j \in \partial i} (1 - \pi_{j \rightarrow i}^0)) \right]. \quad (5.4)$$

同様に、各リンク $i \rightarrow k$ において、ノード i がカバー状態である確率を以下から計算する。

$$\pi_{i \rightarrow k}^0 = C_{i \rightarrow k}^{-1} \prod_{j \in \partial i \setminus k} (1 - \pi_{j \rightarrow i}^0), \quad (5.5)$$

$$C_{i \rightarrow k} = e^{-y} \left[1 - (1 - e^{-y}) \prod_{j \in \partial i \setminus k} (1 - \pi_{j \rightarrow i}^0) \right]. \quad (5.6)$$

π^0 の初期値は 0 から 1 の間で一様乱数値で設定される。メッセージ伝搬を時刻にてラウンド数 k だけ繰り返す。各ラウンドでは、 N ノードの確率をランダム順に更新する。パラメータ数値は $y = 7$ および $k = 10$ で設定する。この近似方法は、次のようにマルチホップカバーと組み合わせさせる。

- ステップ1 最も高い π_i^1 を持つノード i をセンターとして選択する。
- ステップ2 ネットワークからノード i およびその d ホップ隣接ノードを削除する。
- ステップ3 残っている各ノードの 3 つの確率を再計算し、ネットワークにノードがなくなるまでステップ 1 と 2 を繰り返す。

5.3 求められた配置構成の評価

5.3.1 評価ための初期設定と評価指標

データセンターの負荷および、データセンターと利用者間の最短距離を評価指標として比較する。センターの負荷は、そこに接続している利用者数として定義し、全てのデータセンターが全てのデータを持っていると仮定する。したがって、利用者は最も近いデータ

センターにのみ接続する。利用者は長くても2もしくは3ホップでデータセンターへ接続できるようにする。このために、従来のGH法とMH法によってデータセンターの配置を決定する。さらに、データセンターに対する攻撃シナリオを考慮する。ネットワークに最大のダメージを与えるための攻撃として、攻撃者は最も負荷が高いセンターから意図的に破壊するものとする。攻撃によりいくつかのセンターが機能停止すると、それらに接続していた利用者はより長い接続経路を持つことになる。言い換えると、ある一部の利用者はデータセンターの2もしくは3ホップのカバー範囲に入れないかも知れない。利用者が再びカバーされるためには、構築コストを考慮せずに、両配置法で代替センターを生き残ったノードに設置する方法を考慮する。

攻撃前後のネットワークにおいて、両配置法で作られたセンター構成に対して、データセンターの負荷と距離を測って、1000回の平均値を求める。但し、MH法で得られるセンター構成は、確率 π^1, π^0, π^* の初期値に依存するため毎度異なる。従来のGH法で得られるセンター構成は、ノンカバーのノード数 T が同値なら、その中でランダムに選択するので、構成が僅かに異なる場合がある。

SNDlib [74] から引用した5種類の実ネットワークについてシミュレーションを実行する。実ネットワークの基本特性を表5.1に示す。 N と E は、ネットワークのノード数とリンク数である。 k_{min} と k_{max} は最小次数と最大次数を指す。Cost266から得られた典型的な結果を、以下の4.3.2節、4.3.3節で説明する。

表 5.1: 実ネットワークの基本特性

ネットワーク サンプル	性質				
	N	E	k_{min}	k_{max}	直径
Cost266	37	57	2	5	8
Germany50	50	87	2	5	9
Janos-us-ca	39	61	2	5	10
Coronet CONUS	75	99	2	5	17
India35	35	80	2	9	7

5.3.2 攻撃以前の結果比較

図 5.1 は、5つのデータセンターが各自 2 ホップ以内にある利用者をカバーするために配置された場合、各ノードがデータセンターとして選ばれた頻度を示す。色のグラデーションで、赤色は頻繁に選ばれることを意味し、青色はあまり選ばれないことを意味する。白色のノードは一度も選ばれたことがないことを指す。

図 5.1A のように、GH 法ではほとんどの場合、データセンターとして 3 つのノード（ベルリン、パリ、グラスゴー）が選ばれる。次によく選ばれたノードは、4 つの緑色のノード（アテネ、バルセロナ、セビリア、ザグレブ）である。一方、MH 法では必ずベルリンだけがいつでも選択される。しかしながら、3 つのオレンジ色のノードと 1 つの黄色のノードが頻繁に選択されるにもかかわらず、多くの青色のノードも存在する（図 5.1B）。これは、MH 法により得られるセンター構成では、選択配置されるノード候補の種類が多いことを意味する。MH 法で、各ノードが持つ確率 π^1, π^0, π^* は初期値に依存するからである。MH 法で得られる多数の可能解は、システム保安に利用できる。例えば、従来の GH 法では高次数のノードがデータセンターとして頻繁に選ばれるため、敵対的な外部者が攻撃対象を容易に特定できる。しかしながら、MH 法ではどのノードもデータセンターになりうるため、例えトポロジー情報を知っても攻撃対象を特定することが困難である。

表 5.2 は、Cost266 におけるデータセンターと利用者間の最短距離を示す。5、そして 3 は、両方の方法で 2 ホップもしくは 3 ホップ以内にカバーするために必要なデータセンターの最小数である。最小数のデータセンターを設置した場合、MH 法は GH 法（2 ホップ以内：1.407、3 ホップ：1.913）よりも短い距離（2 ホップ以内：1.396、3 ホップ：1.771）となっている。

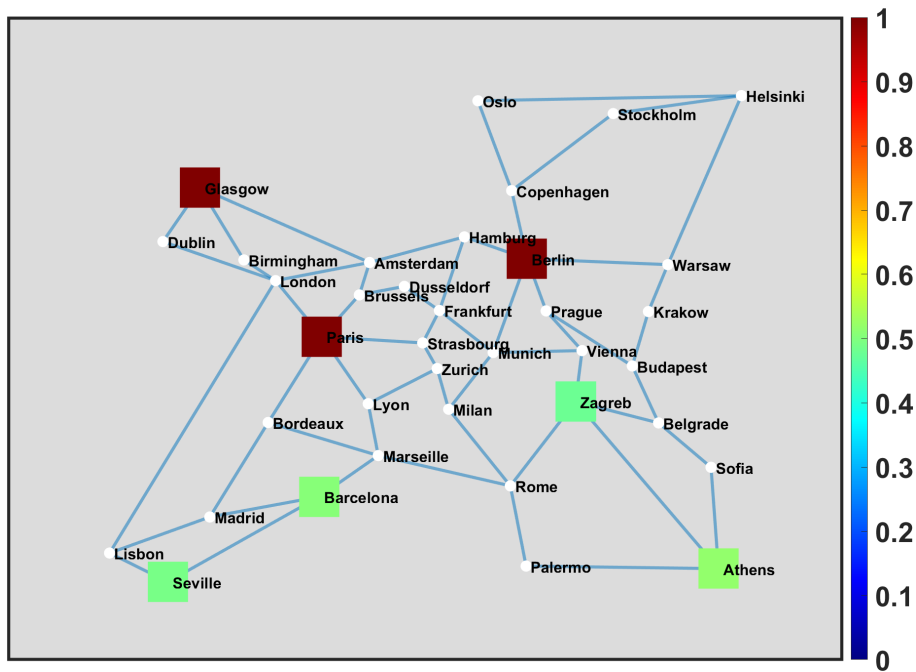
また表 5.2 に示す、空白の値は配置法がその数のデータセンター数ではセンター構成の可能解を見つけられなかったことである。例えば、MH 法で 2 ホップ以内で利用者をカバーするためには 5、6、7 のデータセンターで構成された配置の可能解を提供できる。しかしながら、GH 法は 7 データセンターの可能解だけ見つけた。この結果は、図 5.1 で示したものと関連している。MH 法はより多くの可能解を見つけられる。

図 5.2 は Cost266 における負荷の分散値を示す。青と黄色のバーは、両方で最小数のデータセンターを配置した場合の負荷の分散を表す。2 ホップ、もしくは 3 ホップをカバーする場合にも、MH 法は GH 法よりも低い分散値を持つ。特に、分散値の差は、3 ホップをカバーする場合が、2 ホップをカバーする場合よりも大きくなっている。表 5.1 にある他のネットワークに対しても、センターとして選択されるノードの頻度、平均距

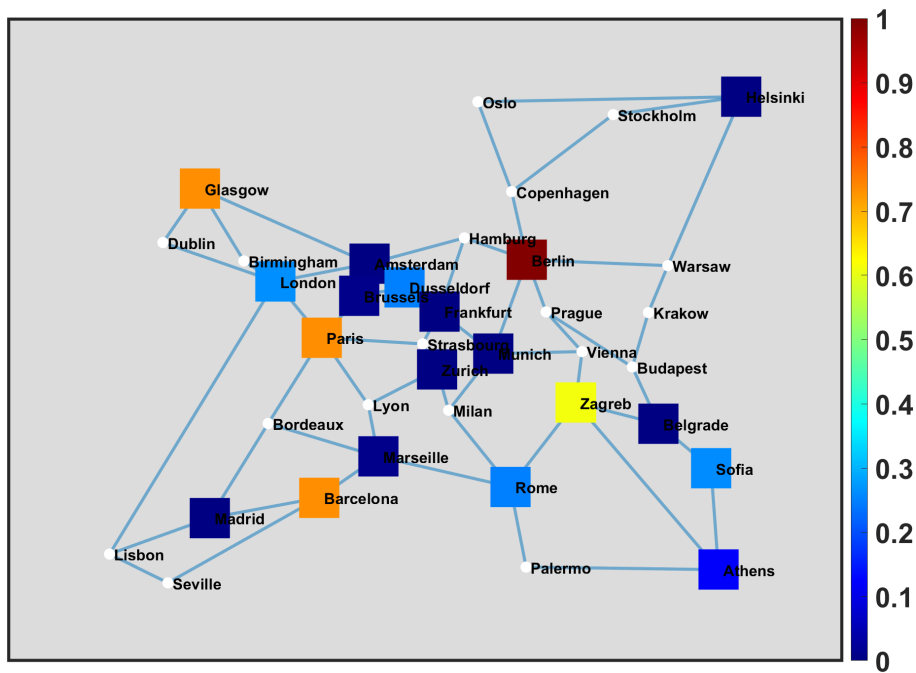
表 5.2: 最短経路の平均値

2 ホップ	5 個のデータセンター	6 個のデータセンター	7 個のデータセンター
GH	1.407	1.323	-
MH	1.396	1.295	1.333
3 ホップ	3 個のデータセンター	5 個のデータセンター	6 個のデータセンター
GH	1.913	-	-
MH	1.771	1.683	1.575

離、負荷の分散値に関して同様の結果が得られている。したがって、攻撃前のネットワークにおいて、MH 法は、従来の GH 法よりも、高い接続性と、同等な負荷のバランスを持つセンター構成を得られる。しかしながら、いくつかの例外が観察される。2 ホップ以内にカバーするために最小数のデータセンターが Janos-us-ca と India35 で配置された場合、GH 法が MH 法よりも低い分散値を持つ。



(a)



(b)

図 5.1: Cost266 で各ノードがデータセンターとして選ばれる頻度。(A)GH 法による結果、(B)MH 法による結果

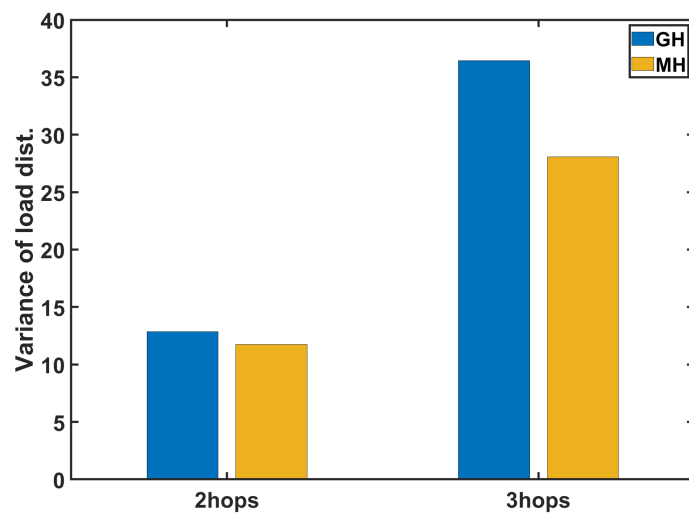


図 5.2: Cost266 に両方を用いて最小数のデータセンターが配置された場合のデータセンターが持つ負荷の分散値。

5.3.3 攻撃後の結果比較

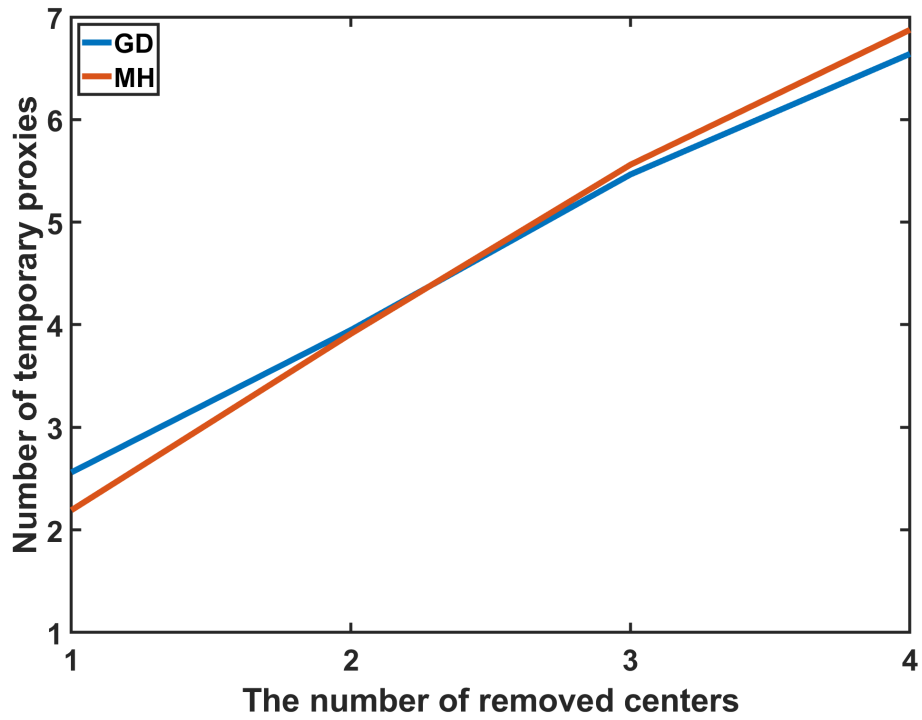
データセンターが攻撃で機能停止された時、既存の接続経路を失った利用者を再びカバーするために、代替サーバーを設置する。Fig. 5.3 は、再カバーするために必要な代替の数である。機能停止されたデータセンター数が多いほど、両方で設置する代替の数も増える (赤色、青色)。しかしながら、MH が GH より多い代替を設置するが、その差は高くても 2 である。

代替を追加したネットワークに対して、平均距離、負荷の分散、負荷の最大値の 3 つの指標を測る。図 5.4 と 5.5 のそれぞれは、最小数のデータセンターが 2 ホップまたは 3 ホップ内の利用者を再カバーした結果を示す。破線と実線は、それぞれ攻撃直後のネットワークと代替を追加したネットワークでの結果である。

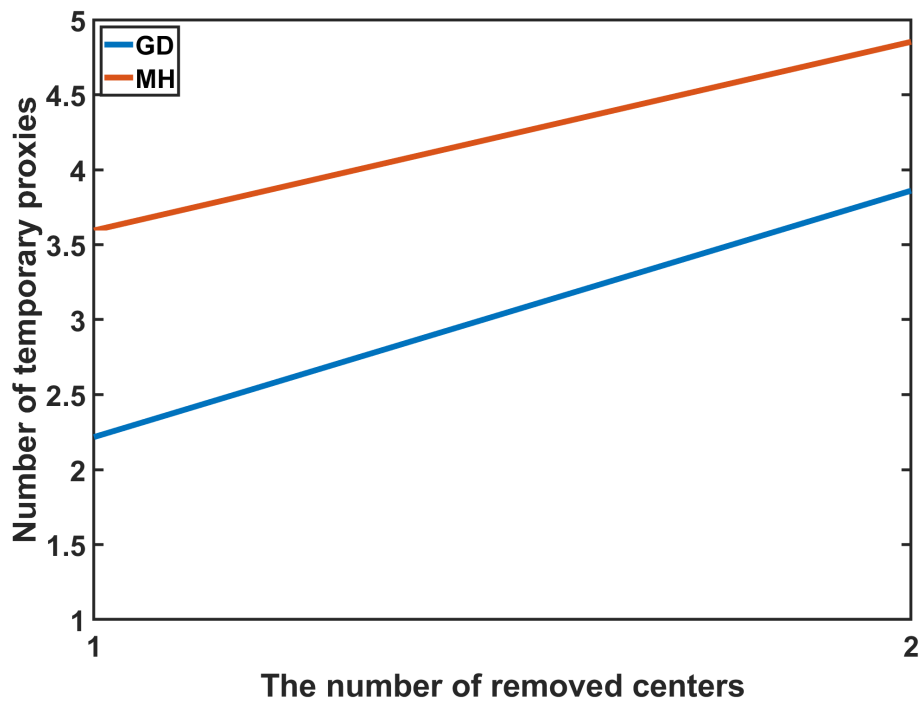
図 5.4(A) のように、両配置法 (赤と青の破線) によって得られる平均距離は、たった 1 つのデータセンターが停止しても、平均距離は 2 を超える。しかしながら、代替を追加した後、平均距離 (赤と青の実線) は大幅に減少する。その値は、表 5.2 に示されていた攻撃以前での平均距離である、1.407 と 1.396 に近づく。図 5.5(A) は、それぞれの減少した値が表 5.2 の 1.913 と 1.771 に近づく。

2 ホップカバーの場合、GH 法による分散は、MH 法によるものよりも高くなる (図 5.4(B) の赤と青の破線)。両配置法によって得られる分散値も、代替を追加することで減少する (図 5.4 と 5.5(b) の赤と青の実線)。さらに、減少した値は、攻撃前の値 (図 5.2) よりも低くなっている。最後に、代替を追加することで最大負荷値が減少することを確認した (図 5.5(c) 参照)。

表 5.1 にある他のネットワークに対しても、平均距離、分散、最大負荷値に関して、同様の結果が得られる。MH 法は、他のネットワークに対して GH 法よりも多くの代替を必要とする。例外として、Janos-us-ca と Coronet CONUS ネットワークにおいて、MH 法による分散の値は、GH 法によるものよりも高くなる。

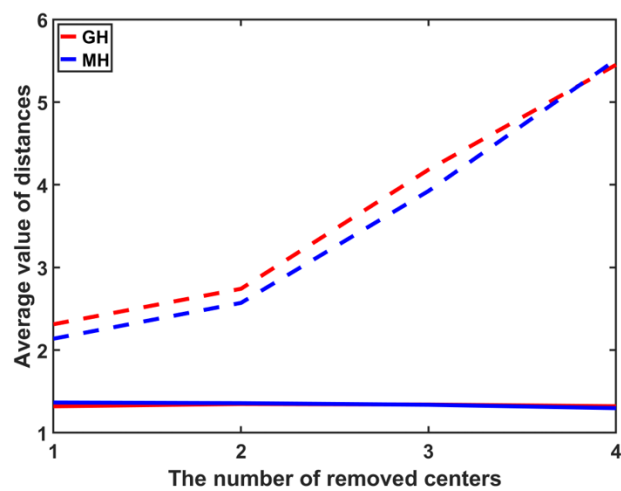


(a)

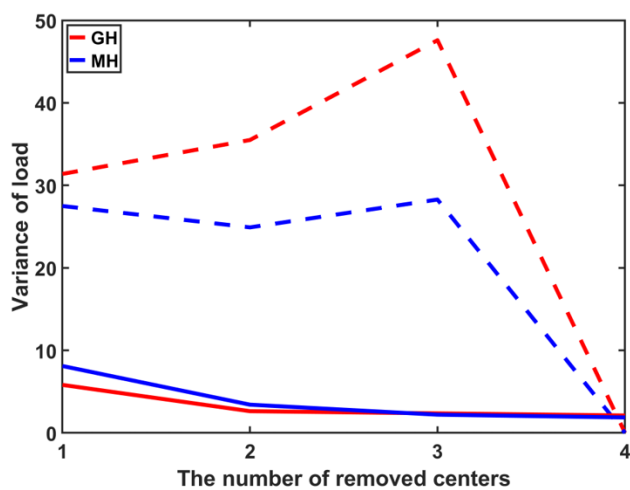


(b)

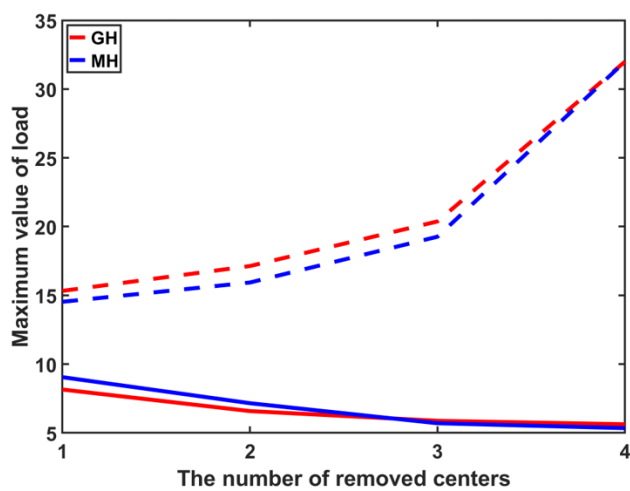
図 5.3: 各配置法で、攻撃後のネットワークに代替を設置する。(a) 2 ホップを再カバーする必要な代替数、(b) 3 ホップを再カバーする必要な代替数



(a)

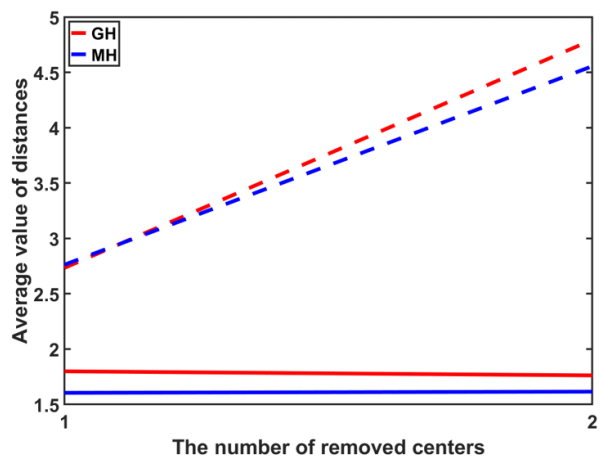


(b)

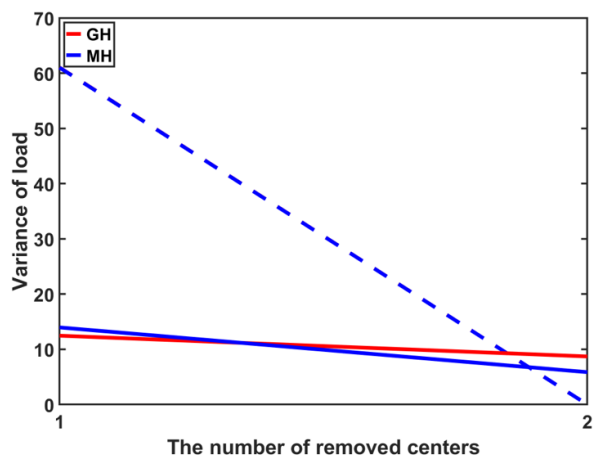


(c)

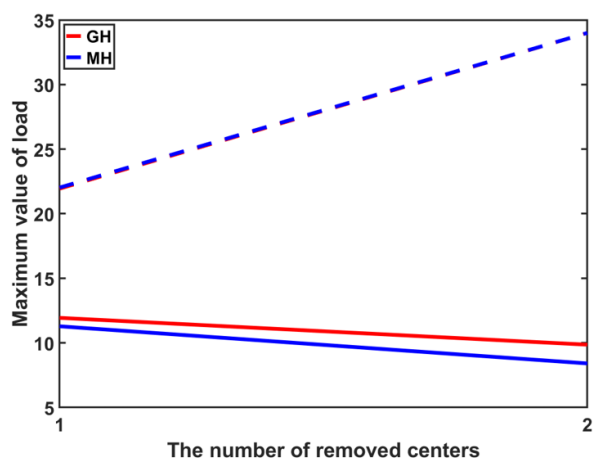
図 5.4: 2 ホップカバーするために、5 個のデータセンターが設置された場合の結果 (A) データセンターと利用者間の平均距離、(B) 負荷の分散、(C) 負荷の最大値。



(a)



(b)



(c)

図 5.5: 3 ホップカバーするために、3 個のデータセンターが設置された場合の結果 (A) データセンターと利用者間の平均距離、(B) 負荷の分散、(C) 負荷の最大値。

5.4 まとめ

GH 法と MH 法を比較して、主にデータセンターと利用者間の平均距離とデータセンター間の負荷分布を調べた。得られた結果は以下の通りである。

- MH 法は、GH 法よりもデータセンター配置問題のための、可能解をより多く提供できる。これはシステムの保安に役立つ。なぜなら、多くのノードが MH 法を通じてデータセンターになる可能性があるので、攻撃者はネットワークの外部から攻撃対象を特定することが難しくなるからである。
- 攻撃前のネットワークでは、MH 法による配置構成は、GH 法によるものより短い平均距離と低い負荷の分散値を持つ。
- 攻撃を受けたネットワークの修復として、代替を追加する方法で平均距離が大幅に減少し、負荷の分散値と負荷の最大値も低下する。

以上、MH 法と代替を追加する修復法により得られるセンターネットワークの構成は、回復力があり、うまく分散されていると結論づけられる。今後の研究としては、少なくとも以下のように記述されたデータセンターの 2 つの特徴は考えられる。クラウドコンピューティングの最近の進歩により、e コマース、ソーシャルネットワーキング、ビデオストリーミングなどのビッグデータアプリケーションをデータセンターネットワークに可能にし、膨大なデータと接続のスループットを生成される。したがって、データセンターは巨大なストレージを要求し、通常は大量のデータを伝送するために光ファイバーケーブルによって接続されている。一方、代替センターを作ることは、巨大なデータ移行に長い時間を要する。さらに、各データセンターが全体ではなく一部のデータしか持たない場合、どのデータを各データセンターでコピーし生成するかを決定する方法を考慮する必要がある。したがって、データの移動とデータの分布についてさらに考慮すべきと考えられる。

第6章

結言

6.1 研究成果の要約

本研究は、ループ・次数分布・コミュニティ構造のような多数ノードの相関関係において、頑健な構造構築に必要なメカニズムの解明と構築手法を目的として、筆者の行ってきた研究成果をまとめたものである。以下、各章を要約する。

第1章では、現実ネットワークは極めて脆弱な構造を持っているため、あらゆる攻撃に対して頑健な構造を構築する必要性を述べる。従来の頑健性研究は主に、2つのノードの関係である次数相関に注目していたことと、次数相関研究の限界について説明した。したがって、多数ノードの相関関係である、ループ、次数分布、コミュニティ構造が頑健性と関係があることを示し、それに応じる本研究の目的を述べ、各章の構成を記述する。

第2章では、本研究で扱うネットワークの基本定義、特性、そして主な解析指標である、頑健性、効率性を説明する。中央集権システムとは違う、コンピューター科学での分散システム概念を説明する。従来から今まで次数相関とループに基づいた頑健性研究、ネットワークから重要なノードを探す方法、最新のコミュニティ構造と頑健性間の関係についての研究を説明し、コミュニティ構造と関係あるソーシャル・キャピタルを記述した。

第3章では、攻撃を受けたネットワークを頑健な構造へ再構築する自己修復法を提案する。ループ強化と次数分布の均質化に基づいた自己修復法は、従来の自己修復法より、高い頑健性と効率性を持つことと、修復法により変化されるネットワークの構造をローカルマップや次数分布で示す。その結果により、ループ強化と次数分布の均質化が頑健な構造に重要であることを分かる。また、提案する自己修復法は、分散処理を前提していたので、ローカルな資源制御を考案する。そして、ローカルな資源制御と分散処理ができるような分散アルゴリズムを記述し、実装結果が従来の逐次アルゴリズムと同じであることを

示す。

第4章では、初期に高いモジュール性を持つレギュラーグラフを構想して、そのネットワークのモジュール性を反モジュール化で減少させながら、ネットワークの頑健性と効率性を確認する。モジュール間を繋ぐノードを優先して壊す MB 攻撃、媒介中心性に基づいた IB 攻撃、偶発的な故障である RF に対して頑健性を調べる。その結果、最適の耐性を持つレギュラーグラフにおいて、高いモジュール性は、レギュラーグラフの頑健性を極めて脆弱な SF ネットワークより低くすることを示す。このように、コミュニティ内の連結だけが密になると、ネットワーク全体が脆弱になる。ソーシャル・キャピタルにおいて、この結果は橋渡し型を重要性を示す。ここで、イントラリンクは結合型、インターリンクは橋渡し型ソーシャル・キャピタルに対応するだと考えられる。パットナムは、既に強力な結合型ソーシャル・キャピタルが持つ危険性を認めたが [31]、本研究ではその危険性を定量的に見せることができる。一方、ランダムレギュラーグラフは RF より、IB 攻撃に対して高い頑健性を持つことを確認する。また、穴居人グラフのように、少数のイントラリンクをインターリンクヘリワイヤすることで、ネットワークは SW 性を持つ。結果的に、モジュール性を減少させると、レギュラーグラフには高い頑健性と効率性が共存できる。次数分布を変化させる従来のモジュール化を修正して、提案した反モジュール化と同様な結果を得られるようにする。

第5章では、ハブのような重要ノードを探索して、そこに情報拡散の拠点（データセンター）を配置する方法により作られたデータセンターの配置構成を比較する。カーバー範囲内あるノード数が1番多いノードをデータセンターとして選択する従来の配置法 (GH 法) と、統計物理的なアプローチとマルチカーバーを融合した新しい配置法 (MH 法) を比較する。その結果、MH 法がより多くのデータセンターの配置構成の可能解を提供できる。これは、外部者が攻撃するために、データセンターを特定することが難しくなるので、システムの保安に役立つ。1番高い負荷を持つデータセンターを壊す攻撃を考慮するが、攻撃以前で、MH 法による配置構成は、GH 法によるものより短い平均距離と低い負荷の分散値を持つことを確認する。攻撃を受けたネットワークの修復として、代替を追加する方法で平均距離が大幅に減少し、負荷の分散値と負荷の最大値も低下することを示す。

6.2 今後の課題

本研究の今後の課題について述べる。

まず第3章で提案された局所分散修復法について、より破壊的だと知られているループ構造への攻撃に対して、修復法の有効性を確認する必要がある。また、長い直径を持った

ネットワークに対しては、修復法の効果が低かったことから、その現象の原因について調査する必要がある。第4章については、まだ未解明である現象についてさらなる調査が必要となる。その現象として、レギュラーグラフが偶発的な故障より、選択的な攻撃に強い明確な理由、モジュール数が最大である場合にモジュール性と頑健性の正比例関係が一部逆転すること、モジュール数が最小である場合にも、一部の破壊率において、モジュール性と頑健性の正比例関係が一部逆転することがある。また、今回には主に媒介中心性に基づいた攻撃法を考慮したが、より強い攻撃に対する頑健性についても調べる必要がある。第5章については、より現実的な制限条件をかけて、配置方法を比べる必要がある。例えば、各データセンターに全てのデータがないよう、データの分布を考慮して、データセンター間の中継を考える必要がある。また、代替センターを設置するための費用に対して、頑健性や負荷分散の評価をする必要もある。

業績リスト

原著論文 (査読あり)

- KIM1 Jaeho Kim, and Yukio Hayashi, "Distributed Self-Healing for Resilient Network Design in Local Resource Allocation Control", *Frontiers in Physics* (2022): 272.
- KIM2 Jaeho Kim, and Yukio Hayashi, "Anti-Modularization for both high robustness and efficiency including the optimal case", *PLOS One* (2023) [Submitted]

口頭・ポスター発表 - 国際学会

1. Jaeho Kim, and Yukio Hayashi, "Self-Healing Strategy for Improving Robustness in Limited Resource" , Conference on Complex Systems (CCS), 2020.12 (Online).
 2. Jaeho Kim, and Yukio Hayashi, "A Proposal of Distributed Algorithm to Reconstruct a Robust Network by Self-healing" , The Thirteenth International Conference on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE), 2021.04 (Porto, Portugal) [Awarded Paper].
 3. Jaeho Kim, and Yukio Hayashi, "Distributed Self-healing under restricted communication: the effect on spatial damages" , In Network Science: 8th International Winter Conference (Netsci-X), 2023.02 (Buenos Aires, Argentina).
 4. Jaeho KIM, and Yukio Hayashi, "High modularity reduces robustness of connectivity even in the optimal case" , Complex Networks and their Applications 2023, 2023. 11 (Menton Riviera, France).
- KIM3 Jaeho KIM, and Yukio Hayashi, "Configuring a data-center network for strong accessibility and load balancing", 10th Anniversary Korea-Japan Joint Workshop on Complex Communication Sciences (KJCCS 2024).

参考文献

- [1] LAN Amara, Antonio Scala, Marc Barthelemy, and H Eugene Stanley. Classes of small-world networks. In *The Structure and Dynamics of Networks*, pages 207–210. Princeton University Press, 2011.
- [2] Réka Albert, Hawoong Jeong, and Albert-László Barabási. Error and attack tolerance of complex networks. *nature*, 406(6794):378–382, 2000.
- [3] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.
- [4] Philippe Wendt, Augusto Voltas-Dorta, and Pere Suau-Sanchez. Estimating the costs for the airport operator and airlines of a drone-related shutdown: an application to frankfurt international airport. *Journal of Transportation Security*, 13(1):93–116, 2020.
- [5] Katrien De Langhe, Els Struyf, Christa Sys, Eddy Van de Voorde, and Thierry Vanelslander. Economic effects and costs of a temporary shutdown of an airport. *Antwerp: University of Antwerp. Retrieved August, 14:2019*, 2012.
- [6] Candid Wueest. The continued rise of ddos attacks. *White Paper: Security Response, Symantec Corporation*, 2014.
- [7] Paul Hines, Jay Apt, and Sarosh Talukdar. Large blackouts in north america: Historical trends and policy implications. *Energy Policy*, 37(12):5249–5259, 2009.
- [8] Cameron A MacKenzie, Joost R Santos, and Kash Barker. Measuring changes in international production from a disruption: Case study of the japanese earthquake and tsunami. *International Journal of Production Economics*, 138(2):293–302, 2012.
- [9] Zhi-Xi Wu and Petter Holme. Onion structure and network robustness. *Physical Review E*, 84(2):026106, 2011.

- [10] Toshihiro Tanizawa, Shlomo Havlin, and H Eugene Stanley. Robustness of onionlike correlated networks against targeted attacks. *Physical Review E*, 85(4):046109, 2012.
- [11] Christian M Schneider, André A Moreira, José S Andrade Jr, Shlomo Havlin, and Hans J Herrmann. Mitigation of malicious attacks on networks. *Proceedings of the National Academy of Sciences*, 108(10):3838–3841, 2011.
- [12] Masaya Murakami, Shu Ishikura, Daichi Kominami, Tetsuya Shimokawa, and Masayuki Murata. Robustness and efficiency in interconnected networks with changes in network assortativity. *Applied Network Science*, 2:1–21, 2017.
- [13] Yukio Hayashi. Growing self-organized design of efficient and robust complex networks. In *2014 IEEE Eighth International Conference on Self-Adaptive and Self-Organizing Systems*, pages 50–59. IEEE, 2014.
- [14] Yukio Hayashi. A new design principle of robust onion-like networks self-organized in growth. *Network Science*, 6(1):54–70, 2018.
- [15] Yukio Hayashi and Naoya Uchiyama. Onion-like networks are both robust and resilient. *Scientific reports*, 8(1):1–13, 2018.
- [16] Alfredo Braunstein, Luca Dall’Asta, Guilhem Semerjian, and Lenka Zdeborová. Network dismantling. *Proceedings of the National Academy of Sciences*, 113(44):12368–12373, 2016.
- [17] Masaki Chujyo and Yukio Hayashi. A loop enhancement strategy for network robustness. *Applied Network Science*, 6(1):1–13, 2021.
- [18] Hai-Jun Zhou. Spin glass approach to the feedback vertex set problem. *The European Physical Journal B*, 86(11):1–9, 2013.
- [19] Fuxuan Liao and Yukio Hayashi. Emergence of robust and efficient networks in a family of attachment models. *arXiv preprint arXiv:2110.03176*, 2021.
- [20] Masaki Chujyo, Yukio Hayashi, and Takehisa Hasegawa. Optimal network robustness against attacks in varying degree distributions. *arXiv preprint arXiv:2301.06291*, 2023.
- [21] Carl Folke. Resilience: The emergence of a perspective for social–ecological systems analyses. *Global environmental change*, 16(3):253–267, 2006.
- [22] Andrew Zolli and Ann Marie Healy. *Resilience: Why things bounce back*. Hachette UK, 2012.
- [23] Yukio Hayashi, Atsushi Tanaka, and Jun Matsukubo. More tolerant recon-

- structured networks using self-healing against attacks in saving resource. *Entropy*, 23(1):102, 2021.
- [24] Lazaros K Gallos and Nina H Fefferman. Simple and efficient self-healing strategy for damaged complex networks. *Physical Review E*, 92(5):052806, 2015.
- [25] Tom Hayes, Navin Rustagi, Jared Saia, and Amitabh Trehan. The forgiving tree: a self-healing distributed data structure. In *Proceedings of the twenty-seventh ACM symposium on Principles of distributed computing*, pages 203–212, 2008.
- [26] Quang Nguyen, Tuan V Vu, Hanh-Duyen Dinh, Davide Cassi, Francesco Scotognella, Roberto Alfieri, and Michele Bellingeri. Modularity affects the robustness of scale-free model and real-world social networks under betweenness and degree-based node attack. *Applied Network Science*, 6(1):1–21, 2021.
- [27] 滝沢誠 共著 白鳥則郎. **分散処理**. 情報科学コアカリキュラム講座. 丸善, 1996.
- [28] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.
- [29] Saray Shai, Dror Y Kenett, Yoed N Kenett, Miriam Faust, Simon Dobson, and Shlomo Havlin. Critical tipping point distinguishing two types of transitions in modular network structures. *Physical Review E*, 92(6):062805, 2015.
- [30] Robert D. Putnam and 潤一 河田. **哲学する民主主義：伝統と改革の市民的構造**. 叢書「世界認識の最前線」. NTT 出版, 2001.
- [31] Robert D Putnam. *Bowling alone: The collapse and revival of American community*. Simon and schuster, 2000.
- [32] Tristan Claridge. Functions of social capital—bonding, bridging, linking. *Social capital research*, 20(1):1–7, 2018.
- [33] Alex Bavelas. Communication patterns in task-oriented groups. *The journal of the acoustical society of America*, 22(6):725–730, 1950.
- [34] Google. Facts about google and competition. 2011.
- [35] Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.
- [36] Alina Campan, Traian Marius Truta, and Matthew Beckerich. Fast dominating set algorithms for social networks. In *MAICS*, pages 55–62, 2015.
- [37] Xiaofeng Gao, Xudong Zhu, Jun Li, Fan Wu, Guihai Chen, Ding-Zhu Du, and Shaojie Tang. A novel approximation for multi-hop connected clustering problem

- in wireless networks. *IEEE/ACM Transactions on Networking*, 25(4):2223–2234, 2017.
- [38] Martin Weigt and Haijun Zhou. Message passing for vertex covers. *Physical Review E*, 74(4):046110, 2006.
- [39] Fuxuan Liao and Yukio Hayashi. Identify multiple seeds for influence maximization by statistical physics approach and multi-hop coverage. *Applied Network Science*, 7(1):52, 2022.
- [40] Marcell Stippinger and János Kertész. Enhancing resilience of interdependent networks by healing. *Physica A: Statistical Mechanics and its Applications*, 416:481–487, 2014.
- [41] Suzhi Bi, Chin Keong Ho, and Rui Zhang. Wireless powered communication: Opportunities and challenges. *IEEE Communications Magazine*, 53(4):117–125, 2015.
- [42] Kaibin Huang and Xiangyun Zhou. Cutting the last wires for mobile communications by microwave power transfer. *IEEE Communications Magazine*, 53(6):86–93, 2015.
- [43] Falko Dressler. *Self-organization in sensor and actor networks*. John Wiley & Sons, 2008.
- [44] Michel Raynal. *Distributed algorithms for message-passing systems*, volume 500. Springer, 2013.
- [45] G Couloris, J Dollimore, and T Kindberg. Distributed systems: Concepts and design 2nd ed. *Addison-Wesley*, 1(99):4, 1994.
- [46] Masaki Chujyo and Yukio Hayashi. Improvement of the robustness against attacks in continuously varying degree distributions. In Rosa Maria Benito, Chantal Cherifi, Hocine Cherifi, Esteban Moro, Luis M. Rocha, and Marta Sales-Pardo, editors, *Book of Abstracts: COMPLEX NETWORKS 2021*, pages 569–571. Springer, 2021.
- [47] Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015.
- [48] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 177–187, 2005.

- [49] Duncan J Watts and Steven H Strogatz. Collective dynamics of ‘small-world’ networks. *nature*, 393(6684):440–442, 1998.
- [50] Lenka Zdeborová, Pan Zhang, and Hai-Jun Zhou. Fast and simple decycling and dismantling of networks. *Scientific reports*, 6(1):37954, 2016.
- [51] Flaviano Morone and Hernán A Makse. Influence maximization in complex networks through optimal percolation. *Nature*, 524(7563):65–68, 2015.
- [52] Salomon Mugisha and Hai-Jun Zhou. Identifying optimal targets of network attack by belief propagation. *Physical Review E*, 94(1):012305, 2016.
- [53] Liangliang Ma, Jing Liu, Boping Duan, and Mingxing Zhou. A theoretical estimation for the optimal network robustness measure r against malicious node attacks. *Europhysics Letters*, 111(2):28003, 2015.
- [54] Albert-László Barabási. Network science. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1987):20120375, 2013.
- [55] Zoran Stanić. *Regular graphs: a spectral approach*, volume 4. Walter de Gruyter GmbH & Co KG, 2017.
- [56] Bruno Requião da Cunha, Juan Carlos González-Avella, and Sebastián Gonçalves. Fast fragmentation of networks using module-based attacks. *PloS one*, 10(11):e0142824, 2015.
- [57] Duncan J Watts. *Small worlds: the dynamics of networks between order and randomness*, volume 36. Princeton university press, 2004.
- [58] Nahuel Almeida, Orlando Vito Billoni, and Juan Ignacio Perotti. Scaling of percolation transitions on erdős-rényi networks under centrality-based attacks. *Physical Review E*, 101(1):012306, 2020.
- [59] Petter Holme, Beom Jun Kim, Chang No Yoon, and Seung Kee Han. Attack vulnerability of complex networks. *Physical review E*, 65(5):056109, 2002.
- [60] Duanbing Chen, Linyuan Lü, Ming-Sheng Shang, Yi-Cheng Zhang, and Tao Zhou. Identifying influential nodes in complex networks. *Physica a: Statistical mechanics and its applications*, 391(4):1777–1787, 2012.
- [61] Shuai Shao, Xuqing Huang, H Eugene Stanley, and Shlomo Havlin. Percolation of localized attack on complex networks. *New Journal of Physics*, 17(2):023049, 2015.
- [62] James PG Sterbenz, David Hutchison, Egemen K Çetinkaya, Abdul Jabbar,

- Justin P Rohrer, Marcus Schöller, and Paul Smith. Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. *Computer networks*, 54(8):1245–1265, 2010.
- [63] Jacek Rak and David Hutchison. *Guide to disaster-resilient communication networks*. Springer Nature, 2020.
- [64] Partha Basuchowdhuri and Subhashis Majumder. Finding influential nodes in social networks using minimum k-hop dominating set. In *Applied Algorithms: First International Conference, ICAA 2014, Kolkata, India, January 13-15, 2014. Proceedings 1*, pages 137–151. Springer, 2014.
- [65] Alan D Amis, Ravi Prakash, Thai HP Vuong, and Dung T Huynh. Max-min d-cluster formation in wireless ad hoc networks. In *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No. 00CH37064)*, volume 1, pages 32–41. IEEE, 2000.
- [66] Thai HP Vuong and Dung T Huynh. Adapting d-hop dominating sets to topology changes in ad hoc networks. In *Proceedings Ninth International Conference on Computer Communications and Networks (Cat. No. 00EX440)*, pages 348–353. IEEE, 2000.
- [67] Qinghong Zhong, Ying Wang, Wenjing Li, and Xuesong Qiu. A min-cover based controller placement approach to build reliable control network in sdn. In *NOMS 2016-2016 IEEE/IFIP network operations and management symposium*, pages 481–487. IEEE, 2016.
- [68] Reuven Bar-Yehuda and Shimon Even. A local-ratio theorem for approximating the weighted vertex cover problem. In *North-Holland Mathematics Studies*, volume 109, pages 27–45. Elsevier, 1985.
- [69] Javier Borge-Holthoefer, Alejandro Rivero, and Yamir Moreno. Locating privileged spreaders on an online social network. *Physical review E*, 85(6):066123, 2012.
- [70] Gouhei Tanaka, Kai Morino, and Kazuyuki Aihara. Dynamical robustness in complex networks: the crucial role of low-degree nodes. *Scientific reports*, 2(1):232, 2012.
- [71] Maksim Kitsak, Lazaros K Gallos, Shlomo Havlin, Fredrik Liljeros, Lev Muchnik, H Eugene Stanley, and Hernán A Makse. Identification of influential spreaders

- in complex networks. *Nature physics*, 6(11):888–893, 2010.
- [72] Shuai Gao, Jun Ma, Zhumin Chen, Guanghui Wang, and Changming Xing. Ranking the spreading ability of nodes in complex networks based on local structure. *Physica A: Statistical Mechanics and its Applications*, 403:130–147, 2014.
- [73] Xian Teng, Sen Pei, Flaviano Morone, and Hernán A Makse. Collective influence of multiple spreaders evaluated by tracing real information flow in large-scale social networks. *Scientific reports*, 6(1):36043, 2016.
- [74] Sebastian Orłowski, Roland Wessäly, Michal Pióro, and Artur Tomaszewski. Sndlib 1.0—survivable network design library. *Networks: An International Journal*, 55(3):276–286, 2010.