

The Bi-Objective Critical Node Detection Problem

Mario Ventresca^{a,*}, Kyle Robert Harrison^b, Beatrice M. Ombuki-Berman^c

^a*School of Industrial Engineering, Purdue University, West Lafayette, USA*

^b*Department of Computer Science, University of Pretoria, Pretoria, South Africa*

^c*Department of Computer Science, Brock University, St. Catharines, Canada*

Abstract

Identifying critical nodes in complex networks has become an important task across a variety of application domains. The Critical Node Detection Problem (CNDP) is an optimization problem that aims to minimize pairwise connectivity in a graph by removing a subset of K nodes. Despite the CNDP being recognized as a bi-objective problem, until now only single-objective problem formulations have been proposed. In this paper we propose a bi-objective version of the problem that aims to maximize the number of connected components in a graph while simultaneously minimizing the variance of their cardinalities by removing a subset of K nodes. We prove that our bi-objective formulation is distinct from the CNDP, despite their common motivation. Finally, we give a brief comparison of six common multi-objective evolutionary algorithms using sixteen common benchmark problem instances, including for the node-weighted CNDP. We find that of the examined algorithms, NSGAII generally produces the most desirable approximation fronts.

Keywords: Networks, critical node detection, multi-objective, evolutionary algorithms

1. Introduction

The problem of identifying critical nodes in a network has recently attracted a significant amount of research attention. These critical nodes may be used to promote or mitigate a diffusive process spreading on the network, or to identify critical junctions through which the process spreads.

*Corresponding author

Email address: mventresca@purdue.edu (Mario Ventresca)

A number of different definitions of a *critical node* have been proposed due to the variety of application domains where discovering such nodes is important. For instance, to define junctions in cell-signaling or protein-protein networks [1], to identify highly influential individuals [2], to determine smart grid vulnerability [3], to discover key points in brain functionality [4], and to determine individuals to target for vaccination or quarantine when mitigating disease spread [5, 6, 7]. A number of other applications in the military domain have also been identified [8].

In this paper we are motivated primarily by the Critical Node Detection Problem (CNDP), as described in [9], where it was also proven to be \mathcal{NP} -hard. Given a network $G = (V, E)$ of $|V| = n$ nodes/vertices and $|E| = m$ links/edges, the goal of the problem is to minimize pairwise connectivity in G by removing no more than K nodes. Several variants of the CNDP have been investigated. In [10], an integer linear programming model with a non-polynomial number of constraints was given and branch-and-cut algorithms were proposed. The particular case where G is a tree structure was shown to be \mathcal{NP} -complete for non-unit edge costs [11]. A polynomial-time dynamic programming algorithm with worst-case complexity $\mathcal{O}(n^3 K^2)$ for solving the unit edge cost problem in graphs was proposed in [12]. Recently, a reformulation of the CNDP was proposed in [13, 14] in order to reduce the number of constraints from $\Theta(n^3)$ to $\Theta(n^2)$. Approximation algorithms based on β -edge disruptors were given in [15, 16] along with an alternative proof for the \mathcal{NP} -hardness of the CNDP. Bicriteria randomized rounding approaches based on an LP-relaxation have been proposed in [6, 17]. Simulated annealing and population-based incremental learning algorithms without approximation bounds were given in [18]. Other recent investigations include [19], which provides multiple greedy constructive heuristics and [20] that examines two local search metaheuristics. A variable neighborhood search was proposed in [21] that outperformed the population-based results of [18].

Using the network data of [13] and [18], [22] demonstrated the effectiveness of two newly proposed heuristics for the CNDP (one heuristic was for the cardinality constrained variant of the problem). Despite the reformulation of [13, 14], traditional mathematical programming techniques have not yet been shown feasible for networks containing hundreds of thousands or millions of vertices. However, a fast greedy algorithm has been recently presented for approximating solutions for such very large scale networks [23]. All of the aforementioned algorithms considered the CNDP as a single-objective problem.

Critical node detection is related to other problems of graph fragmenta-

tion and network interdiction. Graph fragmentation is concerned with the splitting of a given graph according to some set of criteria by removing edges and/or vertices. Some fragmentation problems having the most in common with critical node detection are the minimum contamination problem [24] and the sum-of-squares partitioning problem [25, 7]. Other relevant examples include the minimum multi-cut problem [26], k -way vertex cut [27], classical multi-way cut, multi-cut and k -cut problems [28, 29], in addition to the sparsest cut problem [30]. Further information pertaining to graph fragmentation problems can be found in [31, 32].

Related to network fragmentation is the area of network interdiction, which refers to the monitoring or halting of an adversary’s activity on a network [33]. Without loss of generality, these types of problems are described within a game theoretic framework where an evader operates on the network to maximize some objective function, and the interdictor has the ability to modify the network structure so as to minimize the evader’s objective function. Within this context, multi-objective problems have been investigated using mathematical programming [34] and evolutionary algorithms [35]. The main difference between interdiction and fragmentation problems concerns the explicit consideration of an evader.

1.1. Our Contributions

In this work we propose a bi-objective critical node detection problem that we term the BOCNDP and we provide experimental results based on a variety of standard evolutionary algorithms. Our contributions are more explicitly stated as:

1. *Propose BOCNDP*: Pairwise connectivity is loosely composed of two separate objectives; maximizing the total number of connected components and minimizing the variance in the cardinality (number of nodes) among the connected components, after removing K vertices from the original graph. A single objective formulation of this problem has been studied previously, however, despite the implied bi-objective nature, to date no multi-objective investigations have been performed. We outline the BOCNDP and provide arguments showing its uniqueness from the CNDP as well as highlight other properties and observations.
2. *Experimental comparison*: Six common multi-objective evolutionary algorithms are employed in order to discover solutions to the BOCNDP. The algorithms and brief descriptions are provided in Section 3.1. We compare the ability of these algorithms to arrive at quality solutions against a set of benchmark problem instances described in

[18] and a new set of instances (described below). The Pareto front approximation of each algorithm is also analyzed using the additive epsilon indicator, hypervolume, and spacing measures.

3. *Benchmark networks*: A set of benchmark network instances is created for evaluating node-weighted network fragmentation problems. Two variations of node weights (random and logarithmic) are created by modifying existing benchmark instances for unweighted networks [18]. This set of networks is publicly available online from [36].

It should be noted that this work builds upon our preliminary results presented in [37]. The first major contribution of this article, not present in the previous work, is with regards to the theoretical foundations, from which the formulation of the BOCNDP from the CNDP directly follows. Secondly, we provide proofs that the optimal CNDP solution need not lie within the Pareto front of the corresponding BOCNDP instance (Theorem 1) and that the BOCNDP is \mathcal{NP} -hard (Theorem 2). Lastly, we provide additional empirical analysis with regards to the multi-objective results. Specifically, we additionally compare the entire set of non-dominated solutions found by the optimizers across all runs (on a given benchmark problem), in contrast to comparing on individual runs, thereby giving a more comprehensive analysis of the search capabilities of each optimizer. The additional results are generally agreeable with previous findings, however, they do highlight cases where the low levels of diversity present in the individual Pareto fronts can be misleading with regards to the overall performance of an optimizer. Another experiment is conducted for two variations of node weighted problem instances.

The remainder of this paper is organized as follows. Section 2 describes the BOCNDP and the motivation from the single objective CNDP. In Section 3 the experimental setup is described and a variety of results are shown. Finally, conclusions and avenues for future work are provided in Section 4.

2. The Bi-Objective Critical Node Detection Problem

The BOCNDP is a distinct bi-objective optimization problem that is motivated by the goals of the critical node detection problem. As such, the latter is first discussed before introducing the BOCNDP.

2.1. The Critical Node Detection Problem

The CNDP is an \mathcal{NP} -hard optimization problem formulated by [9, 38]. Let $G = (V, E)$ be an unweighted, undirected graph composed of $|V| = n$

vertices and $|E| = m$ edges. The goal of the CNDP is to determine a subset of vertices $R \subseteq V$ such that the residual graph $G(V \setminus R)$ has minimum pairwise connectivity where $|R| \leq K$. The problem has decision variables [9]:

$$x_{uv} = \begin{cases} 1 & \text{if vertices } u, v \text{ are in the same component of } G(V \setminus R) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$y_u = \begin{cases} 1 & \text{if vertex } u \text{ is deleted in the solution} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

and subsequently mathematically formulated by the following integer program:

$$\text{minimize} \quad \sum_{u,v \in V} x_{uv} \quad (3)$$

$$\text{subject to} \quad x_{uv} + y_u + y_v \geq 1 \quad \forall (u, v) \in E \quad (4)$$

$$x_{uv} + x_{vw} - x_{wu} \leq 1 \quad \forall u, v, w \in V \quad (5)$$

$$x_{uv} - x_{vw} + x_{wu} \leq 1 \quad \forall u, v, w \in V \quad (6)$$

$$-x_{uv} + x_{vw} + x_{wu} \leq 1 \quad \forall u, v, w \in V \quad (7)$$

$$\sum_{u \in V} y_u \leq K \quad (8)$$

$$x_{uv} \in \{0, 1\} \quad \forall u, v \in V \quad (9)$$

$$y_u \in \{0, 1\} \quad \forall u \in V \quad (10)$$

The constraint in Equation (4) enforces the rule that deleted vertices do not share an edge to any other vertex by setting edge x_{uv} as deleted if either or both vertices u and v are deleted. Equations (5)-(7) are focused on ensuring that vertices belonging to the same connected component are properly identified by enforcing a triangle inequality whereby if vertices u, v and v, w are connected in the residual graph, then u and w must also be connected (recall that a subgraph $G_S = (V_S \subseteq V, E_S \subseteq E)$ of G is a connected component if and only if (1) there exists an undirected path between every pair of vertices $u, v \in V_S$ and (2) $\forall v \in (V \setminus V_S)$ there does not exist $u \in V_S$ such that $(u, v) \in E$). The constraint in Equation (8) ensures that no more than K vertices are deleted from G . The node-weighted version of the CNDP (wCNDP) will have Equation (8) replaced with $\sum_{u \in V} \omega(u)y_u \leq W$ where $W \in \mathbb{R}_{>0}$, and $\omega : V \mapsto \mathbb{R}_{>0}$ defines the node weight. As described in the introduction, the CNDP has a variety of applications and has spurred a number of subsequent research endeavors.

2.2. From CNDP to BOCNDP

To better illustrate the BOCNDP we begin with a more general formulation. Let $D = \{C_i\}$ be a finite set of connected components of the residual graph $G(V \setminus R)$, and let $|D|$ be the cardinality of this set (number of connected components). Then, the CNDP can be equivalently written as the problem of determining a subset $R \subseteq V$ such that the residual graph contains $i = 1, \dots, |D|$ connected components that each contain $|C_i|$ vertices [12]:

$$\text{minimize} \quad \sum_{i=1}^{|D|} \binom{|C_i|}{2} \quad (11)$$

$$\text{subject to} \quad \sum_{i \in R} w_i \leq W \quad (12)$$

where $w_i > 0$ are weights associated with each vertex and $W > 0$ is a knapsack constraint. The CNDP formulation given in the previous section is a special case where $w_i = 1 \forall i = 1, \dots, n$. As shown in [15], the CNDP objective function can be rewritten as

$$\sum_{i=1}^{|D|} \binom{|C_i|}{2} = \frac{1}{2} \left(\sum_{i=1}^{|D|} |C_i|^2 - |V| \right) \quad (13)$$

$$= \frac{1}{2} \left(\frac{n_*^2}{|D|} - n_* \right) + \frac{|D|}{2} \text{var}(D) \quad (14)$$

where

$$\text{var}(D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \left(|C_i| - \frac{n_*}{|D|} \right)^2 \quad (15)$$

is the biased sample variance of the cardinalities of the connected components, and $n_* = \sum_{i=1}^{|D|} |C_i|$ is the number of nodes in $G(V \setminus R)$. The following two lemmas highlight the preference of CNDP solution in circumstances of equal size or number of connected components.

Lemma 1 (from [9]). *Let D be a partition of $G = (V, E)$ into $|D|$ connected components obtained by deleting a set of R nodes, where $|R| = K$. Then the objective function $\sum_{i=1}^{|D|} \frac{|C_i|(|C_i|-1)}{2} \geq \frac{(|V|-K) \left(\frac{|V|-K}{|D|} - 1 \right)}{2}$, with equality holding if and only if $|C_i| = |C_j| \forall i, j = 1, \dots, |D|$, where $|C_i|$ is the number of nodes in the i^{th} component of D .*

Lemma 2 (from [9]). *Let D_1 and D_2 be two sets of partitions obtained by deleting $R_1, R_2 \subseteq V$ from $G = (V, E)$, where $|R_1| = |R_2| = K$. Let $|D_1|$ and $|D_2|$ be the number of connected components in D_1 and D_2 , respectively, and let $|D_1| \geq |D_2|$. If each component of D_1 contains an equal number of vertices, then the CNDP objective function value obtained by deleting R_1 is lower than if deleting R_2 .*

From Lemmas 1 and 2, [9] states that the CNDP can be viewed as simultaneously minimizing the variance and maximizing the number of connected components. However, the two statements are made independently of each other and, as we show in Theorem 1, there are solutions whose optimal CNDP solution does not simultaneously minimize the variance and maximize the number of connected components. This suggests the BOCNDP, while motivated by the CNDP, is a different optimization problem and would permit a user to decide among a number of potential graph fragmentations that would better suit their particular circumstance. Such a situation arises frequently in the real-world. For instance, when the user is not able to properly formulate the optimization problem due to stochastic effects, or the value of a node is at least partially assigned qualitatively, nodes have complicated contextual information or missing data, thereby resulting in the need to evaluate solution trade-offs by inspection and/or subsequent methods before a decision can be made.

Consider disrupting a terrorist communication network by neutralizing a subset of individuals. Maximizing the number of connected components in the residual graph could create many smaller, and perhaps less dangerous, cells. However, if many individuals are isolated (i.e., component containing a single individual) then a large connected component may remain intact and the network will not be significantly impacted unless, perhaps, if the isolated individuals are high value targets. The CNDP introduces bias into the objective function toward components containing an equal number of individuals, implying that each individual or community is of equal value, which seems unlikely in context. However, if all connected components have an equal number of individuals then potential information flow between any arbitrary pair of terrorists is minimized and could be beneficial for limiting the transmission of command orders; but this implies that a connected component is unlikely capable of operating independently. Introducing weighted nodes is possible but in practice accurately determining a weight may be extremely difficult or impossible. Alternatively, determining a set of potential solutions that trade off general objectives (maximal fragmentation with minimum variance between the number of nodes in each component)

in conjunction with contextual information by an outside source (as a proxy for node weights) could better solve the problem.

2.3. BOCNDP Formulation and Properties

We therefore propose to reconsider Equation (13). Specifically, since the number of vertices in the original graph n is fixed, the focus lies on maximizing $|D|$ and minimizing $var(D)$ as calculated in Equation (15), and thus the goal is to optimize the bi-objective problem:

$$\text{minimize } var(D) \tag{16}$$

$$\text{maximize } |D| \tag{17}$$

$$\text{subject to } \sum_{i \in R} w_i \leq W \tag{18}$$

Maximizing $|D|$ is equivalent to maximizing the multiplicity of eigenvalue 0 of the graph Laplacian (i.e., for a graph with $|D|$ connected components, eigenvalue 0 has multiplicity $|D|$) [39].

We refer to Equations (16)-(18) as the BOCNDP. Since the BOCNDP is a multi-objective problem, the goal is to best approximate the Pareto front in order to discover and better understand the trade-offs between the objectives. The BOCNDP is a similarly stated, but fundamentally different problem than the CNDP and despite their common motivation, solutions cannot be meaningfully compared because the BOCNDP cannot be transformed into the CNDP. Their differences are highlighted by the following result that shows the optimal CNDP solution need not exist in the Pareto front of the BOCNDP.

Theorem 1. *For a network $G = (V, E)$ with real-valued node weights given by $\omega : V \mapsto \mathbb{R}_{>0}$, the optimal CNDP solution after deleting $K \geq 1$ nodes does not necessarily exist in the BOCNDP Pareto front.*

Proof. (By counterexample) Let D be the optimal partition of G after deleting K vertices, and let $X = |D|$ be the number of connected components and $Y = var(D)$ as calculated by Equation (15). The BOCNDP Pareto front for partitions of $G(V \setminus K)$ will be represented by \mathcal{P} . If $(X, Y) \in \mathcal{P} \forall G$, then it must be that (X, Y) is always non-dominated with respect to both BOCNDP objectives. By providing a counterexample network instance for when the range of ω is constant or non-constant, we show that $(X, Y) \notin \mathcal{P} \forall G$, which proves that the CNDP and the BOCNDP are indeed different problems.

Case 1: constant vertex weight. Without loss of generality $\omega(v) = 1 \forall v \in V$. The network in Figure 1 (a) contains 24 nodes and for $K = 3$,

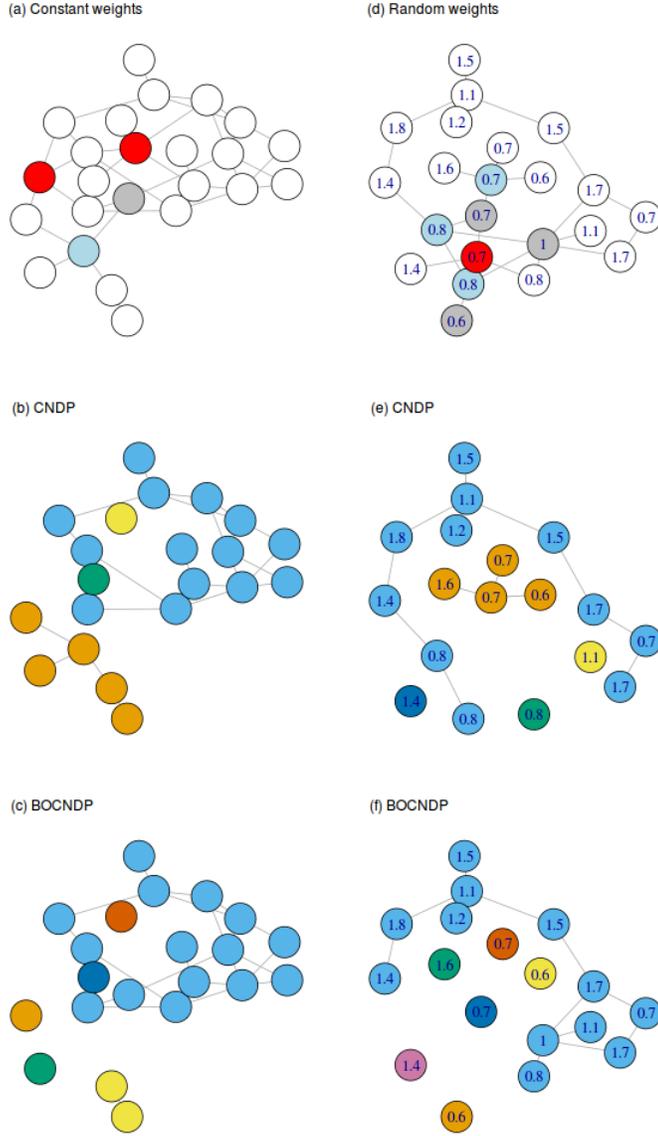


Figure 1: Counterexamples showing that the optimal CNDP solution does not necessarily exist in the BOCNDP Pareto front. (a) and (d) are the constructed networks with red nodes being common between the optimal CNDP solution (in (b) and (e)) and the BOCNDP counter-example (in (c) and (f)), respectively; gray nodes exist only in the optimal CNDP cut set and blue nodes exist only in the BOCNDP cut set. Node colors for figures (b), (c), (e) and (f) are to better highlight the connected components of the residual network after deleting nodes. Node weights are shown for (d)-(f). The network with constant weights contains 24 nodes and uses a constraint $K = 3$, whereas that with random weights contains 22 nodes and has a constraint $W = 3$.

an optimal CNDP solution is shown in (b) with objective value 101 and contains $|D_c| = 4$ connected components of sizes $\{5, 14, 1, 1\}$ nodes, respectively, and variance of $Y_c = 28.1875$, as calculated by Equation (15). However, there exists a BOCNDP solution shown in (c) that has a corresponding CNDP objective value of 106, but contains $|D_b| = 6$ connected components of $\{15, 1, 2, 1, 1, 1\}$ nodes, respectively, and a variance of $Y_b = 26.58333$. Since $|D_b| > |D_c|$ and $Y_b < Y_c$, $(|D_c|, Y_c) \notin \mathcal{P}$.

Case 2: random vertex weights. The network in Figure 1 (d) contains 22 nodes and for $W = 3$ has an optimal CNDP objective value of 61 that is presented in (e) and contains $|D_c| = 5$ connected components of $\{4, 11, 1, 1, 1\}$ nodes, respectively, and a variance of $Y_c = 15.04$. However, there exists a BOCNDP solution shown in (f) that has $|D_b| = 7$ connected components containing $\{1, 12, 1, 1, 1, 1, 1\}$ nodes, respectively, and a variance of $Y_b = 14.81633$, but CNDP objective value of 66. Since $|D_b| > |D_c|$ and $Y_b < Y_c$, $(|D_c|, Y_c) \notin \mathcal{P}$. \square

Moreover, the BOCNDP is straightforwardly shown to be \mathcal{NP} -hard by showing that any objective is \mathcal{NP} -hard. Specifically, we refer to the maximization of the number of connected components through the K-WAY VERTEX CUT problem [27].

Theorem 2. *The BOCNDP is \mathcal{NP} -hard.*

Proof. The objective of maximizing the number of connected components by removal of $R \subseteq V$ from G is known as the K-WAY VERTEX CUT problem [27], which has already been shown to be \mathcal{NP} -hard. Therefore, the BOCNDP is also \mathcal{NP} -hard. \square

3. Experimental Setup and Results

We evaluate a number of standard multi-objective evolutionary algorithms for the BOCNDP using standard benchmark problems from [18]. The experimental results highlight the performance of each algorithm in comparison to a random search, which is used as a control for the experiment. Ideally, the objectives of the BOCNDP would be independent. However, previous studies suggest that variable dependence does not necessarily lead to deteriorated performance and, in some two-objective problems, the addition of highly dependent objectives actually improved performance [40, 41]. We first provide an outline of the experimental setup and parameters.

3.1. Setup

Here we describe the algorithms, their parameterization and the benchmark problem instances. All results were obtained using the MOEA 2.1 Java library [42], and ran on a Linux Mint Debian Edition operating system with an Intel i7-4930K 3.4GHz CPU and 64GB of RAM. Experiments were run for 20 trials.

MO Algorithms: We compare results attained by the MOCHC [43], PAES [44], NSGAI [45], eNSGAI [46], eMOEA [47], and PESA2 [48] algorithms, each of which is briefly outlined below. As a control, we also implement a random search. We use a bit string representation that indicates whether a particular node is present in a solution or not. The empirically-determined population size for each algorithm is equal to the size of the largest component in the current benchmark instance and the mutation rate is inversely proportional to this size. The number of iterations for each algorithm is set as 3000 times the size of the largest network component in the current benchmark. All algorithms utilize the HUX half uniform crossover and other default settings in the MOEA Framework as presented in Table 1. For the ‘Parent selection’ parameter in Table 1, R refers to random selection while T refers to tournament selection with $k = 2$.

MOCHC: The Multi-Objective Cross generational elitist selection, Heterogeneous recombination, and Cataclysmic mutation (MOCHC) algorithm is a variant of the genetic algorithm that couples a conservative selection strategy with highly disruptive recombination in an attempt to produce offspring that are maximally different from the parents.

PAES: The (1+1) Pareto Archived Evolution Strategy (PAES) algorithm is a multi-objective evolutionary strategy that uses an adaptive grid archive to maintain a fixed-size set of diverse solutions.

NSGAI: The Non-dominated Sorting Genetic Algorithm II (NSGAI) sorts the population into sub-populations based on the ordering of Pareto dominance and calculates a crowding distance between members of the sub-populations to provide a ranking within each sub-population such that isolated solutions are preferred. These ranking mechanisms are subsequently used during the selection phase.

eNSGAI: The Epsilon NSGAI (eNSGAI) algorithm is an extension of NSGAI that uses a solution archive based on a relaxation of the traditional dominance relation, namely an ϵ -archive. Furthermore, eNSGAI also has random restarts to (ideally) provide a more diverse set of solutions.

eMOEA: The Epsilon Multi-Objective Evolutionary Algorithm (eMOEA) is a steady-state evolutionary algorithm that uses an ϵ -archive to promote

Table 1: Algorithm parameters. All algorithms had as the maximum number of evaluations, $3000 \times$ the number of nodes in the initial network.

Parameter	MOCHC	PAES	eNSGAI	eMOEA	NSGAI	PESA2
Initial convergence count	25%					
Preserved population	5%					
Convergence value (k)	3					
Parent selection	R		T	T	T	
Archive size		100				100
Bisections		8				8
Epsilon (ϵ)			0.01	0.01		
Injection rate			25%			
Population size (range)			[100, 10000]			
Window size			100			

diversity.

PESA2: The Pareto-Envelope based Selection Algorithm II (PESA2) partitions the objective space into a number of hyperboxes and derives a fitness value for each non-empty hyperbox, rather than assigning fitness values to individual solutions, to promote a selection bias towards less-crowded solutions. The selection phase then uses any standard selection technique to select a hyperbox and the individual used for genetic operations is selected randomly from the chosen hyperbox.

Data: We utilize the benchmark data proposed in [18] and highlighted in Table 2. This data set contains sixteen undirected, unweighted networks created using common complex network generator algorithms: Barabasi-Albert, Watts-Strogatz, Forest Fire, and the Erdos-Renyi random graph. The first three of these models generate networks with a single component whereas the random graph algorithm may contain a number of connected components and only the largest is retained in the data set. Note that the problem name indicates the size of the largest connected component, e.g., ER_234 indicates that the largest connected component contains 234 vertices. Further information about the network structures can be found in [18].

In order to evaluate algorithm performance on weighted networks new benchmark instances were created by assigning vertex weights to the networks in [18] (these can be downloaded from [36]). Two different weighting schemes were considered:

1. Randomly assigned, where $\omega(v) \in [0.2, 3] \forall v \in V$.
2. Logarithmic with node degree d_v , where $\omega(v) = \log(d_v) + 0.5 \forall v \in V$.

Table 2: The sixteen benchmark problems, their associated maximum K -critical nodes for constant-weighted instances, and associated maximum knapsack constraint W for weighted instances.

Name	Vertices	Edges	K	W	Name	Vertices	Edges	K	W
ER_234	250	349	50	70	BA_500	500	499	50	60
ER_465	500	699	80	130	BA_1000	1000	999	75	110
ER_940	1000	1399	140	175	BA_2500	2500	2499	100	200
ER_2343	2500	3499	200	325	BA_5000	5000	4999	150	350
WS_250	250	1249	70	150	FF_250	250	400	13	70
WS_500	500	1499	125	275	FF_500	500	792	25	130
WS_1000	1000	4999	200	565	FF_1000	1000	1633	50	175
WS_1500	1500	4499	265	565	FF_2500	2500	4046	125	300

3.2. Algorithm Performance

Our comparison is from two perspectives: (1) the quality of solutions with respect to the two objectives, and (2) the algorithms’ ability to approximate the Pareto front. Concerning the latter perspective, the additive epsilon indicator [49], hypervolume [50], and spacing [51] measures are considered as they provide both an indication of the quality (i.e., proximity to the true front) and spread of solutions in the approximation front. The additive epsilon and hypervolume indicators are Pareto-compliant while the spacing measure is non-compliant.

Additive Epsilon Indicator (AE): is the minimum ϵ such that for every solution in the approximation front, there exists a solution in the reference front that is no more than ϵ better in all objectives.

Hypervolume (H): measures the volume of objective space dominated by an approximation front.

Spacing (S): quantifies the distribution of solutions in the approximation front.

We rank the algorithms performance using the following procedure. First, a Kruskal-Wallis test is performed to determine whether a performance difference exists between the algorithms. If so, then pairwise Mann-Whitney U-tests are conducted in order to determine where the differences occur and tabulate a “win” accordingly, based on a comparison of medians. Finally, a rank is assigned to each algorithm based on the difference in number of wins and losses. All statistical tests are performed at the 95% confidence level. Note that while full-precision was used for the statistical analysis, only a limited number of significant figures are presented in tabular form due to space limitations. Therefore, in some instances a statistically significant difference is noted for results where there appears to be no disparity.

3.3. Final Solution Quality

In this section we compare the final results obtained for each objective by the six algorithms. Tables 3 to 5 present the average objective values for each of the examined problem instances. Additionally, tables outlining the extreme points ($\max |D|, -$) and ($-, \min \text{var}(D)$) are provided in the supplementary material for this paper. Bold entries denote the highest ranked objective values using the statistical ranking procedure described in Section 3.2. We note that the results for the constant-weighted Watts-Strogatz problems are problematic due to degenerate solutions being produced. Therefore, the results are unreliable and no further analysis is performed with regards to these problems. A more comprehensive analysis would consider different search operators and may alleviate this issue.

When considering maximization of the number of components, it is clear that the NSGAI algorithm is generally superior to the other algorithms, irrespective of the network type or node weighting scheme. Similarly, the eNSGAI algorithm attains the best performance on a number of problems, most notably the Forest Fire networks. Evidently, the non-dominating sorting procedure is highly effective at finding solutions that maximize the number of components.

When considering the variance objective, the results were far more problem dependent. However, the NSGAI or MOCHC algorithms lead to the lowest variance on a majority of problems, with the PAES algorithm occasionally leading to the best variance. Given that minimizing the variance in the BOCNDP is not completely independent from maximizing the number of components, these results may suggest that the BOCNDP has a slight bias towards maximizing the number of components (expected given that the number of components is a factor in computing Equation (15)).

3.4. Pareto Front Approximation

Tables 6 to 8 present the aggregated ranks¹ for each of the multi-objective measures as a means to evaluate the ability to approximate the true Pareto front.

When considering both the constant-weighted and log-weighted problems, as shown in Tables 6 and 7, the NSGAI optimizer generally attains the best average rank for both the hypervolume and additive epsilon measures, but shows significantly poorer results for the spacing measure. Considering

¹The individual ranks and difference scores for each problem instance are provided in the supplementary data.

Table 3: Summary results for the two BOCNDP objectives across the constant-weighted benchmark instances.

Algorithm	Problem	$ D $		$var(D)$		Problem	$ D $		$var(D)$	
		μ	σ	μ	σ		μ	σ	μ	σ
eMOEA	BA_500	312.6	0.5	0.62	0.00	ER_234	61.2	2.3	12.01	13.48
eNSGAI		312.9	0.3	7.23	0.23		61.5	2.5	10.49	9.83
MOCHC		291.7	9.9	1.16	0.31		62.5	2.7	16.23	12.67
NSGAI		312.7	0.4	0.62	0.00		62.2	2.9	15.28	13.89
PAES		311.2	1.6	0.65	0.03		59.5	2.4	5.11	1.88
PESA2		311.9	1.1	0.63	0.01		60.5	2.1	10.12	12.61
Random		239.5	4.6	4.66	0.89		35.6	2.2	430.18	37.34
eMOEA	BA_1000	589.9	0.3	1.01	0.00	ER_465	101.1	3.2	304.82	34.84
eNSGAI		590.0	0.0	1.01	0.00		97.8	4.3	274.04	41.81
MOCHC		563.5	14.5	1.46	0.29		102.1	3.6	286.56	33.43
NSGAI		589.9	0.3	1.01	0.00		101.5	4.3	296.30	35.75
PAES		588.6	1.4	1.03	0.02		90.2	5.9	215.90	78.52
PESA2		589.5	0.9	1.02	0.01		95.2	4.5	250.48	59.13
Random		413.1	9.8	12.79	3.55		48.4	1.4	2009.30	83.49
eMOEA	BA_2500	1119.0	5.7	4.39	0.15	ER_940	190.5	3.4	1062.37	56.88
eNSGAI		1119.9	5.6	4.39	0.16		184.5	3.8	1058.93	86.21
MOCHC		1105.6	11.4	4.76	0.36		179.6	2.2	1486.77	46.17
NSGAI		1117.0	7.4	4.34	0.16		192.8	4.9	1011.98	52.69
PAES		1106.2	6.2	4.70	0.21		183.2	6.3	961.36	127.76
PESA2		1117.2	4.4	4.39	0.10		182.3	4.3	1083.48	83.21
Random		623.8	16.6	91.69	16.10		86.1	1.6	5273.81	108.28
eMOEA	BA_5000	1973.3	7.7	7.17	0.17	ER_2343	320.1	2.2	8689.69	86.22
eNSGAI		1976.4	8.6	7.23	0.23		297.8	4.8	9393.59	202.32
MOCHC		1855.3	13.0	10.95	0.55		276.4	1.7	11608.73	89.20
NSGAI		1978.5	11.5	7.36	0.46		322.1	1.6	8626.23	48.92
PAES		1939.6	10.5	8.11	0.35		298.1	4.7	9313.26	190.47
PESA2		1971.5	5.3	7.40	0.14		295.0	5.1	9487.31	243.45
Random		900.2	21.8	414.01	80.90		110.5	1.9	35968.10	638.26
eMOEA	FF_250	88.3	2.1	1.86	0.18	WS_250	7.2	1.4	3823.42	815.45
eNSGAI		89.0	2.1	1.77	0.18		2.4	0.5	13087.30	2474.35
MOCHC		88.1	2.4	1.82	0.20		6.5	1.3	4460.09	1382.36
NSGAI		89.4	1.8	1.72	0.16		4.6	1.9	7551.23	3460.25
PAES		87.9	2.7	1.77	0.20		1.0	0.0	N/A	N/A
PESA2		88.5	2.2	1.75	0.18		1.0	0.1	N/A	N/A
Random		46.7	7.2	454.93	2401.21		3.0	0.0	10416.60	64.48
eMOEA	FF_500	210.3	2.1	0.97	0.04	WS_500	41.5	2.5	1854.82	198.39
eNSGAI		209.9	2.5	0.94	0.04		17.9	3.6	5967.07	1820.51
MOCHC		211.0	2.4	0.96	0.05		27.1	2.2	3777.43	379.39
NSGAI		209.7	3.1	0.93	0.03		42.4	2.1	1869.55	137.73
PAES		208.1	2.7	0.95	0.05		10.0	3.0	13226.46	4836.73
PESA2		208.1	3.1	0.95	0.04		12.6	4.0	10696.66	5168.13
Random		122.7	5.2	47.44	33.09		7.1	0.3	17999.61	733.81
eMOEA	FF_1000	318.5	5.1	4.03	0.23	WS_1000	4.6	1.4	149116.31	45420.14
eNSGAI		321.0	5.2	4.17	0.33		1.2	0.5	N/A	N/A
MOCHC		301.8	5.9	6.73	0.79		3.4	1.0	204464.44	58524.47
NSGAI		326.9	7.0	4.55	0.85		4.7	1.5	150344.05	54543.17
PAES		319.3	5.6	4.16	0.39		1.0	0.0	N/A	N/A
PESA2		319.3	4.8	4.03	0.26		1.0	0.0	N/A	N/A
Random		143.2	6.7	1474.97	254.84		3.0	0.2	216277.22	21435.87
eMOEA	FF_2500	476.4	5.6	10.52	0.73	WS_1500	32.2	2.7	41583.12	3863.97
eNSGAI		477.7	5.6	10.56	0.67		11.3	4.2	144741.13	58000.05
MOCHC		385.0	8.4	45.16	6.10		15.0	1.7	95784.40	11325.50
NSGAI		486.7	5.4	12.43	1.70		30.6	2.6	44283.78	3951.78
PAES		473.7	7.0	10.62	0.89		1.9	2.6	N/A	N/A
PESA2		476.5	5.6	10.47	0.70		7.0	2.8	274558.48	187676.18
Random		187.2	13.3	7078.06	963.17		6.5	0.7	226824.92	20049.19

the additive epsilon and hypervolume measures, the performance of NSGAI versus the other techniques does not seem to be highly influenced by the network topology, or if so, all approaches are similarly influenced such that their relative performance remains. The inferior performance with respect to the spacing measure is likely a result of not having an explicit mechanism, such

Table 4: Summary results for the two BOCNDP objectives across the log-weighted benchmark instances.

Algorithm	Problem	$ D $		$var(D)$		Problem	$ D $		$var(D)$	
		μ	σ	μ	σ		μ	σ	μ	σ
eMOEA	BA_500	302.9	4.5	0.79	0.12	ER_234	78.2	3.7	2.17	0.91
eNSGAI		317.5	0.8	0.52	0.02		95.5	1.8	0.67	0.08
MOCHC		280.0	14.5	0.51	0.26		94.9	2.7	0.45	0.06
NSGAI		317.3	1.1	0.52	0.01		95.8	1.7	0.63	0.07
PAES		254.9	12.8	0.77	0.32		80.9	3.3	0.53	0.12
PESA2		259.5	9.4	3.09	0.91		61.2	4.0	15.43	10.82
Random		265.9	4.4	2.54	0.42		55.3	3.5	25.54	16.28
eMOEA	BA_1000	607.0	5.7	0.81	0.08	ER_465	137.9	6.5	11.79	10.09
eNSGAI		629.8	1.0	0.59	0.01		176.9	1.9	1.40	0.17
MOCHC		550.4	34.0	0.76	0.49		175.3	4.0	0.85	0.12
NSGAI		628.7	1.9	0.59	0.02		178.3	2.3	1.12	0.09
PAES		524.6	13.5	0.67	0.07		153.7	4.4	1.03	0.10
PESA2		508.6	21.0	4.33	1.57		102.5	9.5	104.76	84.46
Random		508.9	6.0	4.06	0.69		86.0	6.3	215.06	79.65
eMOEA	BA_2500	1214.5	27.4	3.37	0.66	ER_940	226.1	5.7	742.60	51.61
eNSGAI		1322.7	2.7	1.77	0.03		243.2	2.0	655.13	35.36
MOCHC		931.4	82.9	18.81	14.29		232.6	7.8	457.73	45.60
NSGAI		1325.3	2.9	1.69	0.03		245.8	3.0	543.79	29.89
PAES		873.5	24.7	11.16	1.30		197.3	5.4	659.73	76.12
PESA2		950.0	43.8	21.18	6.80		113.8	7.2	2801.69	256.49
Random		944.3	12.9	21.33	4.11		119.5	2.5	2540.98	82.27
eMOEA	BA_5000	2235.2	29.9	4.97	0.50	ER_2343	477.3	5.7	3888.58	75.83
eNSGAI		2446.4	9.9	2.79	0.11		468.7	15.1	4255.21	295.68
MOCHC		1559.9	98.8	50.63	30.56		470.1	4.2	3613.00	54.35
NSGAI		2467.5	9.0	2.44	0.08		499.4	2.8	3506.31	47.69
PAES		1195.0	80.8	1900.00	1447.48		272.3	5.8	8202.48	207.51
PESA2		1664.2	91.4	50.89	22.79		186.1	11.5	15639.56	1095.42
Random		1586.5	26.9	58.61	9.43		193.8	2.1	14673.93	195.68
eMOEA	FF_250	84.4	4.6	4.60	2.03	WS_250	13.0	1.6	1057.95	204.43
eNSGAI		97.3	2.4	1.67	0.25		12.4	1.6	1180.16	214.04
MOCHC		91.1	4.5	1.28	0.24		21.0	1.4	405.11	109.23
NSGAI		96.6	2.7	1.68	0.26		14.2	1.6	926.05	117.55
PAES		82.8	5.6	1.80	1.20		8.7	3.4	1056.90	1204.76
PESA2		67.5	5.5	38.22	30.82		2.8	0.5	6676.80	1228.48
Random		60.0	3.9	42.35	22.75		4.7	0.5	3656.26	239.31
eMOEA	FF_500	196.4	6.6	2.18	0.49	WS_500	59.4	3.9	8.81	7.35
eNSGAI		222.6	1.8	1.02	0.10		71.0	6.4	3.22	0.75
MOCHC		207.9	5.8	0.74	0.09		67.0	6.6	1.78	0.72
NSGAI		222.2	2.5	0.97	0.11		82.5	7.2	1.84	0.44
PAES		193.2	6.2	0.83	0.13		71.1	5.8	0.95	0.51
PESA2		160.9	9.4	8.35	5.31		33.1	6.2	295.20	358.84
Random		145.0	4.9	17.95	13.58		25.7	3.1	455.56	249.20
eMOEA	FF_1000	261.0	7.8	70.07	48.75	WS_1000	52.7	3.1	4562.57	376.66
eNSGAI		310.7	3.3	7.50	0.91		13.1	3.5	27104.17	7346.39
MOCHC		281.0	12.1	16.22	15.02		66.6	3.5	3006.87	239.54
NSGAI		310.8	5.8	6.47	1.13		62.8	4.1	3478.90	401.80
PAES		251.5	6.8	45.34	36.02		38.8	3.7	4728.78	891.91
PESA2		201.4	17.0	450.56	285.04		3.9	0.5	87393.61	10139.50
Random		166.3	9.6	835.91	219.58		6.5	0.5	51185.87	3633.52
eMOEA	FF_2000	468.5	11.0	40.74	8.24	WS_1500	112.1	4.9	3216.61	483.68
eNSGAI		572.7	5.7	11.32	0.80		48.4	15.0	14048.96	5664.27
MOCHC		502.8	19.0	13.94	9.81		109.1	6.2	164.83	275.08
NSGAI		571.0	14.0	8.86	1.40		175.3	3.2	1164.41	156.22
PAES		395.6	9.4	374.98	221.51		119.0	5.2	72.91	90.65
PESA2		379.7	28.6	477.94	607.80		17.3	1.2	49864.89	3584.08
Random		284.3	14.5	1913.19	760.26		22.8	1.0	35310.50	1821.11

as an ϵ -archive, for promoting diversity. When faced with random-weighted problems, MOCHC generally performs better than NSGAI. It is noteworthy that the MOCHC algorithm performs well on both problems with a random network topology (i.e., Erdos-Renyi networks) and a random vertex weighting.

Table 5: Summary results for the two BOCNDP objectives across the random-weighted benchmark instances.

Algorithm	Problem	$ D $		$var(D)$		Problem	$ D $		$var(D)$	
		μ	σ	μ	σ		μ	σ	μ	σ
eMOEA	BA_500	334.7	3.6	0.28	0.05	ER_234	99.9	4.1	0.35	0.09
eNSGAI		344.2	0.8	0.20	0.00		112.6	2.0	0.14	0.01
MOCHC		317.1	17.5	0.41	0.52		114.1	1.1	0.09	0.01
NSGAI		344.0	0.7	0.20	0.00		112.8	1.7	0.13	0.01
PAES		329.6	3.1	0.17	0.01		108.5	2.8	0.13	0.02
PESA2		276.9	11.5	2.08	0.75		66.7	7.8	8.57	6.82
Random		286.3	4.6	1.50	0.22		63.0	3.5	9.12	6.50
eMOEA	BA_1000	659.2	10.6	0.38	0.09	ER_465	185.6	8.7	0.70	0.26
eNSGAI		682.8	0.6	0.23	0.00		225.8	2.0	0.16	0.01
MOCHC		641.1	17.1	0.38	0.18		224.0	2.2	0.13	0.00
NSGAI		682.3	0.9	0.23	0.00		224.2	2.0	0.16	0.01
PAES		626.1	7.1	0.40	0.03		213.2	3.5	0.17	0.01
PESA2		525.9	31.1	3.96	1.79		117.0	17.2	43.64	55.57
Random		529.1	6.6	3.27	0.63		99.1	5.4	57.30	27.77
eMOEA	BA_2500	1476.8	22.6	1.04	0.18	ER_940	307.8	15.1	21.36	32.19
eNSGAI		1580.0	3.5	0.54	0.02		380.4	3.5	1.06	0.09
MOCHC		1469.3	39.7	1.02	0.81		376.9	4.7	0.78	0.06
NSGAI		1585.3	1.8	0.51	0.01		382.6	2.7	0.87	0.06
PAES		787.5	83.5	388.91	396.29		337.2	5.6	1.31	0.19
PESA2		1019.3	68.3	16.94	6.54		129.5	15.8	2111.64	477.68
Random		1003.2	18.9	17.16	3.09		127.0	4.6	2090.59	88.79
eMOEA	BA_5000	2793.5	55.8	1.57	0.24	ER_2343	620.2	12.8	1219.25	124.19
eNSGAI		3015.7	15.8	0.84	0.05		650.5	15.8	1051.85	158.33
MOCHC		2854.9	48.7	1.08	0.19		587.5	16.1	112.86	130.44
NSGAI		3050.3	8.3	0.72	0.01		693.1	13.0	427.60	44.05
PAES		1215.0	114.0	2308.86	1635.00		387.4	7.8	3480.74	167.71
PESA2		1796.6	142.1	39.76	21.20		195.1	18.0	14262.88	1598.92
Random		1681.8	35.9	48.69	10.39		202.1	3.4	13510.86	319.98
eMOEA	FF_250	111.3	3.5	0.29	0.07	WS_250	29.4	2.8	0.51	0.20
eNSGAI		121.1	1.1	0.14	0.01		40.9	1.5	0.14	0.03
MOCHC		121.4	0.6	0.12	0.00		34.8	6.2	0.09	0.05
NSGAI		121.1	0.7	0.14	0.00		39.3	2.8	0.16	0.04
PAES		119.8	1.1	0.13	0.01		32.7	3.1	0.13	0.07
PESA2		75.9	8.7	9.53	7.38		22.8	3.0	3.07	2.18
Random		76.8	3.3	5.69	3.70		18.3	7.6	2.57	1.33
eMOEA	FF_500	237.9	4.3	0.31	0.07	WS_500	93.3	4.9	0.44	0.11
eNSGAI		260.1	0.9	0.13	0.00		117.1	3.0	0.10	0.03
MOCHC		257.5	1.1	0.10	0.00		126.5	2.5	0.04	0.01
NSGAI		260.1	0.7	0.13	0.00		123.5	3.3	0.04	0.02
PAES		249.4	2.6	0.12	0.01		91.4	6.4	0.12	0.06
PESA2		179.4	15.6	3.42	2.01		71.5	7.2	2.19	1.33
Random		173.1	4.7	3.67	1.40		62.4	3.3	3.45	1.34
eMOEA	FF_1000	363.8	11.1	3.41	0.72	WS_1000	107.7	5.3	1.74	0.41
eNSGAI		445.3	1.9	0.86	0.04		130.2	7.8	0.65	0.20
MOCHC		435.6	5.7	0.72	0.05		148.1	6.1	0.48	0.08
NSGAI		446.5	2.9	0.78	0.04		167.8	3.3	0.19	0.01
PAES		398.9	5.7	1.06	0.07		123.1	12.2	0.39	0.12
PESA2		239.0	28.9	216.63	214.92		67.3	12.7	30.16	43.08
Random		199.1	10.1	348.51	144.76		50.5	3.4	72.65	42.52
eMOEA	FF_2000	633.5	25.4	7.25	1.65	WS_1500	185.3	9.8	6.76	1.33
eNSGAI		822.7	8.2	1.57	0.15		226.7	14.7	3.07	0.83
MOCHC		796.9	12.4	1.19	0.10		252.1	11.7	1.27	0.20
NSGAI		834.2	10.2	1.17	0.09		299.4	8.9	0.81	0.10
PAES		599.7	8.4	5.56	0.48		208.0	8.7	1.55	0.22
PESA2		430.7	43.8	283.56	411.07		103.7	20.0	367.34	887.60
Random		327.5	15.2	962.11	534.79		63.5	5.4	1519.00	767.34

In general, the best performance for the spacing measure is problem dependent for constant-weighted networks, with little consistency across various problems. However, for both the log-weighted and random-weighted problem sets, the PESA2 optimizer always attains the best value for the spacing measure, irrespective of the network type. In general, the eMOEA

and eNSGAI algorithms also perform well with respect to the spacing measure. These results strongly indicate that the explicit diversity maintenance strategies employed by the PESA2, eMOEA and eNSGAI algorithms promote well-spread solutions within the Pareto front.

Table 6: Average ranks for the multi-objective measures across the constant-weighted benchmark instances. The ‘BRF’ column indicates the Best Rank Frequency, that is, the number of benchmark problems that the algorithm attained a rank of 1.

Algorithm	Additive Epsilon			Hypervolume			Spacing			
	μ	σ	BRF	μ	σ	BRF	μ	σ	BRF	
Overall	eMOEA	3.700	1.767	1	3.600	1.897	2	2.400	1.897	4
	eNSGAI	2.500	0.527	0	1.900	0.994	4	2.300	1.337	3
	MOCHC	4.500	1.900	1	3.600	2.066	2	5.100	2.079	1
	NSGAI	1.500	1.269	8	1.500	1.269	8	3.500	1.841	1
	PAES	3.300	1.947	3	2.900	1.969	4	4.100	2.183	2
	PESA2	3.500	0.972	0	2.900	1.287	2	2.400	1.075	1
	Random	7.000	0.000	0	5.700	2.497	2	3.200	2.573	5
BA	eMOEA	2.000	1.414	1	2.000	1.414	1	2.000	1.414	1
	eNSGAI	2.500	0.707	0	1.000	0.000	2	2.500	0.707	0
	MOCHC	5.500	0.707	0	3.000	2.828	1	6.500	0.707	0
	NSGAI	1.000	0.000	2	1.000	0.000	2	3.500	0.707	0
	PAES	5.500	0.707	0	3.000	2.828	1	4.000	2.828	0
	PESA2	4.000	0.000	0	2.500	2.121	1	2.500	0.707	0
	Random	7.000	0.000	0	4.000	4.243	1	3.000	2.828	1
ER	eMOEA	3.750	1.708	0	3.500	2.082	1	3.750	2.363	0
	eNSGAI	2.750	0.500	0	2.250	1.500	2	3.250	1.500	0
	MOCHC	3.750	2.630	1	3.000	2.160	1	4.500	1.915	0
	NSGAI	2.250	1.893	2	2.250	1.893	2	4.000	2.449	0
	PAES	2.750	2.363	2	2.250	2.500	3	4.500	2.646	1
	PESA2	3.750	1.500	0	3.250	1.708	1	2.750	1.500	0
	Random	7.000	0.000	0	5.500	3.000	1	1.500	1.000	3
FF	eMOEA	4.500	1.732	0	4.500	1.732	0	1.250	0.500	3
	eNSGAI	2.250	0.500	0	2.000	0.000	0	1.250	0.500	3
	MOCHC	4.750	1.500	0	4.500	1.915	0	5.000	2.708	1
	NSGAI	1.000	0.000	4	1.000	0.000	4	3.000	1.826	1
	PAES	2.750	1.258	1	3.500	1.291	0	3.750	2.062	1
	PESA2	3.000	0.000	0	2.750	0.500	0	2.000	0.816	1
	Random	7.000	0.000	0	6.750	0.500	0	5.000	2.828	1

To provide an overall indicator of performance for the constant-weighted problems, we plot the aggregated set of non-dominated points obtained by each of the optimizers and calculate the hypervolume accordingly using the algorithm of [52]. For the hypervolume calculation, the reference point is taken as the worst observed values for each objective. Furthermore, we generate a reference Pareto front by taking the non-dominated union of each respective Pareto front. Visualizations of the Pareto fronts are provided in Figures 2 to 4 while the hypervolume values can be found in Tables 9 to 11. For clarity (i.e., to prevent skewing of the axes in the Pareto front plots and the values of the hypervolume indicator), results from the random approach

Table 7: Average ranks for the multi-objective measures across the log-weighted benchmark instances. The ‘BRF’ column indicates the Best Rank Frequency, that is, the number of benchmark problems that the algorithm attained a rank of 1.

	Algorithm	Additive Epsilon			Hypervolume			Spacing		
		μ	σ	BRF	μ	σ	BRF	μ	σ	BRF
Overall	eMOEA	4.063	0.998	0	2.813	1.167	2	2.188	1.167	5
	eNSGAI	3.125	1.147	0	2.188	0.750	2	2.000	0.966	6
	MOCHC	1.938	1.124	8	1.750	0.856	8	4.938	1.914	2
	NSGAI	1.688	0.704	7	1.625	1.088	10	2.875	1.586	4
	PAES	3.563	1.459	1	2.500	1.095	3	4.125	1.784	2
	PESA2	6.250	0.683	0	3.125	1.586	2	1.063	0.250	15
	Random	6.063	0.443	0	3.125	1.586	2	4.875	2.500	3
BA	eMOEA	3.750	0.957	0	3.000	1.155	0	1.500	0.577	2
	eNSGAI	2.250	0.500	0	2.000	0.000	0	2.500	1.000	1
	MOCHC	2.750	1.500	1	2.000	0.816	1	5.750	0.957	0
	NSGAI	1.250	0.500	3	1.500	1.000	3	3.250	1.500	1
	PAES	4.750	1.500	0	3.000	1.155	0	5.500	0.577	0
	PESA2	6.000	0.816	0	3.000	1.155	0	1.000	0.000	4
	Random	5.750	0.500	0	3.000	1.155	0	5.250	1.258	0
ER	eMOEA	4.250	1.500	0	2.250	0.500	0	3.500	1.291	0
	eNSGAI	3.250	0.500	0	2.250	0.500	0	1.750	0.957	2
	MOCHC	1.500	1.000	3	1.250	0.500	3	5.750	0.500	0
	NSGAI	2.000	0.816	1	1.750	0.500	1	4.000	1.826	0
	PAES	3.000	1.414	0	2.250	0.500	0	4.250	0.957	0
	PESA2	6.500	0.577	0	2.250	0.500	0	1.250	0.500	3
	Random	6.000	0.000	0	2.250	0.500	0	4.750	2.630	0
FF	eMOEA	4.500	0.577	0	3.750	0.957	0	2.500	0.577	0
	eNSGAI	2.250	0.500	0	2.250	0.500	0	2.500	1.000	0
	MOCHC	2.000	1.155	2	2.000	1.155	2	4.500	1.732	0
	NSGAI	1.500	0.577	2	1.250	0.500	3	3.000	1.155	0
	PAES	4.250	0.500	0	3.500	0.577	0	4.250	1.708	0
	PESA2	5.750	0.500	0	3.750	0.957	0	1.000	0.000	4
	Random	6.500	0.577	0	3.750	0.957	0	7.000	0.000	0
WS	eMOEA	3.750	0.957	0	2.250	1.500	2	1.250	0.500	3
	eNSGAI	4.750	0.500	0	2.250	1.500	2	1.250	0.500	3
	MOCHC	1.500	0.577	2	1.750	0.957	2	3.750	3.202	2
	NSGAI	2.000	0.816	1	2.000	2.000	3	1.250	0.500	3
	PAES	2.250	0.957	1	1.250	0.500	3	2.500	2.380	2
	PESA2	6.750	0.500	0	3.500	2.887	2	1.000	0.000	4
	Random	6.000	0.000	0	3.500	2.887	2	2.500	3.000	3

are omitted in this analysis.

For the BA₅₀₀ benchmark, with the exception of MOCHC, each optimizer ultimately arrives at the same two non-dominated points (see Figure 2a). Given the agreement among the optimizers, it is hypothesized that these two points lie on the true Pareto optimal front for this problem.

With the exception of the ER₂₃₄ benchmark, the overall hypervolume results disagree with those attained using the individual hypervolume values. The inconsistent results on the Erdos-Renyi problems are likely a result of the inherent random structure that provides no exploitable structure for any

Table 8: Average ranks for the multi-objective measures across the random-weighted benchmark instances. The ‘BRF’ column indicates the Best Rank Frequency, that is, the number of benchmark problems that the algorithm attained a rank of 1.

	Algorithm	Additive Epsilon			Hypervolume			Spacing		
		μ	σ	BRF	μ	σ	BRF	μ	σ	BRF
Overall	eMOEA	4.688	0.602	0	2.188	1.047	5	2.313	1.302	7
	eNSGAI	2.688	0.873	1	2.250	1.183	5	1.500	0.730	10
	MOCHC	1.750	0.856	8	1.438	0.512	9	5.375	2.156	2
	NSGAI	1.688	0.704	7	1.500	0.894	11	2.563	1.896	8
	PAES	3.750	1.770	2	2.125	1.088	6	3.125	1.928	6
	PESA2	6.063	0.574	0	2.500	1.461	5	1.063	0.250	15
	Random	6.063	0.574	0	2.500	1.461	5	5.438	1.861	1
BA	eMOEA	4.000	0.816	0	2.000	0.816	1	1.750	1.500	3
	eNSGAI	2.000	0.816	1	2.500	1.732	1	1.000	0.000	4
	MOCHC	2.750	0.500	0	1.750	0.500	1	6.750	0.500	0
	NSGAI	1.500	1.000	3	1.500	1.000	3	3.000	2.309	2
	PAES	5.000	2.828	1	1.500	0.577	2	2.000	2.000	3
	PESA2	5.750	0.957	0	2.750	2.217	1	1.000	0.000	4
	Random	5.500	0.577	0	2.750	2.217	1	5.750	0.500	0
ER	eMOEA	4.750	0.500	0	2.250	0.957	1	2.000	1.155	2
	eNSGAI	2.500	0.577	0	2.250	0.957	1	1.750	0.957	2
	MOCHC	1.250	0.500	3	1.250	0.500	3	5.250	2.872	1
	NSGAI	1.750	0.500	1	2.000	1.414	2	3.250	2.630	2
	PAES	4.250	0.500	0	2.750	1.258	1	2.500	1.915	2
	PESA2	6.250	0.500	0	2.750	1.258	1	1.000	0.000	4
	Random	6.000	0.000	0	2.750	1.258	1	4.500	2.380	0
FF	eMOEA	5.000	0.000	0	2.500	1.291	1	2.250	0.957	1
	eNSGAI	2.500	0.577	0	2.250	0.957	1	1.750	0.500	1
	MOCHC	1.250	0.500	3	2.250	0.957	1	5.000	2.000	0
	NSGAI	1.750	0.500	1	1.500	0.577	2	2.750	1.500	1
	PAES	3.500	1.000	0	1.250	0.500	3	3.000	1.826	1
	PESA2	6.000	0.000	0	2.500	1.291	1	1.250	0.500	3
	Random	6.250	0.500	0	2.500	1.291	1	5.250	2.872	1
WS	eMOEA	5.000	0.000	0	2.000	1.414	2	3.250	1.500	1
	eNSGAI	3.750	0.500	0	2.000	1.414	2	1.500	1.000	3
	MOCHC	1.750	0.957	2	1.250	0.500	3	4.500	2.646	1
	NSGAI	1.750	0.957	2	1.250	0.500	3	1.250	0.500	3
	PAES	2.250	0.957	1	1.750	0.957	2	5.000	0.816	0
	PESA2	6.250	0.500	0	2.000	1.414	2	1.000	0.000	4
	Random	6.500	0.577	0	2.000	1.414	2	6.250	0.957	0

of the optimizers, thereby leading to more variability in the individual fronts attained. In contrast, the overall hypervolume results are agreeable with the individual results for the Forest Fire problems.

Another noteworthy observation is that for nearly all problems, the hypervolume of the reference front is greater than the hypervolume of any one optimizer. This indicates that the reference front consists of points attained by more than one optimizer and thus there is no optimizer that produces a Pareto front that completely dominates all others.

Interestingly, in some cases the observations regarding hypervolume for

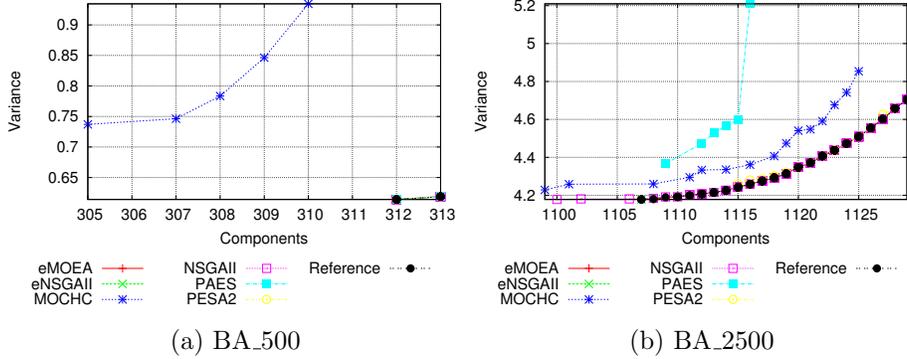


Figure 2: Non-dominated union of obtained Pareto fronts for the constant-weighted Barabasi-Albert benchmark problems.

Table 9: Hypervolume of the non-dominated union of obtained Pareto fronts on the constant-weighted Barabasi-Albert benchmark problems.

Algorithm	BA_500	BA_2500
eMOEA	2.56	27.01
eNSGAI	2.56	27.00
MOCHC	0.62	21.26
NSGAI	2.56	26.98
PAES	2.56	12.57
PESA2	2.56	25.34
Reference	2.56	27.01

the overall set of non-dominated points differ from those attained on comparisons of individual runs. This suggests that the set of solutions attained on any specific run may not be indicative of the overall exploratory power of each optimizer. Moreover, this may be attributed to the lack of diversity in the Pareto front attained on any specific run, however, when considering the overall set of solutions found across multiple runs, we achieve a better representation of the overall search capabilities.

4. Conclusions

In this paper we proposed a bi-objective critical node detection problem (BOCNDP). While it had been recognized previously that the critical node detection problem (CNDP) is inherently bi-objective, this is the first study that formulated the problem as multi-objective. The goals of the BOCNDP are to maximize the number of connected components and minimize the variance of their cardinalities, and we proved the BOCNDP to be distinct from the CNDP.

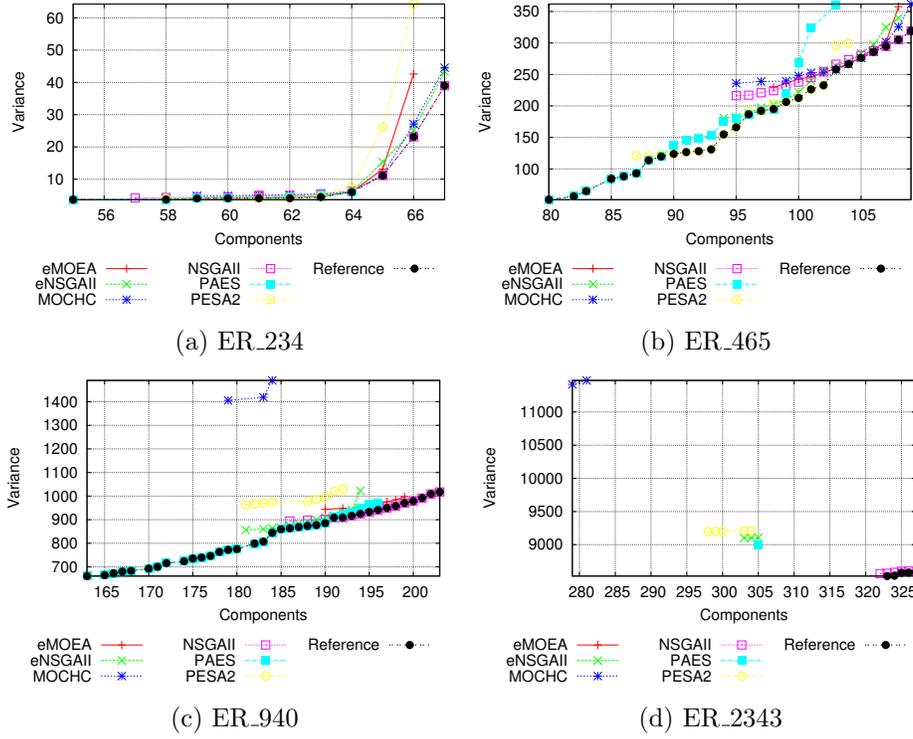


Figure 3: Non-dominated union of obtained Pareto fronts for the constant-weighted Erdos-Renyi benchmark problems.

Table 10: Hypervolume of the non-dominated union of obtained Pareto fronts on the constant-weighted Erdos-Renyi benchmark problems.

Algorithm	ER_234	ER_465	ER_940	ER_2343
eMOEA	612.00	3267.38	19473.84	19715.88
eNSGAI	647.92	4125.17	19057.87	781.99
MOCHC	642.42	3167.43	1642.45	N/A
NSGAI	655.22	3593.59	23026.62	18075.93
PAES	538.61	4514.06	22901.50	N/A
PESA2	575.88	4703.53	15031.79	33.46
Reference	660.32	5445.15	26579.08	19715.88

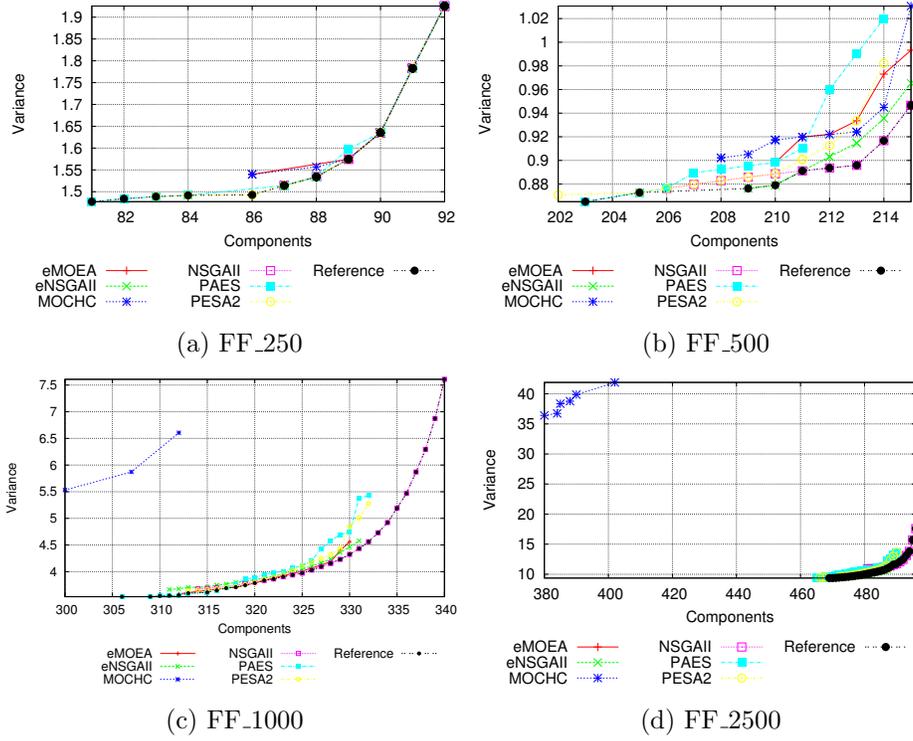


Figure 4: Non-dominated union of obtained Pareto fronts for the constant-weighted Forest Fire benchmark problems.

Table 11: Hypervolume of the non-dominated union of obtained Pareto fronts on the constant-weighted Forest Fire benchmark problems.

Algorithm	FF_250	FF_500	FF_1000	FF_2500
eMOEA	3.41	1.47	114.79	3528.45
eNSGAI	3.64	1.78	115.99	3487.77
MOCHC	3.45	1.42	17.17	37.59
NSGAI	3.64	1.81	134.07	3605.54
PAES	3.70	1.42	117.75	3543.75
PESA2	3.76	1.61	117.89	3543.89
Reference	3.76	1.86	136.17	3727.80

We compared the results using six common multi-objective evolutionary algorithms from two viewpoints: on the attained values of the objectives, and on the ability to approximate the Pareto front. Three well-known multi-objective measures were used to analyze the quality of the BOCNDP solutions from the Pareto front perspective. The multi-objective evolutionary optimizers were tested on a suite of recently proposed benchmark problems, representing a variety of different network properties and sizes. The results indicated that the NSGAI algorithm typically outperformed the other approaches when considering the hypervolume and additive epsilon measures.

Future work includes expanding the number and types of networks, especially to incorporate real-world complex networks. To provide a more comprehensive analysis different search operators should also be considered, in addition to a larger set of multi-objective algorithms and even expanding the definition to other interdiction objectives.

References

- [1] V. Boginski, C. Commander, Identifying critical nodes in protein-protein interaction networks, in: *Clustering Challenges in Biological Networks*, Springer, 2009, pp. 153–166.
- [2] D. Kempe, J. Kleinberg, E. Tardos, Maximizing the spread of influence in a social network, in: *Proceedings of the 9th International Conference on Knowledge Discovery and Data Mining*, 2003, pp. 137–146.
- [3] D. Nguyen, Y. Shen, M. Thai, Detecting critical nodes in interdependent power networks for vulnerability assessment, *IEEE Transactions on Smart Grid* 4 (1) (2013) 151–159.
- [4] K. E. Joyce, P. J. Laurienti, J. H. Burdette, S. Hayasaka, A new measure of centrality for brain networks, *PLoS ONE* 5 (8) (2010) e12200. doi:10.1371/journal.pone.0012200.
- [5] M. Ventresca, D. Aleman, Evaluation of strategies to mitigate contagion spread using social network characteristics, *Social Networks* 35 (1) (2013) 75–88.
- [6] M. Ventresca, D. Aleman, A randomized algorithm with local search for containment of pandemic disease spread, *Computers and Operations Research* 48 (2014) 11–19.

- [7] J. Aspnes, K. Chang, A. Yampolskiy, Inoculation strategies for victims of viruses and the sum-of-squares partition problem, in: Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '05, Society for Industrial and Applied Mathematics, 2005, pp. 43–52.
- [8] J. L. Walteros, P. M. Pardalos, Selected topics in critical element detection, in: Applications of Mathematics and Informatics in Military Science, Vol. 71 of Springer Optimization and Its Applications, Springer New York, 2012, pp. 9–26.
- [9] A. Arulsevan, C. W. Commander, L. Elefteriadou, P. M. Pardalos, Detecting critical nodes in sparse graphs, *Computers and Operations Research* 36 (7) (2009) 2193–2200.
- [10] M. Di Summa, A. Grosso, M. Locatelli, Branch and cut algorithms for detecting critical nodes in undirected graphs, *Computational Optimization and Applications* 53 (3) (2012) 649–680.
- [11] M. Di Summa, A. Grosso, M. Locatelli, Complexity of the critical node problem over trees, *Computers and Operations Research* 38 (12) (2011) 1766–1774.
- [12] B. Addis, M. Di Summa, A. Grosso, Identifying critical nodes in undirected graphs: Complexity results and polynomial algorithms for the case of bounded treewidth, *Discrete Applied Mathematics* 161 (16-17) (2013) 2349–2360.
- [13] A. Veremyev, V. Boginski, E. L. Pasiliao, Exact identification of critical nodes in sparse networks via new compact formulations, *Optimization Letters* 8 (4) (2014) 1245–1259.
- [14] A. Veremyev, O. A. Prokopyev, E. L. Pasiliao, An integer programming framework for critical elements detection in graphs, *Journal of Combinatorial Optimization* 28 (1) (2014) 233–273.
- [15] T. N. Dinh, Y. Xuan, M. T. Thai, E. K. Park, T. Znati, On approximation of new optimization methods for assessing network vulnerability, in: INFOCOM, 2010, pp. 2678–2686.
- [16] T. N. Dinh, Y. Xuan, M. T. Thai, P. M. Pardalos, T. Znati, On new approaches of assessing network vulnerability: Hardness and approximation, *IEEE/ACM Transactions on Networking* 20 (2) (2012) 609–619.

- [17] M. Ventresca, D. Aleman, A derandomized approximation algorithm for the critical node detection problem, *Computers and Operations Research* 43 (2014) 261–270.
- [18] M. Ventresca, Global search algorithms using a combinatorial unranking-based problem representation for the critical node detection problem, *Computers and Operations Research* 39 (11) (2012) 2763–2775.
- [19] B. Addis, R. Aringhieri, A. Grosso, P. Hosteins, Hybrid constructive heuristics for the critical node problem, *Annals of Operations Research* 238 (1) (2016) 637–649.
- [20] R. Aringhieri, A. Grosso, P. Hosteins, R. Scatamacchia, Local search metaheuristics for the critical node problem, *Networks* 67 (3) (2016) 209–221.
- [21] R. Aringhieri, A. Grosso, P. Hosteins, R. Scatamacchia, VNS solutions for the critical node problem, *Electronic Notes in Discrete Mathematics* 47 (2015) 37–44.
- [22] W. Pullan, Heuristic identification of critical nodes in sparse real-world graphs, *Journal of Heuristics* 21 (5) (2015) 577–598.
- [23] M. Ventresca, D. Aleman, A fast greedy algorithm for the critical node detection problem, in: *Combinatorial Optimization and Applications*, Vol. 8881 of *Lecture Notes in Computer Science*, Springer, 2014, pp. 603–612.
- [24] V. S. A. Kumar, R. Rajaraman, Z. Sun, R. Sundaram, Existence theorems and approximation algorithms for generalized network security games, in: *Proceedings of the 2010 IEEE 30th International Conference on Distributed Computing Systems*, 2010, pp. 348–357.
- [25] P. Chen, M. David, D. Kempe, Better vaccination strategies for better people, in: *Proceedings of the 11th ACM Conference on Electronic Commerce*, ACM, 2010, pp. 179–188.
- [26] N. Garg, V. Vazirani, M. Yannakakis, Primal-dual approximation algorithms for integral flow and multicut in trees, *Algorithmica* 18 (1997) 3–20.
- [27] A. Berger, A. Grigoriev, R. van der Zwaan, Complexity and approximability of the k-way vertex cut, *Networks* 63 (2) (2013) 170–1.

- [28] H. Saran, V. Vazirani, Finding k-cuts within twice the optimal, *SIAM Journal of Computing* 24 (1995) 101–108.
- [29] R. Engelberg, J. Konemann, S. Leonardi, J. Naor, Cut problems in graphs with a budget constraint, in: *Proceedings of the 7th Latin American Theoretical Informatics Symposium*, 2006, pp. 262–279.
- [30] S. Arora, S. Rao, U. Vazirani, Expander flows, geometric embeddings and graph partitioning, in: *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, 2004, pp. 222–231.
- [31] *Networks, Virtual Issue - Network Interdiction Applications and Extensions*, Wiley, 2014.
- [32] S. Shen, J. C. Smith, R. Goli, Exact interdiction models and algorithms for disconnecting networks via node deletions, *Discrete Optimization* 9 (3) (2012) 172–188.
- [33] R. A. Collado, D. Papp, Network interdiction – models, applications, unexplored directions, Tech. Rep. RRR 4-2012, Rutgers University (2010).
- [34] J. Salmeron, Deception tactics for network interdiction: A multiobjective approach, *Networks* 60 (1) (2012) 45–58.
- [35] C. M. Rocco, D. E. Salazar, J. E. Ramirez-Marquez, Multi-objective network interdiction using evolutionary algorithms, in: *Reliability and Maintainability Symposium*, 2009, pp. 170–175.
- [36] Weighted cndp and bocndp benchmark network instances, engineering.purdue.edu/~mventresca/index.php?cnd.
- [37] M. Ventresca, K. R. Harrison, B. M. Ombuki-Berman, An experimental evaluation of multi-objective evolutionary algorithms for detecting critical nodes in complex networks, in: *Proceedings of the 18th European Conference on Applications of Evolutionary Computation*, Springer International Publishing, Copenhagen, Denmark, 2015, pp. 164–176.
- [38] B. S.P., Identifying sets of key players in a network, *Computational and Mathematical Organization Theory* 12 (2006) 21–34.
- [39] U. von Luxburg, A tutorial on spectral clustering, *Statistics and Computing* 17 (4) (2007) 395–416.

- [40] H. Ishibuchi, N. Akedo, H. Ohyanagi, Y. Nojima, Behavior of emo algorithms on many-objective optimization problems with correlated objectives, in: *Evolutionary Computation (CEC), 2011 IEEE Congress on*, 2011, pp. 1465–1472.
- [41] R. L. Moritz, E. Reich, M. Bernt, M. Middendorf, The influence of correlated objectives on different types of p-aco algorithms, in: *Evolutionary Computation in Combinatorial Optimisation*, Vol. 8600 of *Lecture Notes in Computer Science*, Springer, 2014, pp. 230–241.
- [42] MOEA Framework, version 2.1, www.moeaframework.org (2014).
- [43] A. Nebro, E. Alba, G. Molina, F. Chicano, F. Luna, J. Durillo, Optimal antenna placement using a new multi-objective CHC algorithm, in: *9th Annual Conference on Genetic and Evolutionary Computation*, 2007, pp. 876–883.
- [44] J. Knowles, D. Corne, The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation, in: *Proceedings of the 1999 Congress on Evolutionary Computation*, Vol. 1, 1999, pp. 98–105.
- [45] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation* 6 (2) (2002) 182–197.
- [46] J. Kollat, P. Reed, Comparing state-of-the-art evolutionary multi-objective algorithms for long-term groundwater monitoring design, *Advances in Water Resources* 29 (6) (2006) 792–807.
- [47] K. Deb, M. Mohan, S. Mishra, A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions, Tech. rep., IIT-Kanpur (2003).
- [48] D. Corne, N. Jerram, J. Knowles, M. Oates, PESA-II: region-based selection in evolutionary multiobjective optimization, in: *Proceedings of the 2001 Genetic and Evolutionary Computation Conference*, 2001, pp. 283–290.
- [49] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, V. Da Fonseca, Performance assessment of multiobjective optimizers: An analysis and review, *Evolutionary Computation*, *IEEE Transactions on* 7 (2) (2003) 117–132.

- [50] E. Zitzler, L. Thiele, Multiobjective optimization using evolutionary algorithms – a comparative case study, in: *Parallel Problem Solving from Nature PPSN V*, Vol. 1498 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 1998, pp. 292–301.
- [51] J. Schott, Fault tolerant design using single and multicriteria genetic algorithm optimization, Master’s thesis, Massachusetts Institute of Technology (1995).
- [52] C. M. Fonseca, L. Paquete, M. López-Ibáñez, An improved dimension-sweep algorithm for the hypervolume indicator, in: *Proceedings of the 2006 Congress on Evolutionary Computation*, IEEE, 2006, pp. 1157–1163.