Soft Computing (2019) 23:12729–12744
https://doi.org/10.1007/s00500-019-03824-8

METHODOLOGIES AND APPLICATION

The bi-objective critical node detection problem with minimum pairwise connectivity and cost: theory and algorithms

Juan Li[1,2,3] · **Panos M. Pardalos**[4] · **Bin Xin**[1,2,3] · **Jie Chen**[1,2,3]

Published online: 14 February 2019

Abstract
An effective way to analyze and apprehend structural properties of networks is to find their most critical nodes. This paper studies a bi-objective critical node detection problem, denoted as Bi-CNDP. In this variant, we do not make any assumptions on the psychology of decision makers and seek to find a set of solutions which minimize the pairwise connectivity of the induced graph and the cost of removing these critical nodes at the same time. After explicitly stating the formulation of Bi-CNDP, we first prove the NP-hardness of this problem for general graphs and the existence of a polynomial algorithm for constructing the ε-approximated Pareto front for Bi-CNDPs on trees. Then different approaches of determining the mating pool and the replacement pool are proposed for the decomposition-based multi-objective evolutionary algorithms. Based on this, two types of decomposition-based multi-objective evolutionary algorithms (MOEA/D and DMOEA-εC) are modified and applied to solve the proposed Bi-CNDP. Numerical experiments on sixteen famous benchmark problems with random and logarithmic weights are firstly conducted to assess different types of the mating pool and the replacement pool. Besides, computational results between two improved algorithms, i.e., I-MOEA/D and I-DMOEA-εC, demonstrate that they behave differently on these instances and I-DMOEA-εC shows better performance on the majority of test instances. Finally, a decision-making process from the perspective of minimizing the pairwise connectivity of the induced graph given a constraint on the cost of removing nodes is presented for helping decision makers to identify the most critical nodes for further protection or attack.

Keywords Bi-objective critical node detection problems · NP-hardness · ε-Approximation · Decomposition-based multi-objective evolutionary algorithms · Mating pool · Replacement pool

1 Introduction

Common networks such as telecommunication, transportation, power systems, and others are exposed to various threats coming from the environment. Any failure of elements of these networks may lead to a complete or partial halt of their services and result in unexpected consequences (Atputharajah and Saha 2009; Liang et al. 2017). An effective way to analyze and apprehend structural properties of networks is to find the most critical nodes. A node is critical if its failure or removal significantly degrades the performance of a network. Once identified, the critical node can

Communicated by V. Loia.

✉ Bin Xin
brucebin@bit.edu.cn

Juan Li
00134476@bit.edu.cn

Panos M. Pardalos
pardalos@ufl.edu

Jie Chen
chenjie@bit.edu.cn

[1] School of Automation, Beijing Institute of Technology, Beijing 100081, China

[2] State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology, Beijing 100081, China

[3] Beijing Advanced Innovation Center for Intelligent Robots and Systems, Beijing Institute of Technology, Beijing 100081, China

[4] Center for Applied Optimization, Department of Industrial and Systems Engineering, University of Florida, Gainesville 32608, FL, USA

be used either to implement protection or attack strategies for networks. The identification of critical nodes in various networks is a fundamental task. In the literature, this problem has attracted a significant amount of research attention in a number of fields, including social network analysis (Borgatti 2006; Fan and Pardalos 2010; Kempe et al. 2010; Leskovec et al. 2007), transportation networks' vulnerability assessment (Jenelius et al. 2006), power grid construction (Salmeron et al. 2015), network risk management (Arulselvan et al. 2007), epidemic control (Zhou et al. 2006), and network immunization strategies (Arulselvan et al. 2009; Kuhlman et al. 2010). Besides, a number of applications in the military domain have also been researched (Walteros and Pardalos 2012).

In this paper, we are motivated primarily by the critical node detection problem (CNDP) described in Arulselvan et al. (2009). Given a network, the CNDP consists in finding a set of nodes, the deletion of which results in minimizing a connectivity measure in the induced graph. Different connectivity measures can be devised according to specific applications of interest. The choice of different measures typically leads to different optimal solutions, as described in Aringhieri et al. (2016a), Shen and Smith (2012) and Veremyev et al. (2014a). In the literature, three mostly common used connectivity measures are: (1) pairwise connectivity, i.e., the number of pair of nodes connected by a path inside the graph; (2) the size of the largest connected component; and (3) the number of connected components (Aringhieri et al. 2016a). Even though these measures are different and can lead to different optimal solutions, they are not totally unrelated. The ideal situation for minimizing the pairwise connectivity is to obtain the largest number of connected components with the smallest variance in their cardinality. This implies that minimizing the pairwise connectivity is a trade-off between minimizing the cardinality of the largest connected component and maximizing the number of connected components. This paper concentrates on the pairwise connectivity measure which is generally enough to determine which nodes are still connected in the induced network.

Considering one of the above-mentioned connectivity measures, several variants of the CNDP have been investigated. Among them, the first two variants, namely CNP and CC-CNP, are the most common used formulation in the literature. Firstly, the basic CNDP [denoted as CNP in the original (Arulselvan et al. 2009; Tomaino et al. 2012)] aims at minimizing the connectivity measure in the induced graph given a constraint on the maximum number of nodes that can be removed. Secondly, in the cardinality constrained critical node detection problem (CC-CNP) (Arulselvan et al. 2011), the objective is to minimize the number of nodes required to be removed given the maximum size of connected components. It belongs to the connectivity constrained formulation which minimizes the number of deleted nodes in order to meet a threshold on certain connectivity metric. The CNDPs are also related to a variety of graph fragmentation problems in the literature. The vertex separator problem (Balas and Souza 2005) and the k-separator problem (Ben-Ameur et al. 2015) have the most in common with CNDP. Other relevant examples include the minimum contamination problem (Kumar et al. 2010), the sum-of-squares partitioning problem (Aspnes et al. 2005; Chen et al. 2010), and so on. Further information pertaining to graph fragmentation problems can be found in Shen and Smith (2012).

Theoretical studies on combinatorial aspects of CNDP for general graphs were first given in Arulselvan et al. (2009) and Di Summa et al. (2011). As mentioned above, Arulselvan et al. presented two variants of the CNDP, namely CNP (Arulselvan et al. 2009) and CC-CNP (Arulselvan et al. 2011). They also proved the complexity of two variants for general graphs and introduced some heuristic algorithms for solving them. Since then, many studies have been presented depending on different connectivity metrics to be checked in the induced graph. Recently, CNDPs on special classes of graphs including split graphs, bipartite graphs, and graphs of bounded treewidth were considered in Addis et al. (2013). Lalou et al. (2016) proposed a new variant of this problem, called component-cardinality-constrained critical node problem (3C-CNP). This variant seeks to find a minimal set of nodes, removal of which constrains the size of each connected component in the induced graph to a given bound. Aringhieri et al. (2019) studied the distance critical node problem. It is a generalization of the critical node problem where the distances between node pairs impact on the objective function. Dinh and Thai (2013) and Dinh et al. (2010) presented a new formulation of CNDP called the β-vertex separator problem. They studied the complexity and inapproximability on general graphs and proposed a pseudo-approximation method and a heuristic approach to solve this problem on general graphs. Similarly, Shen et al. (2013a, b) provided complexity analysis for CNDPs on general graphs and power-law graphs. Aringhieri et al. (2016a) and Veremyev et al. (2014a) used the above-mentioned three different connectivity measures, took into account both the budget and connectivity constraints, and obtained six different variants of the CNDP.

There are exact and heuristic algorithms proposed for CNDP in the literature. The exact algorithms for CNDP include an Integer Linear Program (ILP) for problems with a potentially non-polynomial number of constraints (Di Summa et al. 2012). Other models with a polynomial number of constraints were also studied in Veremyev et al. (2014a, b). A recent work by Pavlikov (2018) provided a model with a polynomial number of constraints, which has the same linear relaxation as the model of Di Summa et al. (2012). Addis et al. (2013) defined a dynamic programming recursion that solves the problem in polynomial time when

the graph has bounded treewidth and unit connection costs. Besides, an approximation algorithm named the bi-criteria randomized rounding approach was proposed in Ventresca and Aleman (2014). However, these exact methods offer limited applicability since most of them are based on the ILP formulation. Recently, several heuristic algorithms have been proposed for CNDP, for example, multiple greedy constructive heuristics (Addis et al. 2016; Ventresca and Aleman 2015) and local search metaheuristics (Ventresca and Aleman 2014; Aringhieri et al. 2016b). A simulated annealing algorithm and a population-based incremental learning algorithm without approximation bounds were applied to CNDP with up to 5,000 nodes (Ventresca 2012). A fast greedy algorithm has been recently presented for approximating solutions for large-scale networks (Ventresca and Aleman 2014). A variable neighborhood search which outperforms the population-based method (Ventresca 2012) was proposed in Aringhieri et al. (2015). Purevsuren et al. (2017) provided results competitive with those of Aringhieri et al. (2016a) for the single-objective CNDP. Readers can refer to a recent survey of Lalou et al. (2018) for a detailed exposition of CNDP and relevant results in the literature.

Most of the aforementioned formulations and algorithms regard CNDP as single-objective problems. These formulations assume that decision makers either have prior knowledge on the maximum number of nodes that can be removed or have the ability to give a threshold of the induced network connectivity. However, it is not easy for decision makers to gain preference knowledge since they know little about the problem itself, and the CNDP is not an exception. In fact, the connectivity of an induced network and the cost of removing nodes are two conflicting objectives and should be considered simultaneously. Aringhieri et al. (2016a) are the first authors to propose fully bi-objective results for CNDP and display Pareto fronts for some instances. Furthermore, Faramondi et al. (2018) also considered the CNDP as a bi-objective problem and firstly provided an explicitly bi-objective approach, i.e., a multi-objective ant colony optimization algorithm. It should be noted that the CNDP is also recognized as a bi-objective problem in Ventresca et al. (2018), but in a different way. To be specific, Ventresca et al. (2018) tackled a bi-objective CNDP where the two objectives are the number of components and the variance of their cardinality, which is kind of a generalization of pairwise connectivity. This paper studies a bi-objective formulation of CNDP (Bi-CNDP) which considers both the connectivity of an induced network and the cost of removing nodes and presents related theoretical and computational results.

For weighted networks, we assume that each node is assigned with a weighted value which is related to the cost of removing it, and decision makers want to minimize the pairwise connectivity of an induced graph and minimize the cost of removing these nodes at the same time. We first prove the NP-hardness on general graphs and the existence of a polynomial algorithm for constructing an ϵ-approximated Pareto front for CNDPs on trees. Then, different types of mating pools and replacement pools are proposed and embedded in decomposition-based multi-objective evolutionary algorithms (MOEAs). Two state-of-the-art decomposition-based MOEAs including MOEA/D and DMOEA-ϵC are modified and applied to solve the Bi-CNDP. Numerical results on sixteen modified famous benchmark problems are conducted to assess effectiveness of different mating pools and replacement pools. Further computational results demonstrate different performances of two improved MOEAs, i.e., I-MOEA/D and I-DMOEA-ϵC, on solving Bi-CNDP. Finally, a decision-making process from the perspective of minimizing the pairwise connectivity of the induced graph given a constraint on the cost of removing nodes is proposed for helping decision makers to identify the most critical nodes.

The rest of this paper is organized as follows. The preliminary definitions related to CNDP are recalled, and then, the formulation of Bi-CNDP and associated theoretical results is given in Sect. 2. Section 3 briefly describes two decomposition-based MOEAs and various approaches of determining the mating pool and the replacement pool. Numerical results among various variants of two decomposition-based MOEAs with different types of mating pools and replacement pools on sixteen commonly used benchmark problems are shown in Sect. 4. Furthermore, Sect. 4 includes computational experiments between two improved MOEAs on these benchmarks and a decision-making process. Finally, Sect. 5 concludes this paper.

2 Mathematical formulations and theoretical results of Bi-CNDP

This section first recalls some basic definitions related to CNDP and then presents mathematical formulations of Bi-CNDP. Next, some important theoretical results related to the complexity analysis of Bi-CNDP are illustrated.

2.1 Mathematical formulations of Bi-CNDP

Let $G = (V, E, C)$ be a weighted and undirected graph composed of n nodes $V = \{v_1, \ldots, v_n\}$ with weight values related to the cost of removing these nodes $C = \{c_1, \ldots, c_n\}$ and m edges $E = \{(v_i, v_j), i, j = 1, \ldots, n\}$, where $(v_i, v_j) \in E \subseteq V \times V$ captures the existence of a relation between node v_i and node v_j. First, some basic definitions of graphs which will be used in the following are recalled.

Definition 1 (*Path*) A **path** over an undirected graph $G = (V, E, C)$ starting at a node $v_i \in V$ and ending at a node $v_j \in V$ is a subset of links in E that connect nodes v_i and v_j.

Definition 2 (*Connected*) An undirected graph G is **connected** if for each pair of nodes v_i and v_j there is a path over G that connects them.

Definition 3 (*Connected component*) A **connected component** of G is a connected subgraph $G_i = (V_i, E_i)$ such that, over G_i (i.e., $V_i \subseteq V, E_i \subseteq E$), no node in V_i is connected to a node in $V \backslash V_i$.

Here, the pairwise connectivity (PWC) (Arulselvan et al. 2009) is adopted as a measure of connectivity of a graph. The PWC represents the number of distinct node pairs connected by a path over G. To be specific, the definition of PWC(G) is given as follows:

$$\mathrm{PWC}(G) = \frac{1}{2} \sum_{v_i, v_j \in V, v_i \neq v_j} x_{ij} \tag{1}$$

where x_{ij} equals 1 if node v_i and node v_j are connected via a path in G, and it is equal to 0 otherwise. PWC(G) is monotonely non-increasing with respect to edge removals, since the removal of an edge cannot increase the number of pairs of nodes connected by a path. Moreover, $\mathrm{PWC}(G) = \frac{n(n-1)}{2}$ holds for a graph G when all pairs of nodes are connected. Therefore, we define the normalized pairwise connectivity as:

$$n\mathrm{PWC}(G) = \frac{1}{n(n-1)} \sum_{v_i, v_j \in V, v_i \neq v_j} x_{ij} \tag{2}$$

It is obvious that nPWC(G) $\in [0, 1]$. It is straightforward to note that nPWC(G) can be regarded as a measure of the degree of connectivity of G, since nPWC(G) is proportional to the fraction of node pairs that are connected at least via one path. The larger nPWC(G) is, the closer G is to a connected graph.

The first goal of the Bi-CNDP is to determine a subset of nodes $R \subseteq V$ such that the induced residual graph $G(V \backslash R)$ has minimum pairwise connectivity nPWC($G(V \backslash R)$). Apart from minimizing the degree of connectivity of the reduced network after the removal, decision makers also consider minimizing the cost of removing selected nodes. Specifically, the cost of removing selected nodes $R \subseteq V$ from G is given as:

$$n\mathrm{Cost}(R) = \sum\nolimits_{v_i \in R} c_i \Big/ \sum\nolimits_{v_i \in G} c_i \tag{3}$$

where c_i represents the weight value associated with node $v_i \in V$. $\sum_{v_i \in G} c_i$ is introduced so that the two objectives are comparable, as they both have values in [0, 1]. In conclusion, the formulation of Bi-CNDP is presented as:

$$\begin{aligned} \text{minimize } & (n\mathrm{PWC}(G(V \backslash R)), n\mathrm{Cost}(R)) \\ s.t. \quad & x_{ij} + y_i + y_j \geq 1, \forall v_i, v_j \in V \\ & x_{ij} + x_{jk} + x_{ki} \neq 2, \forall v_i, v_j, v_k \in V \\ & x_{ij} \in \{0, 1\}, y_i \in \{0, 1\}, \forall v_i, v_j \in V \end{aligned} \tag{4}$$

where $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ is the decision vector, y_i is a Boolean variable, $y_i = 1$ if node $v_i \in V$ is removed from the original graph and $y_i = 0$ otherwise. x_{ij} represents the connectivity between node v_i and v_j, and it equals to 1 if node v_i and node v_j are in the same connected component of $G(V \backslash R)$; otherwise it equals to 0.

The first constraint enforces the separation of nodes to different components. To be specific, deleted nodes do not share an edge to any other node by setting edge $x_{ij} = 0$ as deleted if either or both nodes v_i and v_j are deleted. The second constraint is concerned with a triangle inequality. That is to say, if nodes v_i and v_j, v_j and v_k are in the same component, then nodes v_i and v_k must be in the same component. In this formulation, there is no need to specify a hierarchy between two objectives nor to gain prior information about the psychology of decision makers.

2.2 Theoretical results on Bi-CNDP

Firstly, we transform the Bi-CNDP into a single-objective formulation (denoted as S-CNDP), as shown in (5) by converting the first objective function into an additional constraint function. Then, we can get the following results.

$$\begin{aligned} \text{minimize } & n\mathrm{PWC}(G(V \backslash R)) \\ \text{s.t.} \quad & n\mathrm{Cost}(R) \leq C, \forall v_i, v_j \in V \\ & x_{ij} + y_i + y_j \geq 1, \forall v_i, v_j \in V \\ & x_{ij} + x_{jk} + x_{ki} \neq 2, \forall v_i, v_j, v_k \in V \\ & x_{ij} \in \{0, 1\}, y_i \in \{0, 1\}, \forall v_i, v_j \in V \end{aligned} \tag{5}$$

Theorem 1 The Bi-CNDP is strongly NP-hard on general graphs.

The proof of Theorem 1 is straightforward. Since the S-CNDP formulated in (5) is just the K-CNP described in Arulselvan et al. (2009), the S-CNDP is strongly NP-complete on general graphs (Addis et al. 2013). Furthermore, we know on general grounds that when the complexity status of an ε-constrained version of a multi-objective problem is demonstrated, the multi-objective problem can only be harder to solve, which implies that the Bi-CNDP is strongly NP-hard. Indeed if it were not, a solution for each ε-constrained version could be obtained in (pseudo-)polynomial time, which would imply that S-CNDP is not NP-hard. This

leads to a contradiction. Hence, the Bi-CNDP is strongly NP-hard on general graphs.

The subclass of CNDP over trees was studied in Di Summa et al. (2011). It has been proved that CNDP over trees is still NP-complete when general connection costs are specified. However, the cases where all connections have unit cost are solvable in polynomial time by dynamic programming approaches. Besides, a dynamic programming recursion that solves the problem in polynomial time when the graph has bounded treewidth was proposed in Addis et al. (2013). Besides, the complexity of removing larger node structures has also received significant attention. Granata et al. (2013) introduced the concept of a critical disruption path which means a path between a source and a destination vertex whose deletion minimizes the cardinality of the largest remaining connected component. The proposed network interdiction model seeks to optimally disrupt network operations. In Walteros et al. (2018), authors have proved that the problems of removing critical cliques, stars, and connected subgraphs are all strongly NP-complete.

Finding all Pareto optimal solutions is often computationally problematic for multi-objective discrete optimization, since there are usually exponentially (or infinite) large Pareto optimal solutions. Furthermore, for the simplest problems with two objectives, determining whether a point belongs to the Pareto optimal set is NP-hard (Papadimitriou and Yannakakis 2000). One way to handle these problems is to introduce the ε-approximated Pareto set $P_\varepsilon(x)$ whose definition is given as follows.

Definition 4 (ε-*approximated Pareto set* $P_\varepsilon(x)$) Given a scalar $\varepsilon > 0$, an ε-approximate Pareto optimal set, denoted by $P_\varepsilon(x)$, is a subset of X such that there is no other solution $y \in X$ such that $(1+\varepsilon)\cdot f_i(y) \le f_i(x)$ for all $x \in P_\varepsilon(x)$ and for some $i \in \{1, 2, \ldots, m\}$, where X is a nonempty feasible set for a certain MOP and m represents the number of objectives of the MOP.

This definition says that every other solution is almost dominated by some solution in $P_\varepsilon(x)$, i.e., there is a solution in $P_\varepsilon(x)$ that is within a factor of ε in all objectives. According to theoretical results stated in Papadimitriou and Yannakakis (2000) and the fact that the CNDP is polynomially solvable on trees via dynamic programming (Di Summa et al. 2011), we can have the following statement.

Theorem 2 For CNDPs on trees, there is a polynomial algorithm in n and $1/\varepsilon$ for constructing the approximate Pareto curve $P_\varepsilon(x)$ for the Bi-CNDP formulated in (4), where n represents the size of an instance of the Bi-CNDP.

However, there is no such property for Bi-CNDPs on general graphs. Thus, the exponential size of feasible solutions for Bi-CNDPs on general graphs calls for the use of heuristic algorithms to find Pareto optimal solutions for the proposed Bi-CNDP.

3 Improved decomposition-based multi-objective evolutionary algorithms

Among various heuristic algorithms, the decomposition-based MOEAs have gained much attention during these decades. This paper modifies two decomposition-based MOEAs to deal with Bi-CNDP. Thus, this section presents a brief description of two decomposition-based MOEAs and their variants which will all be used in the next section.

Decomposition is an efficient and prevailing strategy for solving multi-objective problems (MOPs). In decomposition-based MOEAs, an MOP is decomposed into a number of scalar subproblems by using various scalarizing functions. The weighting method, Tchebycheff approach, boundary intersection method, and the ε-constraint method are classical generation methods in the field of mathematical programming and have been adopted for the multi-objective optimization. Zhang and Li (2007) adopted the first three aggregation functions and proposed the multi-objective evolutionary algorithm based on decomposition (MOEA/D). Many variants such as MOEA/D-DRA (Zhang et al. 2009), MOEA/D-AWA (Qi et al. 2014), and so on have been investigated.

Recently, a new MOEA named the decomposition-based multi-objective evolutionary algorithm with the ε-constraint framework (DMOEA-εC) was proposed in Chen et al. (2017) and has demonstrated its superiority over a number of MOEAs. DMOEA-εC firstly incorporates the ε-constraint method into the decomposition strategy and decomposes an MOP into a series of scalar constrained optimization subproblems by assigning each subproblem with an upper bound vector. In decomposition-based MOEAs, all subproblems are optimized simultaneously by only using information from neighboring subproblems. The decomposition-based MOEAs have lower computational complexity than Pareto-based and indicator-based MOEAs. Besides, an external archive population EP is added to store non-dominated solutions found so far. Details of MOEA/D and DMOEA-εC can be found in Zhang and Li (2007) and Chen et al. (2017), respectively.

Performances of these two MOEAs have been witnessed on continuous and discrete benchmark problems, but not on Bi-CNDP. Thus, it is reasonable to adopt them to solve the proposed Bi-CNDP. When applying two MOEAs to Bi-CNDP, there are two important issues that need to be discussed: (1) How to determine the mating pool for selecting parent candidates? (2) How to determine the replacement pool to replace old candidates after a new candidate

is generated? The above-mentioned two issues will be discussed in the next two subsections.

3.1 The mating pool

Given a certain recombination operator, the mating pool plays a vital role in generating new solutions. In this paper, for certain solutions of a subproblem, we discuss four types of mating pools:

(1) -N: the set of solutions of neighboring subproblems of a subproblem serves as the mating pool, and parent solutions are randomly selected from the mating pool.

(2) -P: the whole population is regarded as the mating pool, and parent solutions are randomly selected from the whole population.

(3) -NP: the parent solutions are selected from the set of solutions of neighboring subproblems of a subproblem with a probability δ. Parent solutions are selected from the whole population with a probability $1-\delta$.

(4) -NP-EP: one of parent solutions is selected randomly from the external archive population EP, and the other one is selected according to the third approach.

The four ways of determining the mating pool will be added into MOEA/D, thus obtaining four variants including MOEA/D-N, MOEA/D-P, MOEA/D-NP, and MOEA/D-NP-EP. Similarly, four variants of DMOEA-εC, i.e., DMOEA-εC-N, DMOEA-εC-P, DMOEA-εC-NP, and DMOEA-εC-NP-EP can be developed.

3.2 The replacement pool

When a new candidate is generated, it will be used to compare with and update other old candidates in the replacement pool. For a new solution of one subproblem, we investigate effects of two types of replacement pools in decomposition-based MOEAs.

(1) -L: a newly generated candidate is compared with the set of solutions of neighboring subproblems of a subproblem, thus the replacement takes place locally.

(2) -G: the whole population serves as the replacement pool, which implies a global replacement strategy.

Two types of replacement pool will be integrated into MOEA/D, thus obtaining its variants, i.e., MOEA/D-L and MOEA/D-G. Similarly, two variants of DMOEA-εC including DMOEA-εC-L and DMOEA-εC-G are designed.

4 Numerical experiments

This section is devoted to experimental design for demonstrating the overall quality of solutions found by two MOEAs and their variants over four sets of sixteen benchmark instances proposed in Ventresca (2012). First, details of the sixteen benchmark problems are outlined, then parameter settings are provided. Finally, the experimental results are illustrated. We compare results obtained via two MOEAs and their variants to evaluate the performance of different strategies of determining the mating pool and the replacement pool. Further experiments are conducted to demonstrate different performance of two improved MOEAs on solving Bi-CNDP. Additionally, a decision-making process based on obtained non-dominated solutions is illustrated.

4.1 Benchmark problems

The benchmark set is composed of the graphs proposed in Ventresca (2012), and many results are available for these graphs as single-objective problems. This data set contains sixteen undirected, unweighted graphs belonging to four groups which are created by complex network generator algorithms. Barabasi–Albert (BA) graphs are scale-free networks and proved to be the easiest to process. While the Watts–Strogatz (WS) graphs are designed to mimic a small-world structure with a denser structure, they turn out to be the most challenging ones. Erdos–Renyi (ER) graphs are random graphs and Forest–Fire (FF) graphs reproduce the behavior of how a fire spreads through a forest. None of these graphs is expected to reproduce a real network. However, real networks usually display a mixture of these characteristics. Further information about these networks can be found in Ventresca (2012)[1].

In order to characterize these graphs precisely, Table 1 displays the following quantities: the number of nodes n, the number of edges m, the average degree $<d> = 2 \cdot m \Big/ n$, the number of articulation points nAP[2], the value of the clustering coefficient CC, the average shortest path length D (Aringhieri et al. 2016a), the number of nodes having degree 1 $|D_1|$, and the number of nodes which are neighbors of those in D_1 (Veremyev et al. 2014b), denoted as $|N(D_1)|$ (Aringhieri et al. 2016a). The number of articulation points (nAP) is taken into account since a larger fraction of articulation points usually results in a graph which is easier to fragment. The clustering coefficient CC signals the tendency of nodes to be clustered together. The average shortest path length D indicates the average distance between two nodes taken at random inside the graph.

[1] The numbers of nodes and edges of ER, WS, and FF graphs in the main body of Ventresca (2012) are slightly different from the data set obtained from the website given in Ventresca (2012). Here, these characteristics are given based on the data set downloaded from the website.

[2] A vertex in an undirected connected graph is an articulation point if and only if removing it disconnects the graph. Articulation points represent vulnerabilities in a connected network.

Table 1 Main characteristics of sixteen benchmark instances

Instance	n	m	$<d>$	nAP	CC	D	$\|D_1\|$	$N(D_1)$
BA500	500	499	1.996	164	0.000	5.663	336	149
BA1000	1000	999	1.998	324	0.000	6.045	676	290
BA2500	2500	2499	1.999	825	0.000	6.901	1675	729
BA5000	5000	4999	1.999	1672	0.000	8.380	3328	1475
WS250	250	1246	9.968	0	0.473	3.327	0	0
WS500	500	1496	5.984	0	0.420	5.304	0	0
WS1000	1000	4996	9.992	0	0.483	4.444	0	0
WS1500	1500	4498	5.997	0	0.480	7.554	0	0
ER235	235	350	2.979	48	0.006	5.339	39	37
ER466	466	700	3.004	84	0.002	5.974	69	64
ER941	941	1400	2.976	177	0.005	6.559	147	139
ER2344	2344	3500	2.986	419	0.001	7.516	396	354
FF250	250	514	4.112	83	0.276	4.816	57	50
FF500	500	828	3.312	195	0.247	6.026	160	136
FF1000	1000	1817	3.634	362	0.216	6.173	280	236
FF2000	2000	3413	3.413	725	0.245	7.587	552	477

Since these networks are unweighted in the original, new benchmark instances are created by assigning a weight value to each node of each network. The weight value of each node is regarded as the cost of removing it. We adopt the weight generation method used in Ventresca et al. (2018), as described in the following:

(1) Weights are randomly assigned, where $c(v) \in [0.2, 3]$, $\forall v \in V$;

(2) Weights are logarithmically assigned with node degree d_v, where $c(v) = \log(d_v) + 0.5, \forall v \in V$.

4.2 Parameter settings

For fair comparison, the choice of parameters remains the same for two MOEAs. Specifically, we adopt binary vectors as encoding schemes for solutions. The population size N is set to 300, 400, 500, and 600 for benchmark problems whose number of nodes is $n \leq 500$, $500 < n \leq 1000$, $1000 < n \leq 2500$, and $2500 < n \leq 5000$, respectively. For fair comparison, an external population with the size of $S = \lfloor 1.5 \cdot N \rfloor$ is added to each algorithm, where $\lfloor \cdot \rfloor$ returns the nearest integer in the direction of negative infinity.

Besides, the parameterized uniform crossover (Spears and Jong 1991) and random mutation (Arulselvan et al. 2011) are adopted in generating new solutions. Moreover, control parameters for these reproduction operators are the same as those used in Arulselvan et al. (2011). To be specific, the biased probability of crossover is set as 0.65 and the random mutation probability for each variable of a solution is set as 0.03. Figure 1 provides an example of the parameterized uniform crossover and the random mutation where the number of nodes is 5.

For MOEA/D, DMOEA-εC and their variants, the neighborhood size is $T = \lfloor 0.1 \cdot N \rfloor$, the probability of selecting mate solutions from neighborhood is $\delta = 0.9$, the maximal number of replacement is $n_r = \lfloor 0.01 \cdot N \rfloor$. Inspired by Ventresca (2012), the maximum number of iterations I is set as 2500, 4000, 6000, and 7500 for test instances whose number of nodes n is $n \leq 500$, $500 < n \leq 1000$, $1000 < n \leq 2500$, and $2500 < n \leq 5000$, respectively. For DMOEA-εC and its variants, the iteration interval of alternating the main objective function IN_m is set to $\lfloor 20\% \cdot I \rfloor$. Both algorithms stop when the number of iterations reaches the maximum number, and each algorithm is executed 20 times independently on each instance.

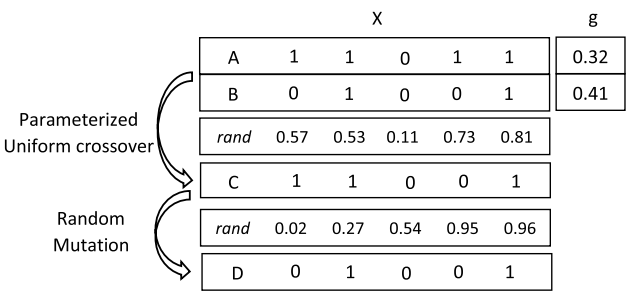

Fig. 1 An example of the parameterized uniform crossover and the random mutation where the number of nodes is 5. A and B are parent solutions. C and D are offspring solutions after crossover and mutation, respectively. *rand* denotes a uniformly randomly distributed value in [0, 1] and g represents the objective value of each solution (i.e., subproblem)

Table 2 Statistical results (Mean[Rank](Std.)) of MOEA/D and its variants with different types of mating pools over 20 independent runs on the sixteen instances with random weights in terms of IGD metrics

Instance	MOEA/D-N	-P	-NP	-NP-EP
BA500	3.80E−03[3](1.57E−04)	3.86E−03[4](1.70E−04)	3.51E−03[2](2.26E−04)	**3.31E−03[1](4.60E−04)**
BA1000	2.58E−02[3](1.05E−03)	2.97E−02[4](8.12E−03)	1.74E−02[2](1.92E−03)	**1.50E−02[1](3.28E−03)**
BA2500	5.63E−04[3](1.93E−04)	5.77E−04[4](1.25E−04)	5.36E−04[2](1.06E−04)	**5.24E−04[1](7.50E−04)**
BA5000	8.64E−03[3](7.33E−04)	8.82E−03[4](6.29E−04)	8.59E−03[2](3.09E−04)	**8.29E−03[1](7.35E−04)**
WS250	3.53E−02[3](6.87E−03)	3.68E−02[4](2.63E−03)	**2.94E−02[1](1.47E−03)**	3.15E−02[2](1.29E−03)
WS500	1.88E−02[3](1.40E−03)	1.96E−03[4](4.59E−03)	**1.45E−02[1](3.45E−03)**	1.47E−02[2](4.78E−03)
WS1000	2.16E−02[3](1.14E−03)	2.24E−02[4](1.85E−03)	1.77E−02[2](1.62E−03)	**1.71E−02[1](1.47E−03)**
WS1500	5.54E−02[3](3.40E−03)	5.60E−02[4](6.06E−03)	**5.20E−02[1](4.12E−03)**	5.42E−02[2](3.50E−03)
ER235	3.75E−02[2](1.08E−03)	4.07E−02[4](2.62E−03)	3.96E−02[3](3.75E−03)	**3.55E−02[1](2.13E−03)**
ER466	9.82E−02[4](5.60E−03)	9.70E−02[3](6.49E−03)	9.26E−03[2](5.12E−03)	**9.22E−03[1](7.89E−03)**
ER941	3.56E−02[3](2.01E−03)	3.73E−02[4](2.81E−03)	3.33E−02[2](1.11E−03)	**3.14E−02[1](1.04E−03)**
ER2344	3.19E−02[3](2.09E−03)	3.26−E02[4]1.99E−03)	**2.88E−02[1](1.50E−03)**	2.92E−02[2](1.63E−03)
FF250	4.79E−02[3](4.71E−03)	5.41E−02[4](2.91E−03)	4.71E−02[2](4.02E−03)	**4.63E−02[1](3.04E−03)**
FF500	4.38E−02[2](3.03E−03)	5.08E−02[4](1.64E−03)	4.76E−02[3](1.08E−03)	**4.29E−02[1](1.30E−03)**
FF1000	3.31E−02[3](1.73E−03)	3.83E−01[4](1.31E−03)	2.75E−02[2](1.97E−03)	**2.61E−02[1](1.08E−03)**
FF2000	2.89E−02[4](1.05E−03)	2.88E−02[3](6.28E−03)	**2.54E−02[1](2.40E−03)**	2.60E−02[2](1.89E−03)
Rank	3.000	3.875	1.813	1.313

Table 3 Statistical results (Mean[Rank](Std.)) of DMOEA-εC-NP-EP and its variants with different types of replacement pools over 20 independent runs on the sixteen instances with logarithmic weights in terms of HV metrics

Instance	DMOEA-εC-NP-EP-L	-G
BA500	1.71E−01[2](1.41E−02)	**1.89E−01[1](1.77E−02)**
BA1000	5.58E−04[2](3.45E−05)	**5.95E−04[1](1.70E−05)**
BA2500	**4.68E−05[1](1.17E−05)**	4.54E−05[2](1.01E−05)
BA5000	6.37E−05[2](1.78E−05)	**6.58E−05[1](2.08E−05)**
WS250	4.16E−01[2](6.41E−02)	**4.49E−01[1](2.97E−02)**
WS500	2.46E−01[2](4.03E−02)	**2.77E−01[1](3.71E−02)**
WS1000	6.11E−02[2](5.67E−03)	**6.51E−02[1](2.36E−03)**
WS1500	7.10E−02[2](5.14E−03)	**7.73E−02[1](3.93E−03)**
ER235	5.37E−02[2](2.36E−03)	**5.78E−02[1](1.44E−03)**
ER466	2.01E−01[2](1.62E−02)	**2.16E−01[1](2.73E−02)**
ER941	**1.73E−02[1](5.03E−03)**	1.68E−02[2](4.89E−03)
ER2344	**6.23E−04[1](9.39E−05)**	6.01E−04[2](8.81E−05)
FF250	5.25E−01[2](2.30E−02)	**5.62E−01[1](3.84E−02)**
FF500	**6.21E−02[1](1.41E−03)**	6.09E−02[2](2.77E−03)
FF1000	7.35E−03[2](7.04E−03)	**7.74E−03[1](4.58E−03)**
FF2000	9.43E−03[2](8.01E−04)	**9.89E−03[1](7.23E−04)**
Rank	1.750	1.250

4.3 Experimental results

This section includes two parts. The first part compares the performance of different variants of two decomposition-based MOEAs, and the second part compares performance of two identified MOEAs obtained according to the experimental results of the first part.

Two commonly used performance metrics, i.e., inverted generational distance (IGD) (Zhou et al. 2005) and hypervolume (HV) (Zitzler and Thiele 1999) are employed to evaluate the performance of compared algorithms.

The IGD metric measures the average distance from a set of uniformly distributed Pareto optimal points over the Pareto front (PF) P^* to the approximation set P. It can be formulated as:

$$\mathrm{IGD}(P^*, P) = \frac{\sum_{x^* \in P^*} d(x^*, P)}{|P^*|} \tag{6}$$

where $d(x^*, P)$ is the minimal Euclidean distance between x^* and any point in P, and $|P^*|$ denotes the cardinality of P^*. If $|P^*|$ is large enough to represent the PF very well, $\mathrm{IGD}(P^*, P)$ could measure both diversity and convergence of P in a sense. To get a smaller IGD value, P must be close to the PF and cover most of the whole PF. Thus, a smaller IGD value indicates a better P.

The HV metric measures the size of the objective space dominated by the solutions in P and bounded by the reference point $\mathbf{r}$. It is defined as:

$$\mathrm{HV}(P, \mathbf{r}) = \mathrm{VOL}\left(\bigcup_{x \in P} [f_1(x), r_1] \times \cdots \times [f_m(x), r_m] \right) \tag{7}$$

where $\mathbf{r} = (r_1, \ldots, r_m)$ is a reference point in the objective space dominated by any Pareto optimal point, and

Table 4 Statistical results (Mean(Std.)) of two improved MOEAs over 20 independent runs on the sixteen instances with random and logarithmic weights in terms of IGD metrics

Instance	I-MOEA/D Random weights	I-DMOEA-εC	I-MOEA/D Logarithmic weights	I-DMOEA-εC
BA500	3.19E-03(1.79E−04)	**1.74E−04**†(5.29E−04)	3.41E−02(1.39E−03)	**2.13E−04**†(1.26E−05)
BA1000	1.57E-02(7.84E−03)	**5.76E−04**†(6.01E−05)	6.16E−03(6.88E−04)	**4.21E−04**†(1.01E−05)
BA2500	4.74E-04(2.21E−05)	**4.64E−04**≈(3.02E−05)	2.77E−02(1.86E−03)	**6.19E−04**†(3.72E−05)
BA5000	7.57E-03(8.13E−04)	**7.34E−04**†(1.30E−05)	6.05E−03(5.38E−04)	**5.48E−04**†(5.53E−05)
WS250	2.23E-02(1.02E−03)	**9.36E−04**†(1.90E−05)	2.28E−02(1.40E−03)	**4.82E−04**†(3.47E−05)
WS500	1.50E-02(1.39E−03)	**6.23E−04**†(3.78E−05)	7.86E−03(2.66E−03)	**6.16E−03**≈(5.53E−04)
WS1000	1.65E-02(1.72E−03)	**1.40E−03**†(6.18E−04)	9.84E−03(8.79E−04)	**8.15E−04**†(6.76E−05)
WS1500	4.89E-02(2.72E−03)	**4.27E−03**†(2.18E−04)	7.15−03(4.77E−04)	**6.42E−04**†(4.61E−05)
ER235	3.52E-02(1.89E−03)	**2.64E−04**†(3.47E−05)	2.19E−02(5.90E−03)	**5.92E−04**†(7.76E−05)
ER466	8.79E-03(8.40E−04)	**8.64E−04**†(5.81E−05)	1.42E−02(1.12E−03)	**2.62E−03**†(1.40E−04)
ER941	2.70E-02(2.29E−03)	**4.62E−03**†(5.73E−04)	2.35E−03(2.91E−04)	**2.52E−03**≈(1.63E−04)
ER2344	2.28E-02(3.79E−03)	**6.99E−03**†(5.31E−04)	7.54E−03(8.93E−04)	**6.17E−04**†(6.04E−05)
FF250	5.20E-02(2.24E−03)	**9.84E−04**†(6.48E−05)	5.19E−02(5.23E−03)	**4.10E−03**†(2.55E−04)
FF500	3.75E-02(4.71E−03)	**2.73E−03**†(7.68E−04)	1.30E−02(1.43E−03)	**9.19E−04**†(1.30E−05)
FF1000	2.28E-02(7.25E−03)	**1.33E−03**†(4.03E−04)	9.05E−02(6.20E−03)	**6.81E−03**†(1.64E−04)
FF2000	2.58E-02(3.50E−03)	**2.81E−03**†(1.92E−04)	3.24E−02(4.18E−03)	**4.46E−03**†(5.11E−04)

Table 5 Statistical results (Mean(Std.)) of two improved MOEAs over 20 independent runs on the sixteen instances with random and logarithmic weights in terms of HV metrics

Instance	I-MOEA/D random weights	I-DMOEA-εC	I-MOEA/D logarithmic weights	I-DMOEA-εC
BA500	9.15E−03(3.78E−04)	**6.81E-02† (2.12E-03)**	5.25E−02(3.31E−03)	**1.89E−01† (2.71E−02)**
BA1000	8.81E−05(1.13E−05)	**4.33E−04† (2.29E−05)**	1.39E−04(1.67E−05)	**5.95E−04† (6.61E−05)**
BA2500	8.87E−06(8.90E−06)	**4.55E−05† (3.87E−06)**	**5.59E−05(6.57E−06)**	4.54E−05§(7.12E−06)
BA5000	6.48E−05(8.76E−05)	**7.56E−05† (6.45E−06)**	4.99E−05(1.50E−06)	**6.58E−05≈ (4.17E−06)**
WS250	3.44E−01(2.29E−02)	**4.04E−01† (1.86E−02)**	3.41E−01(2.41E−02)	**4.49E−01† (1.79E−02)**
WS500	2.89E−01(3.77E−02)	**3.57E−01† (2.91E−02)**	1.58E−01(3.32E−02)	**2.77E−01† (1.51E−02)**
WS1000	**3.95E−02(5.69E−03)**	3.17E−02§(7.54E−03)	**1.75E−01(3.74E−02)**	6.51E−02§(1.38E−03)
WS1500	6.61E−02(7.17E−03)	**7.67E−02† (9.54E−04)**	2.54E−02(3.12E−03)	**7.73E−02† (4.91E−03)**
ER235	2.98E−01(1.97E−02)	**4.23E−01† (5.16E−02)**	3.05E−01(4.18E−02)	**5.78E−01† (4.63E−02)**
ER466	1.16E−01(1.53E−02)	**3.45E−01† (2.79E−02)**	1.24E−01(1.13E−02)	**2.16E−01† (2.34E−02)**
ER941	2.48E−02(4.94E−03)	**4.89E−02† (3.68E−03)**	9.53E−03(8.73E−04)	**1.68E−02† (1.29E−03)**
ER2344	5.41E−05§(7.84E−06)	**2.25E−04(6.17E−05)**	**6.47E−04(1.15E−05)**	6.01E−04≈(2.23E−05)
FF250	1.37E−01(5.51E−02)	**4.22E−01† (3.38E−02)**	2.68E−01(1.36E−02)	**5.62E−01† (4.81E−02)**
FF500	7.11E−02(2.79E−03)	**1.19E−01† (1.54E−02)**	3.88E−02(3.71E−03)	**6.09E−02† (7.46E−03)**
FF1000	4.96E−03(1.56E−04)	**1.72E−02† (3.06E−03)**	3.77E−03(7.26E−04)	**7.74E−03† (6.66E−04)**
FF2000	3.86E−04(3.45E−05)	**5.67E−04† (1.19E−05)**	4.22E−03(2.13E−04)	**9.89E−03† (6.41E−04)**

VOL(·) denotes the Lebesgue measure. Mathematically, for each member in the non-dominated set P, a hypercube $v = [f_1(x), r_1] \times \cdots \times [f_m(x), r_m]$ is constructed with a reference point $\mathbf{r} = (r_1, \ldots, r_m)$ and each member $[f_1(x), \ldots, f_m(x)]$ as the diagonal vertices of the hypercube. The reference point can be found by constructing a vector of the worst objective function values. A larger HV value implies a better P. IGD and HV both take into account diversity and convergence and provide general quality measures of the approximation set P.

In calculating performance metrics, N non-dominated solutions are selected from the external population using the crowding distance approach (Deb et al. 2002). With the purpose of calculating the IGD metric value, P^* is chosen to be the set of non-dominated solutions extracted from the combination of all solutions obtained via two MOEAs and

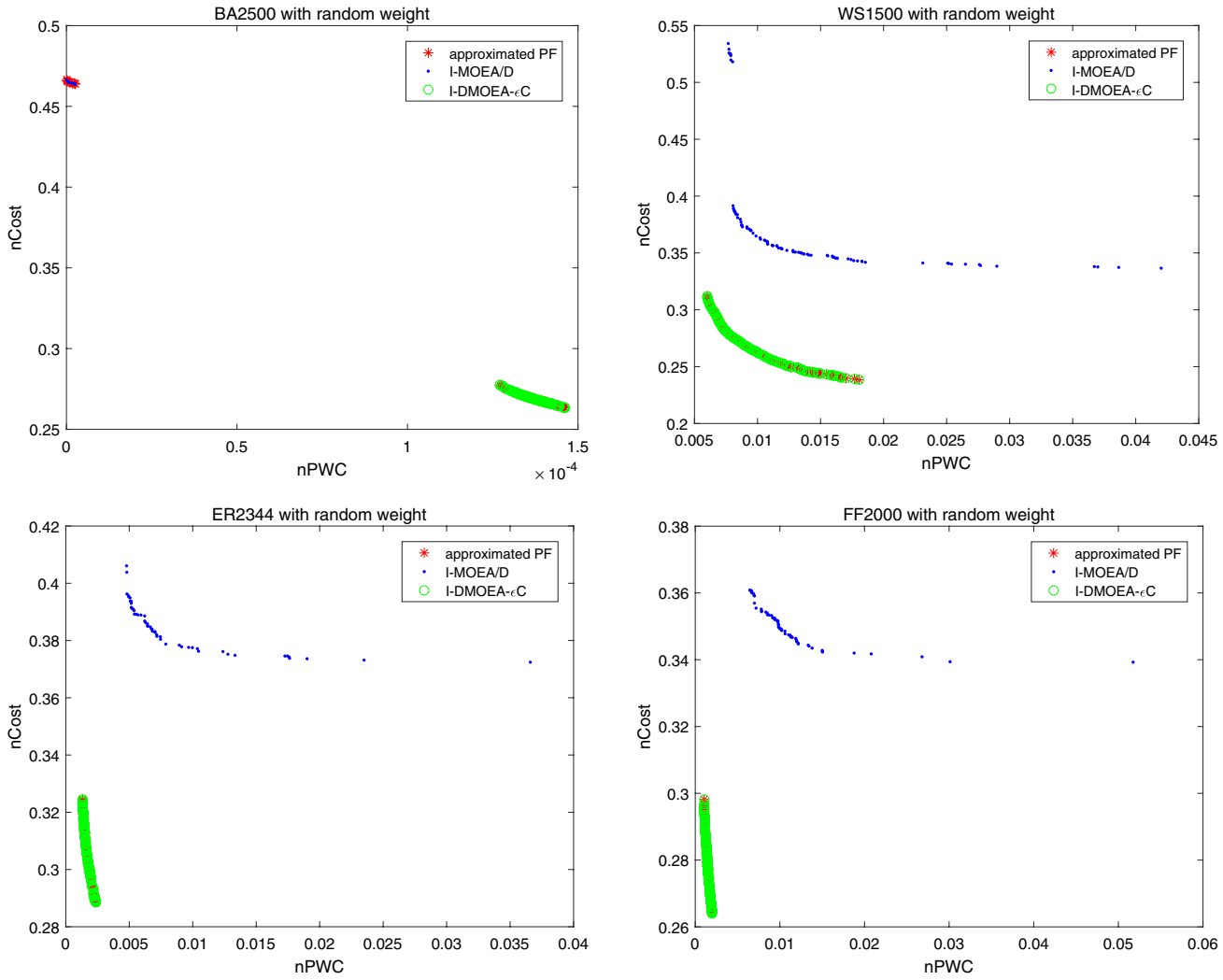

Fig. 2 Final populations in the objective space with the minimum IGD metric value within 20 runs obtained by I-MOEA/D and I-DMOEA-ϵC on BA2500, WS1500, ER2344, and FF2000 instances with random weights

their various variants over 20 independent runs. In order to compute the HV metric value, the reference point is set as 1.1 times the estimated nadir point[3] based on P^* for each instance.

4.3.1 Comparison among various variants

As mentioned above, the determination of the mating pool and the replacement pool are important for evolving the population toward the desired PF. Therefore, effects of various variants of two MOEAs with different strategies of determining the mating pool and the replacement pool are deeply analyzed on all test problems with random and logarithmic weights. The means and standard deviations of IGD metric values over 20 runs of MOEA/D and its variants about the mating pool determination on all test instances with random weights are shown in Table 2. Additionally, the means and standard deviations of HV metrics over 20 runs of variants of DMOEA-εC-NP-EP about the replacement pool determination on all test instances with logarithmic weights are shown in Table 3. The bold data in each table are the best mean metric values for each instance. The mean IGD (HV) values for each instance are sorted in an ascending (descending) order, and the numbers in the square brackets are their ranks. Besides, the mean rank values in terms of the IGD and HV metrics over all test instances are displayed for each variant to have a global view of the performance of all variants.

[3] The nadir point is the upper bound of the PF.

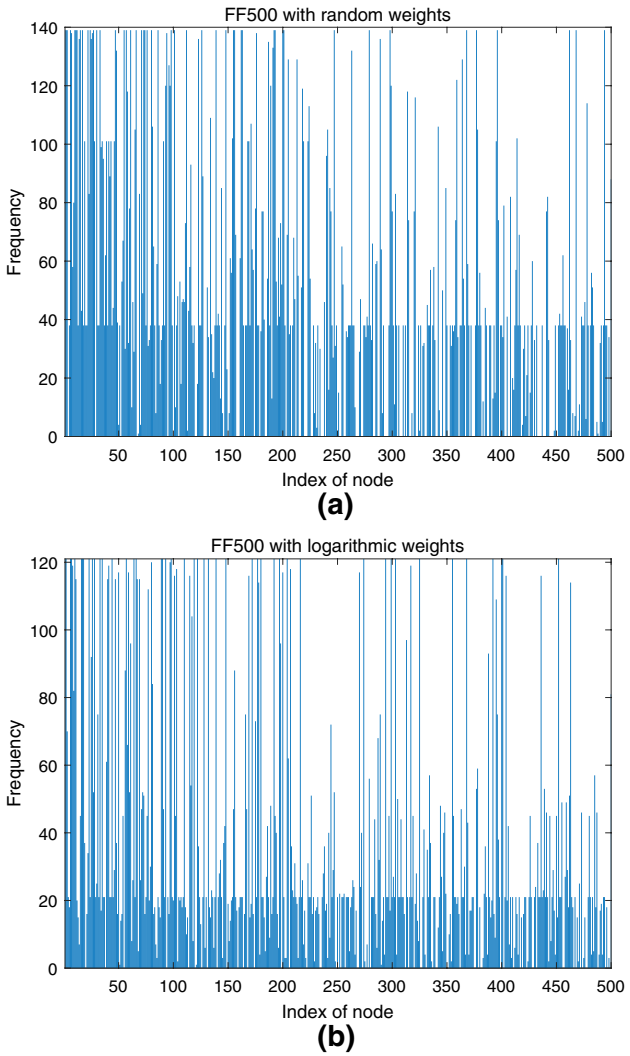

Fig. 3 Frequency of occurrence of each node in the set of final non-dominated solutions for FF500 instances with random and logarithmic weights

As can be seen in Table 2, in terms of IGD metric values, MOEA/D-NP-EP shows obvious advantage over other variants on the majority of test instances with random weights. On the other instances, MOEA/D-NP performs best. Table 2 also reveals the overall rank of the four variants, that is, MOEA/D-NP-EP, MOEA/D-NP, MOEA/D-N, and MOEA/D-P according to the mean rank values. Results in Table 2 highlight the effectiveness of the fourth strategy of determining the mating pool. To be specific, first a parent solution is selected from the external archive population EP. Then, the other one is selected from the neighborhood with a probability δ, and it is selected from the whole population with a probability $1-\delta$. As to DMOEA-εC and its variants about the mating pool determination, similar results can be obtained. That is, DMOEA-εC with the fourth type of mating pool performs the best among all variants.

The superiority of the fourth mating pool determination strategy can be explained in the following two aspects. Firstly, the basic assumption of decomposition-based MOEAs is that neighboring subproblems have similar optimal solutions. Thus, two solutions of neighboring subproblems have a higher chance to produce good solutions which can accelerate convergence. The participation of the whole population in the mating pool with a probability $1-\delta$ is beneficial to give birth to diverse offspring. Additionally, since the external archive population EP is used to store non-dominated solutions found so far, it is reasonable to make best use of it to produce new solutions with high quality.

It can be observed from Table 3 that DMOEA-εC-NP-EP-G outperforms DMOEA-εC-NP-EP-L on the majority of test instances with logarithmic weights in terms of the HV metrics. Similar results can be obtained for MOEA/D and its variants about the replacement pool determination in terms of the HV metrics. The broadened range of the replacement pool makes the replacement more effective, which is good for convergence. Besides, the limited number of replacement takes control of maintaining the diversity of a population. In conclusion, effectiveness of the fourth strategy of determining the mating pool and the global replacement strategy are confirmed experimentally. Thus, the improved MOEA/D and DMOEA-εC with the fourth type of mating pool and the global replacement pool are denoted as I-MOEA/D and I-DMOEA-εC, respectively. They will be employed in the following numerical experiments.

4.3.2 Comparison between two improved MOEAs: I-MOEA/D and I-DMOEA-εC

This part of the experiments are designed to study the effectiveness of I-MOEA/D and I-DMOEA-εC on Bi-CNDPs. Our comparison is made of two perspectives: (1) the comparison between single-objective and multi-objective formulations and (2) the comparison between two improved multi-objective approaches.

Firstly, it has been theoretically proved that the solution of a single-objective problem whose objective is a convex linear combination of the objectives of the MOP is part of the PF of the MOP (Miettinen 1999). This statement is still valid for the Bi-CNDPs, and the optimization of Bi-CNDP can present a set of Pareto optimal solutions for decision makers to have a global view of the problem and make more reasonable decisions.

Secondly, concerning the latter perspective, the IGD (Zhou et al. 2005) and HV (Zitzler and Thiele 1999) are still employed to evaluate the performance of compared algorithms. The means and standard deviations of IGD and HV metric values of two improved MOEAs over 20 independent runs of each algorithm on sixteen instances with random

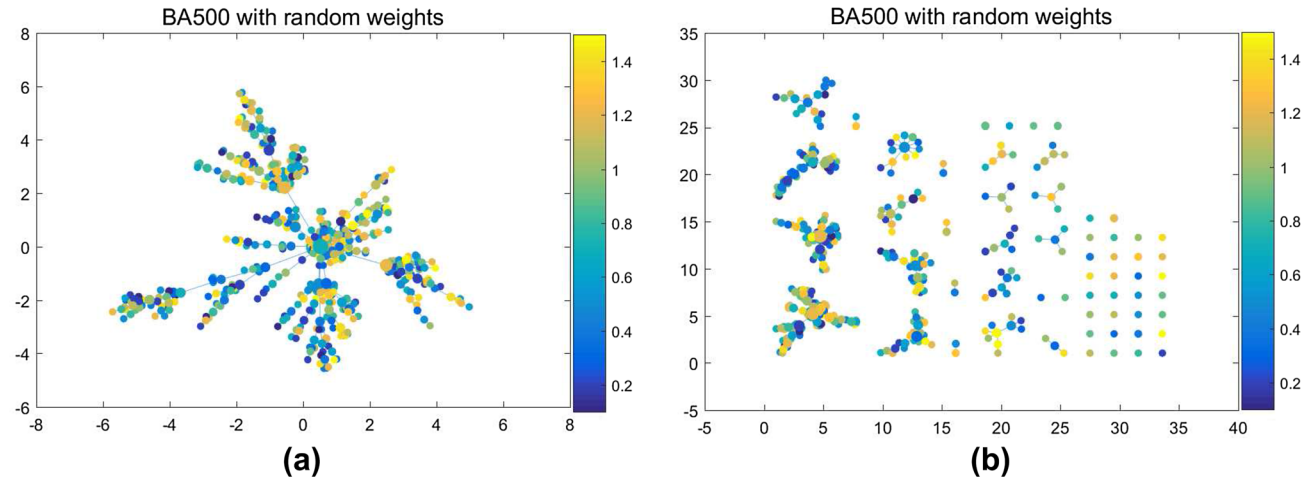

Fig. 4 The network of the BA500 instance with random weights **a** initial; **b** after removing a preferred non-dominated solution with the smallest pairwise connectivity in the induced graph and the total cost of removing nodes less than a predefined threshold 134. The size of each node is proportional to its degree, and the color of each node color is related to its weight value

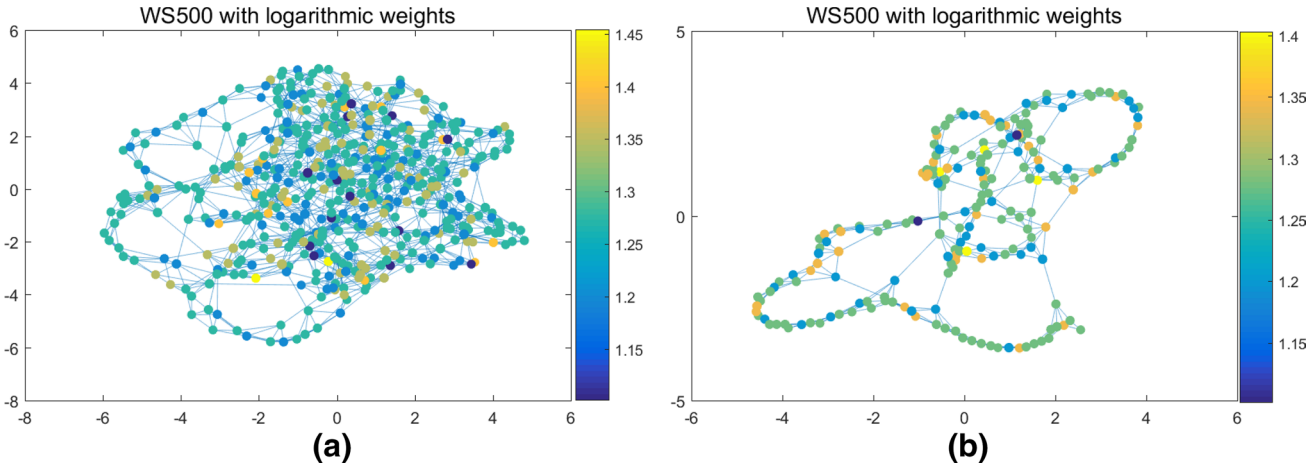

Fig. 5 The network of the WS500 instance with logarithmic weights **a** initial; **b** after removing a preferred non-dominated solution with the smallest pairwise connectivity in the induced graph and the total cost of removing nodes less than a predefined threshold 589. The size of each node is proportional to its degree, and the color of each node color is related to its weight value

and logarithmic weights are shown in Tables 4 and 5. The Wilcoxon's rank sum test at a 95% significance level is conducted to test the significance of differences between the metric values yielded by I-DMOEA-ϵC and I-MOEA/D. †, §, and ≈ indicate that the performance of the I-DMOEA-ϵC is better than, worse than, and similar to that of I-MOEA/D according to the Wilcoxon's rank sum test, respectively. The bold data in each table are the best mean metric values for each instance.

As can be seen in Tables 4 and 5, in terms of IGD metric values, I-DMOEA-ϵC shows a significant advantage over I-MOEA/D on all test instances except for BA2500 with random weights, WS500 and ER941 with logarithmic weights on which two algorithms show competitive performance. As to HV, I-DMOEA-ϵC shows significant superiority over I-MOEA/D on the majority of test instances. To be specific, I-MOEA/D performs better than I-DEMOA-ϵC on BA2500 with logarithmic weights, WS1000 with random and logarithmic weights, and ER2344 with random weights. The two algorithms perform competitively on BA5000 and ER2344 with logarithmic weights. On the remaining test instances, I-DMOEA-ϵC outperforms I-MOEA/D significantly.

Figure 2 shows the distribution of the final solutions with the minimum IGD value within 20 runs found by I-MOEA/D and I-DMOEA-ϵC on BA2500, WS1500, ER2344, and FF2000 instances with random weights.It can be seen from

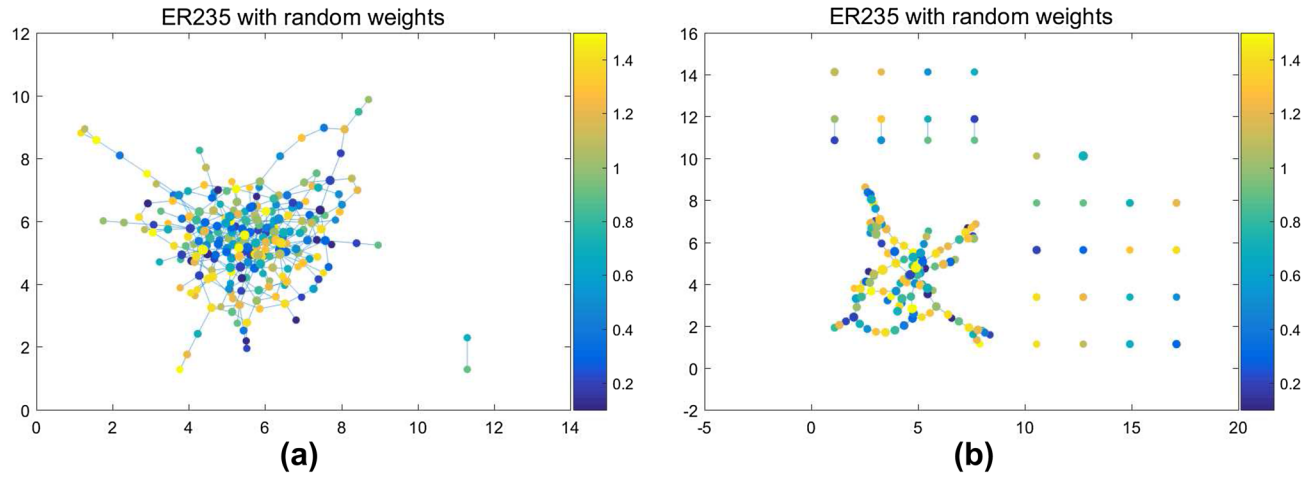

Fig. 6 The network of the ER235 instance with random weights **a** initial; **b** after removing a preferred non-dominated solution with the smallest pairwise connectivity in the induced graph and the total cost of removing nodes less than a predefined threshold 90. The size of each node is proportional to its degree, and the color of each node color is related to its weight value

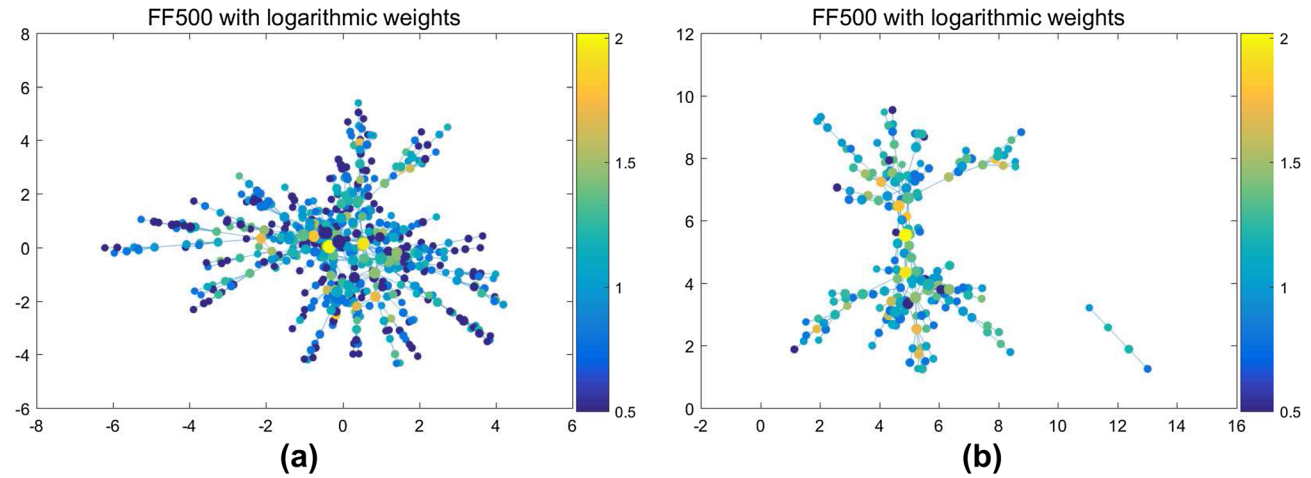

Fig. 7 The network of the FF500 instance with logarithmic weights **a** initial; **b** after removing a preferred non-dominated solution with the smallest pairwise connectivity in the induced graph and the total cost of removing nodes less than a predefined threshold 282. The size of each node is proportional to its degree, and the color of each node color is related to its weight value

Fig. 2 that I-MOEA/D and I-DMOEA-εC exhibit different behaviors and can find different parts of the PF for each instance. To be specific, I-MOEA/D tends to find solutions with high cost values, while I-DMOEA-εC shows opposite behavior. Additionally, results obtained via I-DMOEA-εC achieve better convergence. The reason that I-DMOEA-εC cannot cover the PF well can be attributed to bad estimations of nadir points during the evolutionary process. In summary, Fig. 2 shows that I-MOEA/D and I-DMOEA-εC can find different parts of the PF for each instance and I-DMOEA-εC exhibits better performance compared with I-MOEA/D.

After optimizing the above-mentioned Bi-CNDP via certain MOEA, a set of non-dominated solutions will be obtained. We calculate the frequency of occurrence of each node in the set of final non-dominated solutions for all instances with random and logarithmic weights. Figure 3 shows frequency values of all nodes that exist in the set of final non-dominated solutions for FF500 instances with random and logarithmic weights. According to Fig. 3, we note that frequency values of the majority of nodes of the instance with logarithmic weights are lower than that of the instance with random weights. This suggests that assigning higher weight values to nodes with larger node degree has the effect of reducing frequency values of nodes in the non-dominated solutions. It can be concluded that it is an effective way to spend more resources on protecting nodes

with high node degree in order to increase the robustness of a network.

Furthermore, a decision-making process from the perspective of minimizing the pairwise connectivity of the induced graph given a constraint on the cost of removing nodes is proposed. Specifically, after obtaining a set of non-dominated solutions, decision makers can select a preferred non-dominated solution with the smallest pairwise connectivity in the induced graph given a constraint on the cost of removing nodes. In the end, we exhibit the visual results of the decision-making process for the BA500, WS500, ER235, and FF500 instances, as shown in Figs. 4, 5, 6 and 7. In all figures, we use a color map to represent the topology of the initial network and the network after removing a set of nodes based on a selected Pareto optimal solution. The size of each node is proportional to its degree, and the color of each node is related to its weight value.

5 Concluding remarks

Given a graph, the critical node detection problem consists of finding a set of nodes, deletion of which satisfies one or more metrics in the induced graph. In contrast to most previous approaches, we use a bi-objective formulation, rather than make hypotheses on the psychology of decision makers. In this paper, we propose and study a new variant of this problem called bi-objective critical node detection problem (Bi-CNDP). In this formulation, we assume that removing each node has a cost, and decision makers want to minimize the pairwise connectivity of the induced graph and minimize the cost of removing a set of nodes at the same time.

We firstly prove the NP-hardness of this problem on general graphs and the existence of a polynomial algorithm for constructing the ϵ-approximated Pareto front for Bi-CNDPs on trees. Then, different types of the mating pool and the replacement pool are proposed and integrated in two state-of-the-art decomposition-based MOEAs including MOEA/D and DMOEA-ϵC. Two MOEAs and their variants are applied to solve the proposed Bi-CNDP. Sixteen common benchmark problems are modified by assigning random and logarithmic weight to each node. Computational experiments on all test instances were conducted first to evaluate the performance of different variants about the mating pool and the replacement pool. Then, further numerical experiments are used to compare the performance of two improved MOEAs, i.e., I-MOEA/D and I-DMOEA-ϵC, on Bi-CNDP. Numerical results not only show the effectivenesses of the proposed fourth mating pool and the global replacement strategy, but also demonstrate different behaviors of two improved MOEAs and the superiority of I-DMOEA-ϵC on the majority of test problems. Finally, a decision-making process from the perspective of single-objective is proposed for helping decision makers to identify the most critical nodes with the smallest pairwise connectivity and the total cost of removing nodes less than a predefined threshold.

Future research work will include investigations of designing more effective reproduction operators, embedding problem-specific knowledge during the optimization process, using single-objective methods to further refine solutions obtained via multi-objective approaches, and considering uncertainties in Bi-CNDPs.

Acknowledgements This study was funded by the National Outstanding Youth Talents Support Program 61822304, the NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization under Grant U1609214, the National Natural Science Foundation of China under Grant 61673058, the Foundation for Innovative Research Groups of the National Natural Science Foundation of China under Grant 61621063, and the Projects of Major International (Regional) Joint Research Program NSFC under Grant 61720106011. P. M. Pardalos is supported by the Paul and Heidi Brown Preeminent Professorship in Industrial and Systems Engineering, University of Florida.

Compliance with ethical standards

Conflict of Interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Addis B, Di Summa M, Grosso A (2013) Identifying critical nodes in undirected graphs: complexity results and polynomial algorithms for the case of bounded treewidth. Discrete Appl Math 161(16–17):2349–2360

Addis B, Aringhieri R, Grosso A, Hosteins P (2016) Hybrid constructive heuristics for the critical node problem. Ann Oper Res 238(1):1–13

Aringhieri R, Grosso A, Hosteins P, Scatamacchia R (2015) VNS solutions for the critical node problem. Electron Notes Discrete Math 47:37–44

Aringhieri R, Grosso A, Hosteins P, Scatamacchia R (2016a) A general evolutionary framework for different classes of critical node problems. Eng Appl Artif Intell 55:128–145

Aringhieri R, Grosso A, Hosteins P, Scatamacchia R (2016b) Local search metaheuristics for the critical node problem. Networks 67(3):209–221

Aringhieri R, Grosso A, Hosteins P, Scatamacchia R (2019) Polynomial and pseudo-polynomial time algorithms for different classes of the distance critical node problem. Discrete Appl Math 253:103–121

Arulselvan A, Commander CW, Pardalos PM, Shylo O (2007) Managing network risk via critical node identification. Risk management in telecommunication networks, Springer

Arulselvan A, Commander CW, Elefteriadou L, Pardalos PM (2009) Detecting critical nodes in sparse graphs. Comput Oper Res 36(7):2193–2200

Arulselvan A, Commander CW, Shylo O, Pardalos PM (2011) Cardinality-constrained critical node detection problem. Springer, New York

Aspnes J, Chang K, Yampolskiy A (2005) 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, pp 43–52

Atputharajah A, Saha TK (2009) Power system blackouts—literature review. In: Proceedings of the international conference on industrial and information systems, pp 460–465

Balas E, Souza CCd (2005) The vertex separator problem: a polyhedral investigation. Math Program 103(3):583–608

Ben-Ameur W, Mohamed-Sidi M-A, Neto J (2015) The *k*-separator problem: polyhedra, complexity and approximation results. J Comb Optim 29(1):276–307

Borgatti SP (2006) Identifying sets of key players in a social network. Comput Math Organ Theory 12(1):21–34

Chen P-A, David M, Kempe D (2010) Better vaccination strategies for better people. In: Proceedings of the ACM conference on electronic commerce, pp 179–188

Chen J, Li J, Xin B (2017) DMOEA-εC: decomposition-based multiobjective evolutionary algorithm with the ε-constraint framework. IEEE Trans Evolut Comput 21(5):714–730

Deb K, Agrawal S, Pratap A, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evolut Comput 6(2):182–197

Di Summa M, Grosso A, Locatelli M (2011) Complexity of the critical node problem over trees. Comput Oper Res 38(12):1766–1774

Di Summa M, Grosso A, Locatelli M (2012) Branch and cut algorithms for detecting critical nodes in undirected graphs. Comput Optim Appl 53(3):649–680

Dinh TN, Thai MT (2013) Precise structural vulnerability assessment via mathematical programming. In: Proceedings of the military communications conference, pp 1351–1356

Dinh TN, Xuan Y, Thai MT, Park EK, Znati T (2010) On approximation of new optimization methods for assessing network vulnerability. In: Proceedings of the conference on information communications, pp 2678–2686

Fan N, Pardalos PM (2010) Robust optimization of graph partitioning and critical node detection in analyzing networks. Springer, Berlin

Faramondi L, Oliva G, Panzieri S, Pascucci F, Schlueter M, Munetomo M, Setola R (2018) Network structural vulnerability: a multiobjective attacker perspective. IEEE Trans Syst Man Cybern Syst PP(99):1–14

Granata D, Steeger G, Rebennack S (2013) Network interdiction via a critical disruption path: branch-and-price algorithms. Comput Oper Res 40(11):2689–2702

Jenelius E, Petersen T, Mattsson L-G (2006) Importance and exposure in road network vulnerability analysis. Transp Res Part A Policy Pract 40(7):537–560

Kempe D, Kleinberg J, Tardos E (2010) Maximizing the spread of influence in a social network. Progressive research, pp 137–146

Kuhlman CJ, Kumar VSA, Marathe MV, Ravi SS, Rosenkrantz DJ (2010) Finding critical nodes for inhibiting diffusion of complex contagions in social networks. In: Proceedings of the European conference on machine learning and knowledge discovery in databases, pp 111–127

Kumar VSA, Rajaraman R, Sun Z, Sundaram R (2010) Existence theorems and approximation algorithms for generalized network security games. In: Proceedings of the IEEE international conference on distributed computing systems, pp 348–357

Lalou M, Tahraoui MA, Kheddouci H (2016) Component-cardinality-constrained critical node problem in graphs. Discrete Appl Math 210:150–163

Lalou M, Tahraoui MA, Kheddouci H (2018) The critical node detection problem in networks: a survey. Comput Sci Rev 28:92–117

Leskovec J, Krause A, Guestrin C, Faloutsos C, Vanbriesen J, Glance N (2007) Cost-effective outbreak detection in networks. In: ACM SIGKDD international conference on knowledge discovery and data mining, pp 420–429

Liang G, Weller SR, Zhao J, Luo F, Dong ZY (2017) The 2015 Ukraine blackout: implications for false data injection attacks. IEEE Trans Power Syst 32:3317–3318

Miettinen K (1999) Nonlinear multiobjective optimization. Kluwer Academic Publishers, Boston

Papadimitriou CH, Yannakakis M (2000) On the approximability of trade-offs and optimal access of web sources. In: Proceedings of the symposium on foundations of computer science, pp 86–92

Pavlikov K (2018) Improved formulations for minimum connectivity network interdiction problems. Comput Oper Res 97:48–57

Purevsuren D, Cui G, Qu M, Win NNH (2017) hybridization of GRASP with exterior path relinking for identifying critical nodes in graphs. Int J Comput Sci 44(2):157–165

Qi Y, Ma X, Liu F (2014) MOEA/D with adaptive weight adjustment. Evolut Comput 22(2):231–264

Salmeron J, Wood K, Baldick R (2015) Analysis of electric grid security under terrorist threat. IEEE Trans Power Syst 19(2):905–912

Shen S, Smith JC (2012) Polynomial-time algorithms for solving a class of critical node problems on trees and series-parallel graphs. Networks 60(2):103–119

Shen Y, Nguyen NP, Xuan Y, Thai MT (2013a) On the discovery of critical links and nodes for assessing network vulnerability. IEEE/ACM Trans Netw 21(3):963–973

Shen Y, Dinh TN, Thai MT (2013b) Adaptive algorithms for detecting critical links and nodes in dynamic networks. In: Proceedings of the IEEE military communications conference, pp 1–6

Spears WM, Jong KAD (1991) On the virtues of parameterized uniform crossover. In: Proceedings of the 4th international conference on genetic algorithms, pp 230–236

Tomaino V, Arulselvan A, Veltri P, Pardalos PM (2012) Studying connectivity properties in human protein–protein interaction network in cancer pathway. Springer, New York

Ventresca M (2012) Global search algorithms using a combinatorial unranking-based problem representation for the critical node detection problem. Comput Oper Res 39(11):2763–2775

Ventresca M, Aleman D (2014) A randomized algorithm with local search for containment of pandemic disease spread. Comput Oper Res 48(7):11–19

Ventresca M, Aleman D (2015) Efficiently identifying critical nodes in large complex networks. Comput Soc Netw 2(1):6

Ventresca M, Harrison KR, Ombuki-Berman BM (2018) The bi-objective critical node detection problem. Eur J Oper Res 265(3):895–908

Veremyev A, Prokopyev OA, Pasiliao EL (2014a) An integer programming framework for critical elements detection in graphs. J Comb Optim 28(1):233–273

Veremyev A, Boginski V, Pasiliao EL (2014b) Exact identification of critical nodes in sparse networks via new compact formulations. Optim Lett 8(4):1245–1259

Walteros JL, Veremyev A, Pardalos PM, Pasiliao EL (2019) Detecting critical node structures on graphs: a mathematical programming approach. Networks 73:48–88

Walteros JL, Pardalos PM (2012) Selected topics in critical element detection. In: Daras N (ed) Applications of mathematics and informatics in military science, vol 71. Springer optimization and its applications. Springer, New York, pp 9–26

Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evolut Comput 11(6):712–731

Zhang Q, Liu W, Li H (2009) The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. Technical Report CES-491, The School of Computer Science and Electronic Engineering, University of Essex

Zhou T, Fu Z, Wang B (2006) Epidemic dynamics on complex networks. Prog Nat Sci Mater Int 16(5):452–457

Zhou A, Zhang Q, Jin Y, Tsang E, Okabe T (2005) A model-based evolutionary algorithm for bi-objective optimization. In: Proceedings of the IEEE congress on evolutionary computation, pp 2568–2575

Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithm: a comparative case study and strength pareto approach. IEEE Trans Evolut Comput 3(4):257–271