Applied Soft Computing 24 (2014) 1145–1154

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A multi-objective approach for solving the survivable network design problem with simultaneous unicast and anycast flows

Huynh Thi Thanh Binh [a,*], Bui Thu Lam [b], Nguyen Sy Thai Ha [a], Hisao Ishibuchi [c]

[a] School of Information and Communication Technology, Hanoi University of Science and Technology, Hanoi, Viet Nam
[b] Faculty of Information Technology, Le Quy Don Technical University, Hanoi, Viet Nam
[c] Department of Computer Science and Intelligent Systems, Osaka Prefecture University, Sakai, Japan

ARTICLE INFO

Article history:
Received 12 April 2013
Received in revised form 7 February 2014
Accepted 2 June 2014
Available online 5 July 2014

Keywords:
Survivable network design
Anycast
Unicast
A-SNDP
NSGA-II

ABSTRACT

In this paper, we consider the survivable network design problem for simultaneous unicast and anycast flow requests. We assume that the network is modeled by a connected and undirected graph. This problem aims at finding a set of connections with a minimized network cost in order to protect the network against any single failure. The cost is computed using the *all capacities modular cost* (*ACMC*) *model* and a set of flow demands. We name it as *ACMC-based survivable network design problem* (*A-SNDP*). It is proved that the problem is *NP*-hard. We introduce a multi-objective approach to solve *A-SNDP*. The objectives are to minimize the network cost (*NCost*) and the network failure (*NFail*). Extensive simulation results on instances of Polska, Germany and Atlanta networks showed the efficiency of the multi-objective approach for solving *A-SNDP*.

Introduction

There are many types of connections for data transmission over the Internet. The two most popular types of connections are unicast and anycast. A connection from a node to another is called unicast. An anycast one is also a connection from a node to another; however, the difference is that the destination node has a one or many replica servers which back up for it. The anycast connection is currently used in many applications such as domain name service (*DNS*), web service, overlay network, peer-to-peer (*P2P*) systems, content delivery network (*CDN*), and software distribution [2]. The popularity of the anycast technology is predicted to increase in the near future since many new services using both unicast and anycast paradigms have been being developed [1].

In the Internet, any network failure can cause serious consequences; an example can be seen in the case reported in [12]: a single link failure affected more than 30,000 users and it took 12 hours to fix. Therefore, it is crucial to design survivable networks with simultaneous unicast and anycast flows. In the survivable network design problem (*SNDP*), we would like to minimize the network cost and minimize the network failures simultaneously. In order to decrease the network cost, we have to utilize some links that have the redundant bandwidth for many demands. However, this causes the increase of the number of connections which go over a node, and increasing the frequency of failures as a result. It turns out to be difficult optimizing both objectives at the same time. Therefore, we will alternatively find the acceptable solutions by a multi-objective algorithm.

This paper proposes to deal with *A-SNDP* which uses *all capacities modular cost* (*ACMC*) *model* for calculating the network cost [1]. *A-SNDP* is proved that the problem is *NP*-hard [19]. It first formulates the *A-SNDP* as a multi-objective design problem (called *MA-SNDP*). It then proposes a multi-objective approach for solving *MA-SNDP*; in particular it introduces two new multi-objective genetic algorithms for solving *MA-SNDP*. The first one is a multi-objective genetic algorithm with a scheme of *complete connection encoding—CCE* for solving *A-SNDP*. The second one is a multi-objective genetic algorithm with *connection database based encoding* (*CDE*) for solving *MA-SNDP*. We experimented on the Polska, Germany and Atlanta network instances. We compared the results between multi-objective and single-objective approaches; also we investigated the effect of the choice of an encoding scheme on their performance. Our experimental results showed the efficiency of the multi-objective approach for solving *MA-SNDP*.

The rest of this paper is organized as follows. The “Related works” section describes the related works. The “Background on multi-objective evolutionary algorithms” section is for the background on multi-objective evolutionary algorithms (*MOEAs*).

* Corresponding author. Tel.: +84 903226786.
E-mail address: binh.huynhthithanh@hust.edu.vn (H.T.T. Binh).

http://dx.doi.org/10.1016/j.asoc.2014.06.001
1568-4946/

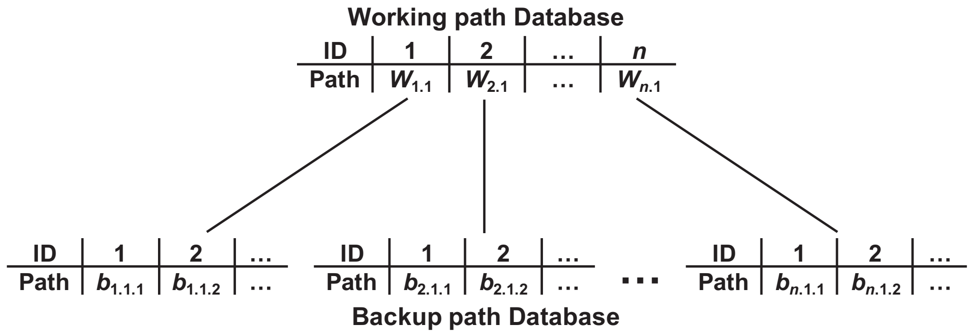

Fig. 1. The demonstration of *CDE* for demand 1where $w_{i.1}$ is the working path, $b_{i.1.j}$ is the backup path of working path $w_{i.1}$, $i = 1 \to n$; j is the ordered number for backup path database.

wP_1	bP_1
wP_2	bP_2
wP_3	bP_3
..	
$wP_{\|D\|}$	$bP_{\|D\|}$

Fig. 2. An example representing an individual built by *CCE*, where each row represents a solution for a demand.

Problem formulation is introduced in the "Formulation of the multi-objective *A-SNDP* (*MA-SNDP*)" section. Our new proposed algorithm is showed in the "Proposed methodology" section. The "Experimental results" section gives our experiments and computational and comparative results. The paper concludes with discussions and future works in the "Conclusion" section.

Related works

SNDP is generally presented in [10], where both economics and reliability are considered in the telecommunication network design. *SNDP* has to guarantee the survivability of a network system and also to minimize the network cost. The most popular way mentioned in many researches is the single backup-path method. The main idea of this method is as following: each connection has a working path and a backup path if there is a single link being failed on the working path, the connection is then switched to the backup path [2–5].

If the network cost is computed using the *all capacities modular cost* (*ACMC*) *model*, we classify the problem as the *ACMC*-based *SNDP* or *A-SNDP* [1]. With single objective approach for solving *A-SNDP*, we optimize the network cost. There are many researches on minimizing the network cost for *SNDP* (see [2,8,10] and references therein). They use branch-and-bounds or branch-and-cut methods to find the optimal solution. These methods can only use for networks with a small number of nodes. For larger networks, they may propose heuristics such as evolutionary algorithms, tabu search [2] and simulated annealing [8]. In [10], Nissen and Gold applied the evolution strategy (*ES*). It was shown by the authors that a larger population in ES can help to achieve a better result than a smaller one by avoiding or delaying convergence on local optimal. However, this algorithm was applied in the network which has only unicast flows.

With the network which has both anycast and unicast flows, Walkowiak et al. presented a heuristic algorithm for solving *A-SNDP* [13]. The main idea of this algorithm is based on flow deviation [8] and local search [14]. They achieved a quite good result with Polska (12 nodes, 36 links, 65 unicast, 12 anycast) network, the detail showed that the average gap of the proposed heuristic to optimal result was 7.11%. Furthermore, they also built a Tabu search algorithm based on hill climbing with some heuristics to solve *A-SNDP* [2]. Experiments on three large instances, which are Polska (12 nodes, 36 links, 65 unicast, 12 anycast), Germany (17 nodes, 52 links, 119 unicast, 13 anycast) and Atlanta (26 nodes, 82 links, 234 unicast, 22 anycast), showed many promising results. In particular with Polska network, they achieved the following average gap to the optimal results: 2.57% for 70% anycast/30% unicast case and 2.00% for 80% anycast/20% unicast case. However, their Tabu search algorithm is still simple and their results cannot be optimal completely.

In [15], Huynh et al. also proposed two heuristics called FBB_1 and FBB_2 for solving *A-SNDP*. The main idea of FBB_1 is to utilize the redundant bandwidth corresponding to the paid cost level in each link. FBB_2 is the combination of FBB_1 and Tabu search. Experiments on three network instances, which are Polska, Germany and Atlanta [2–13], were reported in [15]. With each instance, 10 test sets were randomly created. The results showed that their proposed approach was quite effective with *A-SNDP*. On all instances, FBB_1 and FBB_2 had better results than Tabu search in most of test sets.

In [16], Eduardo et al. introduced a multi-objective approach to the design of electrical distribution networks with two objectives which are the monetary cost and the system failure. Their experimental results demonstrated that the use of multi-objective investment policies was more valuable for energy companies than the traditional single-objective optimal solutions. The

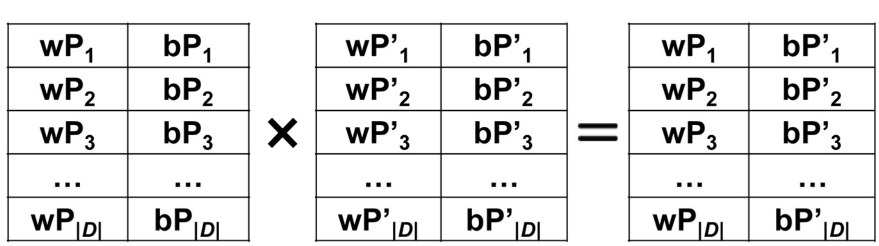

Fig. 3. Illustrates the crossover operator between parent *T* and *T'*, which reproduces the child *X*.

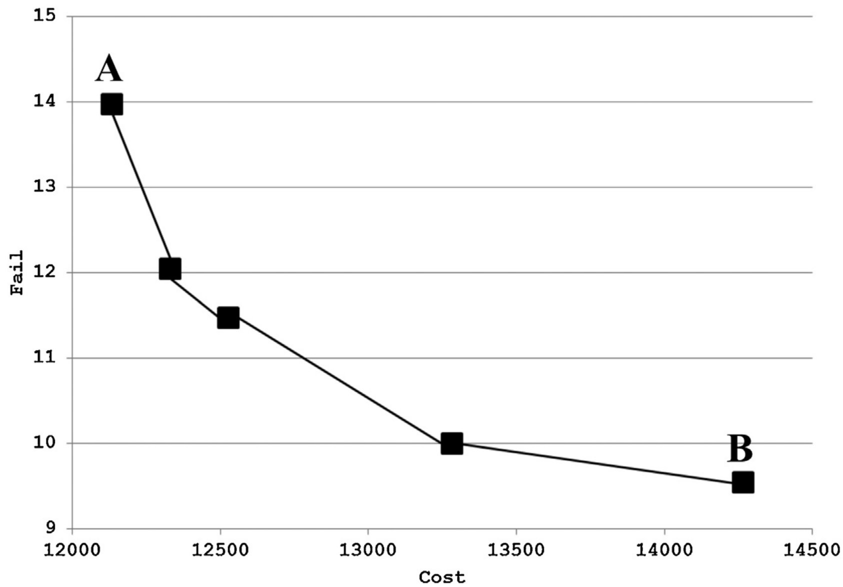

Fig. 4. The non-dominated solutions obtained by *CCE-NSGA*-II on a test set in Polska network. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

usefulness of multi-objective design approaches was proven by their results.

For *A-SNDP*, we can formulate it as a multi-objective problem based on cost and failure. It is difficult to optimize both objectives at the same time. Up to now, there is no multi-objective research for solving *A-SNDP*. It becomes our motivation to carry the research reported in this paper. So, in the next section, we will introduce our newly-designed multi-objective genetic algorithms for solving *A-SNDP*.

Background on multi-objective evolutionary algorithms

Practical problems in real-life usually have more than one objective (or criteria). Solving these problems, we often find a set of trading-off solutions. The set of optimal solutions to the problem are called *Pareto optimal set*. Its projection in objective space is known as the *Pareto optimal front* (*POF*).

Mathematically, in a k-objective unconstrained (bound constrained) minimization problem, a vector function $f(x)$ of k objectives is defined as

$$f(x) = [f_1(x), f_2(x), \ldots, f_k(x)]^T \tag{1}$$

In which x is a vector of decision variables. In evolutionary computation, x represents an individual in the population to be evolved. The value $f_j(x)$, then, describes the performance of individual x as evaluated against the jth objective in the multi-objective optimization problem (*MOP*).

An individual x_1 dominates x_2 if x_1 is not worse than x_2 on all k objectives and is better than x_2 on at least one objective. If x_1 does not dominate x_2 and x_2 also does not dominate x_1, then x_1 and x_2 are said to be non-dominated with respect to each other. If we use the symbol $\preceq$ to denote that $x_1 \preceq x_2$ means x_1 dominates x_2, and the symbol $\triangleleft$ between two scalars a and b to indicate that $a \not\triangleright b$ means a is not worse than b, then dominance can be formally defined asDefinition 1*(dominance):* $x_1 \preceq x_2$ if the following conditions are held:

(1) $f_j(x_1) \triangleleft f_j(x_2)\ \forall j \in [1,2,\ldots,k]$; and,
(2) $j \in [1,2,\ldots,k]$ in which $f_j(x_1) \triangleleft f_j(i_2)$.

In general, if an individual is not dominated by any other individual in the population, it is called a non-dominated solution. All non-dominated solutions in a population form the non-dominated set as formally described in the following definition:Definition 2*(non-dominated set): A set S is said to be the non-dominated set of a population P if the following conditions are met:*

(1) $S \subseteq P$
(2) $\forall s \in S, \nexists x \in P |\ x \preceq s$

If P represents the entire search space, then S is referred to as the *global Pareto optimal set*. If P represents only a sub-space, then S is called the *local Pareto optimal set*. While there can be multiple local Pareto optimal sets, there exists only one global one.

Multi-objective evolutionary algorithms are stochastic techniques being used to find Pareto optimal solutions for *MOP*s [17,20]. There are two key problems that *MOEA*s have to deal with:

- How to get as close as possible to the *POF*. This is challenging because of the stochasticity of the convergence process.
- How to keep solutions diverse. a diverse set of solutions will provide decision makers, designers, etc with more choice.

However, working on a set of solutions instead of only one, makes the measurement of *MOEA* convergence more difficult because one individual's closeness to the *POF* does not act as a measure for the entire set. Unsurprisingly, then, convergence and diversity are commonly used performance criteria when optimization algorithms are assessed and compared with each other.

To date, many *MOEA*s have been developed (i.e. Pareto archived evolution strategy—*PAES*, strength Pareto *EA* 2—*SPEA*2, Pareto frontier differential evolution—*PDE*, non-dominated sorting genetic algorithm II—*NSGA*-II, decomposition based multi-objective evolutionary algorithm *MOEA*/*D* and multi-objective particle swarm optimization *MOPSO*, and the direction based multi-objective evolutionary algorithm *DMEA*) and they are usually classified into two broad categories: with and without elitism. Elitist approach is a mechanism to preserve the best individuals, once found, during the optimization process. The concept of elitism was established at an early stage of evolutionary computation; and to date, it has been widely used in *EA*s. Elitist approach can be realized either

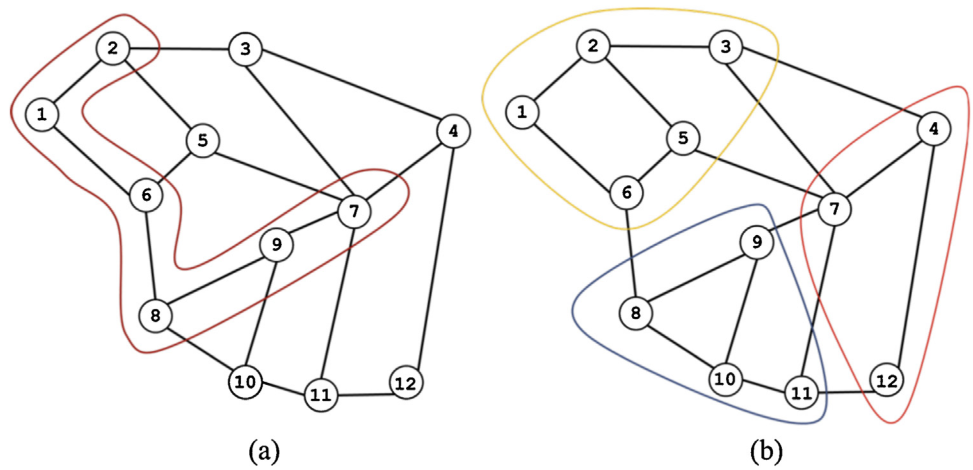

Fig. 5. (a) The state of Polska network with the left-most point A in Fig. 4. (b) The state of Polska network with the right-most point B in Fig. 4.

by placing one or more of the best parents directly into the next generation of individuals, or by replacing only those parents that are dominated by their offspring. Elitist *MOEA*s usually (but not necessarily) employ an external set called the archive to store the non-dominated solutions after each generation.

Formulation of the multi-objective *A-SNDP* (*MA-SNDP*)

For *A-SNDP*, the common criterion is to minimize the network cost while protecting the network against failures. Hence, we can formulate the multi-objective version with two objectives: minimizing the cost and minimizing the failures. To guarantee the survivability, we adopt the protection approach [2–5] where each connection must include a working path and a link-disjoint backup path. The working path is used for data transmission in failure-free state of the network. If there is a single link failure on the working path, the failed connection is then switched to the backup path.

The *MA-SNDP* is formulated as follows:

Input:

- An undirected graph $G=(V, E)$ where V is a set of nodes representing cities, the set of links $E=\{e \mid e=(u, v)\}$ with $u, v \preceq V$ and E is a set of links between cities.
- A set of customer demands $D=\{t_i, s_i, d_i, b_i\}, i=\{1,\ldots,|D|\}$ where

o $t_i = \begin{cases} 0 & \text{if the connection demand is unicast} \\ 0 & \text{if the connection demand is anycast} \end{cases}$

o s_i, $d_i \in V$ are the source and destination nodes of the demand i, respectively.

o b_i is the bandwidth requested by the demand i.

- $B_{e,k}$, $C_{e,k}$ are the bandwidth and the corresponding cost in level k of link e, respectively, $k=\{1,\ldots, K\}$ and K is the maximal level.
- A set of replica nodes $S=\{u_i, v_i\}$ with v_i is the replica node of u_i and v_i, $u_i \in V$.

Constraints:

- All customers' connection demands are satisfied.
- Each connection demand has two disjoint–link paths.

Objectives:

$NCost \rightarrow \min, \quad NFail \rightarrow \min$

With

$$NCost = \sum_e c_e \tag{2}$$

where $c_e = C_{e,k}$ if $B_{e,k-1} < \sum_i R_{ei} < B_{e,k}$, else $c_e = 0$;

$$NFail = \text{Max}\,(fail_1, fail_2, \ldots, fail_n) \tag{3}$$

here, c_e is the cost of link e; R_{ei} *is* bandwidth used by demand i in link $e \in E$, and $i=\{1,\ldots,|D|\}$; $fail_n$ is the number of connections affected when node n is false.

Output:

- A set of paths correspond with each of customer's demand.

If we would like to decrease the network cost, we have to utilize some links that have redundant bandwidth for many demands. However, this causes to increase the number of connections which go over a node, then deteriorates *NFail*. It is difficult to optimize both objectives at the same time. Therefore, we will propose to find acceptable solutions by a multi-objective approach.

Proposed methodology

When applying a multi-objective approach to deal with *MA-SNDP*, there will be no absolutely optimal solution for the two objectives. Instead, a set of non-dominated solutions will be offered. Here, we propose to use a genetic algorithm (*GA*) using dominance relations. The algorithm starts with a population of randomly initialized solutions. This population will be evolved over time. During the evolution process, all good solutions are preserved. In more detail, we used the non-dominated sorting mechanism of *NSGA*-II, *SPEA*2 where the parent population and offspring are combined and sorted in order to generate a population for the next generation. Selection of solutions for producing offspring is also performed as in *NSGA*-II, *SPEA*2 where a scheme of crowding tournament selection is used. However, the crossover and mutation operations are redesigned since the original operators for *NSGA*-II, *SPEA*2 are not suitable for our problem presentation.

Two encoding methods are used in this paper. The first is connection database based encoding (*CDE*). *CDE* encoding was introduced in [18]. The second one is complete connection encoding–(*CCE*).

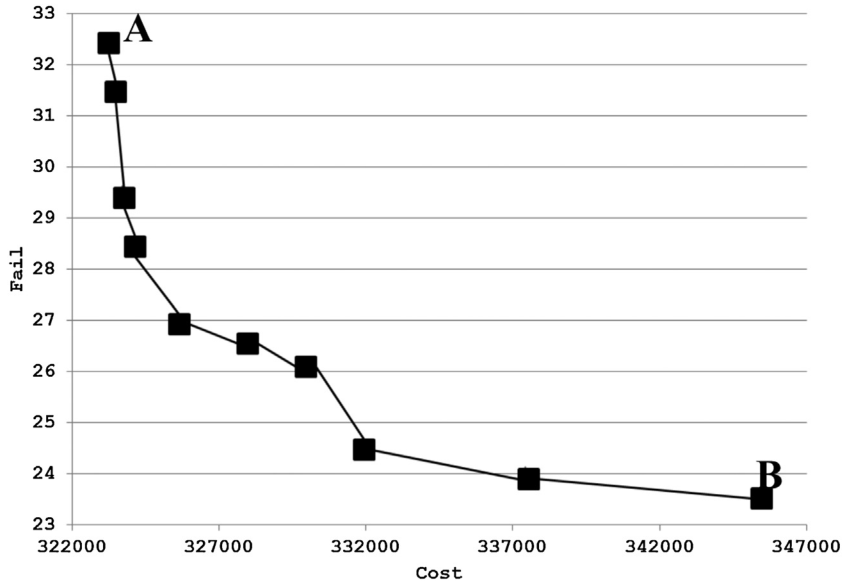

Fig. 6. The non-dominated solutions obtained by *CCE-NSGA*-II on a test set in Atlanta network. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

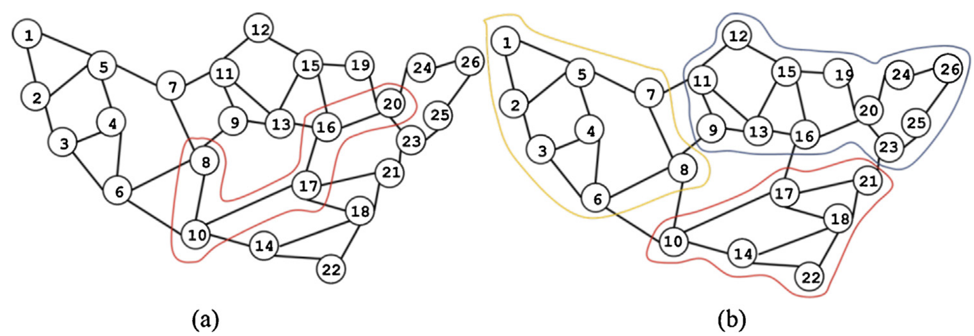

Fig. 7. (a) The state of Atlanta network with the left-most point A in Fig. 6. (b) The state of Atlanta network with the right-most point B in Fig. 6.

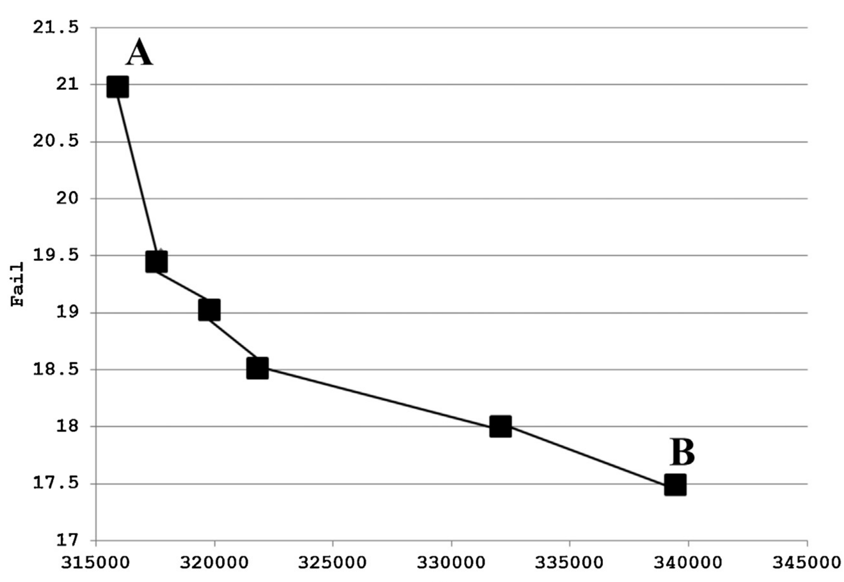

Fig. 8. The non-dominated solutions obtained by *CCE-NSGA*-II on a test set in Germany network.

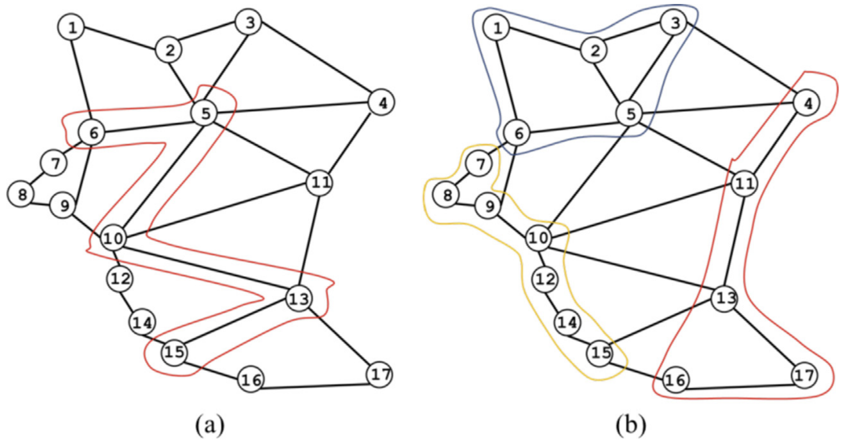

Fig. 9. (a) The state of Germany network with the left-most point A in Fig. 8. (b) The state of Germany network with the right-most point B in Fig. 8. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

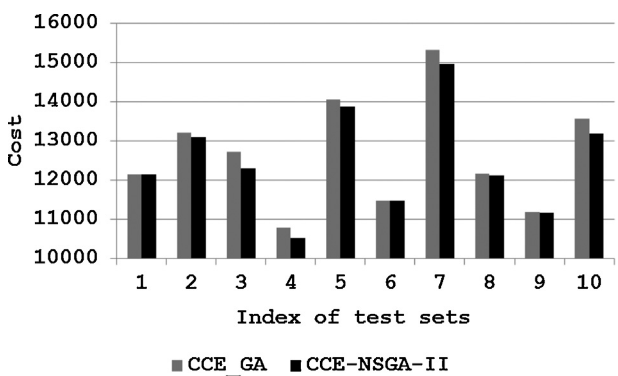

Fig. 10. The comparison between the result found by *CCE-GA* and *CCE-NSGA*-II on Polska network.

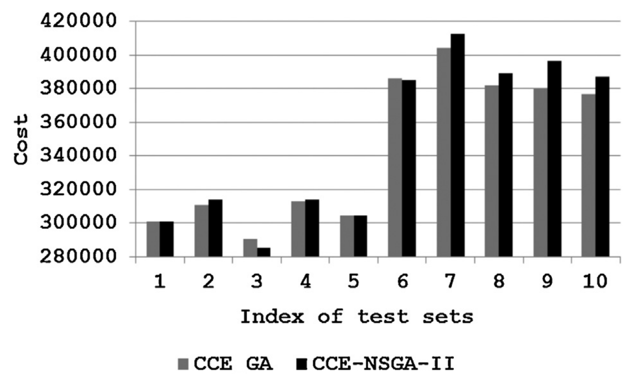

Fig. 11. The comparison between the result found by *CCE-GA* and *CCE-NSGA*-II on Germany network.

Genetic algorithm with connection database based encoding

Connection database based encoding

The paths between two nodes are encoded by connection database based encoding (*CDE*) [18] i.e. using a database to store possible paths between them. The path database is built by using a routine called *FindOtherWays*. The main idea of *FindOtherWays* is as follows: first, we initialize an empty database with two columns: *ID* and *Path*. To find possible paths from *A* to *B*, we use the shortest path algorithm to find the first one, insert it into the column path of this database, then remove a link of this path randomly out of the graph and apply this algorithm again for the next paths. This algorithm will be terminated if we cannot find any paths from *A* to *B* or we achieve the limited number of paths with each demand. Finally, we have a database with *ID* and corresponding paths from *A* to *B*. This is not an exhausted list of paths, its usage implies that once we have a database of paths, we can apply some expert knowledge to remove unnecessary ones.

So, with each connection demand, we use *FindOtherWays* to create a working path database (*WD*). Then, *FindOtherWays* routine is used to create a backup path database (*BD*) for each record of *WD*. Note that, all paths in *BD* of a record are disjoint-link with the path in this record.

With this building, working path and backup path do not have common edges. So if the working path breaks, the backup path will replace and restore data on the working path easily. Thus, the network survivability is always guaranteed for each demand with a single failure case. Each demand in *MA-SNDP* will have working path and backup path databases, and they relate together through the foreign key. Fig. 1 presents the working path and backup path databases of demand 1.

An individual is represented by a pair of strings. Their length is equal to the number of input demands. The first string includes the ID of the working path and the other includes the ID of the backup path of n input demands.

Genetic operators

Crossover operator: We used the one-point crossover.

Mutation operator: A gene is randomly chosen in a chromosome; then its value is replaced with a different value in the corresponding database with this gene.

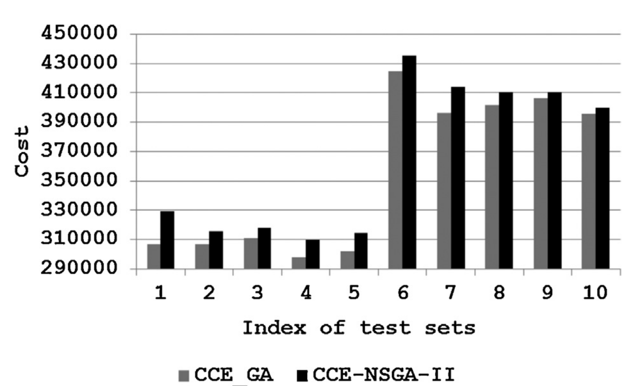

Fig. 12. The comparison between the result found by *CCE-GA* and *CCE-NSGA*-II on Atlanta network.

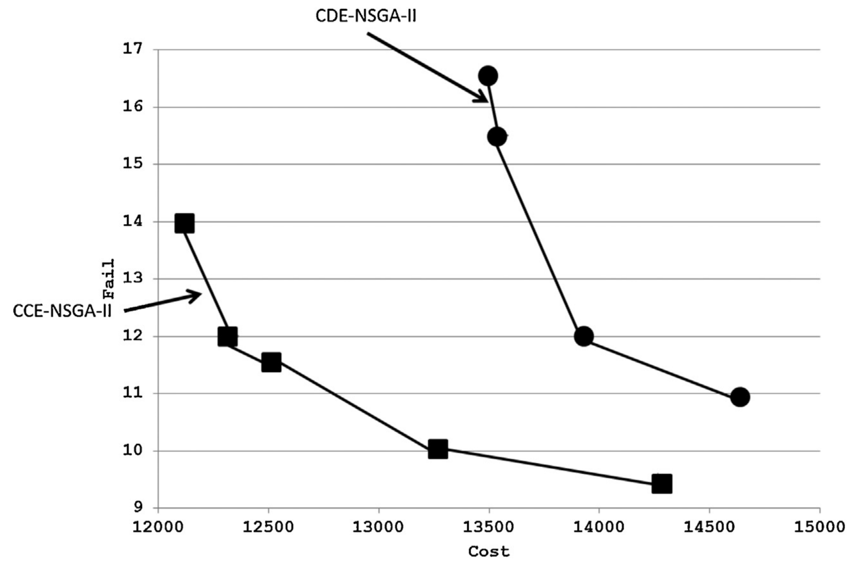

Fig. 13. The non-dominated solutions obtained by a test set found by *CDE-NSGA*-II and *CCE-NSGA*-II on Polska network.

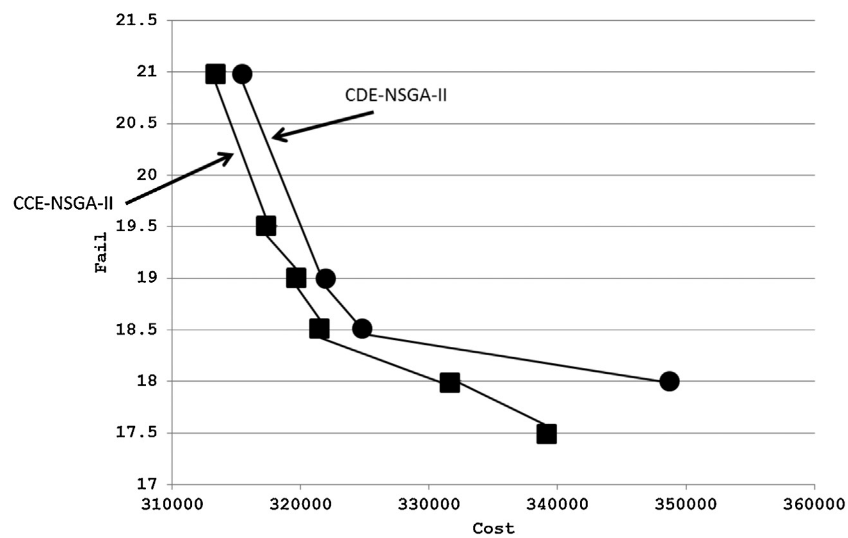

Fig. 14. The non-dominated solutions obtained by a test set found by *CDE-NSGA*-II and *CCE-NSGA*-II on Germany network.

With *CDE* encoding, working path and back up path databases are created, so the computational time was reduce. It was easy to set up crossover and mutation operators. However, the search space is limited due to the fixed working and back up path database. It is difficult to make the database. So, it is not easy to find global optimization. To overcome, complete connection encoding (*CCE*) will be introduced in the next section. The search space of *CCE* encoding is larger than *CDE*, so we hope that we can find global optimization.

A Genetic algorithm with complete connection encoding

Complete connection encoding

For genetic algorithms, encoding is the most important step. There are some methodologies to encode each individual in a population, such as binary encoding, integer encoding . . ., but in this paper, we propose a new encoding mechanism, called complete connection encoding (*CCE*), to encode individuals in *GA*. An individual built by *CCE* is presented as follows: each individual T (i.e. a complete solution) is a set of substrings. Each substring T_i, representing demand i, includes two parts (“working Path wP_i”, “backup Path bP_i”) as in Fig. 1 where “wP_i” and “bP_i” represent the working path and the backup path of the demand i, respectively (Figs. 2 and 3).

To create an individual T, we create each of its substring T_i in turn. wP_i is built using a path finding algorithm. After that, all links of this path will be deleted from the graph so as to find a backup path for the ith demand. Therefore, to initiate an individual that represents a solution of *MA-SNDP*, we need the time is $O(n^2) + O(n)$. In *CCE*, a pair of a working path and a backup path, which is the solution to a demand, is showed clearly in the structure of an individual. We do not need to refer to any outside databases. Therefore, the running time for encoding also decreases considerably. Therefore, the time for encoding also decreases considerably. The diversity of connections in *CCE* is greater than *CDE* [18], which is described in the next section, because the number of chosen connections in *CCE* is unlimited.

Genetic operators

Crossover operator: We apply two different crossover operators: one-point crossover and path crossover. In one-point crossover, we combine the substrings from T_1. T_i with $T'_{i+1}, \ldots, T'_n$ to create the child. In path crossover, we combine the working path of the first

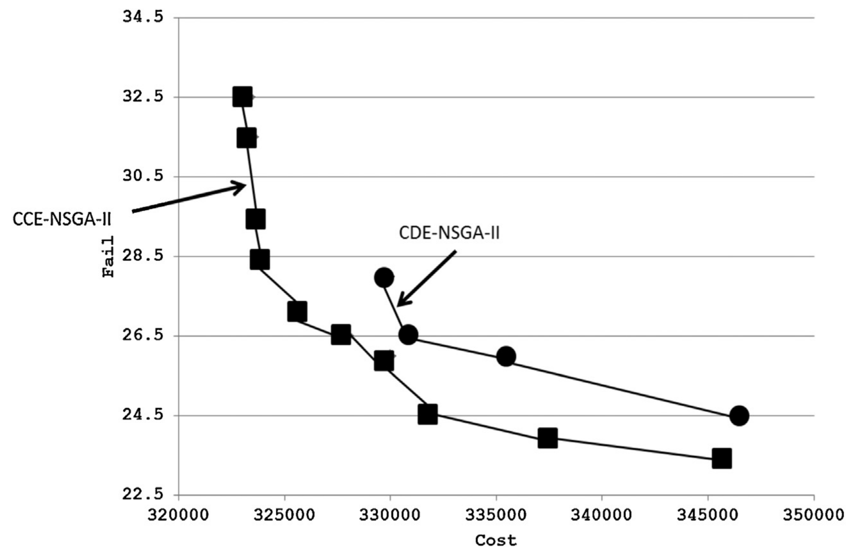

Fig. 15. The non-dominated solutions obtained by a test set found by *CDE-NSGA*-II and *CCE-NSGA*-II on Atlanta network.

parent T with the backup path of the second parent T' to create the child X. With the second crossover, sometimes, the working path and the corresponding backup path are not link-disjoint path any more. Thus, we have to check the child and if any substring violates, it will be replaced by corresponding substring in its parent.

Mutation operator: We choose some individuals in the current population randomly. Then, with each selected individual, we choose one substring i randomly and replace its working path as well as backup path by other couple of link-disjoint paths satisfying the demand i.

Experimental results

Problem Instances

In our experiments, we used three real world instances. They are Polska (12 nodes, 36 links, 65 unicast, 12 anycast), Germany (17 nodes, 52 links, 119 unicast, 13 anycast), and Atlanta (26 nodes, 82 links, 234 unicast, 22 anycast). With each instance, we randomly create 10 test sets that are different from each other in the content of customers' demands.

System setting

We selected *NSGA*-II for our main investigation of multi-objectivity. Experimented four multi-objective genetic algorithms for solving *MA-SNDP*: *CCE-NSGA*-II and *CDE-NSGA*-II. In order to confirm the results, we also used *SPEA*2 for the experiments, resulting *CCE-SPEA*2 and *CDE-SPEA*2.

As stated above, we propose to deal with *A-SNDP* by formulating *A-SNDP* as a multi-objective problem (*MA-SNDP*) and being solved by a multi-objective approach. So, it is necessary to verify the performance of our multi-objective approach on *MA-SNDP* in comparison with the performance of the single objective approach on *A-SNDP*. In order to do so, we also experimented *CCE-GA* on *A-SNDP*, which is similar to *MA-SNDP*. The best cost values obtained by *CCE-GA* and *CCE-NSGA*-II will be compared; also compared the performance of *CCE-NSGA*-II and *CDE-NSGA*-II (regarding both *Ncost* and *NFail*). We compare the experiment results of *CCE-NSGA*-II, *CCE-SPEA*2 to evaluate the efficiency of *CCE*(*CDE*)-*NSGA*-II for solving *MA-SNDP*.

In the experiments, the system was run 50 times for each test set. All the programs were run on a machine with Intel Core 2 Duo U7700, RAM 2GB, Windows 7 Ultimate, and were implemented in Java language.

Our computational experiments were performed under the following parameter specifications: The population size (the number of individuals) is 300, the termination condition (the number of generations) is 300 generations, the one-point crossover probability is 0.33, the path crossover probability is 0.33, and the mutation probability is 0.03. These parameter values were obtained from preliminary computational experiments using a wide range of values for each parameter such as {100, 200, 300, 400, 500} for the population size and {0.01, 0.02, 0.03, 0.04} for the mutation probability. Since we have the two mutation operations, the mutation probability of each mutation operation is about a half of a frequently-used mutation probability in the range [0.5, 1.0].

Computational results

The effect of multi-objectivity

In order to investigate this effect, we selected results obtained from *CCE-NSGA*-II. Note that the similar behavior was also shown by *CDE-NSGA*-II (see the next section). Fig. 4 illustrates the performance of our proposed approach is illustrated by the obtained non-dominated solutions from a single run of *CCE-NSGA*-II.

It is quite obvious that the figure showed a well-spread set of non-dominated solutions obtained by our approach. This indicates a strongly conflicting relation between the two objectives. If they correlate with each other, there will be no trade-off solutions found as in the figure.

Fig. 4 shows the values of the two objectives of individuals in Pareto boundary after 300 generations on a test set in Polska network, with vertical axis represents *NFail* value, and horizontal axis represents *NCost* value. We can see that if the value of *NFail* increases, the value of *NCost* will decrease and vice versa. Therefore, the experiments also prove that we cannot optimize these both objectives at the same time.

In Fig. 5(a), the nodes enclosed by a red line are used by the most connection demands, especially by node 7. The disparity of the connection demands which use the nodes inside red boundary with the ones outside red boundary density is great, the average connection demands density of inside is 9 and outside is 3.5. If one of these nodes is false, it will affect the whole network.

In Fig. 5(b), there is not the disparity of the connection demands density between nodes in network, the connection demands cover regularly on the whole network. We divide this network into three areas, the average connection demands density is 7.4 in the yellow area 6.6 in the blue area, and 6.0 in the red area. The disparity of the connection demands density is quite low. This will prevent the false node from affecting the whole network. However, the building cost is very high. Therefore, the points are more right in Fig. 4, the disparity of the connection demands density on the nodes is lower but the building cost is greater.

Fig. 6 represents the value of objectives of each individual in Pareto boundary after 300 generations of a test set in Atlanta network.

With the left-most point in Fig. 6, the nodes that have the most of the connection demands density are in red boundary, especially by node 8. The average connection demands density of inside is 23.6, and outside is 8.9. Fig. 7 shows that, if one of nodes in the red boundary is false, it will affect the whole networks.

A completely different state of network is represented by the right-most point in Fig. 6. The disparity of the connection demands density between nodes in network is not high, the connection demands cover regularly on the whole network. The average connection demands density of yellow area is 11.1, of blue area is 12, and red area is 13.5. This will restrict the influence of false node to the whole network. However, it is very costly. Therefore, the points are more right in Fig. 6, the disparity of the connection demands density on the nodes is lower but the building cost is greater.

The value of objectives of each individual in Pareto boundary after 300 generations of a test set in Germany network is represented in Fig. 8.

In Fig. 9(b), the connection demands cover regularly on the whole network. The average connection demands density of yellow area is 7.5, of blue area is 8.9, and red area is 8.9. Therefore, the disparity of the connection demands density between nodes in network is not high, and this will prevent the bad influence of false node, but the building network cost is too great.

Fig. 9(a) shows the network state with the left-most point represented in Fig. 8. The red boundary holds nodes that have the most of the connection demands density, especially by 10. The average connection demands density of inside is 17.2, and outside is 4.95. With the quite great disparity of the connection demands density, it is obvious that if the inside nodes are false, it will affect the whole network.

The experiment results in Figs. 10–12 show that the cost of network found by *CCE-GA* and *CCE-NSGA*-II quite equivalent.

The effect of representation

In order to gain the better understanding of our multi-objective design, we also compare the effect of different encoding schemes.

That is why we selected both *CCE* and *CDE* for the comparison. We tested them, recorded results and visualized the results of all three networks on Figs. 13–15.

It is quite clear that both schemes algorithms found spreading non-solutions in all three networks. Once more, this shows the effect of multi-objectivity on this design problem. However, the performance of *CCE-NSGA*-II seemed slightly better that of *CDE-NSGA*-II. Note that in *CCE*, we used the shortest path for mutation, and that might make the algorithm converged quicker than that in *CDE*. And also, with *CDE*, the there is a need to spend a large effort on creating databases for possible paths.

A comparison of performance between MOEAs

In order to confirm the results of *CCE* (*CDE*)-*NSGA*-II, we compare the results with that of *SPEA*2. The obtained non-dominated solutions found by *CCE-NSGA*-II and *CCE-SPEA*2 were together visualized on Figs. 16–18 for all three types of networks. These figures show a clear better set of non-dominated solutions found by *NSGA*-II over *SPEA*2s without any overlapping. Hence we do not need any type of numerical metrics for this comparison. Note that the results of *CDE-NSGA*-II in comparison with that of *CDE-SPEA*2 were in a similar trend as of *CCE*; hence we do not visualize them.

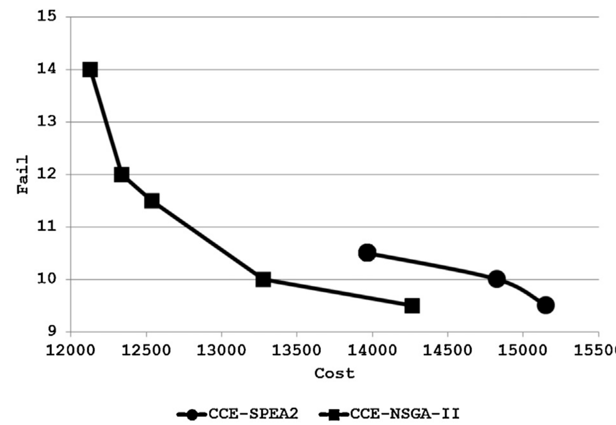

Fig. 16. The non-dominated solutions obtained by a test set found by *CCE-NSGA*-II and *CCE-SPEA*2 on Polska network.

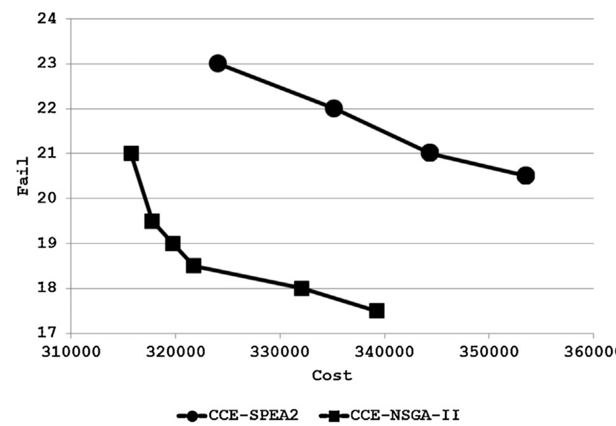

Fig. 17. The non-dominated solutions obtained by a test set found by *CCE-NSGA*-II and *CCE-SPEA*2 on Germany network.

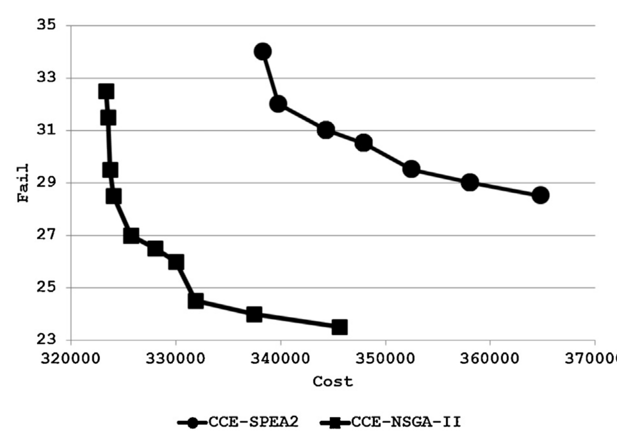

Fig. 18. The non-dominated solutions obtained by a test set found by *CCE-NSGA*-II and *CCE-SPEA*2 on Atlanta network.

Conclusion

In this paper, we proposed a multi-objective design approach for *A-SNDP*. With these multi-objective design method, we found that a good network design depends not only on the network cost but also on many other factures. In particular, we clearly demonstrated the existence of the tradeoff relation between the network cost and the network failure through computational experiments using the proposed multi-objective approach. We experimented on three instances which are Polska, Germany and Atlanta network [2,13]. With each instance, we randomly create 10 test sets that are different from the content of customers' demands. The experimental result showed a great deal on trading off the cost over the failures.

In the future work, we will experiment with more objectives to find better design in the hope of approaching the optimal solution to the problem.

Acknowledgement

This work was supported by the project "Models for the next generation of a robust Internet," grant number 12/2012/HD-NDT, funded by the Ministry of Science and Technology, and Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 107.04.2011.09.

Appendix A. Appendix

```
Procedure CCE-NSGA-II
Begin
Population P, P';
Individual parent1, parent2, child1, child2;
initPopulation(&P);
i = 0;
while (i < generation)
Begin
P' = Ø;
j = 0;
while (j < numParent)//numParent
Begin
selectParent(P, &parent1, &parent2);
crossoverCCE(parent1, parent2, &child1, &child2);
mutationCCE(&child1, &child2, rateMutation);
P' = P' ⋃ child1;
P' = P' ⋃ child2;
j = j + 1;
End;
P = P ⋃ P';
NSGA(&P);
selection(&P);
i = i + 1;
End;
End.
```

```
Procedure CDE-NSGA-II
Begin
Population P, P';
Individual parent1, parent2, child1, child2;
initPopulation(&P);
i = 0;
while (i < generation)
Begin
P' = Ø;
j = 0;
while (j < numParent)
Begin
selectParent(P, &parent1, &parent2);
crossoverCDE(parent1, parent2, &child1, &child2);
mutationCDE(&child1, &child2, rateMutation);
P' = P' ⋃ child1;
P' = P' ⋃ child2;
j = j + 1;
End;
P = P ⋃ P';
NSGA(&P);
selection(&P);
i = i + 1;
End;
```

References

[1] Krzysztof Walkowiak, A flow deviation algorithm for joint optimization of unicast and anycast flows in connection-oriented networks, in: Osvaldo Gervas (Ed.), Computational Science and Its Applications—ICCSA 2008, LNCS, vol. 5072, Springer, Perugia, Italy, 2008, p. 797.

[2] Jakub Gładysz, Krzysztof Walkowiak, Tabu search algorithm for survivable network design problem with simultaneous unicast and anycast flows, Int. J. Electron. Telecommun. 56 (1) (2010) 41–48.

[3] A. Krzysztof Walkowiak, New function for optimization of working paths in survivable MPLS networks, in: Computer and Information Sciences—ISCIS 2006, Springer, Istanbul, 2006, pp. 424–433.

[4] W. Grover, Mesh-based Survivable Networks: Options and Strategies for Optical, MPLS, SONET and ATM Networking, Prentice Hall PTR, Upper Saddle River, NJ, 2004.

[5] V. Sharma, F. Hellstrand, Framework for MPLS-based recovery, in: RFC 3469, 2003.

[6] J. Vasseur, M. Pickavet, P. Demeester, Network Recovery: Protection and Restoration of Optical, SONET-SDH, IP and MPLS, Morgan Kaufmann, San Francisco, CA, 2004.

[7] A. Kasprzak, Algorithms of Flow, Capacity and Topology Structure in Computer Networks Monography, Polish, Wroclaw, 1989.

[8] M. Pioro, D. Medhi, Routing, Flow, and Capacity Design in Communication and Computer Networks, Morgan Kaufmann Publishers, USA, 2004.

[9] K. Walkowiak, Anycast communication—A new approach to survivability of connection-oriented networks, in: Communications in Computer and Information Science, Springer, Berlin, 2003, pp. 378–389.

[10] Volker Nissen, Gold Stefan, Survivable network design with an evolution strategy, in: Success in Evolutionary Computation, Springer, Berlin, 2008, pp. 263–283.

[11] D. Johnson, S. Deering, Reserved IPv6 subnet anycast addresses, in: RFC 2526, 1999.

[12] CDNS, Anycast vs Unicast. http://communitydns.eu/Anycast.pdf [Accessed: 15/7/2014].

[13] Jakub Gladysz, Krzysztof Walkowiak, Optimization of survivable networks with simultaneous unicast and anycast flows, in: ICUMT, Poland, 2009, pp. 1–6.

[14] Roberto Battiti, Mauro Brunato, Franco Mascia, Reactive Search and Intelligent Optimization, Springer, New York, NY, 2008.

[15] Huynh Thi Thanh Binh, Pham Vu Long, Nguyen Ngoc Dat, Nguyen Sy Thai Ha, Heuristic algorithms for solving survivable network design problem with simultaneous unicast and anycast flows, ICIC (1) (2012) 137–144.

[16] Eduardo G. Carrano, Luiz A. E. Soares, Ricardo H. C. Takahashi, Rodney R. Saldanha, and Oriane M. Neto, Electric distribution network multi-objective design using a problem-specific genetic algorithm, IEEE Trans. Power Delivery 21(2) 995–1005.

[17] Eckart Zitzler, Marco Laumanns, Stefan Bleuler, A tutorial on evolutionary multiobjective optimization, in: Metaheuristics for multiobjective optimisation, Springer, Berlin, Heidelberg, 2004, pp. 3–37.

[18] Huynh Thi Thanh Binh, Son Hong Ngo, Dat Ngoc Nguyen, Genetic algorithm for solving survivable network design with simultaneous unicast and anycast flows, BIC-TA (2013) 1237–1247.

[19] A. Hervé Kerivin, Ridha Mahjoub, Design of survivable networks: a survey, J. Netw. 46 (1) (2005) 1–21.

[20] K. Deb, Multiobjective Optimization using Evolutionary Algorithms, John Wiley and Son Ltd., New York, NY, 2001.