Human-centric Computing and Information Sciences
a SpringerOpen Journal

RESEARCH Open Access

# All capacities modular cost survivable network design problem using genetic algorithm with completely connection encoding

Huynh Thi Thanh Binh* and Son Hong Ngo

* Correspondence: binh.huynhthithanh@hust.edu.vn
School of Information and Communication Technology, Hanoi University of Science and Technology, Hanoi, Vietnam

## Abstract

We study the survivable network design problem (SNDP) for simultaneous unicast and anycast flows in networks where the link cost follows All Capacities Modular Cost (ACMC) model. Given a network modeled by a connected, undirected graph and a set of flow demands, this problem aims at finding a set of connections with a minimized network cost in order to protect the network against any single failure. This paper proposes a new Genetic Algorithm with an efficient encoding to solve the SNDP in networks with ACMC model (A-SNDP). Our encoding scheme is simple and allows large search space. Extensive simulation results on real large topology instances show that the proposed algorithm is much more efficient than the Tabu Search and other conventional Genetic Algorithm in terms of minimizing the network cost.

**Keywords:** Survivable network design; All capacities modular cost; Anycast; Unicast; Genetic algorithm

## Introduction

There are many types of connection for data transmission over the Internet. The most popular type of connection is from one node to another, which is called unicast. An anycast connection is also from one node to another, the difference is that the destination node has a one or many replicated servers which back up for it. Anycast has in recent years become increasingly popular for adding redundancy to many Internet services [1–3]. 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), software distribution. The popularity of anycast technology will increase in the near future, since many new services that use both unicast and anycast paradigms are being developed.

In the Internet, any network failure can cause serious consequences, e.g., a single link failure affected more than 30,000 users and it takes 12 hours to fix [3]. Therefore, the design of survivable networks is a crucial problem. For the commnunication networks, the research in [4] presented detailed information on protecting as well as restoring them with specifies techniques such that organization time and money can be saved. Also, in the survivable network design problem (SNDP) with simultaneous unicast and anycast flows, the common objective is to minimize the network cost to protect network against failures. To guarantee the survivability, we adopt the protection approach [5–10] in that 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.

In [5] Gładysz et al. have considered the SNDP for networks using ACMC (All Capacities Modular Cost) link cost model (here after called by A-SNDP). In the ACMC model, a link has many bandwidth levels, each level has a corresponding cost. Many flow demands can go through the same link thus the link cost is defined as the total of required bandwidth from all demands on that link. The network cost is defined as the total of all link cost. The A-SNDP problem is defined as follows. Given a network modeled by an undirected graph where the link cost follows ACMC model and a set of survivable flow demands between node pairs with correspon-ding bandwidth and type of connection (anycast or unicast), this problem aims at finding a set of connection for all flow demands such that the network cost is minimized. The authors in [5] have also proposed a heuristic for A-SNDP using Tabu Search. However, their result is still far from optimal approach.

Genetic algorithm (GA) has been proved effective on NP-hard problem [11]. There have been a lot of research works on using GA approach to solve NP-hard problem, particularly in net-work design problem [9,12]. In our previous work [12], we have developed a genetic algorithm for A-SNDP called CDE-GA that uses Connection Database Encoding for individual representation. However, this encoding method is complex and search space is not diverse enough, thus it limits the performance of GA approach in solving A-SNDP problem.

In this paper, we propose new individual encoding scheme called Complete Connection Encoding (CCE) with genetic algorithm for solving A-SNDP. CCE encoding is simple, and it helps to create more new individuals and enlarge the search space to find better solution in large network instances. We then design the evolution operators using CCE and simulate our proposed algorithm with three typical network instances (Polska, Germany, Atlanta) and two large instances (Germany50, America). We compare the results with Tabu Search [5] and CDE-GA. Results obtained in experimentation show that our proposed algorithm are much better than the compared algorithms in terms of minimizing network cost.

The rest of this paper is organized as follows. Section "Problem description and related works" describes problem and the related works. Our new approach to encode individual and the proposed GA algorithm to solve A-SNDP is shown in Section "Genetic algorithm for solving A-SNDP". Experimental results are given in Section "Experimental results". The paper concludes with Section "Conclusion" by discussion and future works.

## Problem description and related works

The SNDP is generally presented in [13] that considers both economics and reliability in telecommunication network design. The SNDP has to guarantee the survivability of a network system against single or multiple failures and also to minimize the network cost. In both cases, single [5–10] and multiple failures [14], the most popular way mentioned in the literature is the single backup path approach. The main idea of this method is as follows: Each connection has a working path and a backup path, if there is a single link failure on the working path, the failed connection is then switched over its backup path [5–10]. In this work, we consider the protection network from any single failure and assume that one backup path is dedicated for one working path. The approach of using shared backup path [13] is not considered in the scope of this paper.

The A-SNDP problem is described as follows. Given an undirected graph G = (V, E) and a set of customer demand between node pairs with corresponding bandwidth and type of connection (anycast or unicast). Like a unicast connection, an anycast connection is also from one node to another, the difference is that the destination node has one or many replicated servers which back up for it (see Figure 1). In this problem, the cost of a link follows All Capacities Modular Cost model [3], that means a link has many bandwidth levels, each level has a corresponding cost. Many flow demands can go through the same link thus the cost of a link is defined as the total of required bandwidth from all demands on that link. The goal is to find a set of connection for all demands such that the network cost (NCost) is minimized:

$$NCost = \sum_i c_i$$

where $c_i = C_k$, if $B_{k-1} < \sum_j Rij < B_k$

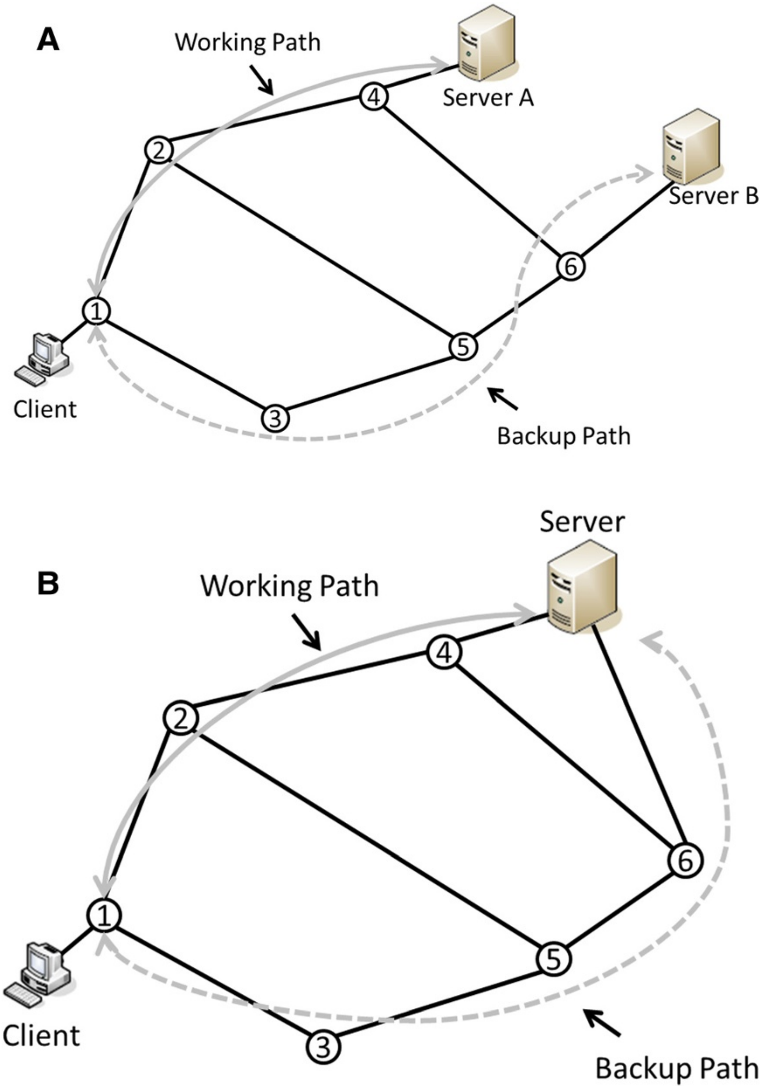


**Figure 1 The example of survivable unicast transmission. A)** Unicast **B)** Anycast.

here, $c_i$ is the cost of link $i$; $B_k$ and $C_k$ is bandwidth and corresponding cost in level $k$; $R_{ij}$ is the required bandwidth from the demand $j$ on the link $i$.

In the literature, there are many papers focus on minimizing the network cost for SNDP problem (see [5,9,15]) and references therein). They use branch – and – bounds or branch – and – cut methods to find optimal solution. These methods can only use for networks with small number of nodes. For larger networks, they have proposed many heuristics such as evolutionary algorithms, tabu search and simulated annealing. In [9], Nissen and Gold applied the evolution strategy (ES) to solve this problem. It is shown by the authors that when using ES, a larger population can achieve better result than a smaller one by avoiding or delaying convergence on local suboptimal. However, this algorithm is 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 [16]. The main idea of this algorithm is based on Flow Deviation method [15] and Local Search algorithm [17]. They achieved a very good result with small networks such as the Polska instance and showed that the average gap of the proposed heuristic to the optimal result is around 7.11%. Furthermore, they also built a Tabu Search algorithm based on hill climbing method with some heuristics to solve this problem [5]. Experiments on three network instances which are Polska (12 nodes, 36 links), Germany (17 nodes, 52 links) and Atlanta (26 nodes, 82 links) showed many promising results. In particular with Polska network (smallest instances), they achieve the average gap to optimal results is 2.57% for 70% anycast/30% unicast case and 2.00% for 80% anycast/20% unicast case. It is observed that by using TS, even on a small network instance like Polska, there is still a significant gap to the optimal result, therefore, the result obtained by Tabu Search algorithm can be further improved.

In [18], Huynh et al. proposed the FBB heuristic (FBB1 and FBB2) for solving A-SNDP. The main idea of FBB1 is based on the use of redundant bandwidth corresponding with paid cost level in each link and that of FBB2 is the combination of FBB1 and the Tabu Search algorithm (TS). Experiments conducted on three network instances (Polska, Germany and Atlanta) show that FBB1 and FBB2 have better result than that of TS [5]. However, the improvement of FBB in compared with TS is still not significant.

In [12], Binh et.al proposed a new algorithm called CDE-GA based on Genetic Algorithm for solving A-SNDP. A new characteristic of this algorithm is to use a new encoding called Connection Database based Encoding (CDE). They also experimented on three instances which are Polska, Germany and Atlanta network [5,16]. With each instance, they randomly create 10 test sets that are different from the content of customers' demands. The results show that the proposed approach is quite effective with A-SNDP. On the big instances (likely Germany and Atlanta network), CDE-GA has much better results than Tabu Search. However, on the small network (likely Polska network), the deviation of the results found by CDE-GA and the results found by Tabu Search is not significant. Other drawback of CDE-GA is that in the mutation operator, only the identification of backup path is changed so it is difficult to create more new paths, thus it limits the search space of GA approach in solving A-SNDP problem.

In the next section, we develop our new Genetic Algorithm by proposing a new scheme to encode an individual for solving the A-SNDP. This scheme is expected to improve the effectiveness in terms of network cost when adopting GA for this problem, especially on large scale network instances.

## Genetic algorithm for solving A-SNDP

### *Individual representation*

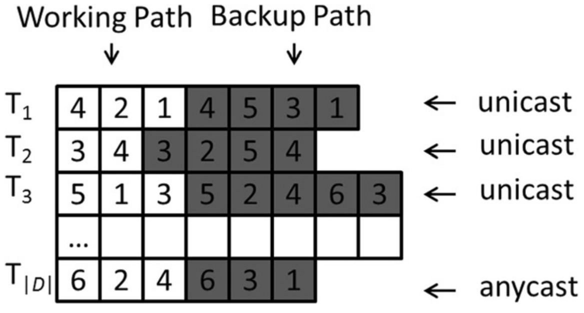
In the design of a Genetic Algorithm, the encoding is the most important task. There are some methods to encode each individual in a population, such as binary encoding, integer encoding… In this paper, we propose a new encoding mecha-nism, called Complete Connection Encoding (CCE) to encode individuals 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 $T_i$, represents a flow demand $i$ and has two parts: the working path and the backup path. Illustration of an individual is shown in Figure 2.

To initialize an individual $T$, we create each of its substring $T_i$ in turn. The working path of $T_i$ is built by using a path finding algorithm. After that, all the link of this path will be deleted from the graph to find the backup path of $T_i$. using the same path finding algorithm. Therefore, to initiate an individual that represents a solution of A-SNDP, we need the time is O(|D|.n$^2$) where $n$ is the number of nodes.

### *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 parent $T$ with the backup path of the second parent $T'$ to create the child $T_{child}$ (see Figure 3). With the second type of crossover, sometimes, the working path and the corresponding backup path are not link-disjoint anymore. Thus, we have to check the child again and if any substring violates the link-disjoint condition, it will be replaced by the corresponding substring from its parent.

**Mutation operator** We choose some individuals in the current population randomly. Then, with each selected indivi-dual, 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 $i$.



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $T_1$ | 4 | 2 | 1 | 4 | 5 | 3 | 1 | | ← unicast |
| $T_2$ | 3 | 4 | 3 | 2 | 5 | 4 | | | ← unicast |
| $T_3$ | 5 | 1 | 3 | 5 | 2 | 4 | 6 | 3 | ← unicast |
| | … | | | | | | | | |
| $T_{\|D\|}$ | 6 | 2 | 4 | 6 | 3 | 1 | | | ← anycast |

**Figure 2 An example representing an individual built by CCE, where each row represents a solution for a demand; |D| is number of flow demands.**

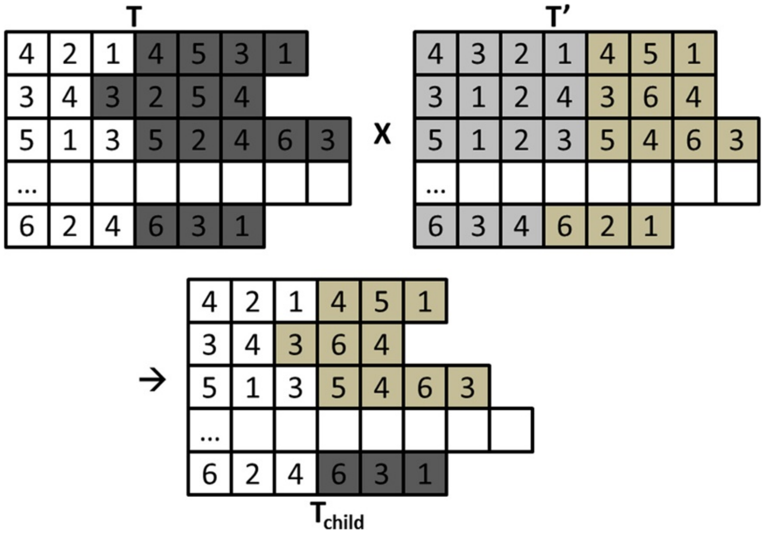


**Figure 3** Illustration of the path crossover operator between parent T and T', which reproduces the child $T_{child}$.

### *Structure of genetic algorithm for solving A-SNDP*

Below is pseudo-code of our proposed algorithm. In line 4, 7 and 14, *noGeneration, noCrossover* and *noMutation* stand for the number of generation, the number of crossover operator and the number of mutation operator, respectively.

```
 1. Procedure CCE-GA
 2. Begin
 3.    initPopulation(P) /*Init a population*/ current = 0
 4.    While(current < noGeneration) {
 5.           Population child = Ø
 6.           h = 0
 7.           While(h < noCrossover) {
 8.           $T_1$, $T_2$ ← SelectParent(P)
 9.                  T' = Crossover($T_1$, $T_2$)
10.                  child = child ∪ T'
11.                  h = h + 1
12.           }
13.           m = 0
14.           While(m < noMutation) {
15.                  T ← SelectParent(P)
16.                  T' = Mutation(T)
17.                  child = child ∪ T'
18.                  m = m + 1
19.           }
20.           P = P ∪ child
21.           P = Selection(P)
22.           current = current + 1
23.    }
   End.
```

**Table 1 Topology instances for experiments**

| Parameters | Networks | | | | |
|---|---|---|---|---|---|
| | Polska | Germany17 | Atlanta | Germany50 | TA2 |
| Nodes | 12 | 17 | 26 | 50 | 65 |
| Links | 36 | 52 | 82 | 176 | 216 |
| Unicast/Anycast | 65/12 | 119/13 | 234/22 | 80/20 | 80/20 |

### Experimental results

We run our proposed algorithm (CCE-GA) independently and compare its performance with the Tabu Search [5] and CDE-GA [12]. All the programs are run on a machine with Intel Core 2 Duo U7700, RAM 2GB, Windows 7 Ultimate, and are installed by C++ and Java language.

### Problem instances

In our experiments, we used five real world instances including the Polska, Germany17, Atlanta, Germany50 and America (TA2) topologies. All can be downloaded from http://sndlib.zib.de [19]. With each instance, we fix the number of flow demands and randomly create 10 test sets which are different from the content of customers' demands (see Table 1).

### Parameters setting

For both CDE-GA and CCE-GA, the number of individual is 300 and the number of generation is 300. This is based on our observation from experiments that when we used larger values, the result is not better. In CCE-GA, we set the one-point crossover probability to 17% and the path crossover probability is also 17%. The selection of crossover probability value is tuned after many experiments with the range from 10% to 50%. The mutation rate is set to a small value (3%). Experiment is repeated 10 times for each test set.

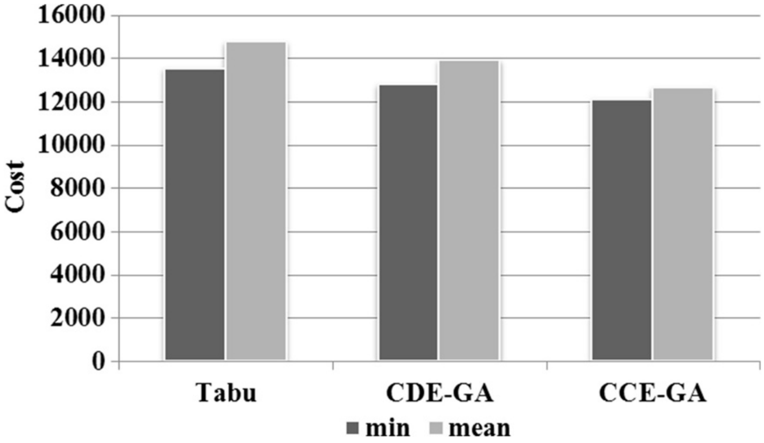


**Figure 4** **Comparision between the best and the average result found by Tabu Search, CDE-GA and CCE-GA on Polska network.**

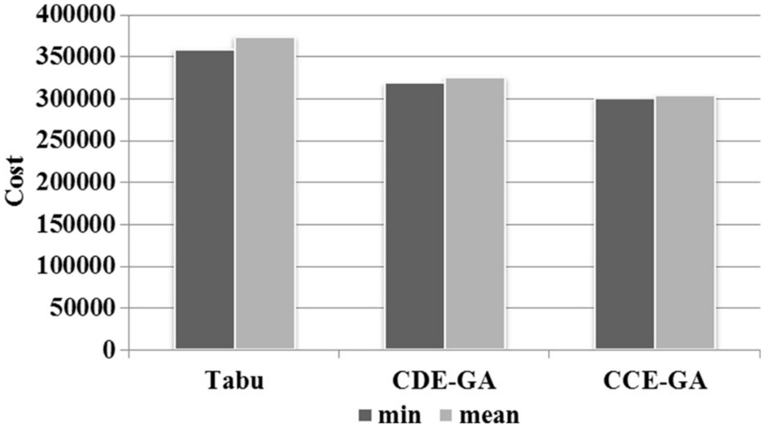


**Figure 5** Comparision between the best and the average result found by Tabu Search, CDE-GA and CCE-GA on Germany17 network.

## Result and discussion

Figures 4, 5, 6, 7 and 8 show the computational results of three compared algorithms on five network instances. For each network, we compare the minimum and the average network cost found. It is notable that the minimum and the average result found by CCE-GA are better than the one found by Tabu Search and CDE-GA on almost problem instances. For example in the Figure 4 (Polska network), the minimum network cost found by CCE-GA are 11% and 6% better than Tabu Search and CDE-GA, respectively.

The average cost found by CCE-GA are 15% and 10% better than Tabu Search and CDE-GA, respectively. The same results can be observed in Figure 8; The best network cost found by CCE-GA are 8% and 4% better than Tabu Search and CDE-GA, while the average network cost found by CCE-GA are better than Tabu Search and CDE-GA about 7% and 6%.

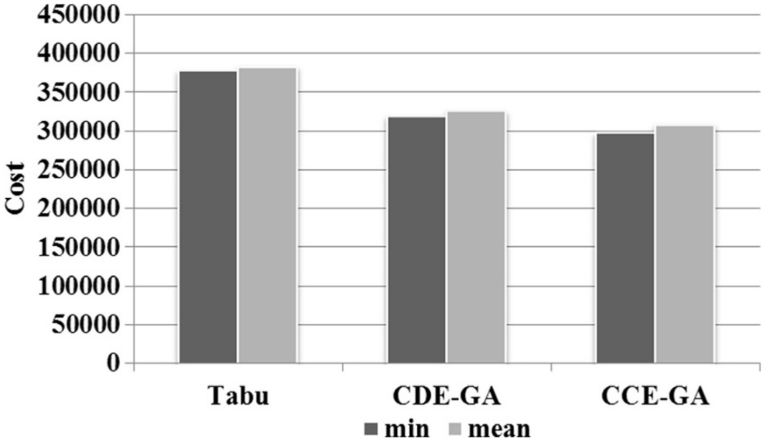


**Figure 6** Comparision between the best and the average result found by Tabu Search, CDE-GA and CCE-GA on Atlanta network.

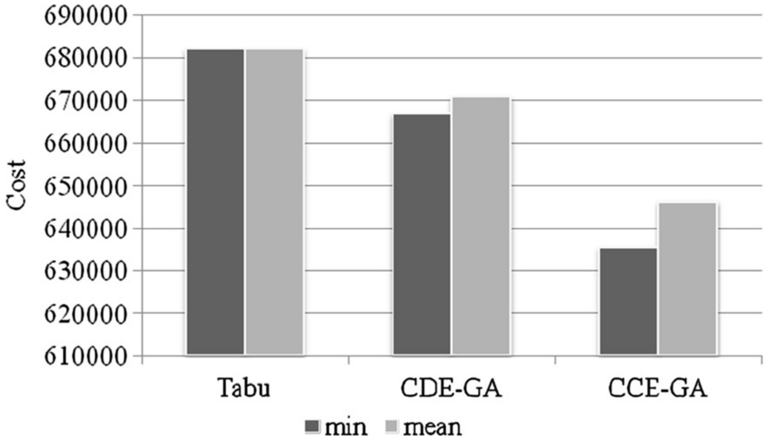


**Figure 7** Comparision between the best and the average result found by Tabu Search, CDE-GA and CCE-GA on Germany50 network.

These above results prove the efficiency of Genetic Algorithms over Tabu Search for this problem. These results also demonstrate the efficiency of our new encoding scheme because it helps to increase the performance of CCE-GA in compare with CDE-GA. As we have explained in previous section, the new encoding scheme can exploit a larger search space. Therefore it can find a better result than that of CDE-GA.

To demonstrate the efficiency of CCE-GA over CDE-GA on large scale networks, we present the detail results on a small network (Polska, Table 2) and on a larger network (TA2, Table 3). For Polska network, the average cost found by CCE-GA are better than that found by CDE-GA on 8/10 problem instances (except the test #3 and the test #5) For TA2 network, the average cost found by CCE-GA are always better than that of CDE-GA. This is because the CDE-GA only works well on small network but not for large scale network. We also observe that for all test sets, the minimum cost found by CCE-GA is much better than that of CDE-GA.

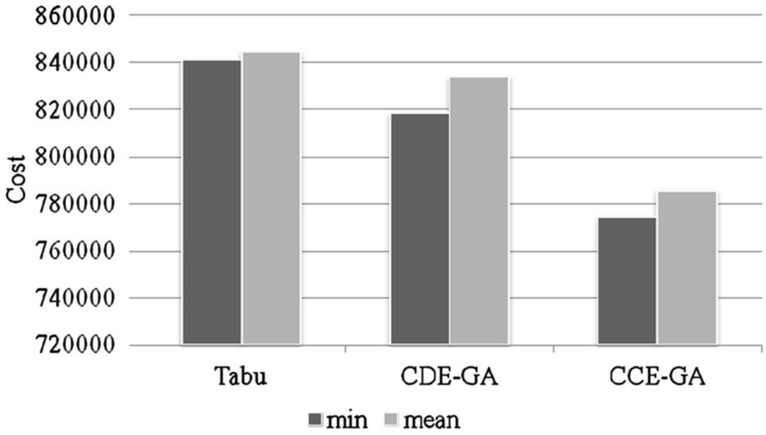


**Figure 8** Comparision between the best and the average result found by Tabu Search, CDE-GA and CCE-GA on TA2 network.

**Table 2 The best and average results found on polska network after 10 running times**

| Test # | Cost | Tabu | CDE-GA | CCE-GA |
|---|---|---|---|---|
| 1 | Min | 13550 | 12812 | 12130 |
| | Mean | 14777 | 13923 | 12653 |
| 2 | Min | 14962 | 13898 | 13200 |
| | Mean | 15270 | 14670 | 13860 |
| 3 | Min | 14010 | 12218 | 12722 |
| | Mean | 15511 | 13289 | 13309 |
| 4 | Min | 11226 | 11484 | 10792 |
| | Mean | 12616 | 12359 | 11045 |
| 5 | Min | 14654 | 13176 | 14054 |
| | Mean | 13691 | 14318 | 14690 |
| 6 | Min | 12714 | 12480 | 11464 |
| | Mean | 12899 | 13262 | 11866 |
| 7 | Min | 16238 | 15850 | 15326 |
| | Mean | 17160 | 16709 | 16305 |
| 8 | Min | 12038 | 12298 | 12168 |
| | Mean | 12320 | 13099 | 12719 |
| 9 | Min | 12042 | 11846 | 11186 |
| | Mean | 12990 | 12386 | 11939 |
| 10 | Min | 15284 | 14590 | 13570 |
| | Mean | 15284 | 15300 | 14556 |

**Table 3 The best and average results found on ta2 network after 10 running times**

| Test # | Cost | Tabu | CDE-GA | CCE-GA |
|---|---|---|---|---|
| 1 | Min | 894440 | 850640 | 848320 |
| | Mean | 905968 | 880571 | 862527 |
| 2 | Min | 843280 | 836120 | 785480 |
| | Mean | 850613 | 837814 | 785480 |
| 3 | Min | 835280 | 817800 | 781480 |
| | Mean | 842613 | 817800 | 781480 |
| 4 | Min | 841280 | 822120 | 798640 |
| | Mean | 842613 | 827586 | 813037 |
| 5 | Min | 858440 | 836120 | 806960 |
| | Mean | 861756 | 844252 | 814589 |
| 6 | Min | 825280 | 806120 | 779800 |
| | Mean | 827413 | 810386 | 789364 |
| 7 | Min | 816120 | 799800 | 778640 |
| | Mean | 818986 | 808289 | 786980 |
| 8 | Min | 841280 | 832120 | 784960 |
| | Mean | 844280 | 835773 | 794862 |
| 9 | Min | 831280 | 814960 | 771480 |
| | Mean | 834746 | 821565 | 784582 |
| 10 | Min | 841280 | 818640 | 774320 |
| | Mean | 844480 | 834137 | 785542 |

**Table 4 The comarision of the average running time (in second) found by tabu search, cde-ga and cce-ga**

| | Polska | Germany17 | Atlanta | Germany50 | TA2 |
|---|---|---|---|---|---|
| CCE-GA | 18 | 21 | 95 | 1951 | 26216 |
| CDE-GA | 13 | 21 | 78 | 941 | 13232 |
| TabuSearch | 1 | 5 | 36 | 666 | 6514 |

To compare the tested algorithms in terms of running time, we calculate the average running time of all 10 test sets for each algorithm. Table 4 shows that CCE-GA takes much more running times in compare with Tabu Search and CDE-GA. But if the running time for Tabu Search, CDE-GA is increased, the best result is not improved.

The above observation can be explained more clearly in Figures 9 and 10, which show the convergence rate of CDE-GA and CCE-GA on TA2 network. Here, because the gap of the convergence rate between CDE-GA and CCE-GA is too large from the first generation to 14th generation so we draw from the 15th generation. It is notable that CCE-GA converge more quickly than CDE-GA from the 15th generation. cost over the failures.

## Conclusion

In this paper, we proposed a new algorithm called CCE-GA for solving A-SNDP. This algorithm uses a new simple encoding scheme called Completely Connection Encoding (CCE) to enlarge the search space to find better solution in large network instances. Experiments are conducted on real network topologies on show that our proposed approach is very efficient in solving A-SNDP. On the big instances, such as Atlanta, Germany50 and TA2 networks, the best and average result found by CCE-GA are much better than CDE-GA and Tabu Search. However, CCE-GA takes much more running times in compare with Tabu Search and CDE-GA to find best result. But if the running time for Tabu Search, CDE-GA is increased, the best result is not improved.

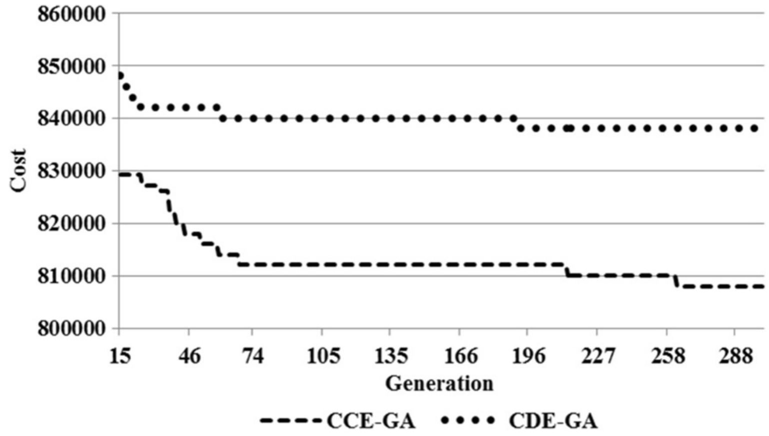


**Figure 9** The convergence rate of CDE-GA and CCE-GA to find the min cost on TA2 network.

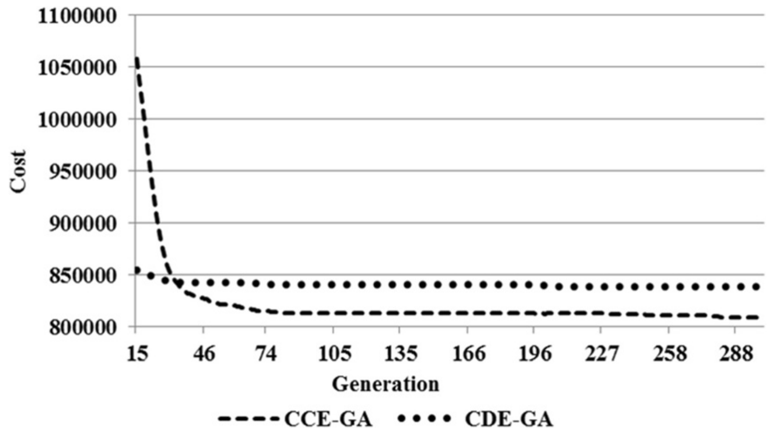


**Figure 10** **The convergence rate of CDE-GA and CCE-GA to find the mean cost on TA2 network.**

In the future, we are planning to improve this algorithm for solving bigger instances in a reasonable time. Moreover, we hope that we can find the other approach with better results for A-SNDP.

### Competing interests

The authors declare that they have no competing interests.

### Authors' contributions

HTTB carried out the design, simulation and drafting the algorithms. SHN participated the design, acquisition of data and helped to draft the manuscript. Both authors read and approved the final manuscript.

### Acknowledgment

This work was supported by the project "*Models for next generation of robust Internet*" funded by the Ministry of Science and Technology, Vietnam under grant number 12/2012-HD-NDT.

Received: 23 April 2014 Accepted: 10 June 2014
Published online: 17 July 2014

### References

1. Johnson D, Deering S (1999) Reserved IPv6 Subnet Anycast Addresses. RFC 2526
2. Ballani H, Francis P (2005) Towards a global IP anycast service. In: SIGCOMM'05. New York, USA, pp 301–312
3. Walkowiak K (2003) Anycast Communication – A New Approach to Survivability of Connection-Oriented Networks. In: Communications in Computer and Information Science. Springer, Berlin, pp 378–389
4. Vasseur J, Pickavet M, Demeester P (2004) Network Recovery: Protection and Restoration of Optical, SONET-SDH, IP and MPLS. Morgan Kaufmann, San Francisco
5. Gładysz J, Walkowiak K (2010) Tabu Search Algorithm for Survivable Network Design Problem with Simultaneous Unicast and Anycast Flows. In: Intl Journal Of Electronics And Telecommunications, vol 56. Versita Publisher, Warsaw, pp 41–48, no. 1
6. Walkowiak K (2008) A Flow Deviation Algorithm for Joint Optimization of Unicast and Anycast Flows in Connection-Oriented Networks. In: Osvaldo Gervas, Computational Science and Its Applications – ICCSA 2008, LNCS, vol 5072. Springer, Perugia, Italy, pp 797–807
7. Walkowiak K (2006) A New Function for Optimization of Working Paths in Survivable MPLS Networks. In: Computer and Information Sciences – ISCIS 2006. Springer, Istanbul, pp 424–433
8. Grover W (2004) Mesh-based Survivable Networks: Options and Strategies for Optical, MPLS, SONET and ATM Networking. Prentice Hall PTR, Upper Saddle River, New Jersey
9. Nissen V, Gold S (2008) Survivable network design with an evolution strategy. In: Success in Evolutionary Computation. Springer, Berlin, pp 263–283
10. Sharma V, Hellstrand F (2003) Framework for MPLS-based recovery. RFC 3469
11. Michalewicz Z (1995) Genetic Algorithms + Data Structures = Evolution Programs, Thirdth edn. Springer
12. Huynh Thi Thanh B, Son Hong N, Nguyen Ngoc D (2013) Genetic Algorithm for Solving Survivable Network Design with Simultaneous Unicast and Anycast Flows. In: the Proceedings of the Eighth International Conference on Bio-Inspired Computing: Theories and Applications. BIC-TA, China, pp 1237–1247
13. Kerivin H, Ridha Mahjoub A (2005) Design of survivable networks: A survey. J Netw 46:1–21
14. Jozsa BG, Orincsay D, Kern A (2003) Surviving Multiple Network Failures Using Shared Backup Path Protection. In: Proceedings of the Eighth IEEE International Symposium on Computers and Communications. Kiris-Kemer, Turkey, June 30-July 03, 2003, pp. 1333
15. Pioro M, Medhi D (2004) Routing, Flow, and Capacity Design in Communication and Computer Networks. Morgan Kaufmann Publishers, San Francisco, CA, USA

16. Gladysz J (2009) Krzysztof Walkowiak: Optimization of survivable networks with simultaneous unicast and anycast flows. Poland, ICUMT, pp 1–6
17. Battiti R, Brunato M, Mascia F (2008) Reactive Search and Intelligent Optimization 1st. Springer, New York, USA, p 182
18. Huynh Thi Thanh B, Pham Vu L, Nguyen Ngoc D, Nguyen Sy Thai H (2012) Heuristic Algorithms for Solving Survivable Network Design Problem with Simultaneous Unicast and Anycast Flows. In: The Eight International Conference on Intelligence on Computing, ICIC-2012. Huangshang, China, pp 137–145
19. Orlowski S, Wessäly R, Pióro M, Tomaszewski A (2010) SNDlib 1.0-Survivable Network Design Library. Networks 55(3):276–286

doi:10.1186/s13673-014-0013-y
**Cite this article as:** Binh and Ngo: All capacities modular cost survivable network design problem using genetic algorithm with completely connection encoding. *Human-centric Computing and Information Sciences* 2014 **4**:13.