



# Network Migration Optimization Using Genetic Algorithms

Stefan Türk, Ying Liu, Rico Radeke, Ralf Lehnert

## ► To cite this version:

Stefan Türk, Ying Liu, Rico Radeke, Ralf Lehnert. Network Migration Optimization Using Genetic Algorithms. 18th European Conference on Information and Communications Technologies (EUNICE), Aug 2012, Budapest, Hungary. pp.112-123, 10.1007/978-3-642-32808-4\_11 . hal-01543164

**HAL Id: hal-01543164**

**<https://inria.hal.science/hal-01543164>**

Submitted on 20 Jun 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Network Migration Optimization using Genetic Algorithms

Stefan Türk, Ying Liu, Rico Radeke, and Ralf Lehnert

Technical University Dresden, Chair for Telecommunications,  
Mommsenstr. 13, 01062 Dresden, Germany  
`tuerk|radeke|lehnert@ifn.et.tu-dresden.de`  
`ying-liu@gmx.de`

**Abstract.** This paper introduces the problem of communication network migration for backbone networks. Heuristic solutions for this problem can be determined by the application of genetic algorithms to the problem. A description of the system model is presented, as well as the used algorithmic approaches and optimization results. Our main goal is the optimization of migration costs, by respecting increasing demands over the migration period, while device costs per bit are decreasing. We will present Crowded DPGA as best found GA to solve the network migration problem.

## 1 Introduction

The traffic demand in optical backbone networks is expected to increase rapidly in the next years [1]. Also the income of network providers is not proportionally increasing to the consumption of network resources due to shrinking tariffs for end users. These forecasts demand a significant cost reduction by strategic planning optimization.

This paper describes how network operators can cope with the increase of backbone traffic in the next years. Network migration describes a technical process of upgrading and exchanging existing hardware/software (infrastructure) to another network technology, which generate calculable cost savings for its owner. The shift from classical Synchronous Digital Hierarchy (SDH) technology towards cost efficient Ethernet services is currently under way, i.e. for high data rate Internet access or global interconnection of company locations [2–4]. The introduction of Carrier Ethernet services is heavily investigated by most providers, since it promises significant cost savings and a simplification in terms of administration and maintenance [5].

In this paper we will focus on the heuristic solution for the specific problem formulation of the migration problem with Genetic Algorithms (GA). This meta heuristic is part of the group of evolutionary algorithms and has been effectively used for various other optimization problems already [6, 7].

The migration is being considered as an multi-period multi layer backbone optimization problem.

The start architecture of our migration approach is an IP/ MPLS/ SDH/ DWDM network. A possible future scenario and migrated network is an IP/ MPLS/ ETH/ DWDM architecture. Node architectures are discussed in detail in [8].

The remaining paper is structured as follows. Chapter 2 describes our network migration model. In chapter 3 GAs are introduced and applied to the migration, followed by the presentation of a reference network in chapter 4. Then our calculation results are shown in chapter 5 and a short conclusion is given in chapter 6.

## 2 Network Migration Model

### 2.1 Introduction

The process of network migration can be described as the stepwise insertion, replacement or removal of components in a network. In our current approach (multi layer migration) Optical Cross Connects (OXC's), Carrier Grade Ethernet switches and IP routers can be added to a network to satisfy the increasing demands of the next years [4]. Within our migration method potential future traffic can be considered in the planning process, giving the advantage of an all period migration optimization.

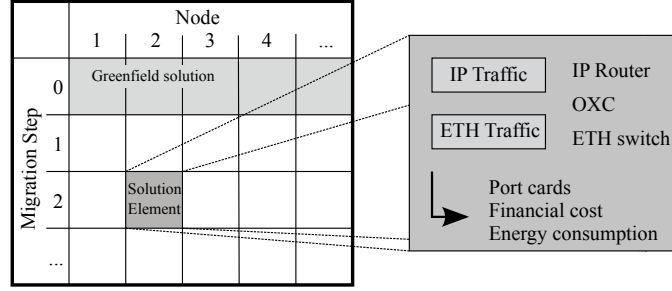
Our network migration model implements for instance two important market-driven factors: on the first hand the network demand increase (40% annual traffic increase [1]) and on the other hand a cost erosion of devices (20% yearly Capital Expenditures (CAPEX) decrease [9]). The detailed description of this cost model can be found in [8].

### 2.2 Solution Model

To solve network migration using algorithms, the appropriate solution to this problem is considered as shown in fig. 1. Each horizontal row represents the information about a specific year, each vertical rank represents the information about a specific node. Every square (or solution element) is regarded as the "gene" inside the solution. It therefore contains information about one node within one year. Elements hold various information, like cost, power consumption, port cards used and slots.

## 3 Genetic Algorithms

GA optimization methods take inspiration from natural evolution and have been widely applied in various fields. In the next sections this meta heuristics will be shortly introduced and the mapping to the migration problem will be shown. Improvements in terms of algorithm performance for our specific problem will be discussed.



**Fig. 1.** Network migration solution  $S$

### 3.1 Classic Genetic Algorithms

Evolutionary Algorithms (EAs) are population-based metaheuristic optimization algorithms. They are inspired by biological evolution using methods such as natural selection, crossover, mutation and survival of the fittest. GAs are the major representatives of EAs, which were firstly researched by *John Holland* and his student *DeJong* in 1975. These are capable of a fast convergence to the optimal solution, which can also be a local optimal solution. This is also the major drawback of this meta-heuristic. It can be minimized using several optimizations, discussed later in this chapter. Basic GA is characterised by the following steps shown in fig. 2a:

**Encoding Individual** The given problem is encoded into a set of individuals. Each individual represents a solution to the given problem, in our case a Migration Solution (MS)  $S$ . An individual can be regarded as a chromosome in a genetic system. Each chromosome consists of genes, which are decision variables for the migration problem. These variables decide the performance of the migration solution or the cost of the individual [7].

**Fitness Function** The quality of new individuals needs to be measured. Since every  $S$  can be evaluated in terms of cost (min) and energy (min) the evolution of the algorithm can be measured using the enhancement of solutions in the designated metric with a fitness function  $f$ . For this paper we chose a cost metric to evaluate the fitness of a solution  $S$ , which is more detailed explained in [4].

**Selection** After each generation a decision has to be made which individuals survive to the next generation. This process is usually performed using generalized replacement, steady state replacement or a fitness based decision (i.e. roulette wheel method [7]). Two individuals  $S_1, S_2$  are then chosen from the new solution pool according to a randomized selection method.

**Crossover** Crossover means interchanging parts of two genotypes, or the exchange of several parts within two solutions. In literature [6] three types are classified:

- single-point crossover: exactly one crossover between two genes
- two-point crossover: exactly two crossovers between two genes
- multi-point crossover: multiple crossovers between two genes

For the crossover of both migration solutions  $S_1$  and  $S_2$  we define a randomly chosen crossover point (uniformly distributed within the length of the migration solution). Whether a crossover happens or not is controlled by the crossover probability  $P_c$  (see tab. 1).

**Mutation** Mutation can be seen as a low probability random change of genes within an individual. In the context of migration this is a random device change within a migration element. The probability of a mutation occurrence is  $P_m$  (see tab. 1). The process of selection, crossover and mutation is illustrated in fig. 3. Two different solutions  $P_1, P_2$  are selected and a crossover point is defined at migration year two and node three. Afterwards child  $C_2$  is mutated in three elements. We only modify devices of a solution element, routing and traffic flows are not affected by mutation.

### 3.2 Crowding

Crowding is a GA enhancement. It is also related to nature and gives the individuals the option to “exploite different niches (resources) in the environment” [10]. The usage of crowding reduces that drawback in comparison to a classic GA, whose individuals very likely converge into one local optimum. The process is described in fig. 2b [11]. After the generation of the initial population all individuals are paired and one pair is chosen. The new offspring is generated by crossover and mutation within this pair. According to a pairing rule each child is paired with one parent to decide if the child should replace the parent.

**Pairing Rule** In [12] pairing describes the process to decide which parent-child pairs are chosen for further improvement. The hamming distance between each child and each parent is taken as criterion for the decision. Having two parents  $P_1, P_2$  and two children  $C_1, C_2$  the pairing decision is made according to eq. 1:

$$\text{if } d(P_1, C_1) + d(P_2, C_2) \leq d(P_1, C_2) + d(P_2, C_1) \quad (1)$$

then use pairs  $P_1$  and  $C_1$ ,  $P_2$  and  $C_2$

else use pairs  $P_1$  and  $C_2$ ,  $P_2$  and  $C_1$

for replacement competition

The distance between migration solution elements can be explained using fig. 3. Since we have no binary problem statement in network migration the amount of different solution elements between two solutions is used within this paper as distance instead of the standard hamming distance.

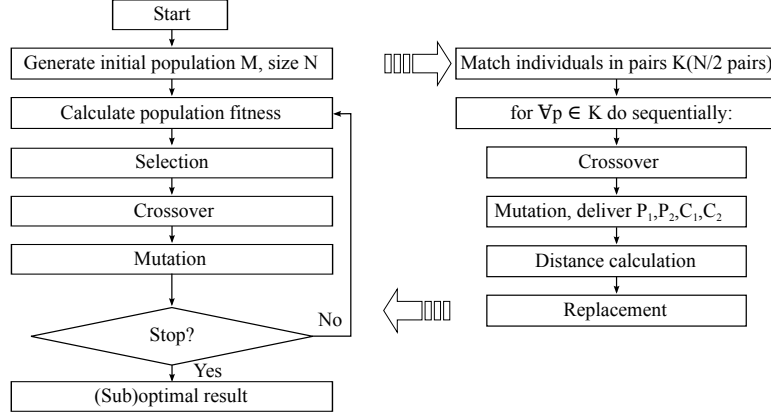


Fig. 2. (a) Basic GA [7]

(b) Crowding GA [11]

**Replacement Rule** After pairing a deterministic or randomized replacement rule is applied to the closest pairs. This phase is used to find the individual within each pair that replaces the old parent [12]. After calculation of fitness values  $f(p)$  and  $f(c)$  for parent and child, replacement rules are applied determining the probability  $p_r$  (that the child replaces the parent). The rule for the deterministic case is given in eq. 2. A rule for a randomized replacement is given in eq. 3 [12]. These equations should be only applied to minimization problems.

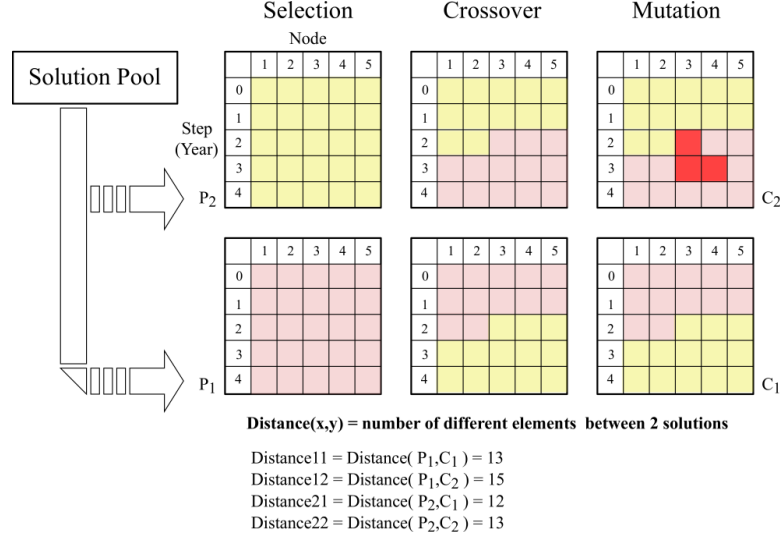
$$p_{r,DET} = \begin{cases} 1 & \text{if } f(c) < f(p) \\ 1/2 & \text{if } f(c) = f(p) \\ 0 & \text{if } f(c) > f(p) \end{cases} \quad (2)$$

$$p_{r,RAND} = 1 - \frac{f(c)}{f(c) + f(p)} \quad (3)$$

### 3.3 Dual Population GA

Another approach for improving GA is the multi-population algorithm which generates two or more subpopulations, providing more individuals to maintain overall diversity. Dual-Population GA (DPGA) is one branch of multi-population GAs, which are shortly introduced in [13]. We implemented the DPGA to solve the migration problem using a main and a reserve population (denoted  $M$  and  $R$ ).  $M$  is used to find the solution with minimal solution cost.  $R$  is necessary to store individuals that strongly differs from the ones in  $M$ .

**Recombination** Since both populations work independent of each other (having their own crossover/inbreeding and mutation phases) a method to exchange information between both populations is necessary. A normal crossover between one individual of each population is applied here (crossbreeding).



**Fig. 3.** Visualizaion of selection, crossover, mutation and distance in the context of network migration

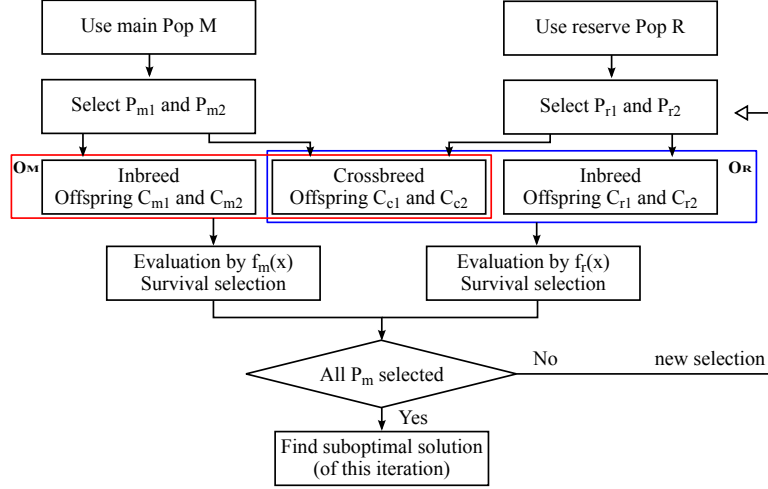
**Fitness Function** Since main and reserve population have different use, their candidates are evaluated using different objective functions. The fitness function for the candidates  $O_M$  of the new main population  $f_m$  is the same as for the classic GA. Every element  $x$  of the candidates  $O_R$  of the reserve population is evaluated within this paper using the fitness function  $f_r$  eq. 4 [14].

$$\min : f_r(\mathbf{x}) = d(\mathbf{m}_{\text{best}}, \mathbf{x}) \quad (4)$$

with:

$\mathbf{m}_{\text{best}}$  : best individual in  $M$   
 $d(\mathbf{m}_{\text{best}}, \mathbf{x})$  : distance between two individuals  $m$  and  $x$

**DPGA Workflow** Figure 4 presents the steps of the DPGA, basing on Park et al. [14]. Their idea is shortly explained in the following. In a first step  $M$  and  $R$  are initialized, afterwards two individuals are selected from each population:  $P_{m1}, P_{m2}$  and  $P_{r1}, P_{r2}$ . Inbreeding and crossbreeding are applied to generate the new offsprings  $C_{m1}, C_{m2}, C_{r1}, C_{r2}$  and  $C_{c1}, C_{c2}$ . Within crossbreeding phase one of the parent individuals ( $P_{m1}$  or  $P_{m2}$ ) is chosen from  $M$ , the other ( $P_{r1}$  or  $P_{r2}$ ) is selected from  $R$  (the one with  $\min(d(P_r, P_m))$ ). Two groups  $O_M$  and  $O_R$  hold the candidates for the next generation of  $M$  and  $R$ . The two fittest individuals of each group are added to the new main and reserve population.



**Fig. 4.** Flowchart of DPGA [11] (one iteration)

### 3.4 Crowded DPGA

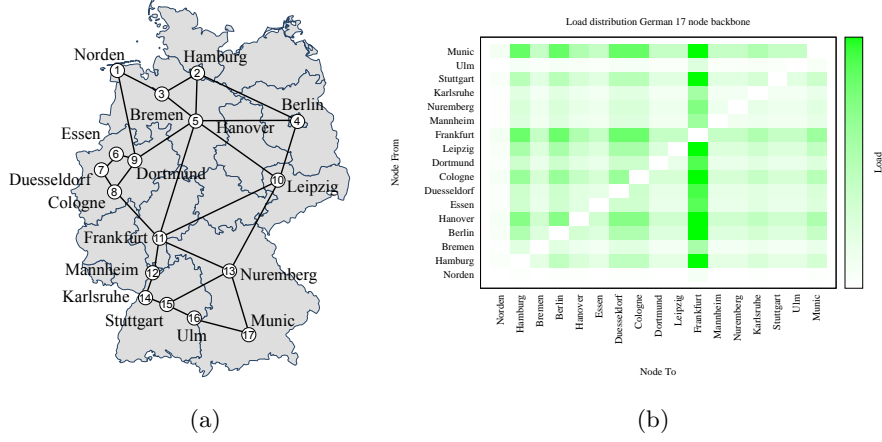
Since crowding and DPGA both improve the performance of the basic GA it is our idea to combine those methods and apply them to the migration problem. Its workflow is similar to DPGA, but the inbreeding phase of  $M$  is replaced by the pairing rule already introduced in section 3.2. Taking advantage of this rule we decide if a child individual replaces a parent after internal crossover and mutation. After this step we take the candidates out of  $O_M$  and  $O_R$  for the new populations. We calculate the fitness of their elements and judge if the stop criterion is already reached (the original size of the main population). The best element of this new  $M$  marks the best found suboptimal solution. A more detailed description of crowded DPGA can be found in [15].

## 4 Reference Network

As reference scenario for our migration we use the German 17-node (fig. 5a) backbone network. The traffic model (fig. 5b) is based on a population investigation that has been described more detailed in [3]. The total migration period is set to 5 years.

All our results are produced on a 64Bit Linux System with 8 Cores and 16GB RAM.



**Fig. 5.** German 17 (GER-17) (a) node backbone scenario (b) and traffic model [8]

Name	Parameter	Range
Population size of $M$	$S_M$	[2,300]
Crossover probability	$P_c$	[0,1]
Mutation probability	$P_m$	[0,1]
Iteration number	$I$	10000
Confidence sample number	$C$	80

**Table 1.** Heuristic parameters

## 5 Results

Figures 6 to 8 show the performance results of the German 17-node backbone network migration. In fig. 6 the presented algorithms are compared for the GER-17 backbone. A bad convergence for random optimization (RO) can be seen, GA and especially crowding DPGA performed best within our experiment.

The influence of main population size setting  $S_M$  was investigated in 7. It can be seen that a semioptimal number of 150 elements performed best in our specific migration scenario. This parameter has to be evaluated more in detail in future experiments using different topologies and node amounts to derive valid statements regarding network migration. Generally a too high  $S_M$  forces the algorithm to work more exploratively, a too low choice tends to a low diversity and therefore to a high addition to the initial solution.

Fig. 8 presents the dependence of crowding DPGA from mutation probability  $P_m$ . We can obtain that a too high  $P_m$  drops the algorithm into a more random working state, which results in a lower convergence rate. An optimal value of  $P_m = 0.02$  was found here.

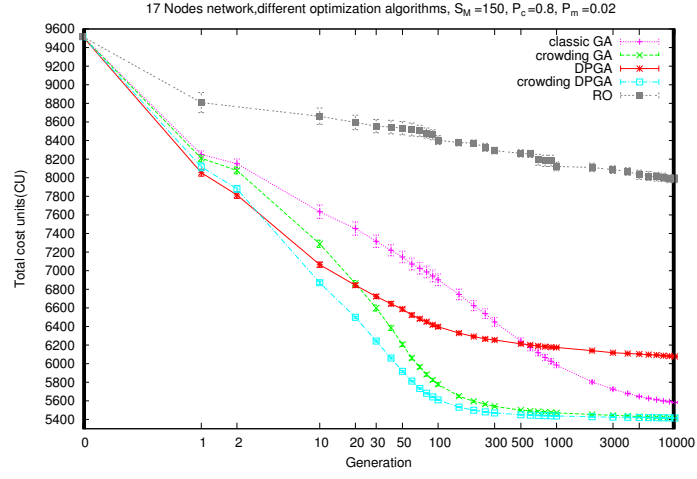
A detailed cost distribution is presented in fig. 9, CAPEX, Implementation Expenditures (IMPEX) and Operational Expenditures (OPEX) values are presented for the optimal result, as well as for the different optical (OP), IP and Ethernet (ETH) basic nodes (BN) and port cards (PC). A high need for ETH equipment can be estimated for the migration years 3 and 4. The negative values for IPBNs mark the disposal of these devices, since these basic nodes can be reduced in size due to more traffic flows on layer two instead of layer three.

The best performance was achieved using the crowded GA with a main population of 150 elements, a  $P_c$  of 0.7, a  $P_m$  of 0.02 and a randomized selection method.

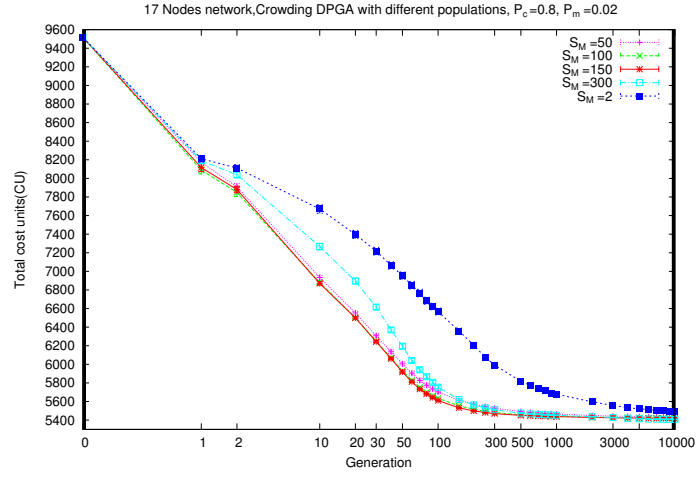
## 6 Conclusion and Perspective

Within this paper the network migration problem was solved using classic GA, crowding GA, DPGA and crowded DPGA algorithms. The modeling of the problem was presented as well as a short introduction to used algorithms and algorithm improvements. Our experiments revealed that the migration problem can be adapted to the needs of GAs and that a further investigation regarding GA is useful for migration since different algorithms behave very differently for our scenario. As best algorithm crowded DPGA was suggested, due to better convergence rate in terms of migration cost.

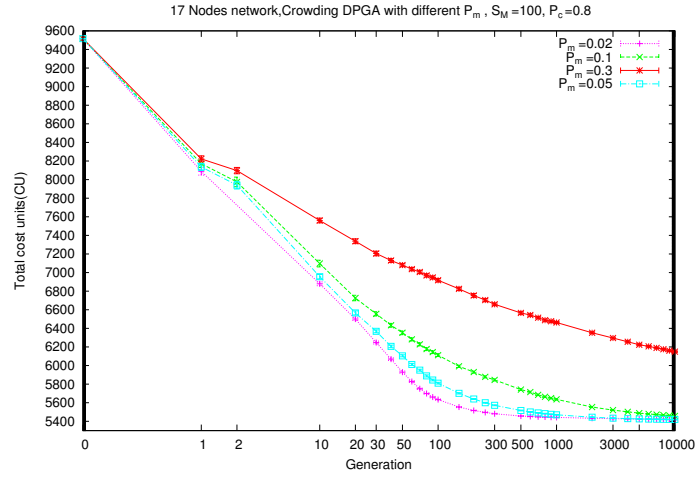
In the future work more scenarios (i.e. european 67-node backbone network) will be investigated to derive general statements which parameters for GA have to be used under which circumstances. Furthermore other solution mappings will be discussed, since the gene structure strongly influences the ability to manipulate solutions using crossover, mutation, etc. Calculation periods will be measured and compared to iterative figures 6 to 8. Afterwards comparisons to other meta heuristics (i.e. PSO) will be presented. A multi-objective investigation using cost and energy factors for the migration should also be provided in future.



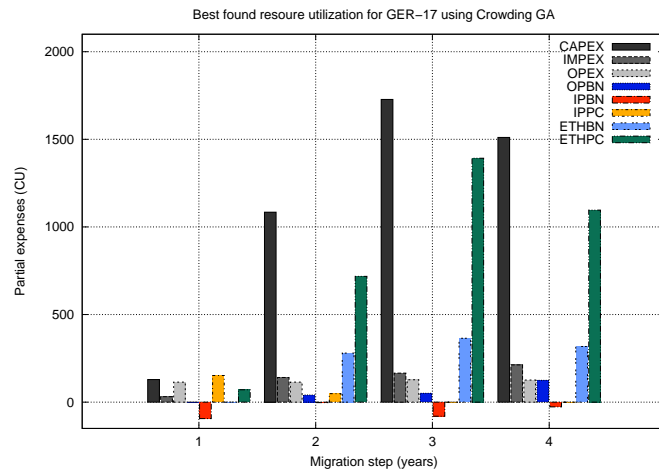
**Fig. 6.** Comparison of different GA migration implementations



**Fig. 7.** Evaluation of main population size



**Fig. 8.** Evaluation of mutation factor



**Fig. 9.** Resource utilization for best migration result

## References

1. Cisco, “Cisco visual networking index: Forecast and methology, 2008-2013,” Tech. Rep., 2009.
2. T. Michaelis, M. Duelli, M. Chamania, B. Lichtinger, F. Rambach, and S. Türk, “Network planning, control and management perspectives on dynamic networking,” in *35th European Conference on Optical Communication*, Vienna, Austria, 2009, p. 7.7.2.
3. S. Türk, R. Radeke, and R. Lehnert, “Network migration using ant colony optimization,” in *9th Conference of Telecommunication, Media and Internet Techno-Economics (CTTE)*, Jun. 2010.
4. S. Türk and R. Radeke, “Optimization of energy efficient network migration using harmony search,” in *17th EUNICE Open European Summer School - Energy-Aware Communications (EUNICE 2011)*, Dresden, Germany, Sep. 2011.
5. Ciena, “The value of otn for network convergence and ip/ethernet migration,” 2009. [Online]. Available: <http://www.ciena.com/files/>
6. S. Sivanandam and S. Deepa, *Introduction to genetic algorithms*. Springer Verlag, 2007.
7. A. Popov, “Genetic algorithms for optimization,” *User Manual*, Hamburg, 2005.
8. S. Türk, S. Sulaiman, A. Haidine, R. Lehnert, and T. Michaelis, “Approaches for the migration of optical backbone networks towards carrier ethernet,” in *3rd IEEE Workshop on Enabling the Future Service-Oriented Internet - Towards Socially-Aware Networks*, Honolulu, Hawaii, USA, 2009.
9. S. Verbrugge, “Strategic planning of optical telecommunication networks in a dynamic and uncertain environment,” Ph.D. dissertation, University of Ghent, 2007.
10. J. Horn, “The nature of niching: Genetic algorithms and the evolution of optimal, cooperative populations,” Ph.D. dissertation, Citeseer, 1997.
11. B. Yuan, “Deterministic crowding, recombination and self-similarity,” in *Evolutionary Computation, 2002. CEC’02. Proceedings of the 2002 Congress on*, vol. 2. IEEE, 2002, pp. 1516–1521.
12. O. Mengshoel and S. Galan, “Generalized crowding for genetic algorithms,” in *Genetic and Evolutionary Computation Conference 2010 (GECCO-10)*, 2010, pp. 775–782.
13. T. Park and K. Ryu, “A dual-population genetic algorithm for adaptive diversity control,” *Evolutionary Computation, IEEE Transactions on*, vol. 14, no. 6, pp. 865–884, 2010.
14. T. Park, R. Choe, and K. Ryu, “Adjusting population distance for the dual-population genetic algorithm,” in *Proceedings of the 20th Australian joint conference on Advances in artificial intelligence*. Springer-Verlag, 2007, pp. 171–180.
15. Y. Liu, “Network migration optimization using genetic algorithms,” Diploma thesis, Technische Universität Dresden, 2011.