# Comparison of Single and Multi-objective Evolutionary Algorithms for Robust Link-state Routing

Vitor Pereira[1], Pedro Sousa[1], Paulo Cortez[2], Miguel Rio[3] and Miguel Rocha[4]

[1] Centro Algoritmi/Dep. of Informatics, University of Minho, Portugal
vitor.pereira@algoritmi.uminho.pt,pns@di.uminho.pt
[2] Centro Algoritmi/Dep. of Information Systems, University of Minho, Portugal
pcortez@dsi.uminho.pt
[3] Dep. of Electronic and Electrical Engineering, University College London, UK
m.rio@ee.ucl.ac.uk
[4] Centre Biological Engineering/Dep. of Informatics, University of Minho, Portugal
mrocha@di.uminho.pt

**Abstract.** Traffic Engineering (TE) approaches are increasingly important in network management to allow an optimized configuration and resource allocation. In link-state routing, the task of setting appropriate weights to the links is both an important and a challenging optimization task. A number of different approaches has been put forward towards this aim, including the successful use of Evolutionary Algorithms (EAs). In this context, this work addresses the evaluation of three distinct EAs, a single and two multi-objective EAs, in two tasks related to weight setting optimization towards optimal intra-domain routing, knowing the network topology and aggregated traffic demands and seeking to minimize network congestion. In both tasks, the optimization considers scenarios where there is a dynamic alteration in the state of the system, in the first considering changes in the traffic demand matrices and in the latter considering the possibility of link failures. The methods will, thus, need to simultaneously optimize for both conditions, the normal and the altered one, following a preventive TE approach towards robust configurations. Since this can be formulated as a bi-objective function, the use of multi-objective EAs, such as SPEA2 and NSGA-II, came naturally, being those compared to a single-objective EA. The results show a remarkable behavior of NSGA-II in all proposed tasks scaling well for harder instances, and thus presenting itself as the most promising option for TE in these scenarios.

**Keywords:** Multi-objective evolutionary algorithms, Traffic Engineering, NSGA, SPEA, intra-domain routing, OSPF

## 1 Introduction

Link-State protocols, such as Intermediate-System to Intermediate-System (ISIS) [12] and Open Shortest Path First (OSPF)[11], are widely used routing protocols. Positive weights, assigned to each link in the network, are used to compute the shortest path (SP) between each source-destination pair, through which network traffic flows. The SPs are obtained using the Dijkstra algorithm [5], and minimize the total sum of link weights in the path. Thus, link weights define how traffic is accommodated onto the underlying network topology, being, in this context, the most important decision factor for the configuration of the traffic routing process. The decision making involved in link weights configuration, usually performed by a network administrator, is not an easy task when the scale of the network and the typically high volume of traffic and flows are taken into consideration. If inadequate, a configuration can cause the misallocation of traffic into the available resources, resulting in packet loss, increasing delays, and, potentially, in the unfulfillment of service level agreements (SLAs).

The Traffic Engineering (TE) problem addressed by this work arises in this context. It consists in finding a set of weights that optimize the congestion levels of the network, for which there are known aggregated traffic demands specified for each source-destination pair. This NP-hard optimization problem has been covered in previous efforts [8, 1] with good results, resorting to several optimization approaches, which include, for instance, Evolutionary Algorithms (EA) in previous work by the authors [14, 13]. Indeed, EA based approaches to TE have been proven to deliver near optimal solutions for the weight setting problem with several advantages when compared with other optimization techniques. Their ability to provide a set of possible solutions, with distinct trade-offs between objectives, enables network administrators to choose from a broader set of configurations, and consequently offers a conscious choice of the most adequate solution. However, distinct EAs have different merits and limitations [15] and consequently some approaches may not offer equally good solutions.

In this context, the present work offers a comparative study of three popular EAs spanning both single and multi-objective alternatives: the Non-dominated Sorting Genetic Algorithm (NSGA-II), the Strength Pareto Evolutionary Algorithm (SPEA2) and a Single-Objective Evolutionary Algorithm (SOEA) previously proposed by the authors. The experimental study allows to compare the performance of the three approaches in two extensions of the described problem, where the weights need to be set for scenarios considering the network's dynamic behavior, namely considering changes on the traffic demands over distinct time periods, in the first case, and the possibility of a single link failure, in the latter case.

The paper proceeds with section 2, describing the experimental model, the framework that sustained the experiments and EAs configuration; section 3 presents the results for the scenario with two distinct traffic demand matrices; section 4 presents the results for the scenarios with a single link failure; finally, section 5 presents the conclusions of this study.

## 2 Experimental Model

Changes on traffic demands and link failures are dynamic conditions that undermine the operational performance of a network. Traffic demands undergo periodic changes during specific periods of time, such as night and day, which affect the congestion levels of the network. To address effective TE under those changes, network administrators could, eventually, perform alterations on the installed weights configuration to induce the redistribution of traffic. However, weight configuration changes cause a temporary instability on the traffic flows due to the distributed nature and convergence time of the routing protocol. Furthermore, changes on traffic paths disrupt the performance of higher level protocols, such as the Transport Control Protocol (TCP) whose connections may become degraded by out of order packet delivery.

There are also similar considerations to be made when re-configuring weights in response to link failures. The majority of these faults are single link failures, and last, usually, a relatively short amount of time [9]. Frequent link weights reconfigurations are thereby not considered a good approach to the problem. A more appealing solution consists in finding a single weights setting that would allow the network to maintain a good performance level against such events. In this case, the weights configuration to seek would guarantee a good traffic distribution in normal network conditions and continue to provide a good congestion level after a link failure or in case of foreseen changes of traffic demands. The next section presents an overview of the mathematical model used to support the simulations.

### 2.1 Mathematical Model

Network topologies are modelled as directed graphs $G(N, A)$, where $N$ represents a set of nodes, and $A$ a set of arcs, with capacity constrains $c_a$ for each $a \in A$. The amount of demand routed on the arc $a$, induced by a particular weight configuration, with source $s$ and destination $t$, is denoted by $f_a^{(s,t)}$. We define the utilization of an arc $a$ as $u_a = \frac{\ell_a}{c_a}$ where $\ell_a$ is the sum of all flows $f_a^{(s,t)}$ that travel over it. A well known piece-wise linear cost function $\Phi_a$, proposed by Fortz and Thorup [7], is used to heavily penalize over-utilized links. The derivative of $\Phi_a$ is defined as:

$$\Phi_a^{'} = \begin{cases} 1 & for \;\; 0 \leq u_a < 1/3 \\ 3 & for \;\; 1/3 \leq u_a < 2/3 \\ 10 & for \;\; 2/3 \leq u_a < 9/10 \\ 70 & for \;\; 9/10 \leq u_a < 1 \\ 500 & for \;\; 1 \leq u_a < 11/10 \\ 5000 & for \;\; u_a \geq 11/10 \end{cases} \tag{1}$$

The single optimization objective consists in distributing traffic demands in order to minimize the sum of all costs, as expressed in Equation 2.

$$\Phi = \sum_{a \in A} \Phi_a \tag{2}$$

A normalized congestion measure $\Phi^*$ is used to enable results comparison between distinct topologies and, for single objective EAs, to linearly combine the normal state congestion value of a network with the congestion after the occurrence of an event. It is important to note that when $\Phi^*$ equals 1, all loads are below 1/3 of the link capacity, while when all arcs are exactly full the value of $\Phi^*$ is 10 2/3. This value will be considered as a threshold that bounds the acceptable working region of the network.

It is now possible to define the general multi-objective optimization problem addressed in this work. Given a network represented by a graph $G = (N, A)$ and one or more demand matrices $D_i$, the aim is to find the set of weights ($w$) that simultaneously minimizes the objective functions $\Phi_1^*$ and $\Phi_2^*$, that, respectively, evaluate the congestion level of the network on a normal state and the congestion level after a change on the network operational conditions. For single objective optimization, the algorithms use a linear weighting scheme where the cost of the solution is given by:

$$f(w) = \alpha \times \Phi_1^* + (1 - \alpha) \times \Phi_2^*,\ \alpha \in [0; 1] \tag{3}$$

### 2.2 Experimental framework

The experimental simulations were run on a publicly available optimization framework, NetOpt [13], previously developed by the authors, in which the optimization meta-heuristic algorithms are provided by a Java-based library, JEColi [6]. An OSPF routing simulator is used to accommodate the traffic demands onto the networks topology arcs, and therefore enabling the application of the congestion evaluation function $\Phi^*$. An overall view of the framework architecture is shown in Figure 1 that also translates the general multi-objective optimization problem defined in the previous section.

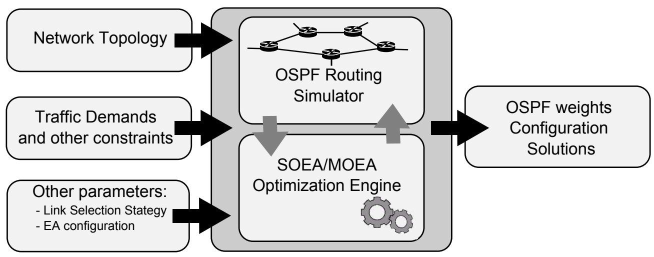


**Fig. 1.** General architecture of the optimization framework

The simulations were run for two synthetic topologies with 30 nodes, named $30_2$ and $30_4$ (the indexes 2 and 4 stand for the average in/out degree of each node), and a real-world backbone topology, the well known Abilene topology. The synthetic topologies were generated by the Brite topology generator [10], using the Barabasi-Albert model, with a heavy-tail distribution and an incremental grow type. The link capacities uniformly vary in the interval $[1; 10]$ Gbits. The characteristics of each topology are summarized in Table 1.

**Table 1.** Synthetic and realistic network topologies

| **Name** | **Topology** | **Nodes** | **Edges** |
|---|---|---|---|
| Abilene | backbone | 12 | 15 |
| $30_2$ | random | 30 | 55 |
| $30_4$ | random | 30 | 110 |

Traffic demand matrices provide, for every ingress point $a$ and every egress point $b$ in the network, the volume of traffic from $a$ to $b$ over a given time interval. For each topology, three distinct levels of traffic demand $D_i$, $i \in \{0.3, 0.4, 0.5\}$, were used in the experiments, where $i$ represents the expected mean of congestion in each link. Larger level values imply more difficult problems, as the volume of traffic to accommodate is greater. The set of demands $D_i$ for the Abilene network were obtained by scaling Netflow data [3] publicly available and measured on March 1st 2004 and September 1st 2004. The set of demand matrices for the synthetic topologies were randomly generate to fulfil the requirements of the expected mean of congestion.

### 2.3 EAs setup and evaluation metrics

Three different EAs were considered in this study, and applied to the contemplated optimization problems. The first approach was the use of a single objective EA (SOEA) based on previous work by the authors [14]. Alternative approaches were provided by two of the most popular multi-objective EAs, namely SPEA2 [16] and NSGA-II [4]. All algorithms were configured to use the same encoding and reproduction operators and their configurations was done to reduce any differences not related with the inherent differences of the optimization engines.

In all EAs, each individual encodes a solution as a vector of integer values, where each value (gene) corresponds to the weight of a link (arc) in the network, and therefore the size of the individual equals the number of links in the network. Although OSPF link weights are integers valued from 1 to 65535, only values in range $[1; 20]$ were considered, allowing to reduce the search space and, simultaneously, increasing the probability of finding equal cost multipaths (ECMP). ECMP offers substantial increases in bandwidth by load-balancing traffic over multiple paths.

The individuals that populate the initial populations were randomly generated, with arc weights taken from a uniform distribution within the reduced range. All EAs resort to the same reproduction operators for solutions combination and genetic diversity:

- *Random mutation*, replaces a given gene by a random value, within the allowed range.
- *Incremental/decremental mutation*, replaces a given gene by the next or by the previous integer value, with equal probabilities, within the allowed range.
- *Uniform crossover*, this operator works by taking two parents as input and generating two offspring. For each position in the genome, a binary variable

is randomly generated: if its value is 1, the first offspring takes the gene from the first parent in that position, while the second offspring takes the gene from the second parent; if the random value is 0, the roles of the parents are reversed.

The single objective EAs use a *roulette wheel scheme* in the selection procedure, by converting the fitness value into a linear ranking in the population. In the experiments, a population size of 100 was considered, and for the MOEAs, an archive of the same size was used. For the SOEA experiments, the final objective value is taken as a linear combination of the two objectives, weighted by a factor ($\alpha$) that defines the trade-off; three values were considered for $\alpha \in \{0.25, 0.5, 0.75\}$. Each simulation configuration was run 30 times with a stopping criteria of 1000 generations.

Three performance metrics that enable results comparison and the evaluation of the MOEA and SOEA algorithms performance were used in the experimental study:

- *C-measure*: It is based on the concept of solution dominance. Given two Pareto Fronts (PF1,PF2), the measure C(PF1; PF2) returns the fraction of solutions in PF2 that are dominated by at least one solution in PF1. A value of 1 indicates that all points in PF2 are dominated by points in PF1, so values near 1 clearly favour the method that generated PF1; values near 0 show that few solutions in PF2 are dominated by solutions in PF1.
- *Trade-off analysis* (TOA): For a pareto front PF1, and given a value of $\alpha$, the solution that minimizes $\alpha \times \Phi_1^* + (1-\alpha) \times \Phi_2^*$ is selected. Parameter $\alpha$ can take distinct values in the range $[0; 1]$, thus defining different trade-offs between the objectives. The values with the same $\alpha$ can be compared among the several multi objective optimizers (MOOs) and also with those from traditional algorithms.
- *Hypervolume*: It is the n-dimensional space that is contained by a set of points. It encapsulates in a single unary value a measure of the spread of the solutions along the Pareto front, as well as the closeness of the solutions to the Pareto-optimal front. We considered as an approximation for the Pareto-optimal front the non dominated solutions of all simulations in the same context, regardless of the algorithm.

The next two sections present more precised definitions of the two studied case problems where changes in the operational conditions of a network undermine its performance. In each case, the results produced by the three algorithms, SOEA, NSGA-II and SPEA2, are discussed and compared.

## 3 Optimization for Two Traffic Demands Matrices

### 3.1 Problem Definition

Traffic demands possess temporal properties that have a significant impact on internet traffic engineering. The diversity of services available on contemporary networks, as well as human behaviors and habits, provoke variations on traffic volumes and flow patterns not accommodated by traditional routing solutions. To acknowledge those variations, for example between two periods, such as night and day, we aim to find a link weight configuration that enables the network to sustain good functional performance in both periods. Thus, given two demand matrices, $D_1$ and $D_2$, that represent the traffic requirements of two distinct periods, we want to find a link weight configuration $w$ that simultaneously minimizes the congestion functions $\Phi_1^*$ and $\Phi_2^*$. Each $\Phi_i^*$ is the normalized cost function $\Phi$ (Equation 2) that evaluate the network congestion considering the traffic demands matrix $D_i$. The SOEA weighted-sum aggregation function for this set of experiments is defined in accordance with Equation 3. The main idea behind the optimization process is that, by compromising the congestion level in each individual scenario, it is possible to obtain a suitable configuration for both matrices. Under the SOEA algorithm, an administrator is able to fine tune adjustments, such as favouring one of the matrices and penalizing the other, by setting the $\alpha$ parameter accordingly in Equation 3. Under MOEA algorithms, the produced solutions feature distinct trade-offs between the objectives which enables network administrators to select the most appropriate solution.

### 3.2 Simulation Results

The experimental results, for each of the three algorithms (SOEA, NSGA-II and SPEA2), are summarized in Table 2 and Table 3 which respectively present the best and the mean fitness values of all runs with distinct trade-offs, organized by traffic demands levels and $\alpha$ values. In the experiments with the $30_4$ network topology only $D0.3$ level traffic demand matrices were considered as for higher levels of demands the obtained congestion values surpass the threshold of 10 2/3, above which the network ceases to operate acceptably. As the size and degree of each node increase, the difficulty of the optimization problem also increases. It is important to mention that, in all simulations, the linear correlation between the two considered traffic demands matrices, $D_1$ and $D_2$, for which the congestion is simultaneously optimized, is approximately 0.5.

The results for the Abilene topology show that all three algorithms were able to converge to the same best solution in at least one of the 30 simulations. The average fitness values, for all levels of demands and trade-offs, Table 3, are also very similar among the three algorithms. The performance metric C-measure, given in Table 4, where the overall mean value for all the distinct instances and runs was computed, reinforces the conclusion that all performances are akin with respect to the Abilene topology. The SOEA, NSGA-II and SPEA2 algorithms were able to provide equally good solutions as all values are of the same magnitude and, consequently, no algorithm's pareto fronts are considered to dominate the others.

For larger network topologies, the performance of the three algorithms starts to diverge. The results for the synthetic topology $30_2$, with 30 nodes and 55 edges, show that the NSGA-II algorithm is able to attain best fitness values for every $\alpha$ and demands level. This can be observed, for instance, with $D0.4$

**Table 2.** Best fitness comparison for two demand matrices optimization

| Algorithm | First Demands | Second Demands | Abilene | | | $30_2$ | | | $30_4$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 0.25 | 0.50 | 0.75 | 0.25 | 0.50 | 0.75 | 0.25 | 0.50 | 0.75 |
| SOEA | | | 1.139 | 1.161 | 1.179 | 1.374 | 1.386 | 1.398 | 2.503 | 2.487 | 2.472 |
| NSGA-II | 0.3 | 0.3 | 1.139 | 1.161 | 1.179 | 1.338 | 1.349 | 1.357 | 1.907 | 1.968 | 1.961 |
| SPEA2 | | | 1.139 | 1.161 | 1.179 | 1.461 | 1.452 | 1.442 | 4.336 | 5.620 | 6.178 |
| SOEA | | | 1.446 | 1.367 | 1.283 | 1.745 | 1.638 | 1.531 | - | - | - |
| NSGA-II | 0.3 | 0.4 | 1.446 | 1.367 | 1.283 | 1.659 | 1.559 | 1.453 | - | - | - |
| SPEA2 | | | 1.446 | 1.367 | 1.283 | 1.878 | 1.718 | 1.559 | - | - | - |
| SOEA | | | 1.522 | 1.522 | 1.521 | 1.951 | 1.985 | 2.019 | - | - | - |
| NSGA-II | 0.4 | 0.4 | 1.522 | 1.522 | 1.521 | 1.841 | 1.882 | 1.916 | - | - | - |
| SPEA2 | | | 1.522 | 1.522 | 1.521 | 2.139 | 2.184 | 2.214 | - | - | - |

**Table 3.** Mean fitness comparison for two demand matrices optimization

| Algorithm | First Demands | Second Demands | Abilene | | | $30_2$ | | | $30_4$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 0.25 | 0.50 | 0.75 | 0.25 | 0.50 | 0.75 | 0.25 | 0.50 | 0.75 |
| SOEA | | | 1.218 | 1.218 | 1.218 | 3.103 | 2.977 | 2.785 | 38.017 | 40.592 | 43.167 |
| NSGA-II | 0.3 | 0.3 | 1.212 | 1.213 | 1.213 | 1.925 | 1.826 | 1.728 | 4.306 | 4.440 | 4.575 |
| SPEA2 | | | 1.213 | 1.213 | 1.214 | 2.186 | 2.064 | 1.942 | 43.983 | 46.922 | 49.862 |
| SOEA | | | 1.489 | 1.399 | 1.308 | 7.946 | 6.115 | 4.285 | - | - | - |
| NSGA-II | 0.3 | 0.4 | 1.482 | 1.393 | 1.304 | 3.538 | 2.860 | 2.183 | - | - | - |
| SPEA2 | | | 1.482 | 1.393 | 1.304 | 4.551 | 3.622 | 2.693 | - | - | - |
| SOEA | | | 1.565 | 1.554 | 1.543 | 10.446 | 10.213 | 9.980 | - | - | - |
| NSGA-II | 0.4 | 0.4 | 1.559 | 1.549 | 1.540 | 2.919 | 2.801 | 2.684 | - | - | - |
| SPEA2 | | | 1.559 | 1.549 | 1.540 | 5.401 | 5.271 | 5.141 | - | - | - |

matrices and $\alpha = 0.5$, where the minimum and average fitness values are, respectively, 1.951 and 10.446 (SOEA), 1.841 and 2.919 (NSGA-II), 2.139 and 5.401 (SPEA2). Although NSGA-II and SOEA best values are very similar, and better than SPEA2 results, the average congestion values for NSGA-II are substantially smaller, that is, the NSGA-II solutions are globally better than those provided by the other two algorithms. The averaged C metric values for the $30_2$ network topology scenarios, Table 4, show that NSGA-II solutions dominate SPEA2 ones in more than 56% on average, with a reverse C-measure of almost 0%. When compared with SOEA solutions, NSGA-II solutions dominate approximately 16% of the SOEA ones and, for the reverse case, SOEA solutions dominate NSGA-II ones in only about 6%. It is therefore possible to conclude, that, for the $30_2$ topology experiment scenarios, the NSGA-II algorithm offers generally better solutions than any of the two other algorithms and that the SPEA2 algorithm had the worst performance of all.

As the size of the used topology increases, the performance of the NSGA-II algorithm detaches from the others. The experiments with the $30_4$ network topology show that while the best values of the three algorithm remain acceptable, NSGA-II features the best solutions, and is the only algorithm whose mean fitness values remain within the acceptable operating limits of the network (Table 3). The C metric values for this new set of simulations are very similar to

**Table 4.** Overall C-Measure for two traffic demand matrices optimization

| | **Abilene** | | | $30_2$ | | | $30_4$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | SOEA | NSGA-II | SPEA2 | SOEA | NSGA-II | SPEA2 | SOEA | NSGA-II | SPEA2 |
| SOEA | - | 0.143 | 0.179 | - | 0.062 | 0.553 | - | 0.029 | 0.602 |
| NSGA-II | 0.150 | - | 0.182 | 0.161 | - | 0.564 | 0.150 | - | 0.600 |
| SPEA2 | 0.110 | 0.113 | - | 0.001 | 0.004 | - | 0.000 | 0.001 | - |

those obtained for the $30_2$ topology, and again, NSGA-II solutions are globally better than those provided by SOEA and SPEA2.

**Table 5. Average hypervolume for two traffic demand matrices optimization**

| **Algorithm** | **Topology** | | |
|---|---|---|---|
| | **Abilene** | $30_2$ | $30_4$ |
| SOEA | 0.002 | 0.585 | 34.359 |
| NSGA-II | 0.002 | 0.321 | 4.868 |
| SPEA2 | 0.002 | 4.592 | 7950.838 |

It is possible to identify a consistency in the performance of all three algorithms where NSGA-II is the algorithm that show comparatively best results. The hypervolume indicators, presented in Table 5, also support that NSGA-II is the best choice algorithm in the context of weights setting optimization for two traffic demand matrices. The NSGA-II pareto fronts are closer to the Pareto-optimal approximation, and better spread, than those provided by SOEA and SPEA2.

## 4 Single Link Failure Optimization

### 4.1 Problem Definition

Link failures on network topologies can occur for different reasons. At the physical layer, a fiber cut or a failure of optical equipment may cause a loss of physical connectivity. Other failures may be related to hardware, such as linecard failures. Router processor overloads, software errors, protocol implementation and misconfiguration errors may also lead to loss of connectivity between routers. Failures may also vary in nature. They can be due to scheduled network maintenance or be unplanned. Although backbone networks are usually well planned and adequately provisioned, link failures may still occur and undermine their operational performance. Several mechanisms can be used to protect an IP network against link failures, such as overlay protection or MPLS fast re-route [2], but protecting all links remains a very difficult task, or even impossible, especially for large network topologies. Thus, protection against failure continues to be link based.

The NetOpt framework supports several criteria to select the failing link, some are dynamic, that depend on the solution that is being evaluated, while others are user choices. The framework also allows to select more than one link to fail simultaneously each corresponding to an optimization objective. This study only considers two of the available single link selection criteria:

- *Highest Load*: The selected link, for each solution being evaluated, is the one that has the highest load for the traffic demands given as parameter. Therefore, distinct solutions may have a different failing link.
- *User Selected*: A network administrator identifies the link against whose failure the network should be protected.

For a given network topology with $n$ links and a traffic demands matrix $D$, the aim is to find a set of weights $w$ that simultaneously minimize the function $\Phi^*_n$, representing the congestion cost of the network in the normal state, and $\Phi^*_{n-1}$, representing the congestion cost of the network when foreseeing that a selected link from the topology will fail. The SOEA weighted-sum aggregation model is described in Equation 4:

$$f(w) = \alpha \times \Phi^*_n + (1-\alpha) \times \Phi^*_{n-1},\ \alpha \in [0;1] \tag{4}$$

An administrator is able to define a trade-off between the objectives by tuning the value of the $\alpha$ weight. When $\alpha = 1$, the optimization is only performed for the normal state topology, without any link failure, whereas when using $\alpha = 0.5$ the same level of importance is given to the two topology states. However, as the link failure optimization can compromise the network congestion level in a normal state, a network administrator may wish to focus on the performance of the normal state network, e.g. using a $\alpha$ value between 0.5 and 1, at the expense of the congestion level in a failed state, that may not occur. Although this feature offers a good tuning tool for administrator, it requires several distinct runs in order to assert the best compromised solution. MOEA algorithms, on the other hand, are able to deliver such knowledge base and choice selection after a single run, and therefore, being more appealing in this context.

### 4.2 Highest Load Link Failure Optimization

The failure of the network link that carries the highest traffic load is one of the worst case scenarios for the failure of a single link in a network. Its failure would translate into the re-routing of the higher amount of traffic and potentially the worst case for out of order TCP packet delivery. Distinct levels of traffic demands, $D0.3$, $D0.4$ and $D0.5$ were used to compare the algorithms in problems with increasing difficulty. For comparison purpose, Table 6 that shows the obtained minimum weighted-sum aggregation fitness values, also includes the optimized congestion values for the networks without link failure optimization, and the respective congestion level after the failure of the link with higher load.

The simulation results show that, for the smallest topology, Abilene, all three algorithm behave alike producing equally good solutions. But, as the topology

**Table 6.** Best fitness values for single link failure weights setting optimization - Highest Load Link

| Topology | Demand | Without Link Failure Optimization | | With Link Failure Optimization | | | | | | Algorithm |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | $\alpha = 0.25$ | | $\alpha = 0.5$ | | $\alpha = 0.75$ | | |
| | | Before | After | Before | After | Before | After | Before | After | |
| Abilene | 0.3 | 1.20 | 1.76 | 1.29 | 1.23 | 1.29 | 1.23 | 1.23 | 1.33 | NSGA-II |
| | | | | 1.34 | 1.21 | 1.33 | 1.22 | 1.24 | 1.35 | SOEA |
| | | | | 1.29 | 1.23 | 1.29 | 1.23 | 1.23 | 1.34 | SPEA2 |
| | 0.4 | 1.53 | 32.22 | 1.63 | 1.58 | 1.63 | 1.58 | 1.55 | 1.70 | NSGA-II |
| | | | | 1.69 | 1.58 | 1.69 | 1.58 | 1.55 | 1.73 | SOEA |
| | | | | 1.64 | 1.58 | 1.64 | 1.58 | 1.55 | 1.70 | SPEA2 |
| | 0.5 | 1.91 | 309.48 | 2.14 | 1.91 | 2.14 | 1.91 | 2.05 | 2.17 | NSGA-II |
| | | | | 2.26 | 1.93 | 2.26 | 1.93 | 2.26 | 1.93 | SOEA |
| | | | | 2.14 | 1.91 | 2.14 | 1.91 | 2.04 | 2.14 | SPEA2 |
| $30_2$ | 0.3 | 1.49 | 14.20 | 1.55 | 1.42 | 1.44 | 1.48 | 1.44 | 1.48 | NSGA-II |
| | | | | 1.56 | 1.58 | 1.56 | 1.58 | 1.54 | 1.61 | SOEA |
| | | | | 1.57 | 1.50 | 1.56 | 1.51 | 1.49 | 1.64 | SPEA2 |
| | 0.4 | 1.79 | 41.44 | 1.83 | 1.76 | 1.75 | 1.80 | 1.75 | 1.80 | NSGA-II |
| | | | | 2.07 | 2.09 | 1.85 | 2.22 | 1.85 | 2.22 | SOEA |
| | | | | 1.95 | 1.93 | 1.91 | 1.96 | 1.91 | 1.96 | SPEA2 |
| | 0.5 | 5.49 | 180.94 | 4.99 | 3.70 | 4.99 | 3.70 | 4.11 | 5.31 | NSGA-II |
| | | | | 12.61 | 17.58 | 12.61 | 17.58 | 12.61 | 17.58 | SOEA |
| | | | | 8.23 | 8.41 | 8.15 | 8.48 | 7.86 | 9.03 | SPEA2 |
| $30_4$ | 0.3 | 3.67 | 73.69 | 2.38 | 2.20 | 2.30 | 2.25 | 2.10 | 2.59 | NSGA-II |
| | | | | 11.14 | 7.91 | 11.14 | 7.91 | 6.04 | 13.64 | SOEA |
| | | | | 59.48 | 29.64 | 28.95 | 47.39 | 28.95 | 47.39 | SPEA2 |
| | 0.4 | 33.93 | 223.04 | 18.66 | 10.13 | 18.66 | 10.13 | 10.07 | 28.42 | NSGA-II |
| | | | | 77.09 | 88.80 | 77.09 | 88.80 | 58.81 | 140.07 | SOEA |
| | | | | 355.03 | 139.57 | 205.65 | 190.92 | 159.03 | 325.12 | SPEA2 |
| | 0.5 | 126.90 | 158.44 | 157.19 | 95.15 | 97.19 | 132.37 | 97.19 | 132.37 | NSGA-II |
| | | | | 310.85 | 180.52 | 310.85 | 180.52 | 224.07 | 277.66 | SOEA |
| | | | | 490.70 | 466.66 | 490.70 | 466.66 | 467.31 | 504.91 | SPEA2 |

size increases, or with the escalation of traffic requirements, NSGA-II is able to obtain solutions which translate into lower congestion values before and after the link failure. In the $30_4$ network topology scenario, with $D0.3$ traffic demands and $\alpha = .5$, the fitness values before and after the link failure are, respectively, 2.19 and 2.29 for NSGA-II; 5.81 and 33.01 for SOEA; 204.29 and 219.19 for SPEA2. These results are even more relevant when comparing with the congestion values when only the congestion of the network in the normal state is optimized by resourcing to a single objective algorithm ($\alpha = 1$). The NSGA-II algorithm was able to provide a better solution while optimizing two objectives than a SOEA algorithm that optimizes a single objective, the congestion of the network before the link failure. This result is observed in all scenarios that are more demanding, allowing to conclude that NSGA-II performs better in these more difficult optimization tasks than SOEA even considering two objectives rather than a single one.

Although congestion values above 10 2/3 are not acceptable within an operational network, the results allow to observe that the more difficult the problem, the greater the difference between the quality of the solutions produced by each of the three EAs. NSGA-II is able to outperform SOEA and SPEA2 in all scenarios. The lack of performance of the SOEA algorithm in more demanding scenarios can be explained by its requirement of a higher number of generations to properly converge. It is also important to acknowledge that even small changes on a single weight can provoke drastic changes on shortest paths and therefore on the congestion value. The crowding distance used in the selection operator of NSGA-II, that keeps a diverse front by making sure each member stays a crowding distance apart, seems to positively influence the algorithm performance.

The C-measure values in Table 7 show that, despite being able to offer solutions with equivalent best fitness for the Abilene topology, the SO algorithm produces more solutions that are neither dominated by NSGA-II or SPEA2 solutions. In contrast, for the more demanding topologies, $30_2$ and $30_4$, NSGA-II solutions dominate approximately 14% of the SOEA solutions, when the reverse is 7% or less. When compared against SPEA2, both NSGA-II and SOEA present better values.

**Table 7.** C-measure of the highest Load link failure optimisation

| | **Abilene** | | | $30_2$ | | | $30_4$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | SOEA | NSGA-II | SPEA2 | SOEA | NSGA-II | SPEA2 | SOEA | NSGA-II | SPEA2 |
| SOEA | - | 0.173 | 0.242 | - | 0.071 | 0.702 | - | 0.056 | 0.887 |
| NSGA-II | 0.007 | - | 0.098 | 0.143 | - | 0.709 | 0.143 | - | 0.887 |
| SPEA2 | 0.010 | 0.108 | - | 0.003 | 0.005 | - | 0.000 | 0.000 | - |

### 4.3 User Choice Link Failure Optimization

A network administrator can consider that a particular link is more crucial than others, because of its capacity or for other reasons. It is therefore important to enable an administrator to select the link that needs to be protected against failure. For this set of simulations, the selected link in each topology is such that it occurs in the largest number of shortest paths when assigning to each link a weight inversely proportional to its capacity.

The minimal congestion values before and after the failure of the selected link, for distinct trade-offs ($\alpha = 0.25, 0.5, 0.75$), are presented in Table 8.

The results of this new test suite consolidate previous observations, that is, for simpler problems, with smaller topologies and lower traffic demand levels, the SOEA and MOEAs algorithms provide equally good solutions, but, as the number of nodes and links increases, or with the growth of traffic demands, NSGA-II is able to deliver better solutions, in the large majority of scenarios, both before and after the link failure. The C metric values, Table 9, are also similar to those observed for the higher load link failure optimization. In average and in the context of simpler problems, SOEA continues to have more solutions

**Table 8.** Best fitness values for single link failure weights setting optimization - User Select Link

| Topology | Demand | Without Link Failure Optimization | | With Link Failure Optimization | | | | | | Algorithm |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | $\alpha = 0.25$ | | $\alpha = 0.5$ | | $\alpha = 0.75$ | | |
| | | Before | After | Before | After | Before | After | Before | After | |
| Abilene | 0.3 | 1.20 | 1.76 | 1.23 | 1.73 | 1.20 | 1.74 | 1.20 | 1.74 | NSGA-II |
| | | | | 1.22 | 1.71 | 1.21 | 1.72 | 1.20 | 1.74 | SOEA |
| | | | | 1.24 | 1.72 | 1.20 | 1.74 | 1.20 | 1.74 | SPEA2 |
| | 0.4 | 1.53 | 25.57 | 1.58 | 33.44 | 1.58 | 33.44 | 1.53 | 33.52 | NSGA-II |
| | | | | 1.56 | 5.26 | 1.56 | 5.26 | 1.56 | 5.26 | SOEA |
| | | | | 1.58 | 33.44 | 1.58 | 33.44 | 1.53 | 33.52 | SPEA2 |
| | 0.5 | 1.91 | 309.48 | 1.97 | 119.30 | 1.95 | 119.32 | 1.93 | 119.34 | NSGA-II |
| | | | | 1.98 | 281.63 | 1.98 | 281.63 | 1.98 | 281.63 | SOEA |
| | | | | 2.01 | 119.29 | 1.95 | 119.32 | 1.93 | 119.34 | SPEA2 |
| $30_2$ | 0.3 | 1.49 | 8.17 | 1.40 | 1.50 | 1.40 | 1.50 | 1.40 | 1.50 | NSGA-II |
| | | | | 1.54 | 4.55 | 1.54 | 4.55 | 1.54 | 4.55 | SOEA |
| | | | | 1.61 | 1.74 | 1.60 | 1.74 | 1.60 | 1.74 | SPEA2 |
| | 0.4 | 1.79 | 58.65 | 1.76 | 1.93 | 1.76 | 1.93 | 1.75 | 1.93 | NSGA-II |
| | | | | 2.10 | 3.40 | 2.10 | 3.40 | 2.10 | 3.40 | SOEA |
| | | | | 2.18 | 2.50 | 2.18 | 2.50 | 2.18 | 2.50 | SPEA2 |
| | 0.5 | 5.49 | 193.16 | 5.79 | 41.19 | 5.44 | 41.51 | 5.44 | 41.51 | NSGA-II |
| | | | | 22.45 | 87.53 | 17.23 | 91.56 | 8.07 | 117.40 | SOEA |
| | | | | 28.13 | 47.86 | 12.86 | 55.66 | 12.30 | 56.43 | SPEA2 |
| $30_4$ | 0.3 | 3.67 | 117.13 | 2.20 | 2.29 | 2.19 | 2.29 | 2.18 | 2.33 | NSGA-II |
| | | | | 5.81 | 33.01 | 5.81 | 33.01 | 5.81 | 33.01 | SOEA |
| | | | | 204.29 | 219.19 | 204.29 | 219.19 | 204.29 | 219.19 | SPEA2 |
| | 0.4 | 33.93 | 98.98 | 10.43 | 9.67 | 9.90 | 10.05 | 9.85 | 10.11 | NSGA-II |
| | | | | 49.27 | 111.08 | 49.27 | 111.08 | 49.27 | 111.08 | SOEA |
| | | | | 456.36 | 509.50 | 456.36 | 509.50 | 440.71 | 544.54 | SPEA2 |
| | 0.5 | 126.90 | 421.07 | 89.91 | 88.98 | 62.21 | 112.74 | 62.21 | 112.74 | NSGA-II |
| | | | | 165.46 | 319.34 | 165.46 | 319.34 | 165.46 | 319.34 | SOEA |
| | | | | 557.49 | 600.71 | 557.49 | 600.71 | 557.49 | 600.71 | SPEA2 |

that are not dominated by any of the non-dominated sets of solutions resulting from NSGA-II and SPEA2 based optimizations. As the difficulty of the problem increases, NSGA-II stands out, providing better sets of non-dominated solutions.

**Table 9.** C-measure of User Choice link failure optimization

| | **Abilene** | | | $30_2$ | | | $30_4$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | SOEA | NSGA-II | SPEA2 | SOEA | NSGA-II | SPEA2 | SOEA | NSGA-II | SPEA2 |
| SOEA | - | 0.162 | 0.203 | - | 0.074 | 0.443 | - | 0.065 | 0.659 |
| NSGA-II | 0.008 | - | 0.063 | 0.122 | - | 0.468 | 0.163 | - | 0.657 |
| SPEA2 | 0.024 | 0.125 | - | 0.014 | 0.019 | - | 0.000 | 0.001 | - |

## 5 Conclusion

The simplicity of link-state protocols, and their reliability proven over the last two decades, continues to justify the use of such routing algorithms in the context of IP backbone networks. However, the dynamic conditions of IP networks, such as changes on traffic demands and disruptions on the underlying topology need to be addressed so that the network continues to ensure a good operational performance even if such events take place. An administrator could react to such changes by re-configuring the link weights but with a temporary negative impact on traffic flows. Other approaches, such as preventive optimization, can effectively take into consideration foreseen changes to compute weight configurations that allow the network to ensure a continues good levels of performance even in dynamic conditions. In this context, two multi objective problems were addressed, that consider changes on traffic demands and single link failure, resourcing to three popular EAs spanning both single and multi-objective: NSGA-II, SPEA2 and single objective EA using weighted-sum aggregation.

The results showed that for simpler problems the single objective optimization approaches provide solutions with best fitness values as good as the MOEA algorithms but, as the difficulty of the problems increases, for more complex network topologies and for more demanding traffic requirements, NSGA-II provides better solutions. By comparing the obtained results with previous work by the authors, it can be observed that SOEA algorithms require a greater number of generations for more demanding problems than MOEA algorithms. The two MOEAs, NSGA-II and SPEA2, rely heavily on their density estimator mechanisms, where the NSGA-II ability to provide a broader spread seems to influence more positively the optimization process than a better solution distribution attained by SPEA2.

Apart from the quality of the solutions other more practical aspects help determine the most appropriate algorithm to the problem. The single objective approaches have an important limitation. They assume, in each individual optimization process, that there is a single optimum trade-off between the objectives. A network administrator needs to guess which value of the weighting trade-off parameter better fits the needs of a network on the addressed operational conditions. In contrast, MOEA algorithms are able to calculate a set of solutions with distinct trade-offs between the two objectives, and let the network administrator decide which solution to implement. Moreover, NSGA-II is able to offer this broader set of solutions within a shorter time than the SOEA using weight-aggregation, or SPEA2 in the same conditions.

## References

1. Aysegl Altin, Bernard Fortz, Mikkel Thorup, and Hakan mit. Intra-domain traffic engineering with shortest path routing protocols. *Annals of Operations Research*, 204(1):56–95, 2013.
2. D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus. Requirements for Traffic Engineering Over MPLS. RFC 2702 (Informational), September 1999.

3. B. RFC 3954 - Cisco Systems NetFlow Services Export Version 9, October 2004.
4. Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evolutionary Computation*, 6(2):182–197, 2002.
5. E. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.
6. P. Evangelista, P. Maia, and M. Rocha. Implementing metaheuristic optimization algorithms with jecoli. In *Proceedings of the 2009 Ninth International Conference on Intelligent Systems Design and Applications*, ISDA '09, pages 505–510, Washington, DC, USA, 2009. IEEE Computer Society.
7. B. Fortz. Internet traffic engineering by optimizing ospf weights. In *Proceedings of IEEE INFOCOM*, pages 519–528, 2000.
8. B. Fortz and M. Thorup. Optimizing ospf/is-is weights in a changing world. *IEEE Journal on Selected Areas in Communications*, 20(4):756–767, 2002.
9. G. Iannaccone, C. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot. Analysis of link failures in an ip backbone. In *Proceedings of the 2Nd ACM SIGCOMM Workshop on Internet Measurment*, IMW '02, pages 237–242, New York, NY, USA, 2002. ACM.
10. A. Medina, A. Lakhina, I. Matta, and J. Byers. Brite: Universal topology generation from a users perspective. Technical report, Boston, MA, USA, 2001.
11. J. Moy. OSPF Version 2. RFC 2328 (Standard), April 1998. Updated by RFC 5709.
12. D. Oran. OSI IS-IS Intra-domain Routing Protocol. Technical report, IETF, February 1990.
13. V. Pereira, M. Rocha, P. Cortez, M. Rio, and P. Sousa. A framework for robust traffic engineering using evolutionary computation. In Guillaume Doyen, Martin Waldburger, Pavel eleda, Anna Sperotto, and Burkhard Stiller, editors, *Emerging Management Mechanisms for the Future Internet*, volume 7943 of *Lecture Notes in Computer Science*, pages 1–12. Springer Berlin Heidelberg, 2013.
14. M. Rocha, P. Sousa, P. Cortez, and M. Rio. Quality of Service Constrained Routing Optimization Using Evolutionary Computation. *Applied Soft Computing*, 11(1):356–364, 2011.
15. K. C. Tan, T. H. Lee, and E. F. Khor. Evolutionary algorithms for multi-objective optimization: Performance assessments and comparisons. *Artif. Intell. Rev.*, 17(4):251–290, June 2002.
16. E. Zitzler, M. Laumanns, and L. Thiele. Spea2: Improving the strength pareto evolutionary algorithm. Technical report, 2001.