



A comparison of multi-objective optimization algorithms for weight setting problems in traffic engineering

Vítor Pereira¹ · Pedro Sousa² · Miguel Rocha¹

Accepted: 3 September 2020 / Published online: 23 October 2020
© Springer Nature B.V. 2020

Abstract

Traffic engineering approaches are increasingly important in network management to allow an optimized configuration and resource allocation. In link-state routing, setting appropriate weights to the links is an important and challenging optimization task. Different approaches have been put forward towards this aim, including evolutionary algorithms (EAs). This work addresses the evaluation of 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 network, with (1) changes in the traffic demand matrices, and (2) link failures. The methods will simultaneously optimize for both conditions, the normal and the altered one, following a preventive TE approach. Since this leads to a bi-objective function, the use of multi-objective EAs, such as SPEA2 and NSGA-II, came naturally; those are compared to a single-objective EA previously proposed by the authors. The results show a remarkable performance and scalability of NSGA-II in the proposed tasks presenting itself as the most promising option for TE.

Keywords Traffic engineering · Intra-domain routing · Link-state routing protocols · Multiobjective evolutionary algorithms · NSGA-II

1 Introduction

Congestion avoidance in whole or part of an Internet Protocol (IP) network is one of the most important problems for Internet Traffic Engineering (TE). Distinct solutions that target optimal traffic congestion levels on a network have emerged in the networking research community, resorting to diversified strategies. Some of those proposals are reactive, and try to avoid congestion by adapting traffic flows at the edge of the network (Villamizar and Song 1994; Floyd and Fall 1999). Other solutions, enabled by new trends like Software Defined

Networking (SDN), maximize the network utilization with hybrid deployments, where some flows are directed according to routing protocol decisions, while others are forwarded reflecting specific administratively installed rules (Jain et al. 2013).

There are also preemptive approaches that consider known or estimated traffic requirements, and try to avoid congestion by optimizing traffic distribution on the available resources (Fortz 2000). Regardless of the congestion avoidance strategy, at the core of the problem lies the necessity to improve resources management and, in this context, routing protocols play a crucial role as they are responsible for global traffic forwarding decisions.

Link-state protocols, such as Open Shortest Path First (OSPF) (Moy 1998) and Intermediate System to Intermediate System (IS-IS) (Gredler and Goralski 2005), are built around a well-known algorithm from graph theory, Dijkstra's shortest path algorithm (Dijkstra 1959). After building a topological database, also called the link-state database, each router calculates the shortest path to each network node, which minimizes the sum of all weights assigned to each link.

✉ Vítor Pereira
vpereira@ceb.uminho.pt

Pedro Sousa
pns@di.uminho.pt

Miguel Rocha
mrocha@di.uminho.pt

¹ Department of Informatics, Centre of Biological Engineering, University of Minho, Braga, Portugal

² Department of Informatics, Centro Algoritmi, University of Minho, Braga, Portugal

When more than a shortest path exists to the same destination, some OSPF and IS-IS implementations perform Equal-Cost Multi-Path (ECMP) (Hopps 2000), splitting traffic evenly along the multiple paths with equal cost weight. This allows a better distribution of traffic on the available resources, and provides substantially better throughput and reliability.

In a link-state context, to optimize congestion is to carefully choose link weights. However, this decision making process, usually performed by a network administrator, is not an easy task when the scale of the network and heavy traffic flow is taken into consideration. If inadequate, a weights' configuration can cause a misallocation of traffic into the available resources, resulting in packet loss, increased delays, and, potentially, in the unfulfillment of service level agreements (SLAs).

The shortest path weight setting optimization problem for congestion avoidance, a multi-commodity flow problem in TE, consists in finding, for known traffic demands, a set of weights that optimizes the congestion levels of the network, and enables traffic to be forwarded without exceeding the capacity of any link. This NP-hard problem has been covered in previous efforts, resorting to different optimization meta-heuristics, such as, for example, Local Search (Fortz 2000), Evolutionary Algorithms (Ericsson et al. 2002; Pereira et al. 2013), or Simulated Annealing (Ben-Ameur et al. 2000). Non-ECMP variations of the problem have also been considered, such as, single shortest paths optimization (Bley et al. 2010), where demands between each origin-destination, within an autonomous system (AS), are routed via a uniquely determined shortest path. More recently, the utilization of non-shortest paths with unequal traffic splitting have also been scrutinized to optimize the utilization of networks' available resources (Pereira et al. 2016). An early survey of the weight setting problem, which also explore some related extensions, can be found in Altin et al. (2013).

Although most approaches only address static conditions on IP networks, in real world scenarios, those conditions are mutable. Traffic requirements change over time, for example, traffic demands undergo periodic alterations due to night and day requirements. Disruptions on the underlying network infrastructure may also occur, such as a link or a switch failure, which also affects how traffic is forwarded on the remaining available resources. In these contexts, the weight setting problem may need to consider multiple conflicting goals, which give rise to a set of trade-off solutions. In fact, it is possible to consider those changes and include them in the link weights optimization process, by formulating adequate multi-objective optimization problems. Shortest path weight setting optimization approaches, that cover such changes, have been studied considering distinct scenarios and contexts. In Altin et al. (2012) and Pereira et al. (2013), the authors aim to find a single link weights configuration able

to optimize congestion over a set of traffic demand matrices, and thus enabling the network congestion to remain at a functional level, even when changes on traffic patterns and volume occur. The problem has also been explored in contexts where the possibility of link failures is equated (Fortz and Thorup 2003; Pereira et al. 2013), or when quality of service (QoS) restrictions, such as delay requirements, are imposed (Rocha et al. 2011).

Multi-objective evolutionary algorithms (MOEAs) have been proven to deliver near optimal solutions for the weights setting problem when different, and sometimes conflicting, goals are targeted (Rocha et al. 2011). They, also, offer several advantages when compared with other optimization techniques. Their ability to provide a set of possible solutions, with distinct trade-offs between objectives, in a single run, enables network administrators to choose a configuration from a broader set, and consequently, enable a conscious choice of the most adequate solution.

However, distinct MOEAs have different merits and demerits (Tan et al. 2002) and, consequently, may not offer equally good solutions for distinct problems. The *no free lunch theorem* for optimization (Wolpert and Macready 1997) clearly states that there is no single best algorithm for all optimization problems. Hence, algorithm selection and settings might involve trial and error.

In this context, the present work presents a comparative study of three popular EAs: the Non-dominated Sorting Genetic Algorithm (NSGA-II), the Strength Pareto Evolutionary Algorithm (SPEA2) and a Single-Objective Evolutionary Algorithm (SOEA) using weighted-sum aggregation, previously proposed by the authors (Rocha et al. 2011). The experimental study is applied to two variants of the described problem, that takes into account the variability of networks' operational conditions, namely, changes on traffic demands and single link failures. The present article extends the work presented in Pereira et al. (2015), including novel results and a more extensive analysis, exploring additional factors that influence the optimization process, such as the dimension of the search space imposed by the considered weight range, as well as the computational efforts required by the optimization methods. A more meticulous description of the conditions, mathematical models and algorithms used in the experiments is presented, as well as a more comprehensive contextualization of this study, reviewing related work.

The remainder of this study proceeds as follows. Section 2 presents the problem statement including both single and multi-objective formulations; Sect. 3 presents an introduction to Evolutionary Multi-Objective Algorithms and describes the ones proposed in the context of this work; Sect. 4 provides some details on the experiments conducted and their setup; Sect. 5 presents and discusses the obtained experimental results; finally, Sect. 6 presents the conclusions.

2 Problem statement and definitions

Changes on network conditions can effectively be modelled to be handled by optimization algorithms, as the ones proposed in this work. In each scenario, two or more objectives can be simultaneously optimized, where one reflects the conditions of a network on a normal state, with known demands and a fixed topology state, while the others reflect some sort of disruption condition. A simultaneous optimization of link-state weights for the different conditions can provide a network with resilience against changes. Furthermore, the introduction of tolerance to changes in the configurations should have an acceptable impact on the performance of the network on a normal state when compared with a single objective optimization.

This section provides the mathematical model used as a foundation for the proposed algorithms and, based on this model, provides formulations, both single and multi-objective, for the problems to be addressed in this work, in the context of TE.

2.1 Mathematical model

Network topologies are modelled as directed graphs $G(N, A)$, where N represents the set of nodes, and A the set of arcs. Each arc has a capacity constrain c_a that limits the amount of traffic it can carry. We denote as $f_a^{(s,t)}$ the amount of traffic, with source and destination pair nodes (s, t) , routed on arc a . The utilization of an arc a can thus be expressed as the fraction $u_a = \frac{\ell_a}{c_a}$, where ℓ_a is the sum of all $f_a^{(s,t)}$, $(s, t) \in N \times N$, that travel over it.

Considering given traffic necessities, and an installed routing configuration, it is possible to evaluate the global congestion level on the network by associating a cost to each arc according to its utilization. A well known piecewise linear cost function Φ_a , proposed by Fortz and Thorup (2000), is used for that purpose. The derivative of Φ_a is defined as:

$$\Phi'_a = \begin{cases} 1 & \text{for } 0 \leq u_a < 1/3 \\ 3 & \text{for } 1/3 \leq u_a < 2/3 \\ 10 & \text{for } 2/3 \leq u_a < 9/10 \\ 70 & \text{for } 9/10 \leq u_a < 1 \\ 500 & \text{for } 1 \leq u_a < 11/10 \\ 5000 & \text{for } u_a \geq 11/10 \end{cases} \quad (1)$$

The sum of all costs, expressed in Eq. 2, enables to evaluate the global congestion level on the network. As Φ_a is heavily penalizing on links with a high or over utilization, lesser congestion measure translate into a better distribution of traffic. The optimization objective is therefore to minimize the congestion measure Φ by finding a routing configuration which better distributes a given traffic matrix, this without

exceeding the capacity of the infrastructure links.

$$\Phi = \sum_{a \in A} \Phi_a \quad (2)$$

$$\Phi^* = \frac{\Phi}{\sum_{(s,t) \in N \times N} d_{st} \times h_{st}} \quad (3)$$

To enable results comparison among distinct topologies, a normalized congestion measure Φ^* is used, expressed in Eq. 3, where d_{st} and h_{st} are respectively the euclidean distance and minimum hop count between nodes s and t . It is important to note that when Φ^* equals 1, all link loads are below 1/3 of their capacity, while when all arcs are exactly full, the value of Φ^* is 10 2/3. This value is to be considered as a threshold that bounds the acceptable working region of the network.

2.2 Multi-objective generic problem

Before we discuss particular problems and algorithms for multi-objective optimization, we define a generic problem that involves multiple conflicting objectives. A multi-objective optimization problem (MOP) involves a number of objective functions that are to be either minimized or maximized, subject to a number of constraints and variable bounds. A MOP, that involves M optimization objectives, can, thus, be formalized as:

$$\begin{aligned} &\text{Minimize } \mathbf{F}(\mathbf{x}) = [F_1(\mathbf{x}), F_2(\mathbf{x}), \dots, F_M(\mathbf{x})]^T; \\ &\text{subject to } g_j(\mathbf{x}) \leq 0, \quad j = 1, \dots, J; \\ &\quad \quad \quad x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, \dots, n. \end{aligned} \quad (4)$$

where a feasible solution $\mathbf{x} \in A^n$ is a vector of n decision variables, $\mathbf{x} = (x_1, x_2, \dots, x_n)$. The feasible set of solutions in the decision space is defined by the constrains $g_i(x) \leq 0$, and by the variable bounds. The multi-objective function vector $\mathbf{F} \in B^M$, where $f_m : A^n \rightarrow B$, constitutes a feasible set in the objective space.

2.3 Multi-objective weight setting problems

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, that impact the congestion levels of the network.

To address those changes, network administrators can perform alterations to the installed weights to redefine the distribution of traffic. However, such changes can lead to possible complications. Weight changes create a temporary instability on the traffic flow due to the distributed nature and convergence time of commonly used routing protocols. Furthermore, changes on traffic paths disrupt the performance

Table 1 Optimization scenarios

Scenario	General description
Resilience to traffic variations	An administrator aims to attain routing configurations able to assure resilience levels to variations in the traffic volumes traversing the network infrastructure. The suggested configurations should be able to support a heterogenous set of traffic volumes that may vary during a given time period (Sect. 2.3.1)
Resilience to link failures	The administrator is interested in improving the resilience levels when dealing with failures of specific links of the network infrastructure. Thus, the objective is that even in the presence of such failure events, the network infrastructure still presents acceptable performance with minimal traffic loss (Sect. 2.3.2)

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 considerations to be made when reconfiguring weights in response to link failures. The majority of link failures are single link failures and usually last for a short amount of time (Iannaccone et al. 2002). Frequent reconfiguration of link weights is thereby not considered a good approach to the problem.

A more appealing solution to these problems consists in finding a single weight setting that would allow the network to maintain a good performance level against such events, that is, find a weight configuration that continues to provide a good distribution of traffic after a link failure or in case of foreseen changes of traffic demands. Table 1 summarizes the two scenarios that are further detailed in the following sub-sections.

2.3.1 Two traffic demands congestion optimization

Traffic demands have 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, lead to variations on traffic volumes and flow patterns not accommodated by tradi-

tional routing solutions. To acknowledge those variations, for example between two periods, such as night and day periods, we aim to find link weights that enable the network to sustain good functional performance in both periods.

Thus, given two traffic demands matrices, D_1 and D_2 , that represent traffic necessities of two distinct periods, we aim to find a set of link weights w that simultaneously minimizes the objective functions Φ_1^* and Φ_2^* , thus defining a bi-objective MOP. Each Φ_i^* is the normalized cost function Φ^* (Eq. 3) that considers the traffic demands matrix D_i .

Additionally, a single-objective problem can also be defined using a weighted-sum aggregation of the different objectives as expressed in Eq. 5:

$$f(w) = \alpha \times \Phi_1^* + (1 - \alpha) \times \Phi_2^*, \alpha \in [0; 1] \quad (5)$$

It is possible to obtain a suitable configuration for both matrices, by compromising the congestion level in each individual scenario. Under Eq. 5, an administrator is able to fine tune adjustments, such as favouring one of the matrices and penalizing the other, by setting the α parameter accordingly.

2.3.2 Single link failure congestion optimization

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 (Cisco Customer Case 2013; Suchara et al. 2011).

Failures may also vary in nature. They can be due to scheduled network maintenance or 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 Multi Protocol Label Switching (MPLS) fast re-route (Awduche et al. 1999), 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.

As it is not possible to create scenarios where the optimization is done to address the failure of all links simultaneously, we assumed in this work that optimization will be done for a scenario with no failures and for an alternative scenario where a selected link fails, thus defining two objective functions.

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 minimizes the function Φ_n^* , representing the congestion cost of the network in the normal state, and

Φ_{n-1}^* , representing the congestion cost of the network when a selected link has failed.

The single-objective version, considering a weighted-sum aggregation model, is provided by Eq. 6:

$$f(w) = \alpha \times \Phi_n^* + (1 - \alpha) \times \Phi_{n-1}^*, \quad \alpha \in [0; 1] \quad (6)$$

An administrator is able to define a trade-off between the objectives by tuning the value of the α 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 α value between 0.5 and 1, at the expense of the congestion level in a failed state, that may not occur.

Several criteria can be used to select the failing link, some are dynamic, that depend on the solution that is being evaluated, while others are user predefined choices. In this study, we only considered two of the possible single link selection criteria:

- *Highest load* The selected link, for each solution being evaluated, is the one that has the highest load when the traffic demands given as parameter are allocated. Therefore, distinct solutions may have a different failing link.
- *Centrality* 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 (a commonly used weight setting strategy used, for instance, in OSPF Cisco implementations).

3 Evolutionary algorithms for weight setting problems

3.1 Single and multi-objective evolutionary algorithms

Evolutionary algorithms (EA) (Bäck 1996) are a particular class of evolutionary computation methods mainly used for optimization problems. They are population based, that is, they maintain a population of search points, rather than a single point, and the evolution of the system involves comparisons and interaction between the points.

In combinatorial optimization, in problems where the search space is typically very large, EAs try to find an optimal solution from a finite set of solutions. Members of the search space are evaluated via a fitness function, which determines how well they solve the particular problem instance.

This is an analogy with natural selection, in which the fittest individuals survive and evolve to produce more adapted solutions. By focusing on the best members of the population, and introducing small variations (mutation) and mating (crossover) operations, it is expected that the population evolves toward good, or even optimal, solutions within a reasonable time.

Suggested in the late 1980s, Multi-objective Evolutionary Algorithms (MOEAs) (Schaffer 1985) are now commonly used to solve optimization problems that need to simultaneously consider multiple objectives. These objectives are frequently conflicting, that is, a solution that improves one of the objectives will eventually degrade at least one of the others. Consequently, there is no single global solution, and it is necessary to determine a set of optimal points, named as a Pareto Front (PF), populated by non-dominated solutions.

Typically, MOEAs use the concept of dominance in the fitness assignment. This idea, initially introduced into EAs by Goldberg (1989), has the main advantage of not requiring the transformation of the multi-objective problem into a single objective one. Furthermore, they are able to generate a diverse set of Pareto optimal solutions in a single run. A solution is said to dominate another if it is not worse than the other in all objectives and, simultaneously, it is strictly better than the other in at least one objective.

There are two important goals, possibly conflicting, in dominance-based approaches, convergence and diversity. Convergence refers to the ability to find a (finite) set of solutions lying on the Pareto-optimal front, while diversity refers to the heterogeneity of such solutions, which should cover the entire range of the Pareto-optimal front. While the distance to the optimal front is to be minimized, the diversity of the generated solutions is to be maximized. Different MOEAs however, provide non-dominated solutions with distinct convergence and diversity for distinct MOPs.

To improve MOEAs diversity and convergence, elitism can also be used. This concept, very popular since the 1990s, consists in maintaining an external set, called archive, that allows to store all the non-dominated or the most preferred solutions found during the search. This archive mainly aims at preventing these solutions from being lost during the optimization process.

Since Schaffer (1984) introduced the Vector Evaluated Genetic Algorithm (VEGA) to solve multi-objective problems, several other MOEAs emerged. Some representative examples include algorithms as the PAES by Knowles and Corne (2000), SPEA (Zitzler and Thiele 1998) and SPEA2 by Zitzler et al. (2001), NSGA (Srinivas and Deb 1994) and NSGA-II by Deb et al. (2002), MOPSO by Coello et al. (2004), PESA-II by Corne (2001), among many more.

Testing all available MOEAs is well beyond the scope of this study. After considering the available possibilities, taking also into account software availability, we selected

two commonly used MOEAs, whose performance has been recognized by the community, the Non-dominated Sorting Genetic Algorithm (NSGA-II), the Strength Pareto Evolutionary Algorithm (SPEA2) and a Single-Objective Evolutionary Algorithm (SOEA) using weighted-sum aggregation. We next provide a brief description of each of these algorithms. These will be put forward in the context of the weight setting problems addressed in this work over the next sections.

3.2 Single-objective evolutionary algorithm for weight setting problems

The SOEA proposed in this work is based on previous work by the authors in the context of the optimization of intra-domain routing weights (Rocha et al. 2011; Pereira et al. 2013). In this SOEA, 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 65,535, we only considered here values in range [1 : 20].

The individuals in the initial population are randomly generated, with the arc weights taken from a uniform distribution. Two mutation and two crossover operators are used to create new solutions:

- *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 inputs 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.
- *Two point crossover* two crossover points are chosen and the genes in positions between these points are exchanged between two mated parents.

A *roulette wheel scheme* (Razali and Geraghty 2011) is used in the selection procedure, after previously converting the fitness values into a linear ranking in the population. Individuals are selected with a probability that is directly proportional to their ranking value, which corresponds to a portion of the roulette wheel.

For the SOEA, the different objective functions were aggregated using weighting coefficients, as explained in the previous section for the different tasks. The multi-objective

optimization problem is, thus, transformed into a scalar optimization problem, that can be addressed resorting to a SOEA. The aggregation function is expressed by:

$$\text{Minimize } \mathbf{F}(\mathbf{x}) = \sum_{i=1}^M w_i F_i(\mathbf{x}) \quad (7)$$

$$\sum_{i=1}^M w_i = 1 \quad (8)$$

where $w_i \geq 0$ are weighting coefficients and F_i are the different objective functions in the MOP. When the aggregated objective functions are normalized, such as the case in the present study, the w_i coefficients translate the relative importance given to each objective.

3.3 Non-dominated sorting genetic algorithm-II

The NSGA-II algorithm attempts to find multiple Pareto optimal solutions in a multi-objective optimization problem, while attending to the three main ideas: (1) It uses an elitist principle; (2) It uses an explicit diversity preserving mechanism; (3) It emphasizes non-dominated solutions. In most aspects, this algorithm does not have much similarity with the original NSGA, but the authors kept the name NSGA-II to highlight its genesis and place of origin (Deb 2001).

At any generation, the offspring population is firstly created by using genetic operators applied over the parent population. In this case, the reproduction operators will be the same defined above for the SOEA. The two solution sets are then combined to form a new population of size $2N$, from which an N dimension population is formed through selection based on dominance. A pseudo-code for NSGA-II is presented in Algorithm 1.

The new population is filled with points of different non-dominated fronts, one at a time. The filling starts with the first non-dominated front (of class 1) and continues with points of the second non-dominated front, and so on, as shown in Fig. 1a. When the last allowed front is reached, and if not all members can be included in the new population, the points with highest crowding distance are chosen. The crowding distance d_i of point i is a measure of the objective space around i which is not occupied by any other solution in the population (Fig. 1d). This, therefore, enables a greater solution diversity.

3.4 Strength Pareto evolutionary Algorithm 2

While NSGA-II uses a dominance depth sorting of solutions to fill the new population, SPEA2 resorts to dominance count and dominance rank sorting strategies on the partially ordered solution space. In a dominance rank strategy, the rank associ-

Algorithm 1 NSGA-II Algorithm

```

1: procedure NSGA-II( $N, f_k(x_k)$ )
2:   Initialize Population  $P'$ 
3:   Evaluate the population against the objectives
4:   Assign Rank based on Pareto Dominance (sort)
5:   Generate Child Population
6:     Parents Binary Tournament Selection
7:     Recombination and Mutation
8:   while not stopping condition do
9:     for each Parent and Child in Population do
10:      Assign Rank based on Pareto (sort)
11:      Generate sets of non-dominated vectors along  $PF_{known}$ 
12:      Loop (inside) by adding solutions to next generation starting from the first front until  $N$  individuals found determine crowding distance between points on each front
13:    end for
14:    Select points (elitist) on the lower front (with lower rank) and are outside a crowding distance
15:    Create next generation
16:    Parent Binary Tournament Selection
17:    Recombination and Mutation
18:  end while
19: end procedure

```

ness assignment scheme, and incorporates a nearest neighbor density estimation and a new archive truncation method.

Algorithm 2 SPEA2 Algorithm

```

1: procedure SPEA2( $N, E, f_k(x_k)$ )
2:   Initialize Population  $P'$  of size  $N$ 
3:   Evaluate objective values
4:   Create an empty Archive  $A$  with capacity  $E$ 
5:   while not stopping condition do
6:     Evaluate the population  $P'$  against the objectives
7:     for each solution in the Population and Archive do
8:       Calculate Raw Fitness
9:       Calculate Solution Density
10:      Compute Solution Fitness based on Raw Fitness and Density values
11:    end for
12:    Copy all non-dominated solutions from  $P'$  and  $A$  to the  $A$ 
13:    if the capacity of  $A$  has not been exceeded then
14:      use dominated individuals in  $P'$  to fill  $A$ 
15:    else
16:      use the truncation operator to remove elements from  $A$ .
17:    end if
18:    Create next generation
19:    Parent Binary Tournament Selection
20:    Recombination and Mutation
21:  end while
22: end procedure

```

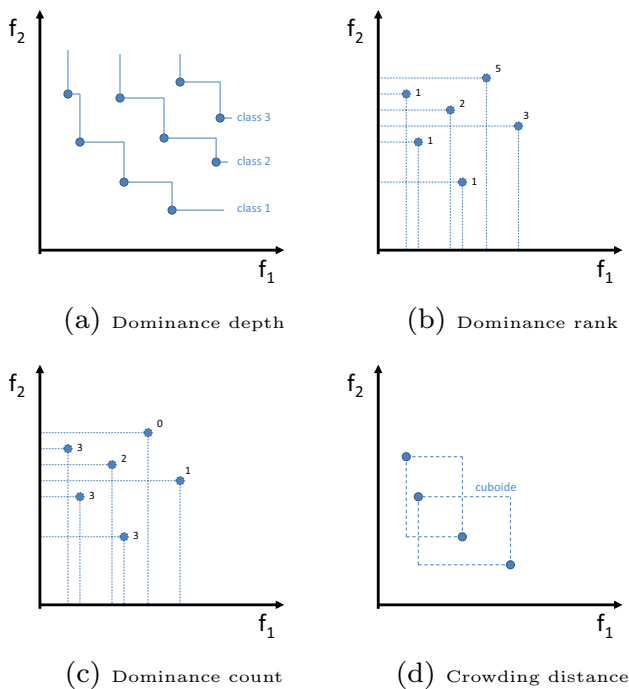


Fig. 1 Dominance strategies and crowding distance

ated with a solution is related to the number of solutions in the population that dominate the considered solution (Fig. 1b), whereas the dominance count of a solution is related to the number of solutions dominated by it (Fig. 1c). Both strategies are used to establish an order between the solutions. The single value fitness rank assigned to each solution evaluates the quality of a solution in relation to the whole population. The SPEA2 algorithm, Algorithm 2, when compared with SPEA, introduces improvements to the dominance fit-

4 Experimental setup

4.1 Algorithm implementation and configuration

A publicly available optimization framework, NetOpt (Pereira et al. 2013), developed by the authors, was used to run the simulations. The framework aims to provide tools to optimize network operational performance by resorting to evolutionary computation methods. The optimization meta-heuristic algorithms are provided by a Java-based library, JEColi (Evangelista et al. 2009) with some modifications and extensions to fulfill the framework requirements. The general architecture of the framework is presented in Fig. 2. In addition to the meta-heuristics library that provides the Genetic, NSGA-II and SPEA2 implementations, NetOpt also includes an OSPF routing simulator. Given a topology, a weight configuration (solution) and one or more traffic matrix, the OSPF simulator computes the routes between all nodes, using the Dijkstra shortest path algorithm, and accommodates accordingly the traffic demands onto the topology arcs. The solution can then be evaluated by applying the congestion function cost from Eq. 3. The process is repeated for each solution.

The experiments were run with the same global configurations so that the results could directly be compared. The population size was set to 100 individuals and, when applicable, with an archive of the same size. The crossover and

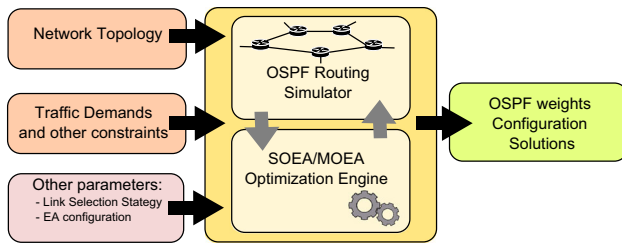


Fig. 2 General architecture of the optimization framework

Table 2 Synthetic and realistic network topologies

Name	Topology	Nodes	Edges
Abilene	Backbone	12	15
30 ₂	Random	30	55
30 ₄	Random	30	110

mutation operators are the ones described in Sect. 3.2, all with a 25% probability of being applied. Each simulation configuration was run 30 times with a stopping criterion of 1000 generations. For the SOEA experiments, trade-offs values α in $\{0.25, 0.5, 0.75\}$ were considered between the optimization objectives.

4.2 Simulation case studies

The simulations were run for two synthetic topologies, denoted as 30₂ and 30₄ (30 identifies the number of nodes, while the indexes 2 and 4 stand for the average in/out degree of each node), as well as for a real backbone topology, the Abilene network topology (Xu Dahai et al. 2011). The synthetic topologies were generated by the Brite topology generator (Medina et al. 2001), 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 2.

The global amounts of traffic that are expected to traverse a network infrastructure are generally expressed as traffic demands. ISPs often use matrices to model such values that can be obtained by several techniques (Technologies 2012; Tune and Roughan 2014). The demand matrix usually summarizes, for each source/destination edge router pair, a given required bandwidth to be supported by the network domain. In this context, when using synthetic networks, the framework is also able to tune the difficulty levels of the optimization problem, by considering distinct levels of traffic demands. The set of demand matrices for the synthetic topologies were randomly generate to fulfill the requirements of the expected mean of congestion. For each network, a set of three distinct traffic demand matrices $D_i, i \in \{0.3, 0.4, 0.5\}$, were used where i represents the expected mean of conges-

tion in each link. Thus, larger values imply more difficult problems, as more traffic needs to be accommodated on the available resources.

In more detail, for each pair of nodes (s, t) , $s \neq t$, the amount of traffic from s to t is modelled by Eqs. 9 and 10, where R is a random number in range $[0, 1]$, $d_{s,t}$ is the Euclidean distance between both nodes, \bar{c}_a is the average capacity of all links, $|E|$ is the number of links in the topology and $H_{s,t}$ the minimum number of hops between s and t . The use of the Euclidean distance in the formulation $D(s, t)$ implies that close pairs of nodes share more traffic.

$$D(s, t) = \frac{R \times \delta}{d_{s,t}} \quad (9)$$

$$\delta = 2 \times \alpha \times \bar{c}_a \times |E| \times \sum_{(s,t) \in N^2} \frac{H_{s,t}}{d_{s,t}} \quad (10)$$

The set of demands D_i for the Abilene network were obtained by scaling Netflow data (RFC 3954 2004) publicly available and measured on March 1st 2004 and September 1st 2004. In problems that have more than one traffic matrix as input, the linear correlation between them is approximately 0.5.

4.3 Comparison metrics

To compare the obtained results and evaluate the performance of the MOEAs and SOEA, in the context of the performed simulations, three metrics were used:

- *C-measure* It is based on the concept of solution dominance. Given two Pareto Fronts (PF1, PF2), the measure $C(\text{PF1}; \text{PF2})$ 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, while values near 0 imply that few solutions in PF2 are dominated by solutions in PF1, and thus that the algorithm that generated PF1 performs worst.
- *Trade-off analysis (TOA)* For a pareto front PF1, and given a value of β , the solution that minimizes $\beta \times \Phi_1^* + (1 - \beta) \times \Phi_2^*$ is selected. Parameter β can take distinct values in the range $[0; 1]$, thus defining different trade-offs between the objectives. The values with the same β can be compared among the several multi-objective optimizers and also with those from traditional single-objective algorithms.
- *Hypervolume* It is the n-dimensional space that is contained by a set of points. It encapsulates in a single 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.

Considering that, for each problem, the Pareto-optimal front is not known, we considered as an approximation the set of the non-dominated solutions of all simulations in the same context, regardless of the algorithm.

5 Results

5.1 Results for the two traffic demand matrices problems

The experimental results, in terms of TOA, for each of the three algorithms (SOEA, NSGA-II and SPEA2), are summarized in Tables 3 and 4, which respectively present the best and the mean fitness values of all runs, organized by traffic demands levels and different values of the trade-off parameter (β). Congestion values need to be below 10 2/3 to be acceptable. For congestion values above this threshold, the network starts to endure heavy packet losses, due to the over utilization of some links, whereas a congestion value of 1 (optimal) translates into all links utilization being under 1/3 of their capacity.

The experiments with the 30₄ network topology only considered the traffic demand matrices with D0.3 level, as for higher levels of demand all obtained solutions have a congestion level that surpasses the threshold value of 10 2/3, above which the network ceases to operate acceptably. As the number of nodes and their degree increase, so does the optimization problem difficulty, and a higher number of iterations is required for the algorithms to converge.

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. This can be considered an easy problem due to the relative small size of the topology. The mean fitness values, for all levels of demands and trade-offs for the Abilene topology, in Table 4, are also very similar among the three algorithms.

The performance metric C-measure, in Table 5, reinforces the conclusion that all algorithms perform equally well. The metric values are all of the same magnitude and, consequently, no algorithm's PF set dominates the others. Although, with respect to the Abilene network topology, SOEA, NSGA-II and SPEA2 provide equally good solutions, NSGA-II and SPEA2 are able to obtain, in a single run, a larger set of non-dominated solutions than the SOEA. The fact that the Abilene network only has 15 edges, which translates into solution vectors with 30 genes (two weights are assigned for each arc, one for each direction) leads to a reduced search space contributing to the fast convergence of all EAs to the same optimal solution.

For larger network topologies, with more nodes and edges, the performances of the three algorithms start to differ. The results for the synthetic topology 30₂, with 30 nodes and 55

Table 3 Best fitness comparison for two demand matrices optimization

Algorithm	First Demands	Second Demands	Abilene			30 ₂			30 ₄		
			$\beta = 0.25$	$\beta = 0.50$	$\beta = 0.75$	$\beta = 0.25$	$\beta = 0.50$	$\beta = 0.75$	$\beta = 0.25$	$\beta = 0.50$	$\beta = 0.75$
SOEA	0.3	0.3	1.14	1.16	1.18	1.37	1.39	1.40	2.50	2.49	2.47
NSGA-II			1.14	1.16	1.18	1.34	1.35	1.36	1.91	1.97	1.96
SPEA2			1.14	1.16	1.18	1.46	1.45	1.44	4.34	5.62	6.18
SOEA	0.3	0.4	1.45	1.37	1.28	1.75	1.64	1.53	-	-	-
NSGA-II			1.45	1.37	1.28	1.66	1.56	1.45	-	-	-
SPEA2			1.45	1.37	1.28	1.88	1.72	1.56	-	-	-
SOEA	0.4	0.4	1.52	1.52	1.52	1.95	1.99	2.02	-	-	-
NSGA-II			1.52	1.52	1.52	1.84	1.88	1.92	-	-	-
SPEA2			1.52	1.52	1.52	2.14	2.18	2.21	-	-	-

Table 4 Mean fitness comparison for two demand matrices optimization: trade-off analysis

Algorithm	First Demands	Second Demands	Abilene			30 ₂			30 ₄		
			$\beta = 0.25$	$\beta = 0.50$	$\beta = 0.75$	$\beta = 0.25$	$\beta = 0.50$	$\beta = 0.75$	$\beta = 0.25$	$\beta = 0.50$	$\beta = 0.75$
			SOEA	0.3	0.3	1.22	1.22	1.22	3.10	2.98	2.79
NSGA-II			1.21	1.21	1.21	1.93	1.83	1.73	4.31	4.44	4.58
SPEA2			1.21	1.21	1.21	2.19	2.06	1.94	43.98	46.92	49.86
SOEA	0.3	0.4	1.49	1.40	1.31	7.95	6.12	4.29	-	-	-
NSGA-II			1.48	1.39	1.30	3.54	2.86	2.18	-	-	-
SPEA2			1.48	1.39	1.30	4.55	3.62	2.69	-	-	-
SOEA	0.4	0.4	1.57	1.55	1.54	10.45	10.21	9.98	-	-	-
NSGA-II			1.56	1.55	1.54	2.92	2.80	2.68	-	-	-
SPEA2			1.56	1.55	1.54	5.40	5.27	5.14	-	-	-

edges (solutions with 110 genes), show that the NSGA-II algorithm is able to attain better fitness values in all trade-offs (β) and demand level pairs. This can be observed, for instance, with *D0.4* traffic demand matrices and a trade-off weight β of 0.25, where the best and mean fitness values are, respectively, 1.951 and 10.446 (SOEA), 1.841 and 2.919 (NSGA-II), 2.139 and 5.401 (SPEA2). Although the NSGA-II and SOEA best values are relatively similar, and both better than SPEA2 results, the mean values for NSGA-II results are substantially better, and therefore, indicate a more robust algorithm able to obtain good solutions more consistently in the different runs.

The averaged C-measure metric values for the 30₂ network topology scenarios, in Table 5, show that the solutions in NSGA-II PFs dominate SPEA2 PFs, on average, in more than 56% of the cases, with a reverse C-measure of nearly 0%. When compared with SOEA PFs solutions, NSGA-II PFs solutions dominate approximately 16% of the SOEA PF solutions and, for the reverse case, SOEA PFs solutions dominate NSGA-II PFs solutions in roughly 6%. The NSGA-II algorithm, for the 30₂ topology scenarios, offered globally better solutions than any of the two other algorithms, while the SPEA2 algorithm had the worst performance of all.

As the size of the topology increases, the performance of the NSGA-II algorithm detaches from the others. The experiments with the 30₄ network topology show that, while the best values of the three algorithms remain acceptable, NSGA-II features the best solutions. More importantly, considering mean results, it is the only algorithm whose fitness values remain within the acceptable operating limits of the network.

The C metric results for this new set of simulations are very similar to those obtained for the 30₂ topology, and again, NSGA-II PFs are globally better than those provided by the SOEA and SPEA2.

The hypervolume indicators, presented in Table 6, also support that NSGA-II is the best choice in the context of link-state weights setting for two traffic demand matrices. The PFs of NSGA-II are closer to the Pareto-optimal approximation than SOEA and SPEA2.

An illustrative representation of the best solutions obtained by each algorithm for the 30₄ network topology, with two *D0.3* traffic matrices is presented in Fig. 3.

5.2 Results for link failure: highest load

The failure of the network link that carries the highest volume of traffic volume configures one of the worst case scenarios in networks. Its failure implies the re-routing of the largest possible 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 purposes, Table 7 that

Table 5 Overall C-measure for two traffic demand matrices optimization: trade-off analysis

	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	–

Table 6 Mean normalized hypervolume for two traffic demand matrices optimization

Algorithm	Topology		
	Abilene	30 ₂	30 ₄
SOEA	1.000	0.127	0.004
NSGA-II	1.000	0.069	0.000
SPEA2	1.000	1.000	1.000

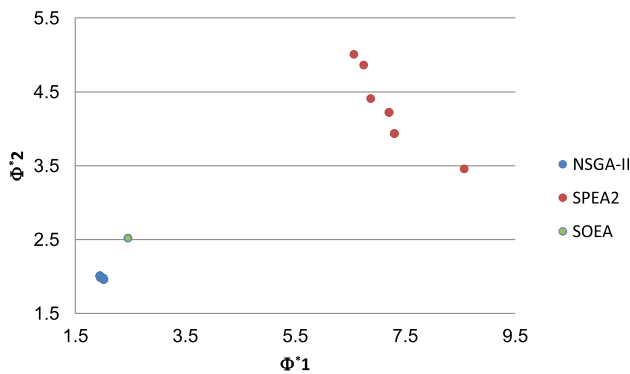


Fig. 3 Two demands congestion optimization (Topology: 30₄, D₁: 0.3 D₂:0.3)

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 (i.e., without any optimization for the future scenario). The single objective optimization was performed by the SOEA algorithm and run with the same previously presented configurations. While the mean value, Table 8, provides a global assessment of each algorithm in all runs, the best TOA value enables to compare each of the objectives fitness values, that is, the congestion values before and after the link failure.

The simulation results show that, for the smallest topology, Abilene, all three algorithms behave alike producing equally good solutions. But, as the topology size increases, or with the escalation of traffic requirements, NSGA-II is able to obtain solutions which translate into lower congestion values, both in the scenarios with or without a link failure, presenting the best solution for both objectives, as well as the best mean TOA congestion values. In some cases, NSGA-II was able

to obtain better results for the first objective, when optimizing both states, than the ones provided by the SOEA when optimizing solely for the first objective, which is a surprising result. This occurred for the more difficult problems, such as, the 30₄ network topology optimization experiments, Table 7.

In the 30₄ network topology scenario, with D0.3 traffic demands and β = 0.5, the fitness values before and after the link failure are, respectively in Table 7, 2.30 and 2.25 for NSGA-II; 11.14 and 7.91 for SOEA; 28.95 and 47.39 for SPEA2. The results allow to observe that the more difficult the problem is, 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 SPEA in all scenarios. The SOEA algorithm low quality results may be explained by its requirement of a higher number of generations to properly converge. The gap in performance between the algorithms is also mirrored in the averaged TOA, as shown in Table 8.

The C-measure values in Table 9 show that, despite being able to provide solutions with equivalent best fitness for the Abilene topology, the SOEA produces more solutions that are neither dominated by NSGA-II or SPEA2 solutions. In contrast, for the more demanding topologies, 30₂ and 30₄, NSGA-II PFs solutions dominate approximately 14% of the SOEA PF solutions, while the reverse is about 7%. When compared against SPEA2 PFs solutions, both NSGA-II and SOEA present better values.

Concerning the hypervolume metric, in Table 10, NSGA-II performs slightly worse for the Abilene topology, but better for the other two networks. Figure 4 presents the best solutions for the single link failure congestion optimization when applied to the 30₄ topology for a D0.3 expected mean of congestion.

5.3 Results for link failure: centrality-based link selection

A network administrator may consider that protection against the failure of a particular link is more crucial than others, because of its capacity, failure probability, or for any other reason. It is, therefore, important to allow an administrator to select the link which is to be protected. In such as context, the optimizations were run selecting the link that occurs in the largest number of shortest paths, higher centrality, when the network is configured with link weights

Table 7 Best fitness values for single link failure weights setting optimization: highest load link

Topology	Demand	Without Link failure optimization		With link failure optimization						Algorithm
		Before	After	$\beta = 0.25$		$\beta = 0.5$		$\beta = 0.75$		
				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

Table 8 Mean fitness comparison for single link failure: highest load link

Algorithm	Demands	Abilene			30 ₂			30 ₄		
		$\beta = 0.25$	$\beta = 0.50$	$\beta = 0.75$	$\beta = 0.25$	$\beta = 0.50$	$\beta = 0.75$	$\beta = 0.25$	$\beta = 0.50$	$\beta = 0.75$
SOEA	0.3	1.53	1.46	1.39	36.29	31.12	25.96	146.21	130.40	114.60
NSGA-II		1.37	1.34	1.30	5.76	6.64	7.51	23.07	30.95	38.83
SPEA2		1.37	1.33	1.30	8.10	11.05	14.00	128.35	135.72	143.10
SOEA	0.4	10.80	7.89	4.97	45.26	38.92	32.59	251.79	238.54	225.30
NSGA-II		3.24	2.69	2.14	11.26	12.35	13.44	74.61	79.91	85.21
SPEA2		3.40	2.79	2.19	12.29	13.00	13.71	417.23	420.31	423.38
SOEA	0.5	175.38	119.23	63.09	149.58	126.89	104.20	–	–	–
NSGA-II		116.54	78.38	40.22	46.97	41.95	36.93	–	–	–
SPEA2		117.56	79.08	40.59	53.18	51.70	50.22	–	–	–

Table 9 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	–

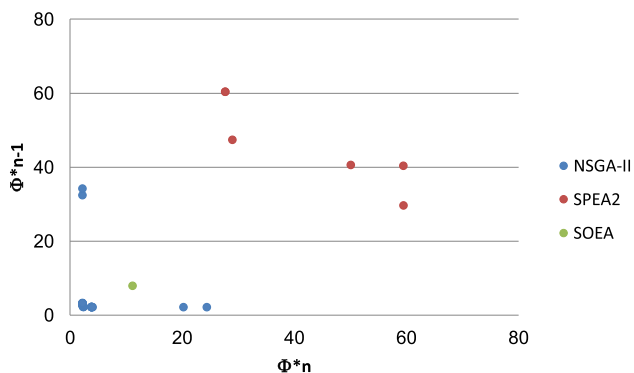


Fig. 4 Single link failure congestion optimization (Topology: 30₄, D:0.3)

Table 10 Mean normalized hypervolume for single link failure optimization

Algorithm	Topology		
	Abilene	30 ₂	30 ₄
SOEA	0.667	0.180	0.048
NSGA-II	1.000	0.178	0.020
SPEA2	0.674	1.000	1.000

inversely proportional to their capacity. The minimal congestion values, before and after the failure of the selected link, are presented in Table 11 while the average TOA congestion values are shown in Table 12. Distinct trade-offs values of $\beta = 0.25, 0.5, 0.75$ were considered.

Results from this new test suite consolidate previous observations. For simpler problems, with smaller topologies and lower traffic demand levels, the SOEA and MOEAs algorithms provide equally good solutions. As the number of nodes and links increases, as well as with growing traffic necessities, NSGA-II delivers better solutions, both before and after the link failure, in the large majority of tested scenarios.

The C metric results, in Table 13, are also similar to those observed for the highest load link failure optimization. SOEA continues to have, in average, more solutions that are not dominated by any of the non-dominated sets of solutions resulting from NSGA-II and SPEA2 based optimization in the context of simpler problems. As the difficulty of the MOP increases, NSGA-II begins to provide better sets of non-dominated solutions.

5.4 Results for distinct weight ranges

As mentioned earlier, solutions are evaluated from a reduced search space, where weights take values within the [1; 20] range. Although this reduced range has been used in several other studies that aim to find optimized weight settings for congestion (Fortz 2000), we here also present a brief compar-

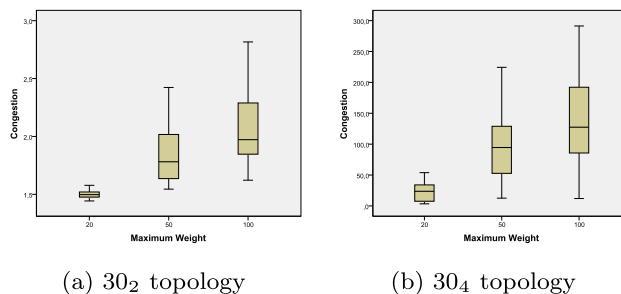


Fig. 5 Maximum weight influence in the congestion value

ison of the effect of such reduced search space on the quality of the obtained solutions.

The experiments consider weight ranges $[1, w_{max}]$, with $w_{max} \in (20, 50, 100)$. A SOEA, with the previously described running configurations, was used to optimize the congestion for each of the network topologies, considering D0.3 traffic matrices. The optimization results obtained by running 30 times each scenario are presented in Fig. 5.

It is possible to observe that the solutions quality depends on the value of the maximum weight parameter. By reducing the search space, the EA converges faster, requiring less iterations. Simultaneously, shorter ranges increase the probability of finding equal cost multi-paths. It is expected that by increasing the number of iterations, solutions with equally good quality will be found independently of the weight range. It is important to observe that the best solution for all ranges was always obtained with the narrower weight configuration, and that by increasing the size of the weight range no better single solution was obtained.

In the context of multi-objective problems, the weight range assigned to each link also has an impact on the convergence and diversity of the solutions. As the weight range increases, the number of iterations, required by the algorithm, to converge to the Pareto-optimal also increases. The range selection choice reflects the importance given to the convergence, versus the diversity of the solutions. As an illustrative example, Fig. 6 presents the best Pareto fronts of 30 runs for MOPs that aim to optimize two distinct D0.3 traffic matrices for the 30₄ network topology.

5.5 Results for EAs running time

The amount of time required to obtain optimized solutions is also a very important factor when choosing an MOEA. As the size of the solutions depends on the number of links, the time required by each MOEA depends mostly on the number of edges on the topology. Some illustrative running times are presented in Table 14 which were obtained with a Core i7 processor.

None of the algorithms uses multi-threading, and, consequently, the optimization process duration can directly

Table 11 Best fitness values for single link failure weights setting optimization: centrality-based link

Topology	Demand	Without Link failure optimization		With link failure optimization						Algorithm
				$\beta = 0.25$		$\beta = 0.5$		$\beta = 0.75$		
		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

Table 12 Mean fitness comparison for single link failure: centrality-based link

Algorithm	Demands	Abilene			30 ₂			30 ₄		
		$\beta = 0.25$	$\beta = 0.50$	$\beta = 0.75$	$\beta = 0.25$	$\beta = 0.50$	$\beta = 0.75$	$\beta = 0.25$	$\beta = 0.50$	$\beta = 0.75$
SOEA	0.3	1.63	1.49	1.36	55.57	37.61	19.65	165.95	126.13	86.31
NSGA-II		1.61	1.48	1.35	1.75	1.66	1.58	7.48	6.83	6.18
SPEA2		1.61	1.48	1.35	2.67	2.44	2.21	426.39	421.56	416.73
SOEA	0.4	19.45	13.49	7.53	74.27	50.65	27.03	165.95	126.13	86.31
NSGA-II		25.52	17.54	9.56	6.37	4.89	3.40	38.74	37.29	35.83
SPEA2		25.52	17.54	9.56	11.03	8.59	6.15	714.89	707.28	699.67
SOEA	0.5	227.12	152.07	77.02	178.50	129.88	81.26	–	–	–
NSGA-II		90.07	60.71	31.36	45.40	34.02	22.65	–	–	–
SPEA2		90.06	60.71	31.36	80.59	71.55	62.51	–	–	–

Table 13 C-measure of centrality-based 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	–

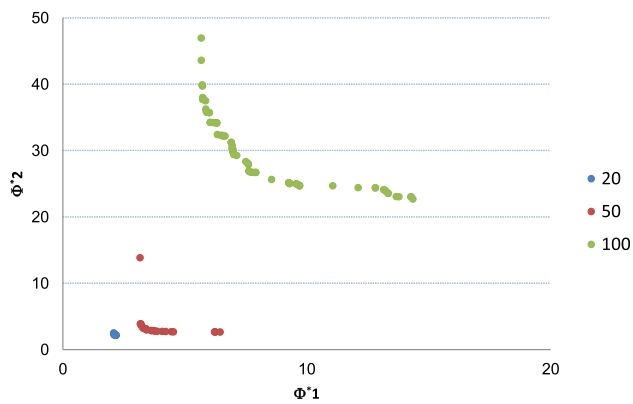


Fig. 6 Pareto fronts with distinct weight range configurations

Table 14 Mean run time per iteration for two demands optimization (in seconds)

Topology	Nodes	Edges	SOEA	NSGA-II	SPEA2
Abilene	12	15	0.061	0.031	0.070
30_2	30	55	0.635	0.735	1.459
30_4	30	110	0.964	1.234	1.889

be compared. The amount of time, per iteration, presented for the SOEA is relative to a single aggregated sum configuration, and not the three trade-offs considered in the experiments.

As an example, the amount of time required for each algorithm, when applied to congestion optimization for the 30₂ topology with a stopping criterion of 1000 iterations, is 31 min for SOEA (3 runs are needed to obtain the 0.25, 0.5 and 0.75 trade-offs), 12.3 min for NSGA-II, and 24.3 min for SPEA2.

It is, nonetheless, possible to improve the time required by each algorithm. One obvious way to achieve this goal is to resort to multi-threading in the evaluation process, where each parallel thread would evaluate a portion of the solutions.

6 Conclusions

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 reconfiguring the link weights assigned to each link, but with a temporary negative impact on traffic flows. Preventive opti-

mization approaches can effectively take into consideration such changes to compute weights that allow the network to continue to ensure good levels of performance even in variable environment conditions.

In this study, two link-state weight setting problems were approached using single and multi-objective EAs: NSGA-II, SPEA2 and single-objective EA using weighted-sum aggregation. Results showed that, for simpler problems, single objective optimization approaches provide solutions with fitness values as good as the MOEA algorithms. As the difficulty of the addressed problems increases, for larger network topologies and more demanding traffic requirements, NSGA-II offers better solutions. Moreover, NSGA-II is able to deliver a broader set of solutions, with distinct trade-offs between objectives, within a shorter computational time. Additionally, results indicate that the use of a reduced search space does not impact negatively the quality of the solutions, enabling a faster convergence of the EAs.

Future work will address the exploration of parallel optimization techniques, such as the parallel evaluation of solutions in a same iteration and Island Models. In Island Models, isolated populations evolve simultaneously and in scheduled moments, or probabilistically, solutions that verify some configured policy migrate between populations. The populations can be homogeneous, that is, share common characteristics, or heterogeneous, with different parameters, such as different operators or even different optimization algorithms.

Acknowledgements This work has been supported by FCT-Fundação para a Ciência e Tecnologia within the R&D Units Project Scope UIDB/00319/2020 and UIDB/04469/2020 units.

References

- Altin A, Fortz B, Ümit H (2012) Oblivious OSPF routing with weight optimization under polyhedral demand uncertainty. *Networks* 60(2):132–139
- Altin A, Fortz B, Thorup M, Ümit H (2013) Intra-domain traffic engineering with shortest path routing protocols. *Ann Oper Res* 204(1):65–95
- Awduche D, Malcolm J, Agogbua J, O'Dell M, McManus J (1999) Requirements for traffic engineering over MPLS. RFC 2702 (Informational)
- Bäck T (1996) Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming. Genetic algorithms. Oxford University Press, Oxford
- Ben-Ameur W, Gourdin E, Liao B, Michel N (2000) Optimizing administrative weights for efficient single-path routing. In: *Proceedings of networks*
- Bley A, Fortz B, Gourdin E, Holmberg K, Klopfenstein O, Pióro M, Tomaszewski, Ümit H (2010) Optimization of OSPF routing in IP networks. In: Koster AMCA, Mñuoz X (eds) *Graphs and algorithms in communication networks: studies in broadband, optical, wireless and ad hoc networks*, Springer, Berlin, Chap. 8, pp 199–240

- Cisco customer case study, maximizing return on network infrastructure investment with Cisco MATE design
- Coello C, Pulido G, Lechuga M (2004) Handling multiple objectives with particle swarm optimization. *IEEE Trans Evol Comput* 8(3):256–279
- Corne D, Jerram N, Knowles J, Oates M (2001) PESA-II: region-based selection in evolutionary multiobjective optimization. In: *Proceedings of the genetic and evolutionary computation conference (GECCO-2001)*, San Francisco, California. Morgan Kaufmann Publishers, pp 283–290
- Dahai X, Mung C, Jennifer R (2011) Link-state routing with hop-by-hop forwarding can achieve optimal traffic engineering. *IEEE/ACM Trans Netw* 19(6):1717–1730
- Deb K (2001) *Multi-objective optimization using evolutionary algorithms*. Wiley, New York
- Deb K, Agrawal S, Pratap A, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6(2):82–97
- Dijkstra E (1959) A note on two problems in connexion with graphs. *Numer Math* 1(1):269–271
- Ericsson M, Resende M, Pardalos P (2002) A genetic algorithm for the weight setting problem in OSPF routing. *J Combin Optim* 6:299–333
- Evangelista P, Maia P, Rocha M (2009) Implementing metaheuristic optimization algorithms with jecoli. In *Proceedings of the 2009 ninth international conference on intelligent systems design and applications, ISDA '09*, Washington, DC, USA. IEEE Computer Society, pp 505–510
- Floyd S, Fall K (1999) Promoting the use of end-to-end congestion control in the internet. *IEEE/ACM Trans Netw* 7(4):458–472
- Fortz B (2000) Internet traffic engineering by optimizing OSPF weights. In: *Proceedings of IEEE INFOCOM*, pp 519–528
- Fortz B, Thorup M (2003) Robust optimization of OSPF/IS-IS weights. In: Ben-Ameur W, Petrowski A (eds) *Proceedings of INOC*, pp 225–230
- Goldberg E (1989) *Genetic algorithms in search, optimization and machine learning*, 1st edn, Addison-Wesley
- Gredler H, Goralski W (2005) *The complete IS-IS routing protocol*. Springer, Berlin
- Hopps C (2000) Analysis of an equal-cost multi-path algorithm, internet engineering task force , IETF , RFC 2992 (Informational)
- Iannaccone G, Chuah C, Mortier R, Bhattacharyya S, Diot C (2002) Analysis of link failures in an ip backbone. In: *Proceedings of the 2nd ACM SIGCOMM workshop on internet measurement, IMW '02*, New York, NY, USA, ACM, pp 237–242
- Jain S, Kumar A, Mandal S, Ong J, Poutievski L, Singh A, Venkata S, Wanderer J, Zhou J, Zhu M, Zolla J, Hözl U, Stuart S, Vahdat A (2013) B4: experience with a globally-deployed software defined wan. In: *Proceedings of the ACM SIGCOMM conference on SIGCOMM (SIGCOMM '13)*, ACM, New York, NY, USA, pp 3–14
- Knowles J, Corne D (2000) Approximating the nondominated front using the Pareto archived evolution strategy. *Evol Comput* 8(2):149–172
- Medina A, Lakhina A, Matta I, Byers J (2001) Brite: universal topology generation from a user's perspective. Technical report, Boston, MA, USA
- Moy J (1998) OSPF Version 2. RFC 2328 (Standard), April . Updated by RFC 5709
- Pereira V, Rocha M, Cortez P, Rio M, and Sousa P (2013) A framework for robust traffic engineering using evolutionary computation. In: *7th International conference on autonomous infrastructure, management and security (AIMS 2013)*, Barcelona, Spain, Springer, LNCS 7943, pp 2–13
- Pereira V, Sousa P, Cortez P, Rio M, Rocha M (2015) Comparison of single and multiobjective evolutionary algorithms for robust link-state routing. In: *Proceedings of evolutionary multi-criterion optimization (EMO2015)*, Springer, Guimarães, Portugal, pp 573–87
- Pereira V, Rocha M, Sousa P (2016) optimizing load balancing routing mechanisms with evolutionary computation. In: *Proceedings of international workshop on future internet and smart networks (FI&SN 2016)*, IOS Press, Ambient Intelligence and Smart Environments, Intelligent Environments, vol 21
- Razali N and Geraghty J (2011) Genetic algorithm performance with different selection strategies in solving TSP. In: *Proceedings of the world congress on engineering 2011, Vol II WCE 2011*, London, U.K, July 6–8
- RFC 3954-Cisco Systems NetFlow services export version 9, B. Claise, Ed., Cisco Systems, October 2004
- Rocha M, Sousa P, Cortez P, Rio M (2011) Quality of service constrained routing optimization using evolutionary computation. *Appl Soft Comput* 11(1):356–364
- Schaffer J (1984) Some experiments in machine learning using vector evaluated genetic algorithms (artificial intelligence, optimization, adaptation, pattern recognition), PhD thesis, Vanderbilt University
- Schaffer J (1985) Multiple objective optimization with vector evaluated genetic algorithms. In: *Proceedings of the 1st international conference on genetic algorithms*, L. Erlbaum Associates Inc., Hillsdale, NJ, USA, pp 93–100
- Srinivas N, Deb K (1994) Multiobjective optimization using nondominated sorting in genetic algorithms. *Evol Comput* 2(3):221–24
- Suchara M, Xu D, Doverspike R, Johnson D, Rexford J (2011) Network architecture for joint failure recovery and traffic engineering. In: *Proceedings of SIGMETRICS11 conference*
- Tan KC, Lee TH, Khor EF (2002) Evolutionary algorithms for multi-objective optimization: Performance assessments and comparisons. *Artif Intell Rev* 17(4):251–290
- Technologies Cariden (2012) Building traffic matrices: introduction to MATE flow collection. White Paper-Version 2
- Tune P, Roughan M (2014) Network design Sensitivity Analysis. In: *The 2014 ACM international conference on measurement and modeling of computer systems*, ACM, SIGMETRICS 14, pp 449–461
- Villamizar C, Song C (1994) High performance TCP in ANSNET. *Comput Commun Rev* 24(5):45–60
- Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1:67–82
- Zitzler E, Thiele L (1998) An evolutionary algorithm for multiobjective optimization: the strength Pareto approach. Technical report 43, Gloriastrasse 35, CH-8092 Zurich, Switzerland
- Zitzler E, Laumanns M, Thiele L (2001) Spea2: improving the strength pareto evolutionary algorithm. Technical report

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.