Business-driven policy optimization for service management

Javier Rubio-Loyola,[1,*,†] Gregorio Toscano-Pulido,[1] Marinos Charalambides,[2]
Marisol Magaña-Aguilar,[1] Joan Serrat-Fernández,[3] George Pavlou[2] and
Hiram Galeana-Zapién[1]

[1] *CINVESTAV Tamaulipas, Ciudad Victoria, Mexico*
[2] *University College London, London, UK*
[3] *Universitat Politècnica de Catalunya, Barcelona, Spain*

SUMMARY

The performance of services offered by network operators has a direct impact on its revenue due to new customer subscriptions, and also on penalties that can apply when services are not provided to an acceptable quality level. Previous research on business-oriented network and service optimization has mainly focused on optimizing individual business indicators, such as profit and revenue, in isolation without analyzing the effect on network configurations and the subsequent impact on other indicators. Given that different business objectives are usually incompatible, a single network configuration cannot optimize them simultaneously. Determining the configuration and the associated trade-offs that satisfy multiple objectives is a complex task. This paper addresses this gap and presents a framework that derives policy configurations that optimize the business value of the network infrastructure. We describe a methodology to quantify business functions considering the dynamics of network events, the dynamics of end-user service usage, the nature of the business indicators, and their relationships with the underlying control methods. The proposed approach addresses the complexity of the target problem through a surrogate-based optimization approach properly tailored to match our application domain needs. We evaluate the effectiveness of the proposed approach through experimentation in a simulation environment we developed over OPNET. Copyright © 2015 John Wiley & Sons, Ltd

Received 20 May 2014; Revised 20 November 2014; Accepted 6 January 2015

1. INTRODUCTION

Differentiated services (DiffServ) have been proposed as a scalable approach for providing quality of service (QoS) in IP networks. The core philosophy is grouping traffic with similar QoS requirements into a limited number of service classes, allocating bandwidth to these classes, and differentiating their forwarding treatment throughout the network. While different QoS levels can be provided, the absence of control over traffic can result in resource starvation and network congestion. Internet service providers (ISPs) operating DiffServ networks can use advanced policy-based control mechanisms to prevent and resolve network congestion, to control service subscriptions and invocations, and to perform dynamic traffic engineering functions for QoS management [1]. In this context, policies are decision-based rules implemented in the network elements and management systems to provide flexibility and constrained programmability of the management functions.

[*] Correspondence to: Javier Rubio-Loyola, Information Technology Laboratory, CINVESTAV Tamaulipas, Ciudad Victoria, Mexico.
[†] E-mail: jrubio@tamps.cinvestav.mx

Although service management solutions have proven to be efficient in providing QoS guarantees, the requirements, methodologies, and incremental efforts to elevate the business value of ISPs operating DiffServ networks and other QoS-oriented domains have remained almost unexplored.

In previous work, we developed an analytical approach for automatically deriving service control policy parameters for DiffServ QoS management, from a business-level perspective [2], with the use of business indicators (BIs), which are indicators that reflect the business strategies of the ISPs. This previous work took a step towards bridging the gap between business value and configuration management by considering the influence of BIs when generating configuration policies. This is realized by a set of linear mapping functions that take into account the impact of BIs over service management policies. Administrator-assigned weights of importance are given to each BI, which are used to derive policy parameters. The mapping functions, however, are simple and may not necessarily result to optimal policy parameter values, mainly because of the possibility of contradictory business objectives. Consider, for example, two representative BIs that relate to the following: (a) the volume of service subscriptions and (b) the level of service satisfaction. A network operator prioritizing the former BI will have a higher economic benefit by taking actions to maximize the number of subscribers. These actions, however, could have a negative impact on service satisfaction because an excessive number of users could eventually cause resource starvation by injecting too much traffic into the network. Consequently, it is necessary to develop appropriate mechanisms that can determine trade-offs between BIs and also to ensure that the underlying policy-based control mechanisms optimize the BIs according to administrative strategies.

This paper presents two novel contributions: (i) an offline optimization framework that provides feedback to network operators in the form of a catalogue of policy configurations that optimize the business value of a given network infrastructure and (ii) a surrogate model that can effectively deal with the complexity and dynamic constraints inherent to the optimization of BIs addressed in this research.

The optimization framework proposed in this paper represents business strategies with BIs, which are monitored and modeled as objective functions of measurable parameters of the managed system. For this, the framework takes into account the dynamics of network events, the diversity of enforceable QoS management actions, the dynamics of end-user service usage, the nature of the BIs, and their relationships with the underlying control methods.

In order to derive optimized policy configurations, the framework uses optimization techniques. Optimization is the procedure of finding and comparing feasible solutions until no better solution can be found [3]. More specifically, our framework uses multi-objective optimization (MOO) techniques because multiple objectives need to be optimized simultaneously. In this work, the term *objective* refers to a BI, and the term *solution* refers to policy values that optimize one or more BIs according to administrative preferences. Determining the most optimal solutions relies on search techniques from which evolutionary algorithms (EAs) have been found to be efficient in locating solutions close to the global optimum, even in highly rugged search spaces [4].

Evolutionary algorithms are population-based techniques that perform multidimensional search and can find more than one solution within an execution. In this work, the term *population* refers to policy configuration values that are considered in the search space. However, despite their success, EAs have limitations when solving computationally expensive optimization problems. Because of the dynamics of network events and service provisioning as well as the range of values configuration policies can take, the optimization of multiple BIs falls under this category of problems. This issue can be addressed by surrogate[1] models such as the ones proposed in the literature [5–8]. Most of these solutions have been developed to solve specific problems, the most successful of which is Pareto efficient global optimization (ParEGO) [6] as it has managed to solve problems that cannot be solved with the other models. Nevertheless, existing techniques are only effective when the number of function evaluations has at most 10 variables [6,9], which are not adequate for the problem addressed in this work; for example, more than 10 policy variables may need to be evaluated. The definition of a surrogate model for BI optimization with dynamic constraints is also a core contribution of this paper.

[1] *Also known as metamodels, emulators, reduced models, approximate models, and response surface models*

This paper presents the policy optimization framework, the surrogate optimization model, and their implementation as an overall solution. The solution is implemented in the OPNET network simulator toolkit and additional software that has been developed to assess the framework's functionality. The system is evaluated with a scenario that describes the practicality of the system as well as the necessary elements for its usage. The system derives a catalogue of policy configurations, which represents the most ideal policy configurations a network operator should deploy in the underlying network infrastructure to effectively implement the network operator's business guidelines.

The paper is organized as follows. Section 2 describes a simple business-driven management approach. Section 3 describes the optimization framework. Section 4 provides details of the surrogate model, and Section 5 presents the experimental validation of the solution. Section 6 presents related work in the area. Finally, Section 7 concludes with directions for future work.

2. A SIMPLE BUSINESS-DRIVEN MANAGEMENT APPROACH

2.1. *Admission control for service management*

As previously proposed in the literature [1], ISPs operating DiffServ networks can realize service management functionality by performing admission control (AC) at different timescales. Static AC has centralized offline functionality and controls the number and type of service subscriptions. A number of configurable parameters can prevent network overloading while maximizing subscribed traffic. Such parameters define service satisfaction factors (i.e., service rates (SRs)), service quality levels, subscription limits, and AC strategies for subscription requests.

Dynamic AC, on the other hand, implements distributed service invocation logic at network ingresses and controls the number of active services as well as the volume of admitted traffic at run time. To regulate the traffic entering the network, dynamic AC functionality defines configurable thresholds that signal critical levels of traffic and subsequent actions for adjusting the SRs and controlling the invocations of subscribed services.

2.2. *Business-driven admission control*

In previous work, we developed an approach for automatically deriving AC parameters, as part of a policy refinement process, from a business-level perspective [2]. This approach took a step towards bridging the gap between business value and configuration management by considering the influence of BIs when generating relevant configuration policies.

The adopted policy refinement process is based on a set of high-level objectives specific to service management functions. Such objectives aim, for example, to control the subscription volume, the service quality, and QoS degradation. These objectives are achieved by low-level policies that perform AC and regulate SRs. All in all, these low-level policies achieve the business goals of ISPs, which are specified in terms of BIs.

Business indicators can be used by service providers to define their preferences in achieving specific goals. They relate to service management objectives (SMOs) and subsequently control the configuration parameters of derived policies. For example, consider three types of BIs that can potentially be used by a service provider: *service satisfaction*, *profit*, and *loss*. The first is directly linked to the level at which a service is perceived by the user, for example, contractual rates, service quality, and availability, whereas the last two relate to the revenue generated by contracted services.

Profit is directly proportional to the subscription volume, because more subscriptions imply higher revenue, but it is inversely proportional to quality because with lower quality levels (e.g., in terms of bandwidth), more services can be accommodated in the available resources. Although more profit can be generated by oversubscribing, the risk of not delivering a service to the contracted quality level increases. This has a negative impact on the *service satisfaction* BI, which is directly proportional to the service quality.

The *service satisfaction* BI also relates to SMOs that control QoS degradation, new service invocations, and active services. High levels of satisfaction imply above-average SRs for active services, high service availability in terms of invocation requests, and late QoS degradation preventative actions.

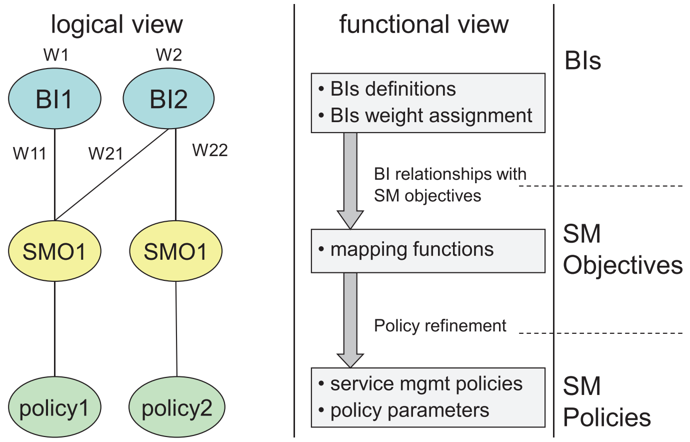

Figure 1. Relationships between business indicators (BIs), objectives, and policies. SM, service management

Profit can be negatively influenced by penalties that an ISP has to pay when a service is not delivered to an acceptable standard, that is, service-level agreement (SLA) violations. High losses can be incurred by a very early (poor usage of resources) or delayed (increased risk of congestion) QoS degradation prevention actions and also by providing below-contractual-average SRs.

It is evident that a number of trade-offs can be considered when achieving the various SMOs. Based on the relationships described previously, we have developed an approach for deriving configuration policies and associated parameter values that reflect the importance of the defined BIs; this is based on the three-level hierarchy depicted in Figure 1. For the sake of clarity, only two BIs are illustrated at the upper layer where each is assigned with a weight representing its importance in achieving a business-level objective. Weights take values from 0 to 1, with the former signifying the least importance of a BI to an ISP.

The BIs are directly linked to individual SMOs that they influence. It is common for a BI to apply to more than one SMO as shown in the left part of Figure 1. In this case, the developed approach assumes that the impact of that BI on the relevant SMOs is of the same magnitude, that is, $W2 = W21 = W22$. The impact of an indicator is represented with a set of mapping functions, which take the associated weight into account. These are used during the refinement process to derive the value of policy parameters according to the business-level objective. The approach treats the effect of those indicators independently by providing separate weighted linear mapping functions, for example, for W11 and W21. The resulting value of a policy parameter is determined by averaging the output of individual functions.

2.3. *Limitations*

The policy refinement process outlined earlier has taken an important step towards bridging the gap between business value and configuration management. However, because of its analytical basis, it lacks the operational means to define quantitatively the fulfillment of high-level objectives. Dynamic network conditions, traffic management operations, run-time execution of policies, and service invocation dynamics are some of the key elements that should be taken into account when deriving policies that maximize business value in practice. Furthermore, the simplicity of the linear mapping functions defined between BIs and service management policies may not necessarily result to optimal policy parameter values, especially in front of contradictory business objectives. As such, the objective of the approach presented next is to extend and improve prior work by addressing these limitations.

3. BUSINESS-DRIVEN POLICY OPTIMIZATION FRAMEWORK

This section describes the policy optimization framework, which includes the components of the policy optimization framework and a concrete methodology to evaluate BIs quantitatively.

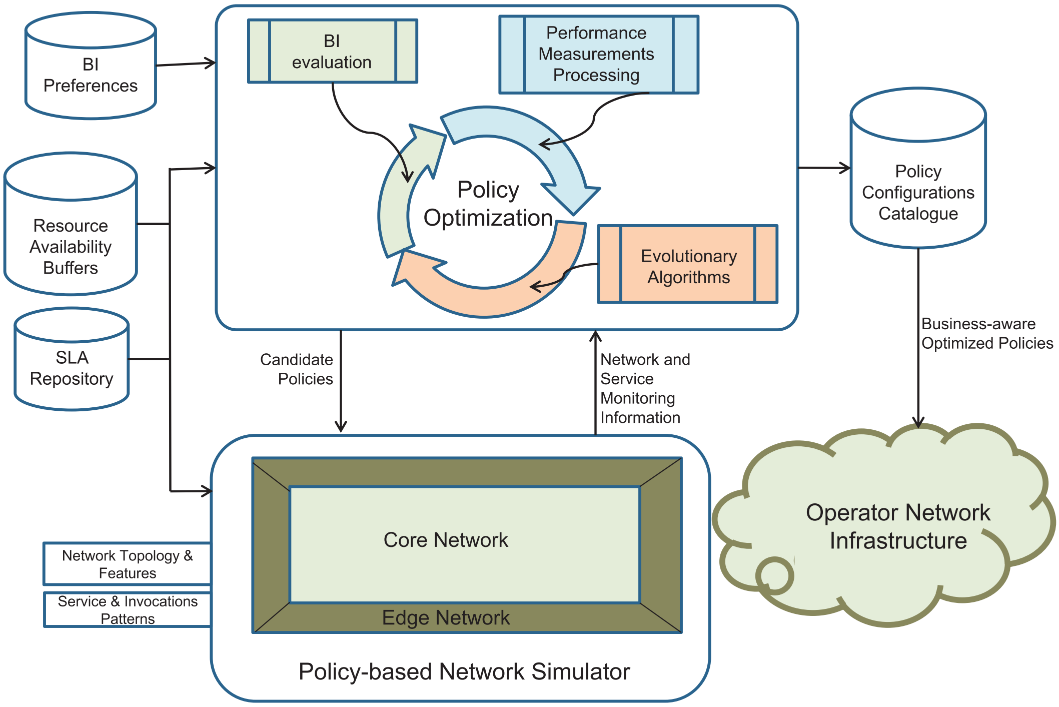
BI Preferences
BI evaluation
Performance Measurements Processing
Policy Optimization
Resource Availability Buffers
SLA Repository
Evolutionary Algorithms
Policy Configurations Catalogue
Candidate Policies
Network and Service Monitoring Information
Business-aware Optimized Policies
Network Topology & Features
Service & Invocations Patterns
Core Network
Edge Network
Policy-based Network Simulator
Operator Network Infrastructure

Figure 2. Policy generation framework. BI, business indicator; SLA, service-level agreement

3.1. Components of the policy optimization framework

The proposed solution is based on an offline policy optimization framework, the main objective of which is to generate a catalogue of service management policies that optimize the business value of the network infrastructure. The various components of this framework are depicted in Figure 2.

The core component of the framework is the *policy optimization* module, which searches the most optimal policies within the policy values' search space to locate solutions that optimize the business objectives. Policy optimization is an iterative process, supported by *EAs* (detailed in Section 4) that are in charge of finding the most optimal values of the service management policies. During their execution, the algorithms produce candidate policies whose optimality in comparison with the business objectives is evaluated iteratively until no better solution can be determined. The EAs are in turn supported by a process that provides numerical values of the objectives to be optimized. Quantifying the BIs is pivotal for the algorithms' search activity, and it is the responsibility of the *BI evaluation* module. The latter follows novel and well-defined methods (detailed in Sections 3.2–3.4) that consider network and service performance information. By correlating such information with business-aware data, they assign numerical values to the BIs accordingly. BI evaluation is supported by the *performance measurement processing* module, whose main responsibility is to handle and manage raw monitoring data from the network simulator such as injected traffic, service invocations, service rejections, and QoS enjoyed by active users.

The *policy optimization* component is configured with the parameters shown in the left part of Figure 2, namely the *BI preferences*, the *resource availability buffers* (RABs), and the *SLA repository*. *BI preferences* define desirable BI levels, which are taken into account when generating the policy configurations. In other words, the enforcement of the policies derived by the optimization process would result in the most ideal configurations to reach the desired preference levels of the BIs. Another configuration set concerns the *RABs*. In the QoS management approach adopted in this work, a *RAB* maintains the aggregate demand of subscribed services per *traffic trunk* (TT), defined as aggregated traffic flows with the same origin–destination and the same performance requirements. Flows with different levels of services are aggregated to different TTs. The policy optimization process is driven by variables and values linked to the TTs of the network.

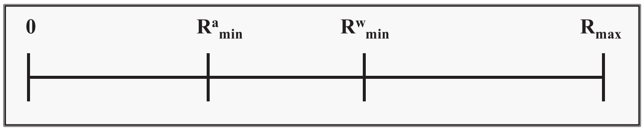

Figure 3. The resource availability buffer

Table 1. Structure of the policy configurations catalogue

BI preferences set *a*	Pareto front for BI preferences set *a*	Pareto point *a*1: BI values	Service management policies for TT_{a11}
			…
			Service management policies for TT_{a1c}
		…	…
		Pareto point *ab*: BI values	Service management policies for TT_{ab1}
			…
			Service management policies for TT_{abc}

BI, business indicator; TT, traffic trunk.

As illustrated in Figure 3, the *RAB* has two main ranges: traffic injection up to $R^{a}_{\min}$ can be used with high confidence even at times of congestion, whereas the area between $R^{w}_{\min}$ and $R_{\max}$ is risky because the network cannot provide QoS guarantees [1]. The *policy optimization component* considers the *RAB* to define the search space of the policy values. Section 5 shows how the values of $R^{a}_{\min}$, $R^{w}_{\min}$, and $R_{\max}$ are derived in practice. Finally, the *SLA repository* is another configuration set for the policy optimization component. The number of service types and their technical specifications formalized therein represent the basis to quantify the BIs during the optimization process. For example, based on the contractual SRs, the *policy optimization* can define whether a service is degraded, at which extent, and for how long. This way, the effect of degradations (and other SLA-aware aspects) can be correlated with appropriate BI numerical values.

During the search process, the *policy optimization* produces candidate policy values, which are fed to the policy-based simulator. The latter is a modified OPNET toolkit with enhanced functionality to support the execution of the service management policies on the fly, with capabilities to produce network and service monitoring information during its simulation runs and also with capabilities to be configured on the fly. When performing simulations, the two configuration data shown in the lower-left part of Figure 2 are considered. The *network topology and features* consist of the access and core routers, links with specific capacities, the TTs of the network, and the DiffServ mapping configurations of the network infrastructure. The simulation of dynamic network conditions and run-time policy execution is performed by considering service invocation dynamics. These are provided as inputs to the simulator in the form of *service and invocation patterns*, which consist of the service characteristics and the service invocation periods including durations (service life cycle of observation times ranging from minutes to hours). This information represents the traffic patterns injected to the network ingress points. With all these inputs, the *policy-based simulator* provides to the policy optimization component raw network and service monitoring information that represents the result of the deployment of the candidate policies generated by the optimization component.

The *policy optimization* component produces optimal policy value sets that are stored in the *policy configuration catalogue*. Table 1 illustrates the structure of the catalogue. For a given set of *BI preferences*, the *policy optimization* produces a Pareto front [10]. Each point in a Pareto front is defined by numerical values of the BIs. These points render the closest optimal BI values in comparison with the desired levels defined by the *BI preferences*. For each Pareto point, the optimization process produces the service management policies for all TTs in the network, which represent the configurations most

Table 2. Service management policies

ID	Policy	Description
P1.1	setTCL(TT,TCL)	Set the TCL threshold with respect to RAB per TT
P1.2	setSR(TT,SR)	Sets the SR of a TT
P2.3	setACth(TT,AC)	Sets the AC limit with respect to RAB per TT

AC, admission control; RAB, resource availability buffer; SR, service rate; TCL, target critical levels; TT, traffic trunk.

aligned to the *BI preferences*. The operator can thus select from the catalogue the most appropriate policy configurations for the TTs to manage its network infrastructure. It is worth mentioning that the policies produced by the framework should be eventually enforced in the actual network. For this purpose, the produced policies could be distributed through Common Open Policy Service-Policy Provisioning or other protocols (e.g., Simple Object Access Protocol or Network Configuration Protocol), and eventually, they should be implemented in the network elements (e.g., the edge routers of the actual policy-based DiffServ/Multiprotocol Label Switching (MPLS) network). The distribution and implementation of policies are processes that lay out of the scope of this paper.

The service management policies we consider in this work are three for each TT, as shown in Table 2. These policies perform dynamic control on the number of active services, as well as on the volume of injected traffic. Policy P1.1 defines a threshold that signals *target critical levels* (TCL) of traffic and TCL $\in [\mathrm{R}^{a}_{\min}, \mathrm{R}_{\max}]$. The closer the TCL is to $\mathrm{R}^{a}_{\min}$, the earlier a notification is issued, whereas values close to $\mathrm{R}_{\max}$ result in delayed proactive actions. The run-time operation of the dynamic control is triggered by TCL crossing alarms, which activate policies P1.2 and P1.3 for adjusting the *SR* and the dynamic *AC* threshold ($\mathrm{AC}_{\mathrm{th}}$) of a TT. The $\mathrm{AC}_{\mathrm{th}}$, which is in the range $[\mathrm{R}^{a}_{\min}, \mathrm{R}_{\max}]$, controls invocations of already subscribed services. The lower it is, the less the chances are of an incoming service being successfully invoked. A new service request will be accepted only if the current utilization of the relevant TT together with the average rate of that service does not exceed $\mathrm{AC}_{\mathrm{th}}$. Finally, the SR parameter, which is in the range $[\mathrm{R}^{a}_{\min}, \mathrm{R}_{\max}]$, adjusts the SRs of active services. The lower the rate, the more the degradation experienced by active services.

A key challenge in the proposed framework is the process of evaluating BIs as this has a number of dependencies including the nature of individual BIs, the network and service measurements, the technology used to deploy the services, the injected traffic, the amount and dynamics of service invocations, and the enforcement of service management policies. The next three sub-sections elaborate on the description, relevance, and most importantly, the methodology to evaluate the three BIs considered in this work: losses due to service invocation rejections (*lossInvRjct*), losses due to performance degradation (*lossSvcDgd*), and service satisfaction (*satisSvc*).

3.2. *Evaluation of losses due to service invocation rejections*

This indicator correlates the losses of a network operator with the rejections of service invocations. When subscribed services are rejected, the operator usually suffers economic penalties. Assigning the highest importance to this BI over others would imply that the operator prioritizes the acceptance of all subscribed services irrespective of the network conditions. This can eventually result in network congestion due to an excess of active services injecting traffic into the network. Congested links can degrade the performance of active services and thus adversely affect the *lossSvcDgd* and *satisSvc* BIs.

The *lossInvRjct* BI tracks the ratio of rejected to accepted services and is influenced by the TCL and AC thresholds. It should be noted that the policies setting these thresholds are executed at the ingress points of each TT, so that the BI is measured while taking into account all TTs in the network and all service invocations in each TT for a period, that is, observation times ranging from minutes to hours.

In order to evaluate this BI, we propose the following methodology, which applies to each TT i in the network:

1. The default behavior of the adopted service management approach considers that when the injected traffic reaches the TCL policy threshold, corrective actions are executed to prevent

network congestion. A default corrective action is the execution of the AC policy, which results in the rejection of new service invocations.

2. When the TCL is crossed upwards and the AC policy is enforced, a count of the number of service rejections is maintained.
3. Service rejections eventually result in the reduction of injected traffic. When the traffic volume falls below the TCL, new invocations are accepted, and the count for rejected services is stopped.
4. The methodology prescribes returning to step ii and continuing the monitoring of the BI until the BI evaluation takes place when the observation time has completed.
5. The following expression quantifies this BI at the evaluation phase:

$$lossInvRjct = \sum_{i=1}^{n} \frac{rjctSvc_{TTi}}{rjctSvc_{TTi} + accSvc_{TTi}} \tag{1}$$

where n is the number of TTs and $rjctSvc_{TTi}$ and $accSvc_{TTi}$ are the total number of rejected and accepted services in the respective TT. The value of the *lossInvRjct* indicator lies between zero and n. A value equal to zero represents that all service invocations are accepted, while a value equal to n would represent that all service invocations are rejected.

3.3. Evaluation of losses due to performance degradation

Congestion is a result of the network resources not being able to accommodate the volume of injected traffic. Under such conditions, performance degradation of active services can occur and subsequently losses for the network operator in the form of penalties because services may not receive their contractual rates. The *lossSvcDgd* indicator correlates these losses with the performance of active services at times of network congestion. Prioritizing this BI over others would result in scenarios where proactive actions prevent network congestion. This would negatively affect the *lossInvRjct* BI as few active services will be maintained because of high invocation rejection rates. In contrast, the BI service satisfaction would be favored as the network would never have congestion, and it would be more probable that the resulting few active services would be fully satisfied most of the time.

This BI is influenced by the TCL and SR policy settings. The evaluation of this BI as a result of enforcing these policies considers a given observation time ranging from minutes to hours. To evaluate this BI, we propose the following methodology, which applies to each TT i in the network:

1. When the volume of traffic is below the TCL, the network is not congested, and the BI is thus not quantified. When the TCL is exceeded, a default corrective action is adopted to adjust the rate of active services by means of the SR policy. However, this can result in active services receiving lower rates than the contracted ones and consequently in service degradation.
2. When congestion occurs, the total number of active services and the volume of injected traffic are quantified at periodic rates (e.g., based on a monitoring interval of >10 s).
3. The number of active services and volume of injected traffic are used to measure the ratio between the average rate of the services currently served and the average SRs for the SLAs allocated in the RAB, which is calculated as follows:

$$lossSvcDgd_{TTi} = \frac{(TR_{in_{TTi}})(numSLAs_{TTi})}{(actSvc_{TTi})(SR_{SLAs_{TTi}})} \tag{2}$$

where $lossSvcDgd_{TTi}$ quantifies the BI in TT i and $TR_{in_{TTi}}$ is the total volume of traffic injected to TT i and $actSvc_{TTi}$ is the number of active SLAs in TT i, both measured at the time of the evaluation. Finally, $numSLAs_{TTi}$ and $SR_{SLAs_{TTi}}$ are the number of SLAs and the sum of the contractual SRs allocated in the RAB of TT i, respectively.

4. The BI is quantified only when the injected traffic is above the TCL. When the volume of injected traffic falls below the TCL, the SR is changed to its original value, and the BI quantification is stopped.

5. The system keeps a record of all BI quantifications. By using all BI evaluation records produced during the simulation (step iv), the average value of $lossSvcDgd_{TTi}$ is computed, which is denoted as $lossSvcDgd_{avgTTi}$. The value of this BI over all TTs in the network is given by

$$lossSvcDgd = n - \sum_{i=1}^{n} lossSvcDgd_{avgTTi} \tag{3}$$

where n is the number of TTs. The value of the *lossSvcDgd* lies between zero and n. A value equal to zero represents that all services have enjoyed at least the average contracted SRs allocated in the RABs of the network during congestion, while a value equal to n would represent that none of the services have enjoyed at least the average contracted SRs allocated in the RABs during congestion as well.

3.4. Evaluation of service satisfaction

Contrary to the *lossSvcDgd* indicator that correlates losses with the performance of active services only during network congestion, the *satisSvc* indicator correlates the business impact of the level of service satisfaction during the whole life cycle of services, namely because service invocations are accepted until their lifetime has ended. Prioritizing this BI over the others would imply that services receive high rates for most of the time at the expense of sacrificing the number of accepted services and hence affecting the *lossInvRjct* BI. In contrast, prioritizing this BI would have a positive effect on the *lossSvcDgd* BI because SRs would rarely be degraded and hence congestion would be highly unlikely.

In order to measure this BI, we propose the following methodology, which applies to each TT i in the network:

1. The BI is quantified during the whole service life cycle, and it is quantified considering two network states: (a) when network resources are not highly utilized and (b) when network resources are heavily utilized.
2. In order to define the state of the network, the methodology considers the percentage of utilization of the shared links supporting a given TT. The link utilization is periodically monitored at intervals of >10 s, which is the granularity of BI evaluations on each TT. The link monitoring functionality on which our methodology relies is supported by the policy-based network simulator described in Section 3.1.
3. Contracted SRs are formalized in SLAs. In the case where services inject traffic at rates lower than the contracted ones, they are considered fully satisfied. When the utilization of all shared links along a given TT is below 85%, we consider that the network is not highly utilized, that services are injecting the desired traffic, and that they are fully satisfied. In these conditions, the BI is assigned with the maximum value when it is evaluated.
4. Services may be prevented from sending traffic at their desired rates because of overutilized network links that are shared between multiple TTs. When the network resources are heavily utilized, the BI is evaluated by considering the ratio between the average rate of the services currently served and the average SRs for the SLAs allocated in the RAB. This ratio is periodically computed, and the results are used for deriving the average value of the BI over a simulation execution.
5. Taking into account all the aforementioned considerations, this BI is evaluated by a combination of the maximum values corresponding to periods of full satisfaction and also when the network is heavily utilized as follows:

$$satisSvc_{TTi} = \begin{cases} 1, & \text{if any shared link's capacity in TT } i < 85\% \\ \frac{(TR_{in_{TTi}})(numSLAs_{TTi})}{(actSvc_{TTi})(SR_{SLAs_{TTi}})}, & \text{if any shared link's capacity in TT } i \geq 85\% \end{cases} \tag{4}$$

where $satisSvc_{TTi}$ quantifies the BI in TT i, $TR_{in_{TTi}}$ is the total volume of traffic injected to TT i, and $actSvc_{TTi}$ is the number of active SLAs in TT i, with the latter two measured at the time of the evaluation. On the other hand, $numSLAs_{TTi}$ and $SR_{SLAs_{TTi}}$ are the number of SLAs and the sum of the contractual SRs allocated in the RAB of TT i, respectively.

6. Taking into account the latter considerations, step (v) certainly implies that BI $satisSvc_{TTi}$ is evaluated periodically until the observation time ends. Considering the number and value of each BI evaluation recorded during the simulation, the average of $satisSvc_{TTi}$ is calculated, and it is denoted as $satisSvc_{avgTTi}$.
7. Finally, the value of this BI over all TTs in the network can be computed by the following expression:

$$satisSvc = n - \sum_{i=1}^{n} satisSvc_{avgTTi} \tag{5}$$

where n is the number of TTs in the network. The value of the *satisSvc* indicator lies between zero and n. A value equal to zero represents that all services have enjoyed at least the average contracted rates allocated in the RABs during the whole service provision. A value of *satisSv* equal to n would represent that none of the services have enjoyed at least the average contracted SRs allocated in the RABs during their provisioning cycle.

4. BUSINESS-DRIVEN OPTIMIZATION SOLUTION

The three BIs are evaluated for a given observation time carried out with a simulation run. Moreover, it is necessary to repeat the simulation iteratively in order to find the most appropriate candidate parameters of the policies to fulfill the desired business goals. This is the ultimate target of the optimization process that we describe hereafter.

4.1. Computational intelligence background

This section gives preliminary information on optimization techniques as background support for the solution proposed in this paper.

4.1.1. Multi-objective problem definitions

Optimization is the procedure of determining and comparing feasible solutions until no better solution can be found [3]. The solutions are considered good or bad in terms of an objective. However, most real-world problems have different objectives that need to be optimized simultaneously. This is also known as a MOO problem or MOP for short. When optimizing this sort of problems, the optimum terms are defined as a set of good trade-off solutions where each one is an efficient solution to the problem. The MOP can be formally stated as follows (we assume that all objectives are equally important, and without loss of generality, we will refer only to minimization problems):

$$\text{Minimize} \qquad F(X) = [f_1(X), f_2(X), \ldots, f_M(X)]^T \tag{6}$$

subject to $\mathbf{X} \in \mathbf{F}$, where $\mathbf{X}$ is a *decision vector* with d *decision variables*, $\mathbf{F(X)}$ is the M-dimensional *objective vector* ($\mathbf{X} \geq 2$), $f_m(\mathbf{X})$ is the m-th objective function ($f_m : \mathbb{R}^d \rightarrow \mathbb{R}$), and $\mathbf{F}$ is the feasible region.

In this case, the goal is to determine, from among all $\mathbf{X} \in \mathbf{F}$, the particular $\mathbf{X}^*$ that yields the optimum value for all the objective functions. However, it is unusual that there is a single solution simultaneously optimizing all the (potentially conflicting) objectives. Instead, there is a need to find a set of *trade-off solutions*.

The most commonly adopted notion of optimality is the so-called Pareto optimality [10], which is based on the *Pareto dominance* relation. The latter is defined as follows: given two solutions $\mathbf{X}, \mathbf{Y} \in \mathbf{F}$, we say that **X** Pareto-dominates **Y**, denoted by $\mathbf{X} \prec \mathbf{Y}$, if and only if

$$\forall m \in \{1, 2, \ldots M\} : f_m(X) \leq f_m(Y) \wedge \tag{7}$$

$$\exists m \in \{1, 2, \ldots M\} : f_m(X) < f_m(Y) \tag{8}$$

otherwise, we say that **Y** is *non-dominated* with respect to **X**. Finally, we say that a point $\mathbf{X}^* \in \mathbf{F}$ is *Pareto optimal* if there is no $\mathbf{X} \in \mathbf{F}$, such that $\mathbf{X} \prec \mathbf{X}^*$. The set of all $\mathbf{X}^* \in \mathbf{F}$ satisfying this condition constitutes the *Pareto optimal set*, whose image in the objective space is called the *Pareto front* or *trade-off surface*.

In the case of the optimization problem presented in this paper, the objective vector **F**(**X**) models the evaluations of the BIs, which are computed with the help of the policy-based network simulations. The *Pareto front* includes all $\mathbf{X}^*$ points where each point gives the values of the decision variables that yield the most optimum values for all BIs. The decision variables model the policy values for the service management policies applying to each TT, whose enforcement would optimize the BIs at run time.

4.1.2. Need for surrogate models

The popularity of MOO has promoted the development of classical techniques from operations research that can produce high-quality solutions under certain conditions [11]. Even though many optimization problems have been successfully solved using classical techniques, many others have not been solved because of the lack of ideal conditions to work properly. Additionally, classical techniques usually produce a single solution, and therefore, it is required to execute the algorithm several times in order to produce the desired number of trade-off solutions, increasing with this the number of fitness function evaluations and thus computational complexity. Moreover, when the search space is large, rugged, and highly restricted and the functions are not differentiable, classical techniques are not effective. EAs are bio-inspired metaheuristics that have been successfully used to solve such problems. The main advantage of these algorithms is their ability to find multiple efficient solutions in one single simulation run. Thus, EAs are an ideal alternative for solving MOPs.

Evolutionary algorithms are population-based techniques that perform a multidimensional search, finding more than one solution within an execution. Furthermore, EAs have been found to locate solutions close to the global optimum even in highly rugged search spaces. Therefore, it is not coincidence that EAs are successfully applied to several MOPs. However, despite their success, when dealing with computationally expensive real-world optimization problems, even these approaches can fail when the number of the fitness function evaluations required to give acceptable results is high and exceeds the available computation time. For this reason, research in EA has proposed several strategies to improve the efficiency of existing approaches. One possible development to achieve this goal is to approximate as much as possible the behavior of such expensive optimization functions, using a fraction of the required computation time. These are called surrogate models (also known as metamodels, emulators, reduced models, approximate models, or response surface models).

A surrogate model is an approximation of a simulation used to construct simpler and lower-computational-cost models; if the original simulation is represented as $f(x)$ and the surrogate model is represented as $f'(x)$, then, $f'(x) = f(x) + e(x)$, where $e(x)$ is the approximated error. The internal behavior of $f(x)$ does not need to be known (or understood); only the input/output behavior is important.[2] A model is constructed based on the response of the simulator to a limited number of intelligently chosen data points. Metamodels generate simple models that capture relations between the relevant information of the input and output variables.

4.2. Surrogate optimization solution

This section describes the algorithmic developments made to address the optimization problem presented in this paper.

[2] *This is known as behavior modeling or black-box modeling.*

The simulation process on which the optimization procedure relies is very time-consuming. BIs are evaluated by considering the results of observations made through network simulations, which can take dozens of minutes per simulation run depending on observation time, the topology, service features, and the computational resources available to execute the simulations.

An appropriate surrogate-based optimization approach that could easily deal with the aforementioned complexity and that can provide efficient results for more than 10 variables has been adopted. The tune-adaptive metamodel-assisted algorithm (TAMAAL) [9] has been applied to solve the policy optimization problem addressed in this paper.

The TAMAAL intersperses evaluations in the metamodel and the evaluation of BIs, and it uses the results of the BI evaluation to feed back the metamodel, thus improving its accuracy. Previous research [12] compared four meta-modeling techniques: radial basis functions [13], support vector regression [14], Kriging design and analysis of computer experiments [15], and polynomial regression [16], according to their accuracy, robustness, efficiency, and scalability with the aim to identify advantages and drawbacks of each meta-modeling technique. This work did not identify a clear winner among the meta-modeling techniques considered, but each metamodel outperformed the others in specific test functions. We used eight scalable unconstrained global multi-objective test function problems from the specialized literature that consider both the number of local minima and the shape of the Pareto front, containing characteristics that are representative of what can be considered 'difficult' in MOO research problems. Therefore, TAMAAL was designed to find the best surrogate approach to be used at a given time for a given problem. This goal is achieved through online adaptation. TAMAAL starts with several meta-modeling techniques and selects the best one according to a metric responsible for measuring the performance of each technique. A block diagram of the TAMAAL approach is shown in Figure 4, and a brief explanation of its main components is provided hereafter.

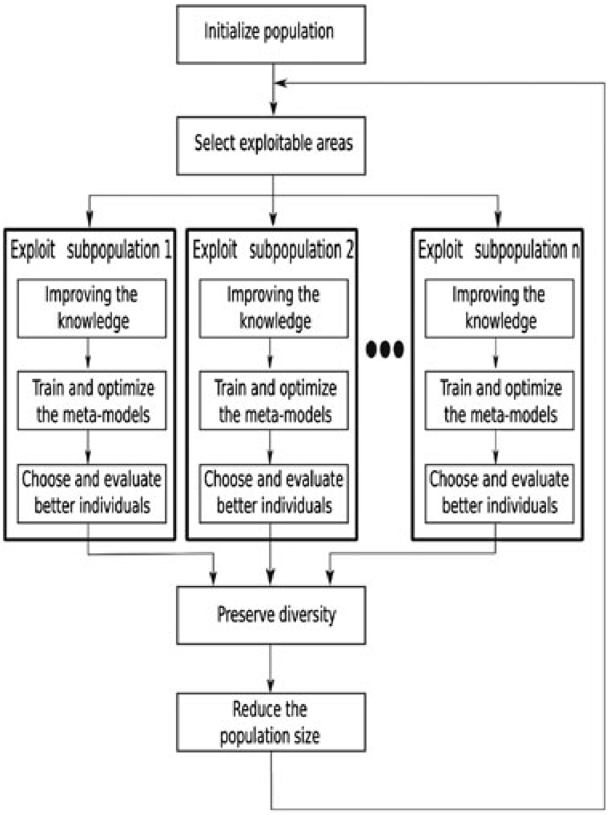
Initialize population
Select exploitable areas
Exploit subpopulation 1
Improving the knowledge
Train and optimize the meta-models
Choose and evaluate better individuals
Exploit subpopulation 2
Improving the knowledge
Train and optimize the meta-models
Choose and evaluate better individuals
Exploit subpopulation n
Improving the knowledge
Train and optimize the meta-models
Choose and evaluate better individuals
Preserve diversity
Reduce the population size

Figure 4. Components of the tune-adaptive metamodel-assisted algorithm approach

4.2.1. Initialization of the population

Like most EAs, TAMAAL initializes its population using randomly generated solutions. It uses the Latin hypercube sampling (LHS) method [17] to achieve this goal, because it is well known that metamodels require a proper distribution of points in order to achieve good accuracy. Therefore, TAMAAL uses the LHS method to create the initial population $\boldsymbol{P}_0$ of size N. In the context of the optimization problem considered in this paper, the population is a random selection of policy values with which the network simulation will be assessed to evaluate the BIs.

4.2.2. Select exploitable areas

The TAMAAL uses a subpopulation scheme in order to guide its effort to promissory regions.[3] This scheme improves the diversity of solutions and prevents the algorithm from becoming trapped in local optima. Also, TAMAAL uses this scheme to search more efficiently, because it assigns different subpopulations to different regions of the search space. To achieve this, TAMAAL clusters the main population in the variable decision space, and if a cluster has promissory solutions, it is selected to become a subpopulation to perform exploitation; otherwise, the cluster is discarded. With the previous procedures, TAMAAL ensures the selection of non-dominated solutions well distributed on the Pareto front.

4.2.3. Exploit a reduced area using a subpopulation

Once the promissory regions are identified, an independent exploitation process is performed on each subpopulation. This process begins by *improving the knowledge* of the subpopulation. The metamodel is subsequently trained and optimized, and the best individuals found by the optimization process will be evaluated using the simulation-based BI evaluation process and added to the main population. For improving the knowledge, TAMAAL uses previously stored solutions that are close to the region of the subpopulation at hand. To achieve this, it encloses the subpopulation using a hypersquare, and then the boundaries of such hypersquare are relaxed by 10%. This operation looks for a wider opening on the search, such that better results are obtained. For this purpose, an external archive that contains solutions evaluated in past generations is used. Then, the subpopulation at hand is filled using the solutions of the external archive enclosed in the hypersquare.

4.2.4. External archive delimitation

The external archive is empty at the beginning of the algorithm execution, and then it is filled with those solutions evaluated so far with the simulation-based BI evaluation process. However, because memory is constrained, the size of the external archive is limited to a maximum value, and the approach retains the closest solution to the centroid of each cluster.

4.2.5. Training and optimizing the metamodels

One meta-modeling approach is created for each objective using the extended subpopulation. Once the meta-modeling is created, a training process starts. After training, the metamodel result is optimized with a MOO algorithm. In this case, the non-dominated sorting genetic algorithm II [18] is used, and it is executed for a few iterations.

4.2.6. Choosing and evaluating the best individuals

After each surrogate model is optimized with the non-dominated sorting genetic algorithm II, TAMAAL selects the best solutions obtained, and it evaluates them with the simulation-based BI evaluation process.

[3] *A promissory region is an area within the search space that could bring the algorithm to an optimal area in terms of the target objectives.*

4.2.7. Preserving diversity

The TAMAAL has been designed to have premature convergence. Therefore, a variation operator is applied. TAMAAL uses a real-parameter mutation operator to mutate the best solutions from the previous step. These solutions are evaluated with the simulation-based BI evaluation process, and they are stored in the external archive.

4.2.8. Reducing the population size

The TAMAAL uses a rank-based approach to maintain a fixed population size of the solutions found.

4.3. Assigning preferences to business indicators

The optimization algorithm outlined previously treats all objectives equally. This means that it produces a *Pareto front* that includes points (solutions), in which each point represents policy values whose enforcement produces optimal results for *all* BIs (objective functions). However, network operators should have the means to prioritize or to assign preferences to the BIs. For this purpose, we have enhanced the previous optimization algorithm with an *a priori* decision-making method in which the network operator would provide knowledge about preferred solutions before the optimization process begins. The method adopted for this work to incorporate the user's preference into TAMAAL is the G-dominance [19]. The resulting approach is called G-TAMAAL.

G-Dominance was proposed by Molina *et al.* [19] as variation of the Pareto dominance in order to incorporate the decision maker's preferences into any meta-heuristic approach. This preference incorporation can follow an *a priori* or interactive approach. G-Dominance refers to a small fitness function modification (placed right after the computation of the fitness function value) that uses penalization as the strategy to assign preferences.

The administrator incorporates its BI preferences for the multi-objective policy optimization process through the use of a scalarizing function based on a reference point (named reference point g) that drives the penalization strategy. Each computed solution is compared with this point in order to evaluate the dominance relationship. The approach penalizes those solutions that are non-dominated with respect to the selected g reference point, or that are dominated by the g reference point (the only solutions that will not be penalized are those that dominate the g reference point). To achieve such a penalization, this approach uses flags that divide the space according to the g reference point. A flag is defined as

$$Flag_g(f) = \begin{cases} 0 & \text{if } f_i \leq g_i \ \forall i = 1, \dots, p \\ 1 & \text{if } g_i \leq f_i \ \forall i = 1, \dots, p \\ 2 & \text{otherwise} \end{cases}$$

where f is a point (in the objective space) and g is a reference point. In this manner, the solutions that satisfy all the aspiration levels are preferred over those partially satisfied solutions. For instance, flags are calculated for all solutions. Solutions with flag equal to 1 and 2 will be penalized.

In Figure 5, we can graphically see the different values obtained by the computation of the flag. In this case, we are interested in those solutions that dominate the g reference point.

It is worth mentioning that the TAMAAL and G-TAMAAL approaches were developed explicitly to handle the complexity of the research problem addressed in this paper.

5. VALIDATION AND ANALYSIS

This section describes the experiments conducted to validate the policy optimization framework. As described earlier, we rely on a policy-driven network simulator based on OPNET to simulate the network and service behaviors and on additional software (*BI evaluation*, *performance measurement processing*, and *EAs*) developed to assess the framework functionality. The developed software can be made available to the interested reader upon request.

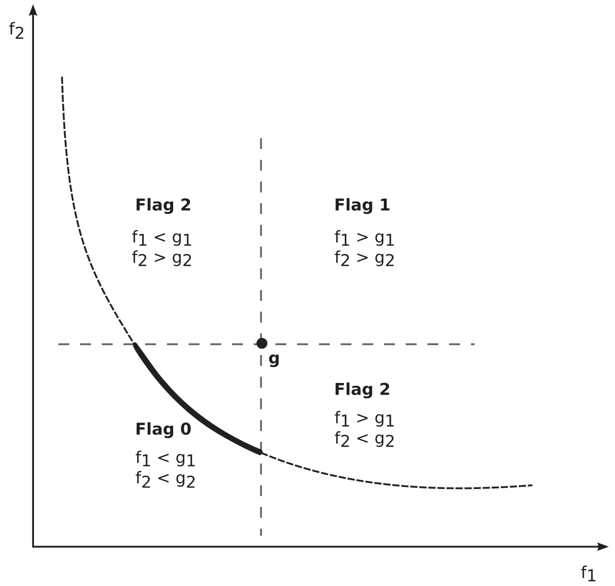

Figure 5. Flags based on a feasible *g* reference point

5.1. Experimentation scenario setup

This section describes the experimentation scenario setup that was used to validate the optimization framework. The setup is configured according to the settings defined by *network topology and features*, *SLA repository*, *RABs*, and *service and invocation patterns*. Moreover, it is worth mentioning that the framework functionality is not restricted to the configurations presented in the following.

5.1.1. Network topology and features

The optimization process is applied to a network infrastructure that consists of 13 nodes and 18 links with the topology shown in Figure 6. The infrastructure has been configured in OPNET to support QoS provisioning in a DiffServ/MPLS-enabled network. The scenario considers six TTs whose ingress/egress points are the following label edge routers (LER): TT1 [LER1–LER5], TT2 [LER2–LER4], TT3 [LER1–LER4], TT4 [LER2–LER6], TT5 [LER3–LER5], and TT6 [LER3–LER6]. The ingress/egress points are configured with MPLS tunnels, and all routers are DiffServ enabled. All links are E1 links with a capacity of 2.048 Mbps.

5.1.2. Service-level agreement repository

The SLA repository contains information about subscribed SLAs. We consider a provider that classifies services as medium (*medQlty*) and high quality (*highQlty*), supporting a total of 30 SLAs from which 15 SLAs are *medQlty* and 15 SLAs are *highQlty* services. The DiffServ QoS classes assigned are AF11 for *medQlty* services and AF41 for *highQlty* services. *medQlty* SLAs are served by TT1, TT2, and TT4, while *highQlty* SLAs are served by TT3, TT5, and TT6 (Figure 6). The SLAs specify an average rate per service based on which a total average rate (TAR) per TT is estimated. The TARs are used for the calculation of the *RAB* described in the following.

5.1.3. Resource availability buffer

In the adopted QoS management approach, the calculation of the RAB is the result of a dimensioning process that considers almost satisfied ($Fctr_{\mathrm{AS}}$) and fully satisfied ($Fctr_{\mathrm{FS}}$) factors and the TAR of the offered services per TT. These factors are used to estimate minimum and maximum traffic demands (TD) per TT, which are in turn calculated with the following expressions:

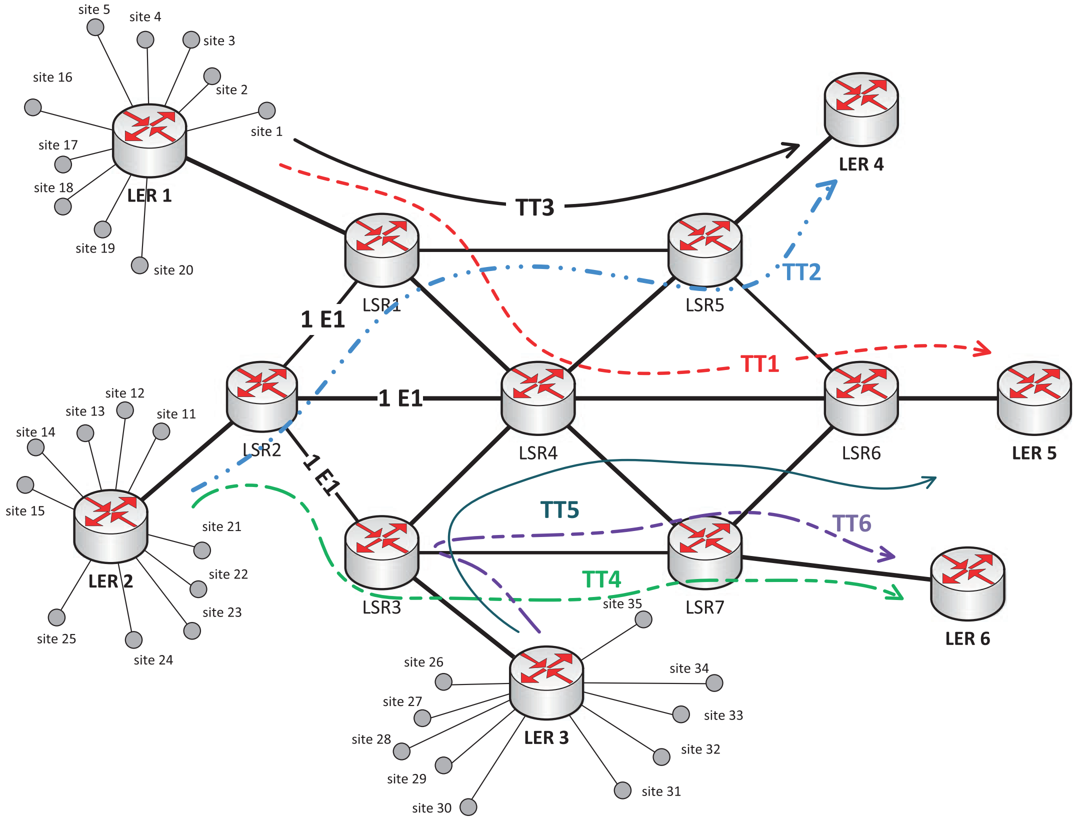

Figure 6. Network topology and traffic trunks

$$\mathrm{TD}_{\mathrm{min}} = (1 - Fctr_{\mathrm{AS}})(TAR) \tag{9}$$

$$\mathrm{TD}_{\mathrm{max}} = (1 + Fctr_{\mathrm{FS}})(TAR) \tag{10}$$

We consider a provider that defines $Fctr_{\mathrm{AS}} = 0.3$ and $Fctr_{\mathrm{FS}} = 0.2$ for *medQlty* services and $Fctr_{\mathrm{AS}} = 0.4$ and $Fctr_{\mathrm{FS}} = 0.3$ for *highQlty* services. $\mathrm{TD}_{\mathrm{min}}$ defines the $\mathrm{R}^{a}_{\mathrm{min}}$ value of the *RAB*. On the other hand, $\mathrm{R}^{w}_{\mathrm{min}}$ is determined by considering the $\mathrm{TD}_{\mathrm{min}}$ values (i.e., $\mathrm{R}^{a}_{\mathrm{min}}$) of the TTs that share physical links. To do this, we first determine the available spare resources after the sum of $\mathrm{R}^{a}_{\mathrm{min}}$ values of the TTs sharing links has been subtracted from the shared link that has the least capacity (the capacity of a shared link is the maximum available for the TTs sharing such link). The spare resources are allocated to the TTs sharing a link proportionally, according to their $\mathrm{TD}_{\mathrm{max}}$ value. The result of this allocation is added to the $\mathrm{R}^{a}_{\mathrm{min}}$ values of the TTs to define the $\mathrm{R}^{w}_{\mathrm{min}}$ values. In the considered topology, there are three shared links: TT2 and TT3 share links LSR1–LSR2, TT1 and TT6 share links LSR2–LSR5, and finally, TT4 and TT5 share links LSR4–LSR7. Finally, $\mathrm{R}_{\mathrm{max}}$ is calculated by subtracting the sum of the $\mathrm{R}^{a}_{\mathrm{min}}$ values of the TTs sharing resources from the capacity of the shared link. Table 3 summarizes the scenario setup.

5.1.4. Service and invocation patterns

The service and invocation patterns define the features of the traffic volume injected into the network. In our scenario, traffic is injected from sites 1 to 30, where each TT serves five sites at a time (e.g., sites 1–5 served by TT1).

We consider bundled FTP, email, and HTTP services that are invoked from each site according to an invocation pattern that consists of an SLA service active time (amount of time a site uses a bundled

Table 3. Summary of experimentation setup

					RAB (Mbps)		
Traffic trunk	Service quality	QoS class	No. of SLAs	TAR per TT (Mbps)	R^{a}_{min}	R^{w}_{min}	R_{max}
TT1	*medQlty*	AF11	5	0.50	0.35	0.959	1.568
TT2	*medQlty*	AF11	5	0.70	0.49	0.984	1.478
TT3	*highQlty*	AF41	5	0.95	0.57	1.064	1.558
TT4	*medQlty*	AF11	5	0.55	0.38	0.991	1.598
TT5	*highQlty*	AF41	5	0.80	0.48	1.089	1.698
TT6	*highQlty*	AF41	5	0.75	0.45	1.056	1.663

QoS, quality of service; RAB, resource availability buffer; SLA, service-level agreement; TAR, total average rate; TT, traffic trunk.

Table 4. Invocation patterns

Traffic trunk	Site service active time (s)	Site invocation time (s)	Minimum invocations per TT	Mean invocations per TT	Maximum invocations per TT
TT1	60	20	5	15.37	20
TT2	65	20	5	18.15	20
TT3	80	30	5	13.7	15
TT4	70	20	5	18.15	20
TT5	85	30	5	15	20
TT6	75	30	5	13.33	20

Table 5. Application configurations

Application	Parameter	Distribution	Value
FTP	Inter-request time	Constant	3 s
	File size	Constant	1.9 MB
E-mail	Inter-arrival time	Poisson	3 s
	E-mail size	Poisson	0.5 MB
HTTP	Page interval	Exponential	3 s
	Object size	Constant	1 MB
	Image	Constant	50 kB
	Large image	Constant	20 kB

service with a given SLA) and an inter-invocation time (mean time between consecutive invocations). We consider an observation time of 5 min and the invocation patterns shown in Table 4. These patterns result in a minimum, maximum, and mean numbers of concurrent service invocations per TT, also shown in Table 4.

Finally, the setup is concluded with the definition of the service characteristics. With this regard, the solution presented in this paper has been evaluated by considering the dynamics of network events and the dynamics of end-user service usage. For testing purposes, the latter have been defined as bundles of FTP, Web browsing, and email applications modeled with the features shown in Table 5. We have selected these TCP-based applications and their features with the aim to evaluate the proposed framework under resource-starving applications. Moreover, the applicability of the solution is by no means limited to these applications. In fact, the solution is robust in that it can consider the dynamics and features of all types of end-user traffic.

5.2. Experimentation results optimizing all business indicators

This section presents the results of the validation scenario targeting the optimization of *all* BIs being treated equally. This represents a situation where the service provider has no specific preference in any

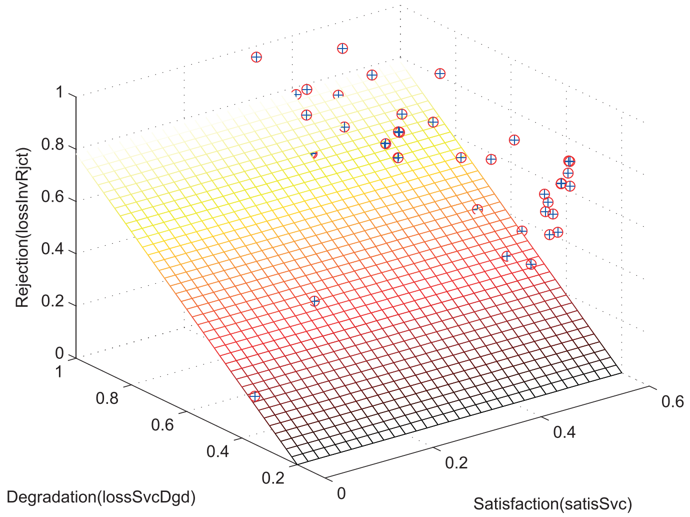

Figure 7. Pareto front points optimizing all business indicators: *satisSvc*, *lossSvcDgd*, and *lossInvRjct*. AC, admission control; SR, service rate; TCL, target critical level

of the three considered BIs. Hence, we run the proposed solution framework with a *preference set* = [0, 0, 0] for the *satisSvc*, *lossSvcDgd*, and *lossInvRjct* BIs, respectively. The result consists of the *Pareto front* with the points defined by the closest BI values for that BI *preference set* = [0, 0, 0]. The Pareto front obtained is shown in Figure 7. It includes the most optimal solutions found by the optimization framework after 31 execution runs, for which, in each execution run, the framework's TAMAAL has performed 1000 evaluations over the real function (OPNET-based simulation), rendering a total of 31,000 simulations.

Specifically, the optimization process produced a total of 460 service management policy sets grouped in 41 Pareto front points shown in Figure 7. This means that points with identical numerical BI values result in different policy sets. Table 6 presents a basic statistical analysis of the 460 policy sets from which two aspects deserve special analysis: (i) the policy sets of the point with the highest statistical mode[4] (point 36) and (ii) the policy set of the *knee point* (point 38), that is, the point with the lower Euclidean distance to the origin (BI *preferences* [0,0,0] in BIs *satisSvc*, *lossSvcDgd*, and *lossInvRjct*). These two aspects are briefly analyzed hereafter.

The Pareto front point with the highest mode (point 36 in Table 6) is statistically representative of the most popular policy sets for the same optimal numerical values of *all* the BIs. Analyzing the 196 policy sets of this point, we found that there is no dominant region for policy values among the derived sets. An example of this behavior is depicted in Figure 8, which shows the distribution of the service management policy values for TT1 of point 36. This makes it evident that the numerical values of the BIs are influenced by the combination of policy values of all TTs in the network, and they do not depend necessarily on closed regions within the RAB of the TTs.

The knee in the Pareto front (point 38) has a Euclidean distance to the origin equal to 0.39836587, and it outperformed the rest of the 460 points. This point has only one set of service management policies (statistical mode equal to 1). From this, we conclude that very popular points in the Pareto front (e.g., those of point 36) do not necessarily coincide with the best alternative to maximize the network business value. The operator would deploy the policy set of the knee point in the real network to optimize equally all the BIs. The associated policy values are shown in Table 7.

5.3. *Experimental results prioritizing business indicators*

This section provides validation results for cases where the network operator wishes to establish preferences among the BIs, that is, not treating all BIs equally, and evaluates the capability of the framework

[4] *The mode is the value that appears most often in a data set.*

Table 6. Pareto front points that optimize all business indicators

	Business Indicator			Point	Mode	Euclidean	Euclidean
Point	satisSvc	lossSvcDgd	lossInvRjct	Mode	Rank	Distance	Dist. Rank
1	0.290468	0.734478	0.89591837	3	20	1.19436147	33
2	0.288287	0.729122	0.99795918	2	24	1.26911419	35
3	0.477503	0.375288	0.42857143	12	9	0.74331936	4
4	0.293806	0.407723	0.89591837	7	14	1.02724375	20
5	0.292344	0.4059	0.99795918	3	21	1.11630747	30
6	0.267896	0.406019	0.96734694	7	15	1.08276488	27
7	0.525888	0.527025	0.69387755	1	30	1.01773258	17
8	0.438534	0.909389	0.79591837	2	25	1.28560743	37
9	0.399932	0.985573	0.78267477	1	31	1.32056031	38
10	0.449641	0.171968	0.75510204	6	16	0.89550495	11
11	0.320812	0.974882	0.97755102	5	17	1.41736419	41
12	0.521338	0.840654	0.71020408	23	3	1.21773655	34
13	0.470607	0.956332	0.93265306	1	32	1.41629219	40
14	0.485808	0.879301	0.86122449	2	26	1.32321097	39
15	0.493302	0.672914	0.78163265	12	10	1.14328899	32
16	0.327497	0.247859	0.95918367	1	33	1.04341827	22
17	0.441313	0.21676	0.5	1	34	0.70124323	3
18	0.32395	0.181786	0.79591837	10	11	0.87833695	10
19	0.293806	0.409832	0.89591837	14	5	1.02808266	21
20	0.268285	0.671039	0.79387755	8	12	1.073551	23
21	0.292344	0.401655	0.99795918	2	27	1.11477096	29
22	0.267896	0.404104	0.96734694	13	7	1.08204825	26
23	0.268285	0.675524	0.79387755	8	13	1.07636011	24
24	0.512869	0.248849	0.75	1	35	0.94205119	13
25	0.496251	0.188862	0.87755102	51	2	1.02568499	19
26	0.506259	0.292169	0.625	2	28	0.85573705	9
27	0.496531	0.191005	0.82978723	1	36	0.98568392	16
28	0.505791	0.295124	0.69047619	2	29	0.90536185	12
29	0.509179	0.271	0.625	1	37	0.85048766	8
30	0.496482	0.215795	0.77777778	4	19	0.94762869	15
31	0.4997	0.191005	0.77777778	1	38	0.9439922	14
32	0.267896	0.407748	0.96734694	3	22	1.08341442	28
33	0.485026	0.335595	0.54081633	5	18	0.8002228	7
34	0.271974	0.561732	0.94897959	3	23	1.13581467	31
35	0.513026	0.68628	0.94897959	13	8	1.2785688	36
36	0.496774	0.229154	0.58367347	196	1	0.79998168	6
37	0.496251	0.184818	0.87755102	16	4	1.02494807	18
38	0	0.353496	0.18367347	1	39	0.39836587	1
39	0.438611	0.272626	0.94897959	14	6	1.08040123	25
40	0.046383	0.229154	0.58367347	1	40	0.62875882	2
41	0.481631	0.229154	0.58367347	1	41	0.79066725	5

to find specific solutions defined by a reference point. The only guideline the network administrator should use to define a reference point lies on setting up values that model tolerance percentage values for each BI, normalized to the number of TTs in the network. Consider that the following tolerance values are defined by the administrator: (i) a maximum of 16% service rejections (tolerance for *lossInvRjct* BI); (ii) no less than 11.6% of services enjoying at least the average contracted SRs allocated in the RABs during congestion (tolerance for *lossSvcDgd* BI); and (iii) a guarantee that no less than 7.6% of services have enjoyed at least the average contracted rates allocated in the RABs during the whole service provision (tolerance for the *satisSvc* BI). The aforementioned percentage values normalized to the six TTs of our running scenario result to the following numerical values for *satisSvc*, *lossSvcDgd*, and *lossInvRjct*, respectively: [0.46, 0.7, 0.98]. With these *BI preferences* and considering the configuration setup described earlier, the framework produced the Pareto front shown in Figure 9,

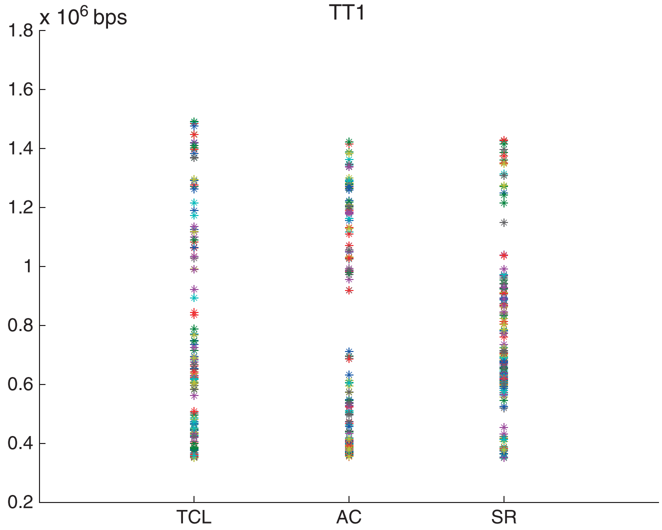

Figure 8. Distribution of policy values for TT1 for point 36

Table 7. Values for policies of the knee point

Policy value	Policy values for TT1–TT6 (bps)					
Target critical level	1,383,730	1,546,095	1,187,704	663,278	1,500,438	805,318
Service rate	1,270,876	729,770	1,024,479	534,309	705,286	1,163,111
Admission control threshold	1,029,852	1,110,586	1,203,230	723,575	1,432,216	489,520

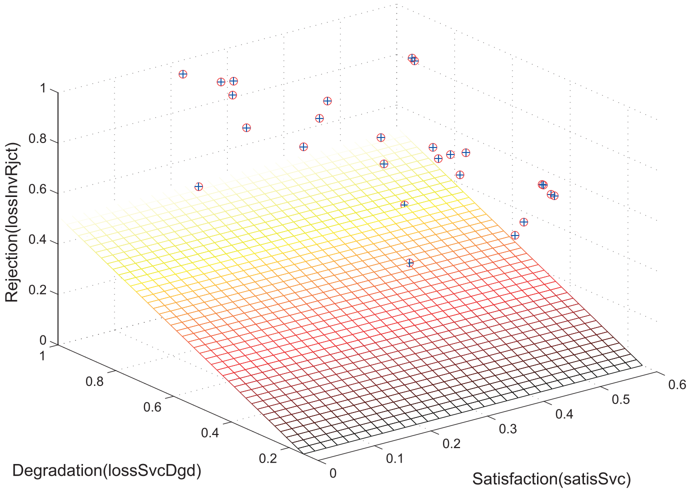

Figure 9. Pareto front points with preferences in the business indicators. AC, admission control; SR, service rate; TCL, target critical level

which plots the most optimal solutions found by the optimization framework, that is, the ones closest to the *BI preferences*.

The optimization process produced 570 service management policy value sets in the 26 Pareto front points shown in Figure 9. Table 8 presents a basic statistical analysis of the 570 policy sets.

The Pareto front point with the highest mode (point 8 in Table 8) produced 153 policy sets. For this solution-specific search, the framework produced policy values falling in dominant regions as

Table 8. Pareto front points with business indicator preferences

	Business Indicator			Point	Mode	Euclidean	Euclidean
Point	satisSvc	lossSvcDgd	lossInvRjct	Mode	Rank	Distance	Dist. Rank
1	0.268285	0.674367	0.79387755	40	4	0.26842738	5
2	0.268726	0.396854	0.86734694	27	7	0.37573122	14
3	0.360379	0.388134	0.83877551	33	6	0.35655168	11
4	0.487694	0.719034	0.98367347	8	9	0.03380451	1
5	0.48281	0.71809	0.99795918	24	8	0.03420638	2
6	0.438534	0.181336	0.79591837	99	3	0.55078052	19
7	0.38811	0.558569	0.55213675	1	15	0.45633087	17
8	0.449641	0.171968	0.75510204	153	1	0.57402456	22
9	0.433105	0.434633	0.79591837	35	5	0.32408183	9
10	0.438534	0.176816	0.79591837	121	2	0.55503904	20
11	0.314494	0.68209	0.94444444	1	16	0.15085411	3
12	0.317137	0.261628	0.94444444	1	17	0.46243275	18
13	0.315811	0.319658	0.94444444	1	18	0.40830712	15
14	0.359354	0.311143	0.81428571	1	19	0.43451192	16
15	0.485026	0.335595	0.54081633	4	10	0.5712264	21
16	0.415086	0.229154	0.58367347	1	20	0.61708017	25
17	0.272331	0.627426	0.92857143	1	21	0.20768134	4
18	0.22817	0.229154	0.58367347	1	22	0.65765935	26
19	0.449641	0.16091	0.75510204	4	11	0.58421266	23
20	0.232188	0.966819	0.50816327	1	23	0.58798009	24
21	0.290038	0.91453	0.73469388	4	12	0.36753952	13
22	0.189468	0.938128	0.99387755	1	24	0.36067311	12
23	0.256971	0.897872	0.89183673	3	13	0.29689538	7
24	0.256971	0.894322	0.94897959	1	25	0.28274384	6
25	0.267896	0.406019	0.96734694	1	26	0.35140984	10
26	0.248465	0.92153	0.93673469	3	14	0.30934525	8

graphically exemplified in the service management policies values for TT1, TT2, and TT3 in Figure 10. This makes it evident that for solutions targeting specific *BI preferences*, it is expected that the most popular solutions will mostly tend towards specific policy values. The knee point has eight service management policy configuration sets. The result of simple statistical analysis of these sets is shown in Table 9, which includes the configurations that have the highest statistical mode per TT and policy value (grey-shaded configurations in Table 9). From this, we conclude that for solutions targeting specific *BI preferences*, it is also expected that the knee point has dominant solutions that tend towards specific policy values. The operator would deploy the policy set 6 of the knee point in the real network to reach the aspiration levels in the *BI preferences*.

6. RELATED WORK

The work presented in this paper is related to relevant works in the areas of network resource management, policy-based management, business-driven management, and computational intelligence's MOO. A non-exhaustive list of related works in the aforementioned areas is briefly described in this section.

Traditional practices for managing network resources are based on some optimization objective such as minimizing the maximum link utilization or the network cost [20] or load balancing optimization [21]. These do not take into account higher-level business objectives and can thus result to network configurations that are not profitable. The performance of services offered by ISPs is an important factor that reflects on the reputation of providers and subsequently on the number of subscriptions they can attract, and also on potential penalties that can apply [22]. The previous work of Xiao and Boutaba [23] investigated models that link network service performance, customer behavior, and market dynamics to profit, while Duan *et al*. [24] evaluate revenue generation as a function of link traffic volume, service

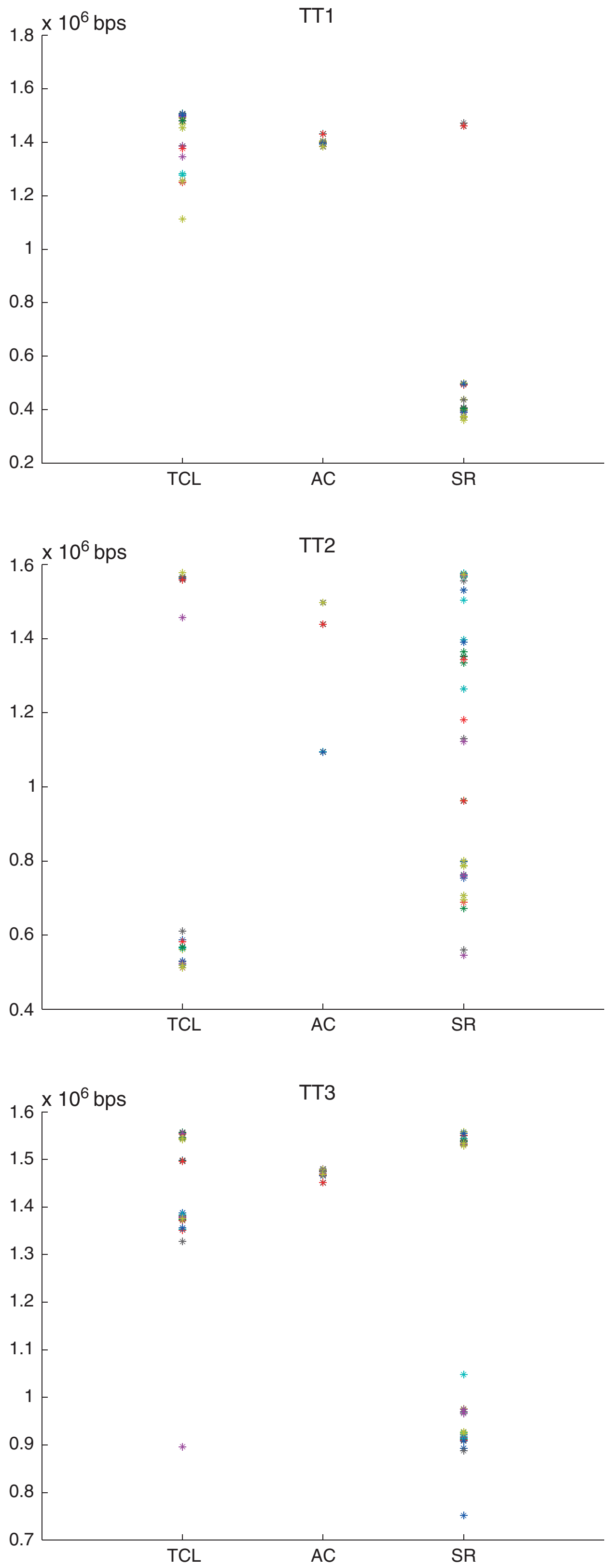

Figure 10. Distribution of policy values for TT1, TT2, and TT3 for point 8. AC, admission control; SR, service rate; TCL, target critical level

Table 9. Service management policies of the knee point

		TT1 (bps)			TT2 (bps)			TT3 (bps)			TT4 (bps)			TT5 (bps)			TT6 (bps)		Freq.
Set	TLC	AC	SR	TLC	AC	SR	TLC	AC	SR	TLC	AC	SR	TLC	AC	SR	TLC	AC	SR	**Mode**
1	1482964.88	420457.46	512684.96	1460229.48	742742.42	1549429.14	1402903.89	1457359.75	1393002.01	1224611.41	1272669.84	742357.76	814910.42	1493969.77	1377871.65	1255333.95	1378697.97	1516656.18	9
2	1478839.23	418823.87	512617.27	1456380.7	748978.65	1544847.96	1392398.59	1457539.8	1390876.49	1225193.6	1257731.23	741938.38	811114.47	1489229.95	1371422.9	1246724.78	1367112.11	1495128.63	0
3	1482823.09	420457.46	512684.96	1454714.45	742742.42	1542728.86	1400932.31	1457359.75	1393002.01	1224598.61	1267649.36	742357.76	813920.83	1492410.06	1377871.65	1254963.35	1378680.36	1504868.6	15
4	1479811.27	420670.96	514285.48	1479074.47	748500.14	1542322.48	1392017.69	1457984.61	1411428.75	1289051.53	1274619.89	742106.7	811023.82	1490622.97	1385921.39	1242502.8	1371372.82	1496508.55	0
5	1479529.99	420670.96	514285.48	1485794.75	748776.75	1543248.51	1396373.09	1457856.86	1394503.32	1376637.61	1292805.95	742111.11	811022.54	1490552.84	1378152.74	1253817.44	1371372.82	1503053.35	1
6	1482823.09	420457.46	512684.96	1454714.45	742742.42	1542728.86	1400932.31	1457359.75	1393002.01	1224598.61	1267649.36	742357.76	813920.83	1493925.67	1377871.65	1255333.95	1378680.36	1516656.18	**17**
7	1478856.89	418823.87	510270.66	1428388.68	748978.65	1535305.36	1391780.6	1456776.67	1390308.84	1224763.74	1257731.23	739699.08	811128.07	1492359.52	1371422.9	1246688.81	1367051.1	1495559.97	0
8	1479855.14	418875.62	510270.66	1428028.32	748869.45	1535605.88	1391783.17	1458339.69	1391351.7	1224827.54	1288529.34	739696.84	810549.09	1492493.59	1382127.1	1243398.84	1372126.64	1495559.97	0

access time, and QoS levels. These focus on optimizing individual BIs (revenue/profit) in isolation, whereas our approach considers multiple objectives jointly when producing network configurations.

Policy-based management has been applied in various application domains to guide the configuration of resources. Flegkas *et al.* [25] focused on network dimensioning policies, while a comprehensive policy set covering traffic engineering and service management operations for IP networks was presented by Charalambides *et al.* [26]. Closer to our research is the work on policy refinement, which is the process of translating high-level objectives to low-level configuration policies. Representative examples include the works of Rubio-Loyola *et al.* [27,28] and Bandara *et al.* [29], which developed refinement approaches for the QoS management domain. These, however, are limited to refining technical objectives, such as managing traffic fluctuations or minimizing the number of SLA violations, and they do not consider optimization of higher-level business objectives. The policy continuum and a policy authoring process are described by Davy *et al.* [30], who propose an approach to assess policy analysis of candidate policies (either newly created or modified) with already deployed policies so that potential policy conflicts are fed back to the policy author. The approach is mostly oriented to detect conflicts and policy consistency, and it is not aimed at maximizing the business value of the underlying infrastructure as considered in this paper. A business-driven SLA refinement and optimization into low-level policy configuration are presented by Aib and Boutaba [31], with a case study on incident management in utility data centers. Only raw monetary profit was used as a BI, as opposed to the wider range considered in this paper.

A popular application domain of business-driven management is that of IT systems. Tosic [32] and Lu and Tosic [33] considered multiple BIs to guide the adaptation of Web service compositions and have also developed a policy language to express business value metrics [34]. Apart from the difficulty in assigning values to these metrics, a simplified method for determining the combined business value (monetary) may result to sub-optimal adaptation decisions. The same drawback is shared by the solution proposed by Bartolini and Salle [35], which dealt with the prioritization of service incidents according to their business impact. The total impact of these incidents is based on a linear relationship with the various BIs. Li and Zhan [36] use machine learning to optimize change management processes. They evaluate the optimality of the resulting decisions according to two BIs, but these are not used to drive the configuration. BIs were also used by Bartolini *et al.* [37] to assess the performance of IT support organization configurations by acting as constraints to define the space of possible solutions. The business impact of service component placement reconfigurations in federated cloud environments was part of an optimization process proposed by Foschini and Tortonesi [38]. The impact model used is rather simple and mainly considers service violation penalties. The finding of price and QoS trade-offs among service providers in competitive environments is analyzed by Maillé *et al.* [39], where decision dependencies and equilibrium compromises are studied by an induction method. The scenarios analyzed however treat the optimization of price settings in isolated scenarios like price per congestion, price per flow unit, and price per flow distribution, contrary to the global MOP addressed in this paper that considers the service life cycle and the dynamic network conditions.

Multi-objective optimization with surrogate models is not new; a wide variety of techniques that use them have been proposed in the past [5–8]. The *ParEGO* [7] has been the most notorious metamodel-based evolutionary multi-objective approach to date. ParEGO uses the Kriging design and analysis of computer experiments metamodel construction technique to model the objective functions. To create the Pareto front, ParEGO uses a series of vectors to assign weighted values to the objective functions according to their importance. At each iteration, a new candidate point is determined by a procedure called *expected improvement maximization*. Although ParEGO has been well tested in different scenarios, its performance decreases rapidly as the number of variables rises [6,9]. Results indicate that ParEGO can significantly reduce its performance when the number of function valuations has more variables. This is a drawback inherited from the Kriging design and analysis of computer experiments approach. Unfortunately, ParEGO cannot replace this approach. The most notorious advance to the state of the art of the work presented in this paper is the TAMAAL approach. TAMAAL is a surrogate model that intersperses evaluations in the metamodel and the simulation-based evaluation to improve its accuracy. The core difference of TAMAAL with the state of the art is that TAMAAL adapts its behavior to follow the best alternative of four meta-modeling techniques on the fly, according to run-time conditions. TAMAAL defines internally a metamodel performance metric based on accuracy, robustness, efficiency, and scalability. This approach is ideal to address the optimization problem targeted in this research, which is computationally expensive and has more than 10 variables.

7. CONCLUDING REMARKS AND FUTURE WORK

This paper described an offline optimization framework that determines service management policy configurations that optimize the business value of the network infrastructure. The approach relies on the representation of business strategies as BIs that have been modeled as objective functions of measurable parameters of the network infrastructure. The paper contributes to the state of the art with the development of a concrete methodology to quantify the objective functions in a business-driven management approach that can effectively drive the optimization process. To quantify BIs, we acknowledge the importance of considering dynamic network events, service usage, delivered QoS, and the relations of these with management functions in the network infrastructure.

Another important gap addressed in the paper is the lack of effective optimization methods for the multi-objective and computationally expensive optimization problem targeted by this research. In this regard, we developed an appropriate surrogate-based optimization approach, namely TAMAAL, which effectively dealt with the complexity of the target problem and generated solutions for cases with which current solutions cannot cope and are not efficient, that is, for a large number of function evaluations.

The solution presented in this paper was fully implemented in software components to assess the framework functionality. The practicality and effectiveness of our approach were demonstrated with the execution of an illustrative scenario that showed the robustness of the proposed concepts. Statistical analysis demonstrated that the proposed framework can be an efficient and powerful tool to improve the business value of network infrastructures. To the best of our knowledge, a holistic solution like the one presented in this paper has not been described in the literature, and as such, it represents an important contribution of the paper.

The evaluation results demonstrate the framework's feasibility in maximizing the business value of a network infrastructure. The approach has been designed as an offline process that could assist operators to make better decisions on how to manage the network infrastructure according to business-oriented guidelines. In this paper, our approach has been developed and validated in the context of a DiffServ QoS management; however, the framework could be applied in other domains, for example, cloud infrastructures, where configuration parameters need to be optimized from a business viewpoint. The challenging issues in adapting our framework to other domains are the formal considerations needed to associate the managed system performance and customer behavior with concrete business aspects of the target domain. Future work will be directed to explore this research area.

We developed and validated a framework to determine policy configurations that find trade-offs among business objectives by taking into account a MOO approach. Future work involves the study of strategies to maximize business value and the quality of provided services under different application domains like software-defined networks and network function virtualization. In particular, multi-objective techniques can be applied in this domain to find trade-offs among contradictory objectives like energy saving, QoS provision, and resource utilization.

The BIs considered in this work are typical of an access network's operator. Future work would be directed to find a trade-off between self-organizing networks) Long-term Evolution functions, which by nature inherently have conflicts due to incompatibility of their management actions.

Other areas that will be targeted are the optimization of pricing strategies and customer frustration reactions to service degradation. We expect these two important aspects to have a direct impact on service usage, service subscription, and invocation for network and service providers. This may possibly require the elaboration of more complex business strategies and possibly the involvement of realistic models of customer reactions to price and service disruptions, which have been left out of this paper.

ACKNOWLEDGEMENTS

This work is supported in part by the European Union IST-FLAMINGO Network of Excellence (#318488) and the project TEC2012-38574-C02-02 from Ministerio de Economa y Competitividad. A special acknowledgment is given to OPNET's University Program for their support.

REFERENCES

1. Mykoniati E, Charalampous C, Georgatsos P, Damilatis T, Goderis D, Trimintzios P, Pavlou G, Griffin D. Admission control for providing QoS in IP DiffServ networks: the TEQUILA approach, *IEEE Communication Magazine* 2003; **41**(1).
2. Rubio-Loyola J, Charalambides M, Aib I, Serrat J, Pavlou G, Boutabam R. Business-driven management of differentiated services. In *IEEE/IFIP Network Operations and Management Symposium (NOMS),* Osaka, Japan, 2010; 240–247.
3. Kalyanmoy D. *Multi-objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons: Chichester, UK, 2001. ISBN 0-471-87339-X.
4. Coello Coello CA, Van Veldhuizen DA, Lamont GB. *Evolutionary Algorithms for Solving Multi-objective Problems*, Kluwer Academic Publishers: New York, March 2002. ISBN 0-3064-6762-3.
5. Voutchkov I, Keane AJ. Multiobjective optimization using surrogates. In *Adaptive Computing in Design and Manufacture. Proceedings of the Seventh International Conference*, Parmee IC (ed.) The Institute for People-centered Computation (IP-CC): Bristol, UK, April 2006, pp. 167–175.
6. Emmerich MTM, Giannakoglou KC, Naujoks B. Single and multiobjective evolutionary optimization assisted by Gaussian random field metamodels, *IEEE Transactions on Evolutionary Computation* 2006; **10**(4): 421–439.
7. Knowles J. ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems, *IEEE Transactions on Evolutionary Computation* 2006; **10**(1): 50—66.
8. Chafekar D, Shi L, Rasheed K, Xuan J. Multiobjective GA optimization using reduced models, *IEEE Transactions on Systems Man and Cybernetics Part C, Applications and Reviews* 2005; **35**(2): 261—265.
9. Montemayor G. On the use of surrogated models in multiobjective evolutionary algorithms. *Masters Thesis*, Center for Research and Advanced Studies of the National Polytechnical Polytechnic Institute (CINVESTAV), Parque Científico y Tecnológico TECNOTAM – Km. 5.5 carretera Cd. Victoria-Soto La Marina C.P. 87130 Cd. Victoria, Tamaulipas, México, December 2011.
10. Pareto V. *Cours D'Economie Politique*, F. Rouge, Lausanne: Switzerland, 1896.
11. Miettinen KM. *Nonlinear Multiobjective Optimization*, Kluwer Academics Publishers: Boston, Massachusetts, 1999.
12. Montemayor-García G, Toscano-Pulido G. A study of surrogate models for their use in multiobjective evolutionary algorithms. In *2011 8th International Conference on Electrical Engineering, Computing Science, and Automatic Control (CCE'2011),* IEEE Press, Mérida, Yucatán, México, October 2011; 1–6.
13. Hardy R. Multiquadric equations of topography and other irregular surfaces, *Journal of Geophysical Research* 1971; **76**: 1905–1915.
14. Vapnik V. *Statistical Learning Theory.* Wiley-Interscience: New York, NY, September 1998.
15. Sacks J, Welch W, Mitchell T, Wynn H. Design and analysis of computer experiments, *Statistical Science* 1989; **4**(4): 409–423.
16. Myers RH. *Classical and Modern Regression with Applications*, 2nd edition, Duxbury/Thomson Learning: California, 1990. ISBN: 0534380166.
17. McKay MD, Beckman RJ, Conover WJ. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 1979; **21**(2): 239–245.
18. Deb K, Agrawal S, Pratap A, Meyarivan T. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *KanGAL Report 200001*, Indian Institute of Technology: Kanpur, India, 2000.
19. Julián Molina J, Santana LV, Hernández-D'iaz AG, Coello Coello CA, Caballero R. g-Dominance: reference point based dominance for multiobjective metaheuristics, *European Journal of Operational Research* 2009; **197**(2): 685–692.
20. Fortz B, Rexford J, Thorup M. Traffic engineering with traditional IP routing protocols, *IEEE Communications Magazine* 2002; **40**(10): 118–24.
21. Halinger G, Nunzi G, Meirosu C, Fan C, Andersen FU. Traffic engineering supported by inherent network management: analysis of resource efficiency and cost saving potential, *John Wiley & Sons, International Journal of Network Management* 2011; **21**(1): 45–64.
22. Wang Y, Lo H-P, Yang Y. An integrated framework for service quality, customer value, satisfaction: evidence from Chinas telecommunication industry, *Information Systems Frontiers* 2004; **6**(4).
23. Xiao J, Boutaba R. Assessing network service profitability: modeling from market science perspective, *IEEE TON* 2007; **15**(6): 1307–1320.
24. Duan Z, Zhang Z, Hou Y. Service overlay networks: SLAs, QoS, and bandwidth provisioning, *IEEE/ACM Transactions on Networking* 2003; **11**(6): 870–883.
25. Flegkas P, Trimintzios P, Pavlou G. A policy-based quality of service management architecture for IP DiffServ networks, *IEEE Network* 2002; **16**(2): 50–56.
26. Charalambides M, Flegkas P, Pavlou G, Rubio-Loyola J, Bandara AK, Lupu EC, Russo A, Dulay N, Sloman M. Policy conflict analysis for DiffServ quality of service management, *IEEE Transactions on Network and Service Management (TNSM)* 2009; **6**(1): 15–30.
27. Rubio-Loyola J, Serrat J, Charalambides M, Flegkas P, Pavlou G. A methodological approach towards the refinement problem in policy-based management systems, *IEEE Communications Magazine* October 2006; **44**(10): 60–68.
28. Rubio-Loyola J, Serrat J, Charalambides M, Flegkas P, Pavlou G. A functional solution for goal-oriented policy refinement. In *Seventh IEEE International Workshop on Policies for Distributed Systems and Networks,* Ontario Canada, 2006; 133–142.
29. Bandara A, Lupu EC, Russo A, Dulay N, Sloman M, Flegkas P, Charalambides M, Pavlou G. Policy refinement for IP differentiated services quality of service management, *IEEE Transactions on Network and Service Management (TNSM)* 2006; **2**: 2–13.

30. Davy S, Jennings B, Strassner J. The policy continuum policy authoring and conflict analysis, *Computer Communications* 2008; **31**(13): 2981–2995.
31. Aib I, Boutaba R. On leveraging policy-based management for maximizing business profit, *IEEE Transactions on Network and Service Management (TNSM)* December 2007; **4**(3): 25–39.
32. Tosic V. Autonomic business-driven dynamic adaptation of service-oriented systems and the ws-policy4masc support for such adaptation, *International Journal of Systems and Service-Oriented Engineering* January 2010; **1**(1): 79–95.
33. Lu Q, Tosic V. Support for concurrent adaptation of multiple Web service compositions to maximize business metrics. In *Proceedings of IFIP/IEEE International Symposium on Integrated Network Management (IM),* Dublin, Ireland, May 2011; 241–248.
34. Lu Q, Tosic V. MiniMASC: a framework for diverse autonomic adaptations of Web service compositions. In *Proceedings of IEEE Symposia and Workshops at UIC/ATC,* Xian, China, 2010; 241–248.
35. Bartolini C, Salle M. Business driven prioritization of service incidents. In *Proceedings of 15th IFIP/IEEE DSOM,* Davis, CA, USA, 2004; 64–75.
36. Li H, Zhan Z. Business-driven automatic IT change management based on machine learning, In *Proceedings of IEEE/IFIP Network Operations and Management Symposium (NOMS),* Maui, HI, USA, April 2012; 1374–1377.
37. Bartolini C, Stefanelli C, Tortonesi M. Potential benefits and challenges of closed-loop optimization processes for IT support organizations. In *Proceedings of IEEE/IFIP Network Operations and Management Symposium (NOMS),* Maui, HI, USA, April 2012; 1392–1398.
38. Foschini L, Tortonesi M. Adaptive and business-driven service placement in federated cloud computing environments. In *Proceedings of IFIP/IEEE International Symposium on Integrated Network Management (IM),* Ghent Belgium, May 2013; 1245–1251.
39. Maillé P, Tuffin B, Vigne JM. Technological investment games among wireless telecommunications service providers, *John Wiley & Sons, International Journal of Network Management* 2011; **21**(1): 65–82.

AUTHORS' BIOGRAPHIES

Javier Rubio-Loyola is a research scientist at CINVESTAV Tamaulipas, México. He holds an Engineering degree in Communications and Electronics and an MSc degree in Digital Systems, both from Instituto Politécnico Nacional of México and a PhD in Telecommunications from Universitat Politècnica de Catalunya in Barcelona. He has participated in a number of Spanish and IST-European research projects mainly in the network management area. His research interests focus on network management, autonomic networking, and service engineering.

Gregorio Toscano-Pulido is a researcher at CINVESTAV Tamaulipas, México. He holds a PhD degree in Computer Science from Cinvestav México. His main research area is on multi-objective evolutionary computing.

Marinos Charalambides is a senior researcher at the University College London. He received a BEng (First Class Hons.) in Electronic and Electrical Engineering, an MSc (Distinction) in Communications Networks and Software, and a PhD in Policy-based Management, all from the University of Surrey, UK, in 2001, 2002, and 2009, respectively. He has been working in a number of European and UK national projects since 2005, and his research interests include software-defined networking, policy-based management, in-network caching, and online traffic engineering.

Marisol Magaña-Aguilar holds an Engineering degree in Computer Science from the Technical Institute of Morelia and an MSc degree in Computer Science from the Centre for Research and Advanced Studies (CINVESTAV) México. Her research interests are on computer networking topics.

Joan Serrat-Fernández received his degree of Telecommunication Engineer in 1977 and a Doctor degree in Telecommunication Engineering in 1983, both from Universitat Politècnica de Catalunya -UPC-. Currently, he is a full professor at UPC where he has been involved in several collaborative projects with different European research groups, both through bilateral agreements or through participation in European funded projects. His topics of interest are in the field of autonomic networking and service and network management.

George Pavlou is professor of Communication Networks in the Department of Electronic and Electrical Engineering, University College London, UK, where he co-ordinates research activities in networking and network management. He received a Diploma in Engineering from the National Technical University of Athens, Greece, and MSc and PhD degrees in Computer Science from the University College London, UK. His research interests focus on networking and network management, including aspects such as traffic engineering, quality of service management, policy-based systems, autonomic networking, information-centric networking, and software-defined networks. He has been instrumental in a number of European and UK research projects that produced significant results with real-world uptake and has contributed to standardization activities in ISO, ITU-T, and IETF. He has been the technical program chair of several conferences, and in 2011, he received the Daniel Stokesbury award for ‘distinguished technical contribution to the growth of the network management field’.

Hiram Galeana-Zapién received his Electronics and Telecommunications Engineering degree from the Autonomous University of Nuevo Leon, Mexico, in 2001, and his PhD degree in Telecommunications Engineering from the Technical University of Catalonia (UPC), Barcelona, in 2011. From February 2001 to September 2004, he worked as a network provisioning engineer for Avantel (today Axtel) in Mexico. In October 2004, he joined the Mobile Communications Research Group at the Department of Signal Theory and Communications, UPC, where he obtained his PhD. Since June 2011, he has been with the Information Technology Laboratory at Cinvestav-Tamaulipas, where he is currently an assistant professor. His research interests mainly focus on mobile radio communication systems, with special emphasis on radio resource management, self-organizing networks, and mobile phone sensing applications for location-based services.