

Multi-Objective Optimization in Dynamic Content Adaptation of Slide Documents

Habib Louafi, *Member, IEEE*, and Stéphane Coulombe, *Senior Member, IEEE*,
and Mohamed Cheriet, *Senior Member, IEEE*,

Abstract—In mobile Web conferencing, slide decks should be optimized before delivery to meet the constraints and environments of target mobile devices. To deliver optimally adapted slides, a trade-off between the visual aspect and delivery time must be reached. Static adaptation methods are CPU-intensive, and require large storage space. The dynamic approach is attractive as the optimal version is created on the fly when the actual slide is to be shared. Existing dynamic solutions are optimized for the resolution of the target mobile device and use good visual quality settings. However, they do not control the resulting data size, which creates serious usability issues, such as increasing the delivery time. Prediction-based methods require much less memory and processing resources than static approaches while yielding an excellent user experience. In this paper, we propose a multi-objective dynamic content adaptation framework, in which we maximize the visual quality and minimize the delivery time simultaneously. We compare our solution with an ideal optimal point, called utopia, and with all the optimal solutions (Pareto front) provided by a static exhaustive system. The obtained results show that our framework yields solutions very close to the utopia and, for the majority of the documents tested, the obtained solutions are on the Pareto front.

Index Terms—Dynamic content adaptation, multi-objective, optimization, quality of experience, QoE, visual quality, delivery time, JPEG, XHTML.

1 INTRODUCTION

MOBILE devices are ubiquitously used in many contexts and targeted by many applications, ranging from sophisticated applications relying on and taking advantage of powerful cloud resources, to simple and standalone ones [1], [2]. Due to the constrained nature of mobile devices (e.g., limited memory, small resolution, limited connectivity), the content they consume must be adapted to their characteristics and environment [3]–[5]. Furthermore, for certain types of applications, the adaptation itself should be achieved in a timely manner, which could be very challenging. For instance, in mobile Web conferencing, a timely (or on-the-fly) adaptation is sometimes required due, for example, to a lack of idle time to perform the adaptation.

In Web conferencing, slide decks should be shared synchronously between the host and all participants. One may consider sending the entire presentation (e.g., PowerPoint) to the mobile user before the meeting. However, many mobile devices may not support that format, or don't have sufficient memory space to store the whole presentation. More importantly, they

won't be synchronized with the host and lose track of which slide is currently presented. The Web represents a good alternative to facilitate synchronization via Web browsers, which are already installed.

When mobile devices are involved in such applications, slides must be adapted before delivery in order to meet their constraints (e.g., memory, supported formats), network conditions (e.g., bitrate and latency), but also to provide the mobile user with the best experience possible [5]–[8]. Furthermore, in certain situations, the adaptation must be achieved on the fly, while the end-user is still waiting on the line. In such a case, the problem of computing the optimal transcoding parameters and using them to create the optimal adapted slide becomes very challenging.

In this paper, we extend the work by Louafi et al. [5], in which the problem of computing the optimal adapted slide is formulated by a scalarization-based optimization method (single-objective optimization). They evaluated the quality of each adapted slide as the product of its visual quality and delivery time. However, the problem we are facing is, in its nature, a multi-objective problem that can be solved by various techniques, and the scalarization is only one way in this domain. Therefore, we propose a general framework, in which the problem of computing the optimal adapted content is formulated as a multi-objective optimization problem. Indeed, instead of relying on a QoE measure formulated as a product of visual and transport qualities, we formulate the problem as simultaneously maximizing the visual quality and minimizing the delivery time. The two functions

- H. Louafi and M. Cheriet are with Synchromedia Lab., École de technologie supérieure, Université du Québec, 1100 Notre Dame Street West, Montreal, H3C 1K3, Quebec, Canada.
E-mail: habib.louafi.1@ens.etsmtl.ca; mohamed.cheriet@etsmtl.ca
- S. Coulombe is with the Department of Software and IT Engineering, École de technologie supérieure, Université du Québec, 1100 Notre Dame Street West, Montreal, H3C 1K3, Quebec, Canada.
E-mail: stephane.coulombe@etsmtl.ca

are conflicting, that is, maximizing (minimizing) one function simultaneously maximizes (minimizes) the other one. This makes the problem very challenging. More importantly, instead of providing the end-user with a single optimal solution, with a multi-objective formulation, a set of optimal solutions (those lying on the Pareto-front) are generated, from which the end-user can select the one that matches his preferences.

We compare our solution with an ultimate optimal point, also called utopia, and with all the optimal solutions (Pareto front) provided by a static exhaustive system. Further, we compare the performance of our framework with that proposed in [5]. The obtained results show that the new proposed framework yields optimal solutions much closer to utopia than those provided by the framework of [5], and for the majority of the documents tested, the obtained optimal solutions lie on the Pareto front.

The paper is organized as follows: In section 2, we review several content adaptation methods proposed in the literature. In section 3, we formulate the problem of computing the optimal adapted content. In section 4, we review some notions from the multi-objective optimization field and their application to our problem. In sections 5 and 6, we show how we propose to evaluate and estimate the visual quality and delivery time, respectively. In sections 7 and 8, we present the experimental setup used to validate the proposed framework as well as the obtained results. Lastly, section 9 concludes the paper.

2 RELATED WORK

To deliver the optimal adapted slide, the obvious solution consists in creating multiple versions of each slide before the meeting, and when a new slide is to be presented on the host side, the system selects and delivers the best one, evaluated under a certain quality criterion. This technique is well-known as static content adaptation [3], [9]. Fig. 1 illustrates the process of delivering the optimal adapted content using a static strategy. It presents two major drawbacks. First, it requires a large storage space to store all the adapted versions of each slide of each presentation. Depending on the nature of the connected mobile devices, not all the created versions will likely be used. Second, it is not always possible to have the time required to create all these versions. Moreover, the optimal version may not be part of the set of created versions, and in this case a lower quality version is delivered instead. This aspect is related to the granularity level of the static content adaptation method (how many versions are created). The higher the number of versions, the better the solution but the higher the storage and the computational complexity. Note that human supervision may also be required to filter the created versions in order to eliminate the non usable ones.

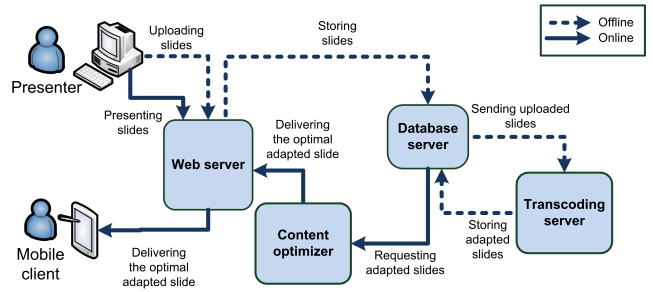


Fig. 1. Static content adaptation architecture. The dashed lines represent the operations performed offline (during the idle time), and the solid ones are performed online (during the presentation).

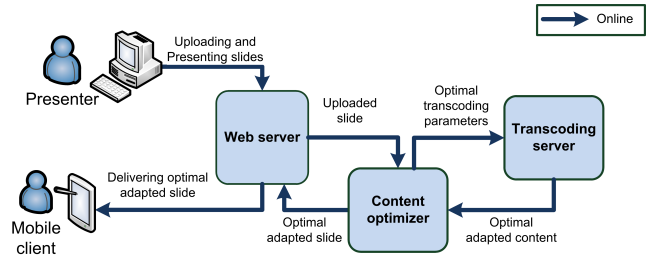


Fig. 2. Dynamic content adaptation architecture. All the operations are performed online (during the presentation).

The dynamic content adaptation strategy is very attractive, as the content is adapted on the fly, without creating any version in advance or requiring storage. In the mobile Web conferencing context, when a new slide is to be shared, an adapted version is created for each mobile device. Fig. 2 shows how the content can be optimized and delivered with a dynamic strategy. Existing dynamic solutions create an adapted version that fits the resolution of the target mobile device and has good visual quality [10]. For instance, JPEG images could be transcoded using the resolution of the target mobile device and a quality factor value of around 80. However, there is no control over the resulting data size of the adapted content, which creates serious usability problems. The adapted content could therefore be rejected by the mobile device due to a lack of memory [11], [12]. Even when the adapted content satisfies the memory constraint, there is no guarantee that it will provide an adequate end-user experience, which is affected, among other things, by the visual aspect of the content, as well as the time required to deliver it. The prediction-based dynamic content adaptation approach is even more attractive, as it allows the estimation of various aspects of the transcoded content (e.g., data size and visual quality) prior to any transcoding operation. Numerous solutions have been proposed to estimate the visual quality, data size or transcoding time of transcoded JPEG or GIF images. Han et al. [6] proposed methods to estimate the data size of transcoded JPEG and GIF images as well as the transcoding time. However, these methods do not consider the visual aspect as

they concentrate only on the transmission time, which is based mainly on the data size and transcoding time of the adapted content. Other research studies have been proposed to predict the transcoded JPEG image data size and visual quality [13]–[15]. The obtained results are very reliable and take into account the scaling and quality parameter changes. However, in practical applications, these two pieces of information (visual quality and data size) need to be combined in order to quantify the adapted content quality or user experience, thus allowing the computation of the optimal version. Existing solutions usually set one parameter and optimize for the other one. For instance, in [15], the authors optimize the visual quality of transcoded JPEG images for a given data size value used as a constraint.

Louafi et al. [5] proposed a prediction-based dynamic content adaptation framework that estimates near-optimal transcoding parameters used to create the best adapted content possible for each mobile device. The framework exploits the predicted visual quality and data size of transcoded JPEG images [13]–[15], and applies them to the adaptation of enterprise documents such as Microsoft PowerPoint slides or Open Office Impress files. In their paper, to evaluate the adapted content quality, they propose a quality of experience (QoE) measure expressed as the product of the visual quality and transport quality of the adapted content. The former was evaluated using the SSIM index metric [16], mapped to its corresponding MOS values to estimate the real human perception [17]. The transport quality was modeled with a Zmf membership function [18], whose parameters are the delivery time and the user's preferences with respect to tolerable waiting time. The obtained results were quite good, but sensitive to the Zmf curve used in modeling the transport quality.

The choice of the cost function is crucial in content adaptation, and numerous approaches are possible for making this choice. Content adaptation methods proposed in the literature often use weighted additive functions (SAW: simple additive weighting), in which the adapted content characteristics and the network parameters are weighted and summed to evaluate the adapted content quality [9], [19], [20]. This approach is not really accurate as not all the parameters are compensatory to be summed [21], [22]. Instead, the authors of [5] proposed to use the weighted product method (WP), which consists in multiplying two high-level computed functions: visual quality and transport quality. In fact, both WP and SAW belong to a special branch of the multi-objective optimization field, namely *scalarization*, which consists in combining the objective functions into one representative function. Still, it is not always clear which scalarization function handles what problem best [23]. Therefore, it is more natural to formulate the problem at hand as a multi-objective optimization problem, in which the objective

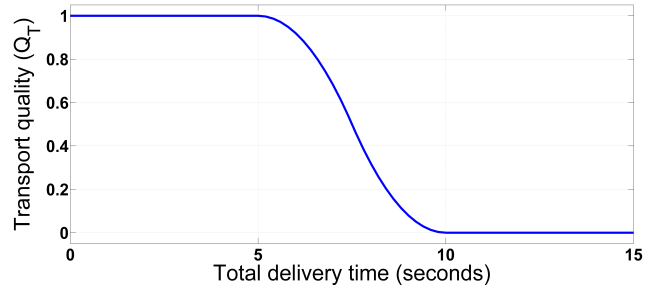


Fig. 3. Zmf-based transport quality modeling. Extracted from [5]

functions are the visual quality and the delivery time.

In [5], in order to evaluate the transport quality, the authors used the Zmf membership function that takes into account the delivery time and the user's preferences in terms of tolerable waiting time. Though the Zmf function represents a powerful tool for modelling the behavior of the delivery time, it acts as a sophisticated threshold, that is, from a certain delivery time value, the transport quality falls to zero, and thus eliminates all the contents having a delivery time higher than that value (see Fig. 3). Furthermore, within a certain waiting time interval, the Zmf function falls rather aggressively, and thus amplifies the effect of the adapted content data size prediction error used in the transport quality evaluation. This was adjusted in [5] by introducing a file size ratio that acts as a safety parameter. Nonetheless dealing with the actual delivery time instead of the Zmf-based transport quality would be desirable.

Regarding the user's preferences, instead of merely applying a tolerable waiting time on the transport quality, it would be better to articulate the user's preferences in both objective functions, i.e., the visual quality and the delivery time. This would be more adequate than applying the user's preferences only on the transport quality, as used in [5]. For instance, the user could weight the visual quality and delivery time to express the relative importance between them.

3 PROBLEM STATEMENT

Let \mathcal{C} be an enterprise document (e.g., PowerPoint) comprised of n pages denoted c_k . This can be formulated as follows: $\mathcal{C} = \{c_k\}_{k=1}^n$. For each page c_k , let $\mathcal{W}(c_k)$ and $\mathcal{H}(c_k)$ be its width and height, in pixels, respectively.

To reach a vast variety of mobile devices, which have different features (resolution, memory size, etc.), and to provide the mobile user with the best experience possible, enterprise documents must be adapted. To that end, an optimal transcoding parameter combination should be computed to be used in adapting each enterprise document page c_k .

In the following, we focus on adapting slide decks into JPEG- and XHTML-based Web pages to be visualized by mobile Web browsers, which are commonly

installed on mobile devices. In the case of JPEG-based Web pages, the whole slide is adapted to a JPEG image and wrapped into a Web page skeleton. In the case of XHTML-based Web pages, the enterprise document components are adapted separately and wrapped into a Web page skeleton. In the latter case, the XHTML-based Web page produced will be comprised of the same number of components as the enterprise document. For the transcoding parameters, we propose to use a target format (f), a scaling parameter (z), and a quality factor (QF). Let \mathcal{P} be the set of transcoding parameter combinations possible:

$$\mathcal{P} = \{(f, z, QF)\} \quad (1)$$

where:

- $f \in \{\text{JPEG}, \text{XHTML}\}$ is the target format into which each page c_k is adapted.
- $z \in]0, 1]$ is a scaling factor used to scale down c_k to obtain the desired resolution.
- $QF \in]0, 100]$ is the quality factor used to control the quality of the JPEG image.

These parameters are applied as follows:

- If $f = \text{JPEG}$, the JPEG-based Web page created will be comprised of one JPEG image adapted from c_k using z and QF .
- If $f = \text{XHTML}$, the XHTML-based Web page created will be comprised of image and text components. The same value of z is used to scale down the image and text components, and the same value of QF is used to change the quality of images.

Let T be the transcoding operation that adapts each page c_k into JPEG- and XHTML-based Web pages using the transcoding parameters f , z , and QF . It is given by:

$$\begin{aligned} T : \mathcal{C} \times \mathcal{P} &\rightarrow \mathcal{C}^{f,z,QF} \\ c_k \times (f, z, QF) &\mapsto c_k^{f,z,QF} \end{aligned} \quad (2)$$

where $\mathcal{C}^{f,z,QF}$ represents the set of adapted Web pages created by T from all pages of \mathcal{C} using all parameter combinations of \mathcal{P} , and $c_k^{f,z,QF}$ is the adapted Web page created from c_k using f , z and QF .

Let D be the target mobile device and $\mathcal{W}(D) \times \mathcal{H}(D)$, $\mathcal{S}(D)$, and $\mathcal{F}(D)$ be its screen maximum resolution (width and height), permissible file size, and set of supported formats, respectively.

Note that for a given page c_k , only a subset of its adapted versions $c_k^{f,z,QF}$ are feasible (can be rendered) on D . Formally, we say that $c_k^{f,z,QF}$ is feasible on D if it satisfies the following constraints:

$$\begin{aligned} \mathcal{S}(c_k^{f,z,QF}) &\leq \mathcal{S}(D) \\ \mathcal{W}(c_k^{f,z,QF}) &\leq \mathcal{W}(D) \\ \mathcal{H}(c_k^{f,z,QF}) &\leq \mathcal{H}(D) \\ f &\in \mathcal{F}(D) \end{aligned} \quad (3)$$

From these feasible versions, we are interested in computing the optimal one(s), i.e., those that maximize the user's quality of experience (QoE).

Generally, in QoE evaluation, different factors should be taken into account, such as the content, the end-user (who pays for the service, who uses the service), the service provider, cost, etc. [24], [25]. However, according to Kuipers et al. [26], in the adaptation of audio-visual content, the following three parameters should be considered in designing a QoE framework.

- 1) The audio-visual quality of the content before delivery (at the source).
- 2) The content quality of service QoS , affected by the conditions of the network in use.
- 3) The human perception (how the content is appreciated audio-visually).

In the case of adaptation and delivery of slide documents, from these three requirements, we conclude that the QoE of the delivered content can be expressed by two factors: its visual quality, $\mathcal{Q}_V(c_k^{f,z,QF}, D)$, and delivery time, $\mathcal{T}_D(c_k^{f,z,QF}, D)$. The first one expresses how the content is appreciated visually before delivery, whereas the second one interprets the impact of the delivery time on the appreciation of the content.

The problem can be formulated as maximizing \mathcal{Q}_V with respect to a given \mathcal{T}_D threshold. However, it is not obvious to establish a \mathcal{T}_D threshold, as this depends greatly on each user. That is, we may have users sensitive to the visual aspect and ready to sacrifice the delivery time to obtain good visual quality, and vice-versa. Therefore, based on these two factors, we formulate the problem of computing the optimal adapted content as a multi-objective optimization problem, in which we want to maximize \mathcal{Q}_V and minimize \mathcal{T}_D with respect to the constraints cited in (3). That is:

$$\begin{aligned} \text{MOPT} \quad &\max \quad \mathcal{Q}_V(c_k^{f,z,QF}, D) \\ &\min \quad \mathcal{T}_D(c_k^{f,z,QF}, D) \\ &\text{s.t.} \quad c_k^{f,z,QF} \text{ is feasible on } D \text{ (see (3))} \end{aligned} \quad (4)$$

We want to optimize the two objectives \mathcal{Q}_V and \mathcal{T}_D simultaneously, and thus, compute a set of optimal solutions that present the different compromises between these two objectives. In such problems, it is not straightforward to decide on the optimal solution, as the objective functions are conflicting. For instance, to minimize the delivery time, we should aggressively transcode the content, and thus reduce the visual quality. Conversely, to maximize the visual quality of the content, we should increase the content data size (up to a saturation point from which no visual quality improvement can be noticed), which increases the delivery time. Therefore, the first goal is to provide the end-user with a set of optimal solutions, and then propose to use an on-the-shelf mechanism (goal attainment) to select the preferred one by involving the end-user. This will be discussed in section 4.2.

4 MULTI-OBJECTIVE OPTIMIZATION

For a given page c_k and a target mobile device D , we define \mathcal{U} as the function between the set of feasible solutions (transcoding parameter combinations satisfying the constraints of (3)), denoted Λ , and its corresponding objective space (denoted $\hat{\Lambda}$). The latter, also called the criteria space, is comprised of a set of $(\mathcal{Q}_V, \mathcal{T}_D)$ combinations. \mathcal{U} is given by:

$$\mathcal{U} : \Lambda \rightarrow \hat{\Lambda}$$

$$(f, z, QF) \mapsto (\mathcal{Q}_V(c_k^{f,z,QF}, D), \mathcal{T}_D(c_k^{f,z,QF}, D)) \quad (5)$$

For a given page c_k , let $x_i = (f_i, z_i, QF_i) \in \Lambda$ and $x_j = (f_j, z_j, QF_j) \in \Lambda$ be two feasible solutions, and $y_i = (\mathcal{Q}_V(c_k^{f_i,z_i,QF_i}, D), \mathcal{T}_D(c_k^{f_i,z_i,QF_i}, D)) \in \hat{\Lambda}$ and $y_j = (\mathcal{Q}_V(c_k^{f_j,z_j,QF_j}, D), \mathcal{T}_D(c_k^{f_j,z_j,QF_j}, D)) \in \hat{\Lambda}$ their images by \mathcal{U} .

We say that y_i dominates y_j (formally: $y_i \succ y_j$) if either:

$$\mathcal{Q}_V(c_k^{f_i,z_i,QF_i}, D) > \mathcal{Q}_V(c_k^{f_j,z_j,QF_j}, D) \quad \text{and} \quad (6)$$

$$\mathcal{T}_D(c_k^{f_i,z_i,QF_i}, D) \leq \mathcal{T}_D(c_k^{f_j,z_j,QF_j}, D)$$

or

$$\mathcal{Q}_V(c_k^{f_i,z_i,QF_i}, D) \geq \mathcal{Q}_V(c_k^{f_j,z_j,QF_j}, D) \quad \text{and} \quad (7)$$

$$\mathcal{T}_D(c_k^{f_i,z_i,QF_i}, D) < \mathcal{T}_D(c_k^{f_j,z_j,QF_j}, D)$$

It is clear that solution x_i is better than solution x_j . Therefore, we want first to identify the set of feasible solutions that cannot be dominated by any other solution. In such a set of solutions, any improvement in one objective will be compensated by a deterioration in the other one. In multi-objective optimization, such solutions are called *Pareto-optimal solutions* (which we denote by Λ_p) and their images in the objective space are called *Pareto-optimal set* (denoted $\hat{\Lambda}_p$). These solutions satisfy the following property.

$$\forall x_p \in \Lambda_p, \nexists x \in \Lambda \mid y \succ y_p \quad (8)$$

where x and x_p are two solutions in the feasible and the Pareto-optimal spaces respectively, and y and y_p their images in the objective and Pareto-optimal set spaces respectively.

From Λ_p , the transcoding system can select an optimal solution and use it to create and deliver an optimal adapted content to the end-user. However, each Pareto-optimal solution presents a trade-off between the visual quality and the delivery time. Fig. 4, for instance, shows all possible solutions of a slide and its Pareto-optimal solutions. Therefore, we need a mechanism that helps us to favor one solution over the others. In this paper, we propose to involve the end-user in discriminating the solutions and selecting the one that expresses his needs; better visual quality versus shorter delivery time.

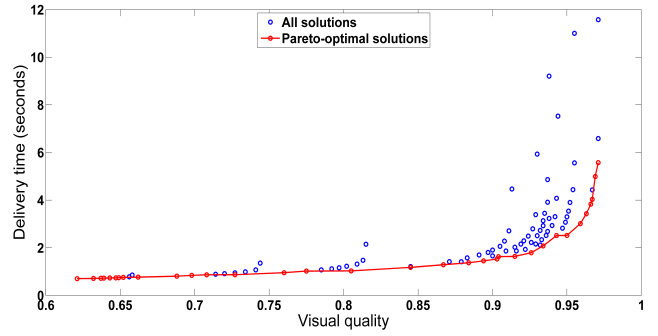


Fig. 4. Example of all valid transcoding solutions for a slide and its Pareto-optimal ones

4.1 User preference articulation

Different methods have been proposed to articulate preferences in order to compute the optimal solution that represents the perception of the end-user (called the decision maker in some fields) [23], [27], [28]. These methods are classified into two categories: a priori and a posteriori. In a priori articulation, the preferences are applied to the solution space and then the Pareto-optimal set is evaluated, from which the optimal point can be computed. In this case, the user preferences are applied to all the feasible solutions, which changes the objective function values, and consequently the Pareto-optimal set. In other words, different preference combinations may result in different optimal-Pareto sets. Conversely, with a posteriori articulation, the Pareto-optimal set is first computed, and then, the optimal one is identified using the user preferences.

In fact, the preference articulation is tied to the optimization method it uses. For instance, Marler and Arora [23] presented an interesting review of the various optimization methods classified into two groups: methods with a priori or a posteriori preference articulation.

Technically, preferences can be articulated using linguistic expressions (e.g., less important, important, and very important) or ranking weights supplied by the end-user to express the relative importance between the objectives. When linguistic expressions are used, they should be converted, using fuzzy membership functions for instance, into scores, which can be incorporated into the optimization process. For simplicity, we propose to use ranking weights to articulate preferences in this paper.

For a user with a mobile device D , let $w_v(D)$ and $w_t(D)$ be the weights associated with \mathcal{Q}_V and \mathcal{T}_D , respectively. To express the relative importance between the objective functions, $w_v(D)$ and $w_t(D)$ should respect the following properties:

$$w_v(D) + w_t(D) = 1$$

$$w_v(D) > 0, w_t(D) > 0 \quad (9)$$

4.2 Optimal solution evaluation

To solve multi-objective optimization problems, numerous optimization methods have been proposed in the literature [23]. In this paper, we propose to use the *Goal attainment* method [27], which tries to identify, from the Pareto-optimal set, the closest point to a goal point (also called *utopia*). The latter represents an ideal point, which is generally not attainable. This method is formulated as follows:

$$\begin{aligned} & \text{Minimize } \lambda \\ & x \in X \\ & \text{s.t. } F_i(x) - w_i \lambda \leq F_i^o, \quad i = 1, 2, \dots, l \end{aligned} \quad (10)$$

where X represents the solution space. $F_i(x)$ are the objective functions, and w_i and F_i^o are their associated weights and utopia points, respectively. λ is an unrestricted scalar to minimize and l is the number of objective functions.

In other words, we want to minimize the distance between each objective function and its utopia point, and so the problem can be formulated by:

$$\text{Minimize } \frac{F_i(x) - F_i^o}{w_i}, \quad i = 1, 2, \dots, l \quad (11)$$

The utopia point can be a point, not necessarily in the feasible solution space, with the highest value of Q_V and the lowest value of \mathcal{T}_D . Let $y^o = (Q_V(c_k^{f^o, z^o, QF^o}, D), \mathcal{T}_D(c_k^{f^o, z^o, QF^o}, D))$ be this utopia point and $x^o = (f^o(c_k, D), z^o(c_k, D), QF^o(c_k, D))$ its image by \mathcal{U}^{-1} .

For a content c_k and a target mobile device D , let $x^* = (f^*(c_k, D), z^*(c_k, D), QF^*(c_k, D))$ be the optimal solution, whose image in the objective space is the closest to the utopia point. It is given by:

$$x^* = \arg \min_{x \in \Lambda, y = \mathcal{U}(x)} \text{dev}(y, y^o) \quad (12)$$

where, $\text{dev}(y, y^o)$ is the weighted deviation between y and y^o .

In our problem, to be able to use deviation, or any distance measure, such as the Euclidean one, the objective functions should be dimensionless, ideally normalized (values confined between 0 and 1) [23]. As it is presented in section 5, the two objective functions, Q_V and \mathcal{T}_D , are normalized. This prevents one function from being shadowed by the other one with a higher scale of values.

Using the weight values introduced in section 4.1, $\text{dev}(y, y^o)$ is given by:

$$\begin{aligned} \text{dev}(y, y^o) = & \left| \frac{Q_V(c_k^{f, z, QF}, D) - Q_V(c_k^{f^o, z^o, QF^o}, D)}{w_V(D)} \right| \\ & + \left| \frac{\mathcal{T}_D(c_k^{f, z, QF}, D) - \mathcal{T}_D(c_k^{f^o, z^o, QF^o}, D)}{w_T(D)} \right| \end{aligned} \quad (13)$$

5 VISUAL QUALITY AND DELIVERY TIME EVALUATION

To solve equation (12) and compute the optimal solution dynamically, we need first to evaluate Q_V and \mathcal{T}_D and then show how these functions can be predicted. For completeness and to make the paper self-contained, Q_V and \mathcal{T}_D evaluation and estimation are presented in this section and in the next one respectively.

5.1 Visual Quality evaluation

For a page c_k and its adapted version $c_k^{f, z, QF}$, let $c_{k,i}$ and $c_{k,i}^{f, z, QF}$ be their components, respectively. Formally, we have:

$$\begin{aligned} c_k &= \{c_{k,i}\}_{i=1}^{m(k)} \\ c_k^{f, z, QF} &= \{c_{k,i}^{f, z, QF}\}_{i=1}^{m(k, f)} \end{aligned} \quad (14)$$

where $m(k)$ and $m(k, f)$ are the total number of components of c_k and $c_k^{f, z, QF}$, respectively. Ignoring the XHTML wrapper, which has no impact on quality in both the JPEG and XHTML transcoded format cases, we have:

$$m(k, f) = \begin{cases} m(k) & \text{if } f = \text{XHTML} \\ 1 & \text{if } f = \text{JPEG} \end{cases} \quad (15)$$

As proposed in [5], the visual quality of adapted content can be evaluated as a weighted sum of the visual qualities of its components, with the weights being the areas they occupy. That is, the larger the area, the larger the weight that should be applied. Thus, we have:

$$Q_V(c_k^{f, z, QF}, D) = \frac{\sum_{i=1}^{m(k, f)} \mathcal{A}(c_{k,i}^{f, z, QF}) Q_V(c_{k,i}^{f, z, QF}, D)}{\sum_{i=1}^{m(k, f)} \mathcal{A}(c_{k,i}^{f, z, QF})} \quad (16)$$

where $Q_V(c_{k,i}^{f, z, QF}, D)$ and $\mathcal{A}(c_{k,i}^{f, z, QF})$ are the visual quality of the adapted component $c_{k,i}^{f, z, QF}$ and the area it occupies, respectively.

Slide documents are generally comprised of text and images. The visual quality of images can be evaluated using any well-known full reference image quality metric, such as PSNR or SSIM [16]. Conversely to images, text is only affected by the scaling parameter, and so its visual quality is not affected. Therefore, without loss of generality, we assume that adapted text components can be rendered perfectly, and preserve their visual quality. Thus, we propose to evaluate Q_V of each component as follows:

$$Q_V(c_{k,i}^{f, z, QF}, D) = \begin{cases} Q_I(c_{k,i}^{f, z, QF}, D) & \text{if } c_{k,i}^{f, z, QF} \text{ is an image} \\ 1 & \text{if } c_{k,i}^{f, z, QF} \text{ is text} \end{cases} \quad (17)$$

where Q_I is the image visual quality, which can be evaluated using, for example, the SSIM index mea-

sure. SSIM is a full reference metric that requires both the original image and its transcoded version. That is, to evaluate the quality of an adapted image, the SSIM index measures the resulting image fidelity of the adapted image compared to that of the original one [16].

Note that when the values of Q_v are already confined between 0 and 1, such as when SSIM is used, there is no need for normalization.

5.2 Delivery time evaluation

The delivery time is comprised of the time required to perform the adaptation operation, plus the time taken by the adapted content to reach its recipient. For adapted content $c_k^{f,z,QF}$ and a target mobile device D , the delivery time can be computed by:

$$\mathcal{T}_D(c_k^{f,z,QF}, D) = \frac{S(c_k^{f,z,QF})}{N_B(D)} + S_L(D) + T_L(c_k^{f,z,QF}) \quad (18)$$

where:

- $S(c_k^{f,z,QF})$ is the data size of $c_k^{f,z,QF}$.
- $N_B(D)$ and $N_L(D)$ are the bitrate and latency of the network used by D , respectively.
- $S_L(D)$ is the server latency. It evaluates the time spent by the request, made by D , on the server (i.e., in the queue) waiting to be processed. It depends on the server performance and on the number of requests in the queue. For a given server, this value may be different for each device.
- $T_L(c_k^{f,z,QF})$ is the transcoding latency. It evaluates the time required by the adaptation operation to complete. It depends on the original content, c_k , and the actual transcoding parameters f , z , and QF . It can be estimated based on past transcoding operations. On high-end computers, this value can be small.

To normalize the delivery time, numerous methods have been proposed [23], some of which use the utopia point, which is not always known. In this paper, we propose to use a function that does not change the delivery time behavior, as was the case with the Zmf function. Therefore, we propose to use the following normalization function:

$$\mathcal{T}_D^{norm} = \frac{\mathcal{T}_D}{\mathcal{T}_D^{max}} \quad (19)$$

where \mathcal{T}_D^{norm} and \mathcal{T}_D^{max} are the normalized and maximum delivery time values, respectively. \mathcal{T}_D^{max} can be obtained from experimental results from previous transcodings, or using a sufficient maximum value.

For clarity, in the following, we will use the term \mathcal{T}_D to represent normalized delivery time, instead of using \mathcal{T}_D^{norm} .

TABLE 1
Predicted SSIM values computed for $QF_{in} = 80$ and $z_v = 0.4$ (adapted from [14])

QF	z									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
10	0.25	0.43	0.55	0.62	0.69	0.73	0.76	0.79	0.80	0.82
20	0.30	0.52	0.65	0.73	0.79	0.82	0.85	0.87	0.88	0.89
30	0.33	0.56	0.69	0.77	0.83	0.86	0.89	0.90	0.91	0.92
40	0.35	0.58	0.72	0.80	0.85	0.88	0.90	0.92	0.92	0.94
50	0.36	0.61	0.74	0.82	0.87	0.90	0.92	0.93	0.94	0.95
60	0.38	0.63	0.76	0.84	0.89	0.92	0.93	0.94	0.95	0.96
70	0.39	0.65	0.78	0.86	0.90	0.93	0.94	0.95	0.95	0.97
80	0.42	0.68	0.81	0.89	0.93	0.95	0.96	0.96	0.97	1.00
90	0.45	0.72	0.85	0.92	0.95	0.96	0.97	0.97	0.98	0.99
100	0.49	0.78	0.91	0.97	0.98	0.98	0.99	0.99	0.99	1.00

6 VISUAL QUALITY AND DELIVERY TIME ESTIMATION

In this section, we review how the visual quality (Q_v) and the delivery time (\mathcal{T}_D) can be estimated. For more details, we refer the reader to [5].

6.1 Visual quality estimation

When the target format $f = \text{JPEG}$, the adapted content will comprise only one JPEG image, $c_{k,1}^{\text{JPEG},z,QF}$, transcoded at z and QF . Let $c_{k,1}^{\text{JPEG},1,80}$ be an image created using $z = 1$ and $QF = 80$, which will be used as a reference image. Actually, each slide is converted into a reference image $c_{k,1}^{\text{JPEG},1,80}$, from which the various images can be created. This reference image is needed for predicting the visual quality and data size of transcoded images.

Using the SSIM index, the visual quality of the adapted content (16) becomes:

$$\begin{aligned} Q_v(c_k^{f,z,QF}, D) &= Q_v(c_{k,1}^{\text{JPEG},z,QF}, D) \\ &= \text{SSIM}(c_{k,1}^{\text{JPEG},z,QF}, c_{k,1}^{\text{JPEG},1,80}) \end{aligned} \quad (20)$$

This visual quality can be estimated using predicted SSIM values computed for QF_{in} , z_v , and various values of z and QF , and tabulated in [14]. QF_{in} and z_v are, respectively, the original (reference) image quality factor and a scaling parameter, derived from the viewing conditions, at which the two images (original and transcoded) should be scaled to compute their SSIM. Table 1, for instance, shows predicted SSIM values computed for $QF_{in} = 80$, $z_v = 0.4$, and various combinations of z and QF . In this case, the estimated visual quality of the adapted content becomes:

$$\begin{aligned} \widehat{Q}_v(c_k^{f,z,QF}, D) &= \widehat{\text{SSIM}}(QF_{in}(c_{k,1}^{\text{JPEG},1,80}), z_v, z, QF) \\ &= \widehat{\text{SSIM}}(80, z_v, z, QF) \end{aligned} \quad (21)$$

where $\widehat{\text{SSIM}}(QF_{in}, z_v, z, QF)$ is the estimated SSIM value that can be extracted from [14] using z_v , $QF_{in} = 80$, z , and QF . For example, using Table 1, we obtain: $\widehat{\text{SSIM}}(80, 0.4, 0.5, 70) = 0.90$.

When $f = \text{XHTML}$, the content and its adapted version will comprise the same number of components. Using the SSIM index, the visual quality of the

adapted content's components (17) becomes:

$$\mathcal{Q}_V(c_{k,i}^{f,z,QF}, D) = \begin{cases} \widehat{\text{SSIM}}(c_{k,i}^{\text{XHTML},z,QF}, c_{k,i}) & \text{if } c_{k,i} \text{ is an image} \\ 1 & \text{if } c_{k,i} \text{ is text} \end{cases} \quad (22)$$

where $c_{k,i}^{\text{XHTML},z,QF}$ is the XHTML adapted version of $c_{k,i}$.

Similarly, using predicted SSIM values, tabulated in [14], we can estimate the visual quality of each adapted component as follows:

$$\widehat{\mathcal{Q}}_V(c_{k,i}^{f,z,QF}, D) = \begin{cases} \widehat{\text{SSIM}}(QF_{in}(c_{k,i}), z_v, z, QF) & \text{if } c_{k,i} \text{ is an image} \\ 1 & \text{if } c_{k,i} \text{ is text} \end{cases} \quad (23)$$

where $QF_{in}(c_{k,i})$ is the quality factor of $c_{k,i}$, and $\widehat{\text{SSIM}}(QF_{in}(c_{k,i}), z_v, z, QF)$ its estimated SSIM value after adaptation, which can be extracted from [14] using $QF_{in}(c_{k,i})$, z_v , z , and QF . Thus, the \mathcal{Q}_V of the adapted content can be estimated using (16).

6.2 Delivery time estimation

To estimate the total delivery time (18), the adapted content data size, network bitrate and latency should be estimated at runtime if they are not available. For instance, the network bitrate can be estimated at runtime, using one of the various proposed algorithms [29], and the network latency can be estimated by pinging the mobile device, or using a mean value of probings that could have been performed when the user registered to the application [30]. Based on research proposed in [13], it is possible to estimate the data size of transcoded JPEG images subject to changing their scaling and quality factors.

For a JPEG image $c_{k,i}$ and its transcoded version $c_{k,i}^{\text{JPEG},z,QF}$, the relative data size between them is given by:

$$r(c_{k,i}^{\text{JPEG},z,QF}, c_{k,i}) = \frac{\mathcal{S}(c_{k,i}^{\text{JPEG},z,QF})}{\mathcal{S}(c_{k,i})} \quad (24)$$

These predicted relative data sizes are tabulated in [13], from which an example is extracted and presented in Table 2. This table presents predicted relative data sizes computed for $QF_{in} = 80$ (QF of the original image) and various values of z and QF .

After estimating the relative data size between each component and its transcoded version, we can estimate the data size of the adapted content, and consequently, the delivery time \mathcal{T}_D . The process is as follows:

When $f = \text{JPEG}$, using the reference JPEG image created before $c_{k,1}^{\text{JPEG},1,80}$, the adapted content data size can be formulated as follows:

$$\mathcal{S}(c_k^{\text{JPEG},z,QF}) = r(c_{k,1}^{\text{JPEG},z,QF}, c_{k,1}^{\text{JPEG},1,80}) \times \mathcal{S}(c_{k,1}^{\text{JPEG},1,80}) + \psi \quad (25)$$

TABLE 2

Predicted relative data sizes computed for $QF_{in} = 80$ (adapted from [13])

QF	z									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
10	0.03	0.04	0.05	0.07	0.08	0.10	0.12	0.15	0.17	0.20
20	0.03	0.05	0.07	0.09	0.12	0.15	0.19	0.22	0.26	0.32
30	0.04	0.05	0.08	0.11	0.15	0.19	0.24	0.21	0.34	0.41
40	0.04	0.06	0.09	0.13	0.17	0.22	0.28	0.34	0.40	0.50
50	0.04	0.06	0.10	0.14	0.19	0.25	0.32	0.39	0.46	0.54
60	0.04	0.07	0.11	0.16	0.22	0.28	0.36	0.44	0.53	0.71
70	0.04	0.08	0.13	0.18	0.25	0.33	0.42	0.52	0.63	0.85
80	0.05	0.09	0.15	0.22	0.31	0.41	0.52	0.65	0.78	0.95
90	0.06	0.12	0.21	0.31	0.44	0.59	0.75	0.93	1.12	1.12
100	0.10	0.24	0.47	0.75	1.05	1.46	1.89	2.34	2.86	2.22

where ψ is the data size of the XHTML wrapper of the Web page.

Using the predicted relative data size tabulated in [13], the adapted content data size can be estimated as follows:

$$\widehat{\mathcal{S}}(c_k^{\text{JPEG},z,QF}) = \widehat{r}(c_{k,1}^{\text{JPEG},z,QF}, c_{k,1}^{\text{JPEG},1,80}) \times \mathcal{S}(c_{k,1}^{\text{JPEG},1,80}) + \psi \quad (26)$$

$$\widehat{r}(c_{k,1}^{\text{JPEG},z,QF}, c_{k,1}^{\text{JPEG},1,80}) = \widehat{r}_I(QF_{in}(c_{k,1}^{\text{JPEG},1,80}), z, QF) = \widehat{r}_I(80, z, QF) \quad (27)$$

where:

- QF_{in} is the QF of the reference JPEG image.
- $\widehat{r}_I(80, z, QF)$ is the estimated relative data size, which can be extracted, for example, from Table 2.

When $f = \text{XHTML}$, the data size of the adapted content can be formulated as follows:

$$\mathcal{S}(c_k^{\text{XHTML},z,QF}) = \sum_{i=1}^{m(k)} r(c_{k,i}^{\text{XHTML},z,QF}, c_{k,i}) \times \mathcal{S}(c_{k,i}) + \psi \quad (28)$$

where $\mathcal{S}(c_{k,i})$ is the data size of the component $c_{k,i}$ and $r(c_{k,i}^{\text{XHTML},z,QF}, c_{k,i})$ the relative data size between $c_{k,i}$ and its XHTML transcoded version. As before, ψ represents the XHTML wrapper data size.

Similarly, using the predicted relative data size of transcoded JPEG images [13], we can estimate the adapted content data size. Thus, we have:

$$\widehat{\mathcal{S}}(c_k^{\text{XHTML},z,QF}) = \sum_{i=1}^{m(k)} \widehat{r}(c_{k,i}^{\text{XHTML},z,QF}, c_{k,i}) \times \mathcal{S}(c_{k,i}) + \psi \quad (29)$$

$$\widehat{r}(c_{k,i}^{\text{XHTML},z,QF}, c_{k,i}) = \begin{cases} \widehat{r}_I(QF_{in}(c_{k,i}), z, QF) & \text{if } c_{k,i} \text{ is an image} \\ 1 & \text{if } c_{k,i} \text{ is text} \end{cases} \quad (30)$$

where:

- $QF_{in}(c_{k,i})$ is the QF of $c_{k,i}$.
- $\widehat{r}_I(QF_{in}(c_{k,i}), z, QF)$ is the estimated relative data size between the image $c_{k,i}$ and its transcoded version $c_{k,i}^{\text{XHTML},z,QF}$, which can be extracted from [13] (e.g., Table 2).

We have shown how the adapted content data size can be estimated in both cases, JPEG and XHTML. Thus, using (18), we can estimate \mathcal{T}_D .

Now, having the possibility to estimate \mathcal{Q}_V and \mathcal{T}_D of each adapted content without performing any transcoding operation, we can estimate the optimal adapted content version by solving (4).

7 EXPERIMENTAL SETUP

To validate the proposed framework, we created a corpus of OpenOffice Impress presentation documents. They were created with a Java application we developed that uses OpenOffice APIs [31] to create, analyze, and parse documents, and to extract the important information required in computing \mathcal{Q}_V and \mathcal{T}_D . Besides creating the documents automatically, it parses each one, identifies the embedded images and text boxes, extracts their dimensions (height and width), evaluates the images' SSIM-index as well as their corresponding mapping to MOS scores, and computes their data sizes. For simplicity, each document comprises one slide composed of one text box and one image, and its position on the slide is set randomly. These text boxes and images were collected from various websites. To span a wide variety of slides, the areas occupied by text boxes and images in the slide are as follows:

$$\begin{aligned} \text{image size} &: 0\%, 10\%, 20\%, \dots, 100\% \\ \text{text size} &: 0\%, 10\%, 20\%, \dots, 100\% \end{aligned} \quad (31)$$

It is allowed that text boxes and images to be overlapped (partially or totally). Let \mathcal{V} be this set of OpenOffice Impress documents.

Let D be a target mobile device that has a resolution of 640×360 , and is connected to a 3-G network with a bitrate $BR(D) = 50$ kbps and network latency $NL(D) = 488$ ms [30]. For comparison with the solution proposed by [5], we use the same weight values for both \mathcal{Q}_V and \mathcal{Q}_T objectives, as follows:

$$w_V(D) = 0.5 \quad , \quad w_T(D) = 0.5 \quad (32)$$

Note that each weight value in (13) is the divisor of a deviation between a point and its utopia. Therefore, a higher value reduces the importance of the deviation to which it is applied. For instance, a higher $w_V(D)$ value will make the visual quality less important. Conversely, a lower weight value increases the importance of the deviation to which it is applied. When we refer to weights as penalties, this relationship is implied.

It is clear that the utopia point represents the ultimate optimal point. Although unreachable, it represents a good target, with which we compare our proposed dynamic system (how far we are from utopia). Thus, we propose to use the following point as utopia:

$$(\mathcal{Q}_V^o, \mathcal{T}_D^o) = (1, 0) \quad (33)$$

It is also important to compare our framework with the one presented in [5], which we denote as SCADS (*scalarization-based dynamic system*). We also want to compare both systems (ours and SCADS) with an exhaustive static system, which creates a set of adapted contents, evaluate their \mathcal{Q}_V and \mathcal{T}_D , and generate a set of optimally adapted contents representing the Pareto-optimal set. This system adapts each document $c_k \in \mathcal{V}$ into JPEG- and XHTML-based Web pages using OpenOffice XHTML and JPEG filters, using the following transcoding parameter values:

$$\begin{aligned} f &\in \{\text{JPEG, XHTML}\} \\ z &\in \{0.1, 0.2, 0.3, \dots, 1\} \\ QF &\in \{10, 20, 30, \dots, 100\} \end{aligned} \quad (34)$$

Note that the native OpenOffice XHTML-based filter is very limited and contains numerous bugs that we have fixed, and that the integration of the improved version of the filter is under process [32].

Thus, the systems to be compared are:

- The scalarization-based dynamic system (SCADS): the dynamic system proposed in [5].
- The multi-objective-based dynamic system (MODS): our new proposed dynamic system.
- The exhaustive static system (EXSS): the exhaustive static system.

Regarding the SCADS dynamic system, we collect, from [5], the optimal estimated transcoding parameters $\hat{f}^*(c_k, D)$, $\hat{z}^*(c_k, D)$ and $\hat{QF}^*(c_k, D)$, and their resulting (not estimated) \mathcal{Q}_V and \mathcal{T}_D (normalized \mathcal{T}_D , called \mathcal{Q}_T).

Regarding our MODS dynamic system, for each $c_k \in \mathcal{V}$, its \mathcal{Q}_V and \mathcal{T}_D are estimated using the same set of transcoding parameter combinations of (34). By solving (4) on the estimated \mathcal{Q}_V and \mathcal{T}_D solution spaces, we identify the optimal transcoding parameters ($\check{f}^*(c_k, D)$, $\check{z}^*(c_k, D)$, $\check{QF}^*(c_k, D)$), and use them to obtain their actual (not estimated) optimal \mathcal{Q}_V and \mathcal{T}_D .

Regarding the EXSS static system, for each $c_k \in \mathcal{V}$, 200 adapted slides are created using the same set of transcoding parameter combinations of (34). Then, for each adapted version, its \mathcal{Q}_V and \mathcal{T}_D are computed, which constitutes the computed (not estimated) solution space. We solve (4) on this computed solution spaces and identify the optimal transcoding parameters ($f^*(c_k, D)$, $z^*(c_k, D)$, $QF^*(c_k, D)$) and their corresponding computed optimal \mathcal{Q}_V and \mathcal{T}_D . Also, we use (6) and (7) to extract the Pareto set from the computed solution space.

The objective is to show how much our proposed solution is far from the utopia (optimal, but unreachable), and how it compares with the scalarization-based dynamic system developed earlier.

Although MODS provides a set of optimal solutions (Pareto-optimal set), we also want to see where the optimal solutions obtained by both dynamic systems,

MODS (computed using the goal-attainment method) and SCADS, are located on the Pareto set generated by the EXSS system (computed, not estimated, solutions). The latter represent the different optimal trade-offs between the visual quality and delivery time.

The optimal solutions obtained by the SCADS and MODS dynamic systems, as well as all the possible solutions generated by the EXSS system, were stored in data structures as follows:

$$\begin{aligned} SCADS_k^* &= [c_k, \hat{f}^*, \hat{z}^*, \hat{Q}F^*, \mathcal{Q}_V(c_k^{\hat{f}^*, \hat{z}^*, \hat{Q}F^*}, D), \mathcal{T}_D(c_k^{\hat{f}^*, \hat{z}^*, \hat{Q}F^*}, D)] \\ MODS_k^* &= [c_k, \hat{f}^*, \hat{z}^*, \hat{Q}F^*, \mathcal{Q}_V(c_k^{\hat{f}^*, \hat{z}^*, \hat{Q}F^*}, D), \mathcal{T}_D(c_k^{\hat{f}^*, \hat{z}^*, \hat{Q}F^*}, D)] \\ EXSS_k &= [c_k, f, z, QF, \mathcal{Q}_V(c_k^{f, z, QF}, D), \mathcal{T}_D(c_k^{f, z, QF}, D)] \\ EXSS_k^* &= [c_k, f^*, z^*, QF^*, \mathcal{Q}_V(c_k^{f^*, z^*, QF^*}, D), \mathcal{T}_D(c_k^{f^*, z^*, QF^*}, D)] \\ EXSS_k^p &= [c_k, f^p, z^p, QF^p, \mathcal{Q}_V(c_k^{f^p, z^p, QF^p}, D), \mathcal{T}_D(c_k^{f^p, z^p, QF^p}, D)] \end{aligned}$$

where $SCADS_k^*$, $MODS_k^*$ are two vectors storing the best transcoding parameter combinations and their \mathcal{Q}_V and \mathcal{T}_D obtained with the SCA and our MOPT dynamic systems, respectively. The $EXSS_k$ is a table that contains all the solutions provided by the static exhaustive system, from which the Pareto set of each document is extracted ($EXSS_k^p$), and the computed optimal solution ($EXSS_k^*$).

8 EXPERIMENTAL RESULTS

8.1 Comparison of SCADS, MODS and EXSS optimal solutions

Unlike the SCADS system, which provides a single optimal solution, the MODS method we propose provides a set of optimal solutions. While the SCADS solutions are based on a QoE measure defined as a product of the visual and transport qualities, the MODS solutions represent the various trade-offs between the visual quality and the delivery time. To show this aspect graphically, we present, in Figs. 5 and 6, for a selected document, the SCADS optimal solution and the MODS optimal solutions obtained, with different weight combinations, as well as the EXSS solutions and their Pareto-optimal set. We can observe that the MODS solutions in the figures are located at different positions on the Pareto front, and represent different visual quality and delivery times compromises. What is somewhat unexpected is that both solutions for weight combinations (0.5, 0.5) and (0.2, 0.8) overlap. This is due to the imprecision in the quality and estimation models of [14].

Besides the flexibility of providing a set of optimal solutions offered by the MODS system compared to SCADS, we go further bellow, and compare the performance of the two systems against two targets:

- *Utopia*: how much the optimal solutions obtained are close to utopia.
- *EXSS Pareto-optimal set*: how much they are close to the EXSS Pareto-optimal set.

The first target is more stable and widely used in the literature, but is unreachable. The second target represents a reasonable set of comparison points.

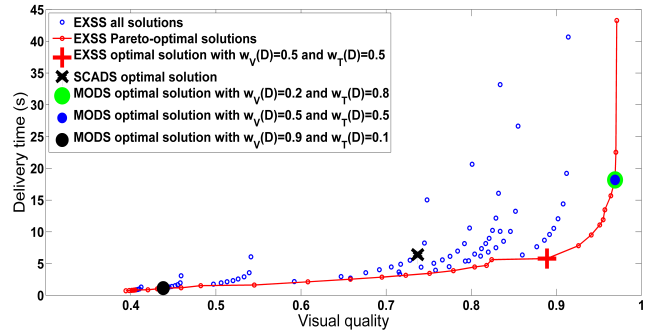


Fig. 5. JPEG optimal solution obtained by SCADS and those obtained by MODS with different weight combinations versus the EXSS Pareto-optimal set

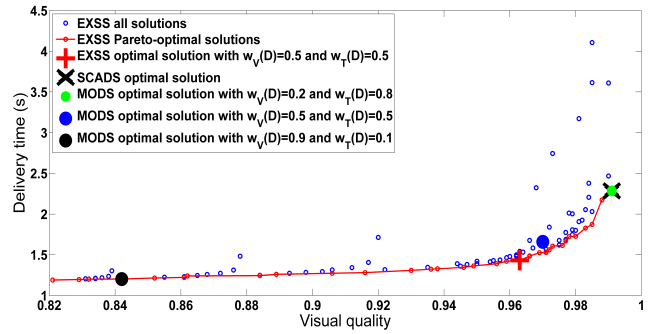


Fig. 6. XHTML optimal solution obtained by SCADS and those obtained by MODS with different weight combinations versus the EXSS Pareto-optimal set

8.2 Relative absolute deviation from utopia

For each document c_k from \mathcal{V} , we compute the absolute deviation of the optimal solutions obtained by the SCADS, MODS and EXSS systems from utopia, using $w_v(D) = 0.5$ and $w_t(D) = 0.5$. We then go on to compute the relative absolute deviation from utopia to a negative ideal, which corresponds to the farthest point: $y^- = (\mathcal{Q}_V^-, \mathcal{T}_D^-) = (0, 1)$ (i.e., zero visual quality with longest time to arrive). For a point y , the relative absolute deviation from utopia to y^- is given by:

$$\frac{\text{dev}(y, y^o)}{\text{dev}(y^-, y^o)} \quad (35)$$

where y^o , y^- are the utopia point (1,0) and the farthest point from utopia (0, 1), respectively (bottom-right and top-left corners of Fig. 5). dev is the absolute deviation, which is evaluated using (13).

The results are presented in Figs. 7 and 8, for JPEG and XHTML, respectively. Since utopia cannot be reached, the relative error value itself is not sufficient. We must also consider how close a method is to EXSS. Clearly, the optimal solutions obtained by our proposed MODS dynamic system exhibit less variance than those obtained by the SCADS dynamic system, although on average, they may appear to perform comparably. In the case of XHTML, the optimal solutions obtained by the MODS dynamic system are much closer to EXSS and exhibit less variance than

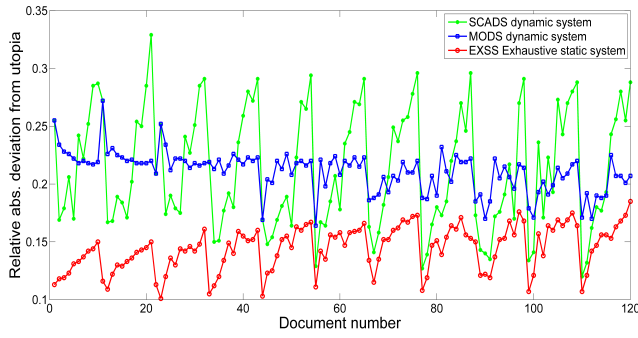


Fig. 7. JPEG relative absolute deviation from utopia to y^- computed for the optimal solutions obtained by SCADS, MODS and EXSS systems, using $w_V(D) = 0.5$ and $w_T(D) = 0.5$

TABLE 3

Average relative absolute deviation from utopia to y^- and their variances

Systems	Average relative absolute deviation		Variance ($\times 10^{-3}$)	
	JPEG	XHTML	JPEG	XHTML
SCADS	0.214	0.112	2.619	3.504
MODS	0.211	0.100	0.296	3.200
EXSS	0.144	0.095	0.376	3.329

those obtained by the SCADS dynamic system, except for the first ten documents, in which there is no image, but only text of different sizes (see (31)). In fact, for the first ten documents, the solution space is comprised of one point (all the points overlap) with a $Q_V = 1$ and T_D being the same for all the adapted contents. This is quite reasonable, as both Q_V and T_D are not sensitive to scaling and quality changes for documents comprised of only text components. However, for the three systems (SCADS, MODS and EXSS), the estimated solutions are still very close to each other. Note that, the curves follow a certain periodicity due to the order of the documents used. In each ten documents, the area of images is fixed (but increases by 10% for each ten documents) and the areas of text vary from 10% to 100%.

From a statistical point of view, Table 3 presents the average absolute relative deviations from utopia to y^- and their variances. The averaged results confirm that the optimal solutions obtained by the MODS dynamic system are slightly closer to utopia than those obtained by the SCADS dynamic system. However, the variances confirm the high variability of the SCADS optimal solutions for JPEG as compared to those obtained by MODS (2.62×10^{-3} vs. 0.29×10^{-3}). On the other hand, the MODS method does not perform as well as EXSS for JPEG (0.21 vs. 0.14), but is very close for XHTML (0.10 vs. 0.09). As expected, the XHTML format is more accurate than the JPEG one, since in XHTML, only embedded components are estimated, and not the entire slide, which is the case of JPEG.

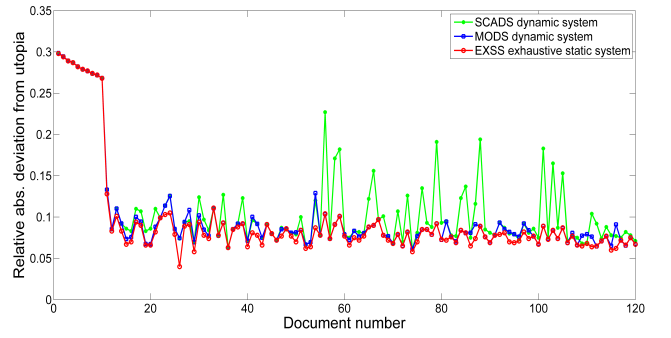


Fig. 8. XHTML relative absolute deviation from utopia to y^- computed for the optimal solutions obtained by SCADS, MODS and EXSS systems, using $w_V(D) = 0.5$ and $w_T(D) = 0.5$

8.3 Minimum deviation from the Pareto front

Another very important aspect is to show where the solutions obtained by both the SCADS and MODS dynamic systems are located on the Pareto front. We intend to show how much the SCADS and MODS optimal solutions are far from (or close to) the different trade-offs between visual quality and delivery time (the non-dominated solutions). First, for each document, we compute the Pareto optimal set from the set of solutions provided by the EXSS system. Then, we compute the absolute deviation between the SCADS and MODS optimal solutions and all the solutions that are on the EXSS Pareto front, and we identify the minimal ones. For each point y , the minimum absolute deviation is computed as follows:

$$\text{mindev}(y) = \arg \min_{y_i \in \Lambda_p^{\text{EXSS}}} \text{dev}(y, y_i) \quad (36)$$

where Λ_p^{EXSS} is the EXSS Pareto optimal set. Similarly, using the farthest point, $y^- = (Q_V^-, T_D^-) = (0, 1)$, we compute the relative minimum absolute deviations as follows:

$$\frac{\text{mindev}(y)}{\text{dev}(y^-, y^o)} \quad (37)$$

where dev is the absolute deviation, which is evaluated using (13).

Figs. 9 and 10 show the relative minimum absolute deviation between the optimal solutions obtained by the SCADS and MODS dynamic systems and their closest points from the EXSS Pareto optimal front, for JPEG and XHTML, respectively.

These figures show that the MODS optimal solutions (obtained with the goal attainment method) are closer to the EXSS Pareto-optimal solutions than are the SCADS optimal solutions. In this scenario, for JPEG all the MODS optimal solutions are on the Pareto front, whereas for XHTML, the majority is.

Statistically, Table 4 shows the averages of the relative minimum absolute deviations from the EXSS Pareto optimal front as well as their variances. This gives a global view of the distance between the

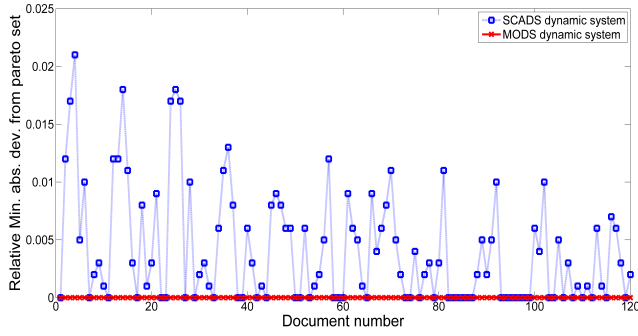


Fig. 9. JPEG relative minimum absolute deviation between SCADS and MODS optimal solutions, and the EXSS Pareto optimal set

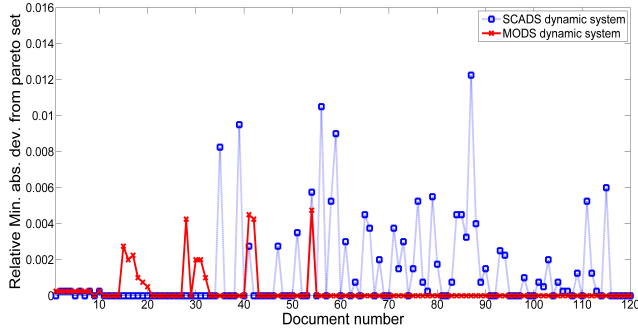


Fig. 10. XHTML relative minimum absolute deviation between SCADS and MODS optimal solutions, and the EXSS Pareto optimal set

optimal solutions obtained by both the SCADS and MODS dynamic systems and the EXSS Pareto optimal front. Globally, both methods provide solutions that are close to the Pareto front, but the MODS ones are significantly closer.

On a final note, although the dynamic system presented in [5] provides good solutions, the method we propose in this paper provides many advantages:

- The solutions are closer to the EXSS Pareto optimal front, compared to those of the SCADS system. Furthermore, our method exhibits much smaller deviation variances from the EXSS Pareto optimal front as well as from the utopia.
- It is more flexible, as it provides the decision maker (or end-user) with all the possible optimal solutions (trade-offs), and allows him to select the one that satisfies his preferences (higher visual quality vs. shorter delivery time).

This paper also allowed us to once again validate the performance of the dynamic system by [5]. The authors of [5] validated the performance of their proposed framework against a static exhaustive system, which is quite reasonable. However, in this paper, we compared their framework with two important information: the utopia point (ultimate optimal) and the Pareto front.

TABLE 4
Average relative minimum absolute deviation from the EXSS Pareto optimal set, and their variances

Systems	Average relative min. abs. dev. ($\times 10^{-1}$)		Variance ($\times 10^{-3}$)	
	JPEG	XHTML	JPEG	XHTML
SCADS	0.040	0.013	0.024	0.006
MODS	0	0.003	0	0.001

8.4 Examples of optimal solutions for selected documents

In this section, we select three examples of both JPEG and XHTML to show the EXSS Pareto optimal solutions, and where the optimal ones obtained by the SCADS and MODS systems are located in comparison. Figs. 11 and 12 show the selected examples for JPEG and XHTML, respectively. The documents are selected to show different behaviors of the three systems: MODS, SCADS and EXSS. For JPEG and in this scenario, all the MODS optimal solutions obtained with the weight combinations (0.5, 0.5) and (0.2, 0.8) overlap. In other words, with a weight combination of (0.5, 0.5), the MODS system reached its maximum Q_v . We tested the system with different weight combinations, such as (0.1, 0.9), (0.05, 0.95), (0.01, 0.99), to obtain higher Q_v , and the results were the same.

Note that all the SCADS optimal solutions are within the delivery time range of [0,10] because they are constrained by the Zmf membership function used to normalize the delivery time. In the MODS dynamic system, we do not have this restriction, but the end-user can use his preferences to favor one objective over the other. Alternatively, in the optimization problem (4), it is also possible to add a constraint on the delivery time, such as discarding all solutions whose delivery times are higher than 7 seconds (retaining only $\mathcal{T}_D(c_k^{f,z,Q^F}, D) < 7s$).

9 CONCLUSION

In mobile Web conferencing, slide documents should be adapted to satisfy the end-user's mobile constraints and environment. To that end, a compromise should be sought between the visual aspect of the adapted content and its delivery time in order to provide the end-user with the optimal quality of experience.

In this paper, we formulated the problem of computing the optimal adapted content as a multi-objective optimization problem, with the objectives being the visual quality and the delivery time. We compared the performance of the proposed solution with two important references. First, we used an utopia point, which represents the ultimate optimal. It represents a desirable target with which we compared our dynamic system. Second, we used the Pareto front, which is a set of optimal solutions representing the different trade-offs between the visual quality

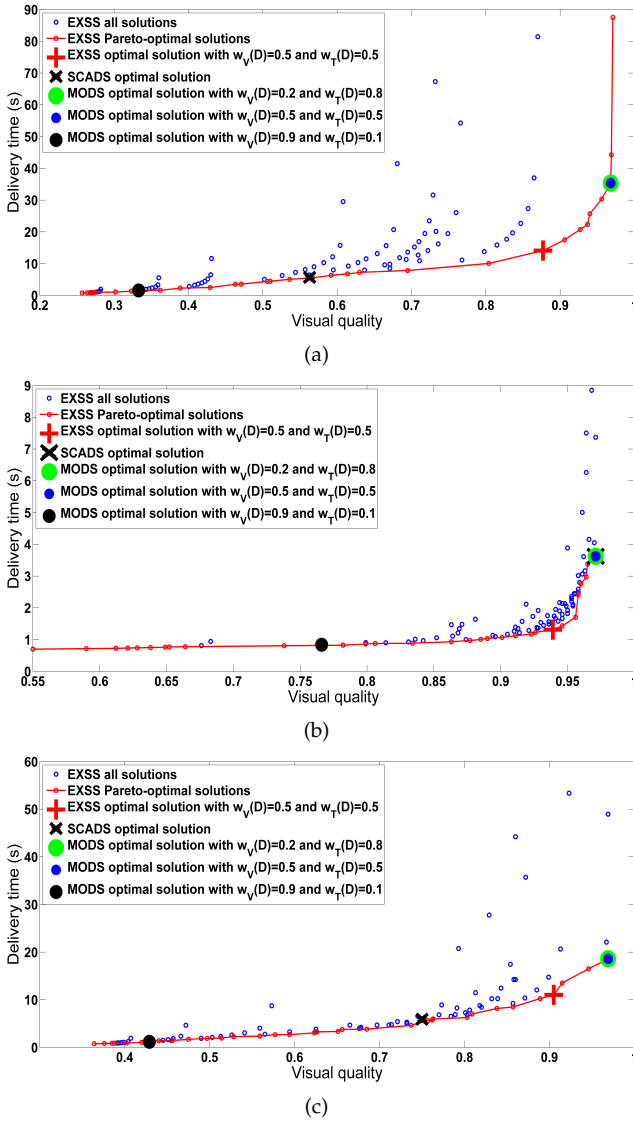


Fig. 11. JPEG results obtained for three selected documents with different weight combinations

and delivery time. Furthermore, we compared our solution with that proposed in [5].

The experiments showed that the results obtained by our system exhibit, for JPEG, less variance than those presented in [5], and are comparable on average. In the case of XHTML, the optimal solutions obtained were much closer to utopia, and had less variance than those obtained by [5]. Although the solutions of both the proposed system and those of [5] were close to the Pareto optimal front, those of the proposed system were much closer. Furthermore, for the majority of documents tested, it reached a value on the Pareto optimal front.

The proposed dynamic system was designed to be generic and flexible, and as a result, future research can be carried out to show its applicability to other content types, such as enterprise documents and video. It is also important to investigate, in future work, the relationship between the adapted slides and

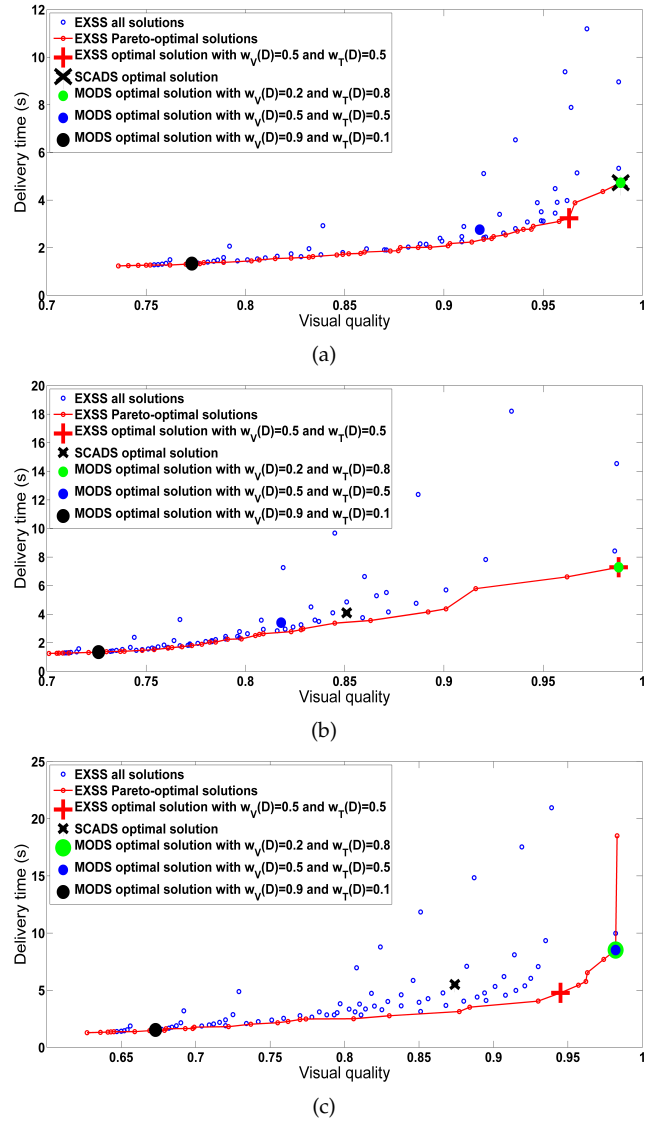


Fig. 12. XHTML results obtained for three selected documents with different weight combinations

their corresponding energy consumption. Once such a relationship is established, the framework can be extended easily by adding a third objective function, called for example energy consumption.

REFERENCES

- [1] A. Karadimce and D. Davcev, "Adaptive Multimedia Content in Mobile Cloud Computing Environment," in *Cloud Networking (CLOUDNET)*, 2012 IEEE 1st International Conference on, 2012, pp. 209–211.
- [2] S. Wang and S. Dey, "Adaptive Mobile Cloud Computing to Enable Rich Mobile Multimedia Applications," *Multimedia, IEEE Transactions on*, vol. 15, no. 4, pp. 870–883, 2013.
- [3] W. Lum and F. Lau, "User-Centric Content Negotiation for Effective Adaptation Service in Mobile Computing," *IEEE Transactions on Software Engineering*, vol. 29, no. 12, pp. 1100–1111, Dec. 2003.
- [4] S. Coulombe and G. Grassel, "Multimedia adaptation for the multimedia messaging service," *IEEE Communications Magazine*, vol. 42, no. 7, pp. 120–126, Jul. 2004.
- [5] H. Louafi, S. Coulombe, and U. Chandra, "Quality Prediction-Based Dynamic Content Adaptation Framework Applied to Collaborative Mobile Presentations," *IEEE Transactions on Mobile Computing*, vol. 12, no. 10, pp. 2024–2036, Oct. 2013.

- [6] R. Han, P. Bhagwat, R. LaMaire, T. Mummert, V. Perret, and J. Rubas, "Dynamic Adaptation in an Image Transcoding Proxy for Mobile Web Browsing," *IEEE Personal Communications*, vol. 5, no. 6, pp. 8–17, 1998.
- [7] H. Louafi, S. Coulombe, and U. Chandra, "Efficient Near-Optimal Dynamic Content Adaptation Applied to JPEG Slides Presentations in Mobile Web Conferencing," in *The 27th IEEE International Conference on Advanced Information Networking and Applications (AINA-2013)*, Mar. 2013, pp. 724–731.
- [8] —, "Robust QoE-aware prediction-based dynamic content adaptation framework applied to slides documents in mobile Web conferencing," *Springer Multimedia Tools and Applications*, pp. 1–38, 2014.
- [9] W. Lum and F. Lau, "User-Centric Adaptation of Structured Web Documents for Small Devices," in *19th Int. Conf. on Advanced Information Networking and Applications (AINA'05)*, vol. 1. IEEE, 2005, pp. 507–512.
- [10] Y. Xiao, Y. Tao, and W. Li, "A Dynamic Web Page Adaptation for Mobile Device Based on Web2.0," in *Advanced Software Engineering and Its Applications*. IEEE, Dec. 2008, pp. 119–122.
- [11] T-Mobile, "Memory full notification error or incoming messages rejected on Android," Online, 2012, accessed on 21 July 2014. [Online]. Available: <http://support.t-mobile.com/docs/DOC-3260>
- [12] Vodafone, "Galaxy Ace memory full - incoming texts are being rejected?" Online, 2013, accessed on 21 July 2014. [Online]. Available: <http://forum.vodafone.co.uk/t5/Samsung/Galaxy-Ace-memory-full-incoming-texts-are-being-rejected/td-p/1531006>
- [13] S. Pigeon and S. Coulombe, "Computationally Efficient Algorithms for Predicting the File Size of JPEG Images Subject to Changes of Quality Factor and Scaling," in *2008 24th Biennial Symposium on Communications*. IEEE, Jun. 2008, pp. 378–382.
- [14] S. Coulombe and S. Pigeon, "Quality-aware Selection of Quality Factor and Scaling Parameters in JPEG Image Transcoding," in *2009 IEEE Symp. on Computational Intelligence for Multimedia Signal and Vision Processing*, Mar. 2009, pp. 68–74.
- [15] —, "Low-complexity Transcoding of JPEG Images With Near-optimal Quality Using a Predictive Quality Factor and Scaling Parameters," *Image Processing, IEEE Transactions on*, vol. 19, no. 3, pp. 712–721, Mar. 2010.
- [16] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [17] H. Sheikh, M. Sabir, and A. Bovik, "A Statistical Evaluation of Recent Full Reference Image Quality Assessment Algorithms," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3440–3451, Nov. 2006.
- [18] The MathWorks, "Z-shaped built-in membership function," 2012, accessed on 19 September 2012. [Online]. Available: <http://www.mathworks.com/help/toolbox/fuzzy/zmf.html>
- [19] A. Richards, M. Antoniadis, V. Witana, and G. Rogers, "Mapping User Level QoS from a Single Parameter," in *Proceedings of the International Conference on Multimedia Networks and Services (MMNS '98)*, Nov. 1998.
- [20] Y. Zhang, S. Zhang, and S. Han, "A New Methodology of QoS Evaluation and Service Selection for Ubiquitous Computing," in *Wireless Algorithms, Systems, and Applications*, ser. LNCS. Springer Berlin / Heidelberg, 2006, vol. 4138, pp. 69–80.
- [21] L. Lee and R. Anderson, "A Comparison of Compensatory and Non-Compensatory Decision Making Strategies in IT Project Portfolio Management," Online, 2009, accessed on 10 October 2012. [Online]. Available: <http://aisel.aisnet.org/irwitpm2009/9>
- [22] A. Dieckmann, K. Dippold, and H. Dietrich, "Compensatory versus Noncompensatory Models for Predicting Consumer Preferences," *Judgment and Decision Making*, vol. 4, no. 3, pp. 200–213, 2009.
- [23] R. Marler and J. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and Multidisciplinary Optimization*, vol. 26, no. 6, pp. 369–395, 2004.
- [24] K. Laghari and K. Connelly, "Toward total quality of experience: A QoE model in a communication ecosystem," *Communications Magazine, IEEE*, vol. 50, no. 4, pp. 58–65, Apr. 2012.
- [25] T. Hobfeld, R. Schatz, M. Varela, and C. Timmerer, "Challenges of QoE management for cloud applications," *Communications Magazine, IEEE*, vol. 50, no. 4, pp. 28–36, Apr. 2012.
- [26] F. Kuipers, R. Kooij, D. De Vleeschauwer, and K. Brunnström, "Techniques for measuring quality of experience," in *Wired/Wireless Internet Communications*, ser. Lecture Notes in Computer Science. Springer-Verlag Berlin/Heidelberg, 2010, vol. 6074, pp. 216–227.
- [27] F. Gembicki and Y. Haimes, "Approach to performance and sensitivity multiobjective optimization: The goal attainment method," *Automatic Control, IEEE Transactions on*, vol. 20, no. 6, pp. 769–771, Dec. 1975.
- [28] A. Beben, J. Batalla, W. Chai, and J. Śliwiński, "Multi-criteria decision algorithms for efficient content delivery in content networks," *Annals of Telecommunications - Annales des Télécommunications*, 2013.
- [29] H. Ningning and P. Steenkiste, "Evaluation and characterization of available bandwidth probing techniques," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 6, pp. 879–894, Aug. 2003.
- [30] P. Svoboda, F. Ricciato, W. Keim, and M. Rupp, "Measured WEB Performance in GPRS, EDGE, UMTS and HSDPA with and without Caching," in *IEEE Int. Symp. on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Jun. 2007, pp. 1–6.
- [31] OpenOffice, "The OpenOffice.org API Project," Online, 2010, accessed on 19 September 2012. [Online]. Available: <http://api.openoffice.org>
- [32] H. Louafi, "Improving the XHTML export filter," Online, 2013, accessed on 21 March 2014. [Online]. Available: https://fosdem.org/2014/schedule/event/improving_the_xhtml_export_filter/



Habib Louafi (S'11) holds an Engineering degree in Computer Science from the University of Oran (Algeria), an M.Sc. from the Université du Québec à Montréal (UQAM), and a Ph.D. from École de technologie supérieure (ÉTS is a part of the Université du Québec network). He is currently working as a post-doctoral fellow at Synchromedia Laboratory for multimedia communication in telepresence (ÉTS). His fields of interest include mobile computing, context-aware systems, QoE, content adaptation and collaborative mobile Web conferencing.



Stéphane Coulombe (S'90-M'98-SM'01) received a B.Eng. in Electrical Engineering from the École Polytechnique de Montréal, Canada, in 1991, and a Ph.D. from INRS-Telecommunications, Montréal, in 1996. He is a Professor in the Software and IT Engineering Department, École de technologie supérieure (ÉTS is a part of the Université du Québec network). From 1997 to 1999, he was with Nortel Wireless Network Group in Montreal, and from 1999 to 2004, he worked with the Nokia Research Center, Dallas, TX, as Senior Engineer and as Program Manager in the Audiovisual Systems Laboratory. He joined ÉTS in 2004, where he currently carries out research and development on video processing and systems, media adaptation, and transcoding. Since 2009, he has held the Vantrix Industrial Research Chair in Video Optimization.



Mohamed Cheriet is a full professor in the Automation Engineering Department at École de technologie supérieure, a constituent of the Université du Québec network. His expertise includes document image analysis, optical character recognition, mathematical models for image processing, pattern classification models, and learning algorithms, as well as perception in computer vision. Cheriet received a PhD in Computer Science from the University of Pierre et Marie Curie. He co-founded the Laboratory for Imagery, Vision, and Artificial Intelligence (LIVIA) and founded and directs the Synchromedia Consortium (Multimedia Communication in Telepresence) at the Université du Québec.