Efficient Service Selection in Multimedia Documents Adaptation Processes

Zakaria Laboudi\textsuperscript{1(✉)}, Abdelkader Moudjar\textsuperscript{2}, Asma Saighi\textsuperscript{1}, and Hamana Nazim Hamri\textsuperscript{1}

\textsuperscript{1} RELA(CS)2 Laboratory, University of Oum El-Bouaghi, Route de Constantine BP 321, 04000 Oum El-Bouaghi, DZ, Algeria
\texttt{laboudi.zakaria@univ-oeb.dz}

\textsuperscript{2} MISC Laboratory, University of Constantine, 2, Nouvelle ville Ali Mendjeli BP 67A, 25100 Constantine, DZ, Algeria

Abstract. Pervasive systems help access to multimedia documents at any time, from anywhere and through several devices (smart TV, laptop, tablet, etc.). Nevertheless, due to changes in users’ contexts (e.g. noisy environment, preferred language, public place, etc.), restrictions on correct access to these documents may be imposed. One possible solution is to adapt their contents using adaptation services so that they comply, as far as possible, with the current constraints. In this respect, several adaptation approaches have been proposed. However, when it comes to selecting the required adaptation services, they often carry out this task according to predefined configurations or deterministic algorithms. Actually, the efficient selection of adaptation services is one of the key-elements involved in improving the quality of service in adaptation processes. To deal with this issue (i.e. the efficient selection of adaptation services), we first provide an enriched problem formulation as well as methods that we use in problem-solving. Then, we involve standard and compact evolutionary algorithms to find efficient adaptation plans. The standard version is usually adopted in systems that are not subject to specific constraints. The compact one is used in systems for which constraints on computational resources and execution time are considered. The proposal is validated through simulation, experiments and comparisons according to performance, execution time and energy consumption. The obtained results are satisfactory and encouraging.

Keywords: Context-aware pervasive systems · Multimedia documents adaptation · QoS-aware adaptation service selection · Evolutionary algorithms

1 Introduction

Pervasive computing is one of the latest emerging trends of technology. It aims at creating an environment endowed with communication and computation capabilities. In contrast to desktop computing paradigm, pervasive systems can be
implemented in any device, in any place and at any time. This helps mobile systems to sense their environment and thus to adapt their behavior accordingly. Indeed, the technological progress is contributing in making smart devices as a way to understand the context of collected data and concern activity due to their ability to send, collect, store, process and communicate data [23]. In such cases, the context is associated with three important aspects [30]: where you are; who you are with and what resources are nearby (smart objects, networks, etc.).

This study focuses on efficient access to multimedia documents in context-aware pervasive systems. These documents include media objects of different natures: texts, images, audios, videos, etc. In practice, they are used in many fields such as e-learning, healthcare, news and tourism. In fact, pervasive computing can improve their presentation since it helps ensure multi/cross-device compatibility (e.g. smart TV, laptops, smartphones). In spite of this, as users’ contexts are subject to changes over time, some restrictions may be imposed on proper access to these documents. One possible solution is to adapt their contents so that they conform as much as possible to the current context; below a scenario for illustration.

Sarah is a University student. In the morning, she has to go to the University and wants to consult courses through her smartphone in the bus. These documents contain a mixture of texts and illustrative videos, audios and images. Thus, she may use the cellular network, which is a paid service. The system identifies that: 1) the screen size of the smartphone is not suitable for playing the current content, 2) auditory contents should be avoided in the bus since it is a public place and 3) Sarah should avoid playing large data contents (e.g. videos) to save access to the data through Internet. Immediately, a notification is shown on Sarah’s smartphone suggesting to adapt the document by enlarging and rearranging media objects, converting auditory contents into texts and using low quality media objects.

The above scenario depicts the importance of understanding the context to infer the current users’ contextual constraints and thus to perform useful actions. In the literature, there exist several approaches for multimedia documents adaptation (MDA) in pervasive environments [6,8–12,14,15,17,27–29]. In general, the adaptation processes within these approaches begin with sensing users’ context information. Then, they analyze these information to identify the constraints that make the context non-compliant with the original documents’ features. Finally, they infer the adaptation actions and perform them using adaptation services to provide adapted documents.

Even so, most of the existing works in the field do not address the efficient selection of services among a large set of candidates, especially when quality parameters must be taken into account (e.g. price, reliability, etc.) [6]. As discussed in [7], this aspect is a key-element for improving the quality of service (QoS) in adaptation processes. Practically speaking, when it comes to selecting a subset of required services, most of adaptation approaches, whatever their nature, often carry out this task through a pre-established selection or deterministic algorithms. Currently, the number of adaptation services has become
important so that they are delivered under various forms and models. In particular, for the same functionality, there may exist many instances that differ in the technical details and non-functional features. This makes their selection difficult.

To deal with this issue, we first provide an enriched formulation to model the considered problem. This is a combinatorial problem that depends on the numbers of services and providers, hence the optimization of a function over a set of values. Therefore, we involve quantum-inspired evolutionary algorithms since they allow both local and global search. Also, conventional and compact versions of genetic algorithms are used. The compact versions are adopted in systems where constraints on resources and execution time are considered. The proposal is validated through simulation, experiments and comparisons according to performance, execution time and energy consumption. Overall, the obtained results are promising and encouraging. In fact, although there are several proposals to deal with services selection in other fields (e.g. web/cloud service selection), this work is different since: 1) it uses an enriched problem formulation compared to the ones used in the literature, 2) the solving method involves compact meta-heuristics. To the best of our knowledge, this is the first initiative of its kind to deal with this issue.

The remainder of this paper is organized as follows. Section 2 reviews the state-of-the-art about MDA in context-aware pervasive systems as well as the issues related to the selection of adaptation services. Section 3 provides the problem formulation of the efficient selection of adaptation services. It also gives details about the methods and algorithms proposed for problem solving. Section 4 discusses the experiments and the obtained results. Finally, Sect. 5 gives some concluding remarks and ongoing works.

2 Multimedia Documents Adaptation: State-of-the-Art

2.1 Multimedia Documents Adaptation in Pervasive Systems

MDA in context-aware pervasive systems has been the subject of much research that selects and executes adaptation services to provide adapted documents \[6,8–12,14,15,17,27–29\]. As reported in \[23\], context-aware pervasive systems include three basic elements: sensing, thinking and acting. Next, we detail these elements in the field of MDA. It should be noted that, though there exist several adaptation approaches, they differ only in the way they implement these elements and in the documents’ models they deal with.

- **Sensing:** it is defined as the acquisition of data about the physical world used by the pervasive systems through sources of information like sensors and sensing devices. The context includes several sensed information that can be organized into categories \[23\]: physical context (e.g. noise level), user context (e.g. location), computational context (e.g. battery level) and temporal context (e.g. agenda). These information are arranged according to a context model. For more details about context modeling, the reader should refer to \[33\].
Thinking: the information gathered from the sensors is unprocessed; it should then be analyzed further to take desired actions. The values of the context elements contribute in identifying the constraints impeding the proper execution of documents (see Table 1). The aim is to infer the conflicts for which users’ contexts do not comply with documents’ features, often through “if...else” rules; the semantic web rule language is an example [14,29].

Table 1. Typical context elements, constraints and conflicts

| Context element | Value      | Constraints                  | Conflicts                                      |
|-----------------|------------|-------------------------------|------------------------------------------------|
| Location        | Lab        | Constrained place             | Auditory content should be avoided             |
| Preferred language | English | Unsuitable language          | The user cannot understand the content         |
| Current activity | Driving    | Visual deficiency            | Hands and eyes cannot be used properly         |

Acting: the last step is to infer and execute adaptation actions as per the current conflicts, to provide adapted documents (see Table 2). Each action is carried out through a set of abstract tasks applied to media objects. Then, adaptation plans (paths) are built by binding each abstract task to one service from a repository of adaptation services. These latter are described by specific properties such as service id, input/output parameters, action type (e.g. transmoding, transcoding, transformation, etc.) and quality parameters (e.g. price, reliability, etc.), and implemented using several services’ types (e.g. web and cloud services) [12].

Table 2. Description of typical adaptation actions

| Adaptation actions   | Description                                      |
|----------------------|--------------------------------------------------|
| Exclude audio        | Converting audio objects into texts and, muting and subtitling videos |
| Language translation | Translating the original document from one language to another preferred by the user or to sign language for dumb persons |
| Voice-commands       | Set of vocal commands used when users’ hands cannot be used |

2.2 Classification of Multimedia Adaptation Approaches

Depending on where the decision-making and adaptation actions take place, multimedia adaptation approaches are divided into four categories as well as their hybridization [6]. The choice regarding which category to use depends on many factors such as devices’ computing power, the available resources, etc.
– **Server-side adaptation**: the devices playing the documents represent the client-side that sends adaptation requests to a server. This latter takes charge of the whole operation of documents adaptation (e.g. [29]).
– **Client-side adaptation**: the devices playing the documents are supposed to be able to perform the adaptation process by themselves (e.g. [8]).
– **Proxy-based adaptation**: the adaptation process involves a proxy between the client and the server which acts as a mediator (e.g. [11]).
– **Peer-to-peer adaptation**: the devices playing the multimedia documents may communicate with each other but also with several platforms, to execute adaptation services (e.g. [12]).

### 2.3 Adaptation Services Selection Issues

As reported in [7], the QoS in any given adaptation process is defined through an evaluation of the adaptation services according to two kinds of quality parameters: the *cost* and *benefit*. The cost depends on services’ features such as pricing, response time, etc. The benefit depends on media objects’ features such as image resolution, video speed, etc. Globally, the QoS relies on user’s preferences and its context usage while the quality of outputs is related to the nature of objects.

By reviewing several MDA approaches, we could note that most of their QoS methods focus mainly on the benefit aspect. In other words, they do not care about the efficient selection of adaptation services according to their costs (i.e. adaptation service QoS parameters). Hence, they adopt either a pre-established selection or deterministic algorithms (e.g. [14,15,29]). Actually, a good adaptation system should not aim only at dealing with users’ contexts and preferences, but also optimizing adaptation service QoS values.

Currently, multimedia services are available in different types and in large numbers, covering a wide range of features. We cite as examples Aylien APIs for media intelligence [1], MeaningCloud APIs for text processing [2], Google Cloud for media and entertainment [3] and media services on Amazon Web Services [4]. Such providers offer their services to consumers according to different models such as on-fly, on-demand, real-time, scheduled, etc. The aim is to converge towards everything as a service. This wealth has led to a significant diversity so that for the same functionality, there may exist several candidate services that differ in details and characteristics. It follows that the adaptation process should select adaptation services from a multitude of instances, those that best meet both functional and non-functional features of adaptation plans.

Much research has been done on QoS-aware service composition, in particular for web and cloud services [31,38]. It is defined as an NP-hard problem, usually modeled as a multi-choice and multi-dimension 0–1 knapsack problem. The composition process is carried out following a workflow model that involves abstract descriptions of services. The main forms of workflow structures are sequence, choice, parallel split and loop. For each abstract task, it is asked to select one concrete service among a set of candidates. Several methods are proposed to deal with this issue; we distinguish *static* and *adaptive* approaches [32].
• **Static approaches**: they perform the service composition according to a prior knowledge about QoS values without considering dynamic changes in QoS (e.g. [20, 21, 25, 26, 35, 37]). They belong to three subcategories:
  - **Exact methods**: they seek optimal solutions using deterministic methods such as constraint programming and linear integer programming; however, the computational complexity does not always allow finding them.
  - **Approximate approaches**: they allow finding approximate solutions using heuristics and meta-heuristics; the particle swarm optimization and genetic algorithms are good examples.
  - **Pareto-optimization approaches**: they involve the Pareto optimality which introduces a set of potential solutions that are optimal with respect to one or more objective functions, but not all.

• **Adaptive approaches**: these approaches have extended the service composition problem to finding optimal solutions in cases where QoS values are not known prior (e.g. [22, 24, 34, 36]). They can adapt to changes in QoS values of the service environment. These approaches belong to two subcategories:
  - **Internal composition approaches**: they react to environmental changes by rebuilding a service composition either from ground up or from the point of fault within the composite service. They use several techniques such as artificial intelligence planning techniques and reinforced learning.
  - **External composition approaches**: they use adjustable adapters that bridge the gap between the service workflow and the dynamically changing service environment. They use several techniques such as social network analysis and protocol-based approaches.

3 Contributions

This section introduces our contributions for the efficient selection of adaptation services.

3.1 Context and Motivations

This work deals with the efficient selection of adaptation services so as to improve the QoS in MDA processes. It particularly focuses on the cost aspect by considering the price, response time and reliability. The price depends on the budget allocated for invoking paid services. The response time is linked to the performance of services. Finally, the reliability refers to the degree of maintaining the quality regarding the network and the queries processed by services.

In fact, many researchers have embarked on a frantic race to apply optimization methods to services’ selection paradigm. Certainly, this is a key-element but it should not be an end in itself; because, what is more important is to put such algorithms in contexts. In this line of thinking, our work aims at integrating the optimization of service selection within an adaptation system. As part of
ongoing works, we are working on designing a generic architecture that can be adapted to a wide range of the approaches discussed above. To do so, we involve Multi-Agents Systems (MAS) to model the three basic elements of context-aware pervasive systems. Indeed, MAS are a very good tool for the effective management of environmental changes, in particular sensing, perception, adaptability, communication, autonomy, cooperation, negotiation, intentionality and distribution. These features are advantageous for context-aware pervasive systems that involve properties such as proactiveness, context understanding (surrounding), smartness, mobility, cross-platform, self-tuning, adaptation, etc.. The idea consists in placing the agents depending on the approach’s category adopted while keeping the same communication protocols between them. The service selection methods will then be used once the negotiation process between the providers and consumers agents is finished.

The main contributions of this paper are summarized as follows:

1. **An enriched problem formulation** in the light of the objectives sought as well as solving methods that meet the problem definition. Starting from the fact that existing formulations of the services selection paradigm deal with QoS parameters through benchmarks or probability distributions (simulation) (e.g. [20,25,34,36]), our formulation reinforces the problem definition by modeling the offers provided by services’ providers. This is an influential aspect since it reflects many real-world situations in practice. It is noteworthy that, as the present work does not depend on any specific approach, only sequential composition workflows are considered. The other models can be transformed into the sequential model using transformation techniques [18].

2. **Synthetic methods** for generating adaptation service QoS values according to scenarios based on the degree of competition amongst adaptation services providers. This makes the problem modeling closer to real-world situations; especially that it still lacks benchmarks for making tests and comparisons. Indeed, there are currently few tools dedicated to the description of adaptation services [7]. In fact, most of existing works that deal with the services selection paradigm generate datasets (QoS values) through probability distributions or benchmarks without considering the correlation between these values (e.g. [20,25,34,36]). We argue then that the proposed synthetic methods allow dealing with large, dynamic and various datasets leading to more reliable and deeper analysis and comparisons between solving-methods. On another side, although benchmarks enable making comparisons between solving-methods performances, they allow, however, dealing only with a set of predefined values for QoS parameters. Thus, synthetic methods may be more suitable for solving-methods analysis comparing to benchmarks.

3. **Optimization methods** for providing a composition of services that converges towards the optimal adaptation paths. Depending on the adaptation approach category, the adaptation services’ selection may be performed either by the devices playing the documents or by remote machines. Unlike most of existing services selection methods that focus mainly on optimizing the cost function, in the context of adaptation services selection, further vital concerns
should be taken into account such as the available computational resources and execution time constraints. For instance, although Pareto optimization approaches are expected to achieve better results, they are however unsuitable for devices endowed with limited computational resources since they are generally CPU-time and memory consuming operations. This leads us to care about three aspects: performance, execution time and energy consumption. Accordingly, approximate approaches are more suitable since they vary from local to global search methods using one or multiple solutions, unlike Pareto optimization that considers only the global search. Thus, it is useful to compare between several approximate approaches according to the number of solutions they deal with. In such a context, evolutionary algorithms may excel since they are one of the most adopted approaches for web and cloud service selection [31, 38], in particular that they include both standard and compact versions. The former deal with many solutions; thus, they run generally on systems that are not subject to specific constraints. The latter deal with very few number of solutions; thus they can run on systems for which constraints on resources (CPU, memory, battery, etc.) and execution time are considered. They also show a simplicity of implementation and ease in setting parameters [19]. In fact, the current number of optimization methods is very large. It would then be possible for us to consider other approaches expect that the time constraints did not allow us to test them all. We keep this point for future works.

3.2 Problem Statement and Modeling

Now, we formulate the problem of efficient selection of adaptation services. For this purpose, we adopt some of the basic concepts and notations used in [18].

1. Service class: we denote by \( S = \{S_1, S_2, \ldots, S_k\} \) the set of \( k \) abstract tasks composing the abstract composite service (ACS) inferred by the adaptation process, where \( S_i = \{1..k\} \in S \) refers to a single task;
2. Concrete service: each abstract task \( S_i \in S \) represents a functionality that can be implemented by one concrete service among a subset of candidates, denoted by \( I_i = cs_{ij} \), from the repository of all available adaptation services. The concrete composite service (CCS) (adaptation path) is an instance of the ACS that binds each abstract task \( S_i \in S \) to a concrete service \( cs_{ij} \in I_i \).
3. Service provider: Each concrete service \( cs_{ij} \in I_i=1..k \) comes from one and only one provider. The set of all providers is denoted by \( P = \{P_1, P_2, \ldots, P_m\} \), where \( m \) is the number of providers. For each provider \( P_u=1..m \), the adaptation services offered are grouped into a set denoted by \( SP_u = \{csp_{uv}\} \).
4. QoS criteria: three non-functional criteria are considered: the price, response time and reliability. For any given concrete service \( cs_{ij} \in I_i=1..k \), these criteria are defined as a vector \( Q(cs_{ij}) = [q_p(cs_{ij}), q_t(cs_{ij}), q_r(cs_{ij})] \), which represents the price, response time and reliability, respectively. The QoS criteria of the CCS are defined as a vector \( Q(CCS) = [q_p(CCS), q_t(CCS), q_r(CCS)] \) representing the price, response time and reliability, respectively.
5. **Fitness functions**: it aims at finding the optimal CCS by selecting services that minimize functions $f_{p,t}$ and maximize $f_r$, as given in equations (1,2,3):

$$f_{p,t}(x) = \sum_{i=1}^{K} \sum_{cs_{ij} \in I_i} x_{ij} \times q_{p,t}(x_{ij}) - g_{p,t}(x)$$ (1)

with \( g_{p,t}(x) = \sum_{u=1}^{m} g_u(x) \)

$$f_r(x) = \prod_{i=1}^{k} \sum_{cs_{ij} \in I_i} x_{ij} \times q_r(cs_{ij})$$ (2)

Subject to:

$$f_{p,t}(x) \leq C_{max}, \quad f_r(x) \geq R_{min} \quad \text{and} \quad \sum_{cs_{ij} \in I_i} x_{ij} \leq 1$$ (3)

where \( x = ((x_{11}, x_{12}, \ldots, x_{1|i_1}), (x_{21}, x_{22}, \ldots, x_{2|i_2}), \ldots, (x_{k|I_k})) \), \( x_{ij} \in \{0, 1\} \) indicates the absence / presence of concrete services \( cs_{ij} \in I =_{1..k} \) in the CCS. \( C_{max} \in \{P_{max}, T_{max}\} \) is a maximum cost for the price and response time not to be exceeded while \( R_{min} \) is a minimum lower bound for the reliability. In addition, to each provider \( P_{u=1..m} \) is assigned an extra profit, denoted by \( g_u \). It is earned in accordance with the price and response time of services \( cs_{pu} \in SP_{u=1..m} \); details are given on Sect. 3.3. For example, if the consumer invokes multiple services coming from provider \( P_u \), this latter may make reduction on the total price or offer more resources and privileges to improve the response time. This increases the degree of competition amongst the providers. To evaluate the quality of the CCS, the Simple Additive Weighting (SAW) technique is used [18]. It converts vector \( Q(CCS) \) into a single normalized real value as follows:

$$w_1 \frac{Q_{max,p}(ACS) - f_p(x)}{Q_{max,p}(ACS)} + w_2 \frac{Q_{max,t}(ACS) - f_t(x)}{Q_{max,t}(ACS)} + w_3 \frac{f_r(x) - Q_{min,r}(ACS)}{1 - Q_{min,r}(ACS)}$$ (4)

Where \( w_1 + w_2 + w_3 = 1 \).

\( Q_{max} \) and \( Q_{min} \) represent the minimum and maximum values of QoS criteria in all possible instances of the ACS, calculated as shown in equations (5,6).

$$Q_{min,r} = \prod_{i=1}^{k} \min_{cs_{ij} \in I =_{1..k}} q_p \times (CS_{ij})$$ (5)

$$Q_{max,p,t} = \sum_{i=1}^{k} \max_{cs_{ij} \in I =_{1..k}} q_{p,t} \times (CS_{ij})$$ (6)
3.3 Solving Methods

For large problem’s instances, approximate and Pareto optimization approaches are more suitable. In this case, for each candidate solution \( x \), functions \( f_{p,t,r} \) are calculated according to a vector \( x' \) obtained through a repair function. This latter withdraws some concrete services from vector \( x \) until the constraints on price, response time and reliability become satisfied (\( P_{\text{max}}, T_{\text{max}} \) and \( R_{\text{min}} \)). Two repair methods are proposed: random repair and greedy repair.

- **Random repair (Rep1)**: the repair function withdraws randomly concrete services from vector \( x \) as much as necessary until there is no cost overrun.
- **Greedy repair (Rep2)**: the repair function withdraws concrete service \( cs_{ij} \) which maximizes \( |Q(x) - Q(x')| \). The process is repeated as much as necessary until there is no cost overrun. In this way, we keep a maximum number of services hoping that we provide good quality adapted documents.

All that remains is to define the gain function \( g \). Three methods are proposed: *No gain*, the gain based on exemption and the gain based on cost reduction.

- **No gain**: Providers do not offer any benefit based on the services requested.
- **The gain based on exemption (buy \( n \), get one free)**: each provider \( P_u \in P \) may make exemptions by offering the lower-cost service for free if the customer requests more than \( n \) services from set \( S_P u \).
- **The gain based on cost reduction (buy \( n \), benefit from cost reduction)**: each provider \( P_u \in P \) may offer a discount (cost reductions) for the customer according to a rate \( DR \in [0, 1] \) if more than \( n \) services are requested from set \( S_P u \).

3.4 Synthetic Methods to Simulate Adaptation Service QoS Values

To simulate adaptation service QoS values, we need methods to generate them. To this end, we inspire from the methods used in studying the knapsack problem [16]. The difficulty of problem-solving depends on the number of abstract tasks and the available concrete services. It also depends on the degree of correlation between QoS values; the more they are correlated (i.e. close each other’s), the more the problem-solving is expected to be difficult. In addition, if each provider offers multiple concrete services to consumers, then the selection will be even more difficult since it leads competitive offers between providers.

Three types of correlations between adaptation service QoS values are proposed: *uncorrelated costs*, *weakly correlated costs* and *strongly correlated costs*, as given on Table 3. Let \( r \) and \( v \) be two positive parameters chosen empirically. Also, let \( q_{p,1..k}, q_{t,1..k} \) and \( q_{r,1..k} \) be random values generated following a uniform distribution over the interval \([1, v]\) (denoted by uniform \((1, v)\)).

Higher correlation leads to smaller value for the difference: \( \{\max_{i=1..k} \{q_{p,t,r}(cs_{ij})\} - \min_{i=1..k} \{q_{p,t,r}(cs_{ij})\} \} \), and thus for the standard deviation. Note that in the case of *weak correlation*, the price, response time and reliability of concrete services \( cs_{ij} \in I_i=1..k \) related to each abstract task \( S_i=1..k \in S \) are...
Table 3. Synthetic methods for generating adaptation service QoS values

| Correlation types        | Price $q_p(cs_{ij})$ | Response time $q_t(cs_{ij})$ | Reliability $q_r(cs_{ij})$ |
|--------------------------|-----------------------|-------------------------------|-----------------------------|
| Uncorrelated             | uniform $(1, v)$      | uniform $(1, v)$              | uniform $(1, v)/v$          |
| Weak correlation 1       | $q_p + \text{uniform}(-r, r)$ | $q_r + \text{uniform}(-r, r)$ | $q_r + \text{uniform}(-r, r)/(v + r)$ |
| Weak correlation 2       | $q_p(cs_{ij}) + \text{uniform}(-r, r)$ | $q_r(cs_{ij}) + \text{uniform}(-r, r)/(q_p(cs_{ij}) + r)$ | $q_r(cs_{ij}) + \text{uniform}(-r, r)/(q_p(cs_{ij}) + r)$ |
| Strong correlation       | $q_p(cs_{ij}) + r$    | $q_r(cs_{ij}) + r$            | $q_r(cs_{ij}) + r + \text{uniform}(0, r)$ |

Constraints types

| Constraints types | $P_{max}$ | $T_{max}$ | $R_{min}$ |
|-------------------|-----------|-----------|-----------|
| Unbounded cost ($C_{max 1}$) | $+\infty$ | $+\infty$ | 0         |
| Average cost ($C_{max 2}$) | $\frac{1}{2}Q_{max 1}(CCS)$ | $\frac{1}{2}Q_{max 2}(CCS)$ | $\sqrt{Q_{min 2}(CCS)}$ |

weakly correlated. In the case of weak correlation 2 and strong correlation, the response time and reliability of any given concrete service $cs_{ij}$ are correlated with its price.

On another side, by using the unbounded cost option, the solution will include all concrete services requested (there will be no constraints on costs). Otherwise, it will include about half of the services selected, which influences negatively on the adapted content with respect to the current context.

3.5 Finding Approximate Solutions Using Evolutionary Algorithms

Evolutionary algorithms (EA) are a class of meta-heuristics that have proven to be effective for finding approximate solutions to several optimization problems. They deal with populations of candidate solutions (individuals) encoded in accordance with the current problem. The evolution process follows an iterative procedure where at each iteration, the individuals are evaluated according to a fitness function, selected and recombined in order to generate new population. The subclasses of EA differ in the way they encode the individuals as well as in the way they implement the selection and recombination operators.

In the present work, three EAs are used: quantum-inspired evolutionary algorithms (QIEA) [16], genetic algorithms (GA) [16] and compact genetic algorithms (CGA) [19]. QIEAs combine principles inspired from EA and quantum computing. They encode the solutions as quantum registers composed of qubits (superposition of states), which is, in fact, a probabilistic model. The recombination of individuals is performed through quantum operators such as the measure and the interference. QIEAs allow both local and global search over solutions’ spaces (they can work on populations of size ranging from one to many solutions). GAs are based on principles inspired from natural selection and modern genetics theories. They encode the individuals as chromosomes that are recombined through crossover and mutation operators. Unlike standard GAs, a CGA uses estimated probability density functions to manipulate a virtual population encoded as a probability vector, which represents in fact a compacted population of size $N_p$. At each iteration, two chromosomes are randomly generated using
the probability vector and evaluated. The pair of individuals \([\text{winner}, \text{loser}]\) is then used to update the probability vector. To further improve the performances, we propose to apply the crossover and mutation operators to the pair \([\text{winner}, \text{loser}]\). Details about these EAs are omitted for space reasons (further details can be found in [16,19]).

Each candidate solution \(x = ((x_{11}, \ldots, x_{1|I_1|}), (x_{21}, \ldots, x_{2|I_2|}), \ldots, (x_{1k}, \ldots, x_{k|I_k|}))\), \(x_{ij} \in \{0, 1\}\) is encoded as a binary chromosome composed of \(k\) genes; \(k\) is the number of abstract tasks in the inferred ACS [21]. Each gene \(g_{i=1..k}\) corresponds to service class \(S_i\). The length of gene \(g_i\) depends on the number of services in set \(I_i\). Each chromosome is evaluated according to the formulas and methods presented above. The structure of chromosomes allows building quantum registers and probability vectors that keep the same form by substituting only every bit by a qubit and a probability value, as shown in Fig. 1.

4 Experiments, Numerical Results and Discussion

Our proposal is validated by making experiments, gathering the results and analyzing them. For this purpose, four instances of EA - denoted by \#GA, \#CGA, \#QIEA\(_1\) \& \#QIEA\(_2\) - are executed for a sufficient number of runs (over 50 runs) by covering all possible cases. Each of them is iterated for a maximum number of 100 iterations. We mention that by performing more than 50 runs, this made no difference. The hardware and software configurations are as follows: laptop endowed with \(i7\ 3537U\ CPU\ (2.5\ GHz)\) and \(8\ Gb\) of memory, the Java language and the Eclipse IDE. The parameters settings are given below.

- **Optimization problem parameters**: number of abstract tasks \(k = 8\); number of concrete services \(nb = 70\); number of providers \(m = 15\); \(v = 10\); \(r = 5\); \(w_{i=1..3} = 0.5, 0.4\) and \(0.1\), respectively; the gain is based on cost reduction with discount rates \(DR_{p,t}\) generated as uniform \((0.1, 0.2)\); threshold for discounting \(n = 3\).
- **\#QIEA\(_{1,2}\) parameters**: the population size \(N_p\) is equal to 5 for \#QIEA\(_1\) and to 1 for \#QIEA\(_2\). The initial amplitudes of qubits \((\alpha_i, \beta_i)\) are set to \(\sqrt{1/2}\). The interference operator is the same as in [13].
- **#CGA and #GA parameters**: both virtual and real populations are of size $N_p = 52$. In addition, instances #CGA and #GA use elitist selection, one cut-point crossover, swap mutation with rate of 2% and total replacement.

The parameters settings were configured experimentally. In fact, by executing several instances that differed in parameters’ values and scenarios regarding providers’ gain policies, the results did not show us any improvements. Tables 4 and 5 show the fitness and execution times for each EA instance; $b$, $m$ and $w$ refer to best, mean and worst results recorded over the 50 runs, respectively.

**Table 4. Performances of GA, CGA and QIEA (%).**

| Correlation       | Cost type | #GA Rep1 | #GA Rep2 | #CGA Rep1 | #CGA Rep2 | #QIEA1 Rep1 | #QIEA1 Rep2 | #QIEA2 Rep1 | #QIEA2 Rep2 |
|-------------------|-----------|----------|----------|-----------|-----------|-------------|-------------|-------------|-------------|
| No correlation    | $C_{max1}$ | b        | 58.03    | 58.04    | 59.83     | 57.77       |
|                   |           | m        | 51.09    | 53.40    | 56.97     | 53.81       |
|                   |           | w        | 46.37    | 49.84    | 53.22     | 49.62       |
|                   | $C_{max2}$ | b        | 74.71    | 76.83    | 77.68     | 78.87       | 74.54       | 77.68       |
|                   |           | m        | 64.59    | 65.80    | 68.97     | 69.72       | 65.81       | 69.14       |
|                   |           | w        | 60.46    | 61.97    | 64.12     | 64.72       | 60.58       | 61.59       |
| Weak-correlation 1| $C_{max1}$ | b        | 53.32    | 56.31    | 57.26     | 55.56       |
|                   |           | m        | 46.86    | 50.36    | 53.27     | 50.39       |
|                   |           | w        | 42.82    | 46.58    | 49.80     | 46.63       |
|                   | $C_{max2}$ | b        | 76.19    | 78.58    | 78.58     | 77.90       | 78.92       | 76.84       |
|                   |           | m        | 66.93    | 69.38    | 71.07     | 72.13       | 76.85       | 68.68       |
|                   |           | w        | 61.38    | 64.35    | 66.58     | 67.24       | 72.39       | 63.48       |
| Weak-correlation 2| $C_{max1}$ | b        | 58.67    | 59.84    | 62.18     | 60.96       |
|                   |           | m        | 52.84    | 56.03    | 58.98     | 55.96       |
|                   |           | w        | 48.25    | 53.32    | 56.65     | 52.31       |
|                   | $C_{max2}$ | b        | 79.83    | 83.87    | 82.74     | 84.09       | 84.09       | 84.09       |
|                   |           | m        | 69.17    | 77.71    | 75.74     | 77.34       | 82.92       | 80.94       |
|                   |           | w        | 59.71    | 71.71    | 69.30     | 70.32       | 80.22       | 64.69       |
| Strong correlation| $C_{max1}$ | b        | 54.66    | 57.45    | 61.63     | 59.63       |
|                   |           | m        | 45.29    | 50.50    | 56.41     | 51.66       |
|                   |           | w        | 39.75    | 44.90    | 51.22     | 40.37       |
|                   | $C_{max2}$ | b        | 81.58    | 85.49    | 81.63     | 84.87       | 83.98       | 85.49       |
|                   |           | m        | 68.79    | 79.02    | 75.44     | 82.73       | 77.52       | 84.46       |
|                   |           | w        | 62.49    | 71.05    | 69.59     | 79.25       | 69.47       | 82.38       |
Table 5. Execution time for GA, CGA and QIEA (recorded in seconds)

| Correlation | Cost type | #GA | #CGA | #QIEA1 | #QIEA2 |
|-------------|-----------|-----|------|--------|--------|
|             |           | Rep1| Rep2 | Rep1   | Rep2   | Rep1   | Rep2   | Rep1 | Rep2   |
| No correlation | $C_{max 1}$ | b   | 0.1880 | 0.0280 | 0.0340 | 0.0060 |
|              |           | m   | 0.1948 | 0.0300 | 0.0359 | 0.0070 |
|              |           | w   | 0.2250 | 0.0490 | 0.0410 | 0.0080 |
|             | $C_{max 2}$ | b   | 0.2530 | 1.8630 | 0.0360 | 0.0440 | 0.3900 | 0.0090 | 0.0770 |
|              |           | m   | 0.2618 | 2.0552 | 0.0382 | 0.0466 | 0.4203 | 0.0093 | 0.0863 |
|              |           | w   | 0.2920 | 2.5520 | 0.0430 | 0.3500 | 0.0520 | 0.4590 | 0.0110 | 0.0960 |

Discussion and Analysis

Table 4 shows that CGA and QIEA perform better than GA, with a preference to #QIEA1. By analyzing the behavior of these algorithms, it was found that the probabilistic models of CGA and QIEA are more suitable for the exploration and exploitation of the search space. Table 4 draws also our attention to the effect of the repair mechanisms on the evolutionary process within all EAs. Indeed, one can compare the results obtained when considering $C_{max 1}$ (repair methods are not used since the costs are unbounded) and $C_{max 2}$ (repair methods are involved since the costs are restricted) in all correlation methods. In particular, the greedy repair shows more efficiency than random repair since the services are selectively withdrawn. We point out that, in the case of $C_{max 1}$, the fitness is negatively influenced by the reliability value that decreases when the number of selected services grows.

Overall, the execution time for GA is greater than for CGA and QIEA. This is expected since the number of individuals evaluated at each iteration is about of $N_p/2$ while #CGA and #QIEA1 deal only with 4 and 5 individuals, respectively; especially that the largest portion of the execution time has elapsed in the evaluation phase. Another observation is that the execution time in greedy repair method is greater than in random repair. This is because the greedy repair makes many tests before withdrawing services, which increases the computational complexity.

Regarding the energy consumption ($E$), Abdelhafez et al. showed in a recent work [5] that for sequential GAs, it is directly proportional to time ($T$) and power ($P$): $E \approx P \times T$. According to Table 5, one observes that: $T_{GA} \approx a \times TC_{GA} \approx b \times T_{QIEA1} \approx c \times T_{QIEA2}$; $a$, $b$ and $c$ are constants greater than 1. As a result, the energy consumption can be rewritten as follows: $E_{GA} \approx a \times E_{CGA} \approx b \times E_{QIEA1} \approx c \times T_{QIEA2}$. Undoubtedly, CGA and QIEA may really consume less energy compared to GA.

In view of the foregoing, #CGA and #QIEA are more efficient than #GA while consuming less energy and computational resources. This makes them suitable to the adaptation approaches categories discussed above, in particular when these algorithms are performed by mobile and battery-based devices. Actually, we could not do more in-depth analysis due to space reasons.
Finally, we assess our proposal against other research-works. Unfortunately, the performances of our algorithms could not be compared with other approaches due to the lack of benchmarks. In addition, the adopted problem formulation is different. The main advantages are summarized as follows:

- Most of existing approaches dedicated to services selection paradigm target mainly the optimization of QoS parameters and the improvement of execution time (see for instance [20,25,26,34–36]). This is necessary but not sufficient on its own. For instance, the authors in [26] have made a comparative study of many-objective evolutionary algorithms for efficient composition of QoS-aware web services. Even though these approaches could achieve better results, the excessive computational cost could still limit the general adoption of such algorithms, in particular for devices endowed with limited computational resources. Thus, our proposal considers further aspects such as the available computational resources and time constraints which are essential in the context of MDA.
- Although there are several works applying metaheuristics for problems dealing with energy minimization in a target scenario such as energy consumption of web and cloud services (e.g. [35]), very few works focus on analyzing energy consumption of metaheuristics themselves [5]. This is relevant for green computing since the adaptation services’ selection may be frequently performed by MDA processes, especially when the number of users grows.
- The proposal can be viewed as a complementary element to existing MDA approaches since most of them do not take into account the cost aspect.
- The proposal shows adaptability insofar as it treats the service selection problem at a high level of abstraction, regardless of any specific approach.
- The proposal is flexible since it allows several options for selecting adaptation services in accordance with the available computational resources.

Our proposal shows also some limitations which are recapped as follows:

- The proposed methods belong to static service selection approaches and thus they do not show adaptation to changes in QoS values over time.
- The collection of services involves the creation of a repository that must be maintained continuously. This may require service discovery mechanisms.

Nevertheless, as this study has not yet reached its end, we plan to overcome these limitations using MAS, which is actually a work that we have already started.

5 Conclusion

The aim of this paper was to propose new mechanisms for the efficient selection of adaptation services used by MDA processes. For this purpose, the problem was first modeled as an optimization problem following an enriched formulation. Then, synthetic methods are proposed to generate QoS values due to the lack of benchmarks. Finally, standard and compact versions of evolutionary algorithms are used in order to efficiently select adaptation paths according to price, response time and reliability of services. The proposal was validated through
simulations and experiments. The numerical results showed us that CGA and QIEA are more efficient than GA in terms of performance and execution time while consuming less energy and computational resources.

Currently, we are working on design a generic MAS-based architecture that can be configured to several adaptation approaches. The agent paradigm is involved to model the elements of context-aware pervasive systems namely sensing, thinking and acting as well as the efficient management of adaptation services. We will also perform more experiments by implementing, analyzing and comparing several services selection approaches ranging from Pareto-based to exact methods, with a special focus on those involving MAS to deal with changes in services’ environments (e.g. [25,37]).

Acknowledgments. We would like to thank the Direction Generale de la Recherche Scientifique et du Developpement Technologique (DGRSDT) in Algeria, for supporting this research work. Also, the authors would like to thank Dr. Amer Draa from MISC Laboratory - University of Constantine 2, Algeria, for the feedback and discussions on optimization concerns and issues.

References

1. https://aylien.com/media-monitoring-api/
2. https://www.meaningcloud.com/
3. https://cloud.google.com/solutions/media-entertainment
4. https://aws.amazon.com/fr/media-services/
5. Abdelhafez, A., Alba, E., Luque, G.: A component-based study of energy consumption for sequential and parallel genetic algorithms. J. Supercomput. 75(10), 6194–6219 (2019). https://doi.org/10.1007/s11227-019-02843-4
6. Adel, A., Philippe, R., Sébastien, L.: Multimedia documents adaptation based on semantic multi-partite social context-aware networks. Int. J. Virtual Commun. Soc. Netw. (IJVCSN) 9(3), 44–59 (2017)
7. Adel, A., Sébastien, L., Philippe, R.: Enrich the expressiveness of multimedia document adaptation processes. In: Semantic Multimedia Analysis and Processing, pp. 185–218. CRC Press (2017)
8. Belhadad, Y., Refoufi, A., Roose, P.: Spatial reasoning about multimedia document for a profile based adaptation. Multimedia Tools Appl. 77(23), 30437–30474 (2018). https://doi.org/10.1007/s11042-018-6080-8
9. Bettou, F., Boufaida, M.: An adaptation architecture dedicated to personalized management of multimedia documents. Int. J. Multimedia Data Eng. Manage. (IJMDEM) 8(1), 21–41 (2017)
10. Derdour, M., Roose, P., Dalmau, M., Ghoualmi-Zine, N.: An adaptation platform for multimedia applications CSC (component, service, connector). J. Syst. Inf. Technol. 14(1), 4–22 (2012)
11. Dromzée, C., Laborie, S., Roose, P.: A semantic generic profile for multimedia document adaptation. In: Intelligent Multimedia Technologies for Networking Applications: Techniques and Tools; Techniques and Tools, pp. 225–246 (2013)
12. Hai, Q.P., Laborie, S., Roose, P.: On-the-fly multimedia document adaptation architecture. Procedia Comput. Sci. 10, 1188–1193 (2012)
13. Han, K.H., Kim, J.H.: Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. IEEE Trans. Evol. Comput. 6(6), 580–593 (2002)
14. Jannach, D., Leopold, K.: Knowledge-based multimedia adaptation for ubiquitous multimedia consumption. J. Netw. Comput. Appl. 30(3), 958–982 (2007)
15. Khallouki, H., Bahaj, M.: Multimedia documents adaptive platform using multi-agent system and mobile ubiquitous environment. In: 2017 Intelligent Systems and Computer Vision (ISCV), pp. 1–5. IEEE (2017)
16. Laboudi, Z., Chikhi, S.: Comparison of genetic algorithm and quantum genetic algorithm. Int. Arab J. Inf. Technol. 9(3), 243–249 (2012)
17. Lakehal, A., Alti, A., Laborie, S., Philippe, R.: Ontology-based context-aware recommendation approach for dynamic situations enrichment. In: 2018 13th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP), pp. 81–86. IEEE (2018)
18. Le, D.-N., Nguyen, G.N.: A new ant-based approach for optimal service selection with E2E QoS constraints. In: Intan, R., Chi, C.-H., Palit, H.N., Santosso, L.W. (eds.) ICSIIT 2015. CCIS, vol. 516, pp. 98–109. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46742-8_9
19. Lee, J.Y., Kim, M.S., Lee, J.J.: Compact genetic algorithms using belief vectors. Appl. Soft Comput. 11(4), 3385–3401 (2011)
20. Li, C., Li, J., Chen, H.: A meta-heuristic-based approach for QoS-aware service composition. IEEE Access 8, 69579–69592 (2020)
21. Li, Y., Yao, X., Liu, M.: Cloud manufacturing service composition optimization with improved genetic algorithm. Math. Probl. Eng. NA 2019, 1–19 (2019)
22. Liu, J.W., Hu, L.Q., Cai, Z.Q., Xing, L.N., Tan, X.: Large-scale and adaptive service composition based on deep reinforcement learning. J. Vis. Commun. Image Represent. 65, 102687 (2019)
23. Mahalle, P.N., Dhotre, P.S.: Context-aware pervasive systems. Context-Aware Pervasive Systems and Applications. ISRL, vol. 169, pp. 49–66. Springer, Singapore (2020). https://doi.org/10.1007/978-981-32-9952-8_3
24. Medeiros Campos, G.M., Souto Rosa, N., Ferreira Pires, L.: Adaptive service composition based on runtime verification of formal properties. In: Proceedings of the 50th Hawaii International Conference on System Sciences (2017)
25. Naseri, A., Navimipour, N.J.: A new agent-based method for QoS-aware cloud service composition using particle swarm optimization algorithm. J. Ambient Intell. Hum. Comput. 10(5), 1851–1864 (2019). https://doi.org/10.1007/s12652-018-0773-8
26. Ramírez, A., Parejo, J.A., Romero, J.R., Segura, S., Ruiz-Cortés, A.: Evolutionary composition of QoS-aware web services: a many-objective perspective. Expert Syst. Appl. 72, 357–370 (2017)
27. Saighi, A., Laboudi, Z.: A novel self-organizing multi agent-based approach for multimedia documents adaptation. In: 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), pp. 1–6. IEEE (2020)
28. Saighi, A., Laboudi, Z., Roose, P., Laborie, S., Ghoualmi-Zine, N.: On using multiple disabilities profiles to adapt multimedia documents: a novel graph-based method. Int. J. Inf. Technol. Web Eng. (IJITWE) 15(3), 34–60 (2020)
29. Saighi, A., Philippe, R., Ghoualmi, N., Laborie, S., Laboudi, Z.: HaMA: a handicap-based architecture for multimedia document adaptation. Int. J. Multimedia Data Eng. Manage. (IJMDEM) 8(3), 55–96 (2017)
30. Schilit, B., Adams, N., Want, R.: Context-aware computing applications. In: 1994 First Workshop on Mobile Computing Systems and Applications, pp. 85–90. IEEE (1994)
31. She, Q., Wei, X., Nie, G., Chen, D.: QoS-aware cloud service composition: a systematic mapping study from the perspective of computational intelligence. Expert Syst. Appl. 138, 112804 (2019)
32. Shehu, U.G., Epiphaniou, G., Safdar, G.A.: A survey of QoS-aware web service composition techniques. Int. J. Comput. Appl. 89, 10–17 (2014)
33. Strang, T., Linnhoff-Popien, C.: A context modeling survey. In: Workshop on Advanced Context Modelling Reasoning and Management (2004)
34. Thangaraj, P., Balasubramanie, P.: Meta heuristic QoS based service composition for service computing. J. Ambient Intell. Hum. Comput., 1–7 (2020). https://doi.org/10.1007/s12652-020-02083-y
35. Wang, S., Zhou, A., Bao, R., Chou, W., Yau, S.S.: Towards green service composition approach in the cloud. IEEE Trans. Serv. Comput. 99, 1–14 (2018)
36. Yuan, Y., Zhang, W., Zhang, X., Zhai, H.: Dynamic service selection based on adaptive global QoS constraints decomposition. Symmetry 11(3), 403 (2019)
37. Zertal, S., Batouche, M., Laboudi, Z.: A novel hybrid optimization-based approach for efficient development of business-applications in cloud. Int. J. Inf. Syst. Serv. Sect. (IJISSS) 12(4), 14–35 (2020)
38. Zhao, X., Li, R., Zuo, X.: Advances on QoS-aware web service selection and composition with nature-inspired computing. CAAI Trans. Intell. Technol. 4(3), 159–174 (2019)