Bus Inf Syst Eng 60(5):415–430 (2018)
https://doi.org/10.1007/s12599-018-0552-2

RESEARCH PAPER

Multi User Context-Aware Service Selection for Mobile Environments

A Heuristic Technique

Michael Bortlik · Bernd Heinrich · Michael Mayer

Received: 31 October 2017 / Accepted: 8 July 2018 / Published online: 20 July 2018

Abstract Modern service systems build on top of service dominant designs which encompass contextualization (value-in-context) and collaboration (value-in-use) between users and service providers. Processes in this domain often require the consideration of both context information (e.g., location or time of day) and multiple participating users where each user probably has its own preferences and constraints (e.g., restricted overall budget). However, selecting a suitable service provider for each action of a process, especially when some of these actions are conducted together by several users, can be a complex decision problem in multi user context-aware service systems. Consequently, exact approaches are not fit to solve such a service selection problem in appropriate time. Thus, the paper proposes a heuristic technique applying a decomposition of the users' global constraints and a local service selection. In this way, the aim is to determine a feasible service composition for each participating user while taking the users' individual preferences and constraints as well as context information into account. The evaluation of the heuristic technique shows, based on a real-world scenario in the tourism domain, that the proposed approach is able to achieve close-to-optimal solutions while efficiently scaling with problem size and therefore can support decision makers in multi user context-aware service systems.

Keywords Service selection · Context-aware · Multi user processes · Mobile environments · Service systems engineering · Heuristic technique

1 Introduction

Based on a progressive development of new mobile technologies (e.g., smartphone, tablet), applications and services, mobile business has gained in importance during the last decade (Statista 2017a, b). Businesses started capitalizing on this development to deliver their goods and services tailored to a new mobile lifestyle (Kabir and Akhtar Hasin 2011). The development also supports the construction of service systems (cf. Alter 2012) in form of a context-aware interplay of stationary and mobile devices, services and users (Zaplata et al. 2009). Thus, individual and professional processes conducted by multiple users jointly using services accessible anywhere and anytime are enabled (Zaplata et al. 2009; Deng et al. 2016). In this regard, context information such as location, time of day or temperature that can be gathered via the sensory capabilities of modern mobile devices play a vital role as they allow for a high degree of individualization (cf. Zhang et al. 2009; Böhmann et al. 2014; Amin et al. 2016). In general, context information can be defined as "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves" (Dey 2001, p. 5).

Accepted after two revisions by the editors of the special issue.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s12599-018-0552-2) contains supplementary material, which is available to authorized users.

M. Bortlik · Prof. Dr. B. Heinrich (✉) · M. Mayer
University of Regensburg, Universitätsstr. 31,
93053 Regensburg, Germany
e-mail: Bernd.Heinrich@ur.de

Modern service systems, especially those focusing on context-awareness, build on top of service-dominant designs (Edvardsson et al. 2011; Alter 2012; Böhmann et al. 2014). Service-dominant designs encompass contextualization, value bound to a given context (value-in-context), and collaboration, which describes the process of value co-creation and co-consumption (value-in-use). Co-creation resp. co-consumption in this regard means that the value of a considered service is created by service providers and users together resp. employed by multiple users together (Vargo and Lusch 2004; Grönroos 2011). This means that some actions from otherwise possibly different processes of users have to be conducted simultaneously together by more than one user. This can be necessary due to the needs or preferences of users aiming to use and employ special customer services with additional value. Processes containing actions that have to be conducted simultaneously together by multiple users are called *multi user context-aware processes* in the following and are in the focus of this paper.

In mobile environments the amount of available information can easily overwhelm a user and lead to an information overload problem (Zhang et al. 2009; Shen et al. 2012a). Thus, users require decision support to find and use the appropriate service for a given context. This issue is further intensified when considering multiple users and their individual preferences and needs. From this viewpoint, the underlying decision problem can be characterized as a selection problem. Here, we aim to select the most suitable services while dealing with individual context information of users and their coordination (e.g., to conduct actions simultaneously together). Our aim is to provide a selection method that can be used by service systems (Alter 2012) for multiple users with the consideration of context information in mobile environments.

Providing such a method and thus supporting a corresponding service system can be valuable in different application fields. For example, information service providers such as Yelp or TripAdvisor can be used to retrieve informational entities (and thus service offers) about restaurants, museums, sights or other real-world objects. In this regard, a multi user context-aware process in the tourism domain can represent a city day trip by a group of users (tourists). The providers (i.e., Yelp, TripAdvisor, etc.) of the informational entities referring to real-world objects and the users with their mobile devices form the participants of a service system (cf. Maglio et al. 2006; Alter 2012; Lyons and Tracy 2013; Frost and Lyons 2017). Moreover, a city day trip (process) consists of multiple actions (e.g., having lunch, visiting a museum), where some actions may be conducted together by several users, requiring the coordination of the users' actions. Furthermore, each action can be supported by different informational entities referring to real-world objects that differ only in attributes such as price or selection of food. To find favorable entities, the individual preferences (e.g., for a certain user, price could be more relevant than selection of food) and global end-to-end constraints (e.g., restricted overall budget) of each user need to be taken into account. Additionally, context information such as business hours and location (of both users and informational entities) is also highly relevant. The huge amount of informational entities that can support a certain action (e.g., almost 10,000 restaurants can be visited in the city of Berlin, Germany[1]) further contributes to an information overload problem, requiring that users need to be supported in their decision making (a discussion of further application fields of multi user context-aware service systems like healthcare or disaster relief assistance can be found in the Online Appendix 1).

The underlying decision problem can be characterized as a service selection problem: By modeling and representing real-world objects as informational entities or service objects, service selection methods can be utilized as decision support. More precisely, a multi user context-aware service selection problem can be defined as the problem of determining the specific composition of service objects for each user that fits best to the users' current context as well as to their individual preferences and global constraints regarding non-functional properties (NFP) such as price and duration (cf. Zeng et al. 2004; Yu et al. 2007; Alrifai et al. 2012). Furthermore, if there exist any dependencies between the service compositions of individual users, the coordination of the users' actions is also required. This includes the mandatory simultaneous use of one or more service objects by several users together (cf. Wang et al. 2010; Wanchun et al. 2011; Benouaret et al. 2012), which will be in the focus of this work.

In mobile environments, often a huge number of different alternative service objects for the realization of each action of a process (cf., e.g., the tourism domain) exists. This leads to large service selection problems. Due to the NP-hardness of the underlying decision problem, finding an optimal solution in appropriate time is not practical, as computation time increases exponentially subject to problem size (cf. Zhang et al. 2012; Moghaddam and Davis 2014; Zhang et al. 2016). These performance issues are amplified by the increased problem complexity that results from considering multiple users and context information (cf. Lewerenz 2015). This is due to existing dependencies within a user's service composition (resulting from context information) as well as between different service compositions of several users (resulting, e.g., from the mandatory

[1] https://www.yelp.com/search?cflt=restaurants&find_loc=Berlin%2C+Germany. Accessed 25 June 2018.

simultaneous use of service objects by several users) need to be taken into account. To address these issues, the use of a heuristic technique instead of an exact service selection approach is envisioned. Existing works presenting exact service selection approaches also state the need to develop heuristic techniques (e.g., Zeng et al. 2004; Klöpper et al. 2010; Xu and Jennings 2010. As to the best of our knowledge no such heuristic technique for solving multi user context-aware service selection problems exists, the research question of our work is as follows:

How to design a heuristic technique for the multi user context-aware service selection that determines a close-to-optimal solution in a short amount of time while scaling efficiently with problem size?

In this regard, we position our work within service systems engineering as a method to enable and enhance the contextualization and collaboration within multi user context-aware service systems and processes (Böhmann et al. 2014). The remainder of this paper is structured as follows: In the next section, we introduce a running example to illustrate our research in more detail. Then, we analyze the existing literature and, based on this analysis, discuss the research gap and our contribution. This is followed by the introduction of our model setup, which is the foundation for the presentation of our heuristic technique in the succeeding section. In the evaluation section, we analyze the performance and scalability as well as the solution quality of our approach. Finally, we conclude our paper with a short summary and an outline of limitations and further research.

2 Running Example

We introduce a running illustrative example as a basis for discussing our approach. This example is based on a scenario of five users being on a day trip in the city of Munich, Germany (see Fig. 1). All users share a process consisting of several actions (i.e., service classes) such as *visiting a museum or zoo*, *having lunch at a restaurant*, *visiting a sight, fun attraction or nature (e.g., park), city tour* and *café*. Furthermore, all users want to have lunch together at a restaurant (e.g., "Sababa", "Ni House") and want to have a city tour together in the afternoon, which can be understood as the mandatory simultaneous use of the same service object in both cases. An action resp. service class in the focus for a mandatory simultaneous use is denoted by *Focus Class (FC)* in the following. Thus, in a FC all users have to use the same service object at the same time. Moreover, in the example, all service objects are described by the attributes *price* and *duration*. All five users have their individual preferences and global end-to-end constraints regarding these NFP (e.g., restricted budget for the whole city trip). Additionally, the service objects representing real-world objects are further described by the context information *GPS position* (in terms of latitude and longitude) and *business hours* (cf. Fig. 1). While the GPS position can be used to determine the distance between service objects, business hours indicate whether a service object is available with respect to the time of day. For example, the restaurant "Ni House" is closed until 11:00 am. Consequently, having lunch at "Ni House" is only possible after 11:00 am whereas "Sababa" opens already at 10:00 am. Therefore, depending on the time of day each of the users starts the day trip (i.e., the users' initial contexts) and on the time each of them spends in the same or different museums or zoos, they will be able to have lunch together at a specific restaurant or not. This means that a temporal coordination of the users is necessary. In order to determine a (close-to-)optimal solution of this service selection problem, all users' preferences and constraints regarding price, duration and distance, all users' initial context as well as the dependencies resulting from context information (i.e., GPS position, business hours) and

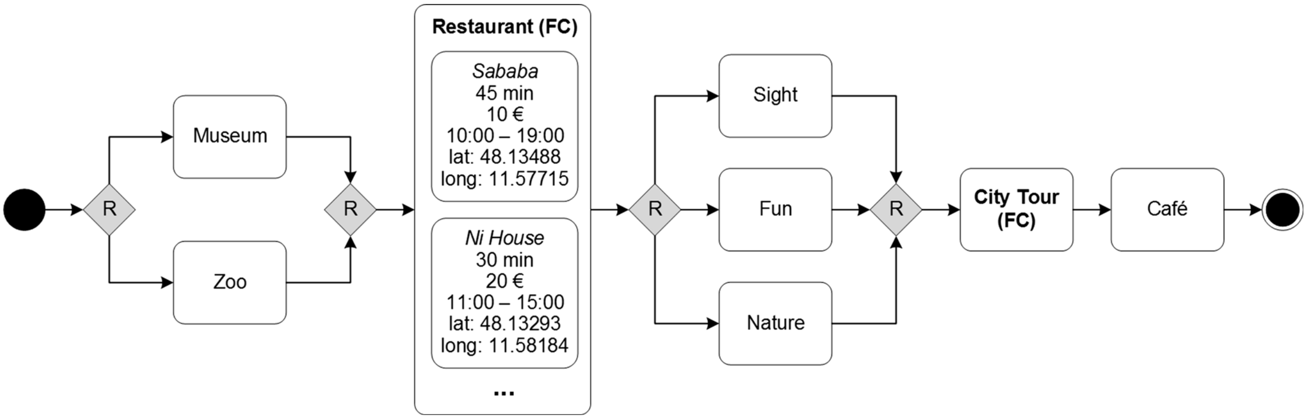

Fig. 1 Illustrative example of a city trip

multiple users (in terms of the FCs restaurant and city tour) need to be taken into account. The heuristic technique presented in this paper aims to address such decision problems in a multi user context-aware service system.

3 Literature Background and Research Gap

In the following, we will discuss existing service selection literature that deals with the consideration of dependencies resulting from both multiple users and context information. This discussion is based on a search of related work conducted in aisnet.org, Web of Science, ACM, IEEE Xplore, INFORMS and ScienceDirect. Here we used the search terms (1) *"multi* user" AND "service selection"* (character "*" as wildcard for any character(s)) as well as (2) *"context" AND "service selection"*, which resulted in 82 papers for (1) and 330 papers for (2). We then analyzed whether these works present service selection approaches relevant for our research. Moreover, in order to identify further relevant works, we also conducted a backward and forward search based on these papers. After screening titles, keywords and abstracts and removing duplicates 16 works regarding (1) and 101 works regarding (2) remained. Based on reading both introduction and summary of these papers, we extracted only the papers that take dependencies resulting from multiple users and context information into account. This resulted in 9 works for search term (1) and 7 works for search term (2). The results indicate that much service selection literature considering multiple users or context information exists, but only 16 works propose (mathematically defined) service selection approaches that deal with dependencies resulting from multiple users or context information. In the next paragraph we discuss this literature and present the identified research gap and our contribution.

3.1 Literature Background

Besides traditional single user service selection approaches (cf., e.g., Zeng et al. 2004; Ardagna and Pernici 2007; Yu et al. 2007; Alrifai et al. 2012), a variety of approaches that consider multiple users (e.g., Wang et al. 2010, 2014; Kang et al. 2011; Wanchun et al. 2011; Benouaret et al. 2012; He et al. 2012; Jin et al. 2012a, b; Liang et al. 2013; Heinrich et al. 2015a; Zhu et al. 2017) or context information (e.g., Zhou et al. 2008; Yu and Reiff-Marganiec 2009a; Xu and Jennings 2010; Yuan et al. 2013; Zhang et al. 2013a; Heinrich and Lewerenz 2015; Lewerenz 2015; Deng et al. 2016; Zhang et al. 2016) exists.

To begin with approaches that consider multiple users, Benouaret et al. (2012), Wanchun et al. (2011) and Wang et al. (2010) focus on the mandatory use of the same service (object) by several users – as we do. However, they consider only one single action (i.e., one service class) and thus not a complete process. He et al. (2012), Jin et al. (2012a) and Zhu et al. (2017) address capacity limits of services in their works, and Heinrich et al. (2015a) propose a multi user service selection approach for processes where users have preferences with respect to other users. However, all of these works do not deal with any context information (see also Table 1).

In contrast, context information and resulting dependencies are considered, for instance, by Heinrich and Lewerenz (2015) in terms of business hours and the distance between different service objects. Shen et al. (2012a) address the distance between different devices of a user in their approach. Yu and Reiff-Marganiec (2009a) consider price discounts for certain sets of services when determining suitable services for scenarios such as organizing a meeting or planning a trip. The service selection approach of Deng et al. (2016) considers the variation of the mobile network's signal strength in case a user is moving around. All of the approaches taking into account context dependencies resulting from context information but do not cope with multiple users.

Existing literature considering multiple users *or* context information can be distinguished in works proposing exact approaches and works presenting heuristic techniques for solving the service selection problem. While exact approaches can assure the optimal solution, they usually are faced with performance issues due to the high problem complexity, in particular caused by the consideration of multiple users or context information. Therefore, other researchers have proposed heuristic techniques that aim to overcome these issues while still achieving a close-to-optimal solution. For instance, Ai and Tang (2008) and Zhang et al. (2013b) provide genetic algorithm approaches, where the dependencies resulting from context information are regarded in the crossover and mutation phase of the algorithm. Moreover, Zhu et al. (2017) utilize artificial bee colony optimization to solve the multi user service selection problem taking capacity limits of services into account. Other authors such as Yu and Reiff-Marganiec (2009a) and Zhang et al. (2013a) restrict the consideration of context information to certain areas of the whole process and solve the service selection problem based on that. A similar idea is followed by Jin et al. (2012a) and Lewerenz (2015) who transform the user's global end-to-end constraints regarding the NFP into local constraints to adapt local service selection. Table 1 summarizes the literature discussion.

Table 1 Summary of literature discussion

	Consideration of		Type of approach	
	Multiple users	Context information	Exact	Heuristic
Wanchun et al. (2011)	Mandatory use of only one but the same service	N/A	X	
He et al. (2012)	Only capacity limits of services	N/A	X	
Heinrich et al. (2015a)	User preferences referring to other users	N/A	X	
Benouaret et al. (2012), Wang et al. (2010)	Mandatory use of only one but the same service	N/A		X
Jin et al. (2012a), Zhu et al. (2017)	Only capacity limits of services	N/A		X
Heinrich and Lewerenz (2015), Shen et al. (2012a), Xu and Jennings (2010)	N/A	Several pieces of context information such as price discounts and distance	X	
Ai and Tang (2008), Deng et al. (2016), Lewerenz (2015), Yu and Reiff-Marganiec (2009a), Zhang et al. (2013a, b)	N/A	Several pieces of context information such as price discounts and distance		X

3.2 Research Gap and Contribution

As discussed in the literature background, several service selection approaches that either address context information or multiple users exist. An approach that allows to consider both is – to the best of our knowledge – missing so far. Moreover, due to the high complexity of the multi user context-aware service selection problem, a heuristic technique that determines a close-to-optimal solution in a short amount of time while scaling efficiently with problem size is needed. Therefore, we aim at developing such an approach in this work.

To do so, we will utilize the concepts of decomposition and local service selection which refer to the selection of locally optimal service object(s) regarding a sub process of a multi user context-aware service system. As stated in literature (cf. Alrifai et al. 2012; Jin et al. 2012a; Suriarayanan et al. 2015), these concepts provide a promising foundation for developing heuristic techniques for service selection as the overall problem complexity can be distributed to several smaller service selection problems. Additionally, local selection is efficient in terms of computation time compared to global optimization (Sun and Zhao 2012). However, local selection without considering the users' global end-to-end constraints regarding the NFP (e.g., restricted overall budget) may result in possibly infeasible service compositions. To address this issue, we propose an optimized decomposition of global end-to-end constraints into local constraints, which can then be used in local selection. Furthermore, in order to enable the consideration of context information and multiple users in decomposition and local service selection, our heuristic technique is based on a stateful representation (cf., e.g., Heinrich and Lewerenz 2015). This representation allows to coordinate the users' actions based on dependencies resulting from context information and between service compositions of individual users.

In conclusion, our aim is to present a heuristic technique that is able to consider both context information and multiple users while providing close-to-optimal solutions and a good scalability with problem size.

4 Model Setup

To enable a better differentiation between existing knowledge and our contribution, we introduce existing definitions and concepts in this section that can serve as a foundation for our service selection approach.

4.1 NFP Model and Utility Function

In a first step, we consider a sequential process consisting of several service classes S_i (i.e., actions), with $i = 1$ to I (cf. Ardagna and Pernici 2007; Yu et al. 2007) and multiple participating users $a \in A$. Each service class encompasses a set of functionally equivalent service objects s_{ij} (with $j = 1$ to J_i). Consequently, all service objects of a certain service class differ only in their values of the considered NFP. A service composition is a concrete realization of a process (i.e., a set of service objects with exactly one service object for each service class of the process). The used formal notation throughout this work is summarized in Online Appendix 2.

To deal with context information, context-aware (CA) and non-context-aware (NCA) attributes need to be distinguished (cf. also Yu and Reiff-Marganiec 2009b; Xu and Jennings 2010; Lin et al. 2012; Heinrich and Lewerenz 2015). NCA and CA attributes together form the set of NFP N considered in a multi user context-aware service selection problem. Furthermore, we denote the quantified value of an attribute $\propto \in N$ for service object s_{ij} by $q_{ij}^{\propto}$. The quantified values (e.g., distance) of a CA attribute are dependent on context information and thus on preceding and succeeding service objects. For NCA attributes, this is not the case. We begin with the simpler case of NCA attributes. Initially, we assume that all NCA attributes are independent from each other. Therefore, depending on the quantified values $q_{ij}^{\propto}$ of the NCA attributes and the user's individual preferences, a certain service object is more or less favorable for the user compared to another functionally equivalent service object. When more than one attribute is considered in a service selection problem, a common approach to enable the assessment of different service objects is to apply a utility function that aggregates the quantified values $q_{ij}^{\propto}$ of a service object to a single utility value U (cf., e.g., Zeng et al. 2004; Ardagna and Pernici 2007; García et al. 2008; Alrifai et al. 2012; He et al. 2012; Lin et al. 2012). Here, we adapt the widely used utility function (cf., e.g., Ardagna and Pernici 2007; Alrifai et al. 2012; Jin et al. 2012a; Lin et al. 2012; Guidara et al. 2014) described in detail by Alrifai and Risse (2009), which applies the simple additive weighting (SAW) method: The quantified values of a service object are first normalized to support comparability between different dimensions of the attributes, and afterwards weighted with the user's individual preferences regarding these attributes (see Alrifai et al. 2012). In this way, the utility value $U_{a_{ij}}$ for each service object s_{ij} and each user a can be calculated. The utility value of a certain service object is usually different for each user as each user has his individual preferences.

The overall utility value of all users' service compositions can then be determined by summing up the utility values $U_{a_{ij}}$ of the selected service objects of all users $a \in A$. The binary decision variable $x_{a_{ij}}$ indicates whether a certain service object s_{ij} is selected by user a ($x_{a_{ij}} = 1$) or not ($x_{a_{ij}} = 0$):

$$\sum_{a \in A} \sum_{i=1}^{I} \sum_{s_{ij} \in S_i} U_{a_{ij}} x_{a_{ij}} \tag{1}$$

Moreover, a certain service composition for a user is only feasible if it does not violate any constraint. In our case such constraints are, on the one hand, the user's global constraints $Q_a^{\propto}$ regarding each NCA attribute $\propto$ and, on the other hand, constraints regarding two or more users such as the mandatory simultaneous use of the same service object by the users.

The kind of utility calculation presented in term (1) requires that the utility value $U_{a_{ij}}$ can be determined for each service object s_{ij} independently. This means in particular that dependencies between different service objects are not considered here. However, in the case of CA attributes, the quantified value of a service object may depend on other (preceding and succeeding) service objects, which means, it may rely on context information and context dependencies. Examples for CA attributes are distance, price discounts on a certain set of service objects, discomfort with growing travel time and business hours. Similar to NCA attributes, each user also has his own preferences and global end-to-end constraints $Q_a^{\propto}$ for these CA attributes. As the utility value of a service object is determined based on its quantified values, in case of CA attributes, context dependencies between service objects influence their utility values. To enable the determination of the quantified values of CA attributes, we utilize the existing concept of a stateful representation of context dependencies which is introduced in the following paragraph.

4.2 Decomposition Based on Stateful Representation of Context Information

As stated above, our approach for multi user context-aware service selection utilizes the concepts of decomposition and local service selection. Because local selection itself is not able to deal with global NFP constraints, the decomposition of the users' global NFP constraints into local constraints is needed in order to determine a feasible service composition. Thus, in the following, we discuss how existing approaches cope with single user decomposition and context information in order to form a foundation for our approach.

To decompose global NFP constraints with respect to NCA attributes, existing approaches (e.g., Alrifai et al. 2012; Jin et al. 2012a; Surianarayanan et al. 2015) determine a specific number of quantified NFP values for each service class i and attribute $\propto$ serving as candidates for local constraints. Then a dedicated value (which is further referred to as benefit value[2]) for each local constraint candidate is calculated based on its potential capability to enable close-to-optimal solutions during local selection.

[2] Please note that the term *benefit* does not represent the same concept as the previously introduced term *utility*, which refers to an assessment of a single service object or an entire service composition.

The local NFP constraints are then determined by maximizing the overall benefit value while satisfying the global NFP constraints.

As described above, the determination of the quantified NFP values of CA attributes for a certain service object does not only depend on the service object itself but also requires to take preceding or succeeding service objects of the service composition into account. Thus, considering context information results in context dependencies within one user's service composition. To cope with CA attributes we use the concept of world states (cf. Ghallab et al. 2004; Heinrich and Schön 2015) which maps the context dependencies onto a state space (i.e., a stateful representation of the dependencies). The basic idea is to model the feasible values of context information for each CA attribute (e.g., time of day, GPS position, etc.), based on the user's initial context (e.g., time and GPS position at start) and all service objects (e.g., duration and GPS position of corresponding real-world object) in the preceding service classes of the process. Each combination of such feasible values of context information is described by a world state ws_{ik} with i referring to the corresponding service class S_i and k as index for the world states in that service class. A world state therefore consists of exactly one state variable for each CA attribute, containing its corresponding value. In this way, the utility of a service object can be determined using the utility function proposed above and with respect to a considered world state (based on its quantified values for all NCA and CA attributes). We denote combinations of world state and service object as *world-state-service-object combinations* (WSC; for an illustration of the state representation, cf. Online Appendix 3).

Adopting this stateful representation allows to decompose a context-aware service selection problem: For each service class S_i a number Υ of suitable local constraint candidates $lcc^{\varphi}_{i\propto}$ (with $\varphi = 1$ to Υ) regarding a CA or NCA attribute $\propto$ can be determined based on its quantified values. Subsequently, the benefit value $B\left(lcc^{\varphi}_{i\propto}\right)$ for each local constraint candidate can be calculated considering the created state space (for a detailed discussion cf. Alrifai et al. 2012; Lewerenz 2015). By solving the corresponding optimization model, the local NFP constraints $LQ^{\propto}_{a_i}$ for each service class S_i regarding the attribute $\propto$ are determined. These determined local constraint candidates then form the user's vector of local NFP constraints $LQ_{a_i} = \left[LQ^1_a, \ldots, LQ^N_a\right]^T$ for each service class S_i that can be used in a local selection approach.

Moreover, the consideration of context information makes a temporal coordination of users necessary. This occurs, for instance, when taking a time-dependent availability of service objects (e.g., business hours) into account. Such a temporal coordination is also required when considering the mandatory simultaneous use of one or more service objects by multiple users. While a temporal coordination can be achieved by means of a state variable representing *time* in each world state, there is also the need to consider possible waiting times: It may be beneficial for a user to wait a certain amount of time, for example, to be able to use a service object which currently is not available but will be a few moments later instead of directly choosing a less favorable service object. In order to consider potential waiting times as well as the loss of utility caused by waiting, Heinrich et al. (2015a) introduce waiting time as additional NFP and special waiting service classes right in front of each regular service class. They further model attributes representing *time* (i.e., duration, waiting time) as discrete values (i.e., in discrete steps, such that every waiting service class encompasses a defined number of waiting services, each being described by a different specific amount of waiting time). In this way, the utility value of a waiting service can be determined similar to a regular service object.

To illustrate the existing concepts presented above, Fig. 2 contains an excerpt of the running example described in Sect. 2 (focusing on only two users, the actions "Museum", "Restaurant (FC)" and "Sight", and the NFP *duration*, *price* and *business hours*). First, waiting actions (actions 0, 2 and 4) are added right before each of the three actions (1) visiting a museum, (3) having lunch at a restaurant and (5) visiting a sight to enable a temporal coordination of the users' actions. Second, the state space for this service selection problem is created by using the existing concept of a stateful representation of context dependencies. Figure 2 illustrates the state spaces for both users with respect to the Action 3 "Restaurant (FC)": All world states encompass the context information required for the consideration of the CA attribute business hours (depending on the time of day) where the values are based on the duration of the previous service objects and the initial start time of each user (as part of the user's initial context). By combining each service object and world state all feasible WSCs of a user for an action are determined.[3]

5 Heuristic for Multi User Context-Aware Service Selection

In this section, we present our heuristic technique to support a multi user context-aware service system. This

[3] Please note that in Fig. 2, WSCs which are infeasible due to business hours (e.g., the combination of ws_{30} and s_{31} referring to restaurant "Ni House" for User 1 at 10:45) have already been removed from the state space.

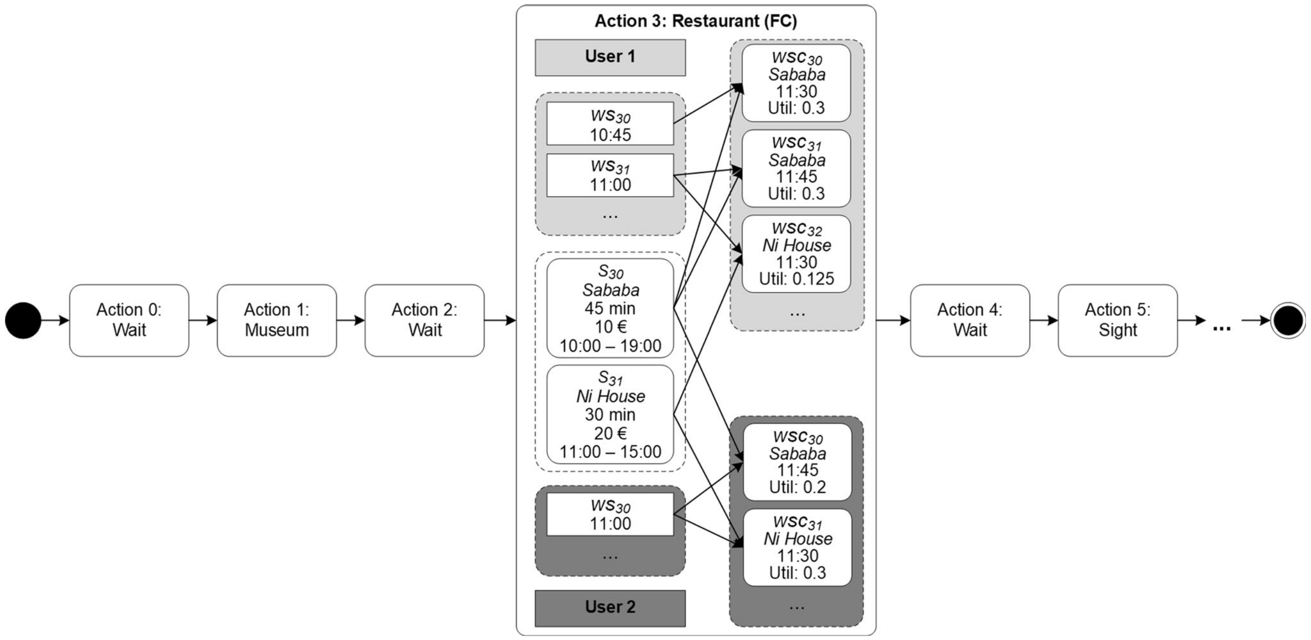

Fig. 2 Excerpt of city trip example modeled as stateful representation

technique consists of the two stages *Multi User-oriented Decomposition* and *Local Service Selection.*

5.1 Multi User-Oriented Decomposition

In the last section, we described existing concepts, which do not consider any dependencies among different users' service compositions like resulting from the mandatory simultaneous use of the same service object by multiple users in the Focus Classes (FCs) which also needs to be addressed. Regarding multiple users, we have to consider that each user may have its own set of local NFP constraints, because usually each user has different global constraints and context information (e.g., location or time of day). To address this issue, the decomposition is initially conducted for each user a separately, leading to local NFP constraints LQ_{a_i} (with $i = 1$ to I representing the different service classes) for each individual user. Second, we need to ensure that the determined local NFP constraints of each user do not violate the mandatory simultaneous use of the same service object in a FC. This happens if the local NFP constraint for an attribute is determined such that at least one user would be excluded from using a service object in the FC in regard to his own context information and local NFP constraints. To avoid this, the local NFP constraints regarding the FC determined for all users must enable the selection of at least one WSC referring to the same service object and context information. To consider these dependencies between different users' service compositions we define additional restrictions in our decomposition optimization model. For this purpose, we introduce the general concept of *common WSCs* (ComWSCs) which encompass WSCs from different users referring to the same service object and context information. For each FC the sets of ComWSCs are built based upon the following three criteria:

(a) All WSCs of a set of ComWSCs refer to the *same service object* of the FC.

(b) All WSCs of a set of ComWSCs refer to world states with the *same values for each state variable* (i.e., each world state encompasses the same context information).

(c) From *each user* participating in the mandatory simultaneous use of the FC, exactly one WSC must exist in every set of ComWSCs.

In our example of Fig. 2, wsc_{31} (User 1) and wsc_{30} (User 2) form a set of ComWSCs because they (a) refer to the same service object s_{30} "Sababa", (b) the context information *time of day* equals to 11:00 am and (c) the set encompasses one WSC from each user. Precisely, there are two different sets of ComWSCs regarding the FC "Restaurant" due to the two different manifestations of time (10:45, 11:00) and the two restaurants considered in the example: $ComWSC_0 = \{(wsc_{31}, wsc_{30})\}$[4] and $ComWSC_1 = \{(wsc_{32}, wsc_{31})\}$.

Therefore, our decomposition optimization model that allows to determine the local NFP constraints for each user

[4] The first WSC in the parentheses is from User 1, the second WSC from User 2.

from the set of local constraint candidates $lcc^{\varphi}_{i\propto}$ is formulated as follows:

$$\max_{x^{\varphi}_{i\propto}} \sum_{i=1}^{I}\sum_{\varphi=1}^{\Upsilon} B(lcc^{\varphi}_{i\propto}) * x^{\varphi}_{i\propto} \tag{2}$$

$$\text{s.t.} \sum_{i=1}^{I}\sum_{\varphi=1}^{\Upsilon} lcc^{\varphi}_{i\propto} * x^{\varphi}_{i\propto} \leq Q^{\propto}_{a} \tag{3}$$

$$\sum_{\varphi=1}^{\Upsilon} x^{\varphi}_{i\propto} * Z(lcc^{\varphi}_{i\propto}) = 1 \quad \forall i \in I_{FC} \tag{4}$$

$$\sum_{\varphi=1}^{\Upsilon} x^{\varphi}_{i\propto} = 1 \quad \forall i = 1,\ldots,I \quad \text{with } x^{\varphi}_{i\propto} \in \{0,1\} \tag{5}$$

With each local constraint candidate $lcc^{\varphi}_{i\propto}$ having its individual benefit value $B(lcc^{\varphi}_{i\propto})$, the aim of the objective function (2) is to maximize the accumulated benefit value over all selected local constraint candidates for a user. Here, $x^{\varphi}_{i\propto}$ are decision variables that indicate whether the corresponding candidate $lcc^{\varphi}_{i\propto}$ is selected as local NFP constraint ($x^{\varphi}_{i\propto} = 1$) or not ($x^{\varphi}_{i\propto} = 0$). Restriction (3) ensures that the sum of the values of all selected candidates $lcc^{\varphi}_{i\propto}$ for the attribute $\propto$ is less than or equal to the user's global NFP constraint $Q^{\propto}_{a}$ and thus the global constraints are satisfied.[5]

Restrictions (4) integrate the concept of ComWSC which, as described above, is necessary to guarantee the mandatory simultaneous use of a service object by multiple users: We introduce I_{FC} as the set of the indices of all FCs of the process. Furthermore, we also introduce $Z(lcc^{\varphi}_{i\propto})$ which equals to 1 if there is at least one set of available ComWSCs when the local constraint candidate $lcc^{\varphi}_{i\propto}$ is chosen as local NFP constraint for FC $i \in I_{FC}$, and equals to 0 otherwise. Multiplied with the binary decision variable $x^{\varphi}_{i\propto}$, these restrictions ensure that only local constraint candidates that allow the selection of at least one ComWSC in the local selection for FC $i \in I_{FC}$ can be chosen. Referring to our illustrative example, the local constraint candidate $lcc^{1}_{3Price} = 10€$ has one possible ComWSC ($ComWSC_0$). Thus, this local constraint candidate would allow the selection of at least one ComWSC and therefore $Z\left(lcc^{1}_{3Price}\right) = 1$. The last restrictions (5) are used to assure that exactly one local constraint candidate $lcc^{\varphi}_{i\propto}$ is selected for each service class S_i.

By solving the proposed decomposition optimization model for every user and every global NFP constraint we receive local constraints LQ_{a_i} for all service classes of the process. These local constraints LQ_{a_i} are used in our local service selection approach to fulfill the global NFP constraints of each user during selection.

5.2 Local Service Selection

Based on the local NFP constraints from our Multi User-oriented Decomposition, the approach for local service selection can be proposed. The purpose of local service selection is to find both a feasible and close-to-optimal service composition under the consideration of the mandatory simultaneous use of the same service object in the FCs. To achieve this goal, we present our local service selection approach for multiple users and a single FC.

A solution of local service selection encompasses a WSC for each service class and every user. Due to the mandatory simultaneous use of the same service object in the FC, a solution is feasible if – and only if – a WSC which is part of the same set of ComWSCs for the FC is selected for all users. Selecting services for each service class and user independently similar to existing approaches (cf. Jin et al. 2012a; Lewerenz 2015; Surianarayanan et al. 2015) is not promising here. This is due to the fact that the different initial contexts, preferences and local NFP constraints of every user would most likely lead to different optimal world states (as part of the optimal WSC) for the FC and therefore the users would not be temporally and spatially coordinated. Thus, local service selection has to assure that the same context information in conjunction with the same service object for every user is selected with respect to the FC. To achieve this, we introduce our local service selection algorithm consisting of three selection steps: (1) *ComWSC selection*, (2) *backward selection* and (3) *forward selection*. Each step addresses one different part of the process (cf. Fig. 3) and is presented in the following.

We propose to start with (1) *ComWSC selection* at the FC as the definition of ComWSCs (cf. Multi User-oriented Decomposition) guarantees the mandatory simultaneous use of the same service object in the FC. This step allows for a successful service selection since the state space of each user ensures that service objects for the remaining preceding resp. succeeding actions can be selected. This is due to the fact that the state space is created based on the initial context of each individual user and therefore only WSCs in the ComWSCs and the succeeding actions exist that can be accessed from the initial context. To determine which ComWSC should be selected, the local selection of our heuristic technique is based on two fundamental criteria:

(a) Feasibility: Only ComWSCs that fulfill the local NFP constraints of every user can be selected

[5] Please note that this refers only to attributes that have to be minimized (e.g., price), for attributes that have to be maximized the less-than-or-equal sign has to be replaced by the greater-than-or-equal sign.

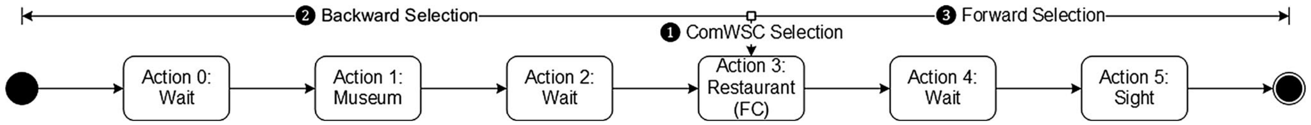

Fig. 3 Local service selection for the city trip example

(b) Optimality: Selection of the ComWSC that leads to the highest aggregated utility among all users

Regarding our city trip example, the aggregated utility is 0.5 for $ComWSC_0$ and 0.425 for $ComWSC_1$ (cf. Fig. 2). Since both ComWSCs are feasible, the algorithm selects $ComWSC_0$, and thus wsc_{31} for User 1 and wsc_{30} for User 2. To sum up (1), *ComWSC selection* considers the dependencies between the users at the FC and the local service selection approach can further proceed with a (2) *backward selection* and (3) *forward selection* addressing the remaining actions of a process.

(2) *Backward selection* starts for each user at the WSC selected for the FC. Going backwards and step by step, it determines a WSC for every action until the beginning of the process is reached. To do so, again local selection under the consideration of the feasibility and optimality criteria takes place. Additionally, a path between the WSC to be selected and the previously selected WSC (i.e., a path between the corresponding world states in the user's state space) needs to exist. In our example, this means that for Action 2 the WSC providing the highest utility which has a path to WSC wsc_{31} (and therefore to world state ws_{31}) is selected subject to User 1's local NFP constraints for this service class. This selection is repeated for all service classes until the beginning of the process is reached and conducted for each user.

Afterwards, (3) *forward selection* is used to select the remaining WSCs from the FC to the end of the process. The WSCs are selected in the same manner as in (2) *backward selection*, with the difference that the succeeding WSCs must contain a world state connected to the previously selected WSC. In case of our city trip example, this means that with respect to User 1's local NFP constraints the WSC with the highest utility in Action 4 and a path to wsc_{31} is selected. Similar to (2) *backward selection*, this selection is repeated until the end of the process is reached and is conducted for each user.

To sum up, by applying the three selection steps we aim to select a feasible, close-to-optimal WSC for each action of the process and user, and thus a service composition for each user in which all users use the same service object in the FC simultaneously.

Above, we discuss the local service selection for multiple users and a *single* FC. From a methodical perspective, it is necessary to present two extensions of this approach (cf. Online Appendix 4). First, we have to consider *multiple* FCs, which is explicated in Online Appendix 4.1. Second, as a core aspect of the heuristic technique, we have to deal with potential local selection failures (i.e., in case local selection cannot find a feasible WSC for a certain action in the first run, for instance, due to context dependencies). To overcome this issue, we include the concept of backtracking in our local service selection approach by identifying the world state responsible for the selection failure and to assure that this world state cannot be selected again. To do so, the world state is put onto a so-called blacklist and an additional feasibility criterion is added to the selection steps, which inhibits the selection of WSCs associated with blacklisted world states. A discussion of the concept of backtracking is shown in Online Appendix 4.2.

6 Evaluation

In this section, we evaluate our heuristic technique regarding solution quality (i.e., providing close-to-optimal solutions) as well as performance and scalability. For this purpose, we compare our method with two exact service selection approaches as well as two heuristic techniques by means of a simulation experiment using real-world data in the tourism domain. Thus, our evaluation design follows the compositional style of *simulation- and metric-based benchmarking* (cf. Prat et al. 2015). We implemented our heuristic technique in Java and use the mathematical programming solver GUROBI[6] for solving the optimization models of the decomposition stage. To ensure a correct implementation of our algorithm, we conducted intensive testing of the source code, namely manual code reviews by persons other than the programmers, unit tests, runs with extreme values and plausibility checks. Moreover, we feel confident that our approach provides correct feasible solutions since for 10,000 randomly generated service selection problems (with a maximum problem size of 429,981,696 possible combinations of service objects) each determined heuristic solution was a feasible solution out of the set of all feasible solutions provided by an exhaustive enumeration.

First, we evaluate our heuristic technique (abbreviated with *HA*) by comparing its results to the results of two exact approaches that were extended to allow the

[6] http://www.gurobi.com. Accessed 25 June 2018.

determination of the optimal solution of a multi user context-aware service selection problem. The first exact approach, which we denote by *StatefulGO* (GO = Global Optimization), is based on an approach proposed by Heinrich and Lewerenz (2015) and uses a stateful representation of context information similar to our approach. It then applies integer programming to solve the corresponding optimization model for *single user* context-aware service selection problems. The second one, *StatelessGO*, is derived from Heinrich et al. (2015a) who propose an approach for multi user service selection without considering any context information. In this approach, the existing dependencies among different users are modeled directly within a knapsack optimization model. In order to apply these two exact approaches in our evaluation, we extended them to cope with multi user context-aware processes. With regard to these extensions, we implemented both extensions as fair as possible, taking into account all possible optimizations to reduce their computation time.

Furthermore, we evaluate our technique *HA* by comparing its results to the results of other potential heuristics for multi user context-aware service selection. Because of the lack of such heuristics in existing literature (cf. Sect. 3), we examine "competing" approaches in two ways: First, we use the criterion TimeLimit[7] in GUROBI, which can be applied to both exact approaches *StatefulGO* and *StatelessGO* by setting their maximum computation time equal to the computation time of *HA*. However, these approaches were not able to find a feasible solution in the given time, thus this criterion is not applicable to evaluate the proposed heuristic technique (a more detailed discussion can be found in Online Appendix 5). Second, we use the criterion SolutionLimit[8] in GUROBI in combination with both exact approaches. Here, we set the criterion SolutionLimit to 1, which means that GUROBI is forced to return the first feasible solution found. Regarding the second criterion SolutionLimit, we were able to define two new heuristics *StatefulSL* and *StatelessSL* and to analyze and compare the results to the results of *HA*.

Our evaluation is based on a real-world scenario of the tourism domain. More precisely, several users conduct a day city trip sharing the same process (cf. running example). Here, we consider the NCA attributes *duration*, *waiting time*, *price* and *favorite score* (an attribute used for modeling user favorites) as well as the CA attributes *distance* and *business hours*. Furthermore, we use TripAdvisor[9] to determine suitable real-world service objects with their NFP values. To take the necessary transport between two succeeding actions into account, we integrate an additional service class representing "transport" before each regular action of the process. In detail, the different services of such a transport service class represent different transport options (walk, bike, car) for the users, each with its corresponding NFP values for duration, price and favorite score. In this way, the process in our evaluation consists of alternating transport and focus service classes with a waiting service class right before each transport or focus service class. Furthermore, each user has his individual initial context and end point. The *initial* problem size encompasses three users, two FCs with 40 functionally equivalent service objects each, 5 waiting services per waiting service class and 144 transport services per transport service class. For each simulation run, we randomly generate the users' preferences and constraints as well as their initial context (i.e., time of day, GPS position; the basic evaluation configuration is summarized in Online Appendix 6).

To evaluate our approach, we derive three extended configurations from the basic evaluation configuration where in each configuration only one parameter is changed while all other parameters remain as defined in the basic evaluation configuration (i.e., ceteris paribus). To do so, we stepwise increase the value of each analyzed parameter until the computation time of each of the approaches *StatelessGO*, *StatefulGO*, *StatelessSL* and *StatefulSL* reaches a limit of 250 s on average (i.e., the upper limit for a parameter such as *no. of FCs* can be different for each of these approaches). Table 2 illustrates the resulting parameter intervals for the three extended evaluation configurations. Although we cannot compare *HA* with the four other approaches above their particular upper limit per parameter (cf. Table 2), we continue the stepwise increase of each parameter regarding *HA* to further analyze its scalability (e.g., we increase *no. of users* stepwise up to the value 20).

Our evaluation is two-fold: We analyze solution quality by means of a utility comparison and performance and scalability by means of a comparison of the computation time of *HA* with *StatelessGO*, *StatefulGO, StatelessSL* and *StatefulSL.* Thereby, each of the settings of the three evaluation configurations is simulated 50 times. We then determine the average utility (*U*) and computation time (*CT*) over all simulation runs for each of the five approaches. Computation time is measured in seconds [sec] and, for each run, encompasses state space creation, decomposition and local service selection for *HA*, state space creation and building/solving the optimization model for *StatefulGO* and *StatefulSL*, and building/solving the

[7] http://www.gurobi.com/documentation/6.5/refman/timelimit.html#parameter:TimeLimit. Accessed 25 June 2018.

[8] http://www.gurobi.com/documentation/6.5/refman/solutionlimit.html#parameter:SolutionLimit. Accessed 25 June 2018.

[9] http://www.programmableweb.com/api/tripadvisor. Accessed 25 June 2018.

Table 2 Evaluation setup parameters per approach

Extended evaluation configurations	*HA*	*StatefulSL*	*StatefulGO*	*StatelessSL*	*StatelessGO*
(1) No. of service objects per FC	10–120 (in steps of 10) and 140–200 (in steps of 20)	10–120 (in steps of 10) and 140–200 (in steps of 20)	10–70 (in steps of 10)	10–120 (in steps of 10)	10–120 (in steps of 10)
(2) No. of FCs	1–10 (in steps of 1)	1–3 (in steps of 1)	1–2 (in steps of 1)	1–5 (in steps of 1)	1–5 (in steps of 1)
(3) No. of users	2–10 (in steps of 1) and 12–20 (in steps of 2)	2–7 (in steps of 1)	2–4 (in steps of 1)	2–10 (in steps of 1)	2–10 (in steps of 1)

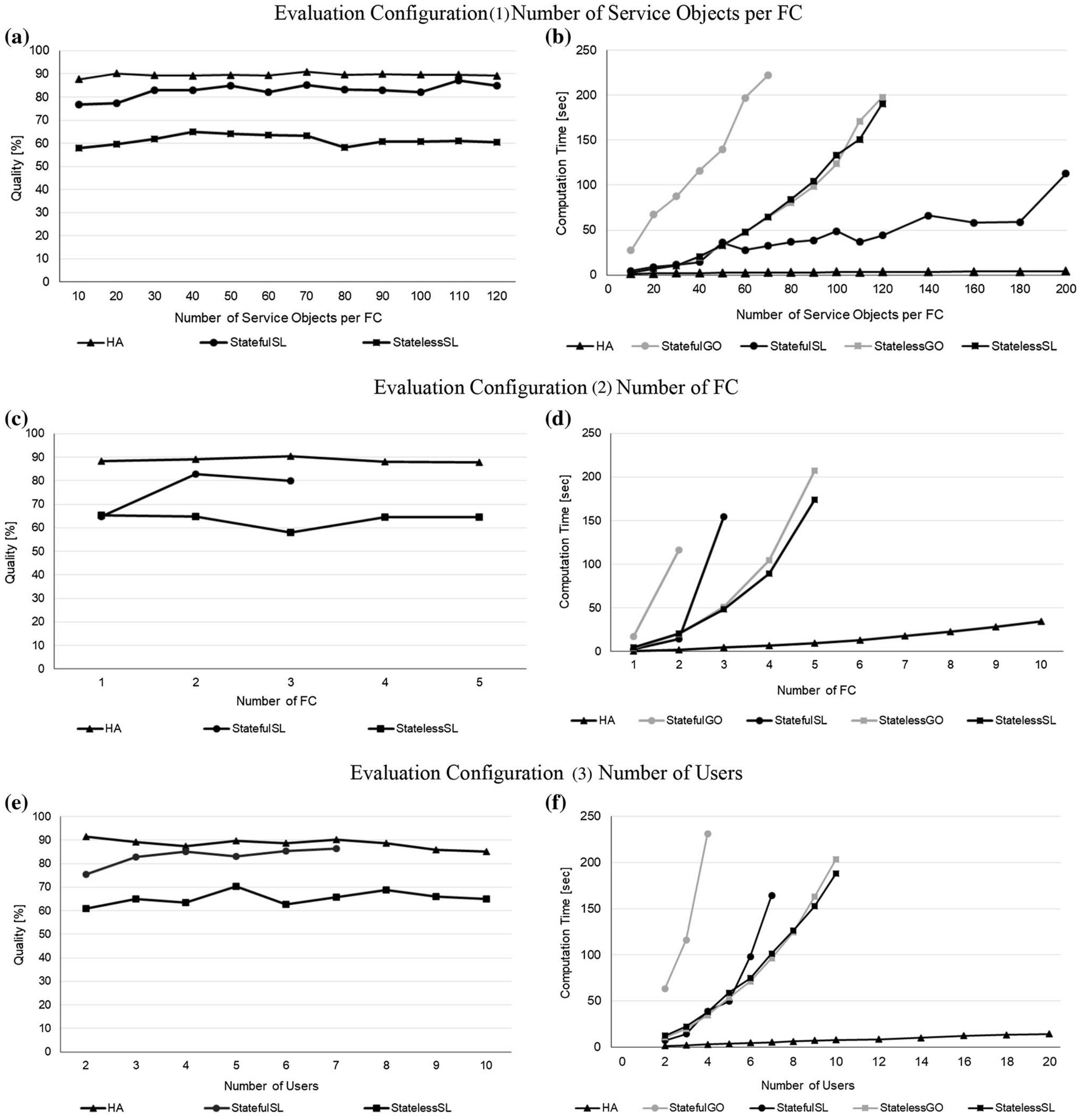

Fig. 4 Evaluation results

optimization model for *StatelessGO* and *StatelessSL*. All simulation runs are conducted on an Intel Xeon E5-2470 v2 processor with 2.40 GHz, 32 GB RAM, Win7 64bit, Java 1.8, and GUROBI Optimizer 6.5. In order to analyze and compare solution quality as well as performance and scalability, the indicators *Quality* and *Computation Time Percentage* (CTP) are used, which are defined as follows:

$$Quality = \frac{U_A - U_{EXACT}^{min}}{U_{EXACT}^{max} - U_{EXACT}^{min}} \quad CTP = \frac{CT_{HA}}{CT_A} \tag{6}$$

The indicator *Quality* is determined by setting the utility U_A achieved by *HA, StatefulSL* and *StatelessSL* in relation to the maximum utility U_{EXACT}^{max} and the minimum utility U_{EXACT}^{min} determined by an exact approach. Furthermore, to assess performance and scalability we use the indicator *CTP* which is determined by dividing the computation time CT_{HA} required by *HA* by the computation time CT_A required by any another approach. In the following, we discuss the evaluation results in terms of solution quality, performance and scalability.

Solution quality: As illustrated in the diagrams (a), (c) and (e) of Fig. 4, the proposed heuristic *HA* reaches an average value of 89.02% of the *Quality* indicator over all evaluation configurations. This solution quality can be compared to the two other heuristic approaches, which reach an average value of only 81.53% (*StatefulSL*) and 63.15% (*StatelessSL*) of the *Quality* indicator. Moreover, the results show that *HA* outperforms *StatefulSL* and *StatelessSL* not only on average, but also in each single evaluation configuration (1)–(3). Especially for the highly relevant evaluation configuration (2) focusing on a stepwise increase of the number of FCs, *HA* provides significantly better results compared to both other heuristics. Further, the solution quality offered by our approach is quite constant for all evaluation configurations [cf. Table 3 for the minimum, maximum and average values of the *Quality* indicator in each evaluation configuration (1)–(3)]. Here, the range between the minimum and maximum value of the *Quality* indicator is between 3.2 and 6.3% for *HA*, 10.3 and 18.1% for *StatefulSL* and 6.9 and 9.5% for *StatelessSL* (see Table 3). Thus, our heuristic technique seems to be able to provide a high and robust solution quality regarding different problem sizes.

In addition, comparable to both heuristics *StatefulSL* and *StatelessSL*, *HA* was nearly always (precisely, in 99.6% of the cases) able to determine a feasible solution (i.e., a feasible service composition). All considered heuristic techniques (*HA*, *StatefulSL* and *StatelessSL*) provide the optimal solution (i.e., a solution with utility equal to the maximum utility U_{EXACT}^{max}) only in about 1% of all simulation runs. In case of *HA*, the obvious reason for this is the local perspective of decomposition and local selection. Consequently, only taking a global perspective (as done by exact approaches) would allow to overcome this issue.

Performance and Scalability: To begin with, the exact approaches *StatelessGO* and *StatefulGO* show the expected increase in computation time [cf. diagrams (b), (d) and (f) in Fig. 4] since the service selection problem is known to be NP-hard (Alrifai et al. 2012). For our technique *HA*, we expect the increase in computation time to be much lower with growing problem size. Indeed, this is supported by our results for all three evaluation configurations: The *CTP* of *HA* compared to both *StatelessGO* and *StatefulGO* is steadily decreasing (cf. Online Appendix 7) when increasing the number of considered service objects per FC, the number of FCs, and the number of users. For instance, regarding evaluation configuration (1) the *CTP* is 54.0% (*StatelessGO*) resp. 7.7% (*StatefulGO*) for 10 service objects compared to 4.3% (*StatelessGO*) and 1.8% (*StatefulGO*) for 70 service objects. This also holds with respect to the heuristic techniques *StatelessSL* and *StatefulSL* as *HA* outperforms both techniques in each simulation run. In addition, although *HA, StatefulSL* and *StatefulGO* have in common the computation time required for state creation, *HA* clearly outperforms *StatefulSL* and *StatefulGO*. This indicates that our *Multi User-oriented Decomposition* in combination with *Local Service Selection* is performing much better than building and solving an optimization model based on the same state space. Furthermore, based on the evaluation results even for larger problem sizes, *HA* seems to provide very good scalability (cf. Fig. 4).

Table 3 Values of the quality indicators per evaluation configuration (1)–(3)

Evaluation configuration	(1)			(2)			(3)		
	Min. (%)	Avg. (%)	Max. (%)	Min. (%)	Avg. (%)	Max. (%)	Min. (%)	Avg. (%)	Max. (%)
HA	87.7	89.5	90.9	87.8	88.8	90.5	85.1	88.5	91.4
StatefulSL	76.8	82.7	87.1	64.8	75.9	82.9	75.4	83.0	86.4
StatelessSL	58.0	61.4	64.9	58.1	63.5	65.4	60.9	65.3	70.4

7 Conclusion, Limitations and Future Research

In this work, we presented a heuristic service selection technique to support multi user context-aware service systems. More precisely, our approach is able to determine a close-to-optimal service composition for multi user processes under the consideration of the users' individual preferences and constraints, context information (especially in mobile environments) and the mandatory simultaneous conduction of one or many actions in the processes.

To the best of our knowledge, existing approaches either neglect the support of context information or multiple users. To address this research gap, we developed a technique consisting of two stages: first, *Multi User-oriented Decomposition* decomposes the users' global NFP constraints into local NFP constraints to address the mandatory simultaneous use of a service object by several users. In the second stage *Local Service Selection* (consisting of three steps) a feasible and close-to-optimal service composition for each user is determined. We further extended this procedure by backtracking to take care of possible local selection failures, which, for instance, can be caused by context dependencies.

Besides its scientific contribution, our heuristic approach also provides some important benefits for practice: Due to the NP-hardness of the service selection problem and the high complexity caused by considering multiple users and context information, existing exact approaches may be unable to determine a solution for larger real-world problems in appropriate computation time. Based on findings from our evaluation, we feel confident that our heuristic technique provides good performance and scalability as well as a high solution quality. Thus real-world multi user context-aware processes can be supported by the provided approach which can be used to enable and enhance the contextualization and collaboration within multi user context-aware service systems and processes (cf. Böhmann et al. 2014). This may be of particular relevance to service systems in dynamic domains such as tourism, healthcare or disaster relief assistance, because the respective participants may require quick and helpful support for decision making. In this regard, the rising availability of mobile devices accompanying the omnipresence of context information and services may awaken the request for coordinated, spontaneous and immediate decision support, for instance, in the tourism domain (e.g., day city trips, visiting a restaurant or meetups). Similar developments may hold for service systems of other domains, for instance, in healthcare, roadside assistance or disaster relief management. In these application contexts, the fast coordination of, for example, diverse disaster relief forces (police, emergency doctors or firefighters) is highly valuable as it is more beneficial for certain actions to be conducted together by forces with complementary professional skills. Here, our proposed approach could help to address this demand.

However, our heuristic technique is also subject to some limitations regarding the feasibility and utility of the determined solutions. These limitations are the starting points for further research. The evaluation has shown that our approach is not always capable of finding an existing feasible solution. One reason for this is that the decomposition of global NFP constraints into local NFP constraints cannot fully take into account context dependencies. This can lead to local NFP constraints of a user that are too restrictive or to local NFP constraints resulting in blacklisted WSCs in all available sets of ComWSCs, making the selection of a common service object not possible. To overcome this limitation, further research may focus on enhancing the decomposition stage by learning from the information of previous local service selections. For instance, the information about a local selection failure at a specific action could be used to repeat the stage *Multi User-oriented Decomposition* with different parameters leading to different local NFP constraints, thus allowing for a new and possibly feasible local service selection. Furthermore, our evaluation has shown that while the proposed heuristic technique is able to achieve a high solution quality, the optimal solution was determined only in about 1% of the evaluated settings. The reason for this is the local perspective of the selection procedure as for each action the service object with the highest utility is selected. Thus, the heuristic technique is vulnerable to finding only local optima. A possible solution to overcome this limitation and to even further increase the solution quality is to integrate probability distributions in local selection and to conduct the stage *Local Service Selection* multiple times moderated by these distributions. Obviously, this requires a trade-off between computational effort and solution quality.

Moreover, considering multi user context-aware service systems and processes, we also have to discuss that the proposed approach does not cope with uncertain context information (cf., e.g., Heinrich and Schön, 2015) that can occur during the execution of a predetermined service composition. Such uncertain information can result from different causes such as data quality problems (e.g., incorrect or outdated NFP values of a service object), altered preferences or restrictions of one or more users, user failures or even variable external influences (e.g., turning weather or unexpectedly long traffic jams during travel). Some works in the research field of service selection deal with uncertain information by proposing proactive strategies that build on more robust models or rule-based supervision already before the execution of the service composition (cf., e.g., Yu and Lin 2005; Ardagna et al.

2011; Shen et al. 2012b; Heinrich et al. 2015b). Moreover, further research could also consider stochastic approaches, which could overcome local optima and also take into account alternative service compositions to deal with uncertain information. Additionally, exceptional events that occur disruptively during the execution of a service composition still have to be considered. Since starting to reselect the whole service composition in case of a disruptive event usually leads to a waste of resources (time, budget, etc.), many researchers in the field of service systems (e.g., Kuster 2008; Mu et al. 2011) and in particular in service selection (e.g., Ardagna and Pernici 2007; Sandionigi et al. 2013; Liang and Du 2017) suggest to only reselect the remaining part of the service composition in order to promptly enable continuing the execution of the composition. This could also be a promising start for a multi user context-aware service re-selection approach able to consider uncertainties at execution time.

In conclusion, our heuristic technique constitutes a first step to provide support for service systems utilizing multi user context-aware service selection.

References

Ai L, Tang M (2008) QoS-based web service composition accommodating inter-service dependencies using minimal-conflict hill-climbing repair genetic algorithm. In: IEEE fourth international conference on eScience, pp 119–126

Alrifai M, Risse T (2009) Combining global optimization with local selection for efficient QoS-aware service composition. In: Quemada J, León G, Maarek Y, Nejdl W (eds) Proceedings of the 18th international conference on world wide web, pp 881–890

Alrifai M, Risse T, Nejdl W (2012) A hybrid approach for efficient web service composition with end-to-end QoS constraints. ACM Trans Web 6:1–31

Alter S (2012) Metamodel for service analysis and design based on an operational view of service and service systems. Serv Sci 4:218–235

Amin MB, Banos O, Khan WA, Muhammad Bilal HS, Gong J, Bui D-M, Cho SH, Hussain S, Ali T, Akhtar U, Chung TC, Lee S (2016) On curating multimodal sensory data for health and wellness platforms. Sensors 16(7):980

Ardagna D, Pernici B (2007) Adaptive service composition in flexible processes. IEEE Trans Softw Eng 33:369–384

Ardagna D, Baresi L, Comai S, Comuzzi M, Pernici B (2011) A service-based framework for flexible business processes. IEEE Softw 28:61–67

Benouaret K, Benslimane D, Hadjali A (2012) Selecting skyline web services for multiple users preferences. In: IEEE 19th international conference on web services (ICWS), pp 635–636

Böhmann T, Leimeister JM, Möslein K (2014) Service systems engineering. Bus Inf Syst Eng 6:73–79

Deng S, Huang L, Hu D, Zhao JL, Wu Z (2016) Mobility-enabled service selection for composite services. IEEE Trans Serv Comput 9:394–407

Dey AK (2001) Understanding and using context. Pers Ubiquitous Comput 5:4–7

Edvardsson B, Ng G, Zhi Min C, Firth R, Yi D (2011) Does service-dominant design result in a better service system? J Serv Manag 22:540–556

Frost R, Lyons K (2017) Service systems analysis methods and components: a systematic literature review. Serv Sci 9:219–234

García JM, Ruiz D, Ruiz-Cortés A, Parejo JA (2008) QoS-aware semantic service selection: an optimization problem. In: Proceedings of the 2008 IEEE congress on services—part I, pp 384–388

Ghallab M, Nau DS, Traverso P (2004) Automated planning: theory and practice. Elsevier/Morgan Kaufmann, Amsterdam

Grönroos C (2011) Value co-creation in service logic: a critical analysis. Mark Theor 11:279–301

Guidara I, Guermouche N, Chaari T, Tazi S, Jmaiel M (2014) Pruning based service selection approach under QoS and temporal constraints. In: de Roure D (ed) IEEE international conference on web services (ICWS). IEEE, Piscataway, pp 9–16

He Q, Han J, Yang Y, Grundy J, Jin H (2012) QoS-driven service selection for multi-tenant SaaS. In: IEEE 5th international conference on cloud computing, pp 566–573

Heinrich B, Lewerenz L (2015) Decision support for the usage of mobile information services: a context-aware service selection approach that considers the effects of context interdependencies. J Decis Syst 24:406–432

Heinrich B, Schön D (2015) Automated planning of context-aware process models. In: European conference on information systems. https://doi.org/10.18151/7217352

Heinrich B, Klier M, Lewerenz L, Mayer M (2015a) Enhancing decision support in multi user service selection. In: 36th international conference on information systems, Fort Worth

Heinrich B, Klier M, Lewerenz L, Zimmermann S (2015b) Quality-of-service-aware service selection: a novel approach considering potential service failures and nondeterministic service values. Serv Sci 7:48–69

Jin H, Zou H, Yang F, Lin R, Shuai T (2012a) A novel method for optimizing multi-user service selection. JCIT 7:296–310

Jin H, Zou H, Yang F, Lin R, Shuai T (2012b) Using bipartite graph for resolving multiple requests conflicts. In: International joint conference on service sciences, pp 46–50

Kabir G, Akhtar Hasin MA (2011) Evaluation of customer oriented success factors in mobile commerce using fuzzy AHP. J Ind Eng Manag 4:361–386

Kang G, Liu J, Tang M, Liu X, Fletcher KK (2011) Web service selection for resolving conflicting service requests. In: IEEE international conference on web services, pp 387–394

Klöpper B, Ishikawa F, Honiden S (2010) Service composition with Pareto-optimality of time-dependent QoS attributes. In: Maglio PP, Weske M, Yang J, Fantinato M (eds) Service-oriented computing, vol 6470. Springer, Heidelberg, pp 635–640

Kuster J (2008) Providing decision support in the operative management of process disruptions. GITO, Berlin

Lewerenz L (2015) A heuristic technique for an efficient decision support in context-aware service selection. In: 36th international conference on information systems, Fort Worth

Liang H, Du Y (2017) Dynamic service selection with QoS constraints and inter-service correlations using cooperative coevolution. Future Gener Comput Syst 76:119–135

Liang Z, Zou H, Guo J, Yang F, Lin R (2013) Selecting web service for multi-user based on multi-QoS prediction. In: IEEE international conference on services computing, pp 551–558

Lin D, Shi C, Ishida T (2012) Dynamic service selection based on context-aware QoS. In: Proceedings of the IEEE ninth international conference on services computing, pp 641–648

Lyons K, Tracy S (2013) Characterizing organizations as service systems. Hum Factors Ergon Manuf Serv Ind 23:19–27

Maglio PP, Srinivasan S, Kreulen JT, Spohrer J (2006) Service systems, service scientists, SSME, and innovation. Commun ACM 49:81

Moghaddam M, Davis JG (2014) Service Selection in web service composition: a comparative review of existing approaches. In: Bouguettaya A, Sheng QZ, Daniel F (eds) Web services foundations. Springer, New York, pp 321–346

Mu Q, Fu Z, Lysgaard J, Eglese R (2011) Disruption management of the vehicle routing problem with vehicle breakdown. J Oper Res Soc 62:742–749

Prat N, Comyn-Wattiau I, Akoka J (2015) A taxonomy of evaluation methods for information systems artifacts. J Manag Inf Syst 32:229–267

Sandionigi C, Ardagna D, Cugola G, Ghezzi C (2013) Optimizing service selection and allocation in situational computing applications. IEEE Trans Serv Comput 6:414–428

Shen Y, Wang M, Tang X, Luo Y, Guo M (2012a) Context-aware HCI service selection. Mob Inf Syst 8:231–254

Shen Y, Yang X, Wang Y, Ye Z (2012b) Optimizing QoS-aware services composition for concurrent processes in dynamic resource-constrained environments. In: IEEE 19th international conference on web services, pp 250–258

Statista (2017a) Mobile professional services market size worldwide 2014–2018. https://www.statista.com/statistics/501755/worldwide-mobile-professional-services-market-revenue/. Accessed 20 Mar 2017

Statista (2017b) Number of available apps in the Apple App Store from July 2008 to January 2017. https://www.statista.com/statistics/263795/number-of-available-apps-in-the-apple-app-store/. Accessed 16 Feb 2017

Sun SX, Zhao J (2012) A decomposition-based approach for service composition with global QoS guarantees. Inf Sci 199:138–153

Surianarayanan C, Ganapathy G, Ramasamy MS (2015) An approach for selecting best available services through a new method of decomposing QoS constraints. Serv Oriented Comput Appl 9:107–138

Vargo SL, Lusch RF (2004) Evolving to a new dominant logic for marketing. J Market 68:1–17

Wanchun D, Chao L, Xuyun Z, Chen J (2011) A QoS-aware service evaluation method for co-selecting a shared service. In: IEEE international conference on web services, pp 145–152

Wang H, Zhang J, Wan C, Shao S, Cohen R, Xu J, Li P (2010) Web service selection for multiple agents with incomplete preferences. In: IEEE/ACM international conference on web intelligence-intelligent agent technology, pp 565–572

Wang S, Hsu C-H, Liang Z, Sun Q, Yang F (2014) Multi-user web service selection based on multi-QoS prediction. Inf Syst Front 16:143–152

Xu L, Jennings B (2010) A cost-minimizing service composition selection algorithm supporting time-sensitive discounts. In: IEEE international conference on services computing, pp 402–408

Yu T, Lin K-J (2005) Adaptive algorithms for finding replacement services in autonomic distributed business processes. In: Proceedings of autonomous decentralized systems, pp 427–434

Yu HQ, Reiff-Marganiec S (2009a) A backwards composition context based service selection approach for service composition. In: IEEE international conference on services computing, pp 419–426

Yu HQ, Reiff-Marganiec S (2009b) Automated context-aware service selection for collaborative systems. In: van Eck P, Gordijn J, Wieringa R (eds) Advanced information systems engineering, vol 5565. Springer, Heidelberg, pp 261–274

Yu T, Zhang Y, Lin K-J (2007) Efficient algorithms for web services selection with end-to-end QoS constraints. ACM Trans Web 1(1):6

Yuan Y, Zhang X, Sun W, Cao Z, Wang H (2013) Optimal web service composition based on context-awareness and genetic algorithm. In: international conference on information science and cloud computing companion, pp 660–667

Zaplata S, Kunze CP, Lamersdorf W (2009) Context-based cooperation in mobile business environments. Bus Inf Syst Eng 1:301–314

Zeng L, Benatallah B, Ngu AHH, Dumas M, Kalagnanam J, Chang H (2004) QoS-aware middleware for web services composition. IEEE Trans Softw Eng 30:311–327

Zhang D, Adipat B, Mowafi Y (2009) User-centered context-aware mobile applications—the next generation of personal mobile computing. Commun Assoc, Inf Syst, p 24

Zhang M, Ranjan R, Haller A, Georgakopoulos D, Strazdins P (2012) Investigating decision support techniques for automating cloud service selection. In: IEEE 4th international conference on cloud computing technology and science. Piscataway, pp 759–764

Zhang M, Liu C, Yu J, Zhu Z, Zhang B (2013a) A correlation context-aware approach for composite service selection. Concurr Comput 25:1909–1927

Zhang Z, Zheng S, Li W, Tan Y, Wu Z, Tan W (2013b) Leveraging genetic algorithm to compose web services in a context-aware environment. In: IEEE international conference on systems, man and cybernetics, pp 829–834

Zhang C, Zhang L, Zhang G (2016) QoS-aware mobile service selection algorithm. Mob Inf Syst 2016:1–6

Zhou T, Zheng X, Song WW, Du X, Chen D (2008) Policy-based web service selection in context sensitive environment. In: IEEE congress on services part 1, pp 255–260

Zhu W, Yin B, Gong S, Cai K-Y (2017) an approach to web services selection for multiple users. IEEE Access 5:15093–15104