Information Systems Frontiers
https://doi.org/10.1007/s10796-023-10392-8

Service Re-Selection for Disruptive Events in Mobile Environments: A Heuristic Technique for Decision Support at Runtime

Michael Bortlik[1] · **Bernd Heinrich**[1] · **Daniel Lohninger**[1]

Accepted: 28 March 2023

Abstract

Modern service-based processes in mobile environments are highly complex due to the necessary spatial–temporal coordination between multiple participating users and the consideration of context information. Due to the dynamic nature of mobile environments, disruptive events occur at runtime, which require a re-selection of the planned service compositions respecting multiple users and context-awareness. Thereby, when re-selecting services the features performance, solution quality, solution robustness and alternative solutions are essential and contribute to the efficacy of service systems. This paper presents an optimization-based heuristic technique based on a stateful representation that uses a region-based approach to re-select services considering multiple users, context information and in particular disruptive events at runtime. The evaluation results, which are based on a real-world scenario from the tourism domain, show that the proposed heuristic is superior compared to competing artifacts.

Keywords Service re-selection · Mobile environments · Runtime decision support · Disruptive events · Multi user processes · Context-awareness

1 Introduction

The development of mobile environments in the form of mobile technologies (such as Smartphones, IoT-Devices) and mobile business is steadily increasing (Muhammad et al., 2018; Statista, 2019a, b). Services that use context information (e.g., sensory capabilities of mobile devices) to provide individual solutions for users gain in importance, which can be seen, for instance, by the increasing market value of 11.99 billion $ in 2015 to 44.95 billion $ in 2021 for location based services (Allied Market Research, 2021). Besides location information, a further important dimension of context is the participating user (user context), which includes the interaction among users (Baccari & Neji, 2016; Grotherr et al., 2018; Ma et al., 2015; Roda et al., 2018; Shen & Yang, 2011; Tung et al., 2014; Weinert et al., 2020; Zhang et al., 2009). Processes in mobile environments, which include multiple participating users can be found in several domains such as healthcare or disaster relief assistance, field work in companies, everyday efficiency and planning, roadside or in tourism (cf. Gavalas et al., 2014; Neville et al., 2016; Ventola, 2014; Zhang et al., 2009). This development supports the construction of *multi user context-aware* service systems, in which mobile technologies enable the realization and support of individual processes.

Determining and realizing processes by an individual service composition for each user at *planning time*, including the selection within a high number of available candidate services, context information (e.g., location or time of day) and the coordination of multiple users, is a known problem in the literature (e.g., Bortlik et al., 2018; Heinrich & Mayer, 2018). However, due to the dynamic nature of mobile environments (cf., e.g., Nagarajan & Thirunavukarasu, 2020; Sheng et al., 2014), several so-called *disruptive events* may occur *at runtime* (Bobek & Nalepa, 2017). Guided by Bearzotti et al. (2012), we define a disruptive event as a real-world event at runtime that significantly change planned values of service candidates, user constraints etc. Thus, disruptive events occurring at runtime can affect all users resulting in service compositions selected at planning time that are no longer optimal if not even no longer feasible. For instance, the disruptive event that a business – selected as service at planning time – is closed and thus become no

✉ Bernd Heinrich
Bernd.Heinrich@ur.de

[1] Information Systems, University of Regensburg, Universitätsstraße 31, 93053 Regensburg, Germany

Published online: 09 May 2023

longer available (cf. Sheng et al., 2014; Zhang et al., 2018; Zheng et al., 2014), results in a need for re-selection at runtime. As the computational effort for determining an optimal solution for such re-selection tasks – considering all possible disruptive events and their changes of planned values – would be very high (the selection problem is known to be NP-hard (non-deterministic polynomial-time hardness); cf. Abu-Khzam et al., 2015; Alrifai et al., 2012), we propose an optimization-based heuristic technique. Thus, our work refers to the following research question:

> *How to design an optimization-based heuristic technique for re-selecting services under consideration of multiple users, context information and in particular disruptive events at runtime?*

To address this research question, we present a novel heuristic technique for service re-selection at runtime for multi user context-aware service systems. Our approach is based on the meta-heuristic local selection (e.g., Gendreau & Potvin, 2010) and a stateful representation (cf., e.g., Heinrich & Lewerenz, 2015). To enable a performant service re-selection at runtime considering multiple users, context-awareness and in particular disruptive events, the presented approach carefully divides the underlying process into regions and efficiently select a service composition by using feasibility checks and a measure based on the stateful representation. Furthermore, the concept of a stateful representation is substantially extended by a dynamic re-structuring of states at runtime to deal with disruptive events. The evaluation results show that for the presented heuristic technique the computation time increases proportionally in almost all settings, while competing approaches in general have an exponential increase in runtime. More precisely, our technique requires on average only about 1.6% of the computation time compared to existing approaches although achieving a solution quality similar to these approaches (88.8% compared to an optimal solution). This reveals a high performance while retaining a comparable high solution quality. Moreover, our technique achieves an average (solution) robustness of 92%, which describes the proportion of the predetermined services at planning time that can be retained during re-selection at runtime. This represents a significant improvement over existing approaches. Finally, in contrast to existing approaches, our technique is able to present several alternative feasible solutions in near-time especially for multiple participating users. The evaluation results show that an alternative solution can be selected within 6.1 seconds on average for multiple users in case a disruptive event occurs.

The remainder of the paper is organized as follows: In the next section, the background (i.e., Methodical Foundations and Related Work) as well as the research gap are discussed. In the third section, we present our research methodology followed by the introduction of a real-world scenario from the tourism domain in section four. In the fifth section, we propose our model setup and the main components of our heuristic technique, in particular an optimization model and an algorithm for multi user context-aware service re-selection. In the evaluation section, we analyze different features of our approach compared to competing artifacts based on the real-world scenario and a simulation experiment. Subsequently, the results and implications are discussed. Finally, we conclude our paper with a short summary and an outline of limitations and further research.

2 Background and Research Gap

2.1 Methodical Foundations

The selection of several services, thus determining a service composition, leads to a problem which in its basic form (Quality of Service-aware service selection) is related to the selection of exactly one item for each of n available sets of items without violating a family of constraints, while maximizing the overall utility (Caserta & Voß, 2019). Thus, the underlying decision problem can be characterized as the well-known MMKP (Multi-Choice Multi-Dimensional Knapsack Problem). The MMKP is an advanced form of the general Knapsack Problem with an additional *multi-choice* property, i.e., the selection of items is performed on sets of items instead of a single item set and further, the resources within a MMKP are *multi-dimensional* since there are multiple resource constraints for the knapsack (Ardagna & Pernici, 2006). In our problem context, the multi-choice property is given by the selection of items (= candidate services) grouped in multiple sets (= tasks), for which each item is characterized by a specific value (= utility). Furthermore, the multi-dimensionality is determined by multiple resources (e.g., duration of a candidate service), which constrain the knapsack (= user's service composition). When considering multiple users, the selected candidate services of a user are dependent on the selected candidate services of the other users in terms of optimality (i.e., utility) and feasibility (i.e., constraints). Thus, the considered MMKP is of higher complexity since user-based dependencies have to be additionally modelled when determining each user's service composition.

There are different meta-heuristics in the literature (Gendreau & Potvin, 2010) that can be used as a basis for solving the MMKP in the context of multi user context-aware service re-selection. Especially the concepts of the meta-heuristic *Tabu search* are promising for solving this problem, because (1) Tabu search is basically a *local search* strategy (Gendreau & Potvin, 2005), which searches the whole neighborhood *deterministically* allowing for a higher degree of solution robustness. Furthermore (2), the existing *memory concept* is highly promising (Gendreau & Potvin, 2010) in order to provide access to relevant (existing) information during the iterations of the re-selection and thus enables to efficiently and effectively explore the search space by learning from previous solutions (Blum & Roli, 2003; Boussaïd et al., 2013; Gendreau & Potvin, 2010; Gogna & Tayal, 2013). In particular (2a), the concept of *short-term memory* (Tabu list) facilitates to keep track of the most recently visited solutions and forbids (or allows) specific moves towards them (Blum & Roli, 2003). Thus, the neighborhood of the current solution is restricted to the solutions that do not belong to the Tabu list (Blum & Roli, 2003). In addition (2b), the concept of *long-term memory* contains a pool of previously generated solutions, which can be used to learn and restart the search (Gendreau & Potvin, 2005). This allows in analogy to buffer states of already examined solutions to enable fast decision support at runtime even for problems which are NP-hard. The considered MMKP faces a high degree of complexity which is discussed in literature and caused by dependencies referring to multiple users, context-awareness and disruptive events at runtime (e.g., Heinrich & Mayer, 2018):

1. **User-based dependencies** Including *multiple users* cause the MMKP to be solved considering each users' service composition (i.e., characterized by multiple tasks of candidate services) in compliance with the user's preferences and constraints. Such preferences and constraints of participating users comprise also temporal-based dependencies arise in case (a subset of) users want to simultaneously select particular candidate services each assigned to a task leading to dependencies between various service compositions. Furthermore, by considering temporal-based dependencies certain candidate services which can (potentially) be added to a service composition are available or not (e.g., depending on temporal availability of a candidate service). Following, changing the candidate services within a service composition of one user affect other service compositions of (subsets of) related users. This requires the inclusion of temporal-based dependencies between multiple users in the optimization model, which in turn leads to additional user-dependent decision variables for each candidate service and thus increases the complexity of the MMKP.
2. **Context-based dependencies** Context-based dependencies result from the consideration of *context information* (e.g., location of a candidate service or a user) in a MMKP. Context is usually addressed by the concept of states, which represent the contextual dependencies between candidate services of consecutive tasks. More precisely, the selection of a candidate service within a task can lead to context information, which directly affects the selection of candidate services (for other users) in subsequent tasks (cf. Heinrich & Lewerenz, 2015) and thus increase the complexity of the MMKP (Mostofa Akbar et al., 2006; Sbihi, 2007). Furthermore, the quantified values for subsequent candidate services (e.g., the distance between the locations of two candidate services), used to calculate the utility, depends on the specific context information.
3. **Event-based disruption** As *disruptive events* at runtime can cause a re-selection of one or more service compositions from one or more users, the complexity of the corresponding MMKP increases. In particular, the re-selection and therefore the service compositions are dependent on the number and the type of the occurring disruptive events as well as the affected candidate services, which are unknown at planning time. For example, the disruptive event that a business is actually closed at runtime, which was selected as service at planning time and thus become no longer available, can affect a wide range in the planned service compositions such as exceeding or falling short of a threshold, time conflicts, capacity conflicts, changing context information, changing preferences, candidate service failures or delays. Due to the dynamic nature of disruptive events at runtime and the variety of event types, typically different affected candidate services must be re-selected.

Given these dependencies referring to multiple users, context-awareness and disruptive events, in the literature of service science and re-selection are multiple works, which discuss central features and goals to assess the re-selection solution (Ardagna & Pernici, 2007; Cao et al., 2007; Caserta & Voß, 2019; Di Napoli et al., 2021; Khan et al., 2002; Mostofa Akbar et al., 2006; Wang et al., 2019; Yu & Lin, 2007). The following describes the features to be addressed in more detail:

1. **Performance** In case of disruptive events, support of the participating users in near-time is needed as waiting times in mobile applications have a negative impact on the users' satisfaction. In particular, a fast and interactive support is a key indicator for usability and thus a key feature for success of mobile applications. Thus, users in mobile environments require near-time support when interacting with mobile applications (Galletta et al., 2004; Hoxmeier & DiCesare, 2000; Li & Chen, 2019). Therefore, finding a solution at runtime with high performance (near-time optimization) is vital for the efficacy of service systems in mobile environments (Harrison et al., 2013; Saleh et al., 2017; Seffah et al., 2006; Tan et al., 2013).

2. **Solution quality** Users expect a high solution quality when re-selecting services. On the one hand, finding an optimal service composition in appropriate time is not realistic at runtime due to the high complexity of the underlying decision problem (Moghaddam & Davis, 2014; Zhang et al., 2012b). On the other hand, finding only a feasible service composition is necessary but not sufficient. Therefore, a near-optimal solution should be aimed for.
3. **Solution robustness** Considering re-selection at runtime, fundamental changes to the predetermined service composition at planning time have a negative impact on the users' satisfaction (Barber & Salido, 2015; Rahmani & Ramezanian, 2016; Seffah et al., 2006; Tan et al., 2013). Therefore, users expect a largely robust service composition at runtime, where in case of disruptive events a small number of changes occur to the predetermined service composition (Barber & Salido, 2015; Rahmani & Ramezanian, 2016).
4. **Alternative solutions** The provided service composition at runtime may not meet the users expectations due to, for example, imprecise or incomplete context information or changing user preferences (David et al., 2014). Thus, determining and presenting alternative feasible solutions in near-time is important. Based on such alternative solutions, proactive user interaction at runtime is possible (Evers et al., 2014) to give the user the opportunity to change the service composition proposed in the, for instance, mobile environment.

As discussed in literature, these features contribute to the efficacy of multi user context-aware service systems since the response to disruptive events and the corresponding adaptation of service compositions represent a key role in the design of these systems (Alter, 2017; Faieq et al., 2021; Hidri et al., 2019). However, existing approaches for multi user context-aware service systems primarily focus on the (adaptation) requirements of the service provider. As a result, the perspective of the users (i.e., service consumers, cf. Faieq et al., 2021) and their required features (e.g., high performance, cf. Frey et al., 2017) are mostly not included in the design of service systems. Therefore, addressing these features is highly important for research and practice, especially for the further development of multi user context-aware service systems (Frey et al., 2017).

2.2 Related Work

In the following, we discuss existing approaches from literature, which can be used in general for service re-selection. Therefore, we applied a literature search process consisting of three phases (cf. Fig. 1). The discussion is based on a literature search of related work (Phase 1) conducted in aisnet.org, Web of Science, ACM, IEEE Xplore, INFORMS, ScienceDirect and Springer, which was performed using 36 keywords resulting in 156 papers (Task I). Moreover, in order to identify further relevant works, we also conducted a backward and forward search based on these papers resulting in 35 further articles and 191 articles in sum (Task II). After a more detailed analysis of the relevant articles (Phase 2) by screening title, keywords and abstracts, 69 papers remained (Task III). Based on reading introduction and summary, we only considered works that contain dependencies between multiple users and context information resulting in 53 articles (Task IV). A detailed text analysis resulted in 38 relevant approaches that are meet our research topic in general (Task V). Thereupon, we structure the literature (Phase 3) firstly on the basis whether the approaches consider multiple users, context-awareness and in particular disruptive events and secondly based on the four central features discussed in the methodical foundations (cf. Section 2.1). A summary of the systematized related work can be found in Table 1 (Task VI), which is explicitly created based on the dependencies and features introduced in Section 2.1.

(A): The first group of service system approaches mainly deal with *user-based dependencies* (e.g., Mayer, 2017; Wanchun et al., 2011). With regard to multiple users, there are several approaches that allow the same candidate services to be used by multiple users (Heinrich et al., 2015; Mayer, 2017; Wanchun et al., 2011; Wang et al., 2010, 2014), while other approaches address capacity limits of services in order to provide multi user support (He et al., 2012; Jin et al., 2012; Kurdija et al., 2019; Pang et al., 2020; Zhu et al., 2017). However, only Mayer (2017) presents an approach for service re-selection in order to cope with disruptive events occurring at runtime but do not deal with context-based dependencies at all. An analysis of the features shows, that some of the works aim at *performance* and therefore He et al. (2012), Heinrich et al. (2015), Jin et al. (2012), Kurdija et al. (2019), Mayer (2017), Wang et al. (2010) and Zhu et al. (2017) can in general provide results in near-time. In regard to *solution quality*, all of the approaches address as a foremost goal the selection of feasible solutions (e.g., Pang et al., 2020). Additionally, these works can be divided in exact approaches (i.e., providing an optimal solution, e.g., Wang et al., 2014) and heuristic techniques (i.e., providing a near-optimal solution, e.g., Kurdija et al., 2019). For the re-selection at runtime the use of heuristics is clearly favored, because they can enable near-optimal solutions while reducing the computational effort. Nevertheless, Mayer (2017) uses an exact approach at runtime to obtain the optimal solution. Finally, none of the existing works contrib-

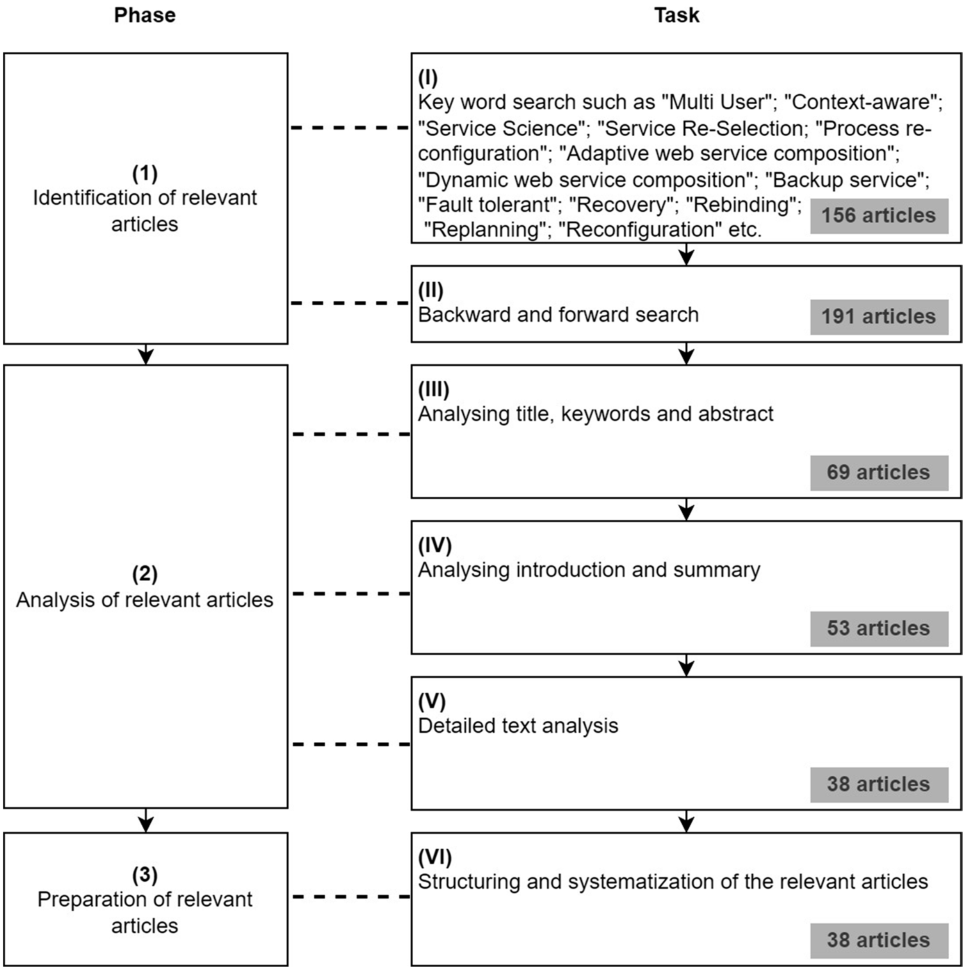
Phase
Task
(1)
Identification of relevant articles
(2)
Analysis of relevant articles
(3)
Preparation of relevant articles
(I)
Key word search such as "Multi User"; "Context-aware"; "Service Science"; "Service Re-Selection; "Process re-configuration"; "Adaptive web service composition"; "Dynamic web service composition"; "Backup service"; "Fault tolerant"; "Recovery"; "Rebinding"; "Replanning"; "Reconfiguration" etc.
156 articles
(II)
Backward and forward search
191 articles
(III)
Analysing title, keywords and abstract
69 articles
(IV)
Analysing introduction and summary
53 articles
(V)
Detailed text analysis
38 articles
(VI)
Structuring and systematization of the relevant articles
38 articles

Fig. 1 Literature search process

utes to the features *solution robustness* as well as *alternative solutions*.

(B): Furthermore, there is a group of several service system approaches that deal with *context-awareness* (e.g., Faieq et al., 2019; Lewerenz, 2015; Shen et al., 2012) and thus address context dependencies between various candidate services based on available context information such as price or distance. Existing literature that deals with context-awareness can be further divided into selection and re-selection works (including *disruptive events* at runtime). In the service systems literature with regard to context-awareness various types of re-selection approaches have been proposed such as fault tolerant strategies (Angarita et al., 2013, 2014; Fekih et al., 2019b; Zheng & Lyu, 2009), process reconfiguration (Shen & Yang, 2011; Zhang et al., 2012a) and adaptive web service compositions (Aouatef et al., 2008; Ardagna & Pernici, 2007; Buys et al., 2011; Cao et al., 2015; Cherif et al., 2019; Ma et al., 2015; Sandionigi et al., 2013; Sedighiani et al., 2021; Tretola & Zimeo, 2019). In a large part of these approaches, *performance* is an important feature to assess the solution (e.g., Bortlik et al., 2018; Xu & Jennings, 2010). In contrast, near-time optimization in order to enable support at runtime is not addressed by any of the presented works. The analysis of the *solution quality* shows that all approaches enable a feasible solution. Furthermore, only a few articles can provide an optimal solution (i.e., Heinrich & Lewerenz, 2015; Sandionigi et al., 2013; Shen et al., 2012; Xu & Jennings, 2010; Zheng & Lyu, 2009), while most of the approaches describe a heuristic technique in order to achieve a near-optimal solution (e.g., Fekih et al., 2019a). Besides that, Angarita et al. (2013), Angarita et al. (2014), Ardagna and Pernici (2007), Sandionigi et al. (2013), Shen and Yang (2011), Zhang et al. (2012a) and Zheng and Lyu (2009) focus on the feature *solution robustness*. However, just Zhang et al. (2012a) describes explicitly the reduction of changing services as goal. All

Table 1 Overview of existing approaches

	Service system approach	Approaches considering			Features of the solution						
		User-based depen-dencies	Context- based depen-dencies	Event-based disruption	1. Solution quality			2. Solution robustness	3. Performance		4. Alternative solutions
					Optimal solution	Near-optimal solution	Feasible solution		Performance	Near-time optimization	
A	Wanchun et al. (2011), Wang et al. (2014)	X	-/-	-/-	X	-/-	X	-/-	-/-	-/-	-/-
	He et al. (2012), Heinrich et al. (2015)	X	-/-	-/-	X	-/-	X	-/-	X	X	-/-
	Jin et al. (2012), Kurdija et al. (2019), Wang et al. (2010), Zhu et al. (2017)	X	-/-	-/-	-/-	X	X	-/-	X	X	-/-
	Pang et al. (2020)	X	-/-	-/-	-/-	X	X	-/-	-/-	-/-	-/-
	Mayer (2017)	X	-/-	X	X	-/-	X	-/-	X	X	-/-
B	Heinrich and Lewerenz (2015), Shen et al. (2012), Xu and Jennings (2010),	-/-	X	-/-	X	-/-	X	-/-	X	-/-	-/-
	Ai and Tang (2008), Deng et al. (2016), Faieq et al. (2019), Fekih et al. (2019a), Lewerenz (2015), Yu and Reiff-Marganiec (2009), Zhang et al. (2013a)	-/-	X	-/-	-/-	X	X	-/-	X	-/-	-/-
	Zhang et al. (2013b)	-/-	X	-/-	-/-	X	X	-/-	-/-	-/-	-/-
	Sandionigi et al. (2013), Zheng and Lyu (2009)	-/-	X	X	X	-/-	X	X	X	-/-	-/-
	Cao et al. (2015)	-/-	X	X	-/-	X	X	-/-	-/-	-/-	-/-
	Fekih et al. (2019b)	-/-	X	X	-/-	X	X	-/-	X	-/-	-/-
	Ardagna and Pernici (2007), Shen and Yang (2011)	-/-	X	X	-/-	X	X	X	X	-/-	-/-
	Aouatef et al. (2008), Buys et al. (2011), Ma et al. (2015), Sedighiani et al. (2021), Tretola and Zimeo (2019)	-/-	X	X	-/-	-/-	X	-/-	-/-	-/-	-/-
	Zhang et al. (2012a)	-/-	X	X	-/-	-/-	X	X	-/-	-/-	-/-
	Cherif et al. (2019)	-/-	X	X	-/-	-/-	X	-/-	X	-/-	-/-
	Angarita et al. (2013), Angarita et al. (2014)	-/-	X	X	-/-	-/-	X	X	X	-/-	-/-
C	Heinrich and Mayer (2018)	X	X	-/-	X	-/-	X	-/-	X	-/-	-/-
	Bortlik et al. (2018)	X	X	-/-	-/-	X	X	-/-	X	-/-	-/-

other approaches only implicitly deal with the feature solution robustness without explicitly presenting techniques to improve the robustness of solutions. Finally, none of the considered approaches contributes to the feature *alternative solutions*.

(C): Finally, there is only a limited amount of articles that deal with *user-based as well as context-based dependencies* (Bortlik et al., 2018; Heinrich & Mayer, 2018). Both approaches allow the same candidate services to be used by multiple users while respecting context dependencies between these candidate services. Nevertheless, none of these approaches discusses event-based disruptions. Considering the feature *performance,* it becomes apparent that both approaches cannot enable near-time optimization. Furthermore, in regard to *solution quality*, Heinrich and Mayer (2018) presents a technique to obtain an optimal solution, whereas Bortlik et al. (2018) focus on a heuristic technique. Finally, neither approach contributes to the features *solution robustness* and *alternative solutions*. A detailed comparison of our approach with the approach by Bortlik et al. (2018) showing significant differences can be found in the appendix of this work.

In summary, none of the identified works provides an approach that can select solutions with fast support, high quality and robustness while respecting multiple users, context-awareness and in particular disruptive events.

2.3 Research Gap

As discussed in the related work, there are several service system approaches that deal with multiple users, context-awareness or disruptive events. However, a multi user context-aware re-selection approach that copes with all these concepts, especially addressing the feature performance and near-time optimization at runtime, is – to the best of our knowledge – missing so far. At runtime, there are high user demands on the solution of a re-selection, in particular solutions in near-time (i.e., performance), solution quality, solution robustness and alternative solutions. Existing service system approaches from the literature indeed refer to the complexity of user-based and context-based dependencies but cannot address high-performance and robust solutions at runtime while maintaining a high solution quality. However, as discussed in the methodical foundations, these features are important for designing service systems. In particular, features contribute to the efficacy of multi user context-aware service systems since the response to disruptive events and the corresponding adaptation of service compositions represent a fundamental part in the design of these service systems (Alter, 2017; Faieq et al., 2021; Hidri et al., 2019). Addressing these features is highly important in research as well as in practice, especially for the further development of multi user context-aware service systems (Frey et al., 2017). To deal with these features, we propose an optimization-based heuristic technique that uses a sophisticated region-based approach to re-select service compositions considering multiple users, context-awareness and in particular disruptive events at runtime. In this regard, we present the first re-selection approach that apply a heuristic on a stateful representation. Thus, the represented context information within these states (i.e., state space) enable our heuristic for near-time optimization as well as feasibility checks and a state space measure. Furthermore, the concept of a state space is significantly extended by a dynamic state space restructuring at runtime to deal with disruptive events.

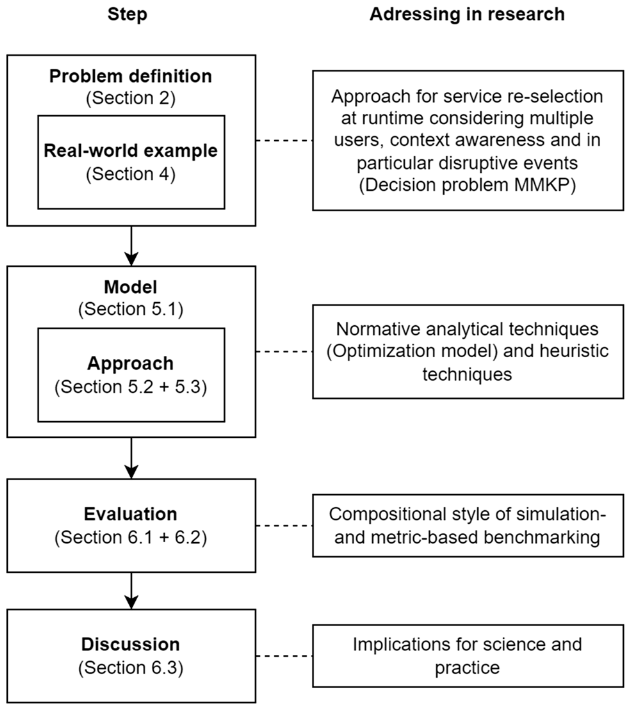
Step
Adressing in research
Problem definition
(Section 2)
Real-world example
(Section 4)
Approach for service re-selection at runtime considering multiple users, context awareness and in particular disruptive events (Decision problem MMKP)
Model
(Section 5.1)
Approach
(Section 5.2 + 5.3)
Normative analytical techniques (Optimization model) and heuristic techniques
Evaluation
(Section 6.1 + 6.2)
Compositional style of simulation- and metric-based benchmarking
Discussion
(Section 6.3)
Implications for science and practice

Fig. 2 Quantitative research approach

3 Research Methodology

Our work is found in general on the quantitative research paradigm (cf. Meredith et al., 1989; Will et al., 2002). In the following, we describe each step of this approach with regard to the work at hand in detail (cf. Fig. 2):

The first step *problem definition* includes the description and discussion of the subject and topic of the study with respect to existing knowledge bases and foundations (cf. Will et al., 2002). Therefore, we ground our research on *multi user context-aware service re-/selection* in the literature under the well-known decision problem MMKP

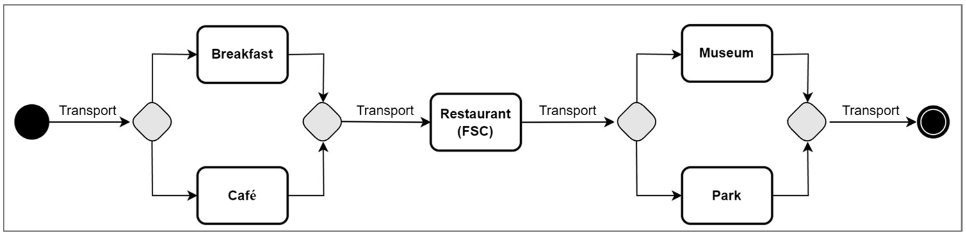

Fig. 3 Excerpt of a process model for a day trip (guided by Hara et al., 2016; Oizumi et al., 2013)

including user-based as well as context-based dependencies and in particular disruptive events at runtime (cf. Section 2.1). Moreover, the scope and the differences of our work have to be discussed compared to existing works that study a related problem (cf. Section 2.2), resulting in a research gap (cf. Section 2.3). To address the illustration of the problem definition (cf. Will et al., 2002), we use a mobile-enabled process from the tourism sector, which represents a valuable part in the further development of service design (cf. Section 4). Based upon this problem definition, we introduce the basic *model notation* as foundation in Section 5.1. In this regard, an optimization model is established, which enables a solution for the underlying decision problem (cf. Section 5.2). Next, we present an algorithm to solve the optimization model efficiently with focus on central features (cf. Section 5.3). More precisely, normative analytical techniques (i.e., optimization model; cf. Meredith et al., 1989) and algorithms (i.e., heuristic techniques; cf. Will et al., 2002) are used for the multi user context-aware service reselection in order to select a near-optimal solution at runtime with high performance. In the evaluation it is assessed in detail how well the proposed algorithm supports to solve the represented mathematical model (Tedeschi, 2006). This evaluation can be done by means of mathematical techniques, simulation, benchmarking etc. and the results can be compared to actual measured results from other approaches (Prat et al., 2015). Thus, the design of our evaluation follows the compositional styles demonstration as well as simulation- and metric-based benchmarking of artifacts (Prat et al., 2015), in which the efficacy, performance or robustness of the artifact is measured and compared with those resulting from other approaches (cf. Section 6.1 and 6.2). Finally, the discussion of the solution, its effectiveness and importance must be presented appropriately to researchers and other relevant audiences such as practitioners. Therefore, Section 6.3 of this paper discusses the results and implications for science and practitioners.

4 Real-world Scenario

The importance of service systems and the associated support of mobile-enabled processes is continuously increasing (Deng et al., 2016) and in this regard, the literature shows that the tourism sector including mobile-enabled processes are gaining in importance, too (Femenia-Serra et al., 2019). Thus, tourism represents a valuable part of the further development of service systems and, in particular, of service design (Koskela-Huotari et al., 2021). To illustrate our approach in the following, we introduce a scenario, which considers an excerpt of a mobile-enabled process of a tourist portal based on a day trip. Therefore, we adopt the structure of this process directly from the service science literature (Hara et al., 2016; Oizumi et al., 2013). In particular, these works describe a day trip process that is composed of a set of activities (e.g., eating) and each associated location of an activity is connected by a transport (e.g., bus). In this regard, a process is further defined as a set of tasks arranged in a temporal order (Corradini et al., 2007) and contains the following tasks in our scenario: *Breakfast or Café*, *Restaurant*, *Museum* or *Park* (cf. Fig. 3).

In the following, we instantiated this process for a day trip based on real-world data of the city of Melbourne (Australia). More precisely, in our scenario, three users participate in the illustrated process and can perform certain tasks such as Breakfast or Café independently of each other but can also conduct tasks together (based on their preferences). A task which is in the focus of the potential simultaneous execution among multiple users is defined as so-called *focus task* in the following. For each (focus) task different candidate services are available. These candidate services (e.g., the Restaurants "Da Guido 365″ and "Little Billy" for task *"Restaurant", cf.* Fig. 3) built on real-world data of the city of Melbourne and are described for example by *name*, *duration*, *opening hours, price, weather suitability* and *GPS coordinates*. In the city of Melbourne there are a plethora of candidate services per task that users can select during a city day trip. For example, in downtown Melbourne there are about 1,250 restaurants, 1,250 cafés, 250 candidate services for arts such as museum, 105 sights such as parks and 220 locations for having breakfast. Given this high number of real candidate services, there exist on average about 1.2 billion possible solutions (i.e., service compositions) per user to realize the city day trip demonstrating the complexity of the selection task. In addition, each user in the process must be transported from a selected candidate service to the next candidate service. This transport is conceptualized by its own type of a task (i.e., transport task) and contains

Table 2 Details of the presented real-world example of *Melbourne*

Aspect (as used in service science literature)	*Real-world example*
Process	Day trip in the city of Melbourne containing nine tasks
Task	*Task 1:* Transport *Task 2 / 3:* Breakfast or Café *Task 4:* Transport *Task 5:* Restaurant *Task 6:* Transport *Task 7 / 8:* Museum or Park *Task 9:* Transport
User	Three users can perform certain tasks such as Breakfast or Café independently of each other but can also conduct tasks together (based on their preferences)
User Preference	Favorite scores for each type of candidate service, etc
Candidate Service *(based on real-world data)*	*Task 2 / 3:* 1,250 Cafes and 220 Locations for having breakfast *Task 5:* 1,250 Restaurants *Task 7 / 8:* 250 Museums and 105 Sights *Task 1, 4, 6, 9:* Car, Bicycle, Walk, Public Transport
Attribute *(based on real-world data)*	Name, duration, opening hours, price and type for each candidate service, etc
Context *(based on real-world data)*	Weather suitability and GPS coordinates for each candidate service, etc
Event Type	Disruptive Events such as closed restaurant, changing weather, exceeding opening hours, etc
Event Occurrence *(based on real-world data)*	Disruptive Events depend on: Ø congestion level in Melbourne per year: approx. 23% Ø share of public transport delay in Melbourne per year: approx. 14.4% # of average rainy days in Melbourne per year: approx. 100 etc

candidate services described by attributes such as *name*, *duration*, *price*, *type* or *favorite score*. Thus, additionally to the five tasks already introduced above, four more transport tasks have to be considered, resulting in nine tasks in sum. In Melbourne there is a wide range of candidate services for transport and therefore cars (including car sharing), bicycles (including bike sharing), walks and Melbourne's public transport, in particular metropolitan trams, metropolitan trains and the Melbourne city bus, can be used. Besides tasks and their candidate services, world states are required to cope with context-awareness (cf. Ghallab et al., 2008; Heinrich & Schön, 2015). In particular, each user has an initial world state based on the initial context (e.g., time and GPS position to start the city day trip) and a corresponding end world state (e.g., time and GPS position at the end of the city day trip).

During a city day trip different types of disruptive events can occur such as closed restaurants, changing weather conditions (e.g., it starts raining) or exceeding opening hours due to delays in the process. Indicators for the occurrence of such disruptive events can be obtained from publicly available information on Melbourne. For example, in 2020 there was a total congestion level of 23%, which extends a 30-minutes drive of the transport candidate service *car* to an average *duration* of 37 min, while at rush hour the congestion level can even reach 39% (TomTom, 2021). Furthermore, 14.4% of all transports in Melbourne of the transport type "metropolitan tram" were not in time in 2019 and therefore the passengers arrived at their destination at least six minutes late (Public Transport Victoria, 2019) also leading to an increase in the attribute *duration*. In addition, in 2020 Melbourne had an average of 100 rainy days per year (Australian Government, 2021), potentially influencing the attribute *weather suitability* of a candidate service (e.g., the restaurant "Da Guido 365" has a weather suitability of "sunshine" because there is only outdoor seating available). Such disruptive events enforce a re-selection at runtime in case the predetermined service compositions of multiple users at planning time are no longer feasible for instance. The most important details from the presented real-world example are summarized in Table 2.

5 Model

5.1 Basic Notations

In this section, we introduce the basic notations for our multi user context-aware service system that can serve as a foundation for the service re-selection approach.

1 We consider a process p referring to a set of tasks T containing all tasks of the process p and $t \in T$ with t describing a single task. A process p is conducted by

one user $a \in A$ or multiple users with A describing the set of users.

2 There is a set of focus tasks F with $F \subseteq T$, which can be conducted by multiple users simultaneously. Furthermore, there is a set of tasks I with $I \subseteq T$, which are executed by each user individually (non- simultaneously). In particular, it holds: $I \cup F = T \wedge I \cap F = \emptyset$. As the sets of tasks T, F and I are conducted by each user, we denote these sets also as T_a, F_a and I_a when referring to a user $a \in A$.

3 Each task t refers to a set of candidate services CS_{at} including the functional equivalent candidate services for a user a and a task t. Thereby, a single candidate service is defined as $cs_{ats} \in CS_{at}$, with s describing the index of the corresponding candidate service.

4 Each candidate service cs_{ats} is characterized by a set NCA of non-context-aware attributes with $nca_n \in NCA$ describing one single attribute such as rating and with n describing the index of the corresponding non-context-aware attribute and a set CA of context-aware attributes with $ca_l \in CA$ describing one single attribute such as time of day and with l describing the index of the corresponding context-aware attribute. The sets NCA and CA form the set of non-functional properties NFP (with $NCA \cup CA = NFP$).

5 To address context-awareness, we use the concept of world states (cf. Section 4). In detail, the set of world states WS_{at} represent the possible context information of user a in a task t. Each world state is defined as ws_{atk} with $ws_{atk} \in WS_{at}$ and with k describing the index of the corresponding world state.

6 To determine the value of a context-aware attribute ca_l the combination of a candidate service and a world state is needed (e.g., GPS coordinates and service destination). This combination refers to a world-state-candidate-service combination wsc_{aty} with $wsc_{aty} \in WSC_{at}$ where WSC_{at} is the set of all world-state-candidate-service combination for a user a in a task t and y describing the index of the corresponding world-state-candidate-service combination.

7 To enable spatial–temporal coordination between multiple users for each focus task in the set of focus tasks F, we use the concept of common world-state-candidate-service combination. A world-state-candidate-service combination wsc_{aty}, which refers to the same world state ws_{atk} and candidate service cs_{ats} by all users of set A is defined as a common world-state-candidate-service combination $cwsc_{tz}$. Thereby, it holds $cwsc_{tz} \in CWSC_t$ with $CWSC_t$ is the set of all common world-state-candidate-service combinations for the focus task t and z describing the index of the corresponding common world-state-candidate-service combination.

8 A service composition sc_a is noted as a realization of a process p for user a in the form of a set of world-state-candidate-service combinations and a set of common world-state-candidate-service combinations with exactly one (common) world-state-candidate-service combination for each task t of the process p.

9 A service composition sc_a for a user a can refer to a global constraint $con_a^{nca_n}$ for each non-context-aware attribute nca_n and to a global constraint $con_a^{ca_l}$ for each context-aware attribute ca_l.

10 We define the value of a non-context-aware attribute related to a candidate service cs_{ats} as $q_{ats}^{nca_n}$ (e.g., for a specific restaurant, the average star rating is 4.8) and the value of a context-aware attribute ca_l related to a world-state-candidate-service combination wsc_{aty} as $q_{aty}^{ca_l}$ (e.g., time of day to visit a restaurant is 12 a. m.). A function to aggregate the quantified values $q_{ats}^{nca_n}$ and $q_{aty}^{ca_l}$ for all attributes to a single utility value is represented by a utility function U. Here, we adopt the widely used utility function described in detail by Alrifai and Risse (2009), which applies the SAW (simple additive weighting) method.

11 At runtime a disruptive event ev_{ad} can occur with $d = 1toD_a$ and D_a defining the number of events for user a. An event ev_{ad} can affect an arbitrary candidate service cs_{ats} or several candidate services, for instance, an event causes a candidate service to become no longer available.

5.2 Optimization Model

To consider multiple users, context-awareness and in particular disruptive events in our approach for service re-selection at runtime, we establish a stateful representation by explicitly modelling a state space containing world states and candidate services. In the following, we present our optimization model for service re-selection at runtime.

Objective Function:

$$\sum_{a \in A} \sum_{t=1}^{T_a} \sum_{y=1}^{WSC_{at}} U\left(wsc_{aty}\right) * wsc_{aty} \rightarrow \max \qquad (1)$$

s. t.:

one CS per Task:

$$\sum_{s=1}^{CS_{at} \setminus BL} cs_{ats} = 1 \ \forall\, t = 1, \dots, \ T_a; \forall\, a = 1, \dots, A \qquad (2)$$

one WS per Task:

$$\sum_{k=1}^{WS_{at}} ws_{atk} = 1 \ \forall\, t = 1, \dots, T_a; \forall\, a = 1, \dots, A \qquad (3)$$

one WSC per Task:

$$\sum_{y=1}^{WSC_{at}} wsc_{aty} = 1 \; \forall \; t = 1, \ldots, T_a; \; \forall \; a = 1, \ldots, A \tag{4}$$

WSC Constraint:

$$\sum_{y=1}^{WSC_{at}} CS\left(wsc_{aty}\right) + WS(wsc_{aty}) - 2 * wsc_{aty} \geq 0 \; \forall \; t = 1, \ldots, T_a \; \forall \; a = 1, \ldots, A \tag{5}$$

WS Constraint:

$$\sum_{y=1}^{WSC_{at}} CREATE\_WS(wsc_{aty}) - wsc_{aty} \geq 0 \; \forall \; t = 1, \ldots, (T_a - 1); \; \forall \; a = 1, \ldots, A \tag{6}$$

one CWSC per Focus Task:

$$\sum_{z=1}^{CWSC_t} cwsc_{tz} = 1 \; \forall \; t = 1, \ldots, F \tag{7}$$

Focus Task Constraint:

$$\sum_{z=1}^{CWSC_t} \left(|A| * cwsc_{tz} - \sum_{a=1}^{A} CWSC(a, cwsc_{tz}) \right) = 0 \; \forall \; t = 1, \ldots, F \tag{8}$$

Non-Context-aware Attribute Constraints:

$$min, \sum_{t=1}^{T_a} \sum_{s=1}^{CS_{at}} q_{ats}^{nca_n} * cs_{ats} \leq con_a^{nca_n} \; \forall \; nca_n \in NCA^-; \; \forall \; a = 1, \ldots, A \tag{9}$$

$$max, \sum_{t=1}^{T_a} \sum_{s=1}^{CS_{at}} q_{ats}^{nca_n} * cs_{ats} \geq con_a^{nca_n} \; \forall \; nca_n \in NCA^+; \; \forall \; a = 1, \ldots, A \tag{10}$$

Context-aware Attribute Constraints:

$$min, \sum_{t=1}^{T_a} \sum_{y=1}^{WSC_{at}} q_{aty}^{ca_l} * wsc_{aty} \leq con_a^{ca_l} \; \forall \; ca_l \in CA^-; \; \forall \; a = 1, \ldots, A \tag{11}$$

$$max, \sum_{t=1}^{T_a} \sum_{y=1}^{WSC_{at}} q_{aty}^{ca_l} * wsc_{aty} \geq con_a^{ca_l} \; \forall \; ca_l \in CA^+; \; \forall \; a = 1, \ldots, A \tag{12}$$

Our optimization model consists of an objective function (1) determining the overall utility value and the constraints (2) to (12). The objective function (1) determines the overall utility by summing up the utility scores $U(wsc_{aty})$ of all selected world-state-candidate-service combinations, over all tasks in T_a and all users in A in the process p with the goal to maximize the accumulated utility value over all selected users' service compositions. To consider only the selected world-state-candidate-service combinations, we use a decision variable wsc_{aty} for each world-state-candidate-service combination where wsc_{aty}= 1 notes that wsc_{aty} is selected and $wsc_{aty} = 0$ that wsc_{aty} is not selected. For re-selecting, constraint (2) ensures that exactly one candidate service cs_{ats} from each task in T_a is selected for each user $a \in A$, which is not on a blacklist BL containing all candidate services that cannot be used for a re-selection due to an event ev_{ad}. To hold the condition that for each user $a \in A$ and for each task in T_a exactly one world state ws_{atk} must be selected, the constraint (3) is also be part of our optimization model. Constraint (4) is used to assure that exactly one wsc_{aty} for each user $a \in A$ is selected for each task in T_a. The constraint (5) links each wsc_{aty} to its corresponding cs_{ats} and ws_{atk} (cf. Section 5.1), which is realized by $CS(wsc_{aty})$ and $WS\left(wsc_{aty}\right)$, within each task in T_a and for each user in A. Further, by constraint (6) is ensured that the resulting world state in the subsequent task (represented by $CREATE\_WS(wsc_{aty})$) for a wsc_{aty} also has to be selected within each task in T_a (except the last task) and for every user in A.

Constraint (7) integrates the concept of common world-state-candidate-service combinations, which is necessary to ensure the simultaneous use of the same world state ws_{atk} and candidate service cs_{ats} by multiple users $a \in A$ with respect to user-based dependencies (cf. Section 2.1). Therefore, we guarantee that exactly one $cwsc_{tz}$ for each focus task in F is selected for all users participating in the process. The constraint (8) checks whether all world-state-candidate-service combinations regarding a selected $cwsc_{tz}$ (i.e., we use the decision variable $cwsc_{tz}$ that is 1 if the corresponding $cwsc_{tz}$ is selected and 0 if not) are used in each users' service composition (represented by $CWSC(a, cwsc_{tz})$).

To ensure feasible solutions, the constraints (9), (10), (11) and (12) consider the global end-to-end constraints for non-context-aware attributes $nca_n \in NCA$ and context-aware attributes $ca_l \in$ CA defined for each user $a \in A$. In this regard, the sets NCA and CA are each divided into a subset NCA^- respective CA^- (with $NCA^- \subseteq NCA$ and $CA^- \subseteq CA$) of attributes that need to be minimized and a subset NCA^+ respective CA^+ (with $NCA^+ \subseteq NCA$ and $CA^+ \subseteq CA$) that need to be maximized. In particular, the constraints (9) and (10) describes if all selected candidate services (i.e., in this regard, we use the decision variable cs_{ats} that is 1 if the corresponding candidate service cs_{ats} is selected and 0 if not) from each task in T_a of the process meet the users' (a) constraints $con_a^{nca_n}$ in regard to all non-context-aware attributes $nca_n \in NCA$. Therefore, all minimizing attributes must not exceed a defined upper constraint with regard to their value $q_{ats}^{nca_n}$ (e.g., the maximum price limit for a user to have lunch in a restaurant is 10 Euro, cf. constraint (9)). Likewise, all maximizing attributes must not fall below a defined lower constraint with regard to their value $q_{ats}^{nca_n}$ (e.g., the average rating of a restaurant where the user is having lunch should be at least 4 stars, cf. constraint (10)). In order to incorporate context-based dependencies, the constraints (11) and (12) describe analogously the feasibility to the global end-to-end constraints of context-aware attributes, which means that the value $q_{aty}^{ca_l}$ is dependent on the selected wsc_{aty} for each task in T_a.

An overview of the formal notation of the optimization model can be found in the appendix of this work.

Table 3 Algorithm for multi user context-aware re-selection

```
Function: MultiUserContextAwareServiceReselection()
Input: Process p, CandidateServices cs[][], World States ws[][], BlacklistedItems bl[], State Space Measures ssm[][], Events ev[], Users a[], State Space sp[][]
Output: A feasible solution regarding the underlying selection problem

BEGIN
  feasibilityInformation[] <- doFeasibilityChecks (p, ws[][], a[])
  region <- setInitialReselectionArea (p, ev[], feasibilityInformation[], a[])
  DO
     fixedWS[] <- fixWorldStates (region, ws[][], a[])
     sp[][] <- extendCurrentStateSpace (region, ws[][], fixedWS[], a[])
     sp[][] <- reduceCurrentStateSpace (region, ws[][], fixedWS[], cs[][], bl[], a[])
     reselectionSuccess <- reselect (p, region, sp[][], bl[], a[])
     IF (reselectionSuccess==false)
      ssm[][] <- getIndicatorForRegionalExpansion (p, region, a[])
      region <- expandReselectionArea (region, p, ev[], ssm[][], a[])
     ELSE
      WHILE (userExpectsAlternative==true & reselectionSuccessAlternative==true)
           reselectionSuccessAlternative <- reselect (p, region, sp[][], bl[], a[])
           userExpectsAlternative <- getUserPreferencesForAlternatives(a[])
      DO
    END IF
  WHILE (reselectionSuccess==false)
END
```

5.3 Algorithm for Multi User Context-Aware Service Re-Selection

The optimization model introduced above relies on a state space comprising candidate services, world states and resulting world-state-candidate-service combinations. This state space must be generated starting with the initial world state of each individual user and then determining the corresponding state transitions throughout the process until the end world state is reached (Bortlik et al., 2018; Ghallab et al., 2008; Heinrich & Schön, 2015). After that, its size must be delimited in such a manner that it allows the optimization model to find a near-optimal solution at runtime with high performance. The algorithm presented in this section addresses this generation of the state space. At the core of the algorithm, the part of the state space in which the disruptive event occurs is generated by a region-based approach in order to enable a performant selection of a solution (Section 5.3.1). In case that no feasible solution exists in the initial region, the algorithm carefully extends to further regions based on a state space measure indicating whether the extensions are promising (Section 5.3.2). Thereby, the algorithm investigates the feasibility of the state space in the considered region and also checks in advance whether there can exist feasible solutions at all before conducting the optimization model (cf. Section 5.2). In case, a region with feasible solutions is found, the resulting state space is examined by our optimization model for multi user context-aware re-selection (cf. Section 5.2). Finally, the algorithm analyses whether several alternative solutions can be selected (Section 5.3.3). The following Table 3 introduces the concepts of our algorithm at a glance, which are discussed in the Sections 5.3.1 to 5.3.3 in detail.

5.3.1 Algorithm for Re-Selection Based on a Single Region

Focus Task-Based Definition of Regions In a first step, the overall process is divided into regions (cf. Fig. 4) for each individual user based on the users' state spaces from the planning time, which aims to limit the search space and to preserve a large part of the process from change. To illustrate our region-based approach, it is assumed w. l. o. g. in the following that User 1 is located in Region 1 of the process and the disruptive event occurs in Region 3 (e.g., closed restaurant). Therefore, a re-selection within only Region 3 is focused on:

Each of these regions within the process refers to exactly one focus task (e.g., a restaurant) and further consists of particular tasks of type Wait, Transport and again Wait (cf. Fig. 4).

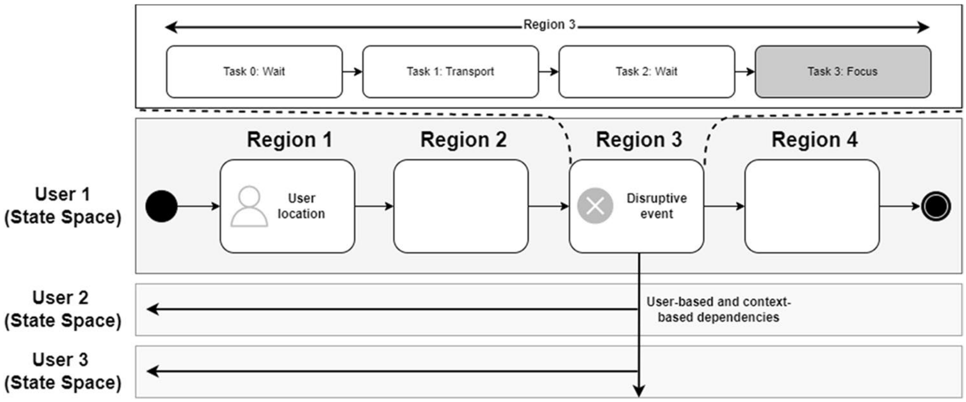

Fig. 4 Region-based approach

The delimitation of a region with exactly these four tasks is mandatory in our approach as it can ensure spatial–temporal coordination and it further allows for a small (atomic) size of the region to keep the re-selection effort low and the robustness high: More precisely, based on the state space, user-based dependencies and context-based dependencies such as time of day and location between multiple users (cf. Section 2.1, cf. also Fig. 4) have to be considered within the defined region in order to select the most suitable services, while dealing with individual context information of users and their coordination. Specifically, in multi user context-aware processes, *waiting tasks* (i.e., Task 0 and Task 2, e.g., waiting time for conducting the transport or focus task) are an important concept to represent a temporal-based coordination between multiple users (e.g., cf. Mayer, 2017), while a *transport task* (i.e., Task 1, e.g., User 1 is transported to another restaurant when an event occurs) support and enable spatial coordination between multiple users.

In order to preserve the context of a user in terms of time and location, the boundaries of the region to be re-selected must be set. Thus, we are the first approach which defines these boundaries in a stateful process based on *world states* as they represent context information in regard to time and location. Therefore, world states can be used to determine effectively the beginning and end of the region in which the disruptive event occurs and a re-selection must be performed. In this regard, Fig. 5 shows exemplary a simplified state space in which the state transitions between world states are represented (for illustration purpose, without their associated candidate services and world-state-candidate-service combinations). The red path constitutes the optimal service composition selected at planning time whereas the green path shows the optimal service composition after re-selection. Given this, the boundaries of Region 3, for instance, for the planned selected service composition can be determined. More precisely, world state ws_{300} is set as the beginning of Region 3 because this world state is automatically determined by the previous part of the service composition (i.e., by world state ws_{230} and its related candidate service). Furthermore, world state ws_{401} is set in the succeeding service composition (i.e., Region 4) as this world state needs to be reached again after the re-selection in Region 3 in order to maintain the state transitions between these neighboring regions.

Event-based re-structuring of the regions state space When a disruptive event occurs, the context within the state space may change for one or more users. As a result, on the one hand, states which reflect the changed context of these users are potentially not present in the state space and, on the other hand, already existing states in the state space may no longer be feasible. Building a completely new state space is very time-consuming and not promising for a runtime approach. Therefore, an elaborated state re-structuring is proposed at runtime, which includes only feasible context information in the state space. In particular, the state space re-structuring consists of three steps: *State space extension*, *State space reduction* and *Feasibility checks* all aiming to obtain a feasible state space:

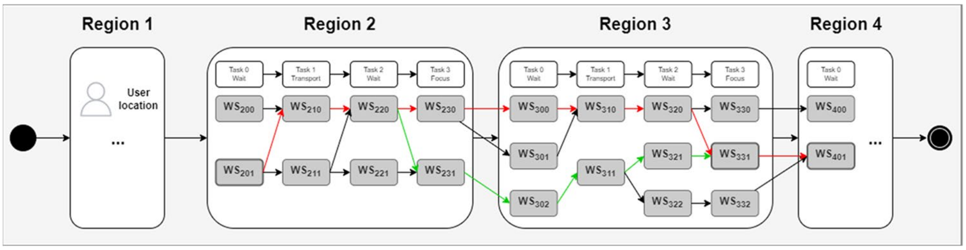

Fig. 5 Representation of state transitions within simplified state space

For *state space extension* a world state is injected in the state space, which reflects the current user context (time and location) and all resulting feasible world states and the associated paths are re-structured forward in the process. Especially, this allows to consider additional feasible states, which were not feasible at planning time. In order to support high performance and robustness, the extension of the states only takes place within the considered region and not for the entire process (i.e., Task 0 to Task 3 in Region 3, cf. Fig. 4). Furthermore, *state space reduction* removes for each individual user all world states and candidate services from the state space that are not feasible (while considering user-based and context-based dependencies within the region, cf. Fig. 4). In particular, only world states and candidate services are considered, which can be re-selected with respect to the fixed beginning and end world state of a region (e.g., Region 3, cf. Fig. 5). This results in a sub state space (i.e., Task 0: $\{ws_{300}\}$, Task 1: $\{ws_{310}\}$, Task 2: $\{ws_{320}\}$, Task 3: $\{ws_{331}\}$) in which large parts of the state space can be excluded from the re-selection process. Finally, we extend the algorithm with *feasibility checks* in order to check the feasibility of the sub state space before a re-selection takes place. For a region, context information is analyzed to check whether feasible solutions can exist in the sub state space at all before executing our optimization model (cf. Section 5.2). In particular, feasibility checks can recognize when no feasible solution can be selected within a regions state space and a region-based expansion is necessary anyway. In this regard, we focused on feasibility checks related to time (e.g., there is no feasible candidate service in the region referring to opening times and the users' context) and location (e.g., there is no feasible transport in the regions state space referring to transport duration). In case that the sub state space contains feasible solutions, the optimization model introduced in Section 5.2 can be performed in order to find a new feasible and near-optimal solution based on this re-structured sub state space of the region.

5.3.2 Algorithm for Re-Selection Based on Region-Based Expansions

In case that within a single region (e.g., Region 3) no feasible solution can be re-selected, further regions have to be considered for re-selection. Precisely, if the initially selected service composition for Region 3 (i.e., marked red path in Region 3, cf. Fig. 5) is potentially no longer feasible (e.g., the candidate service for world state ws_{310} is no longer available and no alternative candidate service exists) other paths may be feasible by considering further regions. Thus, alternative world states have to be re-selected within the process (e.g., marked green path world state ws_{231} instead of world state ws_{230}, cf. Fig. 5). These expansions of the state space have to be done carefully to address computational complexity. To achieve this, we describe two extensions for the presented elaborated region-based algorithm: 1) Expansion of regions based on a state space measure (indicator), and 2) Examination of the feasibility of the state space.

Region-Based Expansion Based on State Space Measure The expansion with a neighboring region (i.e., Region 2 or Region 4, cf. Fig. 5) has a great impact on the performance and solution quality of the re-selection and must therefore be performed in an elaborate manner. In this regard, we choose the neighboring region based on the state space measure *number of world states of all users per region.* This can be reasoned, as adding the neighboring region with the largest number of world states (summed over all users) results in a larger search space and is therefore more promising for finding a feasible solution. Although the time for re-selection of the larger search space results in a higher computation time (i.e., due to the higher number of world states), in general fewer expansions to further regions are required. This leads to a tradeoff between the number of region-based expansions and the size of the considered state space until a feasible solution can be found. However, searching within a larger initial search space is – from the perspective of a heuristic – faster than performing several stepwise expansions each including re-structuring of the state space. Further, we use the measure *number of world states* from the planning state space (as an indicator) because determining the number of world states of the re-structured state space at runtime is very time-consuming. If after expansion no feasible solution can be re-selected in the joined regions (e.g., joining of Regions 2 and 3), a further region is added (e.g., Region 4) at a time (based on the above-mentioned state space measure) until a feasible solution is found by the optimization model or the end of the process is reached (then the heuristic will fail because no feasible solution can be found).

Examining the Feasibility of the State Space As stated in the last sub section, time-consuming expansions must be reduced in order to support high performance. Therefore, our algorithm is able to manage the size and feasibility of the regions state space (cf. Section 5.3.1) in order to select a near-optimal solution with the optimization model presented in Section 5.2. In that respect, feasibility checks provide a sophisticated procedure to determine a feasible state space. In particular, our feasibility check can not only determine that no feasible solution can be re-selected in a single region, but can directly determine the size of a larger region in which the re-selection of a feasible solution is promising (i.e., e.g., in Region 3 there is no feasible solution, then an expansion and re-selection to Regions $2 \cup 3$ takes place directly). In this regard, the restructuring of the state space (i.e., extension and reduction) can be applied dynamically to regions of any size. If nevertheless no

Table 4 Summary of the contribution to the presented algorithm

Concept	*Feature*				*Contribution*
	Performance	Solution quality	Solution robustness	Alternative solutions	
Re-selection based on a single region (Section 5.3.1)	*X*	*X*	*X*		- Determination of the user context in terms of time and location by fixing the boundaries of a region with world states - Dynamic state space extension to address changing user context during re-selection - Dynamic state space reduction to only feasible states, while maintaining user-based and context-based dependencies to neighboring regions
Re-selection based on region-based expansions (Section 5.3.2)	*X*	*X*	*X*		- Elaborated extension of the search space by using a state space measure in order to reduce the number of region-based expansions - Examining the feasibility of the state space in order to reduce the number of time-consuming re-selection steps
Selection of additional solutions (Section 5.3.3)	*X*	*X*		*X*	- Providing proactively several additional solutions at runtime - Re-using already performed re-selection steps (e.g., state space extension or state space reduction) from the initial solution determined at runtime

feasible solution can be re-selected (e.g., due to context interdependencies between multiple users) in the joined regions, a further region must be dynamically added step-by-step until a feasible solution is found (cf. Region-based expansion based on state space measure).

5.3.3 Algorithm for Selecting Alternative Solutions

The re-selected solution may not meet the users' expectations due to, for example, imprecise or incomplete context information or changing user preferences. Therefore, it is important that the algorithm can select several alternative and feasible solutions in near-time with a high solution quality. An alternative solution is given if at least one candidate service from the re-selected solution is modified in the focus task within the considered region even if the rest of the solution still matches the previously selected solution. In this regard, we blacklist this candidate service (or potentially more candidate services based on user preferences) and re-select the alternative solution within the considered region with the (next) higher utility. All steps and information of the region-based approach (e.g., cf. Focus task-based definition of regions, Event-based re-structuring of the regions state space) can be reused from the previous re-selection resulting in significant performance advantages. This approach enables the selection of several alternative solutions while maintaining high performance and solution quality.

The following Table 4 summarizes the contribution and shows which features of the solution are supported by the presented region-based multi user context-aware service system:

6 Evaluation

In this section, we evaluate our heuristic technique regarding the features performance, solution quality, solution robustness and alternative solutions, as discussed in the methodical foundations. The design of our evaluation follows the compositional styles demonstration as well as simulation- and metric-based benchmarking of artefacts (cf. Prat et al., 2015, cf. Section 3). Thus, the evaluation is conducted by means of a *simulation experiment* based on a real-world data set in the tourism domain (cf. Section 4). Further, in order to put the quality of the proposed heuristic (in the following abbreviated with *MUCARS (multi user context-aware re-selection)*) into perspective with respect to the above features, a comparison to existing multi user context-aware service systems is performed.

6.1 Setup and Data Preparation

When aiming at a comparison to extant approaches, it is necessary that these works can cope with multiple participating users, context-awareness and disruptive events. As discussed in the Section Related Work, there are no approaches that address these three aspects. The only technique that considers user-based as well as context-based dependencies and represents a heuristic explicitly focusing on performance as a key criterion for evaluation is the one presented by Bortlik et al., 2018 (in the following abbreviated with *MUCAHA (multi user context-aware heuristic approach)*). For addressing disruptive events, we run the *MUCAHA* approach at every event to the relevant part of the process (i.e., start with the task the user conducts until the end of the process) to be able to handle these events. The second work by Heinrich and Mayer (2018) also dealing with multiple participating users and context-aware service selection presents an approach that provides an optimal (exact) solution at planning time (in the following abbreviated with *MUCAOS (multi user context-aware optimal solution)*). This means, in a runtime setting showing (very) high computational complexity, a broad comparison regarding runtime relevant features like performance, solution robustness and alternative solutions is not meaningful and above all not possible (i.e., finding an optimal solution in appropriate time is not realistic since the underlying decision problem is NP-hard). Thus, we can use *MUCAOS* for comparison purposes only in individual cases (e.g., selected simple settings to determine the feature solution quality).

For our evaluation we draw on a real-world data set in the tourism domain (cf. Section 4). More precisely, in the initially introduced *basic setting*, three users conduct a day trip in the city of Melbourne, which comprises three focus tasks (e.g., having breakfast, having lunch, visiting museum; in an extended setting further tasks such as go shopping are considered). Therefore, we have extracted real-world data from a very popular web portal, which provide information about local businesses. As a result, we obtained candidate services for the tasks *Breakfast*, *Shopping*, *Restaurant*, *Arts*, *Café*, *Sight*, *Beauty* and *Bar* and their corresponding non-functional properties. In particular, for these candidate services, the real data on prices, ratings, locations and distances to parking were extracted. In order to perform the simulation experiment, few further data was defined in addition to real-world data: First, based on the non-context-aware and context-aware attributes and the corresponding non-function properties values, the *constraints of the users* were defined. Moreover, the durations of the selected candidate services were determined depending on the candidate service type (i.e., 15 – 120 minutes in steps of 15 respective 30 minutes). To enable the *transport* of the users between tasks, the candidate services *Walk*, *Bike* and *Car* represent the transport options for each transport task. Therefore, corresponding durations (i.e., 15 – 60 minutes in steps of 15 minutes) and prices (i.e., walk: 0 €, bike: 1 €, car: 4 €) were generated for the different transportation types. Furthermore, to bridge possible *waiting times* of the users, five candidate services for waiting times (i.e., 0 – 60 minutes in steps of 15 minutes) were modeled in the waiting tasks. In addition to these careful definitions, *user-specific data* were determined randomly in each simulation run based on existing real-world data, since no real users are present. In this regard, each user has his own context within the process. Therefore, for each simulation run we randomly create the start and end context (e.g., time and location in the city of Melbourne) for each individual user as well as the values for their preferences (including favorite scores for different candidate service types and transports) for all non-context-aware and context-aware attributes within defined minimum and maximum values. Consequently, we use the non-context aware attributes price, rating, duration, waiting time and favorite score as well as the context-aware attributes location area and parking distance for the description of candidate services in this process. Furthermore, the occurrences of the disruptive events were also randomly generated. To evaluate arising events during the city day trip at runtime (e.g., closed restaurant), we simulate the resulting service failures in the model by randomly determining a task in which the real-world event occurs (i.e., within the set of tasks in the process that the user has not yet conducted). Therefore, a task is randomly selected whose predetermined candidate service from the service composition is no longer available. This procedure ensures that events occur in alternating tasks and thus resulting re-selections are performed on different parts and sizes of the process. In summary, the *basic setting* contains three focus tasks, three users, five events and 50 randomly selected real-world objects (candidate services) per focus task in the area of Melbourne, Australia. Table 5 summarizes the data for the initial basic setting.

Based on the described basic setting, we define four *extended settings*. More precisely, in each extended setting we stepwise increase exactly one parameter (i.e., the number of focus tasks, candidate services, user or events), while all other parameters of the basic setting remain the same (i.e., ceteris paribus). This allows us to analyze the effects of the modified parameters on our evaluation features (e.g., solution quality). The intervals and steps resulting for each extended setting are shown in the following table (cf. Table 6).

In our simulation experiment, we examine the evaluation features *performance*, *solution quality*, *solution robustness* and *alternative solutions*. Each of the interval steps of the extended setting is simulated twenty times and, on this basis, we determine the average results for each evaluation criterion. 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 feasibility checks. After this, each simulation run was performed on an Intel Xeon E5-2650 v4 processor, 512 GB RAM, Debian 9, Java 1.8, and the mathematical solver SCIP Optimization Suite 7.0.1.

Table 5 Data preparation for the initial basic setting

Data Type	*Parameter*	*Task*	*Data preparation*
Real-world data	Price (non-context-aware)	Focus Task	real price
	Rating (non-context-aware)	Focus Task	real rating
	Location Area (non-context-aware)	Focus Task	real location area
	Parking distance (non-context-aware)	Focus Task	real parking distance
Particular definitions of data	Duration (non-context-aware)	Focus Task	15—120 minutes in steps of 15 respective 30 minutes depending on the respective candidate service type
		Transport Task	15—60 minutes in steps of 15 minutes
	Price (non-context-aware)	Transport Task	walk: 0 €; bike: 1 €; car: 4 €
	Waiting Time (non-context-aware)	Waiting Task	0—60 minutes in steps of 15 minutes
	Users' Constraints	-	(0.99 * max. possible aggregated non-functional property value) for all non-context-aware and context-aware attributes for each user
Data generated in each simulation run	Users' Preferences	-	randomly generated for all non-context-aware and context-aware attributes for each user
	Users' Initial Contexts	-	start point is a randomly selected time between 11 – 12 a. m. (in steps of 15 minutes); randomly selected GPS position in the area of Melbourne, Australia, for each user (for the start and the end point)
	Event Occurrence	Focus Task	random selection of the focus task in the future part of the process according to the user location

6.2 Results

In the following, the features performance, solution quality, solution robustness and alternative solutions of the *MUCARS* are set into perspective to the *MUCAHA* heuristic and – if possible – to the *MUCAOS* exact approach. Moreover, when analyzing the results regarding solution robustness, a general baseline is needed for a comparison between the approaches. In this regard, we use the initial results of the *MUCAHA* approach at planning time without any runtime restrictions and without any disruptive events (in the following abbreviated with *BASE_MUCAHA*).

6.2.1 Performance

In this section, we analyze the performance by setting the computation time of *MUCARS* into perspective to *MUCAHA*. Thus, we define the feature *Performance* as follows:

$$Performance = \frac{Computation\ Time_{MUCARS}}{Computation\ Time_{MUCAHA}}$$

We choose to assess the performance relatively to the computation time of *MUCAHA* since this allows a comparison independent from the hardware used for the simulation experiment. With our runtime-optimized approach, we expected less increase in computation time with growing problem size, which is supported by our results in almost all extended settings.

The *MUCARS* approach can select a new solution for the user in only 1.6% on average of the computation time of the *MUCAHA* across all settings and is therefore on average 60 times faster than *MUCAHA*. This proportion remains relatively constant across all four settings (cf. Fig. 6). In particular, for Setting I with ten candidate services the *MUCARS* needs 1.9% and for 200 candidate services 0.7% of the computation time compared to the *MUCAHA* (across all values of the x-axis, the average is 1.2% (11 seconds in average by absolute numbers)). Setting II requires 1.2% for two users and 5.5% for eight users compared to the *MUCAHA* (across all expressions of the x-axis, the average is 2.7%). Furthermore, Setting III shows that the *MUCARS* needs 7.2%

Table 6 Extended settings

Setting	*Extension*	*Intervals and steps of the parameters*
I	# of candidate services per focus task	10 to 120 (in steps of 10) 140 to 200 (in steps of 20)
II	# of users	2 to 8 (in steps of 1)
III	# of focus tasks	1 to 10 (in steps of 1)
IV	# of events	1 to 20 (in steps of 1)

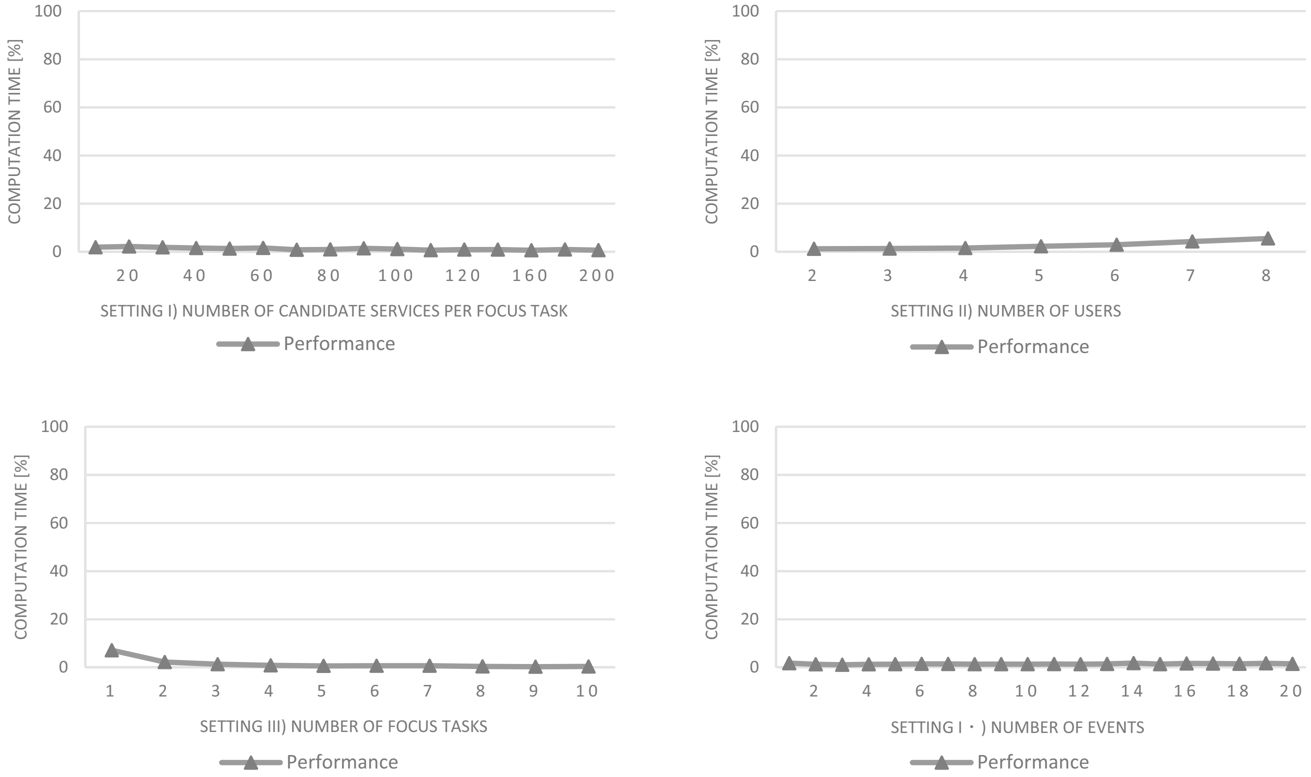

Fig. 6 Evaluation results for the criterion performance

for one focus task and 0.4% for ten focus tasks (across all expressions of the x-axis, the average is 1.5%). Finally, the *MUCARS* in Setting IV requires 1.8% for considering one event and 1.5% for considering 20 events of the *MUCAHA* runtime (across all expressions of the x-axis, the average is 1.5%). The analysis in Setting III shows that the advantages of the *MUCARS* are more significant for large processes (i.e., 7.2% performance for small processes vs. 0.4% performance for large processes) as the *MUCARS* focuses on smaller parts of the process during the re-selection (i.e., region-based approach), which is more advantageous for larger processes. When further analyzing the performance, it is noticeable that in Setting II the only case occurs in which the increase in computation time is no longer proportional for the *MUCARS* starting from six users and the runtime of the *MUCARS* increases more compared to the *MUCAHA*. On the one hand, this is due to the over-proportional increase in common world-state-candidate-services (i.e., concept to ensure the simultaneous use of the same world state and candidate service by multiple users) with increasing number of users. On the other hand, an increasing number of users leads to a smaller solution space due to preferences and restrictions among all users and thus more region expansions are necessary. Hence, both reasons influence the computation time more strongly with a high number of users than with few users leading to an over-proportional increase of the computation time. Nevertheless, the *MUCARS* is still 20 times faster with eight users than the *MUCAHA*.

Finally, the *MUCARS* approach shows a proportional increase in computation time in almost all extended settings despite increasing user-based and context-based dependencies.

6.2.2 Solution Quality

The feature *solution quality* is analyzed by comparing the Utility *U* (cf. Section 5.2) of the optimal solution provided by the *MUCAOS* approach with the corresponding Utility *U* of the approaches *MUCAHA* respectively *MUCARS* (i.e., near-optimal solution) as a percentage. Therefore, the solution quality for each multi user context-aware service selection can be defined as:

$$Solution\ quality_{MUCARS} = \frac{U_{MUCARS}}{U_{MUCAOS}}\quad Solution\ quality_{MUCAHA} = \frac{U_{MUCAHA}}{U_{MUCAOS}}$$

The *MUCAHA* reaches an average value of 93.5% and the *MUCARS* 88.8% of the solution quality over all settings in comparison to the optimal solution from *MUCAOS* at planning time (cf. Fig. 7). These results show that our

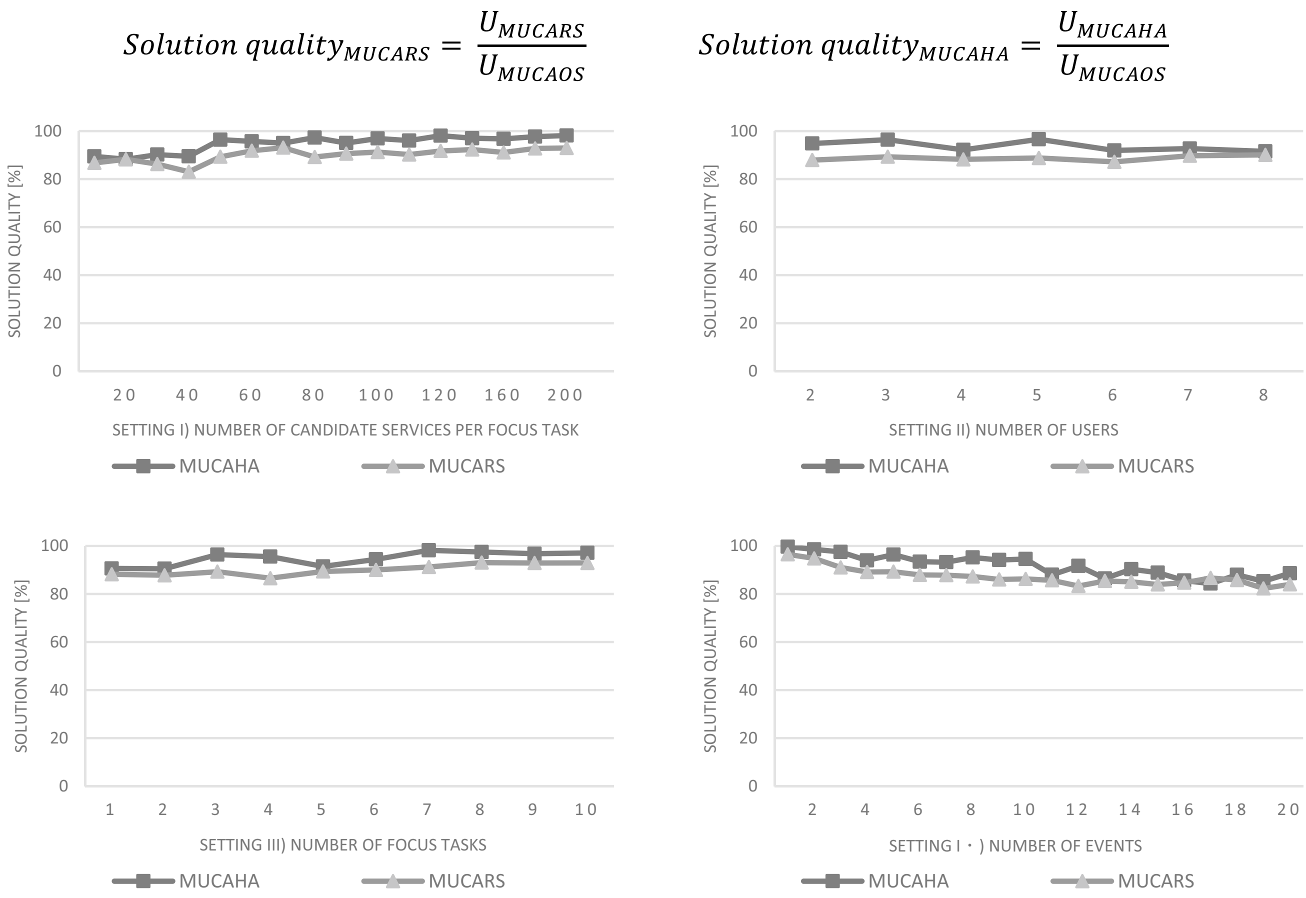

Fig. 7 Evaluation results for the criterion solution quality

approach generally achieves a high solution quality across all settings, although disruptive events are considered at runtime. Therefore, our results mark the lower bound of the actual possible solution quality, which would be even higher compared to runtime results of an exact approach (which is not realistic due to runtime complexity). In more detail, the slight increase in solution quality with an increasing number of candidate services per focus task (i.e., Setting I) can be reasoned by the fact that if a disruptive event occurs, a service composition with a higher solution quality can be re-selected if the number of suitable alternatives (=candidate services) is large. On the other hand, increasing the number of events (i.e., Setting IV) reduces the solution quality of the re-selected service composition, because the search space and thus the amount of suitable alternative candidate services decreases as the number of events increases (compared to the predetermined solution of *MUCAOS*).

In addition, the *MUCARS* approach was able to select a feasible service composition in 100% of all cases. The *MUCAHA* approach could not find a feasible solution in two cases with increasing number of events (cf. Setting IV) and therefore found a feasible solution in 99.8% of all cases. In these two cases, a failed candidate service (i.e., duration: 30 minutes) is replaceable in the same region only by candidate services, which have a longer duration (i.e., duration: 60 minutes). Because an extension of the duration is not feasible due to a defined end time of the process, no feasible solution can be selected by the *MUCAHA*.

6.2.3 Solution Robustness

Furthermore, we analyze the solution robustness across all settings indicating whether there are fundamental *changes* to the service composition determined by the *BASE_MUCAHA* at planning time. A change is any replacement of an original candidate service in one of the focus tasks of any user in the process (excepting the failed candidate service) resulting in a reduction of the solution robustness. The solution robustness for exactly one re-selection can be determined as follows:

$$Solution\ robustness = 1 - \left(\frac{\#\ ReselectedFT}{MaximumFT}\right)$$

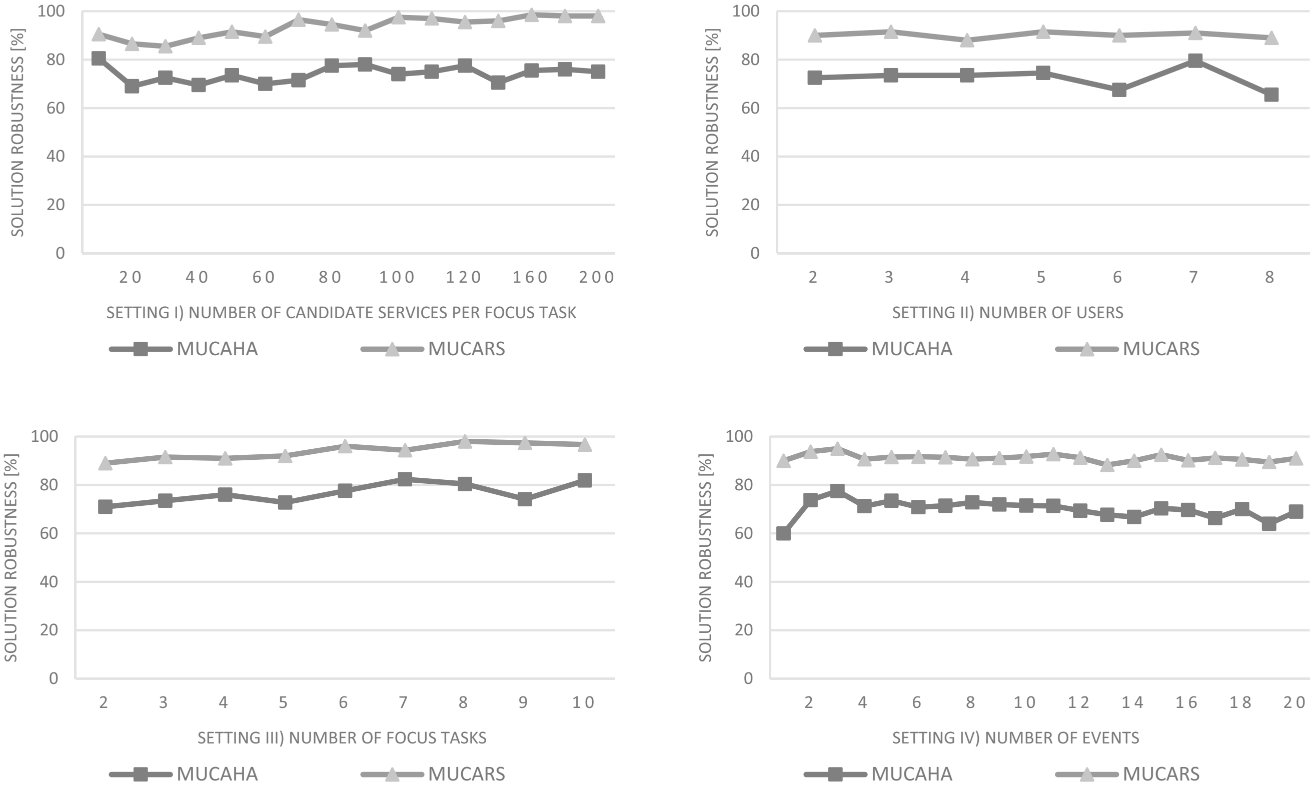

Fig. 8 Evaluation results for the criterion solution robustness

# *ReselectedFT* describes the number of focus tasks whose selected candidate service from the original service composition has changed due to a re-selection at runtime across all users. *MaximumFT* describes the maximum number of focus tasks that can change during a re-selection across all users and processes.

The *MUCARS* approach achieves a continuously high value for solution robustness with an average solution robustness of 92% across all settings (in comparison, the *MUCAHA* achieves 73%; cf. Fig. 8). In particular, the *MUCARS* achieves an average solution robustness of 94% in Setting I (*MUCAHA* 74%), 90% in Setting II (*MUCAHA*

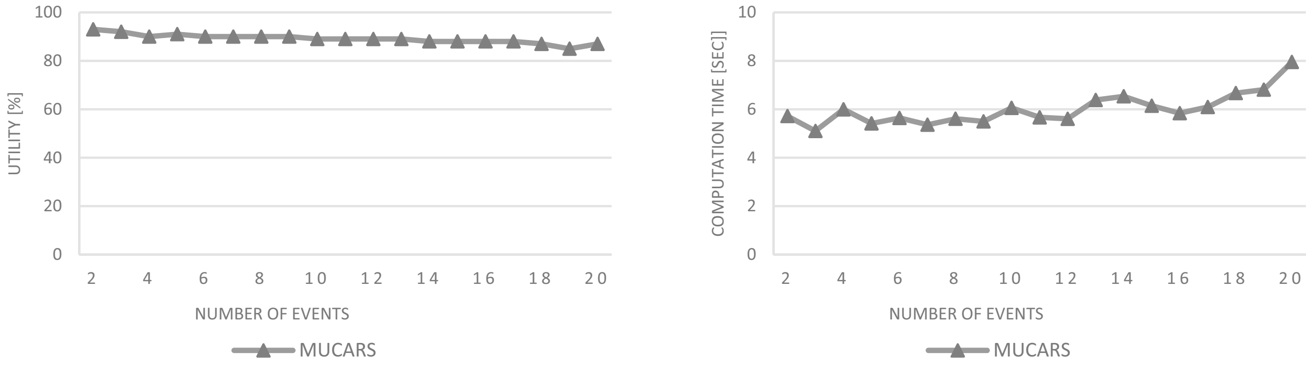

Fig. 9 Evaluation results for the criterion alternative solutions

72%), 94% in Setting III[1] (*MUCAHA* 77%) and 91% in Setting IV (*MUCAHA* 70%). Thus, the advantages of the *MUCARS* over the *MUCAHA* approach are evident in all extended settings.

6.2.4 Alternative Solutions

To evaluate our approach with regard to alternative solutions, we selected five additional alternative solutions (cf. Section 5.3.3) for the *MUCARS* approach for each problem setting towards an increasing number of events to determine both the average *solution quality* and the average *performance* (note that *MUCAHA* does not provide the functionality to determine alternative solutions; cf. Fig. 9).

Considering the highly relevant setting of an increasing number of events at runtime (cf. Setting IV), the *MUCARS* approach provides alternative solutions with an average solution quality of 89% compared to the originally selected service composition at runtime. Furthermore, an alternative solution can be selected in 100% of all settings and can be presented to the user in an average of 6.1 s per event. This further shows the high effectiveness and stability of the *MUCARS* approach.

6.3 Discussion of Results and Implications

We propose a heuristic technique that is able to consider multiple users, context-awareness and in particular disruptive events, while maintaining solutions at runtime with high *performance*, *solution quality*, *solution robustness* and also provide *alternative solutions*. Analyzing the evaluation results, *MUCARS* offers significant advantages compared to the competing algorithm *MUCAHA*. In particular, *MUCARS* can select a solution on average 60 times faster than *MUCAHA* with an overall solution robustness of 92%, while maintaining a high solution quality comparable to *MUCAHA*. Finally, the evaluation reveals that up to five *alternative solutions* can be selected within a few seconds with a high solution quality. Since there is no approach that fulfills all these features at runtime, these contributions show the novelty and efficacy of the approach.

The selection of a fast and robust solution with a nevertheless high solution quality can be obtained by dividing the process into regions (i.e., region-based approach). In particular, we promote the feature solution robustness by clearly delimiting the respective region boundaries with the help of *world states,* which is different from existing approaches in literature where regions are delimited by services (e.g., cf. Lin et al., 2010; Zhai et al., 2009). To cope with the problem of spatial–temporal coordination and the corresponding mapping of the context, the consideration of world states within a region is indispensable in context-aware service systems.

Furthermore, we have significantly extended the concept of regional expansion from the literature (e.g., cf. Gao et al., 2018; Lin et al., 2010; Zhai et al., 2009) by introducing *feasibility checks* and a *state space measure*. The feasibility checks enable to directly examine regions within the state space in which feasible solutions exist. Thus, unnecessary and time-consuming expansions as well as re-selections can be avoided (in contrast, the execution of the feasibility check in the basic setup only needs 5 milliseconds on average). This becomes also evident in a more detailed analysis of our results, which shows that through the feasibility checks across all settings 1,711 region-based expansions can be avoided, since in each of the 1,711 regions a feasible re-selection can be correctly ruled out. This accelerates the selection process and thus leads to a higher performance. In case all feasible solutions can be ruled out in a region due to user-based and context-based dependencies, we introduce a state space measure for the region-based expansion. This measure enables a careful selection of the neighboring region by analyzing the available context information (i.e., mapped by world states) across all users. A deeper analysis shows that better solutions can be achieved when the region with the larger number of world states (i.e., larger search space) is selected thus supporting the feature solution quality. Additionally, fewer expansions are required. In particular, we examined all runs of the basic setting for which an expansion was required. In 100% of these cases, a solution was found directly after the first expansion when the region with the larger number of world states was used. In contrast, expanding the region with the fewer number of world states found a solution in only 25% of these cases, which would lead to further time-intensive expansions. The evaluation results show that in a total of 97% of all our evaluation settings (cf. Table 6) a solution could be selected after the feasible region has been directly examined by *feasibility checks* and a maximum of only one additional regional expansion has been performed with the help of our *state space measure* (cf. Section 5.3.2), which demonstrates the high relevance of the regions selection.

The evaluation results are also favored by the proposed re-structuring of the regions state space. In the literature, the use of states is a common concept to represent, for example, context interdependencies (Heinrich & Schön, 2015). However, the creation of an entire and feasible state space requires a lot of computation time. Our approach expands the concept of states by dynamic and targeted *extension* of the state space based on runtime conditions, thus eliminating the need to create an entire new state space. In this respect, an analysis of our basic setting shows that the *BASE_MUCAHA* needs on average 210 seconds to build an entire state space

[1] Robustness with only one focus task is obviously not determinable.

and the *MUCARS* needs on average only 13.6 seconds for the dynamic extension per event, and thus the computation time can be reduced by 93.5%. On the other hand, we do not perform the selection on the entire state space like other approaches (Bortlik et al., 2018; Heinrich & Mayer, 2018) as this is very time-consuming. Therefore, the state space is narrowed down to the feasible states (based on the valid context information) before the selection is performed. Therefore, a selection of the basic setting (cf. Section 6.1) is not performed on over 156,000 states but only on about 4,000 states, which further improves the performance with simultaneously high solution quality. Further, in contrast to existing service selection approaches from the literature, our approach allows the determination of alternative solutions. By carefully reusing already performed tasks (e.g., Focus task-based definition of regions) from the initial solution selected at runtime, the computation time can be significantly reduced while maintaining a high solution quality. Analyzing our evaluation results in detail shows, that the originally defined region of the initial solution selected at runtime could be reused in 86% for each problem setting towards an increasing number of events (cf. Section 6.2.4) to find a feasible alternative solution. Thus, a further time-intensive expansion to further regions can be dispensed with in these cases.

Our approach also has implications for science and practice. Starting with *implications for science*, service systems (cf. Maglio & Spohrer, 2008) with their characteristics and their complexity regarding user-based dependencies, context-based dependencies and event-based disruptions lead to novel, real decision problems. These problems should not only be discussed conceptually in the scientific literature (Fakhfakh et al., 2020; Edvardsson et al., 2011), but for which concrete solution techniques should also be developed. In this regard, collaboration and contextualization are already an important part of service-dominant design which forms the basis for modern service systems (Alter, 2012; Böhmann et al., 2014; Faieq et al., 2021; Maleki et al., 2018; Yuan & Hsu, 2017). In particular, contextualization comprises information that characterizes the actual state of an environment (e.g., cf. Romero et al., 2020) and thus affects the design of service systems due to resulting uncertainties (Alter, 2017). These uncertainties are already considered in the design of service systems. Therefore, Alter (2017) describes a (context-aware) service system through proposed axioms as a service system that is affected by direct or indirect interactions with the environment in which it operates (i.e., Axiom 4) and further states that the success of a service system depends on responding appropriately to the diversity of situations that the service system will encounter (i.e., Axiom 18). For this reason, *adaptive context-aware service systems* were discussed in the literature (e.g., cf. Bucchiarone et al., 2012; Faieq et al., 2021; Frey et al., 2017). One of the main task of these systems is *managing adaptations based on the existing context information* (Frey et al., 2017). Managing adaptations based on the existing context information has high performance requirements (i.e., real-time, cf. Frey et al., 2017). Existing adaptive context-aware service systems from the literature primarily focus on the requirements of the service provider. In particular, the flexible response of service providers to changes and the corresponding adaptation of business processes (Faieq et al., 2021) in compliance with the service goals pre-established in the Service Level Agreements contracts (Hidri et al., 2019) represent the key role in the design of adaptive context-aware service systems. In our research, we design the system mainly on the user's perspectives (e.g., companies or users who use the services, i.e., service consumers) whereby mechanisms were developed which come close to real-time support. Thus, we substantively extend existing meta-models from the literature (e.g., Hidri et al., 2019) in the sense that user requirements (e.g., high performance) are detailed and included directly in the design of the adapted process. Furthermore, an essential part of managing adaptations is also an *analysis component*. Thus, existing designs of adaptive context-aware service systems firmly anchor this component directly in the meta-model (e.g., cf. Boudaa et al., 2017). Thereby rules are defined, which check whether an adaptation is necessary when events occur. In our approach, we methodically extend and realize this analysis component with *feasibility checks* and a *state space measure* that control the adaptation mechanism at runtime. This procedure opens the research field to develop, for example, further measures to address the existing challenges of user-based dependencies, context-based dependencies and event-based disruptions.

For *practitioners*, our heuristic approach also provides some important benefits in terms of supporting multi user context-aware processes (i.e., maintaining the contextual environment and the collaboration of multiple users in terms of spatial–temporal coordination) where runtime features are elementary. Thereby, the approach can be applied to multi user context-aware processes in different domains relying on the re-selection of services (i.e., service systems) and can have a relevant impact in practice. We will exemplarily discuss this for the tourism domain. Existing mobile applications that support processes such as a city day trip are characterized by certain possibilities for user individualization. For instance, the app *Culture Trip* enables guided tours, the creation of individual plans or the search for sights in the vicinity of the current location. However, in these *tourist apps* or *platforms* (the latter often takes on the role of service integrators, cf. Heinrich et al., 2011) it is currently not possible to handle disruptive events anyway in regard to *businesses* (e.g., restaurants or museums, i.e., service providers) and *users* (i.e., service consumers). The approach proposed in this paper, allows to consider such real-world events combined with a near-time re-selection, i.e., it is possible to continue the planned tourist activity – and potentially also any other planned activities – with a high quality without interruption. By implementing the proposed approach as extension of existing mobile applications in tourism, users can react on disruptive events by re-selecting an alternative service composition, which is robust regarding the initially planned service composition. This can be done by directly using publicly available information on disruptive events (e.g., weather, traffic jam) and proactively inform all affected users and preventively offer alternative solutions. Furthermore, tourist platforms can use the presented approach not only for disruptive events, but also to immediately adapt an existing feasible solution in general. Thus, a re-selection is not only needed to address an external, given disruptive event (e.g., sudden change in weather) but provides the opportunity for proactive adaptation ("initiated events") of the current feasible solution to further improve the user experience. In particular, users can be enabled to actively initiate changes shortly before the start of the tourist activity or during the tourist activity and share them with other participants. Thus, flexible alternatives can be proposed and perceived between users (e.g., based on current conditions, preferences etc.), which greatly increases social interaction and enables the possibility of a spontaneous realization or adaptation of a tourist activity. Moreover, it will be possible for tourist platforms to draw the user's attention based on real-time information ad hoc to service candidates that might appeal to the individual user (or multiple users) on the basis of the preferences or context information (e.g., time, GPS position) and to immediately suggest an alternative solution (i.e., service recommendation). Thus, user experience can be made even more individual. Finally, businesses such as restaurants or museums can also benefit from our approach by providing current data of their business to the tourist platform via an interface. For example, offers for group discounts (i.e., discounts for multiple users) or time-dependent discounts (i.e., e.g., discounts in the evening hours based on context information) can be proposed to users during the realization of a tourist activity in order to proactively enable adaptations of the tour activity. Besides this discussion for the tourism domain, the application of the presented heuristic approach to further activities or domains in the area of multi user context-aware processes is also possible and promising such as logistics or production (Beverungen et al., 2019; Hohmann & Posselt, 2019), healthcare, disaster relief assistance or field work in companies in order to handle disruptions and proactive adaptations within such processes in terms of spatial–temporal coordination.

7 Conclusion, Limitations and Future Research

In this paper we present a heuristic technique for service re-selection, which is aimed at considering multiple users, context-awareness and in particular disruptive events at runtime. In this regard, we propose an approach that can maintain the features performance, solution quality, solution robustness and alternative solutions at runtime in the case of disruptive events. Existing service systems from the literature already refer to the complexity of user-based and context-based dependencies but do not consider disruptive events at runtime. To address this research gap, we developed a heuristic technique, which carefully divides the process into regions and efficiently select a service composition by using feasibility checks and a state space measure. In addition, the concept of a state space is considerably extended by a dynamic as well as precise state space re-structuring at runtime to deal with disruptive events. Alternative solutions are also available to the user in near-time if the selected solution does not meet the users' preferences and expectations. The evaluation results show that *MUCARS* can achieve significant improvements in performance and solution robustness at runtime compared to competing artifacts (i.e., Bortlik et al., 2018), while maintaining a high solution quality.

However, our approach is also subject to some limitations that need to be addressed in future research: *First*, we initially focused on real-world events (e.g., closed restaurant), which leads to a "service failure of a candidate service" in the model. However, mobile environments are volatile and different event types can occur at runtime. Our starting point in this paper could be used to investigate further changes in the model due to real-world events (e.g., changes in non-function properties values such as prices). Here, we are very confident that the presented approach can be extended and transferred to a wide range of further event types as important parts of the proposed heuristic such as the dynamic re-structuring of the state space can be directly re-used. In addition, synergies can arise from the concurrent processing of several different events leading to fewer changes in the process flow for all users. Here, a promising idea may be the use of monitoring and event handling approaches that enable the efficient processing of multiple events at the same time (Ayed et al., 2013; Chen & Rabhi, 2016; Flouris et al., 2017; Kum, 2020; Wang et al., 2017). *Second*, the evaluation shows that with an increasing number of users no proportional increase of the computation time can be achieved. The reason for this is the over-proportional increase in the number of states within the state space required to coordinate and select a solution for multiple participating users. To overcome this limitation, further research in the area of service systems may focus on enhancing the concept of these coordination-relevant states and therefore the mapping of user-based dependencies. Here, a promising idea may be the development of an approach in order to efficiently identify and omit non-relevant states and thus to improve the creation and processing of coordination-relevant states. *Finally*, with *MUCARS* a fast support at runtime is possible and thus the approach enables near-time support (i.e., *MUCARS* can select a solution on average 60 times faster than existing approaches). However, interactive support is becoming increasingly important in mobile environments (Li & Chen, 2019). Since the *MUCARS* runtime is in the lower seconds range (e.g., *MUCARS* requires about 11 seconds in Setting I), real-time interaction is only possible to a limited extent, which possibly influences the user satisfaction (cf. Section 2.1). To improve real-time interaction, a dialogue could be built in that visualizes the user the resulting problem when a disruptive event occurs. The time until the user comprehends this problem and reacts accordingly could then be used to re-select a new solution in the background and present it immediately to the user if preferred. Currently, we work on such concepts of intelligent user interaction with other researchers. In conclusion, the provided multi user context-aware service re-selection approach can serve as promising first step for contributing to the important topic of disruptive events at runtime.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10796-023-10392-8.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data Availability The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Conflict of Interest The authors declare that they have no conflict of interest.

References

Abu-Khzam, F. N., Bazgan, C., Haddad, J. E., & Sikora, F. (2015). On the Complexity of QoS-Aware Service Selection Problem. In A. Barros, D. Grigori, N. C. Narendra, & H. K. Dam (Eds.), *Service-Oriented Computing* (Vol. 9435, pp. 345–352, Lecture Notes in Computer Science). Springer Berlin Heidelberg.

Ai, L., & Tang, M. (2008). QoS-Based Web Service Composition Accommodating Inter-service Dependencies Using Minimal-Conflict Hill-Climbing Repair Genetic Algorithm. In *2008 IEEE Fourth International Conference on eScience (eScience), Indianapolis, USA* (pp. 119–126). IEEE. https://doi.org/10.1109/eScience.2008.110.

Allied Market Research. (2021). Location Based Services Market. https://www.alliedmarketresearch.com/location-based-services-market. Accessed 16 November 2021.

Alrifai, M., & Risse, T. (2009). Combining global optimization with local selection for efficient QoS-aware service composition. In J. Quemada, G. León, Y. Maarek, & W. Nejdl (Eds.), *18th International Conference on World Wide Web, Madrid, Spain, 20.04.2009 - 24.04.2009* (p. 881). ACM Press. https://doi.org/10.1145/1526709.1526828.

Alrifai, M., Risse, T., & Nejdl, W. (2012). A hybrid approach for efficient Web service composition with end-to-end QoS constraints. *ACM Transactions on the Web, 6*, 1–31. https://doi.org/10.1145/2180861.2180864

Alter, S. (2012). Metamodel for service analysis and design based on an operational view of service and service systems. *Service Science, 4*(3), 218–235.

Alter, S. (2017). Service System Axioms that Accept Positive and Negative Outcomes and Impacts of Service Systems. *ICIS 2017 Proceedings 1*. https://aisel.aisnet.org/icis2017/ServiceScience/Presentations/1

Angarita, R., Cardinale, Y., & Rukoz, M. (2013). Dynamic recovery decision during composite web services execution. In L. Ladid, A. Montes, P. A. Bruck, F. Ferri, & R. Chbeir (Eds.), *5th International Conference on Management of Emergent Digital EcoSystems, Luxembourg, Luxembourg, 28.10.2013 - 31.10.2013* (pp. 187–194). ACM Press. https://doi.org/10.1145/2536146.2536152.

Angarita, R., Cardinale, Y., & Rukoz, M. (2014). Reliable composite web services execution: Towards a dynamic recovery decision. *Electronic Notes in Theoretical Computer Science, 302*, 5–28.

Aouatef, C., Iman, B., & Allaoua, C. (2008). Adaptive composition of services in context-aware ambient intelligent systems. In B. C. Desai (Ed.), *the 2014 International C* Conference, Montreal, QC, Canada, 03.08.2014 - 05.08.2014* (pp. 1–5). ACM Press. https://doi.org/10.1145/2641483.2641535.

Ardagna, D., & Pernici, B. (2006). Global and Local QoS Guarantee in Web Service Selection. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, et al. (Eds.), *Business Process Management Workshops* (Vol. 3812, pp. 32–46, Lecture Notes in Computer Science). Springer Berlin Heidelberg.

Ardagna, D., & Pernici, B. (2007). Adaptive Service Composition in Flexible Processes. *IEEE Transactions on Software Engineering, 33*, 369–384. https://doi.org/10.1109/TSE.2007.1011

Australian Government. (2021). Climate statistics for Australian locations: Monthly climate statistics. http://www.bom.gov.au/climate/averages/tables/cw_086071.shtml. Accessed 6 November 2021.

Ayed, D., Asim, M., & Llewellyn-Jones, D. (2013). An event processing approach for threats monitoring of service compositions. In *International Conference on Risks and Security of Internet and Systems (CRiSIS), La Rochelle, France* (pp. 1–10).

Baccari, S., & Neji, M. (2016). Design for a context-Aware and collaborative mobile learning system. *2016 IEEE International Conference on Computational Intelligence and Computing Research, ICCIC 2016*. https://doi.org/10.1109/ICCIC.2016.7919578

Barber, F., & Salido, M. A. (2015). Robustness, stability, recoverability, and reliability in constraint satisfaction problems. *Knowledge and Information Systems, 44*, 719–734. https://doi.org/10.1007/s10115-014-0778-3

Bearzotti, L. A., Salomone, E., & Chiotti, O. J. (2012). An autonomous multi-agent approach to supply chain event management. *International Journal of Production Economics, 135*, 468–478. https://doi.org/10.1016/j.ijpe.2011.08.023

Beverungen, D., Müller, O., Matzner, M., Mendling, J., & Vom Brocke, J. (2019). Conceptualizing smart service systems. *Electronic Markets, 29*, 7–18. https://doi.org/10.1007/s12525-017-0270-5

Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization. *ACM Computing Surveys, 35*, 268–308. https://doi.org/10.1145/937503.937505

Bobek, S., & Nalepa, G. J. (2017). Uncertain context data management in dynamic mobile environments. *Future Generation Computer Systems, 66*, 110–124. https://doi.org/10.1016/j.future.2016.06.007

Böhmann, T., Leimeister, J. M., & Möslein, K. (2014). Service systems engineering. *Business & Information Systems Engineering, 6*(2), 73–79.

Bortlik, M., Heinrich, B., & Mayer, M. (2018). Multi User Context-Aware Service Selection for Mobile Environments. *Business & Information Systems Engineering, 60*, 415–430. https://doi.org/10.1007/s12599-018-0552-2

Boudaa, B., Hammoudi, S., Mebarki, L. A., Bouguessa, A., & Chikh, M. A. (2017). An aspect-oriented model-driven approach for building adaptable context-aware service-based applications. *Science of Computer Programming, 136*, 17–42.

Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences, 237*, 82–117. https://doi.org/10.1016/j.ins.2013.02.041

Bucchiarone, A., Cappiello, C., Di Nitto, E., Pernici, B., & Sandonini, A. (2012). A Variable Context Model for Adaptable Service-Based Applications. *International Journal of Adaptive, Resilient and Autonomic Systems, 3*, 35–53. https://doi.org/10.4018/jaras.2012070103

Buys, J., Florio, V. de, & Blondia, C. (2011). Towards context-aware adaptive fault tolerance in SOA applications. In O. Etzion, A. Gal, S. Zdonik, P. Vincent, & D. Eyers (Eds.), *the 5th ACM international conference, New York, USA, 11.07.2011 - 15.07.2011* (p. 63). ACM Press. https://doi.org/10.1145/2002259.2002271.

Cao, H., Feng, X., Sun, Y., Zhang, Z., & Wu, Q. (2007). A service selection model with multiple QoS constraints on the MMKP. *2007 IFIP International Conference on Network and Parallel Computing Workshops (NPC 2007),* 584–589. https://doi.org/10.1109/NPC.2007.35

Cao, Z., Zhang, X., Zhang, W., Xie, X., Shi, J., & Xu, H. (2015). A context-aware adaptive web service composition framework. *IEEE International Conference on Computational Intelligence & Communication Technology,* 62–66. https://doi.org/10.1109/CICT.2015.68

Caserta, M., & Voß, S. (2019). The robust multiple-choice multidimensional knapsack problem. *Omega, 86*, 16–27. https://doi.org/10.1016/j.omega.2018.06.014

Chen, W., & Rabhi, F. A. (2016). Enabling user-driven rule management in event data analysis. *Information Systems Frontiers, 18*, 511–528. https://doi.org/10.1007/s10796-016-9633-2

Cherif, S., Djemaa, R. B., & Amous, I. (2019). Reflective approach to improve self-adaptation of Web service compositions with autonomic middleware. *International Journal of Pervasive Computing and Communications, 15*, 144–173. https://doi.org/10.1108/IJPCC-01-2019-0001

Corradini, F., Meschini, G., Polzonetti, A., & Riganelli, O. (2007). A rule-driven business process design. *2007 29th International Conference on Information Technology Interfaces,* 401–406. https://doi.org/10.1109/ITI.2007.4283804

David, K., Geihs, K., Leimeister, J. M., Roßnagel, A., Schmidt, L., Stumme, G., et al. (Eds.). (2014). *Socio-technical Design of Ubiquitous Computing Systems*. Springer International Publishing.

Deng, S., Huang, L., Hu, D., Zhao, J. L., & Wu, Z. (2016). Mobility-Enabled Service Selection for Composite Services. *IEEE Transactions on Services Computing, 9*, 394–407. https://doi.org/10.1109/TSC.2014.2365799

Di Napoli, C., Ribino, P., & Serino, L. (2021). Correction to: Customisable assistive plans as dynamic composition of services with normed-QoS. *Journal of Ambient Intelligence and Humanized Computing*. https://doi.org/10.1007/s12652-021-02978-4

Edvardsson, B., Ng, G., Zhi Min, C., Firth, R., & Yi, D. (2011). Does service-dominant design result in a better service system? *Journal of Service Management, 22(4),* 540–556. https://doi.org/10.1108/09564231111155114

Evers, C., Kniewel, R., Geihs, K., & Schmidt, L. (2014). The user in the loop: Enabling user participation for self-adaptive applications. *Future Generation Computer Systems, 34*, 110–123. https://doi.org/10.1016/j.future.2013.12.010

Faieq, S., Front, A., Saidi, R., El Ghazi, H., & Rahmani, M. D. (2019). A context-aware recommendation-based system for service composition in smart environments. *Service Oriented Computing and Applications, 13*, 341–355. https://doi.org/10.1007/s11761-019-00277-7

Faieq, S., Saidi, R., El Ghazi, H., Front, A., & Rahmani, M. D. (2021). Building adaptive context-aware service-based smart systems. *Service Oriented Computing and Applications, 15*, 21–42. https://doi.org/10.1007/s11761-020-00310-0

Fakhfakh, S., Hein, A. M., Jankovic, M., & Chazal, Y. (2020). A meta-model for product service systems of systems. *Proceedings of the Design Society: DESIGN Conference, 1,* 1235–1244. https://doi.org/10.1017/dsd.2020.48

Fekih, H., Mtibaa, S., & Bouamama, S. (2019a). An Efficient User-Centric Web Service Composition Based on Harmony Particle Swarm Optimization. *International Journal of Web Services Research, 16*, 1–21. https://doi.org/10.4018/IJWSR.2019010101

Fekih, H., Mtibaa, S., & Bouamama, S. (2019b). The dynamic reconfiguration approach for fault-tolerance web service composition based on multi-level VCSOP. *Procedia Computer Science, 159*, 1527–1536. https://doi.org/10.1016/j.procs.2019.09.323

Femenia-Serra, F., Neuhofer, B., & Ivars-Baidal, J. A. (2019). Towards a conceptualisation of smart tourists and their role within the smart destination scenario. *The Service Industries Journal, 39*, 109–133. https://doi.org/10.1080/02642069.2018.1508458

Frey, G., Khalgui, M., Romdhani, M., & Fkaier, S. (2017). Context-awareness meta-model for reconfigurable control systems. *(E. Damiani, G. Spanoudakis, & L. Maciaszek, Eds.; pp. 226–234). SCITEPRESS - Science and Technology Publications, Lda.* https://doi.org/10.22028/D291-36618

Flouris, I., Giatrakos, N., Deligiannakis, A., Garofalakis, M., Kamp, M., & Mock, M. (2017). Issues in complex event processing: Status and prospects in the big data era. *Journal of Systems and Software, 127*, 217–236.

Galletta, D. F., Henry, R., McCoy, S., & Polak, P. (2004). Web site delays: How tolerant are users? *Journal of the Association for Information Systems, 5*(1), 1.

Gao, H., Huang, W., Yang, X., Duan, Y., & Yin, Y. (2018). Toward service selection for workflow reconfiguration: An interface-based computing solution. *Future Generation Computer Systems, 87*, 298–311. https://doi.org/10.1016/j.future.2018.04.064

Gavalas, D., Konstantopoulos, C., Mastakas, K., & Pantziou, G. (2014). Mobile recommender systems in tourism. *Journal of Network and Computer Applications, 39*, 319–333.

Gendreau, M., & Potvin, J.-Y. (2005). Metaheuristics in Combinatorial Optimization. *Annals of Operations Research, 140*, 189–213. https://doi.org/10.1007/s10479-005-3971-7

Gendreau, M., & Potvin, J.-Y. (2010). *Handbook of Metaheuristics* (Vol. 146). Springer, US.

Ghallab, M., Nau, D., & Traverso, P. (2008). *Automated planning: Theory and practice*. Elsevier.

Gogna, A., & Tayal, A. (2013). Metaheuristics: Review and application. *Journal of Experimental & Theoretical Artificial Intelligence, 25*, 503–526. https://doi.org/10.1080/0952813X.2013.782347

Grotherr, C., Semmann, M., & Böhmann, T. (2018). Engaging users to co-create-implications for service systems design by evaluating an engagement platform. *Hawaii International Conference on System Sciences (HICSS)*.

Hara, T., Aoyama, K., Kurata, Y., & Yabe, N. (2016). Service design in tourism: encouraging a cooperative relationship between professional design and non-professional design. In S. K. Kwan, J. C. Spohrer, & Y. Sawatani (Eds.), *Global Perspectives on Service Science: Japan* (Vol. 75, pp. 119–135, Service Science: Research and Innovations in the Service Economy). Springer New York.

Harrison, R., Flood, D., & Duce, D. (2013). Usability of mobile applications: Literature review and rationale for a new usability model. *Journal of Interaction Science, 1*, 1. https://doi.org/10.1186/2194-0827-1-1

He, Q., Han, J., Yang, Y., Grundy, J., & Jin, H. (2012). QoS-driven service selection for multi-tenant SaaS. *IEEE Fifth International Conference on Cloud Computing*, 566–573. https://doi.org/10.1109/CLOUD.2012.125

Heinrich, B., Klier, M., Lewerenz, L., & Mayer, M. (2015). *Enhancing decision support in multi user service selection*. 36th International Conference on Information Systems (ICIS).

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. *Journal of Decision Systems, 24*, 406–432. https://doi.org/10.1080/12460125.2015.1080498

Heinrich, B., & Mayer, M. (2018). Service selection in mobile environments: Considering multiple users and context-awareness. *Journal of Decision Systems, 27*, 92–122. https://doi.org/10.1080/12460125.2018.1513223

Heinrich, B., & Schön, D. (2015). Automated Planning of Context-aware Process Models. *European Conference on Information Systems*. https://doi.org/10.18151/7217352

Heinrich, B., Zellner, G., & Leist, S. (2011). Service integrators in business networks—the importance of relationship values. *Electronic Markets, 21*, 215–235. https://doi.org/10.1007/s12525-011-0075-x

Hidri, W., M'tir, R. H., Saoud, N. B. B., & Ghedira-Guegan, C. (2019). A Meta-model for context-aware adaptive Business Process as a Service in collaborative cloud environment. *Procedia Computer Science, 164*, 177–186.

Hohmann, C., & Posselt, T. (2019). Design challenges for CPS-based service systems in industrial production and logistics. *International Journal of Computer Integrated Manufacturing, 32*, 329–339. https://doi.org/10.1080/0951192X.2018.1552795

Hoxmeier, J. A., & DiCesare, C. (2000). System response time and user satisfaction: An experimental study ofbrowser-based applications.. *AMCIS 2000 Proceedings*, 347.

Jin, H., Lin, R., Zou, H., & Shuai, T. (2012). A Novel Method for Optimizing Multi-User Service Selection. *Journal of Convergence Information Technology, 7*, 296–310. https://doi.org/10.4156/jcit.vol7.issue5.36

Khan, S., Li, K. F., Manning, E. G., & Akbar, M. M. (2002). Solving the knapsack problem for adaptive multimedia systems. *Student Information University, 2*(1), 157–178.

Koskela-Huotari, K., Patrício, L., Zhang, J., Karpen, I. O., Sangiorgi, D., Anderson, L., et al. (2021). Service system transformation through service design: Linking analytical dimensions and service design approaches. *Journal of Business Research, 136*, 343–355. https://doi.org/10.1016/j.jbusres.2021.07.034

Kum, D. K. (2020). Extended CEP Model for Effective Enterprise Systems Service Monitoring. *KSII Transactions on Internet and Information Systems (TIIS), 14*(2), 807–825.

Kurdija, A. S., Silic, M., Delac, G., & Vladimir, K. (2019). Fast Multi-Criteria Service Selection for Multi-User Composite Applications. *IEEE Transactions on Services Computing*, 1. https://doi.org/10.1109/TSC.2019.2925614.

Lewerenz, L. (2015). A heuristic technique for an efficient decision support in context-aware service selection. *Proceedings of the 36th International Conference on Information Systems (ICIS)*, 1–20. https://epub.uniregensburg.de/32660/

Li, S., & Chen, C.-H. (2019). The effects of visual feedback designs on long wait time of mobile application user interface. *Interacting with Computers, 31*, 1–12. https://doi.org/10.1093/iwc/iwz001

Lin, K.-J., Zhang, J., Zhai, Y., & Xu, B. (2010). The design and implementation of service process reconfiguration with end-to-end QoS constraints in SOA. *Service Oriented Computing and Applications, 4*(3), 157–168.

Ma, Z., Zhang, X. G., Zhu, Z. Y., & Wang, H. (2015). The implementation of adaptive web service composition. *International Conference on Computer Science and Applications (CSA)*, 303–307. https://doi.org/10.1109/CSA.2015.35

Maglio, P. P., & Spohrer, J. (2008). Fundamentals of service science. *Journal of the Academy of Marketing Science, 36*, 18–20. https://doi.org/10.1007/s11747-007-0058-9

Maleki, E., Belkadi, F., & Bernard, A. (2018). A meta-model for product-service system based on systems engineering approach. *Procedia CIRP, 73*, 39–44.

Mayer, M. (2017). Multi-user service re-selection: react dynamically to events occuring at process Execution. *25th European Conference on Information Systems (ECIS)*, 1807–1821.

Meredith, J. R., Raturi, A., Amoako-Gyampah, K., & Kaplan, B. (1989). Alternative research paradigms in operations. *Journal of Operations Management, 8*, 297–326. https://doi.org/10.1016/0272-6963(89)90033-8

Moghaddam, M., & Davis, J. G. (2014). Service Selection in Web Service Composition: A Comparative Review of Existing Approaches. In A. Bouguettaya, Q. Z. Sheng, & F. Daniel (Eds.), *Web Services Foundations* (Vol. 33, pp. 321–346). Springer New York.

Mostofa Akbar, M., Sohel Rahman, M., Kaykobad, M., Manning, E. G., & Shoja, G. C. (2006). Solving the Multidimensional Multiple-choice Knapsack Problem by constructing convex hulls. *Computers & Operations Research, 33*, 1259–1273. https://doi.org/10.1016/j.cor.2004.09.016

Muhammad, S. S., Dey, B. L., & Weerakkody, V. (2018). Analysis of Factors that Influence Customers' Willingness to Leave Big Data Digital Footprints on Social Media: A Systematic Review of Literature. *Information Systems Frontiers, 20*, 559–576. https://doi.org/10.1007/s10796-017-9802-y

Nagarajan, N., & Thirunavukarasu, R. (2020). Service-oriented Broker for Effective Provisioning of Cloud Services – a Survey. *International Journal of Computing and Digital Systems, 9*, 863–879. https://doi.org/10.12785/ijcds/090508

Neville, K., O'Riordan, S., Pope, A., Rauner, M., Rochford, M., Madden, M., et al. (2016). Towards the development of a decision support system for multi-agency decision-making during cross-border emergencies. *Journal of Decision Systems, 25*(sup1), 381–396.

Oizumi, K., Meguro, Y., Wu, C. X., & Aoyama, K. (2013). Supporting System for Tour Lineup Design based on Tour Service Modeling. In Y. Shimomura & K. Kimita (Eds.), *The Philosopher's Stone for Sustainability* (pp. 143–148). Springer Berlin Heidelberg.

Pang, B., Hao, F., Yang, Y., & Park, D.-S. (2020). An efficient approach for multi-user multi-cloud service composition in human–land sustainable computational systems. *The Journal of Supercomputing, 76*, 5442–5459. https://doi.org/10.1007/s11227-019-03140-w

Prat, N., Comyn-Wattiau, I., & Akoka, J. (2015). A taxonomy of evaluation methods for information systems artifacts. *Journal of Management Information Systems, 32*(3), 229–267.

Public Transport Victoria. (2019). Data and reporting: Monthly performance. https://www.ptv.vic.gov.au/footer/data-and-reporting/network-performance/monthly-performance/. Accessed 6 November 2021.

Rahmani, D., & Ramezanian, R. (2016). A stable reactive approach in dynamic flexible flow shop scheduling with unexpected disruptions: A case study. *Computers & Industrial Engineering, 98*, 360–372. https://doi.org/10.1016/j.cie.2016.06.018

Roda, C., Navarro, E., Zdun, U., López-Jaquero, V., & Simhandl, G. (2018). Past and future of software architectures for context-aware systems: A systematic mapping study. *Journal of Systems and Software, 146*, 310–355. https://doi.org/10.1016/j.jss.2018.09.074

Romero, M., Guédria, W., Panetto, H., & Barafort, B. (2020). Towards a characterisation of smart systems: A systematic literature review. *Computers in Industry, 120*, 103224.

Saleh, A., Ismail, R., & Fabil, N. (2017). Evaluating Usability for Mobile Application. In Unknown (Ed.), *the 2017 International Conference, Hong Kong, Hong Kong, 28.12.2017 - 30.12.2017* (pp. 71–77). ACM Press. https://doi.org/10.1145/3178212.3178232.

Sandionigi, C., Ardagna, D., Cugola, G., & Ghezzi, C. (2013). Optimizing Service Selection and Allocation in Situational Computing Applications. *IEEE Transactions on Services Computing, 6*, 414–428. https://doi.org/10.1109/TSC.2012.18

Sbihi, A. (2007). A best first search exact algorithm for the Multiple-choice Multidimensional Knapsack Problem. *Journal of Combinatorial Optimization, 13*, 337–351. https://doi.org/10.1007/s10878-006-9035-3

Sedighiani, K., Shokrollahi, S., & Shams, F. (2021). BASBA: A framework for Building Adaptable Service-Based Applications. *Journal of Systems and Software, 179*, 110989. https://doi.org/10.1016/j.jss.2021.110989.

Seffah, A., Donyaee, M., Kline, R. B., & Padda, H. K. (2006). Usability measurement and metrics: A consolidated model. *Software Quality Journal, 14*, 159–178. https://doi.org/10.1007/s11219-006-7600-8

Shen, Y., Wang, M., Tang, X., Luo, Y., & Guo, M. (2012). Context-aware HCI service selection. *Mobile Information Systems, 8*(3), 231–254.

Shen, Y.-H., & Yang, X.-H. (2011). A self-optimizing QoS-aware service composition approach in a context sensitive environment. *Journal of Zhejiang University Science C, 12*, 221–238. https://doi.org/10.1631/jzus.C1000031

Sheng, Q. Z., Qiao, X., Vasilakos, A. V., Szabo, C., Bourne, S., & Xu, X. (2014). Web services composition: A decade's overview. *Information Sciences, 280*, 218–238. https://doi.org/10.1016/j.ins.2014.04.054

Statista. (2019a). 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 November 2021.

Statista. (2019b). Size of mobile professional services market by segment worldwide in 2014 and 2018. https://www.statista.com/statistics/501755/worldwide-mobile-professional-services-market-revenue/. Accessed 16 November 2021.

Tan, J., Rönkkö, K., & Gencel, C. (2013). A framework for software usability and user experience measurement in mobile industry. *Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement,* 156–164. https://doi.org/10.1109/IWSM-Mensura.2013.31

Tedeschi, L. O. (2006). Assessment of the adequacy of mathematical models. *Agricultural Systems, 89*, 225–247. https://doi.org/10.1016/j.agsy.2005.11.004

TomTom. (2021). Melbourne traffic: Australia. https://www.tomtom.com/en_gb/traffic-index/melbourne-traffic/. Accessed 6 December 2021.

Tretola, G., & Zimeo, E. (2019). Reactive behavioural adaptation of service compositions. *Journal of Software: Evolution and Process, 31*, 161. https://doi.org/10.1002/smr.2201

Tung, W.-F., Yuan, S.-T., Wu, Y.-C., & Hung, P. (2014). Collaborative service system design for music content creation. *Information Systems Frontiers, 16*, 291–302. https://doi.org/10.1007/s10796-012-9346-0

Ventola, C. L. (2014). Mobile devices and apps for health care professionals: Uses and benefits. *Pharmacy and Therapeutics, 39*(5), 356.

Wanchun, D., Chao, L., Xuyun, Z., & Chen, J. (2011). A QoS-Aware Service Evaluation Method for Co-selecting a Shared Service. In *2011 IEEE International Conference on Web Services (ICWS), Washington, DC, USA* (pp. 145–152). IEEE. https://doi.org/10.1109/ICWS.2011.11.

Wang, H., Peng, S., & Yu, Q. (2019). A parallel refined probabilistic approach for QoS-aware service composition. *Future Generation Computer Systems, 98*, 609–626. https://doi.org/10.1016/j.future.2019.03.053

Wang, H., Zhang, J., Wan, C., Shao, S., Cohen, R., Xu, J., & Li, P. (2010). Web service selection for multiple agents with incomplete preferences. *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 1,* 565–572. https://doi.org/10.1109/WI-IAT.2010.27

Wang, S., Hsu, C.-H., Liang, Z., Sun, Q., & Yang, F. (2014). Multi-user web service selection based on multi-QoS prediction. *Information Systems Frontiers, 16*, 143–152. https://doi.org/10.1007/s10796-013-9455-4

Wang, X., Feng, Z., Huang, K., & Tan, W. (2017). An automatic self-adaptation framework for service-based process based on exception handling. *Concurrency and Computation: Practice and Experience, 29*(5), e3984.

Weinert, T., Janson, A., &#38; Leimeister, J. M. (2020). Does context matter for value co-creation in smart learning services? *SSRN Electronic Journal.* https://doi.org/10.2139/ssrn.3926281

Will, M., Bertrand, J., & Fransoo, J. C. (2002). Operations management research methodologies using quantitative modeling. *International Journal of Operations & Production Management, 22*, 241–264. https://doi.org/10.1108/01443570210414338

Xu, L., & Jennings, B. (2010). A cost-minimizing service composition selection algorithm supporting time-sensitive discounts. *Proceedings - 2010 IEEE 7th International Conference on Services Computing, SCC 2010,* 402–408. https://doi.org/10.1109/SCC.2010.76

Yu, H. Q., & Reiff-Marganiec, S. (2009). A backwards composition context based service selection approach for service composition.

IEEE International Conference on Services Computing, 419–426. https://doi.org/10.1109/SCC.2009.25

Yu, T., & Lin, K.-J. (2007). Service Selection Algorithms for Composing Complex Services with Multiple QoS Constraints. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, et al. (Eds.), *Service-Oriented Computing – ICSOC 2007* (Vol. 4749, pp. 130–143, Lecture Notes in Computer Science). Springer Berlin Heidelberg.

Yuan, S.-T.D., & Hsu, S.-T. (2017). Enhancing service system design: An entity interaction pattern approach. *Information Systems Frontiers, 19*, 481–507. https://doi.org/10.1007/s10796-015-9604-z

Zhai, Y., Zhang, J., & Lin, K.-J. (2009). SOA middleware support for service process reconfiguration with end-to-end QoS constraints. *2009 IEEE International Conference on Web Services,* 815–822. https://doi.org/10.1109/ICWS.2009.126

Zhang, D., Adipat, B., & Mowafi, Y. (2009). User-centered context-aware mobile applications-The next generation of personal mobile computing. *Communications of the Association for Information Systems, 24*(1), 3.

Zhang, J., Lee, J., & Lin, K.-J. (2012a). Context-aware proactive process reconfiguration in service-oriented architecture. *IEEE 14th International Conference on Commerce and Enterprise Computing,* 62–69. https://doi.org/10.1109/CEC.2012.19

Zhang, J., Zhou, A., Sun, Q., Wang, S., & Yang, F. (2018). Overview on Fault Tolerance Strategies of Composite Service in Service Computing. *Wireless Communications and Mobile Computing, 2018*, 1–8. https://doi.org/10.1155/2018/9787503

Zhang, M., Liu, C., Yu, J., Zhu, Z., & Zhang, B. (2013a). A correlation context-aware approach for composite service selection. *Concurrency and Computation: Practice and Experience, 25*, 1909–1927. https://doi.org/10.1002/cpe.2988

Zhang, M., Ranjan, R., Haller, A., Georgakopoulos, D., & Strazdins, P. (2012). Investigating decision support techniques for automating Cloud service selection. *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings,* 759–764. https://doi.org/10.1109/CloudCom.2012.6427501

Zhang, Z., Zheng, S., Li, W., Tan, Y., Wu, Z., & Tan, W. (2013). Leveraging genetic algorithm to compose web services in a context-aware environment. *IEEE International Conference on Systems, Man, and Cybernetics,* 829–834. https://doi.org/10.1109/SMC.2013.147

Zheng, Z., & Lyu, M. R. (2009). A QoS-aware fault tolerant middleware for dependable service composition. *IEEE/IFIP International Conference on Dependable Systems & Networks,* 239–248. https://doi.org/10.1109/DSN.2009.5270332

Zheng, Z., Zhang, Y., & Lyu, M. R. (2014). Investigating QoS of Real-World Web Services. *IEEE Transactions on Services Computing, 7*, 32–39. https://doi.org/10.1109/TSC.2012.34

Zhu, W., Yin, B., Gong, S., & Cai, K.-Y. (2017). An Approach to Web Services Selection for Multiple Users. *IEEE Access, 5*, 15093–15104. https://doi.org/10.1109/ACCESS.2017.2722228

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

Michael Bortlik is a Research Assistant at the Department of Information Systems at the University of Regensburg, Germany. In his research, he focuses on Service Selection. Michael Bortlik published his work in journals such as Business & Information Systems Engineering.

Bernd Heinrich is a Distinguished Professor of Information Systems at the Department of Information Systems at the University of Regensburg, Germany. His research interest focuses on Data and Information Quality, Process Management and Process Planning, Context-aware Services and Mobile Applications. Bernd Heinrich published his research in numerous books and journals (e.g., Business & Information Systems Engineering, Decision Support Systems, Electronic Markets, Information Systems Frontiers, INFORMS Service Science and the Journal of Decision Systems) and presented his work at leading international conferences such as the International Conference on Information Systems (ICIS).

Daniel Lohninger is a Research Assistant at the Department of Information Systems at the University of Regensburg, Germany. In his research, he focuses on Service Selection and Recommender Systems. Daniel Lohninger published his work in journals such as Electronic Markets and Information Systems Frontiers.