# Dynamic Service Selection with End-to-End Constrained Uncertain QoS Attributes

Rene Ramacher and Lars Mönch

Chair of Enterprise-wide Software Systems,
Univerity of Hagen, 58084 Hagen, Germany
{Rene.Ramacher,Lars.Moench}@FernUni-Hagen.de

**Abstract.** Services and service compositions are executed in an uncertain environment with regard to several aspects of quality. Static service selection approaches that determine the entire service selection prior to the execution of a service composition are extensively discussed in the literature. Nevertheless, the uncertainty of quality aspects has been not well addressed in the service selection phase so far, leading to time-consuming and expensive reconfiguration of service compositions at their execution time. Due to the uncertain and dynamic nature of the execution environment, a dynamic service selection approach is highly desirable. In a dynamic service selection, the services are selected during the execution of a service composition taking into account the conditions caused by already executed services. A dynamic service selection contributes to implement robust service compositions to support reliable business processes, where robustness is measured in terms of fulfilling quality constraints of a service composition. In this paper, we examine a dynamic service selection approach based on a Markov decision process. The service selection is considered from a cost minimizing point of view with an end-to-end constrained execution time. A simulation study demonstrates that the dynamic service selection outperforms an optimal static service selection in an uncertain environment with respect to robustness of the service compositions and cost minimizing.

**Keywords:** Dynamic Service Selection, Uncertain QoS, Markov Decision Process.

## 1 Introduction

Complex business functionality is implemented by the composition of already existing services in Service-oriented Architectures (SOAs) [7]. Process models are used to formulate the business or application logic of a service composition. A task within a process model represents certain functionality associated with functional requirements. These requirements have to be provided by a service to execute the corresponding service composition. Therefore, for each task a certain service has to be selected from a set of appropriate services which is known as the service selection problem (SSP). Taken into account the principle of loose coupling, the service selection for each task can be postponed until its execution,

C. Liu et al. (Eds.): ICSOC 2012, LNCS 7636, pp. 237–251, 2012.

i.e., late binding. The quality of a service (QoS) and its cost are considered to distinguish between services providing an identical or a similar functionality [10]. A service selection aims for optimizing certain QoS attributes of a service composition while other QoS attributes are end-to-end constrained [13,14].

Service compositions are executed in an uncertain environment with regard to several QoS aspects. For example, the response time of a service depends on the utilization of the service itself and the underlying network. Because the value of an uncertain QoS attribute is unknown until the service is executed, estimated values are taken into account when a service selection is determined. The actual QoS value will often deviate from the estimated one. The deviation of a QoS value may cause a violation of end-to-end constraints or the deterioration of QoS attributes to be optimized. When end-to-end constraints are exceeded, the execution of the entire service compositions fails.

Most of the approaches for the SSP discussed in the literature deal with a static service selection [4,11,14]. The services for all tasks are selected prior to the execution of the composition in a static service selection. Those approaches are not able to react on unforeseen events like the deviation of QoS values during the execution of a service composition. An apparent approach to deal with deviations is to adjust the service selection for unexecuted tasks in a reconfiguration step taking into account the conditions caused by already executed services [1]. Because the reconfiguration of a service composition considering the entire set of services leads to a high computational and communication effort [5], several efficient reconfiguration approaches based on heuristics are proposed in the literature [5,6]. Due to the uncertainty of several QoS attributes, the execution of a service composition is dynamic and hard to predict. Therefore, a dynamic service selection that takes into account the uncertainty inherent to estimated QoS attributes is appropriate. In contrast to a static service selection, in a dynamic service selection the services are determined individually for each task considering the conditions caused by formerly executed tasks. Although the service selection is carried out locally for each task using an appropriate decision model, a dynamic service selection is able to account for end-to-end constrained QoS attributes.

In this paper, we propose a dynamic service selection approach for sequential service compositions. The SSP is modeled as a Markov Decision Process (MDP) that is able to capture the uncertainty of the QoS attributes. Solving the MDP, we obtain a service selection strategy that is used to drive the dynamic service selection at the execution time. We study the proposed dynamic service selection model for a cost minimizing service selection considering an end-to-end constrained execution time. Its performance is compared to an optimal static service selection approach with reconfiguration by means of randomly generated problem instances. We show that the dynamic service selection ensures robust service compositions fullfiling their end-to-end constraints with lower costs compared to an optimal static service selection approach when an uncertain environment is assumed.

The rest of the paper is organized as follows. The service composition model is formalized in Section 2. The MDP formalism is briefly summarized in Section 3. Furthermore, it is described how the dynamic service selection is modeled as an MDP. The design of the computational experiments and the results are presented in Section 4. Related work is discussed in Section 5. Finally, Section 6 concludes the paper summarizing the major findings and motivating future research.

## 2 Service Composition Model

The process or business logic, i.e., control and data flow, of a service composition is formulated using a process model. We consider abstract process models based on tasks. A task $t_i$ represents a certain functionality with associated functional requirements. An approach to formulate and execute abstract process models is proposed in [2]. The set of services, a service class, that fulfills the functional requirements of a task $t_i$ is denoted with $S_i$. A service from $S_i$ offered by a provider $j$ is denoted as $s_{ij} \in S_i$. Each task $t_i$ has to be bound to a service $s_{ij}$ to execute the corresponding service composition. The binding of the service $s_{ij}$ to the task $t_i$ is indicated by $t_i \leftarrow s_{ij}$. For abbreviation, we denote the service $s_{ij}$ for which $t_i \leftarrow s_{ij}$ holds with $s_i$. In this paper, we consider sequential process models with $n$ tasks. The predecessor of the task $t_{i+1}$ is the task $t_i$, $i = 1, \ldots, n-1$. The task $t_{i+1}$ can only be started after the processing of $t_i$ is completed, i.e., the service $s_{i+1}$ is invoked after the service $s_i$ has been terminated.

The cost of a service and its response time are considered as non-functional attributes. The cost of a service $s_{ij}$, assumed to be known in advance, is denoted with $c(s_{ij})$. In contrast to the cost, the response time of a service $s_{ij}$ can only be estimated prior to the execution of $s_{ij}$. We distinguish between the actual response time $p(s_{ij})$, the expected response time $E[p(s_{ij})]$, and the variance of the response time $Var[p(s_{ij})]$ of a service $s_{ij}$. The quantities $E[p(s_{ij})]$ and $Var[p(s_{ij})]$ can be determined using a monitoring system for gathering execution-related information.

We consider the total cost $c(sc)$ and the total execution time $p(sc)$ for a given service composition $sc$. The cost of a service composition is the sum of costs $c(s_i)$ of the services $s_i$ bound to the tasks:

$$c(sc) = \sum_{i=1}^{n} c(s_i). \tag{1}$$

Since we consider sequential service compositions, the total execution time $p(sc)$ is the sum of the response times of the services bound to the tasks:

$$p(sc) = \sum_{i=1}^{n} p(s_i). \tag{2}$$

The total execution time of a service composition is restricted to a predefined value $R > 0$. The execution of a service composition is feasible if $p(sc) \leq R$ holds. Otherwise, the execution of $sc$ is failed. When penalty costs for failed

service compositions are known, e.g. due to an Service Level Agreement, we denote them by $c^p$.

## 3 Dynamic Service Selection

### 3.1 Markov Decision Processes

An MDP can be used to model decision processes containing uncertainty. Formally, an MDP is represented as a four-tuple $(\Gamma, A, T, R)$ where $\Gamma$ is the set of states the decision process consists of and $A$ is the set of possible actions. The mapping $T : \Gamma \times A \times \Gamma \to [0, 1]$ provides for $\gamma_1, \gamma_2 \in \Gamma$ and $a \in A$ the probability $T(\gamma_1, a, \gamma_2)$ that the process moves to state $\gamma_2$ when action $a$ is performed in state $\gamma_1$. The mapping $R : \Gamma \times A \times \Gamma \to \mathbb{R}$ is the reward function. The value of $R(\gamma_1, a, \gamma_2)$ is for $\gamma_1, \gamma_2 \in \Gamma$ and $a \in A$ the expected utility when action $a$ is performed in state $\gamma_1$ and the process moves to state $\gamma_2$. Solving an MDP results in a strategy $\pi : \Gamma \to A$ that maps each state $\gamma \in \Gamma$ to an appropriate action $a \in A$ such that the expected total reward of the entire process is maximized.

A dynamic service selection based on an MDP requires an appropriate modeling of states and actions as well as determining the transition probabilities and choosing an appropriate reward function. For the researched problem, states are used to represent the task currently to be processed and a time interval in which the processing of a task will be started. The processing of a task $t_i$ is modeled as an action. The state model with the corresponding actions, the state transitions probability, and the cost function as negative reward function are formulated in Subsection 3.2. Based on the proposed state model, Subsection 3.3 describes the probability model used to define the state transition function.

### 3.2 State Model

The execution of a service composition requires the selection of a service $s_{ij} \in S_i$ that is invoked to execute the task $t_i$. With regard to a constrained execution time, the service selection for a certain task has to be driven by the time already consumed by previously executed services. The difference of the consumed time and the execution time restriction $R$ determines the time that is still available to execute the remaining tasks. For this reason, we propose a state model that captures the already consumed time. A set of states $\Gamma_i$ is used for each single task $t_i$. The state $\gamma_{ik} \in \Gamma_i$ is associated with a time interval $[l_{ik}, u_{ik}]$. The processing of task $t_i$ cannot start before $l_{ik}$ and not later than $u_{ik}$ in state $\gamma_{ik}$.

We distinguish $m_i$ states with disjunct time intervals for each task $t_i$, $i = 2, \ldots, n$. The task $t_1$ is represented by the single state $\gamma_{11}$ with $l_{11} = 0$ and $u_{11} = 0$. Moreover, two final states $\gamma_{n+1,1}$ and $\gamma_{n+1,2}$ are introduced. The interval boundaries of these states are set to $l_{n+1,1} = 0$ and $u_{n+1,1} = R$ as well as $l_{n+1,2} = R$ and $u_{n+1,2} = \infty$, respectively. The complete state model consists of $\sum_{i=1}^{n+1} m_i + 3$ states. The set of all states is $\Gamma = \bigcup_{i=1}^{n+1} \Gamma_i$.

The interval boundaries of the states in $\Gamma_i$ with $i = 2, \ldots, n$ are determined taking into account the earliest time ($\underline{p_i}$) and the latest time ($\overline{p_i}$) at which the processing of task $t_i$ can be started. These quantities are calculate as:

$$\underline{p_i} = \sum_{a=1}^{i-1} \min\{E[p(s_{aj})] | s_{aj} \in S_a\}. \tag{3}$$

$$\overline{p_i} = \sum_{a=1}^{i-1} \max\{E[p(s_{aj})] | s_{aj} \in S_a\}. \tag{4}$$

Taking into account $\underline{p_i}$ and $\overline{p_i}$, the interval boundaries $l_{ik}$ and $u_{ik}$ of each state $\gamma_{ik} \in \Gamma_i$ are determined as:

- state $\gamma_{i1}$: $l_{i,1} = 0$ and $u_{i1} = \underline{p_i}$
- state $\gamma_{ik}$, $k = 2, \ldots, m_i - 1$: $l_{ik} = \underline{p_i} + (k-2)\Delta$ and $u_{ik} = \underline{p_i} + (k-1)\Delta$, where $\Delta := \left(\overline{p_i} - \underline{p_i}\right)/(m_i - 2)$
- state $\gamma_{i,m_i}$: $l_{i,m_i} = \overline{p_i}$ and $u_{i,m_i} = \infty$.

The quantities $m_i$ are parameters of the approach. Note that because of our choice of $\Delta$ the model requires $m_i > 2$.

The set of actions $A_i$ that can be performed in each state $\gamma_{ik} \in \Gamma_i$ is derived from the service class $S_i$. Note that each $\gamma_{ik} \in \Gamma_i$ represents the task $t_i$. An action $a_{ij} \in A_i$ corresponds to the invocation of the service $s_{ij} \in S_i$. The final states in $\Gamma_{n+1}$ are associated with the empty action set $A_{n+1} = \emptyset$. The set of all actions is $A = \bigcup_{i=1}^{n+1} A_i$.

The state transitions of the MDP reflect the sequential processing of the tasks of a service composition. The selection of an action $a_{ij} \in A_i$ in state $\gamma_{ik} \in \Gamma_i$ corresponds to the invocation of the service $s_{ij} \in S_i$ to process the task $t_i$. The invocation of the service $s_{ij}$ terminates with a certain probability in the time interval of each state $\gamma_{i+1,y} \in \Gamma_{i+1}$ in which either the processing of the task $t_{i+1}$ can be started for $i < n$ or the processing of the service composition is completed for $i = n$. Accordingly, the decision process moves from $\gamma_{ik}$ to each state $\gamma_{i+1,y} \in \Gamma_{i+1}$ with a certain probability when $a_{ij}$ is selected for $\gamma_{ik}$. The probability $T(\gamma_{ik}, a_{ij}, \gamma_{i+1,y})$ of moving to $\gamma_{i+1,y} \in \Gamma_{i+1}$ depends on the response time of service $s_{ij}$ that is associated with $a_{ij}$ and the time interval represented by $\gamma_{ik}$. Details on calculating the probability $T(\gamma_{ik}, a_{ij}, \gamma_{zy})$ are discussed in Subsection 3.3.

A state model for a simple service composition consisting of three tasks and $m_1 = m_2 = 3$ is shown in Figure 1. For each state except of $\gamma_{11}$, the associated time interval is plotted on the horizontal axis. The state $\gamma_{11}$ is associated with the degenerated interval 0. For the sake of simplicity, some state transitions are only indicated by a dashed line.

The cost function $C$ maps each state transition to the cost $c(s_{ij})$ resulting from the invocation of the service associated with the action $a_{ij}$ and a penalty value $p$ that is applied when the decision process moves to an undesired state. We obtain:

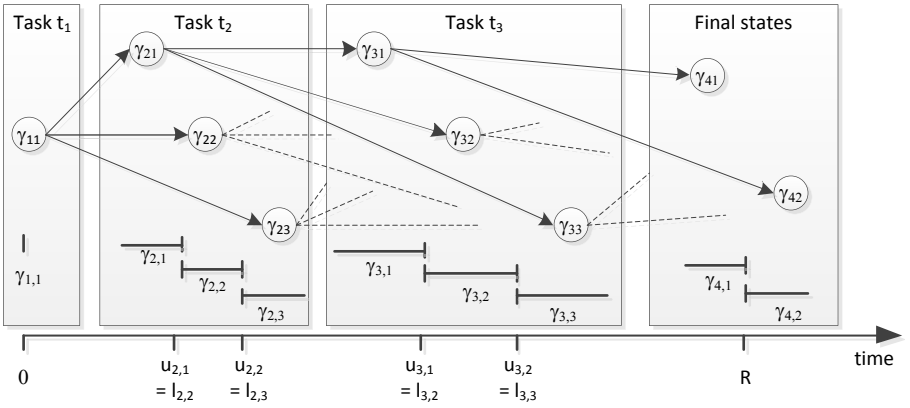


**Fig. 1.** State model of a service composition consisting of three tasks and $m_1 = m_2 = 3$

$$C(\gamma_{ik}, a_{ij}, \gamma_{zy}) = c(s_{ij}) + p. \tag{5}$$

The decision process contains only the single undesired state $\gamma_{n+1,2}$. This is the state the process is moved to when the execution time restriction $R$ is exceeded. Therefore, $p$ is 0 for all states $\gamma_{zy}$ with $z \neq n+1$ and $y \neq 2$. When penalty costs are available (cf. Section 2), then $p$ is set to $c^p$. Otherwise, the value of $p$ is calculated dependening on a risk factor $rf$ as:

$$p = rf \sum_{i=1}^{n} \left( \max\{c(s_{ij}) | s_{ij} \in S_i\} - \min\{c(s_{ij}) | s_{ij} \in S_i\} \right). \tag{6}$$

Modeling the dynamic service selection as an MDP requires that the Markov condition is fulfilled. It states that the optimality of an action does not depend on the sequence of actions already performed. Based on the proposed state model, the Markov condition is fulfilled because the consequences of already performed actions are summarized in the time interval of a certain state. Therefore, the optimality of an action in a certain state only depends on the state and the expected costs of the states the process is supposed to move to.

### 3.3 State Transition Probability Model

The quantity $T(\gamma_{ik}, a_{ij}, \gamma_{zy})$ is the probability that the process moves to $\gamma_{zy}$ when the action $a_{ij}$ is performed in $\gamma_{ik}$. Because of the sequential processing of the service composition, $T(\gamma_{ik}, a_{ij}, \gamma_{zy})$ is 0 for all $\gamma_{zy} \in \Gamma_z$ with $z \neq i+1$. To determine $T(\gamma_{ik}, a_{ij}, \gamma_{i+1,y})$, we assume that the processing of task $t_i$ starts at some point $\beta_{ik} \in [l_{ik}, u_{ik}]$. Thus, the service $s_{ij}$ that is associated with the action $a_{ij}$ is invoked at time $\beta_{ik}$. $T(\gamma_{ik}, a_{ij}, \gamma_{i+1,y})$ is the probability that the invocation of $s_{ij}$ terminates at some point in the time interval $[u_{i+1,y}, l_{i+1,y}]$ represented by $\gamma_{i+1,y}$.

We assume that $\beta_{ik}$ is uniformely distributed on the interval $[l_{ik}, u_{ik}]$, i.e. $\beta_{ik} \sim U[l_{ik}, u_{ik}]$. The assumption is reasonable when the intervals $[l_{ik}, u_{ik}]$ are small which is ensured by an appropriate selection of $m_i$. The response time of a service $s_{ij}$ is modeled as a random variable $\rho_{ij}$. We assume that the distribution information of $\rho_{ij}$ is available either in closed form or in form of an empirical distribution function. The later one can be derived using information from a monitoring system. When the service $s_{ij}$ is invoked in the interval $[l_{ik}, u_{ik}]$ then the time at which the invocation of $s_{ij}$ terminates is a random variable $\Psi_{ijk}$, with $\Psi_{ijk} = \beta_{ik} + \rho_{ij}$. Accordingly, $T(\gamma_{ik}, a_{ij}, \gamma_{i+1,y})$ is the probability that $\Psi_{ijk} \geq l_{i+1,y}$ and $\Psi_{ijk} \leq u_{i+1,y}$ holds. We obtain:

$$T(\gamma_{ik}, a_{ij}, \gamma_{i+1,y}) = (1 - F(l_{i+1,y}))F(u_{i+1,y}), \tag{7}$$

where $F(x) = P(\Psi_{ijk} \leq x)$ is the probability distribution of $\Psi_{ijk}$. The distribution $F$ can be calculated as a convolution of $\beta_{ik}$ and $\rho_{ij}$ because the two random variables are independent. When the convolution of $\beta_{ik}$ and $\rho_{ij}$ is not available in closed form, then $F$ has to be numerically evaluated by sampling $\beta_{ik} + \rho_{ij}$.

In this paper, we assume that the response time of a service $s_{ij}$ follows a normal distribution with $\mu_{ij} = E[p(s_{ij})]$ and $\sigma_{ij}^2 = Var[p(s_{ij})]$, i.e., we have $\rho_{ij} \sim N(\mu_{ij}, \sigma_{ij}^2)$. A closed form expression can be derived for the density function of $\Psi_{ijk}$ in this specific situation. Figure 2 shows the density function of $\Psi_{ijk}$ for an action $a_{ij}$ that is performed in state $\gamma_{ik}$. The integral of the shaded area (b) represents the state transition probability $T(\gamma_{ik}, a_{ij}, \gamma_{i+1,y})$. The probability density of the equal distribution is depicted as the shaded area (a).

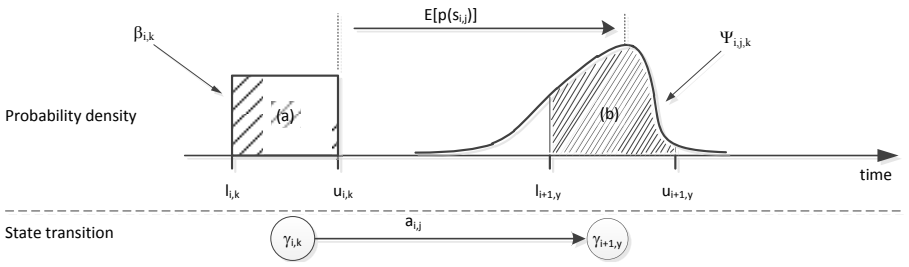


**Fig. 2.** State transition probability $T(\gamma_{ik}, a_{ij}, \gamma_{i+1,y})$ when $\rho_{ij} \sim N(\mu_{ij}, \sigma_{ij}^2)$

### 3.4 Algorithm

A strategy $\pi : \Gamma \rightarrow A$ determines the action $\pi(\gamma) \in A$ that is chosen in the state $\gamma \in \Gamma$. An optimal solution of the dynamic SSP is a strategy $\pi$ that minimizes the expected cost of a service composition. Algorithms to solve an MDP are based on the expected costs $V(\gamma)$ for each state $\gamma \in \Gamma$. With regard to the dynamic service selection, the expected cost $V(\gamma_{ik})$ for a state $\gamma_{ik} \in \Gamma_i$ represents the immediate cost associated with $\gamma_{ik}$, i.e. invocation of a service and penalty costs if an undesired state is reached, and the cost $V(\gamma_{zy})$ of the states $\gamma_{zy}$ the process

is expected to move to. For $z < n$ the cost of each $\gamma_{zy}$ consists of the sum of costs resulting from the invocation of services to process the remaining tasks $t_z, \ldots, t_n$ and the penalty costs that occur when an undesired state is reached. Therefore, the expected cost of a state $\gamma \in \Gamma$ is calculated recursively as:

$$V(\gamma) = \min_{a \in A} \sum_{\gamma' \in \Gamma} T(\gamma, a, \gamma') \left\{ C(\gamma, a, \gamma') + \lambda V(\gamma') \right\}, \tag{8}$$

where $C(\gamma, a, \gamma')$ represents the immediate cost when action $a$ is performed in $\gamma$ and the process is moved to $\gamma'$ (cf. Subsection 3.2). The sum of the immediate costs and the expected costs of a subsequent state $\gamma'$ is weighted with the probability $T(\gamma, a, \gamma')$ that the process is moved to $\gamma'$. Note that the cost of a service associated with the action $a$ is considered in total because the probabilities $T(\gamma, a, \gamma')$ sum up to 1. The penalty cost included in $C(\gamma, a, \gamma')$ when an undesired state $\gamma'$ is reached is considered only as a fraction depending on the probability that the process is moved to $\gamma'$. The equation (8) is known as the Bellman update. The parameter $\lambda$ in expression (8) represents the impact of the costs expected for future states on the costs of earlier states. A standard algorithm to calculate the expected costs for each state $\gamma \in \Gamma$ is given by the backward value iteration. Here, the Bellman update is applied iteratively where convergence of the scheme is ensured by choosing $0 \leq \lambda < 1$.

The dynamic SSP is formulated as a finite MDP with non-stationary expected costs and due to the sequential processing, each strategy $\pi$ is guaranteed to be a proper strategy, i.e. it ends in a finite state of the MDP. Therefore, it turns out that the special case of additive costs, i.e. $\lambda = 1$, is approved without losing the convergence property of the backward value iteration algorithm [8]. Hence, based on the expected cost for each state, the action $a \in A$ chosen in a state $\gamma \in \Gamma$ is determined as follows:

$$\pi(\gamma) = \arg\min_{a \in A} \sum_{\gamma' \in \Gamma} T(\gamma, a, \gamma') \left\{ C(\gamma, a, \gamma') + V(\gamma') \right\}. \tag{9}$$

## 4 Performance Study

A simulation study is conducted to evaluate the performance of the dynamic service selection. The MDP approach described in Section 3 is compared with an optimal static service selection approach with reconfiguration. The average cost of a service composition and the number of failed service compositions are considered as performance indicators. Furthermore, we examine the computational effort required to evaluate the service selection strategy at the execution time of a service composition. Randomly generated problem instances are used. The experimental design is described in Subsection 4.1. Subsection 4.2 elaborates on the optimal static service selection approach with reconfiguration. The results of the computational experiments are presented and discussed in Subsection 4.3.

### 4.1 Generation of Problem Instances

A problem instance consists of $n$ tasks. A set of $m$ services is generated for each task. The services are distinguished into the *gold*, *silver*, and *bronze* categories with respect to their prices and response times. These categories mimic the assumption that a service of a higher price is also expected to have a higher quality, i.e. a lower response time.

The price of a service depends on the categorie the service belongs to and the factor $m_{class}$ that represents the ratio between the price and the quality of the services in different categories. We consider $c^{ref}_{silver} = 100r$, $c^{ref}_{gold} = m_{class} \cdot c^{ref}_{silver}$, and $p^{ref}_{bronze} = 1/m_{class} \cdot p^{ref}_{silver}$ as reference prices for each service category, where $r$ is a realization of a random variable $X \sim U[0, 1]$. Moreover, we consider $p^{ref}_{gold} = 10$, $p^{ref}_{silver} = 50$, and $p^{ref}_{bronze} = 90$ as reference response times for the different service categories.

The price and the expected response time of a service $s_{ij}$ that belongs to the service category $cat \in \{gold, silver, bronze\}$ is chosen as:

$$c(s_{ij}) = c^{ref}_{cat} + (r - 0.5)c^{ref}_{cat}, \tag{10}$$

and

$$E[p(s_{ij})] = p^{ref}_{cat} + (r - 0.5)p^{ref}_{cat}, \tag{11}$$

where $r$ is a realization of an random variable $X \sim U[0, 1]$. The variance of the response time is selected depending on the factor $m_\sigma$ as follows:

$$Var[p(s_{ij})] = m_\sigma E[p(s_{ij})]. \tag{12}$$

Because our comparison approach likewise the majority of existing reconfiguration approaches does not consider penalty costs, we exclude the evaluation of the effects caused by penalty costs from the experimentation. Instead, we evaluate the influence of the risk factor $rf$ (see equation (6)) on the performance of the dynamic service selection. The factors and their levels are summarized in Table 1.

**Table 1.** Factors used to generated problem instances

| Factor | Level | Count |
|---|---|---|
| Number of tasks | $n \in \{5, 10, 15, 20\}$ | 4 |
| Number of providers per task | $m \in \{5, 10, 15\}$ | 3 |
| Expected price difference | $m_{class} \in \{1, 3, 5, 6\}$ | 4 |
| Variance in response time | $m_\sigma \in \{0.1, 0.15, 0.2, 0.25\}$ | 4 |
| Risk factor | $rf \in \{0.3, 0.5, 0.7\}$ | 3 |

### 4.2 Comparison Approach

We use a Mixed Inter Programming (MIP)-based service selection with reconfiguration as an optimal service selection approach. The approach is based on the MIP shown in (13) - (16). The MIP determines a cost minimizing service selection taking into account an end-to-end constrained execution time. In addition to the price $c(s_{ij})$ of a service $s_{ij}$ and its expected processing time $E[p(s_{ij})]$, the MIP contains two parameters $\alpha$ and $\beta$. These parameters are required to use the MIP in a reconfiguration setting. The quantities $\alpha$ and $\beta$ represent the current progress of a service composition. The parameter $\alpha = 1, \ldots, n-1$ identifies the task that is to be processed next. The value of $\beta$ represents the time, passed since the start of processing the service composition.

The MIP contains the binary decision variables $z_{ij}$. The value of $z_{ij}$ is 1, if the service $s_{ij}$ is used to execute the task $t_i$, otherwise 0. The MIP model is formulated as follows:

$$\min \sum_{i=\alpha}^{n} \sum_{j=1}^{m} c(s_{ij})\, z_{ij} \tag{13}$$

subject to:

$$\sum_{j=1}^{m} z_{ij} = 1, \forall i : i = \{\alpha, \ldots, n\} \tag{14}$$

$$\sum_{i=\alpha}^{n} E[p(s_{ij})]\, z_{ij} \leq R - \beta \tag{15}$$

$$z_{ij} \in \{0, 1\}, \forall i, j : i = \{1, \ldots, n\}, j = \{1, \ldots, m\}. \tag{16}$$

The objective function (13) minimizes the costs resulting from the services used to execute the tasks. The equations (14) force that exactly one service $s_{ij}$ is selected for each task $t_i$. The inequality (15) ensures that the expected amount of time to process the remaining tasks $t_k$, $k = \alpha, \ldots, n$ is smaller than the difference of the execution time restriction $R$ the time $\beta$ already passed. The constraints (16) make sure that the decision variables are binary.

Instead of using the reconfiguration in a heuristic manner, i.e. only by exceeding a threshold, the reconfiguration is invoked after the processing of each task.

### 4.3 Simulation Results

The execution of a service composition is simulated using the discrete-event simulation engine SLX. The response time of each service $s_{ij}$ follows a normal distribution $N(E[p(s_{ij})], Var[p(s_{ij})])$. The MIP is implemented and solved by ILOG OPL 6.3 and CPLEX solver 12.3, respectively.

The influence of the number of states used in the MDP approach is evaluated. Therefore, each problem instance is solved with $m_i = 10, 20$, and 40 states for

each $i = 2, \ldots, n+1$, respectively. The corresponding approaches are denoted by $MDP(10)$, $MDP(20)$, and $MDP(40)$.

For each approach $A \in \{MIP, MDP(10), MPD(20), MDP(40)\}$, 100 invocations are simulated for each problem instance. The simulation records the average cost $c^A$ per successfully executed service composition and the percentage of requests $v^A$ that are successfully executed for each approach $A$. Moreover, the simulation keeps tracks of the average computation time $ct^A$, required by the approach $A$ to determine the service selection of all tasks at the execution time of the service composition.

Figure 3 shows an evaluation of the MIP and the MDP(40) approach with respect to the uncertainty expressed through the factor $m_\sigma$. Different values of the risk factor $rf$ are considered for MDP(40). The average cost per service composition is summarized in Figure 3 a). In general, the results state that the MIP is outperformed by the MDP approaches with respect to costs. The advantage of the MDP approaches increases with increasing uncertainty. The lowest average cost are observed for $MDP(40)$ with $rf = 0.3$ independently of $m_\sigma$. The average costs observed for the MDP approaches increase with an increasing risk factor.

Results with repect to $v$ are shown in Figure 3 b). The data reveals that the MIP leads to a large fraction of failed service compositions, i.e., a low value of $v$. The value of $v$ decreases from 87 percent for $m_\sigma = 0.1$ to 81 percent for $m_\sigma = 0.25$. For $m_\sigma = 0.1$ and $m_\sigma = 0.15$, a value close to 100 percent of successfully executed service compositions is observed for all MDP approaches. For the remaining values of $m_\sigma$, it can be concluded that a higher risk factor leads to a higher value of $v$. The execution time restriction is fulfilled in 99 percent for $rf = 0.7$.

An evaluation of $c^A$ with respect to different values of $m_{class}$ is shown in Figure 4 a). The data reveals that in the case of homogenous costs, i.e. $m_{class} = 1$, a similar performance is observed for all approaches. The MIP leads to the minimal average cost per service composition for $m_{class} = 1$. However, the costs determined by the MDP approaches are lower than the cost of the MIP for the remaining values of $m_{class}$. The results point out that a larger number of states

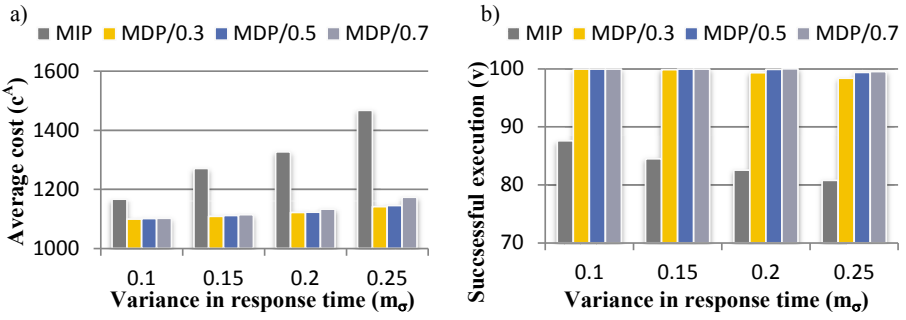


**Fig. 3.** a) Average cost depending on $m_\sigma$ for MIP and MDP(40) with different $rf$ values b) percentage of succesfully executed service compositions depending on $m_\sigma$.

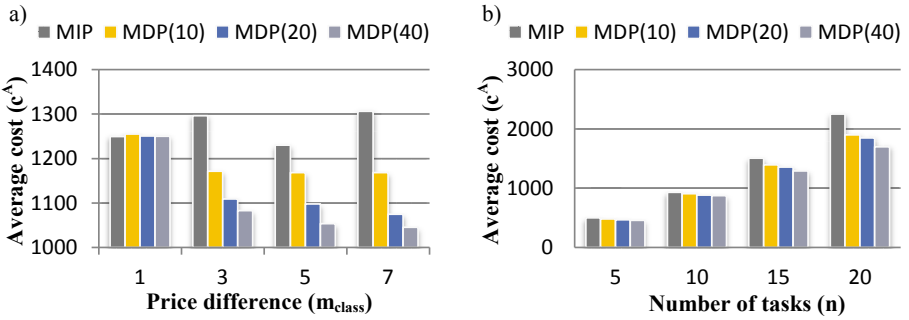


**Fig. 4.** Average cost per service composition depending on $m_{class}$ (a) and on $n$ (b)

improves the performance of the MDP. The lowest values for $c^A$ are obtained for $MDP(40)$. Results of a sensitive analysis with respect to the number of tasks are shown in Figure 4 b). The performance gap of the MIP compared to the MDP increases with an increasing number of tasks.

Figure 5 presents the results of an sensitive analysis of $v$ for different values of $m$ and $n$. It can be observed that $v$ is close to 100 percent for all MDP approaches, independently from $n$ and $m$. The values of $v$ observed for the MIP point out that an increasing number of tasks as well as an increasing number of providers lead to a higher fraction of failed service compositions. Taken into account the number of tasks, the fraction of failed service compositions increases from 9 percent for $n = 5$ tasks up to more than 30 percent for $n = 20$ tasks.

The computation time $ct^A$ required to evaluate the service selection strategy varies from 0.4 seconds for the MDP(40) approach on average for $n = 15$ to less than 0.03 seconds on average for the MDP(10) approach for $n = 5$.

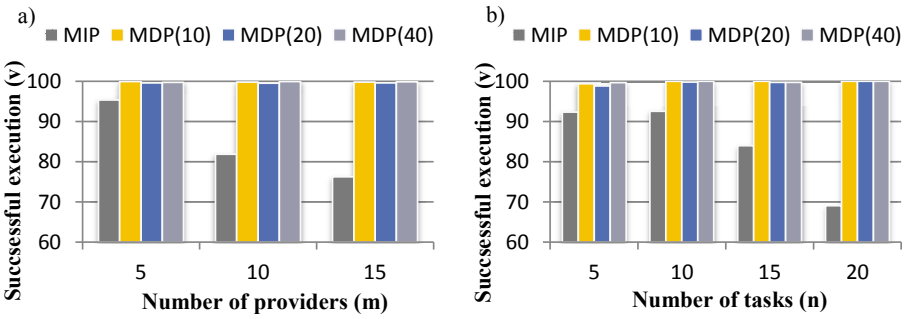


**Fig. 5.** Percentage of succesful executions for different $m$ (a) and $n$ (b)

## 5 Related Work

The service selection with uncertain QoS values has been tackled by researchers in the recent years. The majority of the existing work is devoted to the reconfiguration of service compositions. A reconfiguration is required when services

fail [12] or the QoS values deviate from the original specification [1,5,6]. Service breakdowns are addressed by Yu and Lin [12]. Two algorithms are proposed to reconfigure the service compositions which are already in execution and to provide a backup service selection that is used by future requests of a service composition, respectively. Both algorithms are offline approaches, i.e., the service reconfiguration as well as the backup path are determined simultaneously with the initial service selection. Both approaches are restricted to a single service failure. A failure of several services can neither be handled by the reconfiguration nor by the backup service selection. The violation of end-to-end constrained QoS values due to the deviation of QoS values is not considered at all.

Canfora et al. [1] consider the reconfiguration of end-to-end constrained service compositions with uncertain QoS attributes. An approach for an online reconfiguration of service compositions is presented. Reconfiguration regions are identified taking into account conditional branches, parallelizing operators, and loops. A reconfiguration region consists of all unexecuted tasks that will be reselected to avoid the violation of the end-to-end constraints. The reselection is performed by a common static service selection approach e.g. an Integer Programming-based [14] or Genetic Algorithm-based [4] service selection.

Another reconfiguration approach is discussed by Lin, Zhang, and Zhai [6]. The efficiency of the reconfiguration is addressed. A heuristic is proposed that restricts the size of the reconfiguration regions using a distance measure [6]. The distance is increased iteratively until a reconfiguration region is identified that is of a sufficient size to successful reconfigure the service composition. In addition to restricting the size of a reconfiguration region, Li et al. [5] propose a restriction of the number of tasks considered by the reconfiguration. Supplement services are identified during the initial service selection. The reconfiguration of each task is then narrowed to the set of supplement services to improve the efficiency of the reconfiguration.

Each of the reconfiguration approaches discussed so far has to be regarded as a static service selection. Gao et al. [3] propose an approach for a dynamic service selection based on MDP. The uncertainty with respect to the availability of services is taken into account. The service selection aims for the optimization of a single QoS attribute, e.g. the total cost of a service composition or their execution time.

A further MDP-based dynamic service selection approach for the adaption of processes with inter-process dependencies is discussed by Verma et al [9]. The process execution is affected by exogenous factors. The corresponding implications require an adaption of the service selection at the execution time. The combination of services is restricted by the inter-process dependencies i.e. the combination of services used for different tasks has to be coordinated. As exogenous factor they consider a delayed delivery that requires a decision to wait for the delayed delivery or to place a new order at a different service provider.

End-to-end constrained QoS attributes are neither considered in [3] nor in [9]. Therefore, both approaches are not suitable for a dynamic service selection with an end-to-end constrained execution time. Although delays are considered in [9], the approach is only based on the information that a delay occurs but the

delay itself is not quantified. However, the quantification of the actual delay is mandatory with regard to an end-to-end constrained processing time to drive the service selection for the tasks to be executed.

## 6 Conclusion

Services and service compositions are executed in an uncertain environment with respect to several aspects of quality. The uncertainty is not addressed sufficiently by existing static service selection approaches leading to expensive and time-consuming reconfigurations at the execution time of a service composition. Anticipating deviations resulting from estimated QoS values improve the overall quality of service compositions and their robustness, measured in terms of violated end-to-end constraints.

A dynamic service selection approach modeled as an MDP is proposed in this paper. Modeling the service selection decision as an MDP allows for explicating the uncertainty inherent to several QoS attributes within the decision model. The performance of the proposed approach is studied for a cost minimizing service selection with an end-to-end constrained execution time. An extensive simulation study reveals that in an uncertain environment the dynamic service selection outperforms a static service selection approach with reconfiguration with respect to robustness of the service compositions and to costs. Furthermore, it is shown that the approach is efficient with respect to the computational burden resulting from the service selection decision at the execution time of a service composition.

There are several directions for future research. Although the proposed approach allows for taking into account arbitrary distributed response times, the experiments conducted in this paper are based on the assumption that the response times of the serivces are normally distributed. The normal dsitribution assumption for response times might be violated in a real-world environment. Therefore, further experiments should be performed to assess the performance of the proposed dynamic service selection approach when arbitrary distributions are considered. It should also be evaluated whether the normal distribution assumption represents a sufficient approximation to deal with arbitrary distributed response times or not.

Furthermore, the decision model proposed in this paper is based on sequential process models. Although the sequential process model is a fundamental service composition pattern and serves as a basis for many other service composition patterns with regard to an QoS optimizing service selection [11], a further direction for future research is to extend the model proposed in this paper to more complex process models including conditional branches and parallelizing operators.

## References

1. Canfora, G., Di Penta, M., Esposito, R., Villani, M.L.: QoS-aware replanning of composite web services. In: Proceedings of the IEEE International Conference on Web Services, pp. 121–129 (2005)

2. Casati, F., Ilnicki, S., Jin, L., Krishnamoorthy, V., Shan, M.-C.: Adaptive and Dynamic Service Composition in eFlow. In: Wangler, B., Bergman, L.D. (eds.) CAiSE 2000. LNCS, vol. 1789, pp. 13–31. Springer, Heidelberg (2000)
3. Gao, A., Yang, D., Tang, S., Zhang, M.: Web Service Composition Using Markov Decision Processes. In: Fan, W., Wu, Z., Yang, J. (eds.) WAIM 2005. LNCS, vol. 3739, pp. 308–319. Springer, Heidelberg (2005)
4. Jaeger, M.C., Mühl, G.: QoS-based selection of services: The implementation of a genetic algorithm. In: Proceedings of the KiVS Workshop 2007: Service-Oriented Architectures und Service Oriented Computing (SOA/SOC), pp. 359–370. VDE (2007)
5. Li, J., Ma, D., Mei, X., Sun, H., Zheng, Z.: Adaptive QoS-aware service process reconfiguration. In: Proceedings of the 2011 IEEE International Conference on Services Computing, SCC 2011, pp. 282–289. IEEE Computer Society, Washington, DC (2011)
6. Lin, K.-J., Zhang, J., Zhai, Y.: An efficient approach for service process reconfiguration in SOA with end-to-end QoS constraints. In: Proceedings of the 11th International Conference on Commerce and Enterprise Computing, pp. 146–153 (2009)
7. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: Service-oriented computing: a research roadmap. International Journal of Cooperative Information Systems 17(02), 223–255 (2008)
8. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach, 3rd edn. Prentice-Hall, Inc., Upper Saddle River (2009)
9. Verma, K., Doshi, P., Gomadam, K., Miller, J., Sheth, A.: Optimal adaptation in web processes with coordination constraints. In: Proceedings of the IEEE International Conference on Web Services, pp. 257–264 (2006)
10. Wang, S.G., Sun, Q.B., Yang, F.C.: Towards web service selection based on QoS estimation. International Journal of Web Grid Services 6(4), 424–443 (2010)
11. Yu, T., Lin, K.-J.: Service selection algorithms for web services with end-to-end QoS constraints. In: Proceedings of the IEEE International Conference on E-Commerce Technology, pp. 129–136. IEEE Computer Society (2004)
12. Yu, T., Lin, K.-J.: Adaptive algorithms for finding replacement services in autonomic distributed business processes. In: Proceedings of the International Symposium on Autonomous Decentralized Systems, pp. 427–434 (2005)
13. Yu, T., Zhang, Y., Lin, K.-J.: Efficient algorithms for web services selection with end-to-end QoS constraints. ACM Transactions on the Web (TWeb) 1(1) (2007)
14. Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J., Sheng, Q.Z.: Quality driven web services composition. In: Proceedings of the 12th International Conference on World Wide Web, pp. 411–421. ACM (2003)