

# 基于 DEC\_POMDP 的 Web 服务组合优化算法

赵 莉,李蜀瑜

(陕西师范大学 计算机科学学院,陕西 西安 710062)

**摘 要:**面向服务计算(SOC)和面向服务架构(SOA)技术共同推动了 Web 服务及其组合技术的发展。网络环境的动态变化及其对 Web 服务质量(QoS)的影响,给服务成功组合带来挑战,为服务组合效果满足用户需求带来难题。为了得到经济、省时且成功率高的服务组合策略,综合考虑网络环境的动态变化、服务质量的可变性、用户需求的多样性,采用分散的部分可观测马尔可夫决策(DEC\_POMDP)模型描述多个服务 Agent 的自组织服务组合系统,在基本 Q 学习算法基础上做出改进,求解模型得到组合策略。实验结果表明求解的策略较大地提高了组合服务的成本、时间消耗,且组合成功率较高。采用的 DEC\_POMDP 模型有效地将 Web 服务组合动态过程描述出来,并自适应地更新了 QoS 值,采用 Q 学习算法及时使用了最新的 QoS 值。

**关键词:**服务质量;DEC\_POMDP;自组织;服务组合;Q 学习

**中图分类号:**TP301.6

**文献标识码:**A

**文章编号:**1673-629X(2014)03-0074-05

doi:10.3969/j.issn.1673-629X.2014.03.019

## Web Service Composition Optimization Algorithm Based on DEC\_POMDP

ZHAO Li, LI Shu-yu

(School of Computer Science, Shaanxi Normal University, Xi'an 710062, China)

**Abstract:**Service-Oriented Computing (SOC) and Service-Oriented Architecture (SOA) promoted Web service and its composition of technology development. Changes in the network environment and its impact on Quality of Service (QoS), brought challenge to the successful combination and problems to the service composition effect to meet customer needs. In order to get economic, time-saving and high success-rate service composition strategy, considering the changes in the network environment, variability of QoS and the diversity of user needs, use DEC\_POMDP model to describe the service Agent combination of self-organizing systems, making an improvement based on Q learning algorithm to solve composition of the model. Experimental results show that solving strategy largely improves the services composition cost, time-consuming and the composition success rate is higher. The DEC\_POMDP model describes the dynamic process of Web service composition and adaptively updates the QoS values, the use of Q-learning algorithm makes it in a timely manner using the latest QoS values.

**Key words:**quality of services; DEC\_POMDP; self\_organized; service composition; Q\_learning

## 0 引言

Web 服务采用了开放的、松耦合架构,是一个组件模型,它将应用程序的不同功能单元(服务)通过这些服务之间定义良好的接口和规则联系起来,从而实现特定的服务功能。如何从众多的候选服务中快速的选择出满足用户要求的可靠服务并进行组合已经成为服务计算领域中面临的一个重要问题<sup>[1]</sup>。在选择的过程中要考虑网络环境及可用 Web 服务的 QoS 属性<sup>[2]</sup>,也要考多个满足特定需求的 Web 服务价格和质量优化问题,还要考虑到全局约束和用户的偏好设置<sup>[3]</sup>。

近年来,针对该问题,相关学者展开了大量的研究,孔维梁等介绍了基于工作流、基于 AI 规划及其他组合优化策略,并对动态 Web 服务组合算法进行了研究<sup>[4]</sup>;万里平等提出了一种基于 QoS 的在线 Web 服务全局优化算法,算法优化考虑了用户需求容错机制及网络环境、服务质量的可变性<sup>[5]</sup>;蔡美玲等利用多目标遗传算法,生成了多选择 Web 服务组合的多个选择方案<sup>[6]</sup>;王飞明等引用产生式规则和接口语义规则对 Web 服务建模,引入了 QoS 属性,提出了一种全局优化的分层动态服务组合算法<sup>[7]</sup>。这些服务组合过程,

收稿日期:2013-05-13

修回日期:2013-08-15

网络出版时间:2014-01-07

**基金项目:**国家自然科学基金资助项目(41271387);中央高校基本科研业务费项目(GK201002011);西安市科技局项目(SF1228-3)

**作者简介:**赵 莉(1985-),女,安徽淮北人,硕士研究生,研究方向为 Web 服务;李蜀瑜,博士,副教授,研究方向为嵌入式系统、云计算。

**网络出版地址:**http://www.cnki.net/kcms/detail/61.1450.TP.20140107.1726.051.html

引入了全局的 QoS 属性信息,但是因为没有及时更新组合后系统的 QoS 属性值,所以导致 QoS 值滞后许多,为了能够自适应地获取组合后系统 QoS 值,并且及时对更新后的 QoS 加以利用,文中分析并对比了 POMDP 模型和 DEC\_POMDP 模型,最后决定采用分散的部分可观测马尔可夫决策过程描述组合过程。

在自组织的 Web 服务组合过程中,由于网络环境的制约,参与组合的服务 agent 不能得到整个网络的完全观测。对于单个服务 agent,无论是对单个服务的 QoS 感知,还是对组合后整体服务 QoS 的预知,结果部分的客观测序都是具有部分可观测性的。比较文献[8]中的 POMDP 模型和文献[9-10]提出的 DEC\_POMDP 模型,选择文献[9-10]中的分布式随机控制系统来作为解决组合流程的依据。POMDP 模型适用于集中控制环境中组合系统的描述,但是对于文中的分散服务组合过程,选择采用 DEC\_POMDP 模型。文中基于 DEC\_POMDP 模型,引入 agent 概念<sup>[11-16]</sup>,建立多 Web 服务 agent 组合决策模型,该模型对各服务 agent 的组合策略进行参数编码,并基于相邻服务 agent 交互合作中基于策略梯度估计,通过强化学习<sup>[17-20]</sup>来逼近 DEC\_POMDP 的最优组合,最终获得可接受的组合策略。

## 1 Web 服务组合 QoS 系统模型

文中对组合后的复合服务系统进行建模,每个状态表示组合成功后的一个复合服务状态。在 DEC\_POMDP 模型中,服务 agent 根据顺序(seq)、选择(sel)和循环(loop)三种结构进行组合,最后产生不同组合的全局 QoS ( $ws$ ) 属性。文中借鉴前辈们研究结论重点考虑的属性值包括复合服务的服务费用(Cost)、执行时间(Time)、可用性(Availability)、成功率(Success\_Rate)、信誉度(Reputation),并依据该属性对组合策略进行评价。对公式(1),公式(2)……中的 seq 表示顺序执行组合,sel 表示并行执行组合,而 loop 表示循环执行的组合, $a_i$  则表示第  $i$  个 Web 服务执行与否的动作结果。

(1) 服务费用 (Cost): 指调用服务事务中所需支付的相关费用,包含两部分:请求者的相关费用和提供商的相关费用。

$$C(ws) =$$

$$\begin{cases} \sum_{i=1}^n \text{Cost}(a_i), \text{seq}(a_1, a_2, \dots, a_n) \\ \sum_{i=1}^n T_i \times \text{obs}_i \times \text{Cost}(a_i), \text{sel}(a_1, a_2, \dots, a_n) \\ \sum_{i=1}^n m \times T_i \times \text{obs}_i \times \text{Cost}(a_i), \text{loop}(a_1, a_2, \dots, a_n) \end{cases} \quad (1)$$

(2) 执行时间 (Time): 指服务请求提交到执行结束需要的时间,执行时间的快慢会影响吞吐量的大小。

$$\text{Time}(ws) =$$

$$\begin{cases} \sum_{i=1}^n \text{Time}(a_i), \text{seq}(a_1, a_2, \dots, a_n) \\ \max_{i=1}^n \text{Time}(a_i), \text{sel}(a_1, a_2, \dots, a_n) \\ m \times \sum_{i=1}^n T_i \times \text{obs}_i \times \text{Time}(a_i), \text{loop}(a_1, a_2, \dots, a_n) \end{cases} \quad (2)$$

(3) 可用性 (Availability): 服务在一段时间内,可以被调用的概率。

$$\text{Availability}(ws) =$$

$$\begin{cases} \prod_{i=1}^n \text{Availability}(a_i), \text{seq}(a_1, a_2, \dots, a_n) \\ \prod_{i=1}^n T_i \times \text{obs}_i \times \text{Availability}(a_i), \text{sel}(a_1, a_2, \dots, a_n) \\ m \times \prod_{i=1}^n T_i \times \text{obs}_i \times \text{Availability}(a_i), \text{loop}(a_1, a_2, \dots, a_n) \end{cases} \quad (3)$$

(4) 成功率 (Success\_Rate): 服务执行能成功完成的概率,组合后总成功率由选取服务的成功率构成。

$$\text{Success\_Rate} =$$

$$\begin{cases} \prod_{i=1}^n \text{Success\_Rate}(a_i), \text{seq}(a_1, a_2, \dots, a_n) \\ \sum_{i=1}^n T_i \times \text{obs}_i \times \text{Success\_Rate}(a_i), \text{sel}(a_1, a_2, \dots, a_n) \\ m \times \prod_{i=1}^n T_i \times \text{obs}_i \times \text{Success\_Rate}(a_i), \text{loop}(a_1, a_2, \dots, a_n) \end{cases} \quad (4)$$

(5) 信誉度 (Reputation): 服务供应商对服务可用性的描述。

$$\text{Reputation}(ws) =$$

$$\begin{cases} \min_{i=1}^n \text{Reputation}(a_i), \text{seq}(a_1, a_2, \dots, a_n) \\ \min_{i=1}^n T_i \times \text{obs}_i \times \text{Reputation}(a_i), \text{sel}(a_1, a_2, \dots, a_n) \\ m \times \min_{i=1}^n T_i \times \text{obs}_i \times \text{Reputation}(a_i), \text{loop}(a_1, a_2, \dots, a_n) \end{cases} \quad (5)$$

评价 QoS 由  $C, T, A, S, R$  五个部分组成,其评价价值分为收益型  $qv_g$  和成本型  $qv_c$ ,分别如公式(6)、公式(7)。

收益型  $qv_g$ :

$$qv_{ij} =$$

$$\begin{cases} \frac{Q(ws)_{ij}^{\max} - Q(ws)_j^{\min}}{Q(ws)_j^{\max} - Q(ws)_j^{\min}}, Q(ws)_j^{\max} = Q(ws)_j^{\min} \\ 1, Q(ws)_j^{\max} \neq Q(ws)_j^{\min} \end{cases} \quad (6)$$

成本型  $qv_c$ :

$$qv_{ij} = \begin{cases} \frac{1}{Q(ws)_{ij}} - \frac{1}{Q(ws)_j^{\max}}, & Q(ws)_j^{\max} = Q(ws)_j^{\min} \\ \frac{1}{Q(ws)_j^{\min}} - \frac{1}{Q(ws)_j^{\max}}, & Q(ws)_j^{\max} \neq Q(ws)_j^{\min} \end{cases} \quad (7)$$

针对收益型 QoS, 则  $qv_g$  越大越好, 成本型 QoS,  $qv_c$  越小越好。

## 2 Q 学习算法求解 Web 服务 agent 组合的 DEC\_POMDP 问题

### 2.1 服务 agent 组合的 DEC\_POMDP 模型

假设注册中心有  $n$  个 Web 服务存在, 这  $n$  个 Web 服务可以分为  $m$  类抽象服务。每类抽象服务由一个服务 agent 描述, 多服务 agent 系统采用 DEC\_POMDP 建模, 每个 agent 代表注册中心的一类抽象服务, 每个 agent 有各自的信念、动作、目标、计划和事件。复合服务系统服务由若干 Web 服务组合而成, 假设服务使用与历史状态无关, 即服从马尔可夫过程。

服务组合的过程就是在  $n$  个服务中用 agent 选择  $m$  个服务参与形成一个组合策略。在学习过程中, 每个 agent 会产生其自身的知识库并动态更新, 此时 agent 系统分析用户要求和当前复合服务 QoS 状态作对比, 并调整、制定其后的服务组合目标。下面给出模型中各参数:

状态集  $S$  代表系统状态, 它由构成服务组合策略的  $n$  个 agent 状态组成, 每个 agent 在时刻  $t$  有 0, 1 两种状态, 其中 0 状态表示该 agent 不参与组合或者不符合用户需求, 1 状态表示该 agent 符合用户需求且参与组合。

动作集  $A$ , 在决策时刻  $t$ , 系统状态为  $S$ , 待参与组合的服务节点的集合为  $N$ ,  $m$  为参与组合的服务个数, 其中所有服务节点中候选服务总数为  $n$ , 下一个要参与组合的服务只与当前服务有关, 而与之之前参与组合的服务无关动作值为  $a$ 。

观察值集合  $\Omega_i$ ;  $o_i$  代表 agent  $i$  的观察值, 包含服务 agent 1 -  $i$  组合后的 QoS 属性、后续服务的功能属性、其 QoS 属性, 以及对网络环境的动态观察, 用于建模组合系统中 QoS 值的动态变化情况, 保障系统自组织性, 保障系统可以自适应地更新 QoS。

状态迁移概率  $T: S \times \vec{A} \times S \rightarrow [0, 1]$  表示组合中多服务 agent 系统的转移函数, 代表在组合过程中采取组合动作  $\vec{a}$  后转移到新状态  $s'$  的概率, 一般是根据统计规律, 此处将服务可用性做归一化处理后作为状态转

移概率, 状态迁移概率决定了在当前状况下系统状态转移到另一状态的概率, 概率越大转移到该状态的几率越大。

初始时采用贪婪算法, 选择 QoS 属性好的组合动作。

观察值概率 Obs:  $S \times \vec{A} \times \vec{\Omega} \rightarrow [0, 1]$  表示多服务 agent 系统的观察函数。Obs( $\vec{o} | s', \vec{a}$ ) 表示采取组合动作  $\vec{a}$  后转移到新状态  $s'$  获得观察  $\vec{o}$  的概率, 观察值概率的大小决定了系统当前状态下周围环境对状态转移的影响大小, 它和状态迁移概率共同决定了当前系统状态下迁移到另一状态的几率大小。

收益函数  $S \times \vec{A} \rightarrow Z$  表示组合过程的收益函数。 $R(s, \vec{a})$  表示多 agent 系统在状态  $s$  下采取组合行动  $\vec{a}$  后的组合过程所获利益。

### 2.2 Q 学习算法求解服务组合的 DEC\_POMDP 模型

DEC\_POMDP 模型虽然对组合后的系统做出建模, 但是它的求解却是一个难题。许多研究都致力于求解 DEC\_POMDP 模型, 并取得了一定成果。文中为了求解由 DEC\_POMDP 建模的 Web 服务组合过程, 找到 Q 学习算法。Q 学习采用马尔可夫决策过程 MDP 建模, 采用迭代方法逼近最优解, 以环境-动作的奖赏值  $Q^*(s, a)$  作为度量的参数值。 $Q^*(s, a)$  是 agent 在状态  $s$  下采用动作  $a$  所获得的最大折扣奖赏之和。Q 问题模型包含服务 agent、状态  $S$  和每个状态的动作集合  $A$ 。

将参与组合的第一个服务的 QoS 属性作为算法输入, 组合完成后的复合服务 QoS 属性作为算法输出。对提出的多服务 agent 系统的 DEC\_POMDP 模型, 采用 Q 学习算法来求解。设定 Q 学习中状态集  $S: \{1, 2, \dots, n\}$ ,  $s \in S$ , 动作集  $A: \{0, 1\}$ ,  $a \in A$ 。与 DEC\_POMDP 模型中相似,  $a = 0, Q(s, 0) \geq Q(s, 1)$  不选取此服务,  $a = 1$ , others 选择服务, 执行组合过程。

文中规定 Q 学习算法采用以下规则获得收益值  $r$ :

$$r(s_i, a_i) = \begin{cases} QoS(s) - QoS(ws), & a_i = 1 \\ 0, & a_i = 0 \end{cases} \quad (8)$$

在状态迁移过程中累积代价为:

$$R(s_i, a_i) = \frac{1}{i} \sum_{i=1}^m QoS(s_i) - QoS(ws_i) \quad (9)$$

$R(s_i, a_i)$  为系统从决策步骤  $t_0$  到  $t_m$  时刻, 所获得的累积代价,  $\alpha = \frac{1}{i+1}$  为学习率。那么,  $Q(s_{it}, a_{it}) \leftarrow (1 - \alpha)Q(s_{i(i-1)}, a_{i(i-1)}) + \alpha[R(s_{i(i-1)}, a_{i(i-1)}) + \gamma V_{i-1}(s_{i-1})]$ , 所设计算法步骤如下:

1) 初始化, 令  $t = 1, i = 1, \alpha = 0.5$ , 参与组合的服务

- 数:  $n, Q$  的初值  $Q=0$ ;
- 2) 令  $i=1$ , 随机选定初态  $s_0$  (非终止态), 令  $R(s_0, a_0) = 0$ ;
- 3) 在  $s_i$  下, 根据  $\varepsilon$ -greedy 算法选定  $a_i$ , 执行  $a_i$ , 观察  $s_{i+1}$  和  $t_{i+1}$ , 求出  $R$ ;
- 4) 判断  $s_{i+1}$  是否为终止态, 若不是则  $i = i + 1$ , 返回步骤 3;
- 5) 更新  $Q(s_i, a_i)$ ;
- 6)  $i=i+1$ , 当  $i=m$  时, 学习结束, 否则返回步骤 2。

### 3 实验结果及其分析

#### 3.1 实验环境

此次实验采用的软硬件环境如下: 处理器为 Intel (R) Core(TM) 2 CPU 1.86 GHz, 内存 1.00 GB, 操作系统为 Windows7 旗舰版, 开发语言为 Java, agent 平台为 JADE 平台。

由于目前并没有标准的 Web 服务组合测试集, 文中采用的服务模型有 42 个, 每种服务随机生成 10 个服务实例, 即共采用 420 个服务参与服务选择过程。服务实例的 QoS 属性描述包含服务费用、执行时间、可用性、信誉度和成功率 5 个方面。每个服务实例的 QoS 属性值也在规定的范围内随机生成。

#### 3.2 实验结果

实验验证了 20 种服务实例组合过程的耗费时间和组合成功率。目的在于说明在约束条件下, agent 系统通过不断学习对于组合耗时和成功率的影响。并给出 Q 学习计算的策略下和 STBM 中服务组合耗时和成功率的比较。

实验阶段对 20 个服务组合要求进行 20 次取样, 将使用 STBM 方法和引入 Q 学习策略的组合过程相比较。组合成功率如图 1 所示。由图 1 可见, 在参与组合的服务数量少于 7 个的情况下, 两种组合方法的成功率相当, Q 学习法在初始时与一般的组合方法成功率不如 STBM 算法; 但是随着参与组合系统的服务数量增加, 融入了学习策略的组合算法成功率较之 STBM 方法组合成功率更稳定一些。

对 20 个服务组合要求进行 20 次取样, 将 STBM 方法和引入 Q 学习策略的组合过程相比较。图 2 显示随着参与组合服务数目的增加, 在网络环境经常变动的情形下, 服务组合所耗费时间明显都在增加, 但是将 Q 学习方法求解与 STBM 方法相比较, 可以看出 Q 学习算法求解组合耗费时间在服务数目少时所耗费时间与 STBM 时间相差无几。随着服务数目增加, 两种方法耗费时间虽然都在增加, 但是较之 STBM 算法, Q 学习算法下服务组合随服务数目增加耗费时间的增加渐渐地开始占据优势。

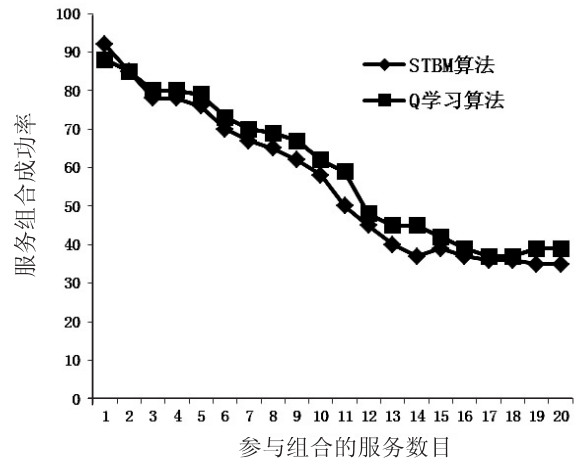


图 1 不同方法下 Web 服务组合成功率比较

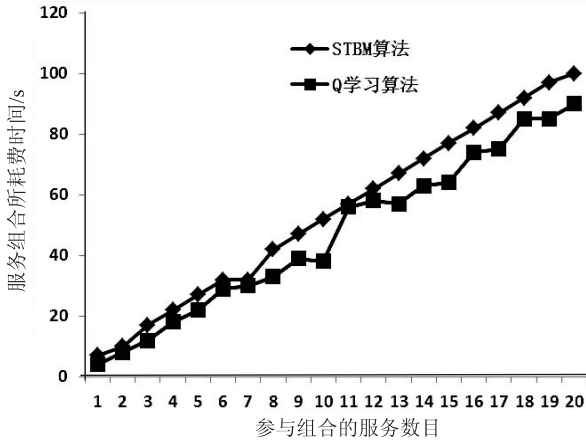


图 2 不同方法下 Web 服务随数目变化所耗费时间比较

### 4 结束语

文中以 DEC\_POMDP 为基础构造 Web 服务组合模型, 提出一种分布式合作的强化学习算法自组织地选择组合策略, 服务 agent 在各个策略中搜索最优的联合组合策略, 减少了组合耗费时间, 提高了服务组合成功率。仿真结果表明, 算法有效降低了服务 agent 感知 Web 服务 QoS 属性的错误率, 减少了预知服务组合 QoS 属性的错误率, 随着时间的增加找到可用服务几率增大, 并能逐步达到接近于最优的组合策略。将来的工作中, 在选择服务之前, 通过服务 agent 的自组织性将 QoS 值较低的单个服务从选择的队列中删除, 达到服务 agent 在参与组合之前预优化处理 agent 中服务总数。这样在选择服务之前就已经约减了质量不合格的服务, 更有利于提高所提 Q 学习算法的执行效率。

#### 参考文献:

[1] Zeng Liangzhao, Benatallah B, Ngu A H H, et al. QoS-aware middleware for Web services composition[J]. IEEE transactions on software engineering, 2004, 30(5): 311-327.



- [2] Hadad E, Lamsade J. TQoS: Transactional and QoS-aware selection algorithm for automatic Web service composition[J]. IEEE transactions on services computing, 2010, 3(1): 73-85.
- [3] Zeng Liangzhao, Benatallah B, Dumas M, et al. Quality-driven Web services composition [C]//Proc of 12th international conference on the World Wide Web (WWW). [s. l.]: [s. n.], 2003: 411-421.
- [4] 孔维梁, 刘清堂, 杨宗凯, 等. 基于动态 QoS 的 Web 服务组合[J]. 计算机科学, 2012, 39(2): 268-272.
- [5] 万里平, 高春鸣, 肖伟, 等. 基于 QoS 的 Web 服务组合的全局优化方法[J]. 计算机工程与应用, 2007, 43(24): 65-68.
- [6] 蔡美玲, 李茂桂, 周杰. 基于多目标遗传算法的多选择 Web 服务组合方法[J]. 计算机工程与应用, 2010, 46(13): 202-205.
- [7] 王飞明, 董荣胜, 胡元闯, 等. 全局优化的分层动态 Web 服务组合[J]. 计算机工程与应用, 2011, 47(12): 72-74.
- [8] Shani G, Pineau J, Kaplow R. A survey of point-based POMDP solvers[J]. Auton Agent Multi-Agent Syst, 2013, 27(1): 1-51.
- [9] Eker B, Akin H L. Solving decentralized POMDP problems using genetic algorithms[J]. Auton Agent Multi-Agent Syst, 2013, 27(1): 161-196.
- [10] Bernstein D S, Givan R, Lmmerman N, et al. The complexity of decentralized control of Markov decision process[J]. Mathematics of operations research, 2002, 27(4): 819-840.
- [11] Bosse T, Hoogendoorn M, Michel C, et al. Modeling collective decision making in groups and crowds: Integrating social contagion and interacting emotions, beliefs and intentions[J]. Auton Agent Multi-Agent Syst, 2013, 27(1): 52-84.
- [12] Chesani F, Mello P, Montali M, et al. Representing and monitoring social commitments using the event calculus[J]. Auton Agent Multi-Agent Syst, 2013, 27(1): 85-130.
- [13] Spiliopoulos L. Strategic adaptation of humans playing computer algorithms in a repeated constant-sum game[J]. Auton Agent Multi-Agent Syst, 2013, 27(1): 131-160.
- [14] Wang Shuying, Shen Weiming, Hao Qi. An agent-based Web service workflow model for inter-enterprise collaboration[J]. Expert systems with applications, 2006, 31(4): 787-799.
- [15] 周宁. Web 服务组合中若干问题的研究[D]. 南京: 南京大学, 2011.
- [16] 刘莉平. 动态 Web 服务组合关键技术研究[D]. 长沙: 中南大学, 2011.
- [17] Azar M, Munos R, Ghavamzadeh M, et al. Speedy Q-learning [C]//Proc of twenty-fifth annual conference on neural information processing systems. [s. l.]: [s. n.], 2011: 1-9.
- [18] Bertsekas D P, Yu Huizhen. Q-learning and enhanced policy iteration in discounted dynamic programming[J]. Mathematics of operations research, 2012, 37(1): 66-94.
- [19] Saad H, Mohamed A, ElBatt T. Distributed cooperative Q-learning for power allocation in cognitive femtocell networks [C]//Proc of Vehicular Technology Conference (VTC Fall). Quebec City: IEEE, 2012: 1-5.
- [20] Chen Zhe, Qiu R C. Cooperative spectrum sensing using Q-learning with experimental validation [C]//Proc of IEEE. [s. l.]: [s. n.], 2011: 405-408.

(上接第 73 页)

#### 参考文献:

- [1] Quadri I R, Gamatié A, Boulet P. Expressing embedded systems configurations at high abstraction levels with UML MARTE profile: Advantages, limitations and alternatives[J]. Journal of systems architecture, 2012, 58(5): 178-194.
- [2] Colbert E. Overview of the UML profile for the architecture analysis & design language [EB/OL]. 2004. <http://aadl.sei.cmu.edu/aadl/documents/041103%20AADL%20Profile%20SEApdf>.
- [3] Marchal B. Working XML: UML, XMI, and code generation, part1 [EB/OL]. 2004-04-15. [http://www.ibm.com/developerworks/library/xwxm23/index.html?S\\_TACT=105AGX52&S\\_CMP=cn-a-x](http://www.ibm.com/developerworks/library/xwxm23/index.html?S_TACT=105AGX52&S_CMP=cn-a-x).
- [4] 李双庆, 曹银龙. 基于 UML 扩展的实时系统建模方法[J]. 计算机工程与设计, 2006, 27(24): 4778-4780.
- [5] Feiler P H, Gluch D P, Hudak J J. The Architecture Analysis and Design Language (AADL): An introduction [R]. [s. l.]: [s. n.], 2006.
- [6] Li Qiuyan, Tian Jie, Pei Qihong, et al. Automatic transformation technology from AADL model to UML model [C]//Proc of 2011 IEEE 3rd international conference on communication software and networks. [s. l.]: [s. n.], 2011: 255-258.
- [7] 许峰, 王志坚. 基于 MDA 的构件开发方法的研究[J]. 计算机应用与软件, 2009, 26(11): 30-33.
- [8] 熊永刚, 唐慧佳. 基于 XMI 的 UML 模型到 XML 文档转换的实现[J]. 计算机应用与软件, 2010, 27(4): 68-70.
- [9] 应一舟, 叶丽君, 郭义喜. 基于 XML 的 UML 时序图向 Petri 网的转换[J]. 计算机工程, 2009, 35(22): 84-87.
- [10] 赵俊峰, 周建涛. UML 时序图向 PNML 转换的研究与实现[J]. 武汉大学学报(理学版), 2011(6): 511-516.
- [11] 严丽, 马宗民, 刘健, 等. 模糊 XML DTD 到 UML 数据模型的转换[J]. 小型微型计算机系统, 2009(4): 586-593.
- [12] 王瀚博, 周兴社, 董云卫, 等. 结构分析和设计语言 AADL 研究[J]. 计算机工程与应用, 2009, 45(16): 1-4.

基于DEC\_POMDP的Web服务组合优化算法

作者：[赵莉](#)，[李蜀瑜](#)，[ZHAO Li](#)，[LI Shu-yu](#)  
作者单位：[陕西师范大学 计算机科学学院, 陕西 西安, 710062](#)  
刊名：[计算机技术与发展](#)

英文刊名：[Computer Technology and Development](#)

年，卷(期)：2014(3)

本文链接：[http://d.g.wanfangdata.com.cn/Periodical\\_wjfz201403019.aspx](http://d.g.wanfangdata.com.cn/Periodical_wjfz201403019.aspx)