

University of Wollongong

Research Online

Faculty of Engineering and Information
Sciences - Papers: Part A

Faculty of Engineering and Information
Sciences

1-1-2014

Ant-inspired multi-phase and multi-party negotiations in the data-intensive service provision

Lijuan Wang
lw840@uowmail.edu.au

Jun Shen
University of Wollongong, jshen@uow.edu.au

Qingguo Zhou
Lanzhou University

Ghassan Beydoun
University of Wollongong, beydoun@uow.edu.au

Follow this and additional works at: <https://ro.uow.edu.au/eispapers>



Part of the [Engineering Commons](#), and the [Science and Technology Studies Commons](#)

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

Ant-inspired multi-phase and multi-party negotiations in the data-intensive service provision

Abstract

The rapid proliferation of enormous sources of digital data and the development of cloud computing have led to greater dependence on data-intensive services. Each service may actually request or create a large amount of data sets. To compose these services will be more challenging. Issues of autonomy, scalability, adaptability, and robustness, become difficult to resolve. In order to automate the process of reaching an agreement in data-intensive service provision, the ant-inspired negotiation mechanism is considered in this paper. There are twostage negotiation procedures in our model, which will provide effective and efficient service selection for service composers. We also present a multi-phase, multi-party negotiation protocol, where the ant colony system is applied for selecting the services. The experimental results show that our ant-inspired negotiation approach can facilitate the data-intensive service provision. Index Terms-Ant colony system, data-intensive service composition, automated negotiation, quality of service.

Keywords

ant, intensive, data, negotiations, party, phase, multi, provision, inspired, service

Disciplines

Engineering | Science and Technology Studies

Publication Details

Wang, L., Shen, J., Zhou, Q. & Beydoun, G. (2014). Ant-inspired multi-phase and multi-party negotiations in the data-intensive service provision. IEEE International Conference on Services Computing (pp. 211-218). United States: IEEE.

Ant-Inspired Multi-Phase and Multi-Party Negotiations in the Data-Intensive Service Provision

Lijuan Wang¹, Jun Shen¹, Qingguo Zhou², and Ghassan Beydoun¹

¹School of Information Systems and Technology

University of Wollongong, Wollongong, NSW, Australia

²School of Information Science and Engineering

Lanzhou University, Lanzhou, P. R. China

E-mail: lw840@uowmail.edu.au, jshen@uow.edu.au, zhouqg@lzu.edu.cn, beydoun@uow.edu.au

Abstract—The rapid proliferation of enormous sources of digital data and the development of cloud computing have led to greater dependence on data-intensive services. Each service may actually request or create a large amount of data sets. To compose these services will be more challenging. Issues of autonomy, scalability, adaptability, and robustness, become difficult to resolve. In order to automate the process of reaching an agreement in data-intensive service provision, the ant-inspired negotiation mechanism is considered in this paper. There are two-stage negotiation procedures in our model, which will provide effective and efficient service selection for service composers. We also present a multi-phase, multi-party negotiation protocol, where the ant colony system is applied for selecting the services. The experimental results show that our ant-inspired negotiation approach can facilitate the data-intensive service provision.

Index Terms—Ant colony system, data-intensive service composition, automated negotiation, quality of service.

I. INTRODUCTION

In the past ten years or so, Web services selection and composition have been attracting much research attention. To obtain a viable business model for composite services, the dynamic service price-setting models are increasingly negotiations-based [1]–[4]. Meanwhile, the rapid proliferation of enormous sources of digital data and the development of cloud computing have led to greater dependence on data-intensive services. Each service may actually request or create a large amount of data sets. To compose these services will be more challenging. Issues of autonomy, scalability, adaptability, and robustness, become difficult to resolve. Indeed, new mechanisms are needed to overcome those issues and to keep the cost and response time of composite services acceptable.

In general, the data-intensive service composition will be cooperatively supported by three stakeholders: the service composers, the service providers, and the data providers. Providers need an approach to regulate and price their resources, either services or data or their combination. They all want to have a good market position maximizing their profits. The decisions of all three stakeholders depend on each other. The data provider sells data sets to multiple service providers in order to maximize the data usage and the profit. The cost and response time of data sets for one service provider are affected by the demand of the others.

The service providers also play the requesters role with respect to the data sets. Thus, service providers will have two aims, one is to lower the access cost and response time of data sets and the other is to maximize their profit and service usage. Also, the service providers compete with other service providers to initiate or maintain a contract with the service composers and are invariably interested in cost saving. The actual usage of services typically encourages the composers to have a long term contract with the service providers. They also select concrete services that best match the QoS requirements. Meanwhile, data-intensive services are typically used in a dynamic and changing environment, and different providers typically have conflicting objectives. In order to automate the process of reaching an agreement in our problem, we exploit a group of agents automatically negotiating to establish agreeable service contracts.

A negotiation process is the interplay of offers and counter-offers between a buyer and a seller, with different criteria and goals, working to identify a mutually acceptable solution. Automated negotiation normally follow negotiation protocols, exchange negotiation objects, and are driven by decision making models [5]. Negotiation protocols govern the negotiation by defining rules such as when the negotiation process ends, what deals can be made, and what sequences of offers are allowed. Negotiation objects are the issues such as price and time over which the negotiation takes place. Decision making models are used for evaluating and generating offers and counter-offers.

The lifetime of our problem is described in Fig. 1. The first step is that the service composer tries to select a set of service candidates while the data provider provides data sets, and the second step is that if a feasible solution which satisfies the service composer's local and global QoS constraints does not exist, negotiations are performed in order to determine new quality values for each involved service. In the lifetime, two-stage negotiation processes are used. In the first stage, a service composer negotiates with multiple service providers over each service in a structured one-to-many negotiation process. In the second stage, each service provider negotiates with a data provider over a set of data sets in a structured one-to-one negotiation process. This paper will focus on the two-stage

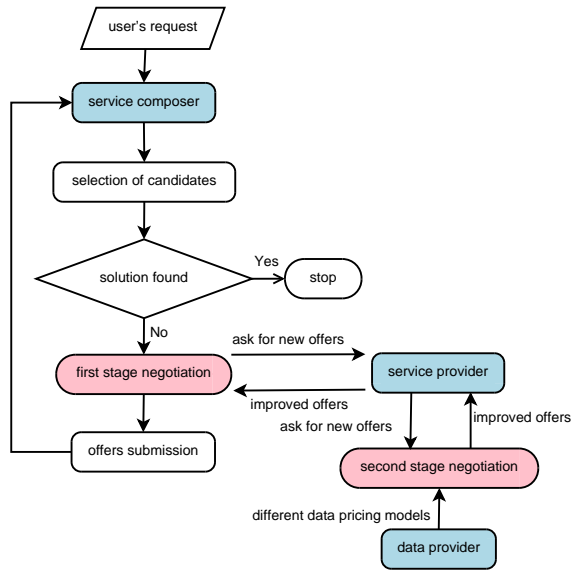


Fig. 1. The lifetime of the data-intensive service provision

negotiation procedures as described in the second step.

The remainder of this paper is organized as follows. Section II introduces the related work. Section III presents a multi-phase, multi-party negotiation protocol. Section IV discusses the decision making model. Section V shows the performance evaluation. Finally, Section VI concludes this paper and proposes future work.

II. RELATED WORK

In a Web service environment, abstract services are the functional descriptions of services, and concrete services represent the existing services available for potential invocation of their functionalities and capabilities. When the functions of several concrete services are consistent with the functional description of an abstract service, these concrete services are the service candidates for the abstract service and QoS attributes are used to distinguish them. Web service composition is the process of changing some existing Web services into a new service to satisfy users' requirements. Web service selection is an important part of this process. Given a request for composite service, which involves a set of abstract services and dependency relationships among them, there is a list of service candidate sets which includes many concrete services for each abstract service. Web service selection refers to finding one service candidate to implement each abstract service according to users' requirements.

The literature has presented many methods for Web service selection and composition. By a detailed analysis of each approach, we find that bio-inspired algorithms can overcome the challenging requirements of data-intensive service provision [6]. It is useful for the provision of data-intensive services to explore key features and mechanisms of biological systems and to emulate biological mechanisms to services system. Compared with integer programming (IP) algorithms [7], [8], linear programming (LP) algorithms [9], or mixed integer

programming (MIP) algorithms [10]–[14], the bio-inspired algorithms usually require much less computation time and are capable of achieving near-optimal solutions, especially in dynamic environments [15], [16]. We have also been applying bio-inspired algorithms to tackle the data-intensive service composition problem [17]–[23]. An economical model of the data-intensive service provision and an extensible QoS model were presented in our previous work [23]. We have presented an enhanced ant colony algorithm and a modified genetic algorithm [19], [20]. However, in our earlier work, we did not consider scenarios where the requirements of the global QoS constraints are significantly severe.

Negotiation has been adopted in service provision in order to get better QoS attributes. An iterative negotiation approach for a service composition was presented in [2]. The aim of the approach was to select services for the service-based systems in the scenarios where the QoS constraints were severe. In [3], the service level agreements for a service composition were established through autonomous agent negotiation. A new negotiation protocol was also proposed to support coordinated negotiation. The authors of [10] introduced an approach for the Web service selection problem with large scale processes and severe QoS constraints. The Web service selection problem was formalized as a MIP problem and loop peeling was adopted for optimization in that paper.

However, the negotiation approaches in the above studies are not able to effectively solve our problem, in which the data plays the dominant role. The cost and response time of services largely depend on the accessing cost and response time of data sets. The negotiation for our problem is a multi-phase and multi-party process. The service composer should be able to negotiate with multiple service providers over each abstract service, while a service provider should be able to negotiate with the data providers over the data sets required by the services. To address the above issues, this paper presents an ant-inspired negotiation approach.

III. THE EXTENDED NEGOTIATION PROTOCOL

We have designed a multi-phase, multi-party negotiation protocol for our problem. The proposal is based on the Iterated Contract Net Interaction Protocol (ICNIP), which is provided by the Foundation of Intelligent Physical Agents (FIPA) [24]. ICNIP is the most widely used negotiation protocol for one-to-many negotiation in the agent community. It is also applicable in one-to-one negotiation, as it is considered to be a particular case of one-to-many negotiation. ICNIP supports recursive negotiation and allows multi-round iterative negotiation to reach an agreement. However, the standard ICNIP is insufficient in supporting QoS negotiation for our problem, as it is unable to allow multi-party negotiation or combine multiple negotiation processes.

The negotiators, namely the service composer, the service provider, and the data provider will be represented by agents. We refer to agents acting on behalf of service composers as composer agents (CAs), agents representing service providers as service agents (SAs), and agents representing data providers

as data agents (DAs). Basically, an agent starts by offering a value for the negotiation object (say, a price) to its opponent. The opponent can accept the offer, exit the negotiation, or make a counter-offer. The negotiation may be iterative as several rounds of offers and counter-offers will occur until one of the agents accepts, or exits the negotiation. The agent might exit the negotiation when the time deadline is reached without an agreement being in place. The iteration is referred to as a negotiation round, and the time deadline is referred to as the number of allowed rounds.

In our model, if a feasible solution that satisfies the global QoS constraints is not found, the CA initiates the negotiation by sending call-for-proposals (cfps) to SAs. Because the cost and response time of data dominate the cost and response time of services, it is possible that SAs negotiate with DAs in advance to reply to the CA. Meanwhile, the CA contacts all SAs of each abstract service in the workflow. It will prepare m cfps in each negotiation round, and these cfps are sent to the m candidate SAs available to take part in the negotiation round. After receiving offers for all abstract services, the CA evaluates the overall QoS attributes based on the deals from various SAs. ICNIP is hence extended to allow a CA to aggregate results of the individual services, which have been selected to execute each abstract service, and to perform the confirmation to instruct SAs in the negotiations. Also, the extended protocol should allow the negotiations between SAs and DAs.

There are three phases in the extended protocol:

1) Phase 1, which allows us to find out how many SAs are available to enter the negotiation, and their offers over the objects to be negotiated. This phase includes one-shot interaction between the CA and all SAs, and it also includes one-shot interaction between all SAs and the DA. The CA is involved in simultaneous negotiations with multiple SAs. Each negotiation between the CA and a SA is private and holds a lot of information. Each SA is unaware of its competitors' status in the current composition. If the CA sends the current winning offers to potential SAs, SAs can analyze their positions and adapt quickly [1], [2]. This in turn significantly reduces the search space by guiding the negotiation process to proceed in the right direction towards optimal solutions. The SAs process the request from the CA and may decide to negotiate with the DA. Before a SA answers the CA, it evaluates the offers received from the DA. The DA may accept the counter-offer from a SA, or the DA may refuse to participate in the negotiation, or the DA may provide a new offer to the SA. Thus, the SA can answer the CA based on the results of the nested negotiation offers. The type of response generated by a SA in this phase is based on the following conditions.

- a) proposal, the SA will respond positively and the process will terminate.
- b) refusal, the SA refuses to participate in the negotiation.

- c) counter-proposal, the SA will respond to the CA with its new offer.

The negotiation protocol is shown in Fig. 2.

- 2) Phase 2, which allows us to iterate the negotiation. In case the first negotiation phase ends up with counter-proposals from SAs, the CA will initiate negotiation with the SAs and send them cfps with the set of current winning offer and counter-offer as its content. The following response options are available to SAs.
 - a) proposal, the SA will respond positively and the process will terminate.
 - b) counter-proposal, the SA responds with a counter-offer.

This negotiation phase is repeated until there is no counter-proposal from SAs or the time deadline is reached. The negotiation protocol is shown in Fig. 3.

- 3) Phase 3, which allows us to end the negotiation. If the second negotiation phase ends up with proposals, the CA will evaluate the overall QoS attributes. If the overall QoS requirements are satisfied, the CA informs all SAs about the acceptance of the offers. Then SAs inform the DA about the acceptance of the dependent offers. The negotiation leads to a signed contractual agreement since it ends with an optimal solution that satisfies the requirements. The negotiation protocol is shown in Fig. 4. In the case that the overall QoS requirements

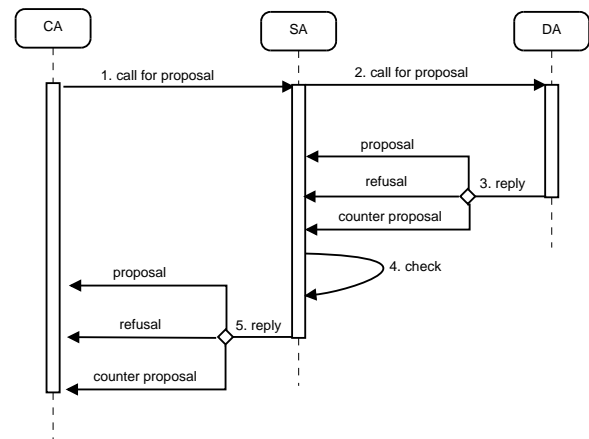


Fig. 2. Protocol to support first phase of negotiation

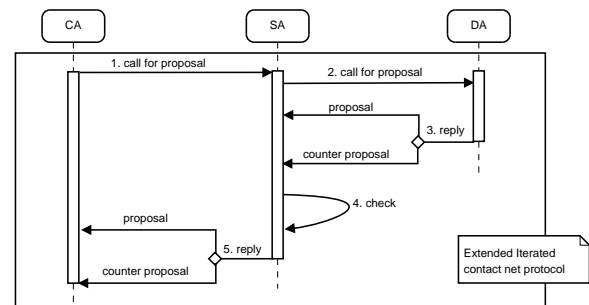


Fig. 3. Protocol to support second phase of negotiation

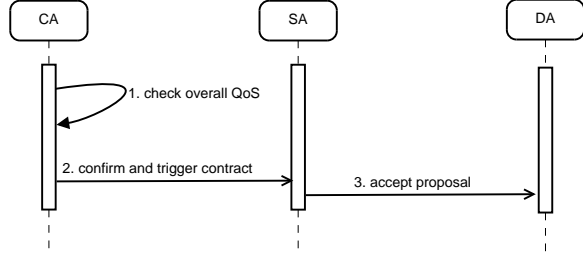


Fig. 4. Protocol to support third phase of negotiation in case of successful negotiation

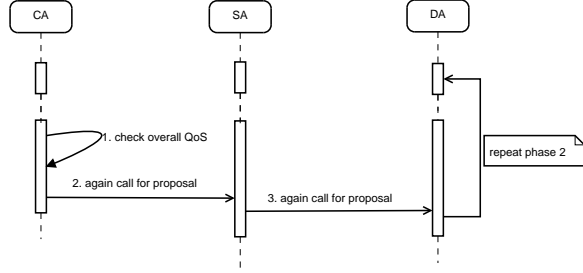


Fig. 5. Protocol to support third phase of negotiation in case of failed negotiation

are not satisfied based on the current deals, the CA will restart the second negotiation phase with respect to one or many abstract services. Then each corresponding SA sends cfps to the DA and commences a new negotiation process. The negotiation protocol is shown in Fig. 5.

IV. DECISION MAKING MODEL

The negotiation process between two agents is a bilateral interaction that consists of a succession of offers and counter-offers. Let a ($a \in \{CA, SA, DA\}$) represent the negotiating agent and o ($o \in \{1, 2, \dots, r\}$) the negotiation object (or issue). Each agent has a defined delimited range to consider the value of an issue. For example, the CA has the maximum price at which it would buy the service and the SA has the minimum price at which it would sell the service. Let $x_o \in [min_o^a, max_o^a]$ be a value for issue o acceptable to agent a . The agent acts alternately, making or accepting offers and counter-offers on the value x_o , or abandoning the negotiation when the time deadline is reached without an agreement being in place. Each agent a has a utility function $U_a(x_o) : [min_o^a, max_o^a] \rightarrow [0, 1]$, representing its satisfaction of the value x_o . For convenience, the utility values are kept in the interval $[0, 1]$. We adopted linear utility function as used in many cost-benefit models in the literature [4], [25], [26]. Each negotiator can also define the utility functions more explicitly [27].

To negotiate on multiple issues (say r), a normalized weight w_o^a ($\sum_{1 \leq o \leq r} w_o^a = 1$) representing the relative importance of each issue o for agent a is assigned under negotiation. Then an agent's utility function for an offer $x = (x_1, x_2, \dots, x_r)$ in

the multi-dimensional space is given by (1).

$$U_a(x) = \sum_{o=1}^r w_o^a * U_a(x_o) \quad (1)$$

We use the variable $x_{b \rightarrow a}^t$ to denote the offer that agent a received from agent b at time t , and use the variable $x_{a \rightarrow b}^t$ to denote the offer that agent a is ready to send to agent b at time t' , and $t < t'$. A constant t_{max}^a is used to represent the time by which agent a must have completed the negotiation. For agent a , if the utility value of offer-to-make $U_a(x_{a \rightarrow b}^t)$, is less than the utility value of the received offer $U_a(x_{b \rightarrow a}^t)$, then the offer is accepted, otherwise a counter-offer is proposed within the time deadline. We use $I^a(t', x_{b \rightarrow a}^t)$ to denote the interpretation of an agent a at time t' of an offer $x_{b \rightarrow a}^t$. According to [26], $I^a(t', x_{b \rightarrow a}^t)$ is defined as (2).

$$I^a(t', x_{b \rightarrow a}^t) = \begin{cases} reject & \text{if } t' > t_{max}^a, \\ accept & \text{if } U_a(x_{a \rightarrow b}^t) \leq U_a(x_{b \rightarrow a}^t), \\ x_{a \rightarrow b}^t & \text{otherwise.} \end{cases} \quad (2)$$

In order to prepare a counter-offer, $x_{a \rightarrow b}^t$, agent a uses a set of tactics that generate new values for each issue in the negotiation set. The time dependent tactics, which use time-based decision functions $\alpha_o^a(t)$ ($0 \leq \alpha_o^a(t) \leq 1$), are often used in negotiation systems [4], [25], [26], [28], [29]. In this case, the counter-offer generated from agent a to b at time t' with respect to each issue o , $x_{a \rightarrow b}^t[o]$, is given by (3).

$$x_{a \rightarrow b}^t[o] = \begin{cases} min_o^a + \alpha_o^a(t)(max_o^a - min_o^a) & \text{if } U_a(x(o)) \text{ is decreasing,} \\ max_o^a - \alpha_o^a(t)(max_o^a - min_o^a) & \text{if } U_a(x(o)) \text{ is increasing.} \end{cases} \quad (3)$$

The counter-offer $x_{a \rightarrow b}^t[o]$ will always be between the value range $[min_o^a, max_o^a]$. When the time deadline is reached, the tactic will suggest offering the reservation value. The reservation value for issue o of agent a represents the value that provides the smallest utility. If the utility function $U_a(x(o))$ is monotonically increasing, the reservation value is min_o^a , otherwise if $U_a(x(o))$ is decreasing, the reservation value is max_o^a .

The time-based decision function $\alpha_o^a(t)$ can be defined as the exponential function, polynomial function, and sigmoid function [4], [26], [29]. For example, the polynomial function can be given by (4).

$$\alpha_o^a(t) = \kappa_o^a + (1 - \kappa_o^a) * \left(\frac{\min(t, t_{max}^a)}{t_{max}^a} \right)^{\frac{1}{\delta}} \quad (4)$$

The variable δ is a parameter used to control the concession rate, and κ_o^a is the initial concession at $t = 0$, where $\alpha_o^a(0) = \kappa_o^a$ ($0 \leq \kappa_o^a \leq 1$) and $\alpha_o^a(t_{max}^a) = 1$.

For each type of agent, the above time dependent tactic, which uses the polynomial function, is used to generate counter-proposals. For each bilateral negotiation between the CA and a SA or between a SA and the DA, an agreement is

Function ACS(G)**Initialization:**

MaxIt: the maximum number of iterations;

noa: the number of artificial ants;

Input:

G: a directed graph;

Output:

S: a service execute path to create a composite service;

optimal: a variable used to indicate whether the solution *S* is an optimal solution. If *S* is an optimal solution, then the value of *optimal* is 1, otherwise 0.

```
1:  $S = \emptyset$ ;  
2:  $step = 0$ ;  
3: while  $step < MaxIt$  do  
4:    $step = step + 1$ ;  
5:   set all ants at start vertex;  
6:   for each ant  $k$  in  $noa$  do  
7:     while ant  $k$  is not at the end vertex do  
8:       ant  $k$  chooses successor  $j$ ;  
9:       ant  $k$  updates its candidate list;  
10:      apply the local updating rule;  
11:     end while  
12:   end for  
13:   when all ants arrive at the end vertex, find the best path  
   from all candidate lists and compare the utility of the  
   best path with the utility of  $S$ . If the utility of the best  
   path is larger than the utility of  $S$ , then  $S$  is replaced  
   by the best path.  
14:   apply the global updating rule using the utility of  $S$ ;  
15: end while  
16: return  $S$  and  $optimal$ .
```

possible if there is some degree of intersection between the reservation intervals of the two agents.

The service composition problem is modeled as a directed graph, and an ant colony system (ACS) is applied by the service composer to find the solution [19]. After CA and SA finish their negotiation process, we need to modify the pheromone information of ACS. Because if too much old pheromone information is maintained, the ants will be stuck in a local optimal solution. We have studied different strategies to modify the pheromone information in order to adapt ACS to handle the dynamic data-intensive service composition problem. The pheromone modification strategy is chosen based on the negotiation results. That is to say, we will choose the pheromone modification strategy according to the number of changes and the frequency of occurrence of changes in the negotiation process. The data-intensive service composition based on ACS is given in Function ACS(G), and the negotiation-based approach is illustrated in Algorithm 1.

V. EXPERIMENTS AND ANALYSES

The aim of this evaluation is to analyze the performance of the proposed ant-inspired negotiation approach: 1) observing the effectiveness of our approach, measured by the success rate

Algorithm 1 The ant-inspired negotiation approach for the data-intensive service provision problem

Initialization:

Tmax: the maximum number of negotiation rounds;

Input:

G: a directed graph;

Output:

S and *optimal*;

```
1:  $S = \emptyset$ ;  
2:  $iteration = 0$ ;  
3:  $[optimal, S] = ACS(G)$ ; // The ACS function with two  
   return values is used to select services.  
4: while  $optimal == 0$  do  
5:    $iteration = iteration + 1$ ;  
6:   if  $iteration > Tmax$  then  
7:     break;  
8:   end if  
9:   for each abstract service do  
10:    for each concrete service do  
11:      CA sends a counter-offer to SA;  
12:      SA sends a counter-offer to DA;  
13:      DA interprets the counter-offer from SA. If DA  
      accepts the counter-offer from SA, the negotiation  
      between SA and DA ends. If DA sends a new offer  
      to SA, then SA interprets the new offer. If SA  
      accepts the new offer from DA, the negotiation  
      between SA and DA ends. If SA sends a counter-  
      offer to DA, then this step is repeated until no  
      counter-offer is proposed within the time deadline  
      of SA and DA.  
14:      SA interprets the counter-offer from CA. If SA  
      accepts the counter-offer from CA, the negotiation  
      between CA and SA ends. If SA sends a new offer  
      to CA, then CA interprets the new offer. If CA  
      accepts the new offer from SA, the negotiation  
      between CA and SA ends. If CA sends a counter-  
      offer to SA, then steps 12-14 are repeated until no  
      counter-offer is proposed within the time deadline  
      of CA and SA.  
15:      if the negotiation between CA and SA is successful  
      then  
16:        change the related parameters of  $G$ ;  
17:      end if  
18:    end for  
19:  end for  
20:  modifying the pheromone matrix of ACS;  
21:   $[optimal, S] = ACS(G)$ ;  
22: end while  
23: return  $S$  and  $optimal$ .
```

of finding an optimal solution; 2) investigating the efficiency of our approach, measured by the number of negotiation rounds and the computation time in each negotiation round; and 3)

studying the effect of the problem size on the performance of our approach. We also implement a MIP approach to solve the same problem and compare our approach with it. We used the open source integer programming system *lpsolve* version 5.5 [30] for the MIP. All the experiments are conducted on computers with Inter Core i5 2500 CPU (3.3GHz and 8 GB RAM).

A. Test Case Generation

In this paper, we restrict our attention to the cost and response time attributes, but other QoS attributes could be easily added to our approach by modifying ACS.

The performance of our approach is affiliated to the size of the data-intensive service provision problem. The size of our problem depends on the number of abstract services used in the composite service, the number of concrete services for each abstract service, and the number of data sets required by each abstract service. For the purpose of our evaluation, we considered different scenarios, where a composite application comprises services from n abstract services, and n varies in our experiments between 10 and 100, in increments of 10. There are m concrete services for each abstract service, and m varies in our experiments between 10 and 100, in increments of 10. Each abstract service requires a set of d data sets, and d varies in our experiments between 1 and 10, in increments of 1. A scenario generation system is designed to generate the scenario for experiments. The system first determines a basic scenario, which includes sequence, conditional and parallel structures. With this basic scenario, other scenarios are generated by either placing an abstract service into it or adding another composition structure as substructure. This procedure continues until the scenario has the predefined number of abstract services.

For each scenario, the price of a data set, the network bandwidth (Mbps) between each data server and the service platform, the storage media speed (Mbps), the size (MB) of a data set and the number of data requests in the waiting queue were randomly generated from the following intervals: [1,100], [1,100], [1,100], [1000,10000] and [1,10]. Each time the number of abstract services, the number of service providers, or the number of data sets was changed, 50 instances of experiments were run and the average results were reported.

B. Result Analysis

The experimental results of all scenarios are grouped into three test sets. In the first set, n varies from 10 to 100, while $m = 10$ and $d = 5$. In the second set, m varies from 10 and 100, while $n = 10$ and $d = 5$. In the third set, d varies from 1 to 10, while $n = 10$ and $m = 10$.

To evaluate the effectiveness of our approach, we compare the success rate of our approach with that of the MIP approach. The success rate is the percentage of instances where a solution could be found. The success rate of the negotiation approach and the MIP approach with respect to the number of abstract services, the number of service providers, and the number of data sets are presented in Figs. 6, 7,

and 8. The results illustrate that the success rate of the MIP approach remains zero in all scenarios, while the negotiation approach maintains a higher success rate compared with the MIP approach. Specifically, the average success rate of our approach is 98.40%, 98.00%, 93.20% in the first, second, and third set of experiment.

The number of negotiation rounds is a relevant efficiency factor of the negotiation approach. A large value in the negotiation round implies that the problem has to be solved

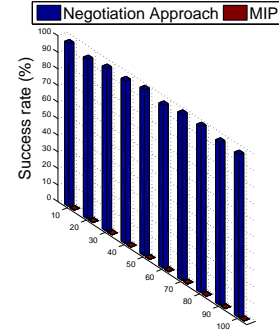


Fig. 6. Success rate vs. number of abstract services

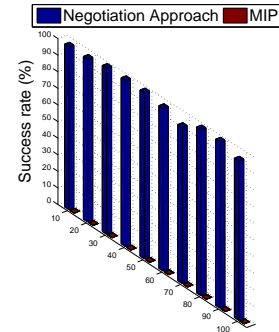


Fig. 7. Success rate vs. number of service providers

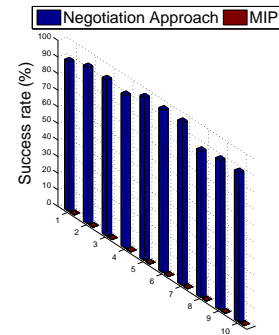


Fig. 8. Success rate vs. number of data sets

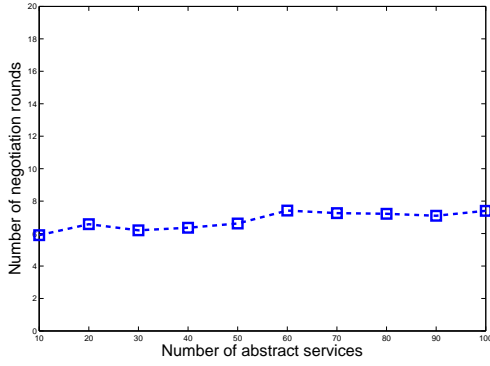


Fig. 9. Number of negotiation rounds vs. number of abstract services

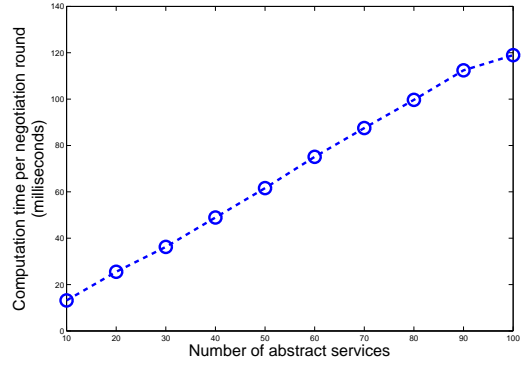


Fig. 12. Computation time per negotiation round vs. number of abstract services

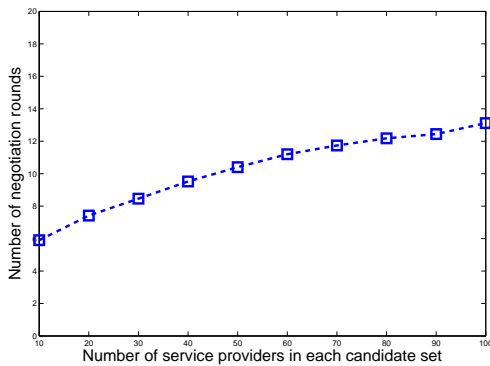


Fig. 10. Number of negotiation rounds vs. number of service providers

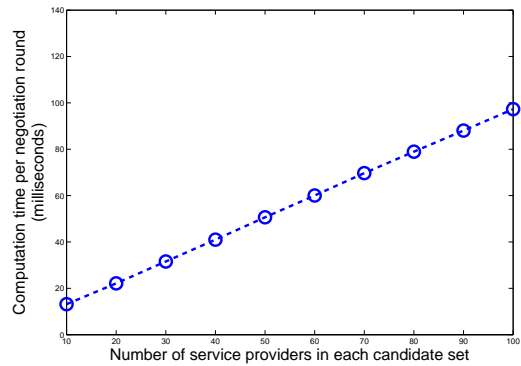


Fig. 13. Computation time per negotiation round vs. number of service providers

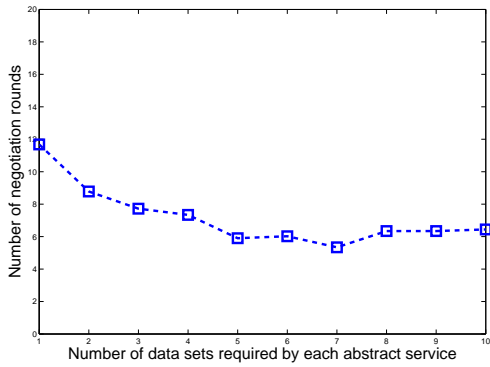


Fig. 11. Number of negotiation rounds vs. number of data sets

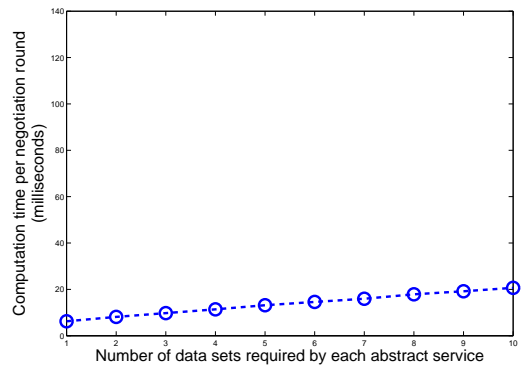


Fig. 14. Computation time per negotiation round vs. number of data sets

via many iteration numbers in the whole negotiation process. Figs. 9, 10, and 11 show the negotiation rounds of the three sets. These figures indicate that the number of negotiation rounds increases when the number of service providers increases. When the number of abstract services or the number of data sets increases, the number of negotiation rounds, on the other hand, does not increase. On average, it took approximately seven rounds to find an optimal solution in the first and third test sets.

Besides the number of negotiation rounds, the time consumption for each round is also a very important factor with respect to the efficiency. Figs. 12, 13, and 14 present the average time consumption per negotiation round of the three sets. The results indicate that the time consumption per round increases when the size of the problem increases. As illustrated, the time consumption per round is linear rather than exponential.

VI. CONCLUSION

The provision of data-intensive service faces new challenges with the rapid proliferation of services and the development of cloud computing. The scope, number, and complexity of data-intensive services are all set to soar in the future. The cost and response time of data sets have a significant effect on the QoS attributes of the composite service. This paper has proposed an ant-inspired negotiation approach for the data-intensive service provision problem. The two-stage negotiation procedures provide effective and efficient service selections for the service composer. A multi-phase, multi-party negotiation protocol is also presented. The experimental evaluation shows that our approach is able to find an optimal solution in scenarios where severe quality constraints are imposed on the problem.

In future work, we will further investigate strategies that service providers and data providers can deploy to reduce the price and decrease the execution time of services. For example, the service provider can subscribe data sets and move some lightweight data sets to the service platform, or the data provider can relocate data sets in order to decrease response time of moving data sets. These strategies can improve the success rate in finding optimal solution and decrease the total cost of providing data-intensive services.

REFERENCES

- [1] K. Hashmi, A. Alhosban, Z. Malik, and B. Medjahed, "Webneg: A genetic algorithm based approach for service negotiation," in *2011 IEEE International Conference on Web Services (ICWS)*. Washington, DC, USA: IEEE Computer Society, 2011, pp. 105–112.
- [2] Q. He, Y. Yang, J. Yan, and H. Jin, "Insc: An iterative negotiation approach for service compositions," in *IEEE 9th International Conference on Services Computing (SCC)*. Washington, DC, USA: IEEE Computer Society, 2012, pp. 170–177.
- [3] J. Yan, R. Kowalczyk, J. Lin, M. B. Chhetri, S. K. Goh, and J. Zhang, "Autonomous service level agreement negotiation for service composition provision," *Future Generation Computer Systems*, vol. 23, no. 6, pp. 748–759, 2007.
- [4] F. H. Zulkernine and P. Martin, "An adaptive and intelligent sla negotiation system for web services," *IEEE Transactions on Services Computing*, vol. 4, no. 1, pp. 31–43, 2011.
- [5] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, M. J. Wooldridge, and C. Sierra, "Automated negotiation: Prospects, methods and challenges," *Group Decision and Negotiation*, vol. 10, no. 2, pp. 199–215, 2001.
- [6] L. Wang and J. Shen, "A critical systematic review of service concretization based on bio-inspired approaches," 2014, <http://ro.uow.edu.au/eispapers/1903>.
- [7] L. Zeng, B. Benattallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "Qos-aware middleware for web services composition," *IEEE Transactions on Software Engineering*, vol. 30, no. 5, pp. 311–327, 2004.
- [8] R. Aggarwal, K. Verma, J. Miller, and W. Milnor, "Constraint driven web service composition in meteor-s," in *Proceedings of IEEE International Conference on Services Computing (SCC '04)*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 23–30.
- [9] V. Cardellini, E. Casalicchio, V. Grassi, and F. L. Presti, "Efficient provisioning of service level agreements for service oriented applications," in *2nd International Workshop on Service Oriented Software Engineering: in Conjunction with the 6th ESEC/FSE Joint Meeting (IW-SOSWE '07)*. New York, NY, USA: ACM, 2007, pp. 29–35.
- [10] D. Ardagna and B. Pernici, "Adaptive service composition in flexible processes," *IEEE Transactions on Software Engineering*, vol. 33, no. 6, pp. 369–384, 2007.
- [11] —, "Global and local qos guarantee in web service selection," in *Business Process Management Workshops*, ser. Lecture Notes in Computer Science, C. J. Bussler and A. Haller, Eds. Berlin, Heidelberg: Springer-Verlag, 2006, vol. 3812, pp. 32–46.
- [12] M. Alrifai, T. Risse, and W. Nejdl, "A hybrid approach for efficient web service composition with end-to-end qos constraints," *ACM Transactions on the Web*, vol. 6, no. 2, pp. 7:1–7:31, 2012.
- [13] Y. Qu, C. Lin, Y. Wang, and Z. Shan, "Qos-aware composite service selection in grids," in *Proceedings of 5th International Conference on Grid and Cooperative Computing (GCC '06)*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 458–465.
- [14] R. Ramacher and L. Monch, "Cost-minimizing service selection in the presence of end-to-end qos constraints and complex charging models," in *IEEE Ninth International Conference on Services Computing (SCC)*. Washington, DC, USA: IEEE Computer Society, 2012.
- [15] G. Canfora, M. Penta, R. Esposito, and M. L. Villani, "An approach for qos-aware service composition based on genetic algorithms," in *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation (GECCO '05)*. New York, NY, USA: ACM, 2005, pp. 1069–1075.
- [16] X. Q. Fan, X. W. Fang, and C. J. Jiang, "Research on web service selection based on cooperative evolution," *Expert Systems with Applications*, vol. 38, no. 8, pp. 9736–9743, 2011.
- [17] J. Shen, G. Beydoun, S. Yuan, and G. Low, "Comparison of bio-inspired algorithms for peer selection in services composition," in *Proceedings of the 2011 IEEE International Conference on Services Computing (SCC)*. Washington, DC, USA: IEEE Computer Society, 2011, pp. 250–257.
- [18] L. Wang and J. Shen, "Towards bio-inspired cost minimisation for data-intensive service provision," in *IEEE First International Conference on Services Economics*. Washington, DC, USA: IEEE Computer Society, 2012, pp. 16–23.
- [19] L. Wang, J. Shen, and G. Beydoun, "Enhanced ant colony algorithm for cost-aware data-intensive service provision," in *IEEE Ninth World Congress on Services*. Washington, DC, USA: IEEE Computer Society, 2013, pp. 227–234.
- [20] L. Wang and J. Shen, "Economical data-intensive service provision supported with a modified genetic algorithm," in *IEEE 2nd International Congress on Big Data*. Washington, DC, USA: IEEE Computer Society, 2013, pp. 358–365.
- [21] L. Wang, J. Luo, J. Shen, and F. Dong, "Cost and time aware ant colony algorithm for data replica in alpha magnetic spectrometer experiment," in *IEEE 2nd International Congress on Big Data*, 2013, pp. 254–261.
- [22] L. Wang, J. Shen, C. Di, Y. Li, and Q. Zhou, "Towards minimizing cost for composite data intensive services," in *Proceeding of the IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. Washington, DC, USA: IEEE Computer Society, 2013, pp. 293–298.
- [23] L. Wang, J. Shen, J. Luo, and F. Dong, "An improved genetic algorithm for cost-effective data-intensive service composition," 2013, the 9th International Conference on Semantics, Knowledge & Grids.
- [24] FIPA, "Iterated contract net interaction protocol specification," 2002, <http://www.fipa.org/specs/fipa00030/SC00030H.pdf>.
- [25] M. Comuzzi and B. Pernici, "An architecture for flexible web service qos negotiation," in *Proceedings of the Ninth IEEE International EDOC Enterprise Computing Conference*, ser. EDOC '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 70–82.
- [26] P. Faratin, C. Sierra, and N. R. Jennings, "Negotiation decision functions for autonomous agents," *Robotics and Autonomous Systems*, vol. 24, no. 3–4, pp. 159–182, 1998.
- [27] J. Wilkes, "Utility functions, prices, and negotiation," HP Labs, Tech. Rep. HPL-2008-81, 2008.
- [28] M. B. Chhetri, J. Lin, S. Goh, J. Zhang, R. Kowalczyk, and J. Yan, "A coordinated architecture for the agent-based service level agreement negotiation of web service composition," in *Proceedings of the Australian Software Engineering Conference*, ser. ASWEC '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 90–99.
- [29] H. Li, S. Y. W. Su, and H. Lam, "On automated e-business negotiations: Goal, policy, strategy, and plans of decision and action," *Journal of Organizational Computing and Electronic Commerce*, vol. 13, no. 1, pp. 1–29, 2006.
- [30] M. Berkelaar, K. Eikland, and P. Notebaert, "Ipsolve: Open source (mixed-integer) linear programming system," 2004, <http://lpsolve.sourceforge.net/>.