2014

Multi-objective ant colony system for data-intensive service provision

Lijuan Wang
University of Wollongong, lw840@uowmail.edu.au

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

Junzhou Luo
Southeast University

Publication Details

Wang, L., Shen, J. & Luo, J. (2014). Multi-objective ant colony system for data-intensive service provision. International Conference on Advanced Cloud and Big Data (pp. 45-52). United States: Institute of Electrical and Electronics Engineers.

Multi-objective ant colony system for data-intensive service provision

Abstract
Data-intensive services have become one of the most challenging applications in cloud computing. The classical service composition problem will face new challenges as the services and correspondent data grow. A typical environment is the large scale scientific project AMS, which we are processing huge amount of data streams. In this paper, we will resolve service composition problem by considering the multi-objective data-intensive features. We propose to apply ant colony optimization algorithms and implemented them with simulated workflows in different scenarios. To evaluate the proposed algorithm, we compared it with a multi-objective genetic algorithm with respect to five performance metrics

Keywords
ant, objective, multi, service, provision, intensive, data, system, colony

Disciplines
Engineering | Science and Technology Studies

Publication Details
Wang, L., Shen, J. & Luo, J. (2014). Multi-objective ant colony system for data-intensive service provision. International Conference on Advanced Cloud and Big Data (pp. 45-52). United States: Institute of Electrical and Electronics Engineers.

Multi-Objective Ant Colony System for Data-Intensive Service Provision

Lijuan Wang, Jun Shen
School of Information Systems and Technology
University of Wollongong, Wollongong, NSW, Australia
Email: lw840@uowmail.edu.au, jshen@uow.edu.au

Junzhou Luo
School of Computer Science and Engineering
Southeast University, Nanjing, P. R. China
Email:jluo@seu.edu.cn

***Abstract*—Data-intensive services have become one of the most challenging applications in cloud computing. The classical service composition problem will face new challenges as the services and correspondent data grow. A typical environment is the large scale scientific project AMS, which we are processing huge amount of data streams. In this paper, we will resolve service composition problem by considering the multi-objective data-intensive features. We propose to apply ant colony optimization algorithms and implemented them with simulated workflows in different scenarios. To evaluate the proposed algorithm, we compared it with a multi-objective genetic algorithm with respect to five performance metrics.**

***Index Terms*—data-intensive service composition, ant colony system, genetic algorithm.**

I. Introduction

Big Data has attracted much research attention. The Gartner Group listed Big Data in the "10 Critical Tech Trends for the Next Five Years" [1]. Data-intensive science is emerging as the fourth scientific paradigm, and new techniques and technologies for the new scientific paradigm are needed [2]. As a result, applications based on data-intensive services have become one of the most challenging applications in service oriented computing and cloud computing. A survey of the challenges, techniques, and technologies of data-intensive applications was presented in [3]. The scope, number, and complexity of data-intensive services are all set to soar in the future. On the one hand, Big Data provides opportunities and potential values. On the other hand, many challenges are arising with respect to the data capture, data storage, data analysis, data searching, data sharing, and data visualization [3]. The service provision, and in particular service composition, will face new challenges such as autonomy, scalability, adaptability, and robustness. Indeed, new mechanisms are needed to overcome those issues. In the following, we will briefly introduce the application of the Big Data problem in scientific research fields.

One of the motivations of our work is the Alpha Magnetic Spectrometer (AMS) experiment, which uses cloud computing to process a huge amount of data. The AMS, also designated AMS-02, is a particle physics experiment module that is mounted on the International Space Station. The purpose of the AMS experiment is to advance knowledge of the universe and lead to the understanding of its origin by searching for antimatter and dark matter while performing precision measurements of cosmic rays composition and flux. The AMS-02 SOC (Science Operation Center) at Southeast University in China (labeled as AMS-02 SOC@SEU) is supported by the IBM-sponsored Cloud Computing Center with 3500 CPU core and 500TB storage. The AMS-02 SOC@SEU typically receives 200GB of data from AMS and generates 700GB of data after processing them, on each day. Scientists and remote users deploy different processes, such as data mining, image processing, thematic map generation, or data query on a large amount of data at AMS-02 SOC. A set of operations is often necessary to provide an appropriate solution to complex scientific applications. The use of Web services technologies provides valuable solutions to speed up the scientific data analysis [4], [5]. A composition of a set of services as a composite service can be reused by other researchers.

The authors of [3] explained that bio-inspired computing was one of the underlying techniques to solve data-intensive problems. The authors stated that biological computing models were better appropriate for Big Data because they had mechanisms with high-efficiency to organize, access, and process data. The authors of [6] already proved that it was useful for service management and discovery to add biological mechanisms to services. One of our earlier studies has presented a hierarchical taxonomy of Web service composition approaches [7]. By analyzing each type of approaches with respect to their optimality, their computational efficiency, and their dynamic complexity, we observed that bio-inspired algorithms, belonging to the sub-optimal methods, could overcome the new challenging requirements of the data-intensive service provision problem. Then we conducted a systematic review of Web service composition and selection based on three bio-inspired algorithms, namely, the ant colony optimization algorithms, the genetic algorithms, and the particle swarm optimization algorithms [7].

The ant colony optimization algorithms are inspired by the foraging behavior of ant colonies, in which a set of artificial ants cooperate to find a solution of a problem by exchanging information via pheromone deposited on a graph edges. The ant colony optimization algorithms iteratively performs a loop constitutes the ants' solution construction and the pheromone update. ACS is an algorithm inspired by the ant system but differs from it in three main aspects [8]. First, the state transition rule provides a way to balance between the exploration of new edges and the exploitation of accumulated knowledge about the problem. Second, a local updating rule is applied while ants construct a path. Third, the global updating rule is applied only to edges which belong to the best ant path. In this paper, we will propose a multi-objective ant colony system (MOACS) to solve the data-intensive service composition problem.

The remainder of this paper is organized as follows. Section II introduces background. Section III investigates how a MOACS could be used to solve the problem. Section IV presents experimental results and analysis. Section V reviews related work. Finally, section VI concludes this paper and proposes future work.

II. Background

A. Pareto-optimal Solutions

The goal of the majority of existing multi-objective optimization algorithms is to find Pareto-optimal solutions. The concept of dominance is used to relate the solutions found in these algorithms.

Definition 2.1 (Dominance): In a minimization problem for all objectives, a solution x_1 dominates another solution x_2 (denotes as $x_1 \prec x_2$) if and only if the two following conditions are true: 1) x_1 is no worse than x_2 in all objectives, namely $F_i(x_1) \leq F_i(x_2)$ ($\forall i \in \{1, 2, \ldots, N\}$, N is the number of objective functions), and 2) x_1 is strictly better than x_2 in at least one objective, namely $F_j(x_1) < F_j(x_2)$ ($\exists j \in \{1, 2, \ldots, N\}$).

Definition 2.2 (Cover): In a minimization problem for all objectives, a solution x_1 is said to cover another solution x_2 (denotes as $x_1 \preceq x_2$) if one of the two following conditions is true: 1) x_1 dominates x_2, namely $x_1 \prec x_2$, or 2) x_1 is equal to x_2 in all objectives, namely $F_i(x_1) = F_i(x_2)$ ($\forall i \in \{1, 2, \ldots, N\}$).

Definition 2.3 (Non-dominated set): Among a set of solutions, the non-dominated set of solutions are those that are not dominated by any member of the set.

A solution is said to be Pareto-optimal if it is not dominated by any other possible solutions. Thus, the Pareto-optimal solutions to a multi-objective optimization problem form the Pareto front or Pareto-optimal set [9]. Pareto-optimal sets are the solutions that cannot be improved in one objective function without deteriorating their performance in at least one of the remaining objective functions.

B. Performance Metrics

There are two goals in a multi-objective optimization: the convergence to the Pareto-optimal set and the maintenance of diversity in solutions of the Pareto-optimal set [10]. These two goals cannot be measured adequately with one single performance metric. Meanwhile, the outcome of a multi-objective optimization run will generally consist of a varying number of non-dominated solutions. Various performance metrics to measure these two goals have been suggested [11]–[14]. Here, we chose the following five performance metrics: 1) the computation time, 2) the overall non-dominated vector generation (ONVG), 3) the comparison metric (C metric), 4) the size of the dominated space, and 5) the summary attainment surface. The first four metrics measure the convergence of the Pareto-optimal solutions, while the fifth metric measures the distribution of the Pareto-optimal set.

1) The Computation Time: The computation time, also called running time, is the length of time required to perform the algorithm.

2) The ONVG Metric: The ONVG metric measures the number of distinct non-dominated solutions in the calculated Pareto-optimal set GP. The larger the value of the ONVG, the more we know about the details of the Pareto-optimal set.

3) The C Metric: The C metric is based on comparing a pair of non-dominated sets by computing the fraction of each set that is covered by the other. C maps the ordered pair (A, B) into the interval $[0, 1]$:

$$C(A, B) = \frac{|\{b \in B, \exists a \in A : a \preceq b\}|}{|B|} \tag{1}$$

where $|B|$ means the number of solutions in the non-dominated set B, and $a \preceq b$ means a covers b. It is important to note that both $C(A, B)$ and $C(B, A)$ have to be considered, since $C(A, B)$ is not necessarily equal to $1 - C(B, A)$. $C(A, B) > C(B, A)$ means that the non-dominated set A has better solutions than B.

4) The Size of the Dominated Space: The size of the dominated space $S(A)$ indicates how well the Pareto-optimal set is approximated by the non-dominated set A of the algorithm [14]. For each non-dominated solution in A, we can compute the values of all objective functions. These values comprise a point in the solution space. Fig. 1 illustrates an example of a dominated space. The greater the size of the space dominated by the non-dominated

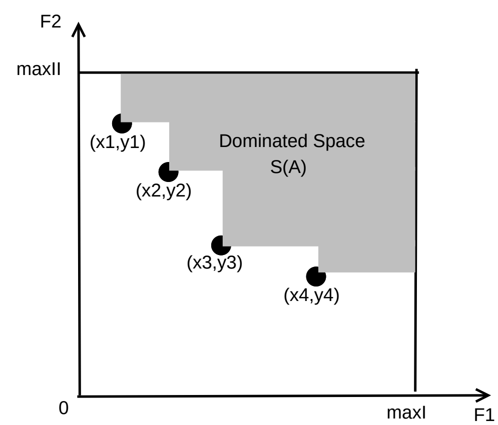

Fig. 1: A space dominated by a non-dominated set

set is, the closer the solutions are to the Pareto-optimal set. Here we use $PS(A)$ to indicate this metric, where $PS(A) = \frac{S(A)}{maxI * maxII} * 100\%$.

5) The Summary Attainment Surface: The summary attainment surface is a visual approach to summarizing a number of runs of a multi-objective optimizer. The authors of [13] proposed an algorithm to plot approximate summary attainment surfaces with any number of objectives, and he suggested that it was more useful to plot the median summary attainment surface to compare the performance of the optimizers. For a two-objective problem, the more the points of the median attainment surface of an algorithm close to the origin of the rectangular coordinate system, the better the solutions of the algorithm are.

III. Multi-Objective Data-Intensive Service Composition Based on MOACS

The data-intensive service composition problem is modeled as a graph with a $start$ vertex and an end vertex. The $start$ vertex is set as the ants' nest and the end vertex is set as the food source. The feasible solutions to the composition problem correspond to paths through the graph. In the graph, all ants are initially positioned at the $start$ vertex and the task of each ant is to find a path from the $start$ vertex to the end vertex. Fig. 2 presents an example of a graph in which data sets, as the inputs and outputs of services, are incorporated.

The proposed MOACS uses a unique ant colony to simultaneously minimize all functions. All objectives share the same pheromone trails. In every iteration, an ant k ($\forall k \in \{1, 2, \ldots, Nants\}$, $Nants$ is the number of ants) constructs one feasible solution, beginning with the $start$ vertex and successively choosing the next vertex from the set of feasible vertices N_i^k (where subindex i represents that ant k is at vertex s_i, s_i is the concrete service which is chosen to implement abstract service AS_i). N_i^k is the neighborhood of vertex s_i when ant k is at it, which includes all direct successors of s_i. For each ant k, a solution is found until it arrives at the end vertex. The key to MOACS is how to determine the state transition rule, the local updating rule, and the global updating rule.

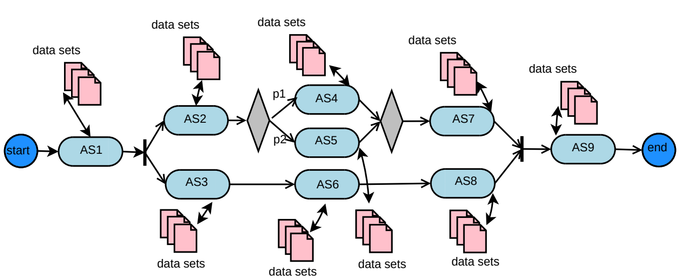

Fig. 2: A graph for data-intensive service composition

A. State Transition Rule

When ant k arrives at vertex s_i, it will choose successor s_j to move to by applying the rule given by (2).

$$j = \begin{cases} \arg\max\limits_{j \in N_i^k} \{ \left[\tau_{ij}\right] \left[[\eta_{ij}^C]^\lambda [\eta_{ij}^T]^{(1-\lambda)} \right]^\beta \}, & \text{if } q \le q_0; \\ \text{randomly selected from } N_i^k, & \text{otherwise.} \end{cases} \tag{2}$$

The variable q is a random variable uniformly distributed in $[0, 1]$, q_0 $(0 \le q_0 \le 1)$ is a parameter, λ is the weight of each objective ($\lambda = k/Nants$), β weights the relative importance of each objective, and τ_{ij} is the pheromone density on edge (s_i, s_j). The variable η_{ij}^C is the heuristic information for the objective function considering service cost, and η_{ij}^T is the heuristic information for the objective function considering service execution time. If $q > q_0$, then j is randomly selected from N_i^k according to the probability distribution given by (3).

$$p_{ij}^k = \begin{cases} \dfrac{\left[\tau_{ij}\right] \left[[\eta_{ij}^C]^\lambda [\eta_{ij}^T]^{(1-\lambda)} \right]^\beta}{\sum\limits_{j \in N_i^k} \left[\tau_{ij}\right] \left[[\eta_{ij}^C]^\lambda [\eta_{ij}^T]^{(1-\lambda)} \right]^\beta}, & \text{if } j \in N_i^k \ ; \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The heuristic information considering service cost is calculated according to $\eta_{ij}^C = 1/Cost(s_j)$. The heuristic information considering service execution time is calculated according to $\eta_{ij}^T = 1/T_{et}(s_j)$. The variables $Cost(s_j)$ and $T_{et}(s_j)$ were described in our earlier study [15].

B. Local Updating Rule

When building a solution of the service composition problem, namely an executed path through the graph, the ants use a local pheromone updating rule that they apply immediately after having crossed an edge (s_i, s_j), which is shown by (4).

$$\tau_{ij} = (1 - \xi)\tau_{ij} + \xi\tau_0 \tag{4}$$

The variable ξ $(0 < \xi < 1)$ is a parameter and τ_0 is initially calculated by $\tau_0 = 1/\big(NN * C(S_0) * T(S_0)\big)$, with NN is the number of nodes in the graph, S_0 is the solution generated by the nearest neighbor heuristic [8]. This is due to the fact that it is a good practice to set the initial pheromone concentration to a value that is slightly higher than the expected amount of pheromone deposited by the ants. The variable $C(S_0)$ represents the overall cost of the solution S_0, and $T(S_0)$ represents the overall execution time of S_0.

C. Global Updating Rule

The global non-dominated set, which the ants found from the beginning of the trial, is stored in the Pareto-optimal set GP. In each iteration, the solution found by each ant is recorded to a set P. After all ants arrive at the end vertex, the local non-dominated set LP is found from P. Each solution in LP is compared with the solutions in GP in order to check if it is non-dominated. If it is a new

Algorithm 1 Multi-objective data-intensive service composition based on MOACS

```
step = 0; //iteration counter
Initialization; //MaxIt is the maximum number of iteration
while 1 do
    step = step + 1;
    P = ∅; // The solutions found by ants for each iteration
    set all ants at the start vertex;
    for each ant k do
        while ant k is not at the end vertex do
            construct a solution using (2) and (3);
            record the solution to P;
            apply the local updating rule (4);
        end while
    end for
    when all ants arrive at the end vertex, find the non-dominated set LP from P;
    update the global non-dominated set GP;
    apply the global updating rule (5) to GP;
    if step > MaxIt then
        break;
    end if
end while
output all solutions in GP.
```

Pareto-optimal solution, it is added to GP and all solutions dominated by the added one are erased from GP. Since all non-dominated solutions are considered as best solutions for a multi-objective optimization problem, we suppose that all non-dominated solutions have the same quality. Therefore, for each solution $\psi^{GP} \in GP$, the pheromone information is globally updated according to (5).

$$\tau_{ij} = (1-\rho)\tau_{ij} + \rho / \big(C(\psi^{GP}) * T(\psi^{GP})\big), \quad \forall (i,j) \in \psi^{GP} \tag{5}$$

The variable ρ $(0 < \rho < 1)$ is the pheromone evaporation rate, $C(\psi^{GP})$ is the cost of a given solution ψ^{GP} while $T(\psi^{GP})$ is the execution time of ψ^{GP}. The implementation of our MOACS is given in Algorithm 1.

IV. Experiments and analysis

In order to verify our proposed MOACS, we compared with a multi-objective genetic algorithm (MOGA). The MOGA is based on the improved version of the non-dominated sorting genetic algorithm (NSGA-II) [10], but we need to customize and modify the NSGA-II in order to handle our problem. In our experiments, a trial testing method is adopted to determine most suitable values for all parameters of MOACS and MOGA. Finally, the parameters of MOACS are: $\beta = 2$, $q_0 = 0.9$, $\rho = 0.1$, $\xi = 0.1$, and $Nants = 100$. The parameters of MOGA are: the population is 100, the crossover probability is 0.9, and the mutation probability is $1/n$ (n is the number of abstract services in a composite service).

A. Test Case Generation

For the purpose of our evaluation, we considered different scenarios where a composite application comprises services from n abstract services, n varies in our experiments between 10 and 50, in increments of 10. There are m concrete services in each service candidate set, m varies in our experiments between 10 and 100, in increments of 10. Each abstract service requires a set of k data sets, k is fixed at 10 in our experiments. 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 service endpoint, 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]. Then every scenario was performed with 21 runs (with 11 being the median line of all 21 attainment surfaces), and every run was stopped after 300 generations. All runs of the same scenario use the same data, and the average results over 21 independent runs are reported.

B. Result Analysis

Table I shows the means of the computation time of each scenario. The upper half of the table indicates that when the number of concrete services increases, the MOACS needs more computation time while the computation time of MOGA remains almost steady. This is because, by using the integer array coding scheme, the change in the number of concrete services will not influence the length of the genome. The lower half of the table indicates that the computation time of both MOACS and MOGA increases when the number of abstract services increases. When the number of abstract services and concrete services is small, MOACS is better than MOGA since the means of the computation time of MOACS are lower than those of MOGA except in the scenario where $n = 10$ and $m = 100$. When there is a large number of concrete services and abstract services, MOGA is more scalable than MOACS.

Table II gives the means of ONVG. The upper half of the table shows that MOGA can get more non-dominated solutions than MOACS except in the scenario where $n = 10$ and $m = 10$. On the other hand, the lower half of the table indicates that MOACS can find more non-dominated solutions than MOGA when the number of abstract services increases except in the scenario where $n = 10$ and $m = 50$.

Table III gives the means of the C metric, where the value in the second column is equal to the value in the third column. The results indicate that the convergence of the Pareto-optimal solutions of MOACS and MOGA is no different, so we cannot say one is better than the other.

Table IV shows the means of $PS(A)$. From this table, we conclude that MOACS is slightly better than MOGA since MOACS always leads to a higher value of $PS(A)$.

Fig. 3 gives an example of the median summary attainment surface of MOACS and MOGA with respect to the varying number of concrete services, and Figs. 4-8 show the median summary attainment surface of both algorithms with respect to the varying number of abstract services. The regions where no difference between the points of the median attainment surfaces of the two algorithms could be found were indicated in gray dots, whereas those regions where the points of the two surfaces were found to differ from each other are plotted in stars and squares, respectively. In the regions where the points of the two surfaces were found to differ from each other, there are three situations: 1) if the points of the median attainment surfaces of MOACS dominate those of MOGA, then the label MOACS is put near the points, 2) if the points of the median attainment surfaces of MOGA dominate those of MOACS, then the label MOGA is put near the points, 3) if the points of the median attainment surfaces of MOACS are not dominated by those of MOGA and the points of the median attainment surfaces of MOGA are not dominated by those of MOACS, then no label is put.

When the number of concrete services increases, MOACS is better than MOGA except for a small number of points, since the points of the median attainment surface of MOACS are closer to the origin of the rectangular coordinate system. When the number of abstract services increases, both MOACS and MOGA have some points where there is no difference between them in the scenario where $n = 10$ and $m = 50$. In the scenario where $n = 20$

TABLE I: Means of Computation Time

Scenarios	*MOACS*	*MOGA*
n is fixed at 10, m varies between 10 and 100, in increments of 10	22.3333	29.6667
	23.6190	30.0952
	24.4762	29.8571
	25.0952	31.3333
	26.5238	29.9524
	27.1905	31.2857
	27.6667	31.1429
	29.5238	30.9048
	30.6667	30.7143
	30.4286	30.2381
m is fixed at 50, n varies between 10 and 50, in increments of 10	26.5238	29.9524
	61.2381	30.7143
	98.9048	30.2381
	141.9524	31.1905
	285.5714	31.7143

TABLE II: Means of ONVG

Scenarios	*MOACS*	*MOGA*
n is fixed at 10, m varies between 10 and 100, in increments of 10	14.9048	14.8571
	32	36.0476
	27.4762	34.4286
	22.0952	27.5714
	35.8571	49.8045
	25.3333	28.0952
	20.1905	27.4286
	38.5238	46.0476
	34.8095	38.3333
	15.3333	16.5714
m is fixed at 50, n varies between 10 and 50, in increments of 10	35.8571	49.8045
	75.4762	69.6196
	90.7619	83.1905
	114.9524	82.3810
	131.5714	84.1429

TABLE III: Means of C Metric

Scenarios	*C(MOACS,MOGA)*	*C(MOGA,MOACS)*
n is fixed at 10, m varies between 10 and 100, in increments of 10	0.9524	0.9524
	0.6298	0.6298
	0.6037	0.6037
	0.8429	0.8429
	0.8311	0.8311
	0.5982	0.5982
	0.6219	0.6219
	0.5127	0.5127
	0.4523	0.4523
	0.5189	0.5189
m is fixed at 50, n varies between 10 and 50, in increments of 10	0.8311	0.8311
	0.4530	0.4530
	0.4452	0.4452
	0.2338	0.2338
	0.2094	0.2094

TABLE IV: Means of $PS(A)$

Scenarios	*MOACS*	*MOGA*
n is fixed at 10, m varies between 10 and 100, in increments of 10	82.65%	82.64%
	84.86%	84.77%
	85.29%	85.25%
	84.65%	84.61%
	85.88%	85.87%
	86.30%	86.12%
	86.17%	86.02%
	86.12%	85.92%
	86.72%	86.38%
	86.33%	86.28%
m is fixed at 50, n varies between 10 and 50, in increments of 10	85.88%	85.87%
	68.96%	68.31%
	51.77%	49.81%
	38.01%	35.96%
	24.74%	21.90%

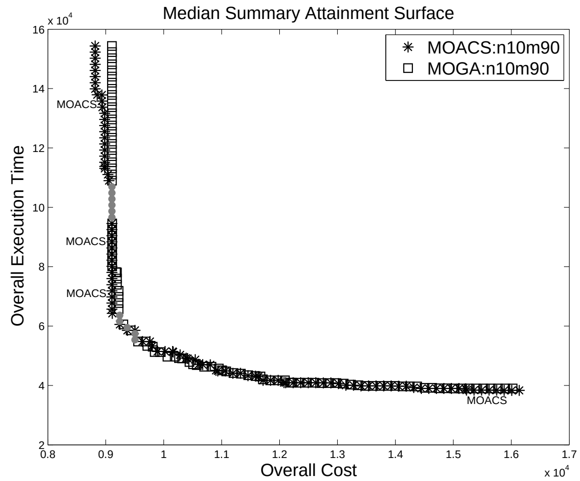

Fig. 3: Median summary attainment surface with $(n, m) = (10, 90)$

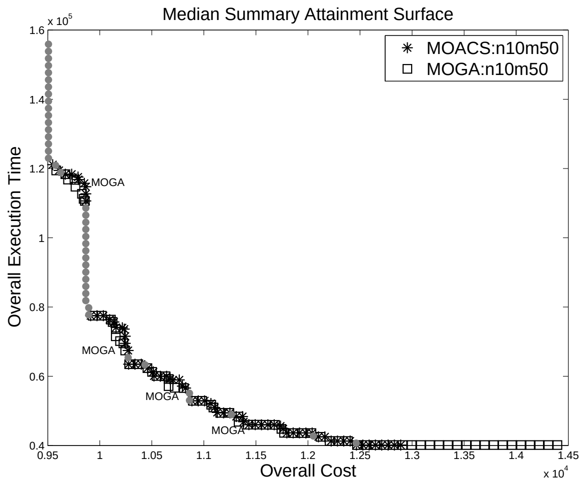

Fig. 4: Median summary attainment surface with $(n, m) = (10, 50)$

and $m = 50$, there are some points where MOGA is better than MOACS. In the remaining scenarios, there is no domination relationship between the points of the median attainment surfaces of the two algorithms, but it is clear that MOGA gives more useful solutions than MOACS which is indicated by the middle parts of the median attainment surfaces of both algorithms.

V. Related work

We have been applying bio-inspired algorithms to tackle the service composition problems [15]–[24]. We investigated how the ant colony optimization was used for peer selection in service composition [16]–[18]. We also presented a survey on bio-inspired algorithms for Web service composition [19]. In [20], we presented an ant colony system (ACS) to solve the single-objective data-intensive service composition problem. In [15], [21], we proposed genetic algorithms to solve the single-objective data-intensive service composition problem. Comparisons with the mixed integer programming (MIP) method and the random selection approaches showed that our approach had better scalability and effectiveness. In [22], we designed a data replica selection algorithm based on ACS.

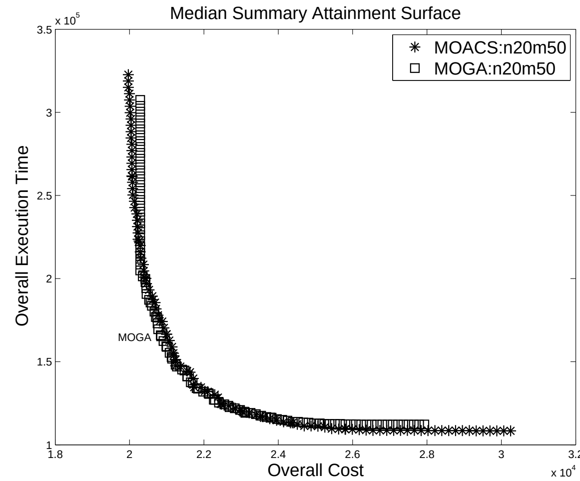

Fig. 5: Median summary attainment surface with $(n, m) = (20, 50)$

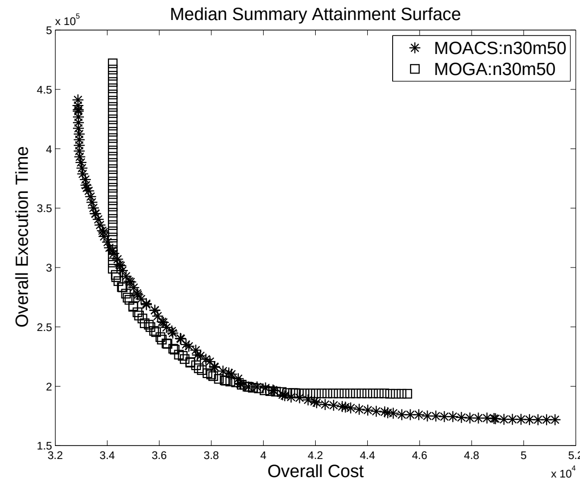

Fig. 6: Median summary attainment surface with $(n, m) = (30, 50)$

In [23], we designed four strategies to modify the pheromone information in order to adapt ACS to handle the dynamic scenarios where new services were provided, some services were discontinued, or the QoS attributes of some services were changed. To evaluate the dynamic ACS, we compared the optimization behavior of each strategy with respect to different strategy-specific parameters. We also recorded the loss in the quality of the best solution and compared it with the solution found by the MIP approach.

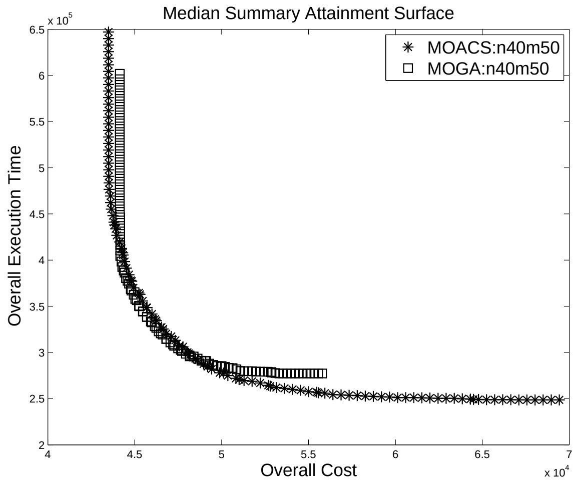

Fig. 7: Median summary attainment surface with $(n, m) = (40, 50)$

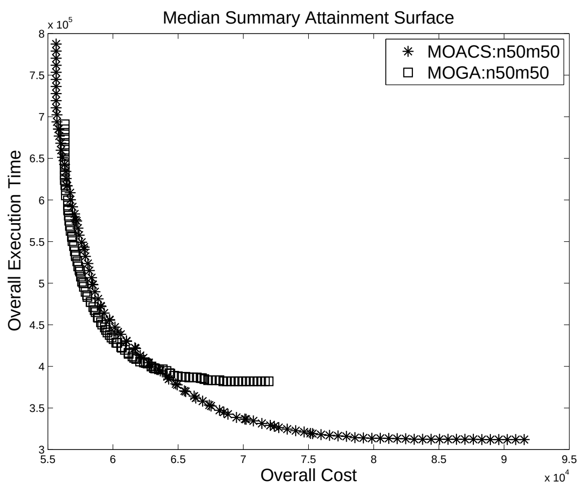

Fig. 8: Median summary attainment surface with $(n, m) = (50, 50)$

The experimental results showed that the performance of each pheromone modification strategy depended not only on its strategy-specific parameter, but also on the number of changes and the frequency of occurrence of changes. Then we proposed an ant-inspired negotiation approach based on the above dynamic ACS [24]. A multi-phase, multi-party negotiation protocol was also designed. The performance of our negotiation approach was compared with the MIP approach. The experimental results showed that our negotiation-based approach, compared with the traditional non-dynamic method, could facilitate the data-intensive service provision with a better outcome.

Many studies have applied multi-objective ant colony algorithms to solve service composition problems. Some studies only considered sequential composition structure [25]–[28], and some studies did not consider the global quality of service (QoS) constraints [25], [27]–[30]. Only single solutions were given by other studies [29], [31], [32]. In [28], the ant system was applied to solve the multi-objective optimization problem, but we applied the ant colony system. We adopted five performance metrics to measure the Pareto-optimal solutions, which is different from all the other studies. This paper also presents a comprehensive comparison of MOACS with a MOGA, which is lacking in other studies [25]–[32]. To the best of our knowledge, our study is the first application of the ACS meta-heuristic to the data-intensive service composition problem with global QoS constraints, where both overall cost and execution time need to be minimized.

VI. Conclusion

Data-intensive service provision faces new challenges with the rapid proliferation of services and the development of cloud computing. The outcomes of our earlier studies confirmed the applicability and efficiency of bio-inspired algorithms to solve data-intensive service provision problems. This paper proposed a new multi-objective ant colony system. The goal was to efficiently obtain a set of non-dominated solutions that simultaneously minimized the total cost and the total execution time. In order to verify the performance of our algorithm, we compared it with a multi-objective genetic algorithm. Both algorithms were simulated on many different scenarios with respect to five performance metrics. The lessons learned from our experimental results are that when we have a large number of concrete services available for each abstract service, a multi-objective genetic algorithm can achieve better solutions in a reasonable time. On the other hand, whenever the number of concrete services available is small, such as in some simple and repetitive scientific computation, a multi-objective ant colony system is to be preferred to a multi-objective genetic algorithm. In our evaluations, we experimented with synthetic datasets without loss of generality.

Acknowledgment

This work is supported in part by National Natural Science Foundation of China under Grants No. 61320106007 and No. 61202449, China National High Technology Research and Development Program under Grants No. 2013AA013503, China Specialized Research Fund for the Doctoral Program of Higher Education under Grants No. 20110092130002, Jiangsu Provincial Key Laboratory of Network and Information Security under Grants No. BM2003201, and Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under Grants No. 93K-9.

References

[1] E. Savitz, "Gartner: 10 Critical Tech Trends for the Next Five Years," 2012, http://www.forbes.com/sites/ericsavitz/2012/10/22/gartner-10-critical-tech-trends-for-the-next-five-years/.

[2] G. Bell, T. Hey, and A. Szalay, "Beyond the Data Deluge," *Science*, vol. 323, no. 5919, pp. 1297–1298, 2009.

[3] C. L. Philip Chen and C. Y. Zhang, "Data-Intensive Applications, Challenges, Techniques and Technologies: A Survey on Big Data," *Information Sciences*, vol. 275, pp. 314–347, 2014.

[4] W. A. Watson, I. Bird, J. Chen, B. Hess, A. Kowalski, and Y. Chen, "A Web Services Data Analysis Grid," *Concurrency and Computation: Practice and Experience*, vol. 14, pp. 1303–1311, 2002.

[5] A. Woolf, K. Haines, and C. Liu, "A Web Service Model for Climate Data Access on the Grid," *International Journal of High Performance Computing Applications*, vol. 17, no. 3, pp. 281–295, 2003.

[6] S. Balasubramaniam, D. Botvich, R. Carroll, J. Mineraud, T. Nakano, T. Suda, and W. Donnelly, "Biologically Inspired Future Service Environment," *Computer Networks*, vol. 55, no. 15, pp. 3423–3440, 2011.

[7] L. Wang and J. Shen, "Multi-Phase Ant Colony System for Multi-Party Data-Intensive Service Provision," *IEEE Transactions on Services Computing, in press, to appear*, 2014.

[8] M. Dorigo and T. Stutzle, *Ant Colony Optimization*. Cambridge, MA, USA: MIT Press, 2004.

[9] H. A. Taboada, J. F. Espiritu, and D. W. Coit, "MOMS-GA: A Multi-Objective Multi-State Genetic Algorithm for System Reliability Optimization Design Problems," *IEEE Transactions on Reliability*, vol. 57, no. 1, pp. 182–191, 2008.

[10] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[11] B. Baran and M. Schaerer, "A Multiobjective Ant Colony System for Vehicle Routing Problem with Time Windows," in *Proceedings of 21st IASTED International Conference on Applied Informatics*. Zurich, Switzerland: IASTED, 2003, pp. 97–102.

[12] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, U. K.: John Wiley & Sons, 2001.

[13] J. Knowles, "A Summary-Attainment-Surface Plotting Method for Visualizing the Performance of Stochastic Multiobjective Optimizers," in *Proceedings of 5th International Conference on Intelligent Systems Design and Applications (ISDA '05)*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 552–557.

[14] E. Zitzler and L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[15] L. Wang, J. Shen, J. Luo, and F. Dong, "An Improved Genetic Algorithm for Cost-Effective Data-Intensive Service Composition," in *Proceedings of The 9th International Conference on Semantics, Knowledge & Grids (SKG)*. Washington, DC, USA: IEEE Computer Society, 2013, pp. 105–112.

[16] 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*. Washington, DC, USA: IEEE Computer Society, 2011, pp. 250–257.

[17] J. Shen and S. Yuan, "QoS-Aware Peer Services Selection Using Ant Colony Optimisation," in *Business Information Systems Workshops*, ser. Lecture Notes in Business Information Processing, W. Abramowicz and D. Flejter, Eds. Berlin, Heidelberg: Springer-Verlag, 2009, vol. 37, pp. 362–374.

[18] S. Yuan, J. Shen, and A. Krishna, "Ant Inspired Scalable Peer Selection in Ontology-based Service Composition," in *IEEE World Congress on Services*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 95–102.

[19] L. Wang, J. Shen, and J. Yong, "A survey on bio-inspired algorithms for web service composition," in *IEEE 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. Washington, DC, USA: IEEE Computer Society, 2012, pp. 569–574.

[20] L. Wang, J. Shen, and G. Beydoun, "Enhanced Ant Colony Algorithm for Cost-Aware Data-Intensive Service Provision," in *IEEE 9th World Congress on Services*. Washington, DC, USA: IEEE Computer Society, 2013, pp. 227–234.

[21] 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.

[22] 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*. Washington, DC, USA: IEEE Computer Society, 2013, pp. 254–261.

[23] L. Wang, J. Shen, and J. Luo, "Impacts of Pheromone Modification Strategies in Ant Colony for Data-Intensive Service Provision," in *Proceedings of the IEEE 21th International Conference on Web Services (ICWS)*. Washington, DC, USA: IEEE Computer Society, 2014, pp. 177–184.

[24] L. Wang, J. Shen, Q. Zhou, and G. Beydoun, "Ant-Inspired Multi-Phase and Multi-Party Negotiation in the Data-Intensive Service Provision," in *Proceedings of the IEEE 11th International Conference on Services Computing (SCC)*. Washington, DC, USA: IEEE Computer Society, 2014, pp. 211–218.

[25] D. B. Claro, P. Albers, and J. Hao, "Selecting Web Services for Optimal Composition," in *Proceedings of 2nd International Workshop Semantic and Dynamic Web Processes (SDWP 2005)*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 32–45.

[26] L. Li, P. Yang, L. Ou, Z. Zhang, and P. Cheng, "Genetic Algorithm-based Multi-Objective Optimization for QoS-aware Web Services Composition," in *Knowledge Science, Engineering and Management*, ser. Lecture Notes in Computer Science, Y. Bi and M. Williams, Eds. Berlin, Heidelberg: Springer-Verlag, 2010, vol. 6291, pp. 549–554.

[27] S. Liu, Y. Liu, N. Jing, G. Tang, and Y. Tang, "A Dynamic Web Service Selection Strategy with QoS Global Optimization Based on Multi-objective Genetic Algorithm," in *Grid and Cooperative Computing - GCC 2005*, ser. Lecture Notes in Computer Science, H. Zhuge and G. C. Fox, Eds. Berlin, Heidelberg: Springer-Verlag, 2005, vol. 3795, pp. 84–89.

[28] W. Zhang, C. K. Chang, T. Feng, and H. Y. Jiang, "QoS-Based Dynamic Web Service Composition with Ant Colony Optimization," in *Proceedings of the 34th Annual IEEE International Computer Software and Applications Conference (COMPSAC '10)*. Washington, DC, USA: IEEE Computer Society, 2010, pp. 493–502.

[29] W. Chang, C. Wu, and C. Chang, "Optimizing Dynamic Web Service Component Composition by Using Evolutionary Algorithms," in *IEEE/WIC/ACM International Conference on Web Intelligence*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 708–711.

[30] Y. Yao and H. Chen, "QoS-aware Service Composition using NSGA-II1," in *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*. New York, NY, USA: ACM, 2009, pp. 358–363.

[31] Y. Xia, J. Chen, and X. Meng, "On the Dynamic Ant Colony Algorithm Optimization Based on Multi-Pheromones," in *7th IEEE/ACIS International Conference on Computer and Information Science (ICIS '08)*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 630–635.

[32] X. Zheng, J. Luo, and A. Song, "Ant Colony System Based Algorithm for QoS-Aware Web Service Selection," in *Proceedings of 4th International Conference on Grid Service Engineering and Management (GSEM)*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 39–50.