4th IEEE Conference on Automation Science and Engineering
Key Bridge Marriott, Washington DC, USA
August 23-26, 2008

Efficient Algorithms for Combinatorial Auctions with Volume Discounts Arising in Web Service Composition

B Prashanth and Y Narahari
Computer Science and Automation,
Indian Institute of Science,
Bangalore, India
prashanthb,hari@csa.iisc.ernet.in

Abstract— **Web services are now a key ingredient of software services offered by software enterprises. Many standardized web services are now available as commodity offerings from web service providers. An important problem for a web service requester is the web service composition problem which involves selecting the right mix of web service offerings to execute an end-to-end business process. Web service offerings are now available in bundled form as composite web services and more recently, volume discounts are also on offer, based on the number of executions of web services requested. In this paper, we develop efficient algorithms for the web service composition problem in the presence of composite web service offerings and volume discounts. We model this problem as a combinatorial auction with volume discounts. We first develop efficient polynomial time algorithms when the end-to-end service involves a linear workflow of web services. Next we develop efficient polynomial time algorithms when the end-to-end service involves a tree workflow of web services.**

I. INTRODUCTION

Web services are emerging as an indispensable technology in recent times because of their reusability, interoperability and loosely coupledness. All this is possible because it is an implementation of well designed architecture called Service Oriented Architecture (SOA). The whole system of SOA is viewed as a collection of services, where a service can be thought as a functionality provided of a component for use by another component. In SOA each component may play any of the three roles: service-broker, service-requester, service-provider:

1) Service Broker: A service-broker acts as a look up service between a service provider and a service requester. Service broker is also called as Registry.

2) Service Provider: A service-provider publishes its services to the service broker.

3) Service Requester: A service-requester asks the service broker where to find a suitable service provider and then binds itself to the provider.

In the case of web services, each Web Service Providers (WSP) publishes and describes all the services he provides in a registry called Universal Description, Discovery, and Integration (UDDI). A Web Service Requester (WSR) uses the UDDI inquiry API to search the UDDI for a suitable service provider. UDDI is a standard designed to provide a searchable directory of Web Services using XML, and also acts as broker between WSP and WSR.

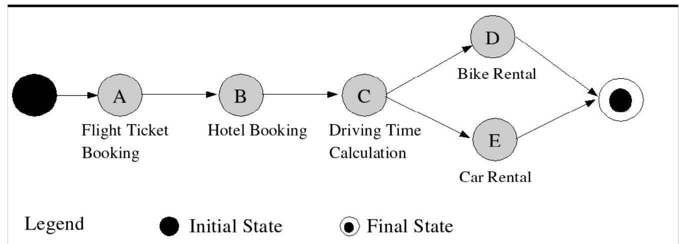

Fig. 1. Statechart of "Travel Planner" composite service

A. Web Service Composition Problem

Usually individual web services cannot satisfy WSR's requirements and generally a WSR asks for a composite service. A composite service (interchangeably called as end-to-end composite service) is a set of stand-alone web services with some dependencies between themselves (Figure 1). It can be easily seen that a composite service is a directed acyclic graph (DAG), where nodes in the graph are services and edges represent dependencies.

Typically there are multiple WSPs who can offer a service at different prices, for example in Figure 1 there can be many number of service providers who can offer Flight Booking Service. This leads the WSR to have different ways of procuring a composite service, but he needs to select one set of WSPs which can collectively provide the composite service at the lowest price. The process of such selection is called web service composition problem. In order to solve this problem WSR needs to conduct a procurement auction and seek the bids from all the WSPs who can provide the services.

B. Motivation for Combinatorial and Volume Discount Auctions in Web Service Composition

There are several advantages [1] to assume that WSPs can not only provide stand-alone services (like Flight Booking service) but also composite services. For example there can be a WSP who can provide Flight and Hotel booking service combindly at lower price than the sum of individual services, or a WSP may have a service which does both bookings together (like a Holiday Package) but he may not be able to provide individual services. By considering these facts into account in the auction, not only bids on the stand-alone services are accepted from WSPs but also bids on the

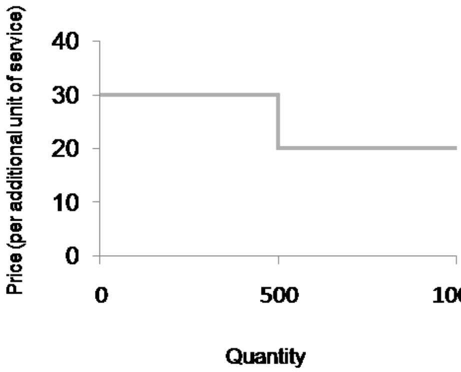

Fig. 2. A Volume Discount bid by a WSP

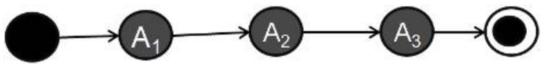

Fig. 3. A simple composite service with 3 states

subset of services which form a path in the composite service graph are also accepted in our setting. So the underlying auction becomes combinatorial as the WSPs can also bid on combination of services which form a path.

In emerging web service markets, many WSPs offer discounts based on the volume of purchase (strikeiron.com, xignite.com, xwebservices.com). So we generalize it and capture the discounts provided by a WSP using a volume discount bid. A typical volume discount bid submitted by a WSP in our setting is shown in the Figure 2. The implication of the bid is that the first 500 instances of service would have a cost of 30 cents per service; the next 500 instances of service would have a cost of 20 cents per service. For example, if a WSR wants 700 instances then, the total payment expected by WSP is $30 \times 500 + 20 \times 200$. So, it costs \$190.

In order to motivate and understand the problem, we present the following example where the WSR wants to procure a linearly structured composite service as shown in Figure 3.

C. A Motivating Example

Let there be a WSR who needs to procure 10 executions of a linear composite service shown in Figure 3. So WSR conducts an auction by seeking volume discount bids from WSPs on bundles of services which form a path in the composite service graph (Figure 3). Then the bids of the WSPs can be of the form shown in Figure 4. In the figure bid for a path $A_i A_{i+1} \ldots A_j$ is represented as A_{ij}. Note that bid on a path of length one means that it is a bid on stand-alone service.

D. Contributions and Outline

There is currently not much work available on web service composition problem in the presence of combinatorial or composite offerings. Moreover, the existing work [1][2] deals

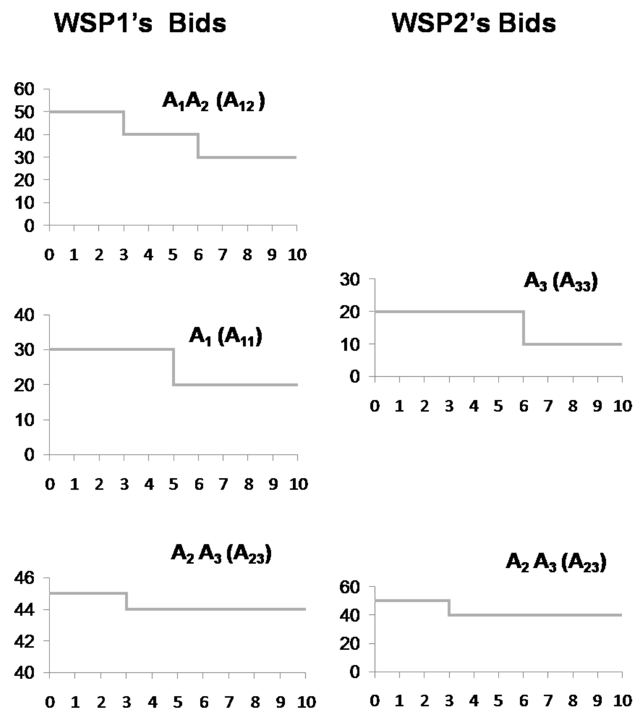

Fig. 4. Volume discount bids of WSP 1 and 2

only with the linear case. There is no work yet on solving the web service composition problem when WSPs provide volume discounts in addition to composite services. In this paper, we model such a problem as the winner determination problem in a procurement auction with combinatorial bids and volume discounts. In this setting, we make two contributions.

- We develop an efficient polynomial time algorithm for multiple instances (executions) of linearly structured end-to-end composite service. We prove that single instance linear web service composition algorithm can be used for multiple instance web service composition. Therefore multiple instance case is polynomial time solvable as the single instance case. This is the subject of Section 3.
- We develop an efficient polynomial time algorithm for multiple instances (executions) of tree structured end-to-end composite service. We show that the single instance composition case reduces to finding minimum weight cycle cover in the constructed graph. Minimum weight cycle cover can be computed in polynomial time. Hence the problem is solved in polynomial time. Multiple instance case can be reduced to single instance case as in linear case. This is the subject of Section 4.

Section 5 concludes the paper with a summary and by presenting several directions in which this work can be enhanced and extended.

We would like to emphasize that we do not consider quality of service (QoS) constraints in our work. QoS constraints have been considered in other papers. However all the solutions available in the presence of QoS constraints are computationally hard if exact solutions are required (being NP-hard problems). Consequently many heuristic based solutions or approximate algorithms have been suggested in the literature. Moreover, none of the existing solutions consider volume discount offerings. In the current paper, we solve the web service selection problem exactly using polynomial time algorithms, in the presence of combinatorial bids and volume discounts, but in the absence of QoS constraints.

II. A Review of Literature

A. Web Service Composition

Zeng et al [3] and Zeng et al [4] present a middleware platform which addresses the issue of selecting the web services for the purpose of their composition in a way that maximizes user satisfaction, expressed as utility functions over QoS attributes. They suggest a multidimensional QoS model which captures nonfunctional properties that are inherent to web services in general, for example availability and reputation. Their model defines number of QoS properties and methods for attaching values for these properties in the context of both stand-alone and composite web services. They describe and compare two selection approaches: one based on local (task-level) selection of services and the other based on global allocation of tasks to services using integer programming (IP). They suggest a dynamic composition approach, in which runtime depends on QoS of the component services.

Gao et al [5] suggest a QoS model that includes capacity and load, which may be useful for capacity planning and QoS negotiation at service provider side. They also illustrate the process of defining an objective function when there are composite constructs. They investigate implicit enumeration and found it impractical.

Canfora et al [6] propose a genetic algorithm (GA) based approach for QoS aware service composition. GAs permit to deal with QoS attributes having non-linear aggregation functions. Also, GAs are able to scale up when the number of WSPs for a particular task increases as compared to linear programming. Moreover, to deal with constraints, it is possible to adopt a fitness function with both static or dynamic penalty.

Zhang et al [7] also uses genetic algorithm to solve the composition problem. It uses the fact that QoS based selection problem is more or less a multistage decision-making problem and solves the problem using genetic algorithm optimized neural network algorithm. Conversion to multistage decision problem is done by considering each task as a stage and the vertices in each stage are the WSP's who can satisfy that task.

Gao et al [8] and Gao et al [9] present a 3-layer organization model and an approach for selecting globally optimal web services based on weighted multistage graph considering interface matching between web services. The approach transforms the problem of selecting the optimal execution plan for composite web services into the problem of selecting the longest path in a weighted multistage graph. Gao et al [8] describes a genetic or immune algorithm to select the optimal execution plan while Gao et al [9] adopts dynamic programming to select optimum web services for composite services dynamically.

Esmaeilsabzali and Larson [10] model the web service composition problem in game theoretic setting. They present an optimal strategy for the service provider when choosing its quality of service.

Richard, Goodwin, and Akkiraju [11] solves the composition problem using Markov decision processes (MDPs). They assume that WSR has beliefs for each path when ever a decision is to be taken in the end-to-end composite service

Mohabey et al [1][2] presents a combinatorial procurement auction for QoS and interface aware web services composition. They consider that WSP provide both stand-alone and composite services for forming an end-to-end composite service and gave a ILP formulation for the problem. But the ILP they gave requires O($m2^n$) variables. Where m is the number of bidders and n is the number of services in the execution path. However, algorithm for execution of ILP is known to be NP-hard. They assume that WSR requests only linear composite service.

B. Computational Complexity of Combinatorial Auctions and Volume Discount Auctions

The winner determination problem in combinatorial auctions in general are NP-hard [12]. However, for some special cases with restricted subsets there exists polynomial time algorithms. In the following we present a quick review on such algorithms.

Rothkopf et al [13] found that if the family of subsets on which a bidder can bid is limited to hierarchical subsets, meaning that every two subsets are disjoint or one is a subset of the other, then the winner determination problem can be solved in polynomial time. The problem of finding an optimal allocation for a combinatorial auction when a linear order exists among the items and where bidders can only bid on subsets of consecutive items is also shown to be polynomially solvable. Furthermore, Rothkopf et al [13] prove that a combinatorial auction where bidders can bid on subsets of a cardinality of at most two has a polynomially solvable winner determination algorithm. Tennenholtz [14] presents a combinatorial network auction, which he proves is computationally tractable. In this auction, the items are assumed to be arranged in a tree, where every node corresponds to an item. The idea is that bids can be submitted only on subsets of items that form a path in the network. If the items are structured in a directed acyclic graph and the bids are allowed on any directed subtree, the winner determination problem already becomes NP-hard [12]. Sandholm [12] also presents some more special cases of combinatorial auction which have polynomial time algorithms.

Volume discount auctions in general are again NP-hard to solve [15].

Fig. 5. A linear composite service

III. A Polynomial time algorithm for linear web service composition

A. The Problem

The WSR requests for k instances (executions) of the composite service $A_1 \ldots A_n$ as shown in the Figure 5. There are m bidders (WSPs) where each bidder submits a discount curves on some possible bundles. The goal is to procure k executions at minimum price.

1) Single Instance Composition: Rothkopf et al [13] presented a dynamic programming solution for the procurement auction when items are structured linearly and where bidders can only bid on subsets that form a path. The same solution can be used for single instance linear web service composition.

2) Multiple Instances Composition: For k executions of linear web service composition with volume discounts, we show that the above single instance solution can be used because of the two following facts about problem.

- all bidders have unlimited supply.
- by increasing the volume, price per unit is not increasing. That is the volume discount curves are non-increasing.

The possible bundles on which a bidder can bid are those which have a path in the end-to-end composite service graph, that is bundles like $A_iA_{i+1} \ldots A_j$ where there is a path from A_i to A_j. In the following sections we represent the bundle $A_iA_{i+1} \ldots A_j$ as A_{ij}.

Example :1

Let a WSR request 10 executions of composite service shown in Figure 3 and there are 2 bidders (WSPs). So each bidder can bid on only the bundles A_{11}, A_{22}, A_{33}, A_{12}, A_{23}, A_{13}. So, bids are of the form shown in the Figure 4.

Figures 6, 7 show pictorial representation of three different feasible solutions to the above example. We prove that any feasible solution of type Fig 6(a) can be converted to Figure 7(b), and Figure 7(b) can be converted to Figure 7(c). So it is enough to search for the optimal solution in the set consisting of the type Figure 7(c).

We define three sets Σ, Θ, Φ in order to prove our result.

- Σ be the set of all feasible solutions, in our example all the three solutions (Figures 7,6) belong to Σ.
- Θ be the subset of Σ, where each element in Θ has the property that it has only one provider for each bundle. In our example both solutions in Figure 7 belong to Θ
- Φ be the subset of Θ, where each element in Φ has the property that number of units purchased of each possible bundle is either k or 0 (example). In our example Figure 7(c) belong to Φ.

We prove that it is enough to search for the optimal solution in Φ. We use $C(.)$ to indicate the cost of the solution (i.e total cost of k instances)

Claim 1 : Every feasible solution α in Σ has a corresponding solution β in Θ whose cost is less than or equal to the cost of α.

$\forall \alpha \in \Sigma, \exists \beta \in \Theta$ where $C(\beta) \leq C(\alpha)$.

Claim 2 : Every feasible solution β in Θ has a corresponding solution γ in Φ whose cost is less than or equal to the cost of β.

$\forall \beta \in \Theta, \exists \gamma \in \Phi$ where $C(\gamma) \leq C(\beta)$.

By claim 1 and 2, one can say that its enough to search for optimal solution in Φ. For the example we discussed, elements in Φ are given in Figure 8.

To find the optimal solution in Φ we just need to have minimum price at k for all bundles and solve it. Therefore the problem is reduced to a single instance combinatorial auction for which there exists an $O(n^2)$ algorithm [13]. So independent of k, m, complexity of our algorithm is $O(n^2)$.

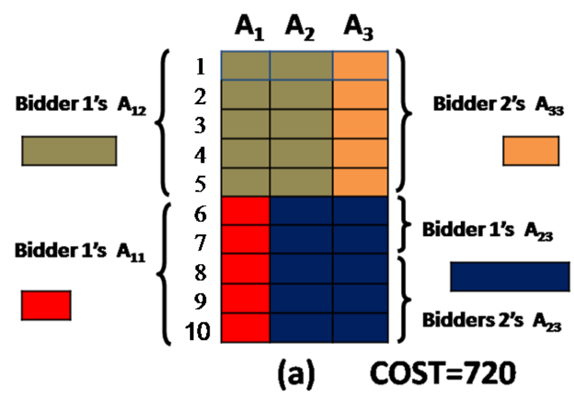

Fig. 6. A feasible solution with 2 providers for bundle A_{23}

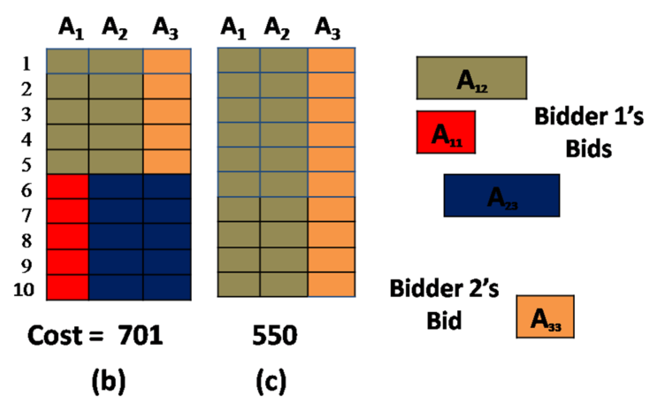

Fig. 7. Feasible solutions with 1 provider for all bundles

Proof of Claim 1: Let $\alpha \in \Sigma$ and let α has a bundle A_{ij} in its solution, where x providers are offering $(n_1, n_2 ... n_x)$ A_{ij}'s at $(p_1, p_2 ... p_x)$ per bundle, let p_{min} be minimum of $p_1, p_2, ... p_x$. It is always true that $(\sum n_i)p_{min} \leq (\sum n_i p_i)$. So we can replace all providers with the provider offering at

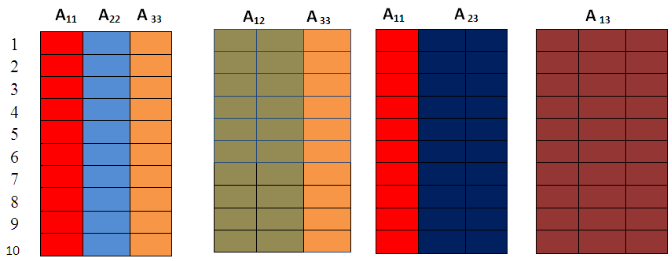

Fig. 8. Diagramatical representation of elements in set Φ

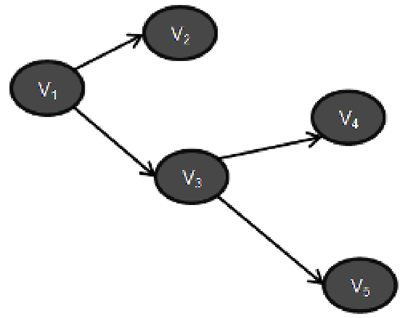

Fig. 9. Tree structured composite web service

p_{min}. Following the same procedure for all bundles we get β which belong to Θ and it is clear that $C(\beta) \leq C(\alpha)$

In our example take α as Figure 6 whose cost is 720. Two providers are offering A_{23}. So x=2, n_1=2, n_2=3, p_1=45, p_2=50 and p_1 is minimum. Therefore we can buy all the A_{23} bundles from bidder 1. It reduces our total cost to 701. Note that our solution is reduced to Figure 7(b).

Proof of Claim 2: Let $\beta \in \Theta$, so β should have k instances of end-to-end composite services (as it is a feasible solution). Let there be y different instances of end-to-end composite services in β each of $(n_1, n_2...n_y)$ instances whose price is $(p_1, p_2, ...p_y)$ per instance. Let p_{min} be min of $p_1, p_2, ...p_y$ its always true that $(\sum n_i)p_{min} \leq (\sum n_i p_i)$. So we can replace all instances with the instance whose cost is p_{min}. Hence the solution is changed to γ which belong to Φ and it is clear that $C(\gamma) \leq C(\beta)$.

In our example 1 by taking Figure 7(b) as β, whose cost is 701. Two types of instances are there in β, they are (A_{12}, A_3) and (A_1, A_{23}). Both the instances are of five copies. Therefore n_1=5, n_2=5, p_1=66 (cost per instance),p_2=74. Cost per unit of first type instance (p_1) is less than second instance. So we can buy all instances of type one. Hence the total cost is 550. Note that our solution is reduced to Figure 7(c).

Hence Claim 1 and 2.

By the above claims, we can say that its enough to search the optimal solution in Φ . Hence we can use the algorithm given in [13] to solve the problem whose complexity is $O(n^2)$ and also it is independent of k, m.

IV. A Polynomial time algorithm for tree structured web service composition

A. The Problem

The WSR requests k instances (executions) of a tree structured composite service as shown in the Figure 9. Let there be m bids (WSPs) where each bid is a discount curve (non-increasing) for the set of services which form a path in the graph. The goal is to procure k executions at minimum price.

First we solve the problem for the single instance tree structured web service composition and later we will show how it can be extended to multiple instances.

1) Single Instance Composition: Tennenholtz [14] presents a single unit combinatorial forward auction for the items which are structured as a tree and where bidders are allowed to bid only on subsets of items that form a path in the network. The auction algorithm presented has polynomial time complexity in number of services. But same algorithm cannot be used for our case as that is a maximization problem (Forward auction), where as we need minimization for our problem (Procurement auction). So, algorithm is modified as follows.

Let $T(V, E)$ be the composite service tree and B is the set of bids. A weighted graph $G(V \cup B, E^{'})$ is constructed by adding the following edges and nodes to T.

- The weight of every edge in T is assigned to 0.
- For every bid $b \in B$ which corresponds to a path v_1 to v_p, a vertex v_b and three edges are added in the following way. Two edges with weights 0 are added from v_b to v_1 and v_b to itself (self loop), third edges is added from v_p to v_b with edge weight equal to the bid value of b.
- For every vertex v in T, we add a self loop which connects v to itself and its weight is assigned to MAX. Where MAX is equal to any maximum number, like the sum of all bid values plus one.

The optimal solution to our problem is obtained from the minimal weight cycle cover of the graph G. Cycle cover of a graph is a set of cycles which cover all the vertices of the graph. The only possible cycles in G are, those formed with the bid vertex v_b and its corresponding path vertices or those formed by self loops. So finding minimum weight cycle cover leads to selecting bids which cover all the vertices V at minimum price. Services (vertices) which can not be provided by any WSP are covered with self loops whose cost is MAX. All the bids which are not in the optimal solution are again covered with self loops whose cost is 0. Hence, the whole graph G is covered by minimum weight cycles.

The minimum weight cycle cover of G can be computed by constructing a new bipartite graph $G'(V', E')$ such that

$$V'(G') = M \cup N \text{ where } M = N = V(G)$$
$$E'(G') = \{(v_i, v_j) | v_i \in M,\ v_j \in N \text{ and } (v_i, v_j) \in E(G)\}$$

It is easy to see that finding minimum weight cycle cover in G is equvialent to finding minimum weight perfect matching in G'. Their exists polynomial time algorithms for computing the minimum weight perfect matching in a bipartite garph ([16]). Hence, the time complexity of minimum weight cycle cover in G is polynomial in n, where n is the number of vertices in G. Note $n = |V \cup B|$, where $B = O(|V^2|)$. Therefore, the whole algorithm for tree structured web service composition is polynomial time executable.

2) Multiple Instances Composition: The solution to k instance tree structured web service composition can be extended from single instance case by following the similar arguments as that of linear structured web service composition. Because the same assumptions which hold for linear composition case also holds here.

V. Conclusions and Future Work

Web services are now a key ingredient of software services offered by software enterprises. Many standardized web services are now available as commodity offerings from web service providers. An important problem for a web service requester is the web service composition problem which involves selecting the right mix of web service offerings to execute an end-to-end business process. Web service offerings are now available in bundled form as composite web services and more recently, volume discounts are also on offer, based on the number of executions of web services requested. In this paper, we have developed efficient algorithms for the web service composition problem in the presence of composite web service offerings and volume discounts. We modeled this problem as a combinatorial auction with volume discounts. We first developed efficient polynomial time algorithms when the end-to-end composite service is a linear workflow of web services. Next we developed efficient polynomial time algorithms when the end-to-end composite service is a tree workflow of web services.

We would like to emphasize that we have not considered quality of service constraints in our work. QoS constraints have been considered in some papers in the literature. However all the solutions available in the presence of QoS constraints are computationally hard if exact solutions are required (being NP-hard problems) or they guarantee only approximate solutions. Moreover, none of the existing solutions consider volume discount offerings. In the current paper, we have solved the web service selection problem exactly using polynomial time algorithms, in the presence of combinatorial bids and volume discounts, but in the absence of QoS constraints.

There are many natural directions in which the work here can be extended and enhanced. We state a few directions below.

- We have developed polynomial time algorithms only in the case of linear and tree type of workflows. The next immediate type to be considered is when the web service workflow is a directed acyclic graph. We believe this problem is NP-hard.
- As already stated, QoS constraints have not been considered in our study. It would be interesting to explore this direction.
- We have implicitly assumed that the bids from all the web service providers are truthful bids. The realistic setting is to consider them as strategic players who may not bid truthfully. Mechanism design is the apt tool to employ to design incentive compatible auctions for web service selection.

References

[1] M. Mohabey, Y. Narahari, S. Mallick, P. Suresh, and S.V. Subrahmanya, "A combinatorial procurement auction for qos-aware web services composition," in *3rd IEEE Conference on Automation Science and Engineering(CASE 07)*, 2007.

[2] M. Mohabey, "ME Thesis:-Incentive Compatible Combinatorial Procurement Auction for Qos and Interface Aware Web Services Composition, Indian Institute of Science, Banglore, India," May 2007.

[3] L. Zeng, B. Benatallah, M. Dumas, and J. Kalagnanam, "Quality driven web services composition," in *International conference on World Wide Web. New York,NY, USA: ACM Press*, 2003, pp. 411–412.

[4] L. Zeng, B. Benatallah, A.H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "Qos-aware middleware for web services composition," in *IEEE Trans. Softw. Eng., vol. 30, no. 5*, 2003, pp. 311–327.

[5] A. Gao, D. Yang, S. Tang, and M. Zhang, "Web service composition using integer programming-based models," in *IEEE International Conference on e-Business Engineering. Washington, DC, USA*, 2005, pp. 603–606.

[6] G. Canfora, M.D. Penta, R. Esposito, and M.L. Villani, "An approach for qos-aware service composition based on genetic algorithms," in *Conference on Genetic and evolutionary computation. NY, USA: ACM Press*, 2005, pp. 1069–1075.

[7] L. Yang, Y. Dai, B. Zhang, and Y. Gao, "Dynamic selection of composite web services based on a genetic algorithm optimized new structured neural network," in *International Conference on Cyberworlds (CW 05)*, 2005, pp. 515–522.

[8] Y. Gao, B. Zhang, J. Na, L. Yang, Y. Dai, and Gong Q., "Optimal selection of web services for composition based on interface-matching and weighted multistage graph," in *International Conference on Parallel and Distributed Computing Applications and Technologies. Washington, DC, USA: IEEE Computer Society*, 2005, pp. 336–338.

[9] Y. Gao, J. Na, B. Zhang, L. Yang, and Q. Gong, "Optimal web services selection using dynamic programming," in *IEEE Symposium on Computers and Communications.Washington, DC, USA: IEEE Computer Society*, 2006, pp. 365–370.

[10] S. Esmaeilsabzali and k. Larson, "Service allocation for composite web services based on quality attributes," in *Seventh IEEE International Conference on E-Commerce Technology Workshops (CECW05). IEEE Computer Society*, 2005.

[11] P. Doshi, R. Goodwin, and R. Akkiraju, "Dynamic workflow composition using markov decision processes," in *IEEE International Conference on Web Services (ICWS 04)*, 2004.

[12] Tuomas Sandholm, "Algorithm for optimal winner determination in combinatorial auctions," in *Artificial Intelligence*, 2002, pp. 135–154).

[13] M. Rothkopf, A. Pekec, and R. M. Harstad, "Computationally manageable combinational auctions," in *Management Science,1998*, 1998, pp. 1131–1147.

[14] Moshe Tennenholtz, "Tractable combinatorial auctions and b-matching," in *Artificial Intelligence*, 2002, pp. 140(231–243).

[15] Anshul Kothari, David C. Parkes, and Subhash Suri, "Approximately-strategyproof and tractable multi-unit auctions," in *ACM Conf. on Electronic Commerce (EC'03)*, 2003, pp. 166–175.

[16] Harold W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.