Seminar in AI: E-Commerce and Computational Economics

Agent-Based Service Composition Through Simultaneous Negotiation in Forward and Reverse Auctions

Authors:Chris Priest, Claudio Bartolini, Andrew Byde

Introduction

- Emergence of E-commerce in the central part of the economy.
- Number of transactions taking place online increases, both between business to consumer and B2B.
- Agent Technology has been proposed as a means of automating some of the sophisticated negotiations.
- This paper focuses on a specific class of business process; service composition.

Introduction (cont'd)

- Over the last decade, companies have focused more on their core competencies, by outsourcing non-core activities to other companies.
- Internet is a suitable intermediary for these types of business arrangements.
- An important role emerges – the service composer. As companies focus on their core competencies, other companies can focus on creating composite packages.
- Algorithm specifications proposed in paper simultaneously purchase component services from a group of forward auctions and sell composite services via reverse auctions.

Service Composition

- Focus is on decision problem if it faces during negotiation. For effectiveness, the service composer will be involved in many interlinked negotiations and must make tradeoffs between them as they progress.
- Ideally (from service composer's perspective), it would be possible to make provisional arrangements without making any commitment.
- Component services may be available at fixed, guaranteed prices. However, there is an increasingly large market selling surplus goods and services via English auction. Therefore it is possible to be undercut by a competitor.

FreightMixer: An Example Scenario

- Is an imaginary transport company that ships goods around the world on behalf of customers.
- Does not own transport infrastructure. Instead, it exploits cheap last-minute sales of excess hold space to meet the needs of its customers.
- It aims to be the cheapest service.
- Acts in two distinct sets of markets:
  - Potential seller in end-to-end cargo services.
  - Potential buyer for hold space in flights and possibly ships.

FreightMixer: An Example Scenario (cont'd)

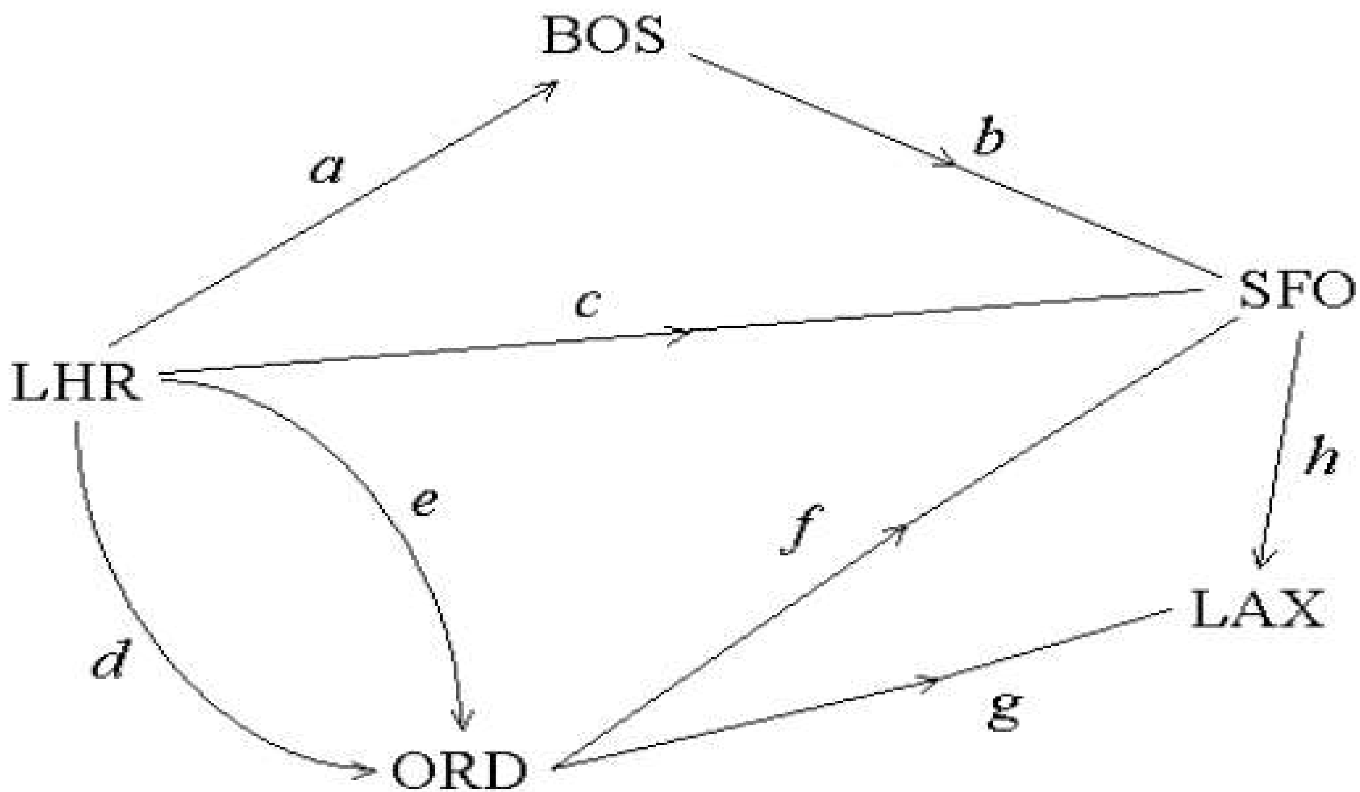
- In its role as service composer, it must
  - Understand requirements of the potential customers which are currently requesting services in the end-to-end cargo markets, and identify a service which could meet their needs.
  - Identify the alternative ways this service can be created from component services.
  - Identify potential sellers of these component services in the markets for hold space on flights.

FreightMixer: An Example Scenario (cont'd)

- A potential buyer, or set of buyers, who are currently requesting a service in the end-to-end cargo marketplaces.
- A service specification which meets the needs of these buyers.
- One or more alternate decompositions of this service into component services.
- A list of sellers in the markets for hold space who are offering to sell individual component services appearing in these decompositions.

FreightMixer: An Example Scenario (cont'd)

- Graph of Services



FreightMixer: An Example Scenario (cont'd)

- Request for Quotes reverse auction
  - 1 tonne crate from London (LHR) to San Francisco(SFO) – current best offer 230.
  - 1 tonne crate from London (LHR) to Los Angeles(LAX) – current best offer 260.
- For Shipment:
  - {LHR-SFO} : i) {LHR-SFO} ii) {LHR-ORD & ORD-SFO} iii) {LHR-BOS & BOS-SFO}
  - {LHR-LAX} : i) {LHR-SFO & SFO-LAX} ii) {LHR-ORD & ORD-LAX} iii) {LHR-ORD & ORD-SFO &SFO-LAX} iv) {LHR-BOS & BOS-SFO & SFO-LAX}

Specification of a Negotiation Algorithm

- All transactions take place through English auctions, reverse auctions and fixed price sales. One-to-one negotiation or double auctions are not considered.

- There are sufficiently large number of buyers and sellers, and sufficient stability in the environment, to allow statistical profiles of expected outcomes in auctions to be built.

Specification of the Decision Problem

- Set of participated auctions $\mathcal{A}$. Start roughly at the same time but may finish at different times. Both forward auctions selling and reverse auctions requiring single good or service.
- Both forward and reverse auctions are English auctions with fixed closing time. Minimum increment/decrement is 1.
- A contract to provide a good g is represented as the sale of a negative good $\overline{g}$. Reverse auctions are modelled as negative English auctions.
- Fixed price sellers are modeled as auctions with a known certain closing price.

Specification of the Decision Problem (cont'd)

- T: technology of the agent

  $(\{a\} \Leftarrow \{b_1, ..., b_n\}, c)$

  $(\emptyset \Leftarrow \{g, \overline{g}\}, 0)$

- G: given bag of goods.

  G': one-step rewrite of G at cost c, if:

  $(A \Leftarrow B, c) \in T,\ B \subseteq G,\ and\ G' = A \cup G \setminus B$

- W(G): set of rewrites of G.

Specification of the Decision Problem (cont'd)

- *v* (G): value received by agent from making use of G outside the auction environment.
- V(G): Valuation of a given bag of goods, G

$$V(G) = \max_{G' \in W(G)} (v(G') - c(G'))$$

- Bid Set: (A, **p**), where $A \subset \mathcal{A}$ and $\mathbf{p} : \mathcal{A} \to \mathbb{Z}$.

  u (A, **p**): Utility to the agent of winning a bid set (A, p)

$$u(A, \mathbf{p}) = V(A) - \sum_{a \in A} \mathbf{p}(a).$$

- Agent maintains probabilistic models of expected outcomes for each auction, based on past performance of similar auctions.

Specification of the Decision Problem (cont'd)

- a: Auction, $a \in \mathcal{A}$ is associated a price distribution.
- P_a (p): believed probability that auction a will close at price p.
- F_a (p): believed probability that auction a will close above price p.

$$F_a(p) = \sum_{p' \geq p} P_a(p')$$

- P_A (**p**): believed probability that the auctions in A will close at prices specified by a price function $\mathbf{p} : \mathcal{A} \rightarrow \mathbb{Z}$

$$P_A(\mathbf{p}) = \prod_{a \in A} P_a(\mathbf{p}(a)).$$

- F_A (**p**): believed probability that the auctions in A will close above prices specified by $\mathbf{p} : \mathcal{A} \rightarrow \mathbb{Z}$

$$F_A(\mathbf{p}) = \prod_{a \in A} F_a(\mathbf{p}(a))$$

Specification of the Decision Problem (cont'd)

- q: current price in auction a
- P_{win} (a, p, q): Believed probability of winning a bid at price p ≥ q in auction a with current price q.

$$P_{win}(a, p, q) := \frac{P_a(p)}{F_a(q)}$$

- P_{win} (A, **p**, **q**): Believed probability of winning a bid at prices **p** in auctions A with current prices **q**.

$$P_{win}(A, \mathbf{p}, \mathbf{q}) = \frac{P_A(\mathbf{p})}{F_A(\mathbf{q})}$$

Specification of the Algorithm

- E (B, A, **q**): Expected utility of a set of auctions B, given a set of observed prices **q**, and given that the agent holds the active bids in auctions A.

$$E(B, A, \mathbf{q}) = V(B) - C(B \cap A, \mathbf{q}) - C(B \setminus A, \mathbf{q} + 1)$$

  - C (S, **q'**): Expected cost of winning the auctions S at prices greater than or equal to **q'.**

$$C(S, \mathbf{q}') = \sum_{\mathbf{p}' \geq \mathbf{q}'} \sum_{a \in S} P_{win}(a, \mathbf{p}'(a), \mathbf{q}'(a))\mathbf{p}'(a)$$

Specification of the Algorithm (cont'd)

- E_c (B, A, **q**): expected value for following the commitment to B

$$E_c(B,A,\mathbf{q}) = E(B,A,\mathbf{q}) +$$
$$+ \sum_{B \subset S \subset A \cup B} P_{ret}(S,A,\mathbf{q}) \Big(V(S) - V(B) - \sum_{a \in S \setminus B} \mathbf{q}(a) \Big)$$

  - Commitment to 'B': the set of auctions that agent chooses to commit for all future time steps.

    S: possible set of auctions that agent may win by committing B.

    P_{ret} (S, A, **q**): probability that the auctions S\B will not be outbid, while the auctions A\S are.

$$P_{ret}(S,A,\mathbf{q}) = \frac{F_{A \setminus S}(\mathbf{q}+1) P_{S \setminus B}(\mathbf{q})}{F_{A \setminus B}(\mathbf{q})}$$

Specification of the Algorithm (cont'd)

At each time step, the algorithm calculates the commitment B which has the largest expected utility, and places minimal bids required to take lead in B\A. In practice,

- Initially the algorithm will identify the set of options which maximize it's a-priori expected utility.
- It will compete in these auctions when outbid.
- If competing bids in forward reduces the expected utility of committed set sufficiently, it may change to another set generating the same composite service.
- If the competing bids are placed in reverse auctions reducing the expected utility of the auction, it may withdraw from that reverse auction and may or may not from all forward auctions associated with it.

Specification of the Algorithm (cont'd)

Problems with the Algorithm:

- Algorithm does not in fact commit, since it re-evaluates its options at each opportunity. The estimate used is that agent chooses a single bundle with best overall expected utility.

- In reality, large number of auctions would make computation difficult to evaluate the expected value function with limited resources.

Specification of the Simplified Algorithm

The efficiency of the algorithm can be improved by making two simplifying assumptions

- Instead of considering every possible bundle of auctions, the algorithm focuses on a *candidate class*, consists of bundles which can be transformed to empty set using production rules.

- Algorithm will ignore any utility gain from unplanned items, when considering the expected utility of switching to a possible bundle.

Specification of the Simplified Algorithm (cont'd)

- First assumption can be validated if we assume the $v(G) <= c(G)$.

- Second assumption simplifies the E_c (B, A, **q**).

$$E_c(B, A, \mathbf{q}) = E(B, A, \mathbf{q}) +$$
$$+ \sum_{a \in A \setminus B} P_{win}(a, \mathbf{q}(a), \mathbf{q}(a)).(v(a) - \mathbf{q}(a))$$

Specification of the Simplified Algorithm (cont'd)

- Calculation of expected return for a given bundle only needs one valuation instead of all possible combinations of valuation.

- Candidate class of bundles, together with their valuations can be generated prior to participation in the auctions, given the set of auctions is known.

Computation of candidate bundle set and bundle valuation

- C: set of candidate bundles
- W, X: working sets that will contain bundles that will end up in C.
- O: bag of offerings available to the agent.
- T: technology of the agent or production rules.
- B: bundle
- V(B): valuation of the bundle B.
- J(G): set of justifications for G. a justification for a bundle G is a couple (G’, r), where G’ is a rewrite of G through r and is a production rule with cost c(r) associated with it.

Computation of candidate bundle set and bundle valuation (cont'd)

$C = \emptyset$
$V(\emptyset) = 0$
$W = (\emptyset)$
For each offering $o \in O$
  For each bundle $G \in W$
    Let $G' = G \cup \{o, \overline{o}\}$
    Let $J(G') = \emptyset$
    Let $W = W \cup \{G\}$

W now contains all achievable bundles of the kind $\{o, \overline{o}\}$

Computation of candidate bundle set and bundle valuation (cont'd)

Repeat
  Let G be a bundle in W
  For each element $a \in G$
    Let $R \subseteq T$ be the set of rules having a as *head*
    For each $r \in R$
      Let D be the *body* of r
      Let $G' = G \bigcup D \backslash \{a\}$
      If $G' \backslash O \neq \emptyset$ then break; G' is an invalid bundle
      Let $J(G') = J(G') \bigcup \{(G, r)\}$
      Let $W = W \bigcup G' \backslash G$
      Let $X = X \bigcup G$
Until $W = \emptyset$

Computation of candidate bundle set and bundle valuation (cont'd)

Repeat

  Let $G \in X$ s.t. $\forall (G', r) \in J(G), G' not \in X^4$

    If $J(G) = \emptyset$ then Let $V(G) = 0$

    Else

      $V(G) = \max_{(G',r) \in J(G)} (V(G') - c(r))$

      ; where c(r) is the cost associated to the rule r

      ; notice that $V(G')$ has been assigned, since $G' not \in X$

    Let $X = X \backslash \{G\}$

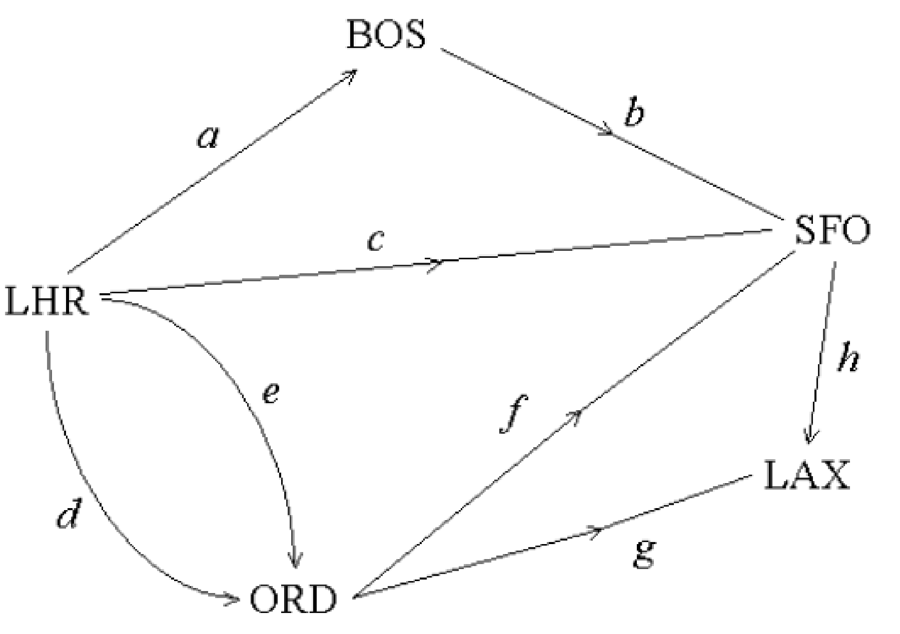
    Let $C = C \bigcup \{G\}$

Until $X = \emptyset$

Example

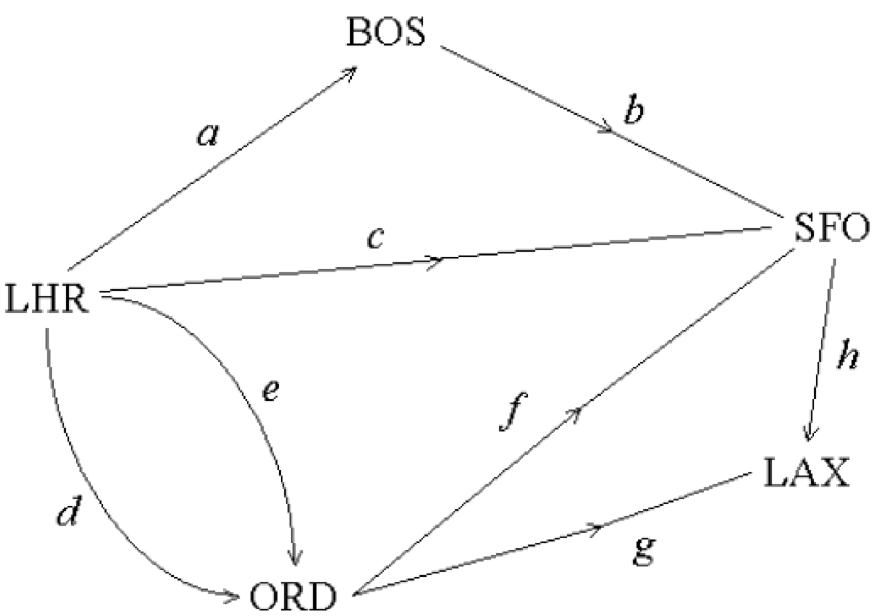
- a: {40, 45,…, 135, 140}
- b: {20, 25,…, 95, 100}
- c: {130, 135, 140, 145, 150}
- d: {50, 55,…, 105, 110}
- e: {80, 85,…, 115, 120}
- f: {30, 35,…, 65, 70}
- g: {20, 25,…, 135, 140}
- h: {70, 75, 80, 85, 90}

- la: {-250, -240,…, -200, -190}
- sfo: {-220, -210, -200, -190, -180}



Example (cont'd)

- sfo ← {a,b}
- sfo ← {c}
- ord ← {e}
- ord ← {d}
- sfo ← {ord, f}
- la ← {ord, g}
- la ← {sfo, h}



Example (cont'd)

- Initial optimal expected payoff for {d, g, c, sfo, la} is (80 + 80 + 140 – 200 - 220) = -120.
- Assume auctions progressed.
  - g has a leading bid of 105 held by another party with an expected purchase price of 125.
  - c is hold by our agent with a bid of 130 and an expected purchase price of 140.
- Assume reverse auction la has a leading offer of 210.

Example (cont'd)

- If the reverse auction, la, is held by another party.
  - This gives an expected closing price of 195.
  - The expected profit of the bundle is now 50.
  - Options:
    - Withdrawal from la auction and keeping sfo, which has an expected profit of 60.
    - If no bids were held in any auction the optimal bundle would be {f, d, sfo} with expected profit of 70. However, E_c (B, A, **q**) shows the expected profit as 44, considering the additional cost of de-committing from auction c.

Example (cont'd)

- If the reverse auction, la, is hold by our agent.
  - This gives an expected closing price of 200.
  - The expected profit of the bundle is now 55.
  - Options:
    - Withdrawal from auction la and switching to bundle {c, sfo}. However, switching contains high risk of accidentally winning auction la and be forced to purchase the goods from fixed-price competitors (260).
    - Better alternative: Rather than using c to generate sfo, it can be used with h to generate la with expected cost of 200. f and d can be used to generate sfo. This gives a total expected profit of 70.

Conclusions and Future Work

- Paper focused on the key problem of effective negotiation for service composition.
- Presented an exact and an effective form of the algorithm.
- In the future, authors plan to extend the algorithm to handle
  - Other forms of negotiation, in particular one-to-one negotiation
  - Various auction types with staggered opening times.