Autonomous Bidding
in the
Trading Agent Competition

Amy Greenwald
Brown University

with

Justin Boyan
ITA Software

Federal Communications Commission
Computing Research Association
Digital Fellows Program

May 7, 2002
Key TAC Features

Simultaneous Auctions

Combinatorial Valuations

- **Complements**
  - $v(X\bar{Y}) + v(\bar{X}Y) \leq v(XY)$
  - camera, flash, and tripod

- **Substitutes**
  - $v(X\bar{Y}) + v(\bar{X}Y) \geq v(XY)$
  - Canon AE-1 and Canon A-1
Examples

FCC auctions

eBay auctions

• proxy bidding agents

• bid up to the value of good $x$

$v(\text{Camera + Flash})$

• autonomous bidding agents

• bid up to the marginal value of good $x$
Bid Determination

Allocation

- given the set of goods I hold, what is the maximum valuation I can attain?

Acquisition

- given the set of goods I hold, and given ask prices in any open auctions, on what set of additional goods should I bid to maximize valuation less costs?

Requisition

- given the set of goods I hold, and given bid prices in any open auctions, on what set of goods should I place asks to maximize valuation plus profits?

Completion

- given the set of goods I hold, and given ask and bid prices in any open auctions, on what set of goods should I place bids or asks to maximize my valuation less costs plus profits?
Overview

- TAC Market Game
- TAC Agent Architecture
- RoxyBot Agent Architecture
TAC Market Game

Score = Valuation – Costs + Profits

Supply

- **Flights**  Inbound and Outbound
- **Hotels**  Grand Hotel and Le FleaBag Inn
- **Entertainment**  Red Sox, Symphony, Phantom

Auctions

- **Flights**  infinite supply, prices follow random walk, clear continuously, no resale permitted
- **Hotels**  ascending, multi-unit, 16th price auctions, transactions clear and random auction closes once per minute, no resale permitted
- **Entertainment**  continuous double auctions, initial endowment, resale is permitted
TAC Market Game

Demand

<table>
<thead>
<tr>
<th>Client</th>
<th>IAD</th>
<th>IDD</th>
<th>HV</th>
<th>RV</th>
<th>SV</th>
<th>TV</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>99</td>
<td>134</td>
<td>118</td>
<td>65</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>4</td>
<td>131</td>
<td>170</td>
<td>47</td>
<td>49</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>147</td>
<td>13</td>
<td>55</td>
<td>49</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>4</td>
<td>145</td>
<td>130</td>
<td>60</td>
<td>85</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>4</td>
<td>82</td>
<td>136</td>
<td>68</td>
<td>87</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>4</td>
<td>53</td>
<td>94</td>
<td>51</td>
<td>105</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>3</td>
<td>54</td>
<td>156</td>
<td>126</td>
<td>71</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>5</td>
<td>113</td>
<td>119</td>
<td>187</td>
<td>143</td>
</tr>
</tbody>
</table>

Feasible Packages

- arrival date prior to departure date
- same hotel on all intermediate nights
- at most one entertainment event per night
- at most one of each type of entertainment
TAC Market Game

Valuation

$$1000 - \text{travelPenalty} + \text{hotelBonus} + \text{funBonus}$$

$$\text{travelPenalty} = 100(|\text{IAD} - \text{AD}| + |\text{IDD} - \text{DD}|)$$

$$\text{hotelBonus} = \begin{cases} 
\text{HV} & \text{if } H = G \\
0 & \text{otherwise}
\end{cases}$$

$$\text{funBonus} = \text{entertainment values}$$

Allocation

<table>
<thead>
<tr>
<th>Client</th>
<th>AD</th>
<th>DD</th>
<th>H</th>
<th>Ticket</th>
<th>Valuation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>G</td>
<td>SV1, RV2</td>
<td>1351</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
<td>G</td>
<td>RV1</td>
<td>1201</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>G</td>
<td>—</td>
<td>1147</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>4</td>
<td>G</td>
<td>RV3</td>
<td>1275</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>3</td>
<td>F</td>
<td>RV1, TV2</td>
<td>1123</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>4</td>
<td>G</td>
<td>TV3</td>
<td>1058</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>3</td>
<td>F</td>
<td>SV1, RV2</td>
<td>1282</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>5</td>
<td>G</td>
<td>TV1, SV3, RV4</td>
<td>1562</td>
</tr>
</tbody>
</table>
TAC Agent Architecture

REPEAT

1. how many copies of each good do i want?

2. on the goods i want, should i bid now or later?

3. for the goods i want now, what am i willing to pay?

UNTIL game over
Bid Determination

Bid on $S \setminus T$

Ask for $T \setminus S$

Bid on $S \setminus T$
Ask for $T \setminus S$
Observations

WD $\cong$ Allocation

- WD: auctioneer seeks the set of combinatorial bids that maximizes profits, given feasibility constraints

WDR $\cong$ Acquisition

- WDR (WD with reserve prices): auctioneer seeks the set of combinatorial bids that maximizes the difference between profits and reserve prices

BD problems in simultaneous auctions $\cong$

WD problems in combinatorial auctions
Pricelines

Buying Priceline
\[ \vec{p}_g = \langle 0, 0, 0, 0, 20, 30, \infty, \infty, \ldots \rangle \]
\[ \forall g, \quad n_{\text{Buy}}(S,g) = \sum_{q \in S} q_g \]
\[ \forall g, \quad \text{Cost}_g(S,P) = \sum_{n=1}^{n_{\text{Buy}}(S,g)} p_{gn} \]
\[ \text{Cost}(S,P) = \sum_{g \in G} \text{Cost}_g(S,P) \]

Selling Priceline
\[ \vec{\pi}_g = \langle 10, 5, 2, 1, 0, 0, 0, 0, -\infty, -\infty, \ldots \rangle \]
\[ \forall g, \quad n_{\text{Sell}}(S,g) = \sum_{q \notin S} q_g \]
\[ \forall g, \quad \text{Profit}_g(S,\Pi) = \sum_{n=1}^{n_{\text{Sell}}(S,g,\Pi)} \pi_{gn} \]
\[ \text{Profit}(S,\Pi) = \sum_{g \in G} \text{Profit}_g(S,\Pi) \]
Formalization

Acquisition
Inputs: set of packages $Q$
    set of buying pricelines $P$
    valuation function $v : Q \rightarrow \mathbb{R}^+$
Output: $S^* \in \arg\max_{S \subseteq Q} (\text{Valuation}(S, v) - \text{Cost}(S, P))$

Requisition
Inputs: set of packages $Q$
    set of selling pricelines $\Pi$
    valuation function $v : Q \rightarrow \mathbb{R}^+$
Output: $S^* \in \arg\max_{S \subseteq Q} (\text{Valuation}(S, v) + \text{Profit}(S, \Pi))$

Completion
Inputs: set of packages $Q$
    set of buying pricelines $P$
    set of selling pricelines $\Pi$
    valuation function $v : Q \rightarrow \mathbb{R}^+$
Output: $S^* \in \arg\max_{S \subseteq Q} (\text{Val}(S, v) - \text{Cost}(S, P) + \text{Profit}(S, \Pi))$
Formalization

Acquisition
Inputs: set of packages $Q$
  set of buying pricelines $P$
  valuation function $v : Q \rightarrow \mathbb{R}^+$
Output: $S^* \in \arg \max_{S \subseteq Q} (\text{Valuation}(S, v) - \text{Cost}(S, P))$

Requisition
Inputs: set of packages $Q$
  set of selling pricelines $\Pi$
  valuation function $v : Q \rightarrow \mathbb{R}^+$
Output: $T^* \in \arg \max_{T \subseteq Q} (\text{Valuation}(T, v) + \text{Profit}(T, \Pi))$

Completion
Inputs: set of packages $Q$
  set of buying pricelines $P$
  set of selling pricelines $\Pi$
  valuation function $v : Q \rightarrow \mathbb{R}^+$
Output: $S^*, T^* \in \arg \max_{S,T \subseteq Q} (\text{Valuation}(S, v) - \text{Cost}(S, P) + \text{Profit}(T, \Pi) - \text{Cost}(T, P))$
Completion $\mapsto$ Acquisition

Buying Priceline
$\vec{p}_g = \langle 0, 0, 0, 0, 20, 30, \infty, \infty, \ldots \rangle$

Selling Priceline
$\vec{\pi}_g = \langle 10, 5, 2, 1, 0, 0, 0, 0, -\infty, -\infty, \ldots \rangle$

1st Reduction

- add reverse of selling pricelines to buying pricelines:
  $\vec{p}_g + \text{reverse}(\vec{\pi}_g) = \langle 1, 2, 5, 10, 20, 30, \infty, \infty, \ldots \rangle$

2nd Reduction

- extend package input set with singleton packages, one for each copy of each good in selling pricelines; assign selling prices as dummy package valuations:
  $\vec{\pi}_g \mapsto 4$ new packages with valuations $10, 5, 2, 1$

Bid Determination in double-sided auctions $\mapsto$
Bid Determination in single-sided auctions
Utility

Acquisition
Inputs: set of packages $Q$
set of buying pricelines $P$
valuation function $v : Q \rightarrow \mathbb{R}^+$
Output: $S^* \in \arg \max_{S \subseteq Q} (\text{Valuation}(S, v) - \text{Cost}(S, P))$
$u(S^*) = \max_{S \subseteq Q} (\text{Valuation}(S, v) - \text{Cost}(S, P))$

Example
valuations
$v(XYZ) = v(XY) = v(YZ) = 500$
$v(X) = v(Y) = v(Z) = v(XZ) = 0$

pricelines
$p(X) = p(Y) = p(Z) = 100$

utilities
$u(XY) = u(YZ) = 300$
Marginal Utility
for the goods i want now, what am i willing to pay?

Acquisition
Inputs: set of packages $Q$
    set of buying pricelines $P$
    valuation function $v : Q \rightarrow \mathbb{R}^+$
Output: $S^* \in \text{arg max}_{S \subseteq Q}(\text{Valuation}(S, v) - \text{Cost}(S, P))$
    $u(S^*) = \text{max}_{S \subseteq Q}(\text{Valuation}(S, v) - \text{Cost}(S, P))$

Answer
$u(x) = u(A \cup \{x\}) - u(A)$, with $p(x) = 0 \& p(x) = \infty$

Example
$u(X) = u(XYZ) - u(YZ) = 400 - 300 = 100$
$u(Y) = u(XYZ) - u(XZ) = 400 - 0 = 400$
$u(Z) = u(XYZ) - u(XY) = 400 - 300 = 100$

Bids
$b(Y) = 300, b(X) = b(Z) = 100$
$v(Y) - p(Y) = 200$
RoxyBot
how many copies of each good do i want?

Acquisition
Inputs: set of packages $Q$
    set of buying pricelines $P$
    valuation function $v : Q \rightarrow \mathbb{R}^+$
Output: $S^* \in \arg\max_{S \subseteq Q} (\text{Valuation}(S, v) - \text{Cost}(S, P))$
    $u(S^*) = \max_{S \subseteq Q} (\text{Valuation}(S, v) - \text{Cost}(S, P))$

Answer
$n\text{Buy}(S^*, g) = \sum_{q \in S^*} q_g$

Example
$n\text{Buy}({XY}, X) = 1$
$n\text{Buy}({XY}, Y) = 1$
$n\text{Buy}({XY}, Z) = 0$
XOR
$n\text{Buy}({YZ}, X) = 0$
$n\text{Buy}({YZ}, Y) = 1$
$n\text{Buy}({YZ}, Z) = 1$
Marginal Utility, Revisited
for the goods I want now, what am I willing to pay?

Acquisition
Inputs: subset of packages $Q$
set of buying pricelines $P$
valuation function $v : Q \rightarrow \mathbb{R}^+$
Output: $S^* \in \arg \max_{S \subseteq Q} (\text{Valuation}(S, v) - \text{Cost}(S, P))$
\[ u(S^*) = \max_{S \subseteq Q} (\text{Valuation}(S, v) - \text{Cost}(S, P)) \]

Answer
\[ u(x) = u(A \cup \{x\}) - u(A), \text{ with } p(x) = 0 \& p(x) = \infty \]

Example
\[ u(X) = u(XY) - u(Y) = 400 - 0 = 400 \]
\[ u(Y) = u(XY) - u(X) = 400 - 0 = 400 \]

Bids
\[ b(X) = b(Y) = 400, \quad b(Z) = 0 \]
\[ v(XY) - p(X) - p(Y) = 300 \]
RoxyBot 2000 Architecture

(A) REPEAT

1. Ping server to update current prices and holdings
2. Estimate clearing prices and build buy/sell pricelines
3. Run completer to find optimal buy/sell quantities
4. Bid/ask marginal valuations

   UNTIL game over

(B) Run allocator
TAC 2000 Statistics

Preliminary Round (~70 Games)

Final Round (13 Games)
**Price Uncertainty**

for the goods i want now, what am i willing to pay?

**Example**

\[ p(x) = 0, \text{ with probability } \frac{1}{2}, \text{ and} \]
\[ p(x) = 200, \text{ with probability } \frac{1}{2}, \]
for all \( x \in \{X, Y, Z\} \)

**Answer**

average marginal utility

**Bidding Policy**

<table>
<thead>
<tr>
<th>( X )</th>
<th>( Y )</th>
<th>( Z )</th>
<th>( u(X) )</th>
<th>( u(Y) )</th>
<th>( u(Z) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>500</td>
<td>0</td>
</tr>
<tr>
<td>200</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>500</td>
<td>200</td>
</tr>
<tr>
<td>0</td>
<td>200</td>
<td>0</td>
<td>0</td>
<td>500</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>200</td>
<td>200</td>
<td>500</td>
<td>0</td>
</tr>
<tr>
<td>200</td>
<td>200</td>
<td>0</td>
<td>0</td>
<td>500</td>
<td>200</td>
</tr>
<tr>
<td>200</td>
<td>0</td>
<td>200</td>
<td>200</td>
<td>300</td>
<td>200</td>
</tr>
<tr>
<td>0</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>500</td>
<td>0</td>
</tr>
<tr>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>300</td>
<td>200</td>
</tr>
<tr>
<td>Bids</td>
<td>100</td>
<td>450</td>
<td>100</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
RoxyBot Under Uncertainty

how many copies of each good do i want?

Answer

sound and complete set of packages

Example

\( n_{\text{Buy}}(\{XY\}, X) = 1 \)
\( n_{\text{Buy}}(\{XY\}, Y) = 1 \)
\( n_{\text{Buy}}(\{XY\}, Z) = 0 \)

Bidding Policy

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>( u(X) )</th>
<th>( u(Y) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>200</td>
<td>0</td>
<td>500</td>
<td>300</td>
</tr>
<tr>
<td>0</td>
<td>200</td>
<td>300</td>
<td>500</td>
</tr>
<tr>
<td>200</td>
<td>200</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>Bids</td>
<td></td>
<td>400</td>
<td>400</td>
</tr>
</tbody>
</table>
# Bidding Under Uncertainty

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>AT Tac</th>
<th>RoxyBot</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>200</td>
<td>0</td>
<td>0</td>
<td>500</td>
<td>300</td>
</tr>
<tr>
<td>0</td>
<td>200</td>
<td>0</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>200</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>200</td>
<td>200</td>
<td>0</td>
<td>300</td>
<td>100</td>
</tr>
<tr>
<td>200</td>
<td>0</td>
<td>200</td>
<td>0</td>
<td>300</td>
</tr>
<tr>
<td>0</td>
<td>200</td>
<td>200</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>200</td>
<td>200</td>
<td>200</td>
<td>−200</td>
<td>100</td>
</tr>
</tbody>
</table>

Scores

- ATTac: 275
- RoxyBot: 300
RoxyBot 2001 Architecture

INPUTS
Truncation Parameter \( t_0 \in [0.5, 1.0] \)
Schedule by which to Decay \( t_0 \)

(A) REPEAT
1. Updates prices and winnings
2. Estimate clearing price distributions
3. Initialize \( d = 0, s = 8, n = 0, \) and \( t = t_0 \)
4. REPEAT
   (a) Sample clearing price distributions
   (b) Compute optimal completion \( D_n \)
   (c) Store \( D_n \) in completion list
   (d) Increment \( n \)
   (e) Tally results
      i. for all items \( i \)
         o initialize \( \#i = 0 \)
         o for all completions \( D_n \)
            − if \( i \in D_n \), increment \( \#i \)
         o if \( \#i/n > t \)
            − increment \( d \)
            − add \( i \) to \( D \)
         o if \( \#i/n < 1 - t \)
            − decrement \( s \)
            − delete \( i \) from \( S \)
   (f) Discard from list inconsistent completions
   (g) Set \( n \) equal to length of completion list
   (h) Decay \( t \)
      UNTIL \( d = s \) or TIME OUT
(B) Run allocator
Future Work

Empirical Testing

○ Completion vs. No Completion
○ Sampling vs. No Sampling
○ ILP vs. LP Relaxation

Theoretical Study

○ timing—optimal stopping problem
○ estimate joint price distributions