

POLITECNICO DI MILANO

Facoltà di Ingegneria



Corso di Laurea in Ingegneria Informatica  
Dipartimento di Elettronica e Informazione

PM-AI & R PROJECT

Progetto di Intelligenza Artificiale  
e Robotica del Politecnico di Milano

AGENTI AUTONOMI PER ASTE ELETTRONICHE:  
PROGETTO, REALIZZAZIONE E ANALISI  
STATISTICA DEL COMPORTAMENTO

Relatore: Dott. Pier Luca LANZI  
Correlatore: Prof. Marco COLOMBETTI

Tesi di Laurea di:  
Alessandro STRADA  
Matr. Nr. 629248

Anno Accademico 2000-2001



## Sommario

Questa tesi ha come argomento lo studio di problematiche relative alla progettazione e alla realizzazione di agenti autonomi per aste elettroniche. Si è partiti da un problema concreto: sviluppare un agente che potesse partecipare alla competizione internazionale per agenti di commercio elettronico, nota come *Trading Agent Competition 2001* (TAC-01). Innanzitutto, è stata effettuata un'analisi sistematica delle diverse problematiche che devono essere affrontate durante la gara ed è stata definita una strategia basata su euristiche. La strategia è stata implementata mediante un'opportuna architettura a regole. Questa architettura costituisce il nucleo dell'agente *polimi\_bot* che è stato realizzato all'interno del Laboratorio di Intelligenza Artificiale e Robotica per partecipare alla TAC-01. La competizione, che si è svolta dal 7 settembre al 14 ottobre 2001, è composta da diverse sessioni di gioco a cui gli agenti partecipano in remoto, collegandosi alle macchine dell'Università del Michigan che ha organizzato l'evento. I risultati della competizione sono stati analizzati con metodi statistici sia per valutare le prestazioni dell'agente *polimi\_bot* sia per verificare quanto questo tipo di competizione sia efficace al fine di valutare differenti strategie nel settore delle aste elettroniche. Dalle analisi effettuate emerge che nella TAC non esiste un vero vincitore. Infatti il breve periodo (poco più di un mese) durante il quale si è svolta la competizione non ha permesso di individuare se un singolo agente avesse prestazioni particolarmente superiori a quelle degli altri. Tuttavia le analisi hanno evidenziato un gruppo di agenti, tra cui *polimi\_bot*, che si sono distinti per l'efficacia delle loro strategie.



# Ringraziamenti

Ringrazio innanzitutto il Prof. Marco Colombetti per avermi proposto questa tesi e per avermi fornito indicazioni fondamentali nell'analisi dei risultati della competizione. Ringrazio inoltre il Dott. Pier Luca Lanzi per avermi seguito costantemente e per avermi fornito tutto il supporto necessario, dimostrandosi sempre estremamente disponibile.

Ringrazio il *TAC-Team* per aver organizzato e gestito l'intera competizione e per la rapidità con cui hanno risolto i problemi che si sono presentati.

Ringrazio la mia famiglia per avermi accompagnato in America, per la finale della TAC che si è svolta a Tampa (Florida). Ringrazio inoltre gli amici del Politecnico e di Vigevano che mi hanno tenuto compagnia e mi hanno sopportato per tutti questi anni.

Infine ringrazio la comunità *open-source* che ha prodotto quasi tutti gli strumenti che ho utilizzato per realizzare questa tesi.



# Indice

<b>Ringraziamenti</b>	<b>iii</b>
<b>Elenco delle figure</b>	<b>vii</b>
<b>Elenco delle tabelle</b>	<b>ix</b>
<b>1 Introduzione</b>	<b>1</b>
1.1 Obiettivi . . . . .	3
1.2 Struttura della tesi . . . . .	4
1.3 Contributi originali . . . . .	5
<b>2 Stato dell'arte</b>	<b>7</b>
2.1 Commercio elettronico . . . . .	7
2.1.1 Storia . . . . .	8
2.1.2 Tendenze . . . . .	11
2.2 Modelli per il commercio elettronico . . . . .	12
2.2.1 Business to business . . . . .	13
2.2.2 Business to consumer . . . . .	14
2.3 Modelli d'asta . . . . .	15
2.3.1 Asta inglese . . . . .	16
2.3.2 Asta a offerta segreta . . . . .	17
2.3.3 Asta olandese . . . . .	18
2.3.4 Asta di Vickrey . . . . .	19
2.3.5 Asta a doppia chiamata continua . . . . .	20

2.3.6	Asta combinatoria . . . . .	23
2.4	Aste elettroniche . . . . .	25
2.4.1	Applicazioni . . . . .	25
2.4.2	Piattaforme . . . . .	27
<b>3</b>	<b>La <i>Trading Agent Competition</i></b>	<b>31</b>
3.1	Introduzione . . . . .	31
3.2	Regole . . . . .	32
3.2.1	I mercati . . . . .	33
3.2.2	La funzione di utilità . . . . .	34
3.3	Analisi della competizione . . . . .	36
3.3.1	Il problema dell'acquisizione . . . . .	36
3.3.2	Il problema dell'allocazione . . . . .	42
3.3.3	Il problema del completamento . . . . .	42
3.4	La piattaforma <i>AuctionBot</i> . . . . .	43
3.4.1	Messaggi e attività . . . . .	44
3.4.2	Interfaccia di programmazione . . . . .	45
<b>4</b>	<b>L'agente <i>polimi_bot</i> per la TAC-01</b>	<b>49</b>
4.1	L'architettura dell'agente . . . . .	49
4.2	Modelli e predittori . . . . .	51
4.2.1	Asta dei biglietti aerei . . . . .	51
4.2.2	Asta delle stanze d'albergo . . . . .	55
4.2.3	Asta dei biglietti per le attrazioni . . . . .	57
4.3	Strategia . . . . .	59
4.4	Allocazione delle risorse . . . . .	62
4.5	Creazione delle offerte . . . . .	63
4.6	Osservazioni . . . . .	66
4.7	Implementazione . . . . .	66
4.8	Formalizzazione delle regole . . . . .	69
4.8.1	Trigger . . . . .	69
4.8.2	Condizioni . . . . .	70

4.8.3	Azioni . . . . .	71
4.8.4	Le regole dell'agente <i>polimi_bot</i> . . . . .	71
4.9	Sviluppi . . . . .	72
<b>5</b>	<b>I risultati della TAC-01</b>	<b>75</b>
5.1	I partecipanti . . . . .	75
5.2	Il torneo preliminare . . . . .	78
5.2.1	Il turno eliminatorio . . . . .	78
5.2.2	Gli agenti finalisti . . . . .	79
5.2.3	Il turno di selezione . . . . .	81
5.3	Le finali . . . . .	81
5.4	Le strategie . . . . .	84
<b>6</b>	<b>Analisi statistica dei risultati</b>	<b>91</b>
6.1	La scelta del metodo . . . . .	91
6.2	Analisi della varianza . . . . .	93
6.3	Procedure di confronto multiplo . . . . .	94
6.4	Osservazioni sull'applicabilità . . . . .	95
6.5	Il turno eliminatorio . . . . .	96
6.5.1	Analisi statistica . . . . .	96
6.5.2	Osservazioni . . . . .	98
6.6	Il turno di selezione . . . . .	99
6.6.1	Analisi statistica . . . . .	99
6.6.2	Osservazioni . . . . .	101
6.7	Conclusioni . . . . .	103
<b>7</b>	<b>Conclusioni</b>	<b>117</b>
<b>A</b>	<b>I parametri dell'agente <i>polimi_bot</i></b>	<b>121</b>
	<b>Bibliografia</b>	<b>123</b>



# Elenco delle figure

2.1	Domanda e offerta. . . . .	22
3.1	<i>Scatter plot</i> di mille simulazioni della camminata aleatoria. . .	38
3.2	Istogrammi delle distribuzioni dei prezzi dei biglietti aerei in vari istanti della gara. . . . .	39
4.1	Architettura dell'agente. . . . .	50
4.2	Stimatore del prezzo dei biglietti aerei. . . . .	54
4.3	Esempi di utilizzo dei due stimatori per le aste degli alberghi.	58
4.4	Confronto tra gli stimatori per le aste dei biglietti per le at- trazioni. . . . .	60



# Elenco delle tabelle

4.1	Tabella dei valori delle attrazioni per due clienti. . . . .	63
4.2	Le regole dell'agente <i>polimi_bot</i> . . . . .	74
5.1	I partecipanti alla TAC-01 . . . . .	78
5.2	Classifica finale del turno eliminatorio della TAC-01. . . . .	80
5.3	Classifica finale del turno di selezione della TAC-01. . . . .	82
5.4	Classifica della prima semifinale della TAC-01. . . . .	83
5.5	Classifica della seconda semifinale della TAC-01. . . . .	84
5.6	Classifica della finale della TAC-01. . . . .	85
6.1	ANOVA sui risultati del turno preliminare. . . . .	96
6.2	ANOVA sui risultati del turno di selezione. . . . .	100
6.3	Test di Scheffé sul turno preliminare. . . . .	104
6.4	Test di Bonferroni sul turno preliminare. . . . .	105
6.5	Test di Tukey sul turno preliminare. . . . .	106
6.6	Test di omogeneità SNK sul turno preliminare. . . . .	107
6.7	Test di omogeneità di Scheffé sul turno preliminare. . . . .	108
6.8	Test di omogeneità di Tukey sul turno preliminare. . . . .	109
6.9	Test di Scheffé sul turno di selezione. . . . .	110
6.10	Test di Bonferroni sul turno di selezione. . . . .	111
6.11	Test di Tukey sul turno di selezione. . . . .	112
6.12	Test di omogeneità SNK sul turno di selezione. . . . .	113
6.13	Test di omogeneità di Scheffé sul turno di selezione. . . . .	114
6.14	Test di omogeneità di Tukey sul turno di selezione. . . . .	115



# Capitolo 1

## Introduzione

Attualmente ci sono centinaia di siti sul *Web* che permettono di comprare e vendere ogni tipo di bene attraverso aste elettroniche. Tenere traccia delle diverse aste che offrono il bene che si vuole acquistare è però complesso e dispendioso in termini di tempo. Per aiutare gli utenti ad interagire con i siti di aste elettroniche, stanno nascendo sempre più servizi che permettono di automatizzare l'interazione con le diverse aste a cui si vuole partecipare. Tuttavia anche questi strumenti falliscono nel caso in cui sia necessario controllare un gran numero di aste che mettono in palio beni diversi, su diversi siti e con formati d'asta differenti. Uno degli obiettivi della ricerca nel campo delle aste elettroniche è quindi lo sviluppo di nuovi e più sofisticati strumenti che permettano di automatizzare l'interazione con i siti di aste elettroniche. Questi strumenti dovrebbero partecipare alle aste con la stessa flessibilità di un utente umano ma contemporaneamente dovrebbero poter gestire un maggior numero di aste rispetto a quelle che potrebbe gestire l'utente. L'interesse nello sviluppo di questi strumenti è in crescita dal momento che il settore delle aste elettroniche sta diventando sempre più importante nell'ambito del commercio elettronico. Per questo motivo molte istituzioni propongono ambienti per lo sviluppo e la valutazione di agenti per aste elettroniche. Ad esempio l'Università del Michigan ha sviluppato la piattaforma *AuctionBot*. Questa piattaforma fornisce agli agenti un ambiente formato da aste elettro-

niche simultanee in cui sono vendute diverse categorie di beni. Utilizzando la piattaforma *AuctionBot*, l'Università del Michigan ha organizzato la *Trading Agent Competition* (TAC), giunta attualmente alla sua seconda edizione. Lo scopo della TAC è stimolare la ricerca nel campo degli agenti software di commercio elettronico (*trading agents*), dando rilievo allo sviluppo di strategie di successo, che permettano di massimizzare i profitti in un ambiente costituito da mercati virtuali. La competizione permette di valutare il comportamento degli agenti all'interno di mercati competitivi e di studiare come gli agenti interagiscano con le norme che regolano i mercati. Tutto ciò fornisce la possibilità di migliorare la progettazione dei meccanismi che governano i mercati. La competizione è anche un banco di prova delle tecnologie che, in questo ambito, sono ancora in una fase sperimentale, come l'utilizzo di agenti autonomi per interagire in mercati complessi. La volontà dell'Università del Michigan è stata quindi di definire un problema che si ponga come punto di riferimento per lo studio degli agenti software di commercio elettronico.

Nella TAC, ogni partecipante è un agente di viaggio che ha come obiettivo l'allestimento di pacchetti di viaggio. Ogni agente agisce per conto di otto clienti che esprimono le loro preferenze riguardo a vari aspetti del viaggio (il volo, il soggiorno e l'intrattenimento). Lo scopo dell'agente è massimizzare la soddisfazione dei suoi clienti, cercando di minimizzare le spese. I partecipanti propongono agenti software per allestire in modo automatico pacchetti di viaggio, interagendo in tre mercati simulati. Ogni mercato vende un tipo di bene: biglietti aerei, prenotazioni per stanze d'albergo e biglietti per assistere ad alcune attrazioni. Per rendere le cose più interessanti, ogni categoria di beni viene scambiata in aste differenti con regole diverse. Una difficoltà ulteriore deriva dall'interdipendenza dei beni: ad esempio la prenotazione delle stanze d'albergo deve corrispondere alle date di arrivo e partenza dei voli di cui si possiedono i biglietti. Alla fine della sessione di gioco, gli agenti ricevono un punteggio che dipende da quanto sono riusciti a rispettare le preferenze di ogni cliente e da quanto hanno speso per i beni. Lo scopo dell'agente è massimizzare il guadagno che dipende dal livello di soddisfazione degli otto

clienti.

Nell'edizione di quest'anno, hanno partecipato ventotto agenti provenienti da università, laboratori di ricerca industriale e società commerciali di tre continenti.

## 1.1 Obiettivi

L'obiettivo della tesi è lo studio di problematiche relative alla progettazione e la realizzazione di agenti autonomi per aste elettroniche. Per affrontare concretamente il problema abbiamo realizzato l'agente *polimi\_bot*, sviluppato per interagire con la piattaforma *AuctionBot*. Per lo sviluppo del nostro agente, abbiamo scelto di utilizzare una strategia euristica, privilegiando in questo modo la semplicità di implementazione pur mantenendo comunque un certo grado di flessibilità nei comportamenti attraverso l'utilizzo di parametri. L'implementazione della strategia è stata realizzata definendo un'opportuna architettura basata su regole che definiscono il comportamento dell'agente in risposta agli eventi che si verificano durante la competizione. Per verificare la qualità della strategia implementata, abbiamo fatto partecipare il nostro agente *polimi\_bot* alla *Trading Agent Competition 2001* (TAC-01). Per velocizzare i tempi di realizzazione e per poter verificare la correttezza della strategia su cui si basa il nostro agente, abbiamo deciso inizialmente di implementare le regole direttamente in C++, rendendo l'agente meno flessibile ma facilitando il compito di trovare e correggere gli errori di implementazione. In seguito abbiamo iniziato a trasformare il nucleo dell'agente, utilizzando un'architettura a regole vera e propria. In questo modo è possibile sperimentare in modo semplice nuove strategie, in previsione della partecipazione alla prossima edizione della TAC. I nostri obiettivi sono duplici. Innanzitutto abbiamo voluto valutare le prestazioni del nostro agente allo scopo di verificare se un'euristica sia sufficiente per ottenere buone prestazioni oppure se sia necessario ricorrere a strategie più complesse basate su metodi di ottimizzazione o su algoritmi di Apprendimento Automatico. In secondo

luogo abbiamo voluto verificare, mediante un'analisi dei risultati effettuata tramite metodi statistici, quanto sia efficace questo tipo di competizione nel valutare le differenti strategie sviluppate per competere nel campo delle aste elettroniche.

## 1.2 Struttura della tesi

Questo lavoro è strutturato nel modo seguente.

Nel Capitolo 2, presentiamo una panoramica sullo stato dell'arte delle tecnologie relative al commercio elettronico, attraverso l'analisi di alcuni modelli del commercio elettronico e delle aste elettroniche. Presentiamo inoltre alcune delle sempre più numerose piattaforme che permettono la gestione di aste su Internet.

Nel Capitolo 3, presentiamo la *Trading Agent Competition* (TAC), descrivendo in primo luogo le regole della competizione, eseguendo un'analisi del problema generale della TAC e descrivendo la piattaforma *AuctionBot*, che è stata utilizzata per lo svolgimento delle gare.

Nel Capitolo 4, presentiamo l'agente *polimi\_bot* che è stato sviluppato all'interno del Laboratorio di Intelligenza Artificiale e Robotica per partecipare all'edizione 2001 della *Trading Agent Competition*. Dopo aver illustrato la struttura dell'agente e averne descritto i componenti principali, spieghiamo come l'agente affronta e cerca di risolvere i vari problemi posti dalla competizione. Descriviamo quindi le procedure utilizzate per l'implementazione delle strategie.

Nel Capitolo 5, descriviamo in dettaglio lo svolgimento dell'edizione di quest'anno della TAC. Introduciamo i partecipanti alla competizione, riportiamo i risultati di tutte le fasi della gara e illustriamo le strategie di alcuni agenti che si sono qualificati per la fase finale della competizione.

Nel Capitolo 6, analizziamo approfonditamente i risultati della competizione utilizzando metodi statistici. Dopo aver discusso le motivazioni che ci hanno portato alla scelta del metodo, procediamo a illustrare i risulta-

ti prodotti dall'analisi che abbiamo effettuato. In questo modo cerchiamo di stabilire quali agenti si siano comportati meglio e cerchiamo di valutare le prestazioni reali dell'agente *polimi\_bot*. Inoltre cerchiamo di valutare l'efficacia della competizione nella promozione di strategie realmente valide.

## 1.3 Contributi originali

La tesi contiene i seguenti contributi originali:

- un'analisi delle dinamiche dei mercati della *Trading Agent Competition 2001* (TAC-01) mediante l'applicazione di *modelli lineari*, al fine di ottenere adeguati *predittori* dell'andamento dei prezzi;
- lo sviluppo di un agente per la competizione basato su strategia euristica, implementata mediante la definizione e lo sviluppo di un'architettura a regole;
- l'analisi approfondita dei risultati della TAC-01 allo scopo di valutare la rilevanza di questo tipo di competizione al fine di evidenziare le strategie migliori. L'utilizzo dell'*analisi della varianza* seguita da una serie di *procedure di confronto multiplo* ci ha permesso di osservare che questo tipo di competizione ha evidenziato un gruppo formato dai migliori agenti, ma non è riuscita a individuare una singola strategia vincente.



# Capitolo 2

## Stato dell'arte

In questo capitolo presentiamo una panoramica sullo stato attuale delle tecnologie relative al commercio elettronico. In particolare analizziamo alcuni modelli del commercio elettronico e più in dettaglio illustriamo le aste elettroniche con le relative applicazioni. Infine presentiamo alcune delle numerose piattaforme che permettono la gestione di aste su Internet.

### 2.1 Commercio elettronico

Il commercio elettronico viene definito sulla base della Comunicazione della Commissione Europea al Parlamento europeo, al Consiglio, al Comitato economico e sociale e al Comitato delle regioni COM (97)157 “Un’iniziativa europea in materia di commercio elettronico”:

Il commercio elettronico consiste nello svolgimento di attività commerciali e di transazioni per via elettronica e comprende attività diverse quali: la commercializzazione di beni e servizi, la distribuzione di contenuti digitali, l’effettuazione di operazioni finanziarie e di borsa, gli appalti pubblici e altre procedure di tipo transattivo delle Pubbliche Amministrazioni.

Il commercio elettronico non si esaurisce quindi nella semplice conduzione della transazione bensì può anche abbracciare altre fasi ed altri aspetti che

riguardano un rapporto commerciale: da quelli di ricerca ed individuazione dell'interlocutore, a quelli di trattativa e negoziazione, da quelli degli adempimenti e delle scritture formali a quelli dei pagamenti e della consegna fisica o elettronica del bene o servizio acquistato o venduto [17].

### 2.1.1 Storia

I computer sono stati usati per la prima volta in applicazioni commerciali negli anni Sessanta, negli Stati Uniti. Grazie alla piattaforma ERMA (*Electronic Recording Machine Accounting*), le banche riuscirono ad automatizzare le procedure relative agli assegni, che stavano diventando troppo numerosi per essere gestiti manualmente. In seguito, negli anni Settanta e Ottanta l'utilizzo dei computer nell'ambito commerciale è cresciuto enormemente. Venne infatti introdotta l'infrastruttura EDI (*Electronic Data Interchange*), che permetteva alle aziende di scambiarsi informazioni in modo standard. In questo modo venne praticamente eliminato l'intervento umano nelle comunicazioni interaziendali [19]. Il grande salto di qualità è però avvenuto negli anni Novanta con l'introduzione del *World Wide Web*, che ha permesso l'estensione dell'utilizzo dei computer in un numero vastissimo di applicazioni commerciali, sia tra aziende (*Business to Business*), sia tra aziende e pubblico (*Business to Consumer*), sia tra persone fisiche (*Consumer to Consumer*). Nella seconda metà degli anni Novanta, si verificarono i successi eccezionali delle quotazioni al NASDAQ di società come *Netscape*, *Yahoo* e *Amazon*. Di conseguenza le *start-up* (nuove aziende che tipicamente si basano su Internet, con grandi potenziali di crescita, ma ad alto rischio) riuscirono a ottenere molto facilmente i finanziamenti da parte degli investitori. Tuttavia, il principale obiettivo di molte di queste nuove compagnie consisteva solamente nel cercare di attirare un gran numero di utenti verso i loro siti. In questo modo riuscivano a raccogliere molte adesioni da parte degli investitori, al momento della quotazione in Borsa. All'inizio quest'idea funzionò e permise a queste aziende di muovere grossi capitali. Questo enorme flusso di denaro attirò parecchi investitori e moltissime compagnie spesero grosse somme per essere

*on-line*. Bisogna notare però, che già all'epoca si potevano rilevare alcuni segnali di avvertimento del fatto che la situazione sarebbe cambiata. Infatti molte grosse compagnie cominciavano ad avere dei problemi a consegnare i prodotti al consumatore, problemi di magazzino oppure problemi nelle comunicazioni con i fornitori. Inoltre praticamente nessuna aveva in previsione attivi di bilancio per diversi anni.

Con il rallentamento dell'economia su scala globale, avvenuto nel 2001, è cominciato un pesante ridimensionamento delle aziende basate sul *Web*. La maggior parte delle società nate esclusivamente come catalizzatori per la raccolta di capitali, è fallita e anche molte aziende quotate in Borsa hanno perduto una grossa parte del loro valore iniziale. Tuttavia, quelle società che hanno operato senza debiti troppo grossi e che erano vicine al pareggio di bilancio, sono ancora attive e probabilmente sopravvivranno alla recessione. Infatti quello che si è imparato in questi anni è che, per avere successo *on-line*, bisogna seguire le stesse regole che garantiscono il successo nelle aziende tradizionali: è necessario pianificare l'attività, in modo da produrre un utile [20]. C'è infatti un selezionato gruppo di imprese *dot-com* che malgrado la situazione contingente, continua a ottenere notevoli crescite di fatturato e al tempo stesso genera profitti. Queste società hanno due cose in comune: operano in mercati orientati al consumatore — che in realtà hanno ottenuto benefici dalla economia stagnante — e hanno modelli unici e adatti a raggiungere il successo su Internet.

Tra le società di successo che operano nell'ambito del commercio elettronico, una delle più conosciute è *Amazon.com* [55] (AMZN). Tuttavia non è quella che ha ottenuto i risultati migliori; infatti *eBay* [56] (EBAY) ha più utenti registrati e ha effettuato transazioni per un valore che ha raggiunto il triplo del suo fatturato nel secondo trimestre del 2001. Inoltre *eBay* è ora la più popolare piattaforma per il commercio elettronico e ormai alle sue aste si vendono milioni di articoli. Secondo la *Nielsen NetRatings*, le aste *on-line* muovono circa il 10% di tutto quello che viene speso nel campo del commercio elettronico, rendendole la terza più importante categoria dopo libri e

vestiti. In questo settore *eBay* è decisamente il leader, con una percentuale di mercato stimata del 64%. Questa azienda, contrariamente a quello che si può pensare, ha ricavato benefici dal rallentamento economico globale. Infatti questa situazione ha convinto i compratori a cercare le migliori occasioni e i venditori a ricevere denaro da tutto il mondo. Il successo di *eBay* è anche imputabile al fatto che questa società non vende al dettaglio e quindi non ha problemi di spese per evadere gli ordini e per effettuare le consegne. Forse ancora più importante è il fatto che non ha assolutamente spese di magazzino.

Un'altra categoria che ha avuto successo nel commercio elettronico, è quella che opera nel settore dei viaggi. Ad esempio, *Expedia* (EXPE), *Travelocity* (TVLY) e *Priceline.com* (PCLN) hanno ottenuto ottimi risultati nei primi due trimestri del 2001. Anche questo settore infatti trae benefici da spese operative relativamente basse, ovvero spese di magazzino e di gestione degli ordini praticamente nulle. Inoltre come *eBay*, anche chi opera nei viaggi *on-line* è stato favorito da quello che si può definire *network effect*, dove l'aggiunta di nuove alleanze con compagnie aeree, alberghi e noleggiatori di auto attrae ulteriori potenziali clienti; l'aumento di traffico e di transazioni, a sua volta, porta a migliori rapporti con i partner e il ciclo si ripete [26]. In realtà gli eventi dell'11 settembre hanno peggiorato molto la situazione di tutti gli operatori di viaggio e questo ha influito anche sulle società citate. Tuttavia questa circostanza è assolutamente fuori dell'ordinario e il forte peggioramento dei risultati di queste società, nell'ultimo trimestre, non è imputabile a una cattiva gestione o a un fallimento del commercio elettronico nel settore dei viaggi.

La situazione italiana ha vissuto gli stessi sviluppi degli Stati Uniti, seppure con i consueti ritardi dovuti alla maggior arretratezza dal punto di vista tecnologico. Infatti si è verificata anche da noi l'esplosione delle attività commerciali *on-line* avvenuta negli ultimi due anni (1999-2000), quindi con alcuni anni di ritardo rispetto agli USA; in seguito anche in Italia il settore ha pesantemente risentito del periodo di recessione e della situazione difficile

verificatasi negli Stati Uniti. Infatti questo settore economico risente, molto più dei settori tradizionali, della dimensione globale dell'economia.

### 2.1.2 Tendenze

In un prossimo futuro, Internet sarà ancora considerata come una fonte di guadagno per le aziende, ma si avrà una visione più realistica delle effettive possibilità del mezzo. Internet sarà molto utilizzata dalle società tradizionali per poter espandere il loro giro d'affari. Infatti le vendite *on-line* sono in crescita, anche durante questo periodo di rallentamento dell'economia e anche l'utilizzo di Internet in generale sta crescendo. Con l'aumento del bacino di utenza di Internet e la diffusione di connessioni veloci, le società che gestiscono la pubblicità saranno incoraggiate a investire sempre di più sul nuovo mercato. Quindi probabilmente, in futuro la maggior parte delle aziende senza una strategia riguardo al *Web* non avrà successo [20]. L'*e-commerce* infatti si sta sempre più diffondendo, anche se con tassi di incremento diversi nelle varie regioni del pianeta, malgrado le oscillazioni dell'economia globale. I principali fattori della crescita dell'*e-commerce* sono: la diffusione di Internet, la diversificazione delle modalità di accesso, la crescente liberalizzazione dei mercati delle telecomunicazioni, la spinta della concorrenza, la necessità di ridurre i costi delle transazioni e la volontà di presidiare più efficacemente i mercati. Le imprese sono sempre più consapevoli delle potenzialità dell'*e-commerce* e della necessità di essere presenti in Internet. Il valore complessivo delle transazioni *business to business* è ancora di gran lunga superiore a quello degli scambi *business to consumer*; quest'ultimo peraltro registra tassi di crescita superiori alle aspettative non solo per i prodotti di intrattenimento, ma anche per i beni semidurevoli. Il modello di comunicazione in rete sta imponendo, inoltre, alla comunicazione stessa nuove caratteristiche: la necessità di attirare con una forte componente di servizio e la possibilità di trarre reciproco vantaggio dall'interazione con il consumatore finale. Anche il quadro normativo è in corso di definizione per quanto concerne l'*e-commerce*,

sebbene, allo stato attuale, esso appare incompleto. Tuttavia ciò non deve rappresentare un ostacolo a chi voglia operare nel mercato virtuale [18].

## 2.2 Modelli per il commercio elettronico

Il commercio elettronico è costituito da operazioni che coinvolgono imprese e individui. Si parte quindi dalla distinzione di base tra imprese e individui, per comprendere le diverse tipologie riunite sotto l'etichetta unica di *e-commerce*, queste tipologie mutano anche radicalmente a seconda dei vari attori in campo e delle diverse modalità di rapporto che si instaurano tra loro [24]. *Business to business* (B2B) e *business to consumer* (B2C) sono universalmente riconosciute come le due forme principali assunte dal commercio elettronico. Si tratta più propriamente di due etichette sotto le quali è lecito ricondurre praticamente tutte le relazioni che è possibile instaurare con l'*e-commerce*: relazioni che hanno luogo per lo più proprio tra impresa e impresa oppure tra impresa e consumatore. È bene però precisare subito che esistono anche altre tipologie, generalmente riconducibili alle due principali, ma con caratteristiche più specifiche. Pensiamo ad esempio al commercio elettronico tra consumatori finali, il cosiddetto *consumer to consumer* (C2C), dove il consumatore è protagonista sia dal lato dell'offerta (proponendo la vendita dei propri prodotti) che da quello della domanda: l'esempio più tipico cui fare riferimento è quello delle aste *on-line*. Un altro caso è quello del cosiddetto *intra-business*: è questo il caso che coinvolge un'azienda con sedi distribuite sul territorio o un insieme di aziende appartenenti allo stesso gruppo. In tali situazioni le esigenze di scambio dei beni e servizi all'interno delle unità della struttura possono essere soddisfatte attraverso la rete ed essere accompagnate da una rendicontazione economica. Le principali caratteristiche del commercio intra-aziendale sono molto simili a quelle del B2B con la restrizione che di norma il gruppo di soggetti coinvolti è rigorosamente chiuso, la classificazione dei prodotti è già data e la transazione economica passa in secondo piano (può essere eseguita a distanza di tempo, a consunti-

vo) [21]. Altro esempio possibile è il cosiddetto *consumer to business* (C2B), un modello ancora poco noto ma probabilmente destinato a vasta diffusione, consistente in una serie di iniziative commerciali (come quelle avviate da *Priceline*) dove è il consumatore a proporre il prezzo per il prodotto a cui è interessato, provocando così l'offerta dei diversi venditori che propongono il bene o servizio desiderato [24].

### 2.2.1 Business to business

L'economia possiede ormai una dimensione globale, non più basata soltanto sull'accumulo di beni, ma sempre più sull'offerta di servizi a struttura integrata e reticolare. In un'economia delle reti, l'interesse individuale determina un corso del tutto diverso. È infatti integrando l'attività economica di ciascuno in un reticolo di relazioni reciproche mutuamente vantaggiose, pensato per ottimizzare lo sforzo collettivo, che il successo di ogni impresa diviene più probabile, grazie a quella che alcuni hanno battezzato *win win strategy* [22].

Il fenomeno è in buona parte già avviato e, stando agli analisti, destinato soltanto a ulteriore espansione. Già oggi infatti esiste un grandissimo numero di interconnessioni tra aziende, fornitori, partner e clienti, tra chi produce materialmente una merce e chi la distribuisce, tra tutti gli attori, insomma, collocati lungo l'intera catena di creazione del valore di un prodotto. Proprio la complessità di tali relazioni ha fatto balzare in primo piano la necessità di migliorare l'assetto logistico ai fini di rendere quanto più efficace possibile la strategia competitiva dell'azienda. Di qui gli evidenti vantaggi offerti dal modello B2B che indica tutte quelle iniziative tese a integrare l'attività commerciale di un'azienda con quella dei propri clienti o dei propri fornitori dove però il cliente non sia anche il consumatore finale del bene o del servizio venduti ma un partner attraverso il quale si raggiungono i consumatori finali [23].

Le innumerevoli transazioni, che sempre più costituiscono il cuore dell'economia mondiale, distribuite lungo tutto l'arco della catena produttiva,

stanno insomma trasferendosi in blocco sulla Rete, con significative riduzioni dei costi e incrementi consistenti dei margini di guadagno: ecco perché svariate e autorevolissime stime indicano proprio il B2B come il settore del commercio elettronico che sembra destinato al maggior sviluppo in un futuro talmente prossimo da confondersi già con il presente [24].

Questo nuovo approccio ha al suo centro un'entità la cui importanza è ormai indiscutibile in pressoché tutti i comparti del moderno assetto sociale: l'informazione, che deve muoversi parallelamente al flusso logistico e in entrambe le direzioni. È proprio la movimentazione delle informazioni a rivelarsi spesso il punto critico nella fornitura di un servizio di alta qualità: è stato infatti calcolato che la quantità di documentazione richiesta per spostare delle merci da una parte all'altra del globo richiede frequentemente tempi più lunghi della stessa movimentazione fisica. Da questo punto di vista, Internet può far conseguire alle aziende grandi vantaggi, rendendo possibile la trasmissione di informazioni, dati e documenti in tempo reale [21].

### 2.2.2 Business to consumer

È la forma più nota, anche se non la più diffusa, di *e-commerce*, e indica sostanzialmente la fornitura di beni e servizi direttamente all'utente finale. In genere avviene tramite un sito *web* dove chi vende mette a disposizione di chi compra un carrello della spesa virtuale in cui inserire tutti gli oggetti da acquistare, tutti "osservabili" da cataloghi appositi, più o meno sofisticati a seconda del sito e dei beni o servizi che l'impresa propone. Il compratore può aggiungere o eliminare i prodotti che vuole, modificarne le quantità, controllare tanto il valore complessivo quanto quello di dettaglio, visionare tutto il contenuto, ecc. L'atto finale della transazione consiste nella compilazione, da parte del compratore, di un *form* con i suoi dati e gli estremi della carta di credito (questo è uno dei punti critici che sollevano il problema, molto discusso, della sicurezza di queste transazioni). Se tutto è andato come doveva, l'operazione si conclude con l'arrivo, in tempi brevi, della merce ordinata direttamente a casa del cliente finale.

I vantaggi, sia per chi vende sia per chi compra, sono abbastanza evidenti: l'azienda può completare le proprie opzioni di vendita estendendo sensibilmente i suoi canali distributivi a costi contenuti grazie alla pubblicazione *on-line*, mentre il cliente può fare acquisti, o prenotare viaggi, o svolgere tutta una serie di operazioni finanziarie senza nemmeno uscire di casa. Naturalmente ci sono anche dei rischi, specie per le imprese: non possedere un sistema distributivo all'altezza delle promesse, soprattutto per quanto riguarda i tempi di distribuzione, può anche danneggiare il fatturato complessivo, incrinando la fiducia del cliente e dunque allontanandolo dal marchio anziché fidelizzarlo. Inoltre, bisogna tener conto del fatto che effettuare vendite dirette molto spesso può essere anche solo un'attività subordinata alla raccolta di informazioni utili a individuare le abitudini di acquisto del cliente, o a instaurare con lui un rapporto di lungo periodo, in modo da incrementare le vendite sia tradizionali che *on-line* [24].

## 2.3 Modelli d'asta

Un meccanismo di contrattazione è essenzialmente un protocollo entro il quale gli agenti interagiscono per determinare un contratto. Le aste costituiscono un caso generale di questi protocolli, come caratterizzato nella definizione ufficiale espressa da McAfee e McMillan [33]:

Un'asta è un'istituzione di mercato con un esplicito insieme di regole, che determinano l'allocazione delle risorse e i prezzi sulla base delle offerte dei partecipanti al mercato.

Un'osservazione importante deve essere fatta prima di discutere dei vari formati d'asta: in generale tutti coloro che partecipano a un'asta hanno una *valutazione privata* dell'oggetto che intendono comprare. Bisogna inoltre considerare che il compratore tende a mantenere il più possibile riservata questa informazione. Infatti non ci sarebbe ragione di rivelare al venditore quanto si è disposti a spendere, subito all'inizio. Oltre a questo bisogna considerare che la ragione dell'acquisto potrebbe essere la possibilità di rivendere il bene

per ottenere un guadagno. Quindi oltre alla valutazione privata, il potenziale compratore deve avere una stima delle valutazioni dei futuri clienti. Ogni partecipante avrà perciò informazioni e valutazioni diverse, ma tutti dovranno cercare di indovinare a quale prezzo verrà venduto il bene.

William Vickrey [3] ha stabilito la tassonomia di base delle aste, imposta sull'ordine secondo cui i prezzi sono quotati e sul modo in cui le offerte sono presentate. Vickrey ha individuato quattro principali categorie che rappresentano la quasi totalità delle aste a chiamata singola. Si noti che la letteratura relativa alle aste è quasi completamente in lingua inglese e quindi non esistono traduzioni in italiano, ufficialmente riconosciute, dei nomi dei vari formati d'asta. Dove è necessario verranno quindi riportati anche i nomi in inglese per permettere il riferimento alla letteratura.

### 2.3.1 Asta inglese

L'asta inglese (*English auction*) è una delle più note forme di asta; è anche conosciuta come asta a prezzo crescente (*ascending-price auction*). Paul Milgrom [4] definisce l'asta inglese nella seguente maniera:

Il banditore inizia con il più basso prezzo accettabile — la base d'asta — e continua a sollecitare successivamente offerte più alte dai potenziali clienti fino a che nessuno aumenterà la sua offerta. Quindi il bene è venduto (*knocked down*) al miglior offerente.

Il successo di un'asta inglese, dal punto di vista del venditore, è maggiore quanto più alto è il livello di competitività dei potenziali compratori. Infatti in un ambiente molto competitivo, il gran numero di rilanci farà crescere di molto il prezzo finale. Invece, dal punto di vista del compratore, la chiave del successo è mantenere bassa la quotazione; ciò tipicamente viene ottenuto forzando il prezzo a crescere di un piccolo incremento per volta. Spesso, in questo modo, un cliente riesce ad acquisire un bene per molto meno della sua massima valutazione perché semplicemente riesce ogni volta a rilanciare di poco. In altri termini, il venditore non riceve necessariamente il valore

massimo che i partecipanti sono disposti a pagare e quindi altri tipi di aste possono rivelarsi più remunerative di quella inglese (dal punto di vista del venditore) [5]. Uno svantaggio del sistema inglese è che questo tipo di asta è soggetta alla formazione di gruppi di partecipanti, che si mettono d'accordo in modo da non rilanciare sulle loro offerte, per mantenere basso il prezzo finale (questi gruppi sono chiamati *rings* in letteratura). Inoltre nelle aste di tipo inglese è abbastanza diffuso il pericolo della “maledizione del vincitore” (*winner's curse*), ossia il fenomeno per cui il vincitore paga un bene più del suo valore reale e ciò accade per l'alto livello di competizione (e per l'inesperienza di alcuni partecipanti).

La miglior strategia per un partecipante ad un'asta di tipo inglese è di offrire poco di più della precedente migliore offerta fino a che il prezzo corrente raggiunge la valutazione del partecipante, al che è necessario fermarsi. Questo è il comportamento ottimo perché l'obiettivo del compratore è acquistare un bene se il prezzo è minore di quello che per lui è il valore, ma vuole anche pagare il meno possibile. Se tutti utilizzano questa strategia, le offerte termineranno sempre quando il prezzo avrà raggiunto la valutazione del potenziale compratore con la seconda più alta valutazione (se si considerano aste che vendono una sola unità del bene). Un vantaggio per il compratore è la disponibilità di informazioni: infatti è possibile osservare e sapere se uno dei partecipanti si ritira e a quale prezzo si ritira. Questo rivela molto riguardo alle valutazioni private degli altri concorrenti e permette di riesaminare la propria valutazione man mano che l'asta prosegue. La strategia di un partecipante è espressa dalla serie delle sue offerte che è una funzione dei seguenti valori: la sua valutazione, la sua precedente stima delle valutazioni degli altri partecipanti e le offerte precedenti degli altri.

### 2.3.2 Asta a offerta segreta

La principale caratteristica di questo tipo di aste (note in letteratura come *first-price sealed-bid auction*) è che le offerte non vengono rivelate agli altri partecipanti (a differenza dell'asta inglese). Anche in questo caso il vincitore

paga esattamente quanto ha offerto. In generale, ogni partecipante può effettuare una sola offerta e questo significa che la fase di preparazione è molto importante. Generalmente un formato a offerta segreta ha due parti distinte: un periodo in cui i partecipanti propongono le loro offerte e una fase in cui le offerte sono confrontate e il vincitore è determinato (in alcuni casi però il vincitore non è annunciato agli altri). In pratica, se si mette in palio una sola unità, ogni partecipante propone un'offerta, che viene mantenuta segreta agli altri partecipanti, e l'offerta più alta vince. Un'importante osservazione deve essere fatta riguardo alla quantità di beni che sono messi all'asta. Infatti quando più beni sono messi all'asta con queste regole, l'asta prende il nome di “asta discriminatoria” (*discriminatory auction*), perché non tutti i vincitori pagano la stessa somma. In un'asta discriminatoria le offerte sono ordinate dalla più alta alla più bassa e i beni sono assegnati uno per uno, partendo dall'offerta più alta fino all'esaurimento dei beni da vendere. Quindi una caratteristica importante di questo formato è che i vincitori possono (e in genere accade) pagare prezzi differenti.

Dal punto di vista dei partecipanti, un'offerta alta aumenta le probabilità di vittoria, ma abbassa il profitto in caso di vincita. Una buona strategia è di proporre un'offerta vicina al reale valore di mercato del bene. Questa strategia aiuta inoltre a evitare la maledizione del vincitore. È difficile individuare un'unica strategia per questo tipo d'aste, perché un'offerta che massimizza il profitto dipende anche dalle azioni degli altri. Bisogna operare una scelta valutandone le conseguenze: se fare un'offerta alta che porterà più spesso a una vincita oppure fare un'offerta bassa che permetterà di trarre un maggiore guadagno in caso di vincita. La maggior parte dei partecipanti cerca di avvicinarsi con le proprie offerte al valore di mercato del bene messo all'asta.

### 2.3.3 Asta olandese

L'asta olandese (*Dutch auction*) è anche nota come asta a prezzo decrescente (*descending-price auction*) ed è (come l'asta inglese) un'asta in cui le offerte sono rese pubbliche a tutti i partecipanti. In questo tipo di aste, l'offerta

parte da un prezzo estremamente alto ed è progressivamente abbassata fino a quando un compratore non accetta di pagare quel prezzo. Se ci sono più beni a disposizione, l'asta continua fino a quando tutte le unità sono state assegnate. In altre parole il primo vincitore prende il suo premio e paga il suo prezzo e i vincitori successivi pagano meno. È opinione comune che il sistema inglese sia inferiore a quello olandese, riguardo a un fatto: il primo sfrutta la competizione per ottenere un prezzo maggiore, ma rischia che le offerte vengano incrementate di poco e che il vincitore paghi un prezzo inferiore alla sua massima disponibilità. Viceversa, nel sistema olandese, se il compratore con il più alto interesse vuole realmente un bene, allora non può attendere troppo a lungo per fare la sua offerta. Questo significa che potrebbe offrire la sua valutazione privata o un valore prossimo ad essa.

Il problema per il partecipante a un'asta olandese è esattamente lo stesso che affronta il partecipante a un'asta con offerta segreta. Con un certo anticipo, dovrà decidere il massimo valore che è disposto ad offrire. Deve inoltre decidere quando fermare l'asta basandosi sulla sua valutazione privata dell'oggetto e su quello che crede sarà la valutazione degli altri partecipanti. Questo tipo di asta è quindi strategicamente equivalente al tipo di asta a offerta segreta, perché nessuna informazione significativa è rivelata nel corso dell'asta, ma si viene a conoscere tutto solo alla fine, quando è troppo tardi.

#### 2.3.4 Asta di Vickrey

L'asta di Vickrey [3] (nota anche come *uniform second-price auction*) prende il suo nome da William Vickrey, vincitore del Premio Nobel in Scienze Economiche nel 1996, che l'ha classificata negli anni Sessanta. In quest'asta le offerte sono segrete e ogni concorrente ignora quindi, le altre offerte. Il bene è vinto dal miglior offerente, che però pagherà un prezzo pari a quello della seconda offerta più alta. In altre parole il vincitore paga meno di quanto aveva offerto. Questo si riferisce a un'asta che mette in palio una singola unità di un bene, tuttavia nel caso di aste che mettono in vendita più unità, i vincitori pagheranno tutti lo stesso prezzo, corrispondente alla prima offerta

non vincente. Sembrerebbe ovvio quindi, che dal punto di vista del venditore, ci sia minor possibilità di guadagno rispetto ad un'asta inglese, in realtà questo non accade. I partecipanti infatti modificano le loro offerte in base alle diverse regole e in questo caso tenderanno a offrire di più che in altre aste. Infatti nessuno teme di pagare un prezzo troppo alto, perché è meno probabile che si verifichi la maledizione del vincitore. Partecipanti aggressivi sono favoriti nell'ottenere il bene, ma pagheranno un prezzo vicino a quello stabilito dal mercato. Il prezzo finale non dipenderà quindi, dalle azioni dei partecipanti ma solo dalle loro offerte.

Paul Milgrom [4] suggerisce che la strategia dominante, per un partecipante a un'asta con questo formato, è di offrire la propria vera valutazione del bene, perché questa offerta domina tutte quelle di valore inferiore alla valutazione, ma nessuna di quelle di valore superiore. Se il partecipante non segue la strategia ottima e offre meno della sua valutazione diminuisce le sue probabilità di vittoria. Viceversa se offre più della propria valutazione rischia di incorrere nella maledizione del vincitore. Nessuna di queste strategie però, influenza il prezzo pagato in caso di vittoria, perché il prezzo finale non è influenzabile in modo diretto da un singolo compratore. Quando tutti i partecipanti adottano la strategia di offrire il loro vero prezzo, il risultato è che il bene è vinto dal miglior offerente a un prezzo pari alla seconda offerta più alta. L'esistenza di una strategia dominante permette ai concorrenti di determinare la loro offerta senza tenere presenti le azioni degli altri.

Varie ragioni fanno dell'asta di Vickrey un'ottima scelta. Infatti, non dovendo temere la maledizione del vincitore, i partecipanti sono più aggressivi e quindi c'è una maggior possibilità di guadagno da parte del venditore; inoltre questo tipo di asta è considerato meno soggetto a collusioni e formazioni di *rings*. In generale rispetto agli altri formati, l'asta di Vickrey è più semplice, richiede meno tempo per preparare le offerte, è meno costosa e permette una partecipazione maggiore.

### 2.3.5 Asta a doppia chiamata continua

Un'asta a doppia chiamata continua (*Continuous Double-Auction* o CDA) non è un'asta classica dove c'è un solo venditore (non rientra quindi nella classificazione di Vickrey), infatti ogni partecipante può indifferentemente comprare o vendere beni. Una CDA pertanto è più simile a un mercato di tipo azionario. In una CDA classica [6], compratori e venditori sono liberi in ogni momento di annunciare pubblicamente domande e offerte. Ogni compratore può accettare l'offerta di un venditore, e ogni venditore può accettare la domanda di un compratore. Una forma di CDA usata nei mercati reali è il formato "a doppia chiamata continua con coda delle richieste" (*continuous double auction with order queue* o *persistent shout double auction* [7]). In questo tipo di mercato, un partecipante può fare un'offerta (di acquisto o vendita) in ogni momento, ma una volta fatta, l'offerta rimane fino a che il partecipante non decide di modificarla o di rimuoverla oppure fino a che viene accettata.

Le caratteristiche di questo tipo di mercato dipendono dagli andamenti della domanda e dell'offerta. Se un bene è offerto a un certo prezzo, i compratori nel mercato vorranno acquistare un numero di beni, pari alla "quantità richiesta" (*quantity demanded*) al prezzo corrente. In generale, maggiore è il prezzo del bene, minore è la quantità richiesta. Se tracciamo la quantità richiesta rispetto al prezzo (si veda la Figura 2.1), otteniamo una curva decrescente, la curva della domanda  $D$  (*demand curve*). Similmente se un bene sta per essere acquistato a un certo prezzo, i venditori vorranno vendere un certo numero di questi beni: la "quantità fornita" (*quantity supplied*) al prezzo corrente. Disegnando la quantità fornita rispetto al prezzo, possiamo costruire la curva dell'offerta  $S$  (*supply curve*).

Al prezzo determinato dall'intersezione delle curve della domanda e dell'offerta, la quantità offerta è pari alla quantità richiesta. Quindi *tutti* i partecipanti (siano essi compratori oppure venditori) possono concludere una transazione a questo prezzo. Questo prezzo è chiamato "prezzo di equilibrio"  $P_0$  (*equilibrium price*) e il numero di beni scambiati è la "quantità di equi-

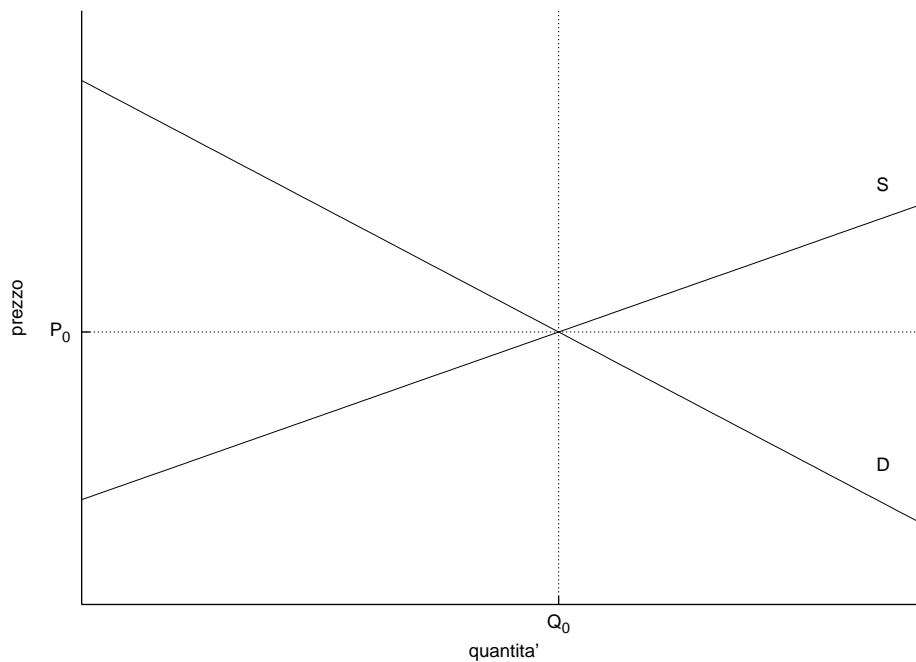


Figura 2.1: Domanda e offerta.

librio"  $Q_0$  (*equilibrium quantity*). In un mercato libero (dove non ci sono interventi esterni) gli scambi tenderanno naturalmente a essere effettuati al prezzo di equilibrio. Se il prezzo corrente è inferiore all'equilibrio, la quantità richiesta sarà maggiore della quantità offerta. Ci sarà quindi "domanda in eccesso" (*excess demand*), che costituisce un incentivo per i compratori ad alzare il prezzo, per assicurarsi che le loro offerte vengano accettate. Similmente, se c'è "offerta in eccesso" (*excess supply*), ci sarà un incentivo per i venditori ad abbassare le loro offerte, per assicurarsi di effettuare qualche scambio. Questo processo auto-correttivo è conosciuto come "determinazione dei prezzi" (*price determination* o *equilibration*). Il meccanismo della CDA è particolarmente efficace nel raggiungimento del prezzo di equilibrio.

Un'analisi di questa efficienza è stata compiuta da Gode e Sunder [15], i quali sostenevano la tesi che alcune strutture di mercato (come le CDA) sono talmente efficienti che, anche se vi partecipano solo agenti con "intelligenza zero" (*zero-intelligence traders*), permettono di raggiungere ugualmente l'e-

quilibrio, con l'unico vincolo per i partecipanti di non fare offerte in perdita. Gli agenti ZI-C (*zero-intelligence constrained*) utilizzati da G. e S. generano infatti le loro offerte a caso da una distribuzione uniforme, limitata dai vincoli di spesa per il bene che si sta trattando. Tuttavia Cliff e Bruten [16] dimostrano che in generale, il valore atteso dei prezzi di una transazione effettuata da agenti ZI-C sarà differente dal prezzo di equilibrio. Infatti la conclusione a cui erano giunti G. e S. era dovuta al fatto che avevano analizzato una situazione particolare, altamente simmetrica. A causa della simmetria, il valore atteso della funzione di densità di probabilità (PDF), ottenuta dalla combinazione della PDF del venditore e di quella del compratore, coincideva con il reale valore di equilibrio del mercato.

### 2.3.6 Asta combinatoria

Le aste di tipo combinatorio (*combinatorial auction*) sono state ideate negli ultimi anni (neanche loro rientrano nella classificazione di Vickrey), per venire incontro a determinate esigenze, soprattutto per quel che riguarda il contesto B2B.

In un'asta combinatoria i partecipanti possono fare offerte su combinazioni di beni. Questo permette di esprimere la complementarità dei beni. Infatti alcuni beni hanno un certo valore se considerati in combinazione con determinati altri. Come conseguenza la somma del valore dei beni presi singolarmente è diversa dal valore dei beni considerati in gruppo (non vale l'additività). In questo tipo di aste il problema di trovare l'insieme delle offerte vincenti che massimizzano il guadagno — il problema della determinazione del vincitore (*Winner Determination problem* o WD) — è totalmente affidato all'organizzatore dell'asta. Invece nel caso di aste simultanee la complessità è lasciata ai compratori. Tuttavia è semplice mostrare come i due approcci siano equivalenti. Nel caso di aste simultanee, dato un insieme di beni acquisiti bisogna scegliere i pacchetti, tra tutti quelli che si possono costruire, che daranno il maggior ricavo, ossia che massimizzeranno la funzione utile del compratore. Nel caso di un'asta combinatoria invece avremo un insieme di offerte su diver-

se combinazioni di beni (pacchetti) e da questo insieme è necessario estrarre il sottoinsieme che ha il valore complessivo massimo (che massimizza quindi la funzione utile del venditore) e che costituisce una soluzione *ammissibile* del problema (ad esempio non deve includere più pacchetti che contengono lo stesso bene).

Trovare la soluzione ottima è però, un problema NP-completo [8] e per questo motivo le aste combinatorie non sono ancora molto utilizzate: ad esempio la *Federal Communication Commission* ha considerato la possibilità di indire un'asta combinatoria per assegnare le bande per le frequenze dedicate alle telecomunicazioni, tuttavia ha scartato l'opzione per problemi di intrattabilità. In seguito però, sono stati proposti numerosi metodi per trovare la soluzione ottima, cercando di minimizzare i tempi impiegati nella sua ricerca. Ad esempio uno di questi metodi è quello proposto da Tuomas Sandholm [9]. La ricerca viene effettuata sul grafo delle allocazioni, generato in modo da evitare cicli e vertici ridondanti; inoltre si cerca di sfruttare il fatto che lo spazio delle offerte è sparso: ciò implica che anche le allocazioni da analizzare sono sparse. L'algoritmo controlla, costruendola, ogni allocazione che ha un valore positivo, esattamente una volta e non costruisce le altre allocazioni. Quindi a differenza della programmazione dinamica, l'algoritmo genera solo quelle parti dello spazio effettivamente popolate da offerte. Lo svantaggio di questo approccio è che il tempo della ricerca dipende dalle offerte analizzate. Dopo aver generato il grafo delle allocazioni, viene utilizzato IDA\* (*Iterative Deepening A\** [10]) come algoritmo di ricerca, che è risultato due ordini di grandezza più veloce del semplice DFS (*Depth-First Search*), e grazie all'utilizzo di diverse regole di pre-elaborazione, che permettono di potare ulteriormente il grafo, questo approccio si è rivelato significativamente più veloce degli algoritmi disponibili in precedenza, rendendo possibile la gestione di aste combinatorie di un gran numero di beni e offerte.

In seguito lo stesso Sandholm in collaborazione con Subhash Suri ha proposto un miglioramento dell'algoritmo sopra descritto. In questa versione dell'algoritmo [11], viene utilizzato un metodo di ricerca più sofisticato per

la determinazione del vincitore, includendo miglioramenti strutturali che riducono le dimensioni dell'albero, strutture dati più veloci, e ottimizzazioni che comprendono la possibilità di identificare e risolvere casi trattabili speciali. Inoltre vengono presentati dei casi più generali come la possibilità di trattare più unità di uno stesso bene.

Altre soluzioni impiegano l'uso del classico algoritmo di *Branch and Bound* cercando di velocizzarlo con l'uso di euristiche e di buone funzioni di valutazione dei nodi [12]. Mentre altri propongono soluzioni approssimate e quindi molto più veloci, che forniscono un estremo superiore e un estremo inferiore della soluzione ottima [13]. Oppure sempre nel campo di soluzioni approssimate, Craig Boutilier e Holger Hoos [29] propongono un nuovo algoritmo di ricerca locale stocastica, *Casanova*, da applicare al problema della determinazione del vincitore. Questo approccio permette di trovare soluzioni di alta qualità (anche ottime) molto più velocemente dei metodi ottimi proposti in precedenza (fino a diversi ordini di grandezza), in particolare per problemi di grandi dimensioni. Inoltre gli autori propongono un linguaggio logico per esprimere naturalmente le offerte combinatorie, nel quale una singola offerta logica corrisponde a un grande numero (spesso esponenziale) di offerte reali. Su questi problemi, *Casanova* si comporta molto meglio dei metodi sistematici.

## 2.4 Aste elettroniche

Nell'evoluzione del commercio elettronico, le aste stanno diventando sempre più importanti, sia per le transazioni B2B sia per gli acquisti da parte dei consumatori.

### 2.4.1 Applicazioni

Come risultato della crescente popolarità di questa forma di commercio elettronico, sempre più imprese stanno offrendo siti dove svolgere aste. Per questo motivo, se si vuole acquistare un bene particolare, ci sono spesso diverse aste

che offrono quell'articolo. Se si vuole ottenere il miglior prezzo, bisogna quindi tenere costantemente sotto controllo tutte queste aste con un *browser web* e piazzare le offerte in modo appropriato. È necessario inoltre, fare attenzione a non acquistare più unità dello stesso bene. Se il numero di aste da seguire è molto grande, il compito risulterà gravoso e richiederà molto tempo da dedicarvi. In più, se si vuole comprare più di un bene (come accade spesso nel caso del B2B), il compito può diventare quasi impossibile da svolgere manualmente [25]. Perciò coloro che mantengono le aste, stanno iniziando a fornire strumenti di supporto per facilitare questo impegno. Sono stati sviluppati strumenti di ricerca come *Auction Beagle* [57] che permette di localizzare e controllare le aste che offrono specifici beni, eliminando il problema di dover continuamente cambiare sito. Altri servizi forniscono informazioni sulle tendenze dei prezzi di beni molto scambiati, per aiutare a decidere la massima offerta. Questi strumenti aiutano, ma rimangono ancora inadeguati per quel che riguarda ad esempio, il contesto del B2B, in quanto non rendono ancora sufficientemente automatizzato il compito [25].

Alcuni siti, dove si svolgono aste inglesi, forniscono alcuni semplici agenti per fare offerte in modo automatico. Questi agenti risiedono sulle macchine che ospitano il sito e fanno offerte al posto del cliente. Tipicamente bisogna inserire il massimo prezzo che si è disposti a pagare e l'agente piazzerà l'offerta minore possibile (offrirà quindi il prezzo attuale più l'incremento minimo possibile stabilito dalle regole). Tuttavia questi agenti non sono capaci di partecipare in aste multiple e quindi, se si utilizza uno di questi agenti, si è obbligati all'acquisto (se l'offerta risulterà vincente). Localmente, si potrà ottenere il più basso prezzo possibile, ma difficilmente da un punto di vista globale, l'asta a cui si partecipa si rivelerà essere quella dove si ottiene il prezzo migliore. Inoltre l'uso di questi agenti costringe a rivelare a chi gestisce l'asta il massimo prezzo che si è disposti a pagare; ciò potrebbe portare a situazioni in cui questa informazione è usata per truffare i clienti (poiché chi mantiene i siti dove si svolgono le aste, riceve in genere una commissione proporzionale al prezzo pagato) [25].

*FreeMarkets* [58] è un esempio di applicazione rivolta al contesto B2B. *FreeMarkets* è una delle aziende leader nella fornitura di soluzioni basate su software per *e-sourcing* — il processo di identificazione delle opportunità, valutazione delle risorse, negoziazione dei contratti e gestione continuativa delle relazioni con i fornitori e di tutto quello che permette di raggiungere i traguardi della propria azienda — e servizi. Questa società fornisce tutti gli strumenti di cui hanno bisogno le imprese per migliorare i loro processi di *sourcing* e permette di ridurre le spese, effettuando il *sourcing on-line*. Inoltre permette di identificare fornitori di qualità e di partecipare a mercati *on-line* per un'ampia gamma di beni e servizi.

### 2.4.2 Piattaforme

Sono disponibili su Internet numerose piattaforme che permettono di gestire aste e mercati elettronici anche molto complessi, rispetto alle semplici aste (quasi tutte di tipo inglese), disponibili sui siti più noti. In generale queste infrastrutture sono orientate all'utilizzo di agenti per automatizzare, almeno in parte, l'acquisizione o la vendita di beni e servizi. Le piattaforme descritte in questo paragrafo sono solo una parte di quelle realmente disponibili, infatti anche questo settore del commercio elettronico è in fase di espansione e il numero delle infrastrutture di questo tipo è destinato a crescere in brevissimo tempo.

Ad esempio, *Agent-Based Market Space* [59] è un'applicazione di commercio elettronico mediata da agenti. Lo scopo è quello di permettere l'automazione del commercio elettronico, incentrato sull'interazione in un mercato (come ricerca, negoziazione, risoluzione delle transazioni), con l'utilizzo di agenti. Ogni partecipante nel mercato (qualsiasi sia il suo ruolo) ha un agente che automatizza le interazioni parzialmente o totalmente. L'approccio utilizzato è quello di sviluppare un'infrastruttura basata su agenti, fondata su modelli d'informazione e d'interazione semplici. L'unità fondamentale dell'informazione è il contratto (rappresentato da documenti strutturati). Il modello d'informazione consiste in un numero di interazioni possibili (atomiche), che

insieme definiscono il linguaggio d'interazione. I seguenti tipi di messaggio formano il linguaggio: Chiedi (*Ask*), Annuncia (*Tell*), Contratta (*Negotiate*), Offri (*Offer*), Accetta (*Accept*) e Rifiuta (*Decline*). Ciò permette di gestire interazioni complesse come la possibilità di pubblicizzare, di effettuare ricerche, di contrattare e di concludere di accordi commerciali.

Un altro progetto è rappresentato da *AuctionBot* [60] (si veda la Sezione 3.4), che è un *Internet server* multiuso per aste, sviluppato all'Università del Michigan. La piattaforma *AuctionBot* può essere utilizzata per creare aste automatizzate su Internet, con la possibilità di specificare le proprie preferenze sul tipo d'asta, oppure può essere utilizzata per fare offerte in aste già esistenti, create su questa piattaforma.

L'Università del Michigan ha anche sviluppato un sistema aperto, basato sulla piattaforma *AuctionBot*, che permette a terze parti di costruire e integrare i loro propri agenti in un mercato di beni e servizi relativi all'informazione. Il progetto UMDL (*University of Michigan Digital Library* [32]) è un esempio di sistema complesso dove gli agenti che rappresentano utenti, contenuti e servizi della biblioteca digitale, comprano e vendono beni e servizi attraverso aste (in particolare vengono utilizzate CDA: si veda la Sezione 2.3.5). Gli agenti possono unirsi alle aste oppure lasciarle in ogni momento e le transazioni avvengono continuamente.

Il progetto *FishMarket* [61] studia come costruire istituzioni, che hanno agenti come intermediari, tali da rivelarsi utili per il commercio elettronico. *FishMarket* è una casa d'aste elettronica, basata sulle convenzioni commerciali del mercato del pesce tradizionale, dove agenti di complessità arbitraria possono fare scambi su una rete affidabile, sotto condizioni eque, robuste e realistiche.

Ancora, *Bazaar* [62] propone di trasformare il commercio elettronico su Internet, fornendo un'infrastruttura aperta per gestire mercati, dove agenti di commercio di varia complessità possono scambiarsi beni e servizi per conto dei loro proprietari. Questo è stato uno dei primi lavori, che ha aperto la strada agli sviluppi a cui stiamo assistendo in quest'ultimo periodo.

Da *Bazaar* è poi stato sviluppato *Kasbah* [31], che è un mercato virtuale basato su *Web* dove agenti autonomi comprano e vendono beni per conto degli utenti che li hanno creati. Questi ultimi specificano i parametri per guidare e vincolare il comportamento globale degli agenti. L'obiettivo di questo sistema è di aiutare a realizzare una trasformazione fondamentale nel modo in cui la gente scambia beni: dalla necessità di uno sforzo per un controllo costante, a un sistema dove un agente software fa la maggior parte del lavoro per conto dell'utente.

Da queste precedenti esperienze, il MIT ha promosso l'iniziativa *Agent-mediated Electronic Commerce* [63] (AmEC), che studia come le tecnologie degli agenti software possano velocizzare la rivoluzione del commercio elettronico. I punti fondamentali riguardano mercati basati su componenti distribuiti, linguaggi aperti ed estendibili e protocolli per localizzare e definire beni e servizi, confronti di prodotti basati sul valore, supporto alle decisioni d'acquisto, protocolli di negoziazione, visualizzazione dei dati e delle attività dei mercati. Inoltre si investigano le questioni relative alla sicurezza e le implicazioni socioeconomiche della prossima generazione di sistemi di commercio elettronico, basate sulla mediazione di agenti.

L'Università del Minnesota ha proposto un'altra architettura, MAGMA (*Minnesota AGent Marketplace Architecture* [30]), per un mercato virtuale basato su agenti, che comprende tutti gli elementi richiesti per simulare un vero mercato. Questi elementi includono un'infrastruttura per la comunicazione, meccanismi per l'immagazzinamento e il trasferimento di beni, servizi bancari come transazioni monetarie e un meccanismo economico per transazioni dirette o tramite intermediari. MAGMA è un'architettura estendibile che fornisce tutti i servizi essenziali per attività commerciali basate su agenti. Questi servizi sono disponibili attraverso un'interfaccia programmabile (API) aperta che permette l'uso di un insieme eterogeneo di agenti, indipendentemente dalla piattaforma e dal linguaggio.



## Capitolo 3

# *La Trading Agent Competition*

In questo capitolo presentiamo la *Trading Agent Competition* (TAC). Descriviamo in primo luogo le regole della competizione, dopo di che proseguiamo con un'analisi del problema generale della TAC. Infine descriviamo la piattaforma *AuctionBot*, che è stata utilizzata per lo svolgimento delle gare.

### 3.1 Introduzione

La *Trading Agent Competition 2001*<sup>1</sup> (TAC-01) è una competizione di commercio elettronico a cui hanno partecipato agenti software da tutto il mondo. Lo scopo della competizione è stimolare la ricerca sugli agenti di commercio (*trading agents*) dando rilievo allo sviluppo di strategie di successo in un ambiente controllato [28]. Questa competizione è stata organizzata dall'Università del Michigan ed è giunta alla seconda edizione. La prima edizione è stata tenuta a Boston, Massachusetts USA nel luglio del 2000, in corrispondenza della Quarta Conferenza Internazionale sui Sistemi Multiagente (*Fourth International Conference on Multiagent Systems* o ICMAS-00). Quest'anno l'evento si è tenuto in concomitanza alla Terza Conferenza ACM sul Commercio Elettronico (*Third ACM Conference on Electronic Commerce*), che si è tenuta a Tampa, Florida USA, dal 14 al 17 Ottobre 2001.

---

<sup>1</sup><http://auction2.eecs.umich.edu>

Le sessioni di gioco della TAC sfruttano il supporto di Internet e la piattaforma *Michigan Internet AuctionBot* [27]. Questa piattaforma si compone di un *server* configurabile dedicato alle aste che implementa diversi tipi di regole per la gestione di vari tipi di aste. È a disposizione degli sviluppatori un'interfaccia di programmazione (API), che dà la possibilità di creare agenti software; in questo modo tali agenti possono partecipare alle aste. Quest'interfaccia si basa sui protocolli TCP e XML. L'Università del Michigan mette a disposizione due *server* dove si sono svolti sia gli allenamenti sia la competizione.

Gli agenti partecipanti alla TAC operano nello scenario delle agenzie di viaggio. Ogni agenzia (rappresentata da un agente) ha un pacchetto di otto clienti da soddisfare. L'obiettivo da raggiungere è l'allestimento di pacchetti di viaggio da vendere ai clienti. Ognuno di questi ha le sue preferenze in termini di date di partenza e di arrivo, di quanto è disposto a pagare in più per alloggiare in un buon albergo e di quanto è disposto a spendere per assistere alle attrazioni disponibili. I punteggi della TAC dipendono dal grado di soddisfazione dei clienti, proporzionale a quanto sono state rispettate le loro preferenze, e dalle spese nette effettuate nei mercati.

## 3.2 Regole

In una sessione di gioco della TAC, ogni agente partecipante alla competizione opera nell'ambito delle agenzie di viaggio e ha come obiettivo l'allestimento di otto pacchetti di viaggio, che verranno in seguito venduti agli otto clienti, che gli sono stati affidati. Tutti i pacchetti prevedono la partenza dall'ipotetica TACtown e l'arrivo a Tampa, città in cui i clienti resteranno per un periodo massimo di cinque giorni. I pacchetti di viaggio consistono dei seguenti componenti: un volo di andata e ritorno, una prenotazione in un albergo e dei biglietti per una o più attrazioni, che permettono di assistere al *wrestling* tra alligatori oppure di visitare il parco divertimenti o il museo. Ognuno dei clienti ha le sue preferenze sui vari aspetti del viaggio: delle da-

te preferite di arrivo e partenza, quanto è disposto a spendere in più per alloggiare nell'albergo migliore di Tampa e quanto è disposto a pagare per assistere ai vari tipi di attrazioni. L'obiettivo dell'agente di viaggio è di massimizzare la soddisfazione complessiva dei suoi clienti (la somma delle utilità dei clienti), cioè di allestire pacchetti più vicini possibili alle loro preferenze, cercando però di contenere al massimo i costi.

La difficoltà del problema risiede nelle interdipendenze tra i diversi beni. Ad esempio il viaggiatore ha bisogno di una stanza d'albergo per ogni notte compresa tra il giorno dell'arrivo e quello della partenza e può assistere a un'attrazione solo durante quell'intervallo di tempo. In più, i clienti hanno preferenze individuali su quali giorni rimanere a Tampa, sul tipo di albergo e su quali sono le attrazioni che desiderano vedere. Tutti i tre tipi di beni (voli, alberghi e attrazioni) sono scambiati in mercati separati, le cui regole sono descritte nel seguito.

### 3.2.1 I mercati

Tutti i mercati seguono il seguente protocollo ad alto livello:

1. un agente sottopone un'offerta,
2. l'asta immediatamente aggiorna la sua quotazione, indicando i prezzi correnti.

Le regole di ogni particolare mercato specificano quando, o sotto quali condizioni, l'asta accetterà le offerte e registrerà le transazioni.

TACAIR è l'unica compagnia aerea che effettua voli tra TACtown e Tampa. C'è un solo volo per ogni tratta (andato o ritorno) al giorno. I biglietti sono venduti in un'asta a singolo venditore. Dato che tutti i clienti devono rimanere almeno una notte a Tampa, non ci sono voli d'andata l'ultimo giorno, né voli di ritorno il primo giorno. TACAIR è rappresentata nel mercato da un agente che impone i prezzi secondo una funzione stocastica, illustrata nella Sezione 3.3.1.

Ci sono due alberghi a Tampa: *Tampa Towers* e *Shoreline Shanties*. Il primo è più confortevole ed è il migliore dei due; per questa ragione ci si aspetta che costi di più. Si noti che un cliente non può cambiare albergo durante il suo viaggio. Le stanze sono scambiate in aste a prezzo crescente: un'asta per ogni albergo e per ogni notte, ognuna con sedici stanze da vendere. Solo gli alberghi possono vendere stanze. Non c'è un'offerta minima per le stanze. La differenza di prezzo tra gli alberghi è basata sulla preferenza per *Tampa Towers*, come stabilito dalla funzione d'utilità del cliente. Dato che i clienti hanno bisogno di una stanza dalla notte del loro arrivo alla notte prima della partenza, non ci saranno alberghi disponibili l'ultimo giorno.

All'inizio dell'istanza di gioco, ogni agente di viaggio riceve un lotto di biglietti per le attrazioni. Ci sono un totale di otto biglietti disponibili per ogni evento ogni giorno. Inoltre all'inizio della sessione di gioco ogni agente riceve dodici biglietti raggruppati come segue: (i) un gruppo di quattro biglietti per lo stesso evento validi per il primo o per il quarto giorno, (ii) un gruppo di quattro biglietti per lo stesso evento validi per il secondo o per il terzo giorno, (iii) un gruppo di due biglietti per lo stesso evento (diverso da quello al punto (i)) validi per il primo o per il quarto giorno e (iv) un gruppo di due biglietti per lo stesso evento (diverso da quello al punto (ii)) validi per il secondo o per il terzo giorno. Gli agenti possono sia comprare sia vendere biglietti. Come per gli alberghi, un cliente non può usare un biglietto, il giorno della sua partenza.

### 3.2.2 La funzione di utilità

Le preferenze di ogni cliente riguardano: il giorno preferito per l'arrivo ( $PA$ ), il giorno preferito per la partenza ( $PD$ ), la somma che il cliente è disposto a pagare in più, per alloggiare nel miglior albergo ( $HP$ ), le tre somme che il cliente è disposto a pagare per assistere ai tre tipi di attrazione ( $AW$ ,  $AP$ ,  $MU$ ). Le preferenze per l'arrivo e la partenza sono scelte in modo casuale per ogni cliente, affinché tutte le possibili coppie di arrivi e partenze siano egualmente probabili. La somma  $HP$ , che il cliente è disposto a pagare in più,

per alloggiare nell'albergo migliore, è scelta uniformemente nell'intervallo da \$50 a \$150. Si noti che il cliente paga questa somma una volta soltanto e non per ogni notte in cui rimane alle *Tampa Towers*. Le somme  $AW$ ,  $AP$ ,  $MU$ , che il cliente è disposto a spendere per assistere rispettivamente al *wrestling* tra alligatori, per visitare il parco divertimenti o il museo, sono scelte uniformemente nell'intervallo da \$0 a \$200.

Un pacchetto di viaggio è composto da: la vera data dell'arrivo ( $AA$ ), la vera data della partenza ( $AD$ ) e un indicatore ( $TTI$ ) che vale 1 se comprende una prenotazione per l'albergo migliore (*Tampa Towers*) e vale 0 se la prenotazione vale per le *Shoreline Shanties*. Un pacchetto di viaggio è *valido* se per ogni notte compresa tra il giorno dell'arrivo e quello della partenza, è stata prenotata una stanza. Si noti che un cliente deve rimanere almeno una notte e che per il giorno della partenza il cliente non necessita di una stanza d'albergo.

Un pacchetto di attrazioni è composto da una serie di biglietti per i vari eventi, e consiste di non più di un biglietto per tipo. Per il calcolo della funzione di utilità, si usa un indicatore per ogni evento ( $AWI$ ,  $API$ ,  $MUI$ ). Ogni indicatore può valere 0 se il cliente non assisterà alla relativa attrazione, oppure 1 nel caso contrario. Un pacchetto di attrazioni è *valido* se tutti i biglietti sono per eventi tenuti in giorni distinti. Inoltre per assistere a un evento il cliente deve essere in città e quindi non sono validi pacchetti contenenti biglietti relativi a eventi che si svolgono in giorni in cui il cliente non è a Tampa. Si noti che un cliente paga la stessa cifra, sia che assista allo stesso evento una sola volta, sia che vi assista più volte.

Il livello di soddisfazione  $u$ , misurato in dollari, di un cliente relativamente a un pacchetto di viaggio *valido* e a un pacchetto di attrazioni *valido* è dato dalla formula:

$$u = 1000 - TP + HB + FB$$

dove  $TP$  rappresenta la penalità per lo spostamento delle date d'arrivo o di partenza,  $HB$  rappresenta il bonus ottenuto se si prenota l'albergo migliore e  $FB$  rappresenta il bonus ottenuto se nel pacchetto ci sono biglietti per le

attrazioni. In particolare la penalità  $TP$  per lo spostamento delle date è dato dalla formula:

$$TP = 100(|AA - PA| + |AD - PD|)$$

Il bonus  $HB$ , ottenuto se il cliente alloggia nell'albergo migliore ( *Tampa Towers*), è dato da:

$$HB = TTI * HP$$

Il bonus  $FB$ , ottenuto se il cliente assiste a una o più attrazioni, è dato da:

$$FB = AWI * AW + API * AP + MUI * MU$$

Si noti che l'utilità  $u$  di un pacchetto *valido* può variare da \$400 a \$1750, mentre un pacchetto *non valido* ha un'utilità pari a zero. Il punteggio finale è calcolato come somma delle utilità  $u$  per ogni cliente da cui si sottraggono le spese effettuate dall'agente e un'eventuale penalità di \$200 per ogni biglietto di un'attrazione venduto senza esserne in possesso.

### 3.3 Analisi della competizione

Il problema che gli agenti partecipanti alla *Trading Agent Competition* devono risolvere può essere scomposto in tre sottoproblemi più semplici: l'*acquisizione* da parte dell'agente dei beni necessari, l'*allocazione* ai clienti dei beni ottenuti e infine il *completamento* dei pacchetti, ovvero la scelta di quali risorse comprare, allo scopo di ottenere pacchetti *validi* partendo dai beni attualmente in possesso [1].

#### 3.3.1 Il problema dell'acquisizione

Questo problema viene affrontato quando è necessario entrare in possesso dei beni desiderati. I tre tipi di beni disponibili si possono acquistare su tre diversi tipi di mercati. Si hanno regole diverse per le aste dei biglietti aerei, le aste delle stanze d'albergo e le aste dei biglietti per le attrazioni.

### Le aste dei biglietti aerei

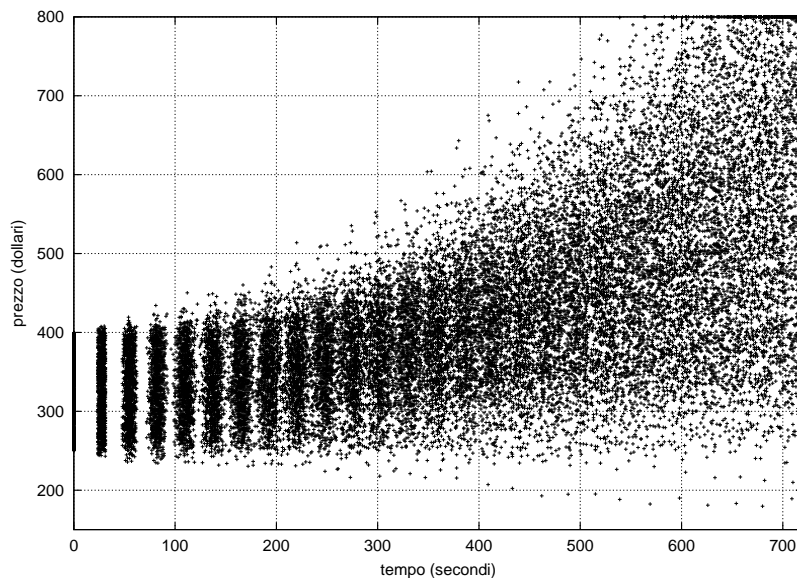
I biglietti sono venduti in aste a singolo venditore (*Single Seller Auctions*), distinte per giorno e tragitto (andata e ritorno). La disponibilità di biglietti aerei è illimitata, inoltre il prezzo dei biglietti aerei è determinato dal venditore e non dipende dalle offerte fatte dai compratori. Ogni offerta che supera il prezzo corrente, produrrà immediatamente una transazione al prezzo stabilito dal venditore. Ogni offerta inferiore al prezzo corrente, rimane attiva fino a che il prezzo corrente non la raggiunge oppure fino a che non viene ritirata. Un'offerta attiva produrrà una transazione pari al suo valore (e non al prezzo corrente).

L'aggiornamento dei prezzi dei voli viene effettuato mediante una passeggiata aleatoria: il prezzo iniziale è scelto uniformemente nell'intervallo tra \$250 e \$400 ed è perturbato ogni  $n$  secondi (con  $n$  che varia da ventiquattro a trentadue) da un valore estratto da una distribuzione uniforme fra  $-10$  e  $x(t)$ . L'estremo superiore finale  $x$  delle perturbazioni è una variabile casuale (non rivelata direttamente agli agenti), scelta per ogni volo indipendentemente da una distribuzione uniforme su  $[10, 90]$ . L'estremo superiore della perturbazione all'istante  $t$ ,  $x(t)$ , è una interpolazione lineare tra  $10$  e l'estremo finale  $x$ :

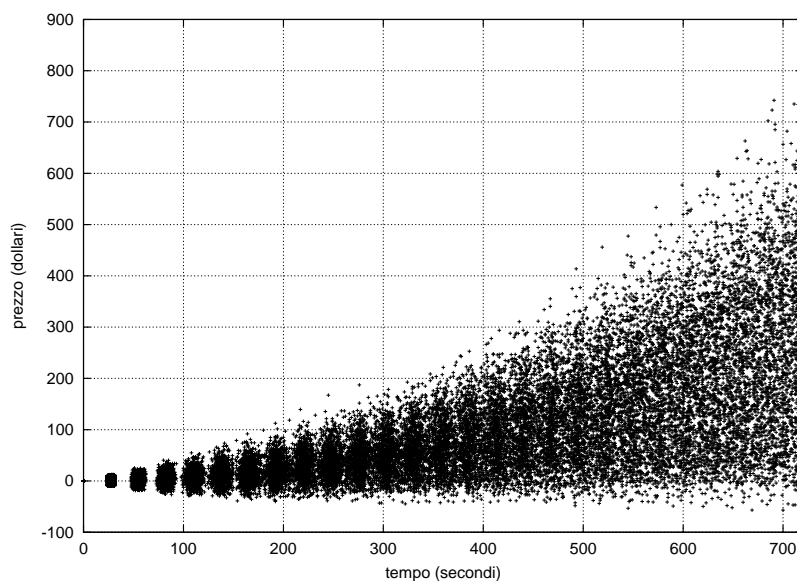
$$x(t) = 10 + \frac{t}{T}(x - 10)$$

dove  $T$  è la durata del gioco, ovvero dodici minuti. Indipendentemente dalle perturbazioni, i prezzi devono rimanere nell'intervallo tra \$150 e \$800.

Osservando lo *scatter plot* in Figura 3.1(a) ottenuto tracciando lo sviluppo del prezzo in mille simulazioni del processo descritto sopra, si può notare che in media il prezzo del biglietto aereo aumenterà con il passare del tempo e più l'agente aspetterà a comprare, più spenderà per ottenere il biglietto desiderato. Questo comportamento si può osservare ancora meglio sullo *scatter plot* in Figura 3.1(b) che riproduce ancora l'andamento del prezzo dei biglietti in mille simulazioni, ma relativamente al prezzo iniziale. Se analizziamo gli istogrammi in Figura 3.2, ovvero le distribuzioni empiriche dei prezzi a vari istanti di tempo, si nota che aumentano sia la media che la varianza al



(a) Valori assoluti.



(b) Valori relativi al prezzo iniziale.

Figura 3.1: *Scatter plot* di mille simulazioni della camminata aleatoria: usando valori assoluti (a) e usando valori relativi al prezzo iniziale (b).

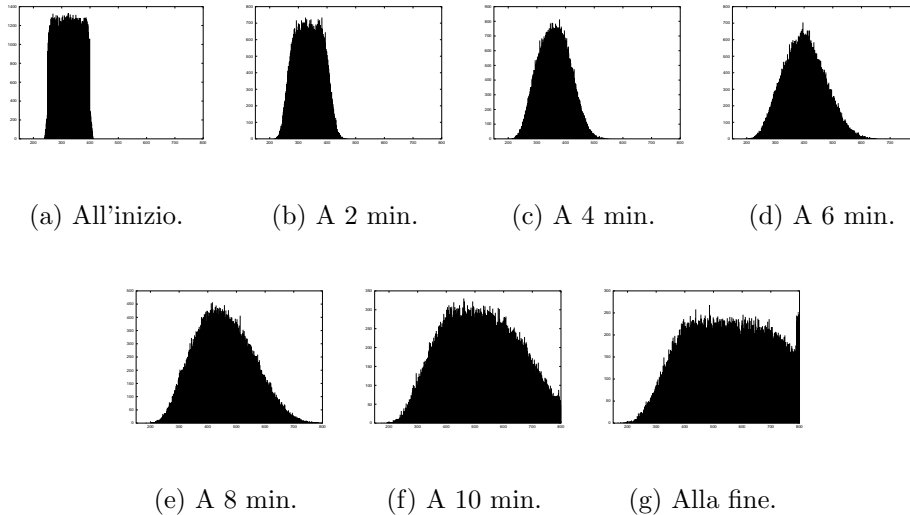


Figura 3.2: Istogrammi delle distribuzioni dei prezzi dei biglietti aerei in vari istanti della gara: all'inizio (a), dopo due minuti (b), dopo quattro minuti (c), dopo sei minuti (d), dopo otto minuti (e), dopo dieci minuti (f) e alla fine (g).

passare del tempo. Gli istogrammi, infatti tendono a dilatarsi e a spostarsi verso prezzi maggiori.

### Le aste delle stanze d'albergo

Le stanze d'albergo sono vendute in un'asta di tipo inglese al sedicesimo prezzo (*16th-Price English Auction*), cioè un'asta in cui le offerte devono essere crescenti e i vincitori delle sedici stanze pagano il sedicesimo prezzo più alto. Il tipo di asta utilizzato è quindi una combinazione di due tipi di aste: quella di tipo inglese (*English Auction*, si veda la Sezione 2.3.1) e quella che in letteratura è conosciuta come asta di Vickrey (*Vickrey Auction*, si veda la Sezione 2.3.4).

Le nuove offerte presentate dagli agenti, devono superare la quotazione corrente. Quest'ultima (indicata in seguito con *ASK*) rappresenta il sedicesimo prezzo più alto. Per essere accettata, ogni nuova offerta  $b$  deve soddisfare

le seguenti condizioni:

1.  $b$  deve permettere di comprare almeno una unità a un prezzo pari almeno a  $ASK + 1$ ;
2. se l'offerta corrente di un agente  $b'$  permettesse di acquistare  $q$  unità al prezzo corrente, allora la nuova offerta  $b$  deve permettere di acquistare almeno  $q$  unità a un prezzo almeno pari a  $ASK + 1$ .

Gli agenti non possono ritirare le loro offerte. Quando l'asta chiude, le sedici stanze verranno assegnate agli agenti che hanno proposto le sedici offerte più alte, a un prezzo pari alla quotazione corrente ( $ASK$ ).

Nell'edizione del Luglio 2000 si è potuto verificare che nonostante le aste degli alberghi siano in teoria una combinazione tra asta inglese e asta di Vickrey, nella pratica si sono rivelate equivalenti ad aste di Vickrey pure [2]. Durante la gara, gli agenti alzavano del minimo possibile le loro offerte per rivelare il meno possibile agli avversari e per impedire che l'asta chiudesse<sup>2</sup>. Infatti gli agenti che hanno ottenuto i punteggi migliori sono quelli che tendevano a offrire la loro valutazione privata (utilità marginale) del bene, all'avvicinarsi della fine della sessione. Questa è infatti la strategia ottima per un'asta di Vickrey. Un'altra caratteristica di questo tipo di aste che si è verificata nella scorsa edizione, è l'assenza quasi totale del fenomeno della "maledizione del vincitore", che si verifica quando un bene viene acquistato a un prezzo superiore al suo valore reale. Infatti il prezzo pagato dai vincitori, che avevano utilizzato la strategia ottima, era generalmente molto inferiore alla valutazione privata, che tipicamente era dell'ordine del migliaio di dollari. E questo fatto si è verificato anche quando, nella fase finale, si sono scontrati molti agenti che adottavano questa strategia.

Per evitare questo tipo di comportamento, per l'edizione di quest'anno, è stato deciso che le aste chiudano ad istanti casuali. In particolare un'asta, scelta a caso tra quelle ancora aperte, viene chiusa ad ogni minuto a partire

---

<sup>2</sup>Nella scorsa edizione c'era una regola che faceva terminare un'asta dopo un certo periodo, non rivelato agli agenti, di inattività

dal quarto. Questo impedisce agli agenti di utilizzare la strategia ottima per un'asta di Vickrey, come nella scorsa edizione, e inoltre rende più problematico cambiare tipo d'albergo durante la gara. Le aste di quest'anno sono quindi effettivamente una combinazione di asta inglese e asta di Vickrey, infatti il prezzo tende generalmente a salire di molto durante la gara, come in un'asta inglese. Gli agenti però, non hanno a disposizione tutte le informazioni che avrebbero in un'asta inglese pura, perché in ogni istante è noto solo il prezzo a cui verrebbero vendute le stanze, se l'asta chiudesse in quel momento, cioè il sedicesimo prezzo più alto. Tuttavia il problema della maledizione del vincitore è poco frequente, come in un'asta di Vickrey, e questo incoraggia l'aggressività nelle offerte. Tutto ciò comunque rende estremamente difficoltoso la stima dei prezzi delle camere d'albergo.

### **Le aste dei biglietti per le attrazioni**

I biglietti delle attrazioni possono essere scambiati in una CDA (*Continuous Double-Auction*, si veda la Sezione 2.3.5), quindi in un mercato simile a quello azionario, dove esistono una domanda e un'offerta.

Più precisamente nella TAC-01 è stata adottata un formato "a doppia chiamata continua con coda delle richieste" (*continuous double auction with order queue*), in cui un partecipante può annunciare domanda o offerta in ogni momento. Una volta posta, la domanda o l'offerta rimangono fino a che il partecipante non decide di modificarle o di rimuoverle oppure fino a che vengono accettate. In particolare nella TAC-01 questo formato è stato leggermente modificato: durante le gare agli agenti vengono comunicate solo l'offerta e la domanda in essere (*outstanding bid e outstanding ask*), cioè la più alta delle offerte e la più bassa delle domande. Inoltre se un'offerta supera la domanda in essere, viene automaticamente effettuata la transazione a un prezzo pari alla domanda in essere.

### 3.3.2 Il problema dell'allocazione

Questo problema è equivalente al problema della determinazione del vincitore in un'asta combinatoria a unità multiple (*Winner Determination Problem in Multi-Unit Combinatorial Auctions*, si veda la Sezione 2.3.6) che a sua volta è equivalente al problema del *Multidimensional Maximum Weighted Set-Packing* (e *Multidimensional Maximum Weighted Clique*) ed è quindi un problema NP-completo [1].

Nell'edizione dello scorso anno sono state utilizzate diverse soluzioni al problema dell'allocazione, sia approssimate sia ottime. In particolare gli agenti che hanno utilizzato un allocatore ottimo si sono piazzati ai primi due posti della graduatoria finale. Per esempio, ATTac-2000 [14] ha utilizzato l'impostazione classica dei problemi di ottimizzazione e ha utilizzato il pacchetto di programmazione lineare "LPSolve" per trovare la soluzione ottima. Sono stati definiti un insieme di variabili, vincoli su queste variabili e una funzione obiettivo. La funzione obiettivo esprime il fatto che si cerca l'allocazione con il massimo valore (ricavi meno costi). RoxyBot [1] ha invece utilizzato un approccio euristico basato sui lavori di Sandholm [11], in cui lo spazio delle differenti allocazioni viene esplorato con l'algoritmo A\* (per la soluzione ottima) e con il *beam search* (per una soluzione approssimata ma ottenibile più rapidamente).

### 3.3.3 Il problema del completamento

Questo problema consiste nel costruire l'insieme dei beni da acquistare, partendo dall'insieme dei beni posseduti attualmente, in modo da ottenere il maggior ricavo, con la minor spesa. Il problema del completamento può essere visto come una generalizzazione del problema dell'allocazione. La soluzione viene cercata tra tutti i pacchetti ammissibili che si possono costruire, invece che su tutti quelli ottenibili con i beni già acquisiti. Il valore di ogni pacchetto, è dato dalla differenza tra la funzione di utilità e i costi stimati dei beni inclusi, invece che il solo valore della funzione di utilità.

Nella scorsa edizione, anche in questo caso, sono state utilizzate soluzioni sia ottime sia approssimate. La maggior parte degli agenti ha optato per una soluzione approssimata, privilegiando la velocità di esecuzione data dalla minor complessità computazionale. Invece gli agenti che si sono classificati ai primi posti, hanno optato per una soluzione ottima. Ad esempio, ATTac-2000 ha utilizzato la stessa formulazione, vista in precedenza, del programma lineare, la cui soluzione oltre all'allocazione ottima delle risorse già possedute, fornisce anche il numero di beni da acquisire. Allo stesso modo RoxyBot ha utilizzato il medesimo metodo di ricerca, già descritto in precedenza, basandosi su delle strutture dati, chiamate vettori di prezzo (*pricelines*), per gestire il fatto di non avere ancora acquisito tutti i beni necessari.

### 3.4 La piattaforma *AuctionBot*

*Michigan Internet AuctionBot* [27] è un *server* per aste flessibile, scalabile e robusto, che supporta sia agenti software sia agenti umani. La piattaforma gestisce molte aste simultaneamente, separando l'interfaccia dalle procedure fondamentali per la gestione delle aste. Questa separazione permette di avere tempi di risposta rapidi e di tollerare interruzioni nel sistema o nella rete, ma necessita accurate procedure di rilevamento dei tempi per assicurare la precisione temporale. La possibilità di specificare le aste in termini di parametri indipendenti, rende il sistema utile per la ricerca sugli agenti software nell'ambito del commercio elettronico, permettendo l'esplorazione dei meccanismi dei diversi tipi di asta.

La formalizzazione della contrattazione tra agenti in termini di aste fornisce una struttura omogenea e permette una migliore comprensione dei modelli teorici studiati nell'ambito della teoria delle aste. Malgrado la diversità dei meccanismi delle varie aste, è possibile definire chiaramente le attività e i messaggi utilizzati nelle interazioni. Questo è importante per chi progetta sistemi multiagente, perché rende la comunicazione strutturata.

La piattaforma *AuctionBot* gestisce un grande numero di aste simultanea-

mente. Per poter partecipare a una di queste aste, un utente deve registrarsi. Gli utenti umani possono controllare il loro accesso attraverso una pagina *web* che presenta una visione organizzata delle loro offerte, delle aste a cui partecipano e delle transazioni già effettuate. Esiste anche un'interfaccia per gli agenti software. Questa interfaccia è un protocollo di messaggi a livello TCP/IP, che permette agli agenti l'accesso a tutte le caratteristiche della piattaforma presenti nell'interfaccia *web*. Gli agenti possono fare offerte, creare un *account* o richiedere informazioni sulle aste. Il codice per interfacciarsi con gli agenti è di dominio pubblico.

### 3.4.1 Messaggi e attività

Ogni asta può svolgere tre tipi di attività fondamentali. Un'asta può **ricevere offerte** (*Receive Bids*); quando ciò avviene, il banditore dell'asta deve verificare che l'offerta soddisfi le regole. Un'asta può **fornire informazioni intermedie** (*Supply Intermediate Information*) che genericamente prendono il nome di quotazioni (*price quote*). Infine un'asta può **effettuare transazioni** (*Clear*): infatti lo scopo principale di un'asta è di determinare lo scambio delle risorse e i pagamenti tra compratori e venditori.

Le aste *devono* avere una fase in cui ricevono offerte e una fase in cui effettuano transazioni, mentre la fase in cui vengono fornite informazioni intermedie è opzionale.

In questa formulazione, non è necessario che l'asta sia coinvolta direttamente nell'esecuzione delle transazioni, sebbene molti di coloro che offrono servizi d'aste su Internet, partecipano al trasferimento di denaro e beni. Ad esempio, la piattaforma MAGMA fornisce questi servizi (si veda la Sezione 2.4.2). Per supportare le attività sopra descritte, sono richiesti i seguenti messaggi:

- **Offerta** (*Bid*): questo messaggio è mandato dall'agente all'asta. Un'offerta specifica la volontà di un agente di vendere o comprare una certa quantità di un bene in funzione del suo prezzo. È possibile includere

altre informazioni, ad esempio le condizioni per cui l'offerta scade e se l'offerta può essere suddivisa.

- **Ritiro dell'offerta** (*Bid Withdrawal*): questo messaggio è mandato dall'agente all'asta. Non tutte le aste permettono a un agente di ritirare la propria offerta.
- **Ammissione dell'offerta** (*Bid Admittance*): questo messaggio è mandato dall'asta all'agente, quando un'offerta soddisfa le regole dell'asta.
- **Rifiuto dell'offerta** (*Bid Rejection*): questo messaggio è mandato dall'asta all'agente, quando un'offerta viola le regole dell'asta.
- **Quotazione** (*Price Quote*): messaggio mandato dall'asta all'agente. Il contenuto preciso varia a seconda dell'asta, ma la definizione che viene utilizzata è il prezzo che l'agente avrebbe dovuto offrire per vincere uno dei beni messi in palio, all'istante in cui la quotazione è stata fornita.
- **Notificazione di transazione** (*Transaction Notification*): messaggio mandato dall'asta agli agenti coinvolti in una transazione. Specifica i termini di uno scambio, inclusi il prezzo e la quantità e le informazioni necessarie per effettuare lo scambio, come le identità del compratore e del venditore.

Un agente può avere solo un'offerta attiva a un determinato istante — una nuova offerta ricevuta dall'asta sostituisce ogni vecchia offerta. All'interno di questo protocollo, le percezioni dell'agente sono limitate all'osservazione della quotazione. In molti casi l'agente può dedurre dalla quotazione di quali e quanti beni è in possesso.

### 3.4.2 Interfaccia di programmazione

L'interfaccia di programmazione per applicativi (API) della piattaforma *AuctionBot* è basata sull'invio di messaggi in formato XML tra *client* e *server*.

L'interfaccia è quindi molto leggera, in quanto si occupa solo della composizione dei messaggi da inviare al *server*. Questa API permette di sviluppare applicazioni *client* lasciando piena libertà di scelta del linguaggio di programmazione e del sistema operativo. Implementando le specifiche della API, gli sviluppatori possono interagire con il *server* che gestisce le aste, come farebbero attraverso l'interfaccia *web*, ma tramite agenti software. Gli obiettivi di progetto per la API includono l'indipendenza dalla piattaforma, la semplicità d'uso, la correttezza, e l'uso di linguaggi di interfacciamento standard.

L'indipendenza dalla piattaforma è stata ottenuta ponendo l'attenzione sul formato dei messaggi che viaggiano dal *client* al *server*, piuttosto che su "librerie" che il *client* usa per interagire con la macchina remota. L'interfaccia è quindi un insieme di convenzioni di comunicazione, piuttosto che una "libreria" di funzioni.

La caratteristica della semplicità di progetto serve sia per l'obiettivo di avere un'interfaccia amichevole per l'utente, sia per l'obiettivo della correttezza. I servizi forniti dalla piattaforma — possibilità di creare aste, fare offerte, ricevere le quotazioni e avvertire i clienti delle transazioni — sono fondamentalmente semplici e quindi un'interfaccia troppo complessa servirebbe solo a confondere l'utente. L'uso di sequenze di caratteri ASCII per trasmettere dati tra *client* e *server*, è stata ispirata dal sistema operativo *Plan 9*<sup>3</sup> dove le periferiche sono controllate da messaggi testuali; questi non hanno problemi di ordinamento dei *byte* e inoltre hanno una semantica chiara per la lettura e la scrittura. Il compito in più di conversione delle stringhe, non presente in un protocollo binario, è compensato dalla semplicità di comporre e di verificare messaggi leggibili dall'uomo [34].

Tutti i messaggi sono codificati nel formato standard XML, così qualsiasi analizzatore sintattico XML può essere utilizzato per interfacciarsi con la API della piattaforma *AuctionBot*. Tradizionalmente, una API è vista come un'insieme di strutture dati e funzioni, la cui implementazione risiede su una macchina locale. I programmatori accedono ai servizi supportati attra-

---

<sup>3</sup><http://plan9.bell-labs.com/plan9dist/>

verso l'operazione di collegamento (*linking*) alla "libreria" che implementa la API (durante la compilazione oppure durante l'esecuzione) e chiamano le sue funzioni tramite un'interfaccia ben definita. Viceversa nel modello utilizzato dall'Università del Michigan, le funzioni della API risiedono su una macchina remota (*server*) e, quindi, non sono direttamente collegate al codice cliente. Le interfacce alle funzioni sono messaggi ben definiti codificati come stringhe che sono inviate attraverso una porta specifica e invocano la funzione richiesta sulla macchina remota. Quest'ultima restituisce un messaggio testuale, contenente il risultato della funzione, al cliente che l'ha invocata.



# Capitolo 4

## L'agente *polimi\_bot* per la TAC-01

In questo capitolo descriviamo l'agente *polimi\_bot* che è stato sviluppato all'interno del Laboratorio di Intelligenza Artificiale e Robotica per partecipare all'edizione 2001 della *Trading Agent Competition*. Dopo aver illustrato la struttura dell'agente e averne descritto i componenti principali, procediamo spiegando come l'agente affronta e cerca di risolvere i vari problemi posti dalla competizione. Infine descriviamo come abbiamo proceduto all'implementazione delle strategie.

L'agente *polimi\_bot* utilizza una strategia euristica. Abbiamo cercato, infatti, di rendere la sua implementazione meno complessa possibile, date le particolarità della competizione che rendono estremamente difficile il compito di cercare e correggere gli errori nel codice.

### 4.1 L'architettura dell'agente

Il comportamento di *polimi\_bot*, come quello della maggior parte degli altri agenti (anche nelle passate edizioni), è ciclico. Durante ogni iterazione, l'agente richiede al *server* lo stato corrente di tutte le aste, stima i costi delle risorse, cerca di modificare l'allocazione attuale per ottenere un punteggio mi-

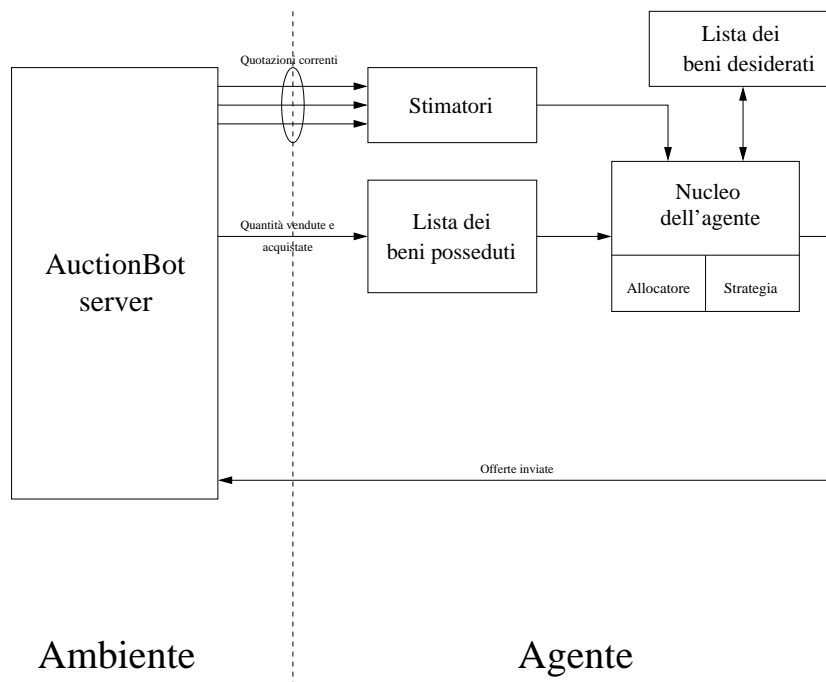


Figura 4.1: Architettura dell'agente.

gliore, basandosi sui costi stimati. Infine l'agente invia le offerte appropriate per i beni desiderati al *server*.

L'ambiente in cui l'agente *polimi\_bot* opera è costituito dalla piattaforma *AuctionBot*. Questa rende disponibile all'agente le informazioni correnti sulle aste e le quantità di beni che l'agente ha comprato o venduto fino a questo momento. Queste informazioni costituiscono le percezioni dell'agente e gli permettono di aggiornare gli stimatori e la lista dei beni posseduti. Di conseguenza il nucleo dell'agente, nel quale è codificata la strategia e che contiene l'allocatore, può aggiornare la lista dei beni che desidera comprare. E in base a questa può inviare le opportune offerte al *server*. L'architettura dell'agente è illustrata in Figura 4.1.

## 4.2 Modelli e predittori

Abbiamo iniziato lo sviluppo dell'agente ricercando modelli sufficientemente accurati per la descrizione delle aste. Infatti partendo da buoni modelli, si possono ottenere buoni predittori dei prezzi dei vari beni e la disponibilità di buoni predittori permette all'agente di ottenere buoni risultati.

Creare un modello per i tre diversi tipi di mercato presenta vari gradi di difficoltà a seconda dell'asta. Il caso più semplice è rappresentato dall'asta dei biglietti aerei, dove il modello è noto e fornito nelle regole della gara. L'asta dei biglietti per le attrazioni presenta una difficoltà di modellazione intermedia, ma è meno critica dell'asta delle stanze d'albergo, ossia un buon predittore del prezzo dei biglietti può essere utile, ma non influenza più di tanto le prestazioni dell'agente. Il caso più complesso riguarda l'asta delle stanze d'albergo, in cui le informazioni a disposizione sono estremamente limitate e inoltre c'è la complicazione ulteriore di non conoscere a priori l'istante in cui le diverse aste (una per ogni giorno e per ogni tipo di albergo) chiuderanno.

Si consideri che spesso le informazioni sono assolutamente insufficienti per uno studio accurato dei modelli. Ad esempio nel caso del mercato dei biglietti aerei, abbiamo una sola osservazione da cui stimare un parametro di una distribuzione, oppure nel mercato delle stanze d'albergo, conosciamo solo la quotazione attuale delle stanze e questo non permette di fare inferenze, ad esempio, sulla strategia degli altri agenti. Quindi nessuno degli stimatori utilizzati ha delle proprietà tipiche dei buoni stimatori, come ad esempio la non distorsione o la consistenza asintotica e perciò dal punto di vista teorico sono dei pessimi stimatori.

### 4.2.1 Asta dei biglietti aerei

Il prezzo dei biglietti degli aerei non dipende dalle offerte fatte dai partecipanti, come nelle aste classiche, ma è determinato dal venditore istante per istante. In pratica il prezzo parte da un certo valore iniziale e viene per-

turbato periodicamente. Il valore di questa perturbazione è estratto da una distribuzione uniforme su  $[-10, x(t)]$ . L'estremo superiore  $x(t)$  è incognito e dipende in modo deterministico dal valore  $x$ , che rappresenta l'estremo superiore finale della perturbazione. Si noti che questo valore non è noto agli agenti.

La chiave di una buona predizione del prezzo dei voli è la possibilità di avere una stima accurata del valore incognito  $x$ , che dà un'indicazione sulla massima oscillazione possibile del prezzo. Per ottenere una stima di questo valore abbiamo cercato di ottenere una stima di  $x(t)$ , utilizzando un limite superiore  $\hat{x}_e$  e un limite inferiore  $\hat{x}_d$ . Si noti che  $x(t)$  rappresenta l'estremo superiore della perturbazione all'istante  $t$  ed è legato deterministicamente a  $x$  dall'equazione seguente:

$$x(t) = 10 + \frac{t}{T}(x - 10) \quad (4.1)$$

dove  $T$  è la durata del gioco, ovvero dodici minuti.

L'algoritmo per ottenere la stima funziona nel modo seguente. Dopo aver ottenuto le informazioni aggiornate sulle aste, l'agente:

1. effettua la differenza tra il prezzo  $p$  all'istante corrente  $t$  e il prezzo all'istante precedente  $t - 1$ :

$$\Delta = p(t) - p(t - 1);$$

2. considera  $\Delta$  come se fosse l'estremo  $x(t)$  e controlla se  $\Delta$  è superiore alla stima per difetto  $\hat{x}_d$ . Se lo è,  $\Delta$  diventa la nuova stima per difetto:

$$\hat{x}_d = \Delta$$

altrimenti la stima rimane invariata;

3. considera  $\Delta$  come se fosse il valor medio della distribuzione uniforme da cui viene estratta la perturbazione, che quindi avrebbe come estremi  $-10$  e  $2\Delta + 10$  e controlla se questo estremo superiore è maggiore della stima per eccesso  $\hat{x}_e$ . Se lo è, la nuova stima per eccesso diventa:

$$\hat{x}_e = 2\Delta + 10$$

altrimenti rimane invariata;

4. calcola la stima di  $x(t)$  come somma pesata delle due stime e utilizza un coefficiente  $\beta$  (pari a 0.85) per smorzare oscillazioni troppo ampie:

$$\hat{x}(t) = \beta(\hat{x}_e w_e + \hat{x}_d w_d) + (1 - \beta)\hat{x}_p(t)$$

dove  $w_e$  è il peso della stima per eccesso ed è pari a 0.4, mentre  $w_d$  è il peso della stima per difetto ed è pari a 0.6 e  $\hat{x}_p(t)$  è la stima all'iterazione precedente;

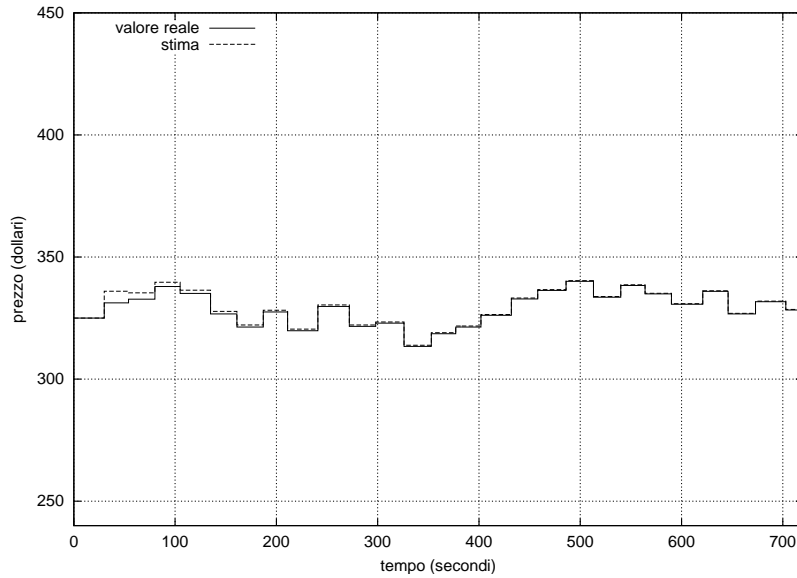
5. calcola la stima di  $x$  invertendo l'equazione 4.1:

$$\hat{x} = 10 + \frac{T}{t}[\hat{x}(t) - 10]$$

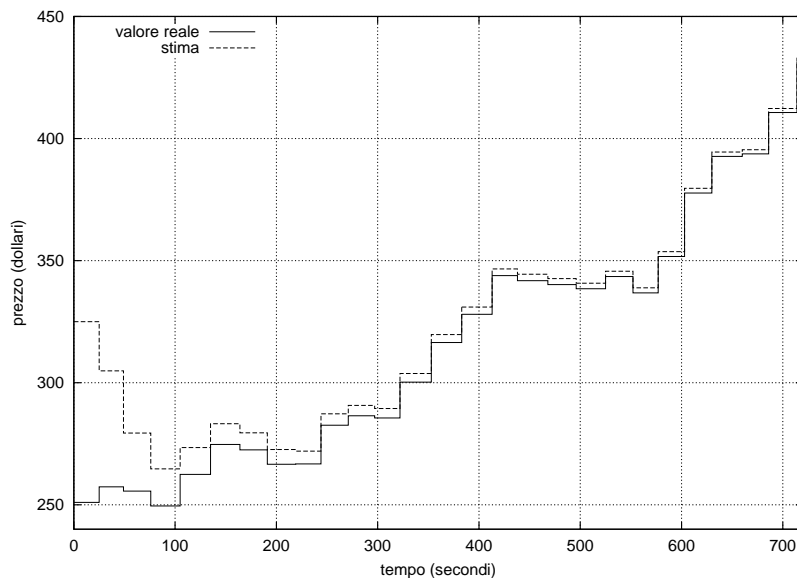
dove  $T$  è la durata dell'asta ed è pari a dodici minuti.

I valori numerici sopra citati sono stati trovati effettuando numerose simulazioni e sono stati scelti perché hanno fornito il minor errore quadratico medio (MSE) tra valore reale e valore stimato. Nei calcoli delle stime, viene anche controllato di non superare il valore massimo che  $x$  può assumere, ossia 90, e di conseguenza, utilizzando l'equazione 4.1, viene verificato che anche le stime di  $x(t)$  non eccedano il valore di saturazione. In particolare questo valore viene più facilmente raggiunto dallo stimatore per eccesso ed è per questo che  $w_e$  è minore di  $w_d$ . In pratica lo stimatore per difetto è lo stimatore di massima verosimiglianza (MLE) per la distribuzione uniforme da cui viene estratta la perturbazione del prezzo, mentre quello per eccesso è lo stimatore ottenuto con il Metodo dei Momenti (MME).

Le prestazioni dello stimatore possono essere verificate in Figura 4.2, dove vengono riportati i due casi limite ottenuti in centomila simulazioni. In particolare la Figura 4.2(a) riporta una simulazione in cui la stima ha inseguito molto bene il valore reale, mentre la Figura 4.2(b) mostra un caso in cui la stima ha cominciato ad essere abbastanza accurata solo dopo alcuni minuti.



(a) Errore minimo (MSE = 1.5177).



(b) Errore massimo (MSE = 345.118).

Figura 4.2: La migliore prestazione dello stimatore del prezzo dei biglietti aerei su 100000 simulazioni (a) e la peggiore prestazione dello stimatore (b).

### 4.2.2 Asta delle stanze d'albergo

Il modello del mercato delle stanze d'albergo è più complicato da descrivere, in quanto si tratta di un modello non lineare in cui il prezzo dipende dalle azioni di tutti gli agenti coinvolti e dove l'unica informazione disponibile durante la sessione di gioco è il sedicesimo prezzo più alto. L'osservabilità del sistema è quindi molto limitata. Inoltre non conoscendo a priori l'istante in cui le aste chiudono, non è neanche possibile applicare i modelli utilizzati lo scorso anno o comunque utilizzare la strategia ottima per un'asta di Vickrey (Paragrafo 2.3.4).

Data la complessità del problema, il modello ottimo non è proponibile per questo tipo di mercato. Abbiamo quindi cercato di ottenere un compromesso tra la complessità del modello e la sua accuratezza. Un modello troppo complesso sarebbe infatti difficile da gestire e inoltre a causa dei vincoli temporali della competizione, un modello troppo complesso dal punto di vista computazionale potrebbe causare un peggioramento delle prestazioni dell'agente. Per questo motivo abbiamo tentato di trovare un'approssimazione lineare accettabile.

Inizialmente abbiamo cercato di creare un modello del processo che descrive l'andamento della quotazione, con un modello autoregressivo di ordine  $n$ , AR( $n$ ). Per selezionare l'ordine sono stati utilizzati i criteri FPE (*Final Prediction Error*) e AIC (*Akaike's Information theoretic Criterion*), che permettono l'identificazione del modello in modo automatico, utilizzando un approccio per tentativi. FPE e AIC rappresentano la misura di quanto uno specifico modello sia adatto a descrivere un processo e entrambi sono funzioni della varianza dei residui del modello (errore di predizione): al diminuire della varianza, aumenta l'accuratezza. Grazie a questi criteri, abbiamo trovato che il risultato migliore (la minor varianza dell'errore di predizione) si ottiene con un modello del primo ordine, quindi un AR(1):

$$y(t) = \alpha y(t-1) + \eta(t).$$

Tuttavia, quando abbiamo proceduto all'identificazione del parametro  $\alpha$ ,

abbiamo trovato un valore di pochissimo superiore a uno. Da questo si ricava che il prezzo attuale è dato dal prezzo precedente più una quantità non nota (il fatto che il parametro sia poco superiore a uno è dovuto al fatto che il prezzo non può scendere). Sfortunatamente queste conclusioni non forniscono alcuna informazione in più di quello che già conoscevamo. Anche l'analisi del residuo  $\eta(t)$  non ha portato a risultati apprezzabili. Infatti non risulta esprimibile in termini di modelli lineari a media mobile, MA(n).

Abbiamo quindi deciso un approccio diverso, basato su due diverse stime del prezzo finale degli alberghi. Il primo stimatore è semplicemente ottenuto tramite regressione lineare dei prezzi comunicati dal *server* fino all'istante corrente. Il secondo è leggermente più complesso e si basa su un modello che utilizza i Processi di Poisson [36]. Abbiamo potuto verificare che quest'ultimo stimatore è meno accurato, quando il prezzo degli alberghi rimane basso e quindi abbiamo stabilito che se il prezzo rimane sotto una certa soglia (\$100 durante le gare), verrà utilizzato lo stimatore basato sulla regressione lineare, mentre al superamento della soglia subentrerà lo stimatore basato sui Processi di Poisson omogenei.

Per ottenere una stima del prezzo finale degli alberghi, abbiamo modellato l'arrivo di un'offerta che modifica il prezzo attuale attraverso il seguente Processo di Poisson di parametro  $\nu$ :

$$N_t \sim P(\nu t)$$

Per le proprietà dei Processi di Poisson, la seguente variabile aleatoria  $T$  indica che il tempo trascorso tra due arrivi è un'esponenziale di parametro  $\nu$ :

$$T \sim \Gamma(1, \nu)$$

Il valore atteso di  $T$  è:

$$E[T] = \frac{1}{\nu}$$

e il suo stimatore di massima verosimiglianza (MLE) è:

$$\hat{T} = \frac{1}{\bar{T}_n}$$

dove  $\bar{T}_n$  è la media campionaria di  $n$  osservazioni della variabile  $T$  ed è lo stimatore di massima verosimiglianza di  $\nu$ .

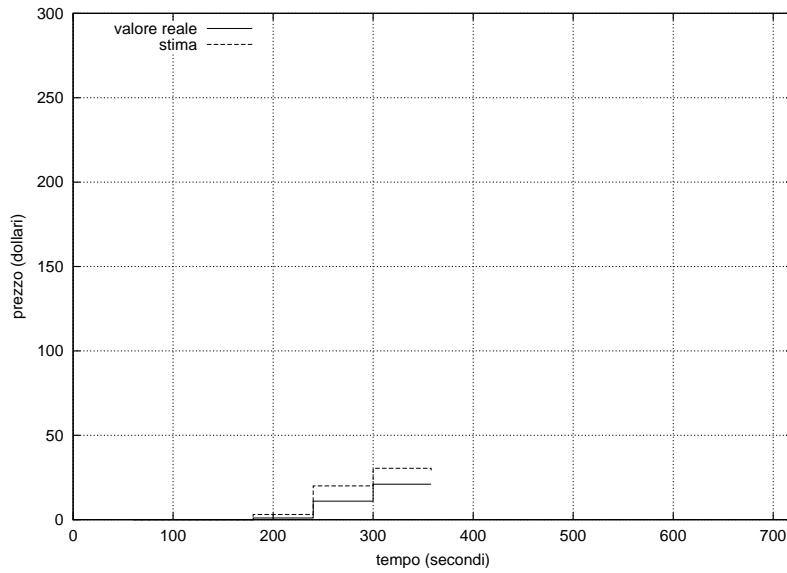
Se l'agente conoscesse quanto manca alla fine dell'asta, sarebbe possibile calcolare una stima del prezzo finale. Purtroppo non è noto a priori quando l'asta terminerà e quindi abbiamo scelto di considerare come termine la scadenza più vicina. Ad esempio, se ci troviamo al secondo minuto, utilizzeremo come termine delle aste il quarto minuto (quando la prima asta chiuderà), mentre se ci troviamo a metà del settimo minuto, la prossima scadenza sarà all'ottavo minuto. La stima del prezzo finale si ottiene dividendo il tempo rimasto per  $\hat{T}$ ; in questo modo otteniamo una stima di quante volte il prezzo corrente aumenterà prima della fine dell'asta. Moltiplicando questo numero per la media campionaria degli incrementi di prezzo, otterremo una stima di quanto il prezzo corrente aumenterà prima della fine dell'asta.

In realtà per la formulazione di questo stimatore sono state fatte diverse ipotesi semplificative che non trovano riscontro nel modello reale. Abbiamo assunto che gli arrivi degli incrementi dei prezzi si possano descrivere mediante un Processo di Poisson, ma per essere descritti come tali, dovrebbero soddisfare alcune proprietà tipiche di questi processi. Ad esempio gli incrementi di prezzo dovrebbero essere indipendenti, se si considerano intervalli di tempo disgiunti, però questo non è garantito a causa del fatto che il modello reale è non lineare e tempo variante. Inoltre bisogna notare che le aste considerate sono eventi ad orizzonte limitato, ipotesi che però non è vera per i processi stocastici in generale. Fortunatamente la limitata accuratezza di questo stimatore è in parte compensata dalla sua semplicità e dal fatto che è computabile in tempi molto rapidi.

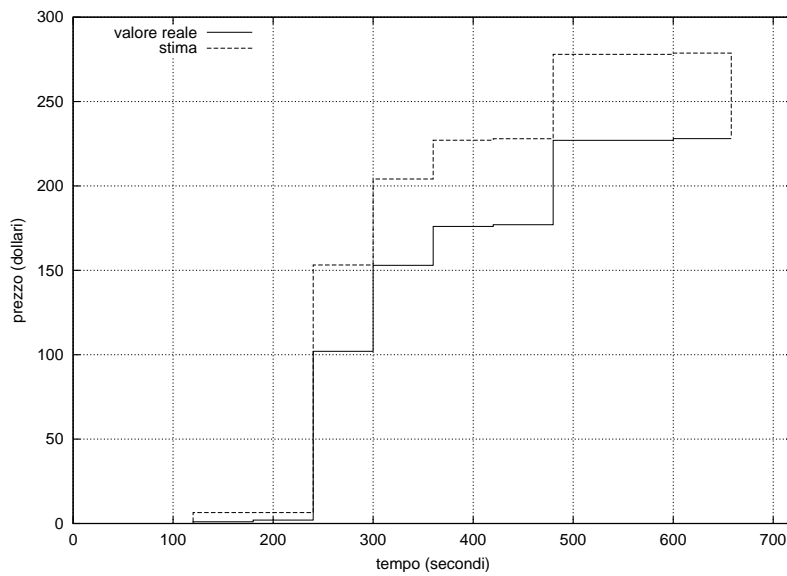
Come esempi del funzionamento di questi stimatori si possono considerare i grafici in Figura 4.3.

### 4.2.3 Asta dei biglietti per le attrazioni

Anche il mercato dei biglietti per le attrazioni non è descrivibile accuratamente mediante un modello lineare. I prezzi infatti dipendono dalle azioni di



(a) Stimatore basato su regressione lineare (*Tampa Towers*, giorno 3, istanza 6994).



(b) Stimatore basato su Processo di Poisson (*Shoreline Shanties*, giorno 4, istanza 6994).

Figura 4.3: Esempi di utilizzo dei due stimatori per le aste degli alberghi.

tutti gli altri agenti, di cui il nostro agente non ha praticamente nessuna informazione. In questo caso, l'agente conosce solo la domanda e l'offerta correnti. Come nei casi precedenti, abbiamo tentato diversi approcci per ottenere uno stimatore dei prezzi sufficientemente accurato.

Inizialmente abbiamo provato a utilizzare come stima, l'intersezione delle due rette di regressione, quella della domanda e quella dell'offerta. In questo modo abbiamo cercato di ottenere quello che in prima approssimazione può essere considerato il prezzo di equilibrio (anche se le quantità messe all'asta non sono rese note agli agenti). Tuttavia questo approccio non ha portato ai risultati sperati. Infatti lo stimatore tende a oscillare troppo e in questo contesto, la caratteristica della mancanza di oscillazioni ha un'importanza almeno pari a quella dell'accuratezza, perché cambi troppo repentini nelle stime portano a prendere decisioni penalizzanti per l'agente.

Abbiamo pertanto deciso di prendere come stima semplicemente la media tra i valori di domanda e offerta attualmente noti. Questo stimatore non è molto accurato, ma risulta avere meno oscillazioni del precedente e quindi gli è stato preferito. In Figura 4.4 è possibile notare come lo stimatore basato sulla regressione lineare abbia delle ampie oscillazioni, soprattutto all'inizio della sessione di gioco, cosa che il secondo stimatore non ha.

## 4.3 Strategia

Il comportamento dell'agente è determinato da poche e semplici regole che partendo dall'analisi delle informazioni a disposizione, permettono di decidere cosa comprare e a chi assegnarlo. Per fare questo l'agente mantiene aggiornate le liste dei beni posseduti e dei beni desiderati e tiene traccia dell'allocazione corrente.

All'inizio della sessione di gioco, quando vengono forniti i dati sui clienti, l'agente costruisce la lista dei beni desiderati, includendo tutto il necessario per soddisfare completamente le richieste dei clienti. Ad ogni cliente viene assegnato un pacchetto che rispetta le sue preferenze. Il tipo d'albergo è de-

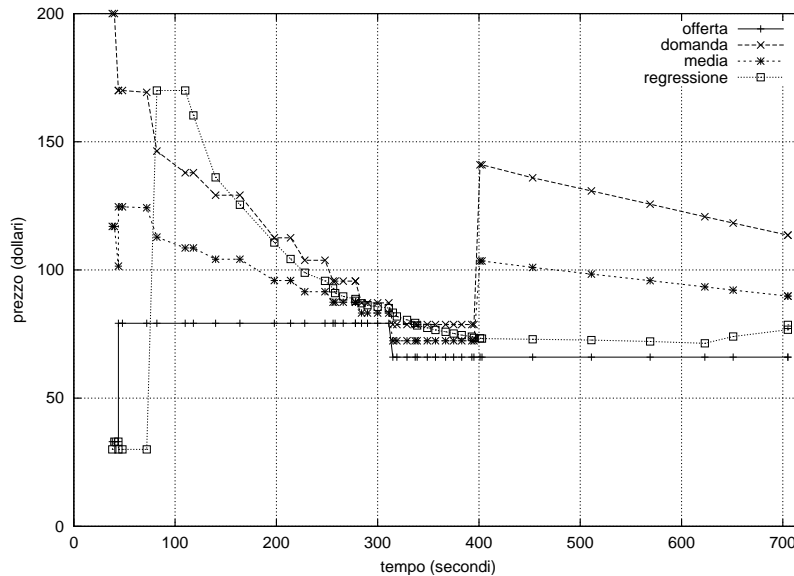


Figura 4.4: Confronto tra gli stimatori per le aste dei biglietti per le attrazioni.

ciso in base a quanto il cliente è disposto a pagare per il miglior albergo diviso per il numero di giorni di permanenza: se questa quantità è superiore a una certa soglia (\$25 durante le gare), cercheremo di prenotare l'albergo migliore per quel cliente. Abbiamo adottato questa strategia, perché il bonus vale per l'intero viaggio e non per notte. In questo modo, i clienti che rimangono pochi giorni — che quindi sono disposti a pagare di più per l'albergo migliore, in proporzione ai giorni di permanenza — tipicamente andranno alle *Tampa Towers*, le cui stanze in genere saranno più care e più difficili da ottenere. Anche i biglietti per le attrazioni saranno allocati in base a quanto i clienti sono disposti a pagare, dando la precedenza ai biglietti che vengono dati gratis agli agenti.

Ad ogni iterazione, verrà deciso per ogni cliente se, modificando il pacchetto a lui attualmente assegnato, si ottiene un potenziale guadagno oppure no. L'agente valuta quindi la rendita stimata di un insieme di pacchetti e tra questi sceglierà quello che promette il miglior guadagno. Verranno valutati tutti i seguenti pacchetti di viaggio alternativi:

- tutti quelli compresi tra il giorno d'arrivo e quello di partenza. Ad esem-

pio se l'allocazione corrente prevede per il cliente il pacchetto  $\langle 2, 5 \rangle$  (arrivo il giorno 2 e partenza il giorno 5), verranno presi in considerazione i pacchetti  $\langle 2, 3 \rangle$ ,  $\langle 3, 4 \rangle$ ,  $\langle 4, 5 \rangle$ ,  $\langle 2, 4 \rangle$  e  $\langle 3, 5 \rangle$ ;

- i pacchetti ottenuti anticipando di un giorno l'arrivo o ritardando di un giorno la partenza. Ad esempio se il pacchetto corrente è  $\langle 3, 4 \rangle$ , l'agente analizzerà i pacchetti  $\langle 2, 4 \rangle$  e  $\langle 3, 5 \rangle$ ;
- se il pacchetto corrente contiene un solo giorno di permanenza, l'agente prenderà in considerazione i pacchetti ottenuti spostando l'intero viaggio di un giorno avanti o indietro. Ad esempio dal pacchetto  $\langle 2, 3 \rangle$  cercherà delle alternative nei pacchetti  $\langle 1, 2 \rangle$  e  $\langle 3, 4 \rangle$ .
- inoltre per ogni pacchetto analizzato, compreso quello corrente, l'agente prenderà in considerazione l'ipotesi del cambio di albergo.

Per ogni pacchetto considerato l'agente calcola la differenza tra i ricavi aggiuntivi (che possono essere anche mancate spese) e le spese aggiuntive (che possono essere anche mancati guadagni, ad esempio quando si passa dall'albergo migliore a quello peggiore), cercando di sfruttare i beni non vendibili già posseduti e non attualmente allocati, che costituiscono un costo non recuperabile.

Per evitare di cambiare pacchetti troppo spesso (soprattutto all'inizio, quando le stime sono meno accurate), la strategia permette di cambiare un pacchetto solo se questo porterà un guadagno potenziale superiore a una certa soglia. Questa soglia aumenta progressivamente durante la sessione di gioco, per penalizzare i cambiamenti verso la fine delle aste, quando è più difficile e rischioso ottenere nuovi beni.

Per allestire un pacchetto *valido*, i beni critici da ottenere sono i biglietti aerei e le stanze d'albergo. E sebbene ci sia una disponibilità illimitata dei primi, solo trentadue stanze sono messe all'asta in totale ogni giorno. Tuttavia la richiesta massima può raggiungere le sessantaquattro unità, cosa che succede se ogni agente ha allocato nello stesso giorno una stanza a ogni cliente. C'è quindi il rischio di non ottenere tutte le stanze d'albergo desiderate e

in genere questo è sufficiente a rendere *non validi* i pacchetti che contengono i beni non più acquisibili. Per rimediare a questa situazione, l'agente verifica se le aste dove si vendono i beni desiderati sono ancora aperte. In caso contrario si attiva un algoritmo di recupero: l'agente effettua la riallocazione, tramite ricerca globale, di tutti i beni posseduti e di quelli desiderati ancora ottenibili e in seguito cerca quei clienti che sono rimasti senza un pacchetto valido. A questi tenta di dare il miglior pacchetto valido ancora ottenibile (quindi con libera scelta di aerei e di stanze d'albergo, le cui aste siano ancora aperte) effettuando una ricerca locale.

## 4.4 Allocazione delle risorse

L'algoritmo di allocazione delle risorse viene utilizzato durante la fase di recupero e all'inizio della sessione di gioco, quando vengono assegnati i biglietti per le attrazioni. Abbiamo utilizzato un approccio *greedy*, perché come spiegato nel Paragrafo 3.3.2, la ricerca dell'allocazione ottima è un problema NP-completo<sup>1</sup>. Abbiamo quindi voluto privilegiare la semplicità, la velocità e anche la possibilità di trattare un maggior numero di clienti, nel caso che le prossime edizioni della competizione lo prevedano.

L'algoritmo funziona così: per ogni cliente si effettua una ricerca locale sull'insieme dei pacchetti validi costruibili, senza considerare le attrazioni, per trovare quello che fornisce il maggior ricavo. Quando tutti i clienti hanno ricevuto, se possibile, un pacchetto valido, si procede all'allocazione degli intrattenimenti, che avviene da un punto di vista globale: si ordinano i valori che ogni cliente dà alle attrazioni dal più alto al più basso e si procede così ad assegnare tutti i biglietti posseduti, stando attenti a creare solo pacchetti validi.

È facile notare come anche l'allocazione dei biglietti non sia ottima. Ad

---

<sup>1</sup>Nel caso di soli otto clienti, il problema è trattabile, come è stato dimostrato lo scorso anno. Infatti i due agenti che hanno ottenuto i punteggi migliori hanno utilizzato un allocatore ottimo

esempio si considerino due clienti con le stesse preferenze per date (entrambi rimangono un solo giorno a Tampa) e alberghi e con i valori per le attrazioni di Tabella 4.1. Se abbiamo un biglietto per il Museo e uno per il Parco, entrambi validi per il giorno in cui i clienti sono in città, il biglietto per il Museo verrà dato al cliente  $x$ , dato che \$150 è il valore maggiore, mentre il biglietto per il Parco verrà dato al cliente  $y$  (unica possibilità rimasta); il valore totale dell'allocazione è \$210. Tuttavia per avere l'allocazione ottima, bisognerebbe dare il biglietto del Museo al cliente  $y$  e quello del Parco al cliente  $x$ , per avere un valore totale di \$230.

Cliente	Museo	Parco
$x$	150	90
$y$	140	60

Tabella 4.1: Tabella dei valori delle attrazioni per due clienti.

Per migliorare le prestazioni, l'algoritmo di allocazione delle risorse viene applicato a cinquemila permutazioni casuali di clienti tra cui si sceglie l'allocazione che fornisce il maggior ricavo [14]. Su una macchina dotata di processore Intel Pentium III a 800MHz e di 128MB di RAM, l'algoritmo impiega mediamente duecento microsecondi.

Durante la ricerca del pacchetto che permette di recuperare un cliente, l'agente invece utilizza un approccio completamente locale per l'allocazione delle risorse: oltre ad alberghi e voli, anche i biglietti per gli intrattenimenti vengono allocati in base alle preferenze del solo cliente considerato.

## 4.5 Creazione delle offerte

Nelle aste dei biglietti aerei, l'agente pone le offerte per la maggior parte dei voli desiderati all'inizio della sessione di gioco, cercando di spendere il meno possibile, ma cercando mantenere una certa flessibilità nella strategia. I rimanenti voli vengono acquistati a metà sessione, quando gli stimatori

sono più accurati e quindi i pacchetti sono già quasi definitivi, e comunque il prezzo dei voli probabilmente non ha ancora raggiunto il massimo.

Nelle aste per le stanze d'albergo, l'agente pone le offerte per le stanze desiderate, aumentando progressivamente l'aggressività delle offerte all'avvicinarsi della fine della sessione, perché tipicamente le ultime aste sono le più combattute. Inoltre l'agente cerca di comprare anche stanze non necessarie, se il prezzo si mantiene sufficientemente basso: se il prezzo è sotto i \$10, l'agente cerca di comprare almeno quattro stanze, mentre se il prezzo è sotto i \$20, l'agente fa un'offerta per almeno due stanze [14].

Nelle aste per i biglietti delle attrazioni, l'agente segue le strategie di un'agente ZIP (*Zero Intelligence Plus*), ideato da Dave Cliff [35] e adattato alle regole della TAC. L'agente ZIP si basa su semplici meccanismi di contrattazione e su semplici forme di apprendimento automatico. La chiave del comportamento di questi agenti risiede nella loro capacità di modificare il loro margine di profitto a seconda delle condizioni del mercato. I fattori in base ai quali l'agente aggiorna il suo margine di profitto sono quattro. Il primo dipende dalla condizione dell'agente: se sia *attivo*, ovvero se partecipa attivamente alle contrattazioni, o *non attivo*, ossia quando ha comprato o venduto tutti beni di sua competenza. Gli altri tre riguardano l'ultima offerta fatta: se il suo valore  $q$  è superiore o inferiore all'offerta che l'agente farebbe in questo istante  $p$ , se si tratta di un'offerta d'acquisto o di vendita e se è stata accettata o respinta (se è avvenuta una transazione o no). Un'agente di vendita ZIP aumenta il suo margine di profitto se l'ultima offerta è stata accettata e  $p \leq q$ . Diminuisce il suo margine solo se è ancora *attivo* e l'ultima offerta era un'offerta di vendita con  $p \geq q$  oppure era un'offerta di acquisto accettata con  $p \geq q$ . Similmente un'agente di acquisto ZIP aumenta il suo margine di profitto quando l'ultima offerta è stata accettata e  $p \geq q$ , mentre diminuisce il margine quando è attivo e l'ultima offerta era un'offerta d'acquisto respinta con  $p \leq q$  oppure era un'offerta di vendita accettata con  $p \leq q$ . La quantità utilizzata per aggiornare i margini di profitto è determinata in modo adattativo, utilizzando un processo di aggiustamento basato sulla "regola delta" di

Widrow-Hoff [37]. Per poter usare questo agente nella TAC abbiamo provveduto a modificare le regole, in modo da adattarle al mercato dei biglietti per le attrazioni. In questo mercato non si conosce l'ultima offerta, ma solo i valori della domanda e dell'offerta correnti. Abbiamo quindi utilizzato questi valori al posto di  $q$  e in particolare abbiamo utilizzato il valore della domanda corrente, quando  $q$  indica l'ultima offerta di acquisto, mentre abbiamo utilizzato il valore della offerta corrente, quando  $q$  indica l'ultima offerta di vendita. Inoltre agli agenti della TAC non viene rivelato quando avviene una transazione (cioè se un'offerta sia stata accettata oppure no) e per questo motivo abbiamo utilizzato una regola euristica per ovviare a ciò. Per le regole del mercato, la domanda corrente, se non avvengono transazioni, può solo salire e similmente l'offerta corrente nelle stesse condizioni può solo scendere. Invece quando avviene una transazione, generalmente l'offerta corrente aumenterà, oppure la domanda corrente aumenterà, perché una transazione elimina le offerte che la hanno generata. Perciò se teniamo traccia dei prezzi, riusciremo a individuare se un'offerta è stata accettata oppure no. La regola fallisce quando viene ritirata un'offerta, in questo caso i valori della domanda e dell'offerta correnti possono essere alterati senza che sia avvenuta una transazione.

Come molti agenti della scorsa competizione, anche *polimi\_bot* riesce a sapere quando la sessione di gioco sta per finire e quindi può modificare il suo comportamento di conseguenza. Quando tutte le aste delle stanze d'albergo sono chiuse oppure, in caso di alta latenza della rete, quando l'agente stima che manchino un certo numero di iterazioni alla fine (tre nelle gare), l'agente non cerca più di modificare l'allocazione corrente e inoltre cerca di vendere tutti i biglietti per le attrazioni inutilizzati al minimo prezzo stabilito (\$30 durante le gare).

## 4.6 Osservazioni

Durante le eliminatorie, l'agente comprava tutti i biglietti aerei all'inizio della sessione di gioco, per cercare di minimizzare le spese del bene, che in molti casi incide di più nei costi. Tuttavia questa strategia è stata modificata durante il turno successivo, perché abbiamo verificato che in questo modo la strategia dell'agente era molto rigida — il cambio di pacchetti diventava infatti molto oneroso — e quindi abbiamo optato per l'acquisizione dei biglietti in due momenti diversi della sessione (all'inizio vengono acquistati circa tre quarti dei biglietti, il resto viene acquistato a metà sessione) per aumentare la flessibilità dei pacchetti.

Durante le varie fasi della TAC è stato possibile notare alcuni punti deboli dell'agente, messi in luce dall'ambiente altamente competitivo delle gare ufficiali, rispetto alle partite di allenamento contro gli agenti messi a disposizione dall'Università del Michigan (*dummy\_buyer*). I problemi più gravi sono nati dai mercati delle camere d'albergo. In alcune situazioni l'agente ha comprato un numero eccessivo di camere: ciò è dovuto al fatto che cambiare albergo a un cliente, durante la sessione, è sempre rischioso, perché un'offerta è accettata solo se “batte la quotazione” (si veda a pagina 39) e quindi, c'è l'eventualità di ottenere stanze in entrambi gli alberghi per lo stesso cliente. Un altro motivo che spiega alcune cattive prestazioni dell'agente, è dato dall'insufficiente accuratezza di alcuni stimatori: abbiamo potuto verificare che, soprattutto nelle aste degli alberghi, gli stimatori non si sono dimostrati sempre affidabili, quindi, in alcune situazioni l'agente non è stato in grado di prevedere che il prezzo in alcune aste sarebbe salito di molto e di provvedere cambiando opportunamente il pacchetto (ad esempio scegliendo l'altro albergo).

## 4.7 Implementazione

Per sviluppare l'agente *polimi\_bot* abbiamo utilizzato il linguaggio C++; in questo modo abbiamo potuto utilizzare le classi *Bidding Agent*, messe a di-

sposizione dall'Università del Michigan, che forniscono un aiuto nella progettazione degli agenti, ponendosi a un livello intermedio tra la strategia ad alto livello da implementare e la piattaforma *AuctionBot*. Le classi *Bidding Agent* mettono a disposizione dei meccanismi che utilizzano la API descritta nella Sezione 3.4.2 allo scopo di aiutare lo sviluppatore a gestire la sincronizzazione delle informazioni tra il *server* e l'agente. Le classi modificano le offerte automaticamente, seguendo le politiche implementate dallo sviluppatore, e tengono traccia delle informazioni ricevute dalla piattaforma *AuctionBot*, come ad esempio le quotazioni correnti, le transazioni avvenute e lo stato delle aste. Inoltre forniscono informazioni sulle offerte: quando scadono e se sono state accettate o respinte. In questo modo chi sviluppa l'agente si deve preoccupare solo dell'implementazione della strategia ad alto livello e non della gestione dei messaggi XML da inviare al *server*, in quanto questa gestione avviene in modo trasparente allo sviluppatore.

Per quanto riguarda la struttura interna dell'agente, abbiamo sviluppato un gruppo di classi che descrive le aste, un altro gruppo di classi che implementa le strategie dell'agente *ZI-Plus* (si veda la Sezione 4.5) e infine una classe che fa da coordinatore e che contiene il nucleo dell'agente, ovvero strutture dati e algoritmi che costituiscono la strategia descritta in Sezione 4.3.

Il gruppo di quattro classi che descrivono le aste è formato da una classe per ognuno dei tre tipi di mercato e da una classe base che contiene tutte le strutture comuni a tutti i mercati. Queste classi si interfacciano con le classi *Bidding Agent* per avere informazioni sullo stato corrente dell'asta e hanno le strutture dati, usate per tenere traccia delle quotazioni. Inoltre le classi che descrivono le aste hanno i metodi che vengono utilizzati dal coordinatore per aggiornare gli stimatori, costruire le offerte e inviarle al *server*.

Il gruppo di tre classi che implementa le strategie dell'agente *ZI-Plus* è costituito da una classe che descrive l'agente venditore, un'altra che descrive l'agente compratore e una classe base astratta che contiene metodi e attributi comuni ai due agenti. La classe che descrive i mercati dei biglietti per

le attrazioni utilizza queste classi al fine di ottenere i prezzi a cui fare un'offerta d'acquisto per i biglietti desiderati o un'offerta di vendita per i biglietti posseduti ma non utilizzati.

Infine il coordinatore implementa il nucleo dell'agente vero e proprio. Infatti contiene gli algoritmi che permettono all'agente di decidere cosa comprare e a chi allocarlo. Contiene le strutture dati che mantengono la lista dei beni posseduti, quella dei beni desiderati e quella dei beni inutilizzati; tiene traccia dell'allocazione corrente e della spesa effettuata; mantiene le informazioni relative al tempo trascorso e al numero di iterazioni effettuate, da cui riesce a stimare l'approssimarsi della fine della sessione di gioco. Nel ciclo principale l'agente esegue i seguenti passi:

1. richiede al *server* quanto tempo è trascorso e in base a questo decide se si trova vicino alla fine della sessione;
2. aggiorna la lista dei beni posseduti, quella dei beni desiderati e quella dei beni non utilizzati, in base alle quantità di beni che ha comprato o venduto nel ciclo precedente;
3. se la lista dei beni desiderati contiene dei beni che non sono più ottenibili, perché il mercato in cui venivano scambiati è chiuso, l'agente procede alla riallocazione delle risorse;
4. aggiorna gli stimatori;
5. cerca di recuperare gli eventuali clienti senza un pacchetto *valido*, costruendo dei pacchetti alternativi;
6. in base ai nuovi stimatori e alle informazioni aggiornate, l'agente ricompila la lista dei beni desiderati, cercando un nuovo insieme di pacchetti *validi* che producano un punteggio atteso più elevato (si noti che tutto questo avviene se non si è vicini al termine della sessione);
7. aggiorna l'allocazione corrente, se ne è stata trovata una migliore;

8. costruisce le offerte (l'aggressività delle offerte è maggiore se si è vicini al termine della sessione);
9. invia le offerte al *server*.

Per mantenere una certa flessibilità, abbiamo reso parametrico il nostro agente. In Appendice A sono riportati tutti i parametri su cui si può agire per modificare il comportamento dell'agente.

## 4.8 Formalizzazione delle regole

È possibile formalizzare le regole che sono alla base della strategia dell'agente *polimi\_bot*, utilizzando regole composte da *trigger*, *condizione* e *azione*. Il *trigger* specifica l'*evento* in seguito al quale viene attivata la corrispondente regola. Quando il *trigger* diventa attivo, viene valutata la condizione; se la *condizione* è verificata, l'*azione* viene eseguita. La *condizione* è opzionale: se non è presente, l'*azione* viene comunque eseguita. La sintassi utilizzata per definire le regole è del tipo:

```
on trigger  
[if condizione]  
do azione
```

Nel seguito sono elencati i *trigger* che possono essere attivati dagli eventi che si possono verificare durante ogni sessione. Sono elencate le diverse funzioni e i diversi predicati utilizzabili per definire le *condizioni*. Sono inoltre elencate le *azioni* da eseguire quando la condizione è verificata. Infine sono elencate le regole che definiscono la strategia dell'agente *polimi\_bot*.

### 4.8.1 Trigger

Durante ogni sessione si possono verificare diversi eventi che attivano i seguenti *trigger*. Il trigger *BidInfoUpdated* si attiva quando l'agente ha ottenuto i dati aggiornati riguardo alle aste. Il trigger *UpdatingDone* si attiva quando l'agente ha portato a termine le operazioni di aggiornamento

delle sue strutture dati (stimatori, liste dei beni posseduti, ecc.). Il trigger *WishedGoodNotAvailable* si attiva quando un bene desiderato non è più acquisibile. Il trigger *ReallocationDone* si attiva quando è avvenuta una riallocazione dei beni. Il trigger *CheckingCurrentUtility* si attiva quando si sta verificando l'utilità fornita dal pacchetto corrente per il cliente esaminato. Il trigger *EvaluatingCurrentClient* si attiva quando si sta valutando se è conveniente cambiare il pacchetto corrente per il cliente esaminato. Il trigger *WishListUpdated* si attiva quando la lista dei beni da acquistare è stata aggiornata.

### 4.8.2 Condizioni

Le *condizioni*, che devono essere valutate ogni volta che si attivano delle regole, sono espressioni booleane in cui possono comparire le seguenti funzioni e i seguenti predicati.

#### Funzioni

Per verificare le condizioni della regola in esame, l'agente utilizza le seguenti funzioni. La funzione  $utility(c, p)$  restituisce l'utilità del pacchetto  $p$  per il cliente  $c$ . La funzione  $evaluate(c, p, p')$  restituisce il guadagno stimato ottenuto assegnando al cliente  $c$  il pacchetto  $p'$ , in sostituzione del pacchetto  $p$ . La funzione  $changeHotel(p)$  restituisce il pacchetto ottenuto cambiando il tipo di albergo nel pacchetto  $p$ . La funzione  $packageThreshold(t)$  restituisce il valore della soglia che permette il cambiamento di un pacchetto all'istante  $t$ . La funzione  $hotelThreshold(t)$  restituisce il valore della soglia che permette il cambiamento del tipo di albergo in un pacchetto all'istante  $t$ . La funzione  $currentTime()$  restituisce l'istante corrente.

#### Predicati

Nella base dei fatti possono essere presenti i seguenti predicati. Il predicato  $CurrentClient(c)$  è vero se si stanno esaminando le alternative per il cliente

c. Il predicato *CurrentPackage(p)* è vero se si stanno esaminando i pacchetti alternativi al pacchetto *p*. Il predicato *AlternativePackage(p)* è vero se *p* è uno dei pacchetti alternativi da esaminare. Il predicato *NewPackage(p)* è vero se si vuole assegnare il pacchetto *p* al cliente esaminato.

### 4.8.3 Azioni

Quando la *condizione* di una regola è stata verificata, l'agente può eseguire una o più delle seguenti azioni. L'azione *assert* viene eseguita quando si vuole asserire un predicato nella base dei fatti. L'azione *delete* viene eseguita quando si vuole eliminare un predicato dalla base dei fatti. L'azione *updateOwnedGoods* viene eseguita quando si vuole aggiornare la lista dei beni posseduti. L'azione *checkWishList* viene eseguita quando si vuole controllare che la lista dei beni desiderati contenga solo beni acquisibili. L'azione *updateUnusedGoods* viene eseguita quando si vuole aggiornare la lista dei beni non utilizzati. L'azione *updateEstimations* viene eseguita quando si vogliono aggiornare le stime dei prezzi. L'azione *startAltPackageEval* viene eseguita quando si vuole procedere alla valutazione dei pacchetti alternativi. L'azione *reallocation* viene eseguita quando si vuole effettuare una riallocazione; L'azione *reallocateCurrentClient* viene eseguita quando si vuole cercare un pacchetto *valido* per il cliente esaminato. L'azione *checkClientUtility* viene eseguita quando si vuole verificare le utilità fornite dai pacchetti attualmente assegnati ai clienti. L'azione *updateWishList* viene eseguita quando si vuole aggiornare la lista dei beni da acquisire. L'azione *updateZIPLimits* viene eseguita quando si vogliono aggiornare i limiti degli agenti ZIP. L'azione *constructBids* viene eseguita quando si vogliono costruire le offerte. L'azione *submitBids* viene eseguita quando si vogliono inviare le offerte al server.

### 4.8.4 Le regole dell'agente *polimi\_bot*

Utilizzando gli elementi appena definiti è possibile specificare le regole che determinano la strategia dell'agente. Queste regole sono elencate in Tabel-

la 4.2. La regola (1) permette di eseguire le operazioni di aggiornamento all'inizio del ciclo. La regola (2) permette di avviare la valutazione dei pacchetti alternativi. La regola (3) permette di iniziare le operazioni di recupero, nel caso in cui un bene desiderato non sia più acquisibile. In particolare la regola avvia la riallocazione dei beni. La regola (4) permette di iniziare la ricerca di quei clienti che sono rimasti senza un pacchetto *valido*. La regola (5) permette di recuperare il cliente esaminato se il suo pacchetto non è *valido*, cercando di assegnargli un nuovo pacchetto. La regola (6) permette di assegnare un pacchetto alternativo al cliente attualmente in esame. Viene valutato se il guadagno atteso, ottenuto cambiando il pacchetto *cp* con il pacchetto *ap*, supera la soglia prestabilita all'istante corrente. Viene inoltre valutato se il pacchetto alternativo *ap* è più conveniente del pacchetto *np*, eventualmente assegnato in precedenza. In caso affermativo al cliente viene assegnato il pacchetto *ap*, sostituendolo a quello assegnato in precedenza. La regola (7) è simile alla precedente. In questo caso però si considera il pacchetto alternativo, ottenuto cambiando tipo di albergo al pacchetto *ap*, se questo cambio porta un guadagno superiore alla soglia (del cambio di albergo) all'istante corrente. La regola (8) permette di effettuare le operazioni necessarie a inviare al *server* le offerte appropriate.

## 4.9 Sviluppi

Per la competizione, le regole sono state inizialmente implementate direttamente in C++ e non sono quindi modificabili senza alterare il codice. Ciò ha fatto perdere flessibilità all'agente ma ci ha permesso di trovare e correggere gli errori di programmazione in modo più semplice e rapido. Durante la competizione la strategia è stata collaudata ed è stato possibile correggere i difetti più gravi nel codice dell'agente. Abbiamo quindi iniziato a trasformare il nucleo dell'agente, utilizzando un'architettura a regole vera e propria. In questo modo sarà possibile sperimentare in modo semplice e veloce un'intera classe di strategie, in previsione della partecipazione alla prossima edizione

della *Trading Agent Competition*.

Id	Regola
1	<pre> <b>on</b> BidInfoUpdated <b>do</b> updateOwnedGoods(); checkWishList();     updateUnusedGoods(); updateEstimations() </pre>
2	<pre> <b>on</b> UpdatingDone <b>do</b> startAltPackageEval() </pre>
3	<pre> <b>on</b> WishedGoodNotAvailable <b>do</b> reallocation() </pre>
4	<pre> <b>on</b> ReallocationDone <b>do</b> checkClientUtility() </pre>
5	<pre> <b>on</b> CheckingCurrentUtility <b>if</b> (CurrentClient(?c) &amp;&amp; CurrentPackage(?p) &amp;&amp;     (utility(?p, ?c) == 0)) <b>do</b> reallocateCurrentClient() </pre>
6	<pre> <b>on</b> EvaluatingCurrentClient <b>if</b> (CurrentClient(?cc) &amp;&amp; CurrentPackage(?cp) &amp;&amp;     NewPackage(?np) &amp;&amp; AlternativePackage(?ap) &amp;&amp;     (evaluate(?cc, ?cp, ?ap) &gt;       packageThreshold(currentTime())) &amp;&amp;     (evaluate(?cc, ?cp, ?ap) &gt;       evaluate(?cc, ?cp, ?np))) <b>do</b> delete(NewPackage(?));     assert(NewPackage(?ap)) </pre>
<i>prosegue alla pagina successiva</i>	

<i>prosegue dalla pagina precedente</i>	
Id	Regola
7	<pre> <b>on</b> EvaluatingCurrentClient <b>if</b> (CurrentClient(?cc) &amp;&amp; CurrentPackage(?cp) &amp;&amp;       NewPackage(?np) &amp;&amp; AlternativePackage(?ap) &amp;&amp;       (evaluate(cc, cp, changeHotel(ap)) &gt;         evaluate(cc, cp, np)) &amp;&amp;       (evaluate(cc, ap, changeHotel(ap)) &gt;         hotelThreshold(currentTime())) &amp;&amp;       (evaluate(cc, cp, changeHotel(ap)) &gt;         packageThreshold(currentTime()))) <b>do</b> delete(NewPackage(?));       assert(NewPackage(changeHotel(ap))) </pre>
8	<pre> <b>on</b> WishListUpdated <b>do</b> updateZIPLimits(); constructBids(); submitBids() </pre>

Tabella 4.2: Le regole dell'agente *polimi\_bot*

# Capitolo 5

## I risultati della TAC-01

In questo capitolo descriviamo in dettaglio lo svolgimento dell'edizione di quest'anno della *Trading Agent Competition*. Dopo aver presentato i partecipanti alla competizione, riportiamo i risultati di tutte le fasi della gara. Infine illustriamo le strategie di alcuni agenti che si sono qualificati per la fase finale della competizione.

### 5.1 I partecipanti

All'edizione 2001 della *Trading Agent Competition* hanno preso parte ventotto agenti provenienti da otto diverse nazioni. La Tabella 5.1 elenca tutti gli agenti partecipanti a questa competizione, indicando anche l'università o l'istituto di appartenenza degli sviluppatori, la piattaforma e il linguaggio con cui gli agenti sono stati sviluppati e la composizione delle squadre autrici degli agenti.

Agente	Appartenenza	Piattaf.	Ling.	Membri
006	Swedish Institute of Computer Science	Unix	Prolog	Erik Aurell, Magnus Boman, Joakim Orrhult, Vincent Hauser, Lars Rasmusson
jboadw	McGill University	Linux	C++	Doina Precup, John Boadway
Mike	Carnegie Mellon University	Linux	C++	Katia Sycara, Yuriy Nevmyvaka, Gregg Economou, Maksim Tsvetovat, Pu Huang
SouthamptonTAC	University of Southampton	Unix	C++	Minghua He, Nick Jennings
UATrader	University of Arizona	Windows	Java	Daniel Zeng, Lihua Cao, Junho Lee
arc-2k	The Chinese University of Hong Kong	Windows	Perl	Chi Lun Tommy Chau
PainInNEC	NEC Research Institute	Linux	Perl	David Pennock, Tracy Mullen, Sun Park, Eric Glover
Spartacus	Stanford University	Unix	C++	Eugene Nudelman, Ryan Porter, Haowen Chen
umbctac	University of Maryland, Baltimore county	Unix	C++	Youyong Zou, Shanshan Liu
ATTac	AT&T Labs - Research	Linux	C++	Peter Stone, Michael Littman, Janos Csirik, David McAllester

*prosegue alla pagina successiva*

<i>prosegue dalla pagina precedente</i>				
Agente	Appartenenza	Piattaf.	Ling.	Membri
TacsMan	Stanford University	Unix	C++	Ryan Porter, Eugene Nudelman, Haowen Chen
Urlaub01	Penn State University	Unix	C++	Oliver Hansch
bang	NCST, Bangalore	Linux	C++	Ravi Prakash Putchala, Balaji Rajendran
livingagents	living systems AG	Linux	Java	Clemens Fritschi, Klaus Dorer
polimi_bot	Politecnico di Milano	Linux	C++	Alessandro Strada, Pierluca Lanzi
tricky	University of Essex	Windows	Java	Maria Fasli, Alexandros Zakas
whitebear	Cornell University	Linux	C++	Ioannis Vetsikas
CaiserSose	University of Essex, UK	Windows	Java	Maria Fasli, Yannis Korres
RedNEC	NEC Research, Rutgers University	Linux	Perl	Tracy Mullen, David Pennock, Sun Park, Eric Glover
Typhon	University of Essex	Windows	Other	Maria Fasli, George Rallidis
gekko	University of Southern California	Unix	C++	David Pynadath, Pragnesh Jay Modi, Ranjit Nair, Paul Scerri
RoxyBot	Brown University	Linux	C++	Amy Greenwald
poligen	Politecnico di Milano	Linux	Java	Davide Cappelletto, Bucci Stefano
DEFend	AT&T Labs - Research	Linux	C++	Peter Stone, Michael Littman, Janos Csirik, David McAllester
<i>prosegue alla pagina successiva</i>				

<i>prosegue dalla pagina precedente</i>				
Agente	Appartenenza	Piattaf.	Ling.	Membri
Retsina	Carnegie Mellon University	Linux	C++	Katia Sycara, Yuriy Nevmyvaka, Gregg Economou, Maksim Tsvetovat, Pu Huang
harami	Bogazici University Istanbul	Unix	Other	Taner Bilgiç, Engin Alper

Tabella 5.1: I partecipanti alla TAC-01

## 5.2 Il torneo preliminare

Il torneo preliminare è iniziato il 7 settembre scorso e si è svolto in due fasi. La prima fase (turno eliminatorio) ha determinato gli agenti qualificati alla fase finale. La seconda fase (turno di selezione) è servita a determinare la composizione dei due gruppi di agenti che, dopo aver superato le eliminatorie, hanno affrontato le semifinali. I punteggi, in base ai quali sono state stilate le classifiche dei vari turni, sono stati calcolati effettuando la media pesata dei punteggi ottenuti nelle varie sessioni di gioco. I pesi aumentavano progressivamente durante le gare. In particolare le sessioni del giorno  $i$  (dove  $i$  è il numero di giorni dall'inizio della gara) avevano un peso pari a  $\frac{1}{2} + \frac{i}{2}$ . Inoltre nel calcolo del punteggio, non sono stati considerati i dieci risultati peggiori ottenuti da ogni agente. Ogni sessione è stata giocata da otto agenti estratti a caso dall'insieme di tutti i partecipanti. Mediamente gli agenti hanno partecipato a più di cinquecento sessioni di gioco in questo torneo.

### 5.2.1 Il turno eliminatorio

Gli organizzatori hanno concesso ai partecipanti tre giorni di prove in ambiente realistico, quindi questo turno è iniziato ufficialmente il 10 settembre, anziché il 7, ed è terminato il 17 settembre. Se si osserva la Tabella 5.2, che

riporta la classifica finale del turno, si possono notare alcune caratteristiche della competizione. Il dato più evidente è che la deviazione standard è generalmente dello stesso ordine di grandezza della media dei punteggi. Ad esempio, la deviazione standard del primo classificato, che è quella minima, è pari a più del 25% del suo punteggio finale. Questo indica che la varianza dei punteggi da sessione a sessione è molto elevata e quindi per avere risultati significativi si sono dovute effettuare centinaia di sessioni. La causa principale dell'alta varianza dei punteggi è data dal fatto che le preferenze dei clienti sono estratte da distribuzioni uniformi. Altre cause possono essere imputabili a difetti di programmazione (che sono stati causa di molti dei punteggi negativi) e al fatto che gli agenti potevano essere modificati durante il torneo preliminare. I fatti dell'11 settembre hanno inoltre influito anche su questa competizione. Il crollo delle Torri Gemelle ha infatti provocato numerose interruzioni nel collegamento tra la rete GARR (che fornisce connettività alle università e agli istituti di ricerca italiani) e gli Stati Uniti. Questo spiega il motivo per cui il nostro agente ha partecipato a circa cento sessioni in meno degli altri. Si noti inoltre che l'agente *dummy\_buyer0*, introdotto dagli organizzatori, non ha realmente partecipato alla competizione ed è servito solo come punto di riferimento, infatti questo agente non è stato modificato durante tutte le gare a cui ha partecipato. Anche l'agente *LongAgent* non ha effettivamente partecipato alla competizione. È stato introdotto dagli organizzatori solo verso la fine di questa fase, per effettuare delle verifiche e ha potuto partecipare solo a poche decine di sessioni.

### 5.2.2 Gli agenti finalisti

Il turno è stato superato da sedici agenti: tra questi, i dodici agenti che hanno ottenuto i migliori punteggi nel turno eliminatorio sono stati automaticamente ammessi alla fase finale. Oltre a questi, gli organizzatori hanno ammesso alle finali gli agenti delle nazioni che altrimenti non sarebbero stati rappresentati e alcuni altri agenti a loro discrezione. Si noti che alle finali non poteva accedere più di un agente per ogni squadra.

#	Agente	$p$	$\bar{X}_n$	$s_n$	Min	Max	$n$
1	SouthamptonTAC	3425	3349	880.136	-715.14	5126.28	265
2	whitebear	3089	2940	1186.809	-2531.72	5025.54	267
3	DEFend	3084	2837	1773.898	-4879.21	5709.49	268
4	TacsMan	2990	2859	1190.803	-3009.81	5260.17	274
5	Retsina	2969	2868	1513.527	-4317.04	5162.4	271
6	Mike	2901	2850	1518.682	-5015.94	5547.61	272
7	Urlaub01	2890	2653	1799.540	-10201	5416	271
8	ATTac	2825	2719	1459.519	-2229.61	5315.77	272
9	umbctac	2784	2315	2787.367	-24912.6	5218.33	270
10	polimi_bot	2645	2134	5285.116	-66160.8	5100.71	186
11	livingagents	2639	1477	9659.059	-126330	4889.15	267
12	harami	2560	2347	1692.544	-7819.35	4824.85	271
13	PainInNEC	2554	2371	1811.077	-8254.82	4805.04	271
14	dummy_buyer0	2472	2357	1392.731	-4382.64	4680.28	268
15	CaaiserSose	2468	2075	2500.733	-13969	4815	270
16	RedNEC	2375	2034	2388.812	-9067.17	4890.77	275
17	Spartacus	2255	2154	1652.612	-5795.7	5015.81	273
18	Typhon	1820	1530	1933.605	-8270.13	5297.08	266
19	LongAgent	1550	1050	1084.147	-1629.8	2921.14	37
20	UATrader	1401	1034	2260.016	-6842.5	4662	271
21	jboardw	1097	529	2295.586	-9126.78	3883.6	266
22	RoxyBot	587	-282	7702.740	-115365	5034.14	272
23	arc-2k	235	227	750.778	-1580.32	2303.74	281
24	tricky	-102	-87	2101.118	-8490.37	4052.36	285
25	gekko	-164	-466	2475.672	-13004	3758.08	268
26	bang	-442	-1165	3443.064	-19898	4644	272
27	006	-591	-1063	3866.071	-43685	4645	268
28	poligen	-1595	-1975	2606.727	-11159.6	3793.23	189

Tabella 5.2: Classifica finale del turno eliminatorio della TAC-01.  $p$  indica il punteggio,  $\bar{X}_n$  è la media non pesata,  $s_n$  è la deviazione standard e  $n$  è il numero di sessioni giocate.

Sono stati ammessi ai turni successivi i seguenti agenti: *SouthamptonTAC*, *whitebear*, *ATTac*, *TacsMan*, *Retsina*, *Urlaub01*, *umbctac*, *polimi\_bot*, *living-agents*, *harami*, *PainInNEC*, *CaiserSose*, *jboadw*, *RoxyBot*, *arc-2k* e *006*.

### 5.2.3 Il turno di selezione

Questo turno è stato affrontato dagli agenti qualificati dalla fase precedente e si è svolto dal 24 settembre al 5 ottobre. Lo scopo di questo turno è stato di ottenere la composizione dei due gruppi di agenti che avrebbero affrontato le semifinali. In particolare i primi quattro e gli ultimi quattro agenti classificati avrebbero formato il primo gruppo, mentre gli altri otto agenti avrebbero formato il secondo gruppo. Ai sedici finalisti sono stati aggiunti gli agenti *ATTac-2000*, *dummy\_buyer0* e *bang*. Il primo è una copia dell'agente che ha ottenuto il miglior punteggio nella scorsa edizione ed è stato utilizzato come punto di riferimento per confrontare le prestazioni delle due competizioni; si noti che l'agente è stato leggermente modificato perché dalla scorsa edizione sono cambiate alcune regole. Il secondo è servito come riferimento rispetto al turno eliminatorio e non è stato incluso nella fase finale. Il terzo è stato incluso dagli organizzatori nel turno di selezione, ma non è stato considerato per la composizione dei gruppi per le semifinali.

In Tabella 5.3 è riportata la classifica finale del turno di selezione, da qui si può ricavare la composizione dei due gruppi che hanno affrontato le semifinali.

## 5.3 Le finali

Le finali si sono svolte il 14 ottobre. Gli organizzatori della competizione hanno invitato gli autori degli agenti qualificati a Tampa (Florida, USA), dove si è tenuto un *workshop* sulla *Trading Agent Competition*. Durante il *workshop* si è discusso della competizione e sono state illustrate le strategie di alcuni agenti. Le finali si sono svolte in due fasi: le semifinali e la finale. Alle semifinali sono stati ammessi tutti gli agenti che si sono qualificati nel turno

#	Agente	Punti	$\bar{X}_n$	$s_n$	Min	Max	$n$
1	SouthamptonTAC	3163.8	3084	855.3	-490.28	4550.07	312
2	whitebear	3120.8	3038	881.4	-1062.63	4871.68	318
3	Urlaub01	3075.5	2922	1197.8	-5812.41	4701.9	319
4	livingagents	3012.2	2855	1251.7	-4201	4710	305
5	TacsMan	2983.8	2838	1065.7	-1288.4	4591.37	315
6	CaiserSose	2869.8	2688	1219.7	-1994.24	4677.5	315
7	polimi_bot	2857.6	2745	980.8	-1717.42	4742.58	316
8	umbctac	2764.8	2677	1288.4	-3629.02	4796.3	313
9	RoxyBot	2731.6	2607	1379.7	-3766.68	5280	316
10	ATTac	2685.9	2594	1115.5	-3631.1	4675.85	317
11	Retsina	2674.9	2506	1517.6	-8380.26	4813.25	311
12	PainInNEC	2574.8	2379	1496.9	-4595.66	4571.1	314
13	ATTac2000	2412.1	2293	1569.2	-6864.69	4647.66	312
14	harami	2156.3	1816	2078.9	-7667.18	4873.98	316
15	dummy_buyer0	1673.0	1504	1523.7	-5758.38	4039.71	319
16	jboadw	1306.8	1189	1334.9	-4220.99	3529.46	318
17	006	1115.0	494	1893.0	-6024	4926	312
18	bang	703.1	707	1852.1	-8025	4626.15	313
19	arc-2k	-35.6	-174	1064.7	-8643.14	3143.49	320

Tabella 5.3: Classifica finale del turno di selezione della TAC-01. Nella colonna  $p$  sono riportati i punteggi, in  $\bar{X}_n$  le medie, in  $s_n$  le deviazioni standard e in  $n$  il numero di sessioni giocate.

eliminatório, organizzati in due gruppi a seconda della posizione ottenuta nel turno di selezione. Alla finale sono stati ammessi i primi quattro agenti classificati delle due semifinali. In particolare nel primo gruppo sono stati inclusi gli agenti: *SouthamptonTAC*, *whitebear*, *Urlaub01*, *livingagents*, *jboadw*, *006*, *bang* e *arc2k*. Nel secondo hanno gareggiato *TacsMan*, *CaiserSose*, *polimi\_bot*, *umbctac*, *RoxyBot*, *ATTac*, *Retsina* e *PainInNEC*. Nelle Tabelle 5.4 e 5.5 sono riportate le classifiche rispettivamente della prima e della seconda semifinale, mentre in Tabella 5.6 è riportata la classifica del turno finale della competizione. È importante notare che il numero di sessioni di gioco affrontate dagli agenti finalisti è estremamente limitato. Gli agenti che hanno superato le semifinali hanno partecipato infatti a sole ventiquattro sessioni in totale. Questo non permette di affermare che l'agente classificato al primo posto sia effettivamente migliore di tutti gli altri. Infatti l'alta varianza delle sessioni non permette di dedurre risultati statisticamente significativi dall'esiguo numero di sessioni giocate nella fase finale. Per questo motivo gli organizzatori sono stati attenti a non usare il termine "vincitore" per indicare l'agente *livingagents*. Si noti che l'agente *polimi\_bot* non ha potuto partecipare alla prima sessione di gioco delle semifinali per un problema di collegamento alla rete.

#	Agente	$p$	$\bar{X}_n$	$s_n$	Min	Max	$n$
1	livingagents	3660.2	3660	893.8	2057.54	4982.63	11
2	SouthamptonTAC	3614.5	3614	747.3	2373.81	4658.15	11
3	Urlaub01	3484.8	3485	924.1	1864.88	4874.64	11
4	whitebear	3469.7	3470	1043.0	1579.2	4645.88	11
5	006	3240.8	3241	1108.1	962.09	4196.88	11
6	arc-2k	1746.3	1746	1948.7	-1038.31	4153.12	11
7	jboadw	1716.7	1717	1281.3	0	4005.92	11
8	harami	94.4	94	2537.2	-6534.55	3380.82	11

Tabella 5.4: Classifica della prima semifinale della TAC-01. Nella colonna  $p$  sono riportati i punteggi, in  $\bar{X}_n$  le medie, in  $s_n$  le deviazioni standard e in  $n$  il numero di sessioni giocate.

#	Agente	$p$	$\bar{X}_n$	$s_n$	Min	Max	$n$
1	Retsina	3293.5	3293	630.9	2306.17	4116.57	11
2	ATTac	3249.2	3249	407.9	2535.69	3941.2	11
3	CaiserSose	3038.1	3038	640.9	1898.67	3806.1	11
4	TacsMan	2966.1	2966	595.2	2331.27	4236.4	11
5	PainInNEC	2905.9	2906	540.6	2142.72	3585.82	11
6	polimi_bot	2834.7	2835	1102.1	0	3796.55	11
7	umbctac	2772.9	2773	813.5	844.83	3880.15	11
8	RoxyBot	2112.4	2112	1478.7	-1778.29	3977.33	11

Tabella 5.5: Classifica della seconda semifinale della TAC-01. Nella colonna  $p$  sono riportati i punteggi, in  $\bar{X}_n$  le medie, in  $s_n$  le deviazioni standard e in  $n$  il numero di sessioni giocate. Si noti che il nostro agente *polimi\_bot* ha ottenuto un punteggio nullo alla prima sessione di gioco, per un problema di rete.

## 5.4 Le strategie

Le strategie utilizzate dagli agenti che hanno partecipato alla fase finale sono estremamente varie, sia negli strumenti adoperati sia nella complessità.

Ad esempio l'agente *livingagents* [44] è basato sulla piattaforma LARS (*living agents running system*), sviluppata dalla *living systems AG*<sup>1</sup> per gestire sistemi multiagente. Gli sviluppatori hanno utilizzato venti agenti di cinque diversi tipi. L'agente *TACManager* si occupa di attendere l'inizio della sessione e di attivare e disattivare gli altri agenti. Gli agenti di tipo *TACDataGrabber* si occupano di acquisire periodicamente dal *server* le nuove informazioni riguardanti le aste e di mettere a disposizione queste informazioni agli altri agenti. Ci sono cinque agenti di questo tipo: uno per l'asta dei voli d'andata, uno per l'asta dei voli di ritorno e tre per le aste dei biglietti per le attrazioni. Gli agenti di tipo *TACClient* provvedono a calcolare le mi-

<sup>1</sup><http://www.living-systems.com>

#	Agente	$p$	$\bar{X}_n$	$s_n$	Min	Max	$n$
1	livingagents	3670.0	3670	622.3	2331.99	4631.84	24
2	ATTac	3621.6	3622	691.6	2079.15	4877.1	24
3	whitebear	3513.2	3513	700.1	1563.85	4695.49	24
4	Urlaub01	3421.2	3421	698.3	1923.09	4640.21	24
5	Retsina	3351.8	3352	668.2	2150.35	4643.28	24
6	SouthamptonTAC	3253.5	3254	1466.9	-3119	4485.62	24
7	CaiserSose	3074.1	3074	656.2	1696.46	3965.41	24
8	TacsMan	2859.3	2859	1054.3	0	4230.26	24

Tabella 5.6: Classifica della finale della TAC-01. Nella colonna  $p$  sono riportati i punteggi, in  $\bar{X}_n$  le medie, in  $s_n$  le deviazioni standard e in  $n$  il numero di sessioni giocate.

giori combinazioni e ad inviarle agli agenti che generano le offerte. Ci sono otto agenti di questo tipo, uno per cliente. Gli agenti di tipo *TACAuctioneer* provvedono a effettuare le offerte provenienti dagli agenti del tipo precedente. Ci sono cinque agenti di questo tipo: uno per i voli d'andata, uno per quelli di ritorno, uno per l'albergo *Tampa Towers*, uno per l'albergo *Shoreline Shanties* e uno per le attrazioni. Infine l'agente *ResultGrabber* si occupa di recuperare i risultati delle precedenti sessioni, che vengono inserite in un'apposita base di dati, e si occupa di aggiornare il prezzo medio degli alberghi attualmente memorizzato. Questo agente non viene attivato durante la gara ma è attivato manualmente tra una sessione e l'altra. La strategia impiegata è molto semplice. All'inizio della sessione vengono attivati tutti gli agenti, vengono richiesti i prezzi dei voli di andata e ritorno e in base a questi e alla media dei prezzi degli alberghi viene calcolato il miglior pacchetto per ogni cliente. Quindi vengono piazzate le offerte relative ai pacchetti allestiti. Per quanto riguarda i biglietti delle attrazioni, l'agente effettua tutte le offerte di acquisto o vendita a \$80 basandosi sull'assunzione che questo sia il prezzo medio. L'agente non cambia, né ritira alcuna offerta, in quanto gli sviluppatori hanno avuto dei problemi relativi alla comunicazione con il *server*,

problemi che non hanno fatto in tempo a correggere e hanno quindi deciso di effettuare tutte le offerte all'inizio della sessione.

L'approccio utilizzato dagli sviluppatori di *ATTac-2001* [45] è invece basato sui dati e sull'apprendimento automatico e mantiene come punto di partenza l'allocatore ottimo utilizzato nella scorsa edizione. All'inizio della sessione, quando vengono rivelati i primi prezzi dei voli, l'agente calcola l'allocatione ottima, in base a quello che possiede e ai prezzi stimati. Decide quindi di comprare quei biglietti aerei il cui costo atteso al quarto minuto è superiore al beneficio atteso per non averli comprati. Allo stesso modo, un minuto prima che ogni asta degli alberghi chiuda, l'agente ricalcola l'allocatione ottima e decide di comprare quegli aerei il cui costo atteso è superiore al beneficio di non comprarli. Infine effettua offerte di valore pari al prezzo stimato per le stanze d'albergo basandosi sui beni posseduti, sui nuovi biglietti aerei ottenuti e su quante stanze prevede di riuscire a comprare. All'ultimo minuto, l'agente compra i voli, se sono necessari al completamento dell'allocatione calcolata. In parallelo a questo processo, l'agente compra e vende i biglietti per le attrazioni basandosi sui prezzi stimati. In particolare la stima dei prezzi per le attrazioni è data dalla media dei prezzi durante il turno di selezione. La stima dei prezzi dei voli è basata sui dati ricavati dalle precedenti sessioni da cui vengono determinati i parametri utilizzati nella descrizione della stima degli incrementi di prezzo. Infine i costi attesi delle stanze d'albergo sono stimati utilizzando i metodi per l'apprendimento automatico applicati ai dati ricavati dalle precedenti sessioni.

Il cuore dell'agente *whitebear* [46] è costituito dal pianificatore (*Planner*). Questo elemento costruisce il problema di ottimizzazione utilizzando i prezzi stimati dei beni da comprare o vendere. Successivamente il pianificatore risolve il problema formulato, utilizzando euristiche che permettono di ottenere velocemente delle soluzioni spesso vicine all'ottimo. Il pianificatore permette anche di rieffettuare velocemente la valutazione dei piani nel caso in cui l'agente non sia riuscito ad ottenere i beni desiderati. Un altro elemento importante è quello che genera le offerte. Per quanto riguarda le aste dei

biglietti aerei, l'agente cercherà di comprare prima quei biglietti il cui prezzo tenderà a salire più rapidamente. Per le stanze d'albergo, l'agente effettua offerte più alte per le stanze che ritiene critiche. Infine nelle aste dei biglietti per le attrazioni, l'agente parte da offerte molto basse che verranno progressivamente aumentate. In seguito l'agente cercherà di vendere i biglietti inutilizzati secondo quanto stabilito dal pianificatore.

Gli sviluppatori dell'agente *Retsina* [47] hanno invece scelto approcci diversi. Hanno infatti realizzato un allocatore basato su metodi di simulazione Monte Carlo di tipo catene di Markov (*Markov Chain Monte Carlo methods* [38]) e di varianti dello schema *Simulated Annealing* [39]. Partendo da un'allocazione casuale, l'agente effettua una passeggiata aleatoria con deriva (*random biased walk* [40]) su tutte le possibili allocazioni. L'agente preferisce spostarsi nelle direzioni che garantiscono una utilità maggiore, ma ogni tanto effettua delle mosse svantaggiose per non rimanere bloccato ai massimi locali. I difetti di questo approccio sono la non ottimalità della soluzione trovata e la probabile lentezza rispetto alla soluzione fornita mediante programmazione lineare. Per la predizione dei prezzi delle varie stanze d'albergo, gli sviluppatori hanno utilizzato una matrice ricavata dall'analisi delle sessioni precedenti. La matrice ha come righe le sedici aste delle stanze d'albergo e come colonne la posizione che ne indica l'ordine di chiusura. Da questa matrice l'agente ottiene una stima del prezzo finale calcolando la media dei prezzi dall'istante corrente al termine della gara. Si noti che gli sviluppatori hanno deciso di realizzare personalmente il sistema di comunicazione con il *server*, senza quindi utilizzare gli strumenti messi a disposizione dagli organizzatori.

Anche l'agente *SouthamptonTAC* [48] sfrutta diverse strategie per affrontare i diversi problemi: per l'allocazione e per il completamento si basa sulla programmazione lineare, mentre per la stima dei prezzi utilizza i metodi della logica *fuzzy*. Ad esempio l'agente cerca di classificare le aste dei biglietti aerei, mediante *Fuzzy Recognition*, in una di quattro categorie: nella prima il cambiamento atteso di prezzo è piccolo, nella seconda il cambiamento è medio, nella terza è grande e nella quarta molto grande. Maggiore è il cam-

biamento atteso e prima l'agente cercherà di acquistare il biglietto relativo. Inoltre l'agente decide quali biglietti comprare in base ai prezzi correnti e a quanti biglietti servono per uno specifico giorno. La logica *fuzzy* è stata utilizzata anche nella predizione dei prezzi finali delle stanze d'albergo. Le regole utilizzate, per stimare di quanto il prezzo aumenterà prima della fine dell'asta, si basano su diversi fattori: ad esempio i prezzi correnti delle stanze, gli incrementi nei prezzi nell'ultimo minuto e l'istante corrente. Per quanto riguarda i biglietti per le attrazioni, l'agente cerca di accontentare quei clienti che sono disposti a pagare molto e cerca di vendere tutti i biglietti non utilizzati. Infine l'agente utilizza due formulazioni del programma lineare per l'allocazione e il completamento. La prima formulazione, utilizzata nei primi undici minuti (quando le aste delle stanze d'albergo sono ancora aperte), calcola quanti e quali biglietti aerei e stanze d'albergo acquistare. Mentre la seconda formulazione viene utilizzata nell'ultimo minuto e calcola quanti e quali biglietti aerei e per le attrazioni comprare.

L'agente *CaiserSose* [49] è strutturato in tre stadi: predizione, scelta delle offerte iniziali e controllo dei prezzi. Il primo stadio osserva i prezzi delle varie aste nei primi tre minuti e basandosi su questi dati l'agente cerca di predire l'aumento dei prezzi. Il secondo stadio decide quali offerte effettuare cercando di alloggiare i clienti che rimangono solo un giorno nell'albergo *Tampa Towers*, mentre tutti gli altri clienti nell'albergo *Shoreline Shanties* (a meno che nella fase di predizione questo albergo non sia risultato in prospettiva, molto più caro del primo). In base alle decisioni prese riguardo a dove far alloggiare i clienti, l'agente poi effettua le relative offerte per i biglietti aerei. Quindi l'agente alloca ai clienti quei biglietti per le attrazioni che garantiscono un buon guadagno e cerca di vendere gli altri. Il terzo stadio si occupa di controllare costantemente i prezzi nelle aste delle stanze d'albergo e provvede a chiamare la funzione di completamento che determina quante stanze comprare in ogni asta. Il completamento è effettuato in base a ciò che l'agente possiede e ai prezzi correnti. L'algoritmo è basato su regole e utilizza un'euristica *greedy* per trovare una soluzione che diminuisca le spese dell'agente.

Invece l'agente *TacsMan* [50] è basato sul componente *Evaluator* che trova la soluzione ottima del problema di ottimizzazione, utilizzando CPLEX<sup>2</sup>. In base alla soluzione l'agente può ricavare le utilità marginali di ogni bene e quindi può porre le offerte opportune nelle aste delle stanze d'albergo e dei biglietti per le attrazioni. Per quanto riguarda i biglietti aerei, l'agente ad ogni minuto valuta diverse opzioni. Per ogni cliente, l'agente calcola il punteggio atteso ottenuto non comprando biglietti, comprando solo l'andata o il ritorno oppure comprando entrambi i biglietti. Di queste possibilità l'agente seleziona quella che garantisce il punteggio maggiore.

Gli agenti si possono quindi dividere in due categorie: quelli che hanno utilizzato un'impostazione basata sui metodi della Ricerca Operativa e che quindi cercano in genere le soluzioni ottime dei problemi dell'allocazione e del completamento, e quelli che hanno utilizzato euristiche per ottenere una soluzione buona (seppure non ottima), senza ricorrere a strumenti esterni, privilegiando semplicità e rapidità di esecuzione. Tra gli agenti non ancora menzionati, l'agente *PainInNEC* [51], ad esempio, può essere inserito nel primo gruppo perché ha utilizzato il pacchetto CPLEX<sup>2</sup> per risolvere il problema di ottimizzazione. Anche l'agente *RoxyBot* [53] può essere messo nel primo gruppo, poiché si basa sull'impostazione di un programma lineare, sebbene non utilizzi algoritmi che garantiscono di trovare sempre la soluzione ottima. Mentre al secondo gruppo appartiene l'agente *UMBCTAC* [52] che è basato sulla versione dello scorso anno, anch'essa euristica. Infine va citato l'agente *dummy\_buyer* [54] che è stato sviluppato dagli organizzatori e, anche se non ha partecipato realmente alla competizione, si è comportato piuttosto bene relativamente alla sua complessità ed è stato un buon punto di riferimento per gli altri agenti. Si tratta infatti di un agente parametrizzato che si basa su regole molto semplici per decidere che cosa acquistare.

---

<sup>2</sup><http://www.cplex.com>



# Capitolo 6

## Analisi statistica dei risultati

In questo capitolo analizziamo approfonditamente i risultati della *Trading Agent Competition 2001*, utilizzando metodi statistici. Dopo aver discusso le motivazioni che ci hanno portato alla scelta del metodo, procediamo ad illustrare i risultati prodotti dall'analisi che abbiamo effettuato. In particolare per rispondere alla domanda: “Ci sono differenze significative nelle prestazioni degli agenti?” abbiamo utilizzato l'*analisi della varianza*. Mentre per rispondere alla domanda: “Quali agenti hanno prestazioni differenti?” abbiamo utilizzato diverse *procedure di confronto multiplo*. Con questi metodi cerchiamo di stabilire quali agenti si siano comportati meglio e cerchiamo di valutare le prestazioni *reali* dell'agente *polimi\_bot*. Inoltre cerchiamo di valutare l'efficacia della competizione nella promozione di strategie realmente valide.

### 6.1 La scelta del metodo

Nella *Trading Agent Competition 2001* (TAC-01) gli agenti sono stati valutati in base alla media dei punteggi ottenuti nelle diverse fasi della competizione, descritte nel Capitolo 5. Allo scopo di valutare le prestazioni dell'agente *polimi\_bot* e di verificare quanto sia efficace la TAC-01 al fine di promuovere le strategie migliori, abbiamo considerato i risultati del turno preliminare e

del turno di selezione, ma non i risultati della finale. Il turno preliminare e quello di selezione sono i più adatti a essere analizzati con metodi statistici, perché durante questi turni gli agenti hanno partecipato a un elevato numero di sessioni. Viceversa abbiamo escluso dall'analisi i risultati della fase finale perché durante quest'ultima fase si sono svolte solamente poche sessioni e pertanto i test utilizzati non ci avrebbero permesso di ottenere risultati statisticamente interessanti. L'analisi dei risultati della competizione consiste nel comparare, utilizzando gli opportuni metodi statistici, le prestazioni ottenute dai vari agenti in ogni turno, per determinare se la classifica rispecchia realmente la bontà delle strategie dei partecipanti. Per effettuare questi confronti, uno dei metodi più utilizzati in letteratura si basa su *t-test* multipli, tramite i quali si confrontano le prestazioni di ogni coppia di agenti. Questo approccio tuttavia non è indicato per confrontare più di due gruppi, perché diminuisce la capacità del test di evidenziare la presenza di differenze significative, ovvero determina una diminuzione della *potenza* dei singoli test. Contemporaneamente l'utilizzo di *t-test* multipli aumenta la probabilità che il test riscontri differenze tra gli agenti quando non ci sono, ovvero aumenta la probabilità di commettere *errori del primo tipo*.

Per questo la metodologia corretta prevede l'utilizzo dell'*analisi della varianza* (ANOVA o *ANalysis Of VAriance* [41]) seguita da appropriate *procedure di confronto multiplo* [41], note anche come *post hoc test*. In particolare abbiamo utilizzato l'*analisi della varianza a una via* (1-way ANOVA), in quanto abbiamo un'unica variabile dipendente costituita dai punteggi degli agenti e un unico fattore o variabile indipendente che permette di associare i punteggi agli agenti corrispondenti. Nella nostra analisi abbiamo escluso i dieci peggiori risultati di ogni agente, dato che in generale i punteggi più bassi derivano da errori nella programmazione e non dalla strategia dell'agente. Inoltre l'esclusione dei dieci peggiori risultati ci permette di migliorare la significatività dei risultati dei test che sono piuttosto sensibili ai valori anomali (*outlier*).

## 6.2 Analisi della varianza

L'*analisi della varianza* (ANOVA) ci permette di rispondere alla domanda: “Ci sono differenze significative nelle prestazioni degli agenti?”. L'ANOVA è la generalizzazione del *t-test* di Student, a molti gruppi. Il *t-test* è generalmente utilizzato per verificare se due gruppi hanno la stessa media. L'ipotesi che i gruppi abbiano media simile è chiamata *ipotesi nulla* ( $H_0$ ), mentre l'ipotesi che i gruppi abbiano media diversa è l'*ipotesi alternativa* ( $H_1$ ). Affinché il test dia risposte significative, la popolazioni da cui sono estratti i gruppi da confrontare devono avere distribuzione *normale* e devono avere la stessa varianza. Il test consiste nel calcolare il valore della statistica *t* e di controllare se questo valore appartenga o meno alla *regione critica*. Questa regione contiene tutti i valori per cui si *rifiuta* l'ipotesi nulla: se *t* cade nella regione critica, i gruppi confrontati avranno medie differenti; se *t* cade al di fuori della regione critica, i gruppi confrontati avranno medie simili. I risultati del *t-test* sono legati al valore del livello di significatività  $\alpha$  (tipicamente tra 0.01 e 0.10) che indica la probabilità di rifiutare l'ipotesi nulla quando essa è vera. Quindi il livello  $\alpha$  è pari alla probabilità di commettere un *errore del primo tipo*, cioè assumere che le medie dei due gruppi siano diverse, quando in realtà sono simili. In generale minore è il valore di  $\alpha$  e più il test è *conservativo*. Un test si dice *conservativo* quando tende a non rifiutare l'ipotesi nulla, ovvero tende a non individuare le differenze tra i gruppi.

Anche per l'analisi della varianza si fa l'ipotesi che i gruppi siano estratti da popolazioni *normali*, con la stessa varianza. L'ANOVA si propone di determinare se le medie dei gruppi siano significativamente differenti. Quindi l'*ipotesi nulla*  $H_0$  è che non ci siano differenze tra le medie, mentre l'*ipotesi alternativa*  $H_1$  è che almeno uno dei gruppi abbia una media differente da quella degli altri. Per determinare se i gruppi abbiano media differente oppure no, è necessario calcolare la *varianza tra gruppi* ( $s_{bet}^2$ ), ovvero la varianza della media campionaria moltiplicata per il numero di osservazioni, e la *varianza intra-gruppi* ( $s_{wit}^2$ ), ovvero la media delle varianze dei gruppi. Se le medie dei gruppi fossero simili, il rapporto tra  $s_{bet}^2$  e  $s_{wit}^2$  dovrebbe essere pari circa a

uno, mentre se le medie fossero diverse, il rapporto sarebbe molto maggiore di uno. Si confronta quindi il rapporto tra le varianze con un *valore critico*, dipendente da  $\alpha$ . Se il rapporto è maggiore del valore critico, si conclude che almeno un gruppo ha media differente dagli altri (si *rifiuta* l'ipotesi nulla).

Si noti che l'analisi della varianza non fornisce però *alcuna indicazione* su quali e quanti gruppi si differenzino dagli altri, ma fornisce invece un'informazione di tipo qualitativo. Per avere informazioni più specifiche bisogna applicare una o più *procedure di confronto multiplo*.

### 6.3 Procedure di confronto multiplo

Le *procedure di confronto multiplo* ci permettono di rispondere alla domanda: "Quali agenti hanno prestazioni differenti?". Queste procedure sono note anche come *post hoc test* perché tipicamente si effettuano dopo l'*analisi della varianza*. Le procedure di confronto multiplo sono generalmente basate sul *t-test* ma includono i necessari correttivi per il fatto che si stanno confrontando più di due gruppi. Viceversa un *t-test* senza correzioni non fornirebbe dei risultati sufficientemente affidabili. Ad esempio supponiamo di dover confrontare tre gruppi, utilizzando un test con livello di significatività  $\alpha = 0.05$ . In questo caso dovremmo effettuare tre *t-test*. Per ogni test avremmo una probabilità di commettere errori del primo tipo (riscontrare differenze tra i gruppi quando non ci sono) pari al 5%. Perciò quando si considerano i tre test contemporaneamente, ci si aspetta di commettere un errore pari circa al 15%, ovvero pari circa alla somma degli errori dei singoli test. Pertanto avremmo un test più pronò a trovare differenze anche quando in realtà i gruppi hanno la stessa media. Introducendo dei fattori di correzione sul parametro  $\alpha$  otteniamo varie procedure che si differenziano per essere più o meno conservative. Nel nostro caso abbiamo utilizzato quattro tipi di test: il test di *Bonferroni*, il test di *Scheffé*, il test di *Tukey* e il test SNK (*Student-Newman-Keuls*). In particolare i test di *Scheffé* e *Bonferroni* sono quelli più conservativi, il test SNK è quello meno conservativo, mentre il test di *Tukey* si pone a un livello

intermedio.

## 6.4 Osservazioni sull'applicabilità

Come riportato in precedenza, i metodi che vogliamo utilizzare si basano su alcune ipotesi relative alle popolazioni da cui sono estratti i gruppi da analizzare. Viene richiesto che le varianze dei gruppi confrontati siano uguali e che le distribuzioni da cui sono estratte le osservazioni che formano i diversi gruppi siano *normali*. Nel nostro caso queste ipotesi non vengono rispettate. Infatti tramite il test di omogeneità della varianza di *Levene* [42] abbiamo verificato che le varianze sono differenti. Tuttavia l'*analisi della varianza* è un metodo robusto e riesce a tollerare le differenze di varianza tra i gruppi che nel nostro caso non sono eccessive. Infatti la componente maggiore della varianza è dovuta alle caratteristiche della competizione ed è quindi comune a tutti gli agenti. Per quanto riguarda la forma delle distribuzioni, abbiamo verificato, tramite il test di normalità di *Kolmogorov-Smirnov* [43], che le popolazioni dei vari gruppi non sono *normali*. La non normalità delle distribuzioni deriva soprattutto dalla forma delle code delle distribuzioni ed è dovuta alle caratteristiche della competizione, come la non omogeneità delle varianze. Infatti i punteggi che gli agenti possono ottenere sono limitati superiormente ma non inferiormente e quindi la distribuzione dei punteggi è asimmetrica. Anche in questo caso però l'ANOVA è robusta, se le distribuzioni hanno determinate caratteristiche. Ad esempio nel nostro caso, la statistica *kurtosis* (indice relativo alla forma di una distribuzione) è positiva per tutti i gruppi. Questa condizione rende l'ANOVA più conservativa rispetto a un test condotto su popolazioni normali. Tuttavia, poiché il test ha evidenziato comunque una differenza nei punteggi (ha rifiutato l'ipotesi nulla), è evidente che la differenza tra le medie è estremamente significativa. Anche l'asimmetria (*skewness*) non altera i risultati del test grazie al *Teorema Limite Centrale*, a patto di avere un numero di casi sufficientemente elevato. Inoltre l'aver eliminato i dieci punteggi più bassi rende più robusto

il test, dato che l'ANOVA è meno affidabile in presenza di molti valori anomali (*outlier*). Per i *post hoc test* valgono le medesime considerazioni fatte per l'analisi della varianza: i *post hoc test* sono robusti nelle stesse condizioni in cui l'ANOVA è robusta e quindi anche nel nostro caso. Un altro fatto da considerare è che le nostre conclusioni sono puramente qualitative e non hanno pretese di generare una classifica alternativa della competizione, ovvero effettuare un ordinamento rigoroso in base ai risultati del test. Non abbiamo quindi la necessità di effettuare dei test più potenti di quelli utilizzati.

## 6.5 Il turno eliminatorio

Questo turno della competizione si è svolto dal 10 al 17 settembre. In questo periodo si sono affrontati ventotto agenti che hanno partecipato mediamente a oltre duecentocinquanta sessioni ciascuno, per un totale di circa mille sessioni complessivamente giocate.

### 6.5.1 Analisi statistica

In Tabella 6.1 sono riportati i risultati dell'*analisi della varianza* effettuata sui punteggi del turno preliminare. Come si può notare, il *p-value* è prossimo a zero. Quando il *p-value* è minore del livello di significatività  $\alpha$ , il risultato del test è il rifiuto dell'ipotesi nulla  $H_0$  che è vera quando le medie dei punteggi sono simili. Nel nostro caso  $\alpha = 0.05$  e  $p\text{-value} < \alpha$ , quindi si rifiuta l'ipotesi  $H_0$ . Pertanto si può affermare con sufficiente sicurezza che in questo turno ci sono agenti con prestazioni significativamente differenti.

	SS	df	MS	F	p-value
Tra-Gruppi	13053123999.118	27	483449037.004	195.196	.000
Intra-Gruppi	17035033771.426	6878	2476742.334		
Totale	30088157770.545	6905			

Tabella 6.1: ANOVA sui risultati del turno preliminare.

Le *procedure di confronto multiplo*, che abbiamo utilizzato per analizzare i punteggi del turno eliminatorio, forniscono due tipi di risultati. In particolare, il test di *Bonferroni* (Tabella 6.4) effettua confronti tra tutte le coppie di agenti, riportando per ogni coppia se le medie sono significativamente diverse oppure no. Invece il test *Student-Newman-Keuls* (Tabella 6.6) è un test di omogeneità che cerca i gruppi di agenti che hanno media simile. Si noti che il test di *Tukey* (Tabella 6.5 e Tabella 6.8) e il test di *Scheffé* (Tabella 6.3 e Tabella 6.7) forniscono entrambi i tipi di risultati.

Le tabelle che riportano i risultati dei test su ogni coppia di agenti si consultano per righe: su ogni riga è riportato il nome di un agente che viene confrontato con gli agenti che lo precedono in classifica. Il simbolo  $\simeq$  su fondo grigio indica che l'agente nella riga corrispondente ha presumibilmente prestazioni comparabili all'agente nella corrispondente colonna (l'ipotesi nulla è stata *accettata* o *non rifiutata*). Il simbolo  $<$  indica che le prestazioni dell'agente nella riga corrispondente sono molto probabilmente inferiori a quelle dell'agente nella corrispondente colonna (l'ipotesi nulla è stata *rifiutata*). I simboli su fondo grigio chiaro indicano che il test ha fornito una risposta solo *marginamente* significativa, ovvero con *p-value* compreso tra 0.01 e 0.10. Inoltre queste tabelle riportano nella prima colonna il tipo di implementazione utilizzata. La maggior parte degli agenti utilizza una strategia euristica (H) o basata su un *solver* (S). Non risulta una netta prevalenza di un metodo sull'altro, anche se nel turno eliminatorio, ai primi posti si sono classificati più agenti basati su *solver*. Non si può parlare però di raggruppamento di strategie. Si noti inoltre che la classificazione in base all'implementazione della strategia non è netta e di alcuni agenti non sono disponibili sufficienti informazioni.

Le tabelle che riportano i risultati dei test di omogeneità si consultano per colonne. In ogni colonna è riportato un gruppo di agenti le cui prestazioni sono presumibilmente omogenee. Maggiore è il *p-value* corrispondente e più omogeneo è il gruppo. Si noti che in generale i gruppi trovati con i test di omogeneità non sono disgiunti.

### 6.5.2 Osservazioni

Innanzitutto bisogna segnalare che i punteggi dell'agente *LongAgent* formano un gruppo anomalo (*outlier*). Infatti questo agente è stato introdotto dagli organizzatori a competizione già iniziata per effettuare delle verifiche. I risultati prodotti dalle analisi relative a questo agente non sono significativi, perché ha partecipato a molte meno sessioni degli altri agenti.

Dalla Tabella 6.3, che riporta i risultati del test di *Scheffé*, si può notare che questo test è il più conservativo di tutti. Infatti è quello che tende a rilevare meno le differenze fra gli agenti e ciò è rappresentato dall'elevata estensione dell'area grigia. Il test di Scheffé evidenzia che gli agenti classificati dal diciottesimo all'ultimo posto hanno prestazioni inferiori a quelle degli altri agenti. Gli agenti classificati dal primo al diciassettesimo posto hanno prestazioni simili, perché l'area bianca (e quella grigio chiaro che rappresenta una zona di transizione) presente nel triangolo che include i primi diciassette agenti è estremamente limitata. Le stesse considerazioni sono confermate dal test di *omogeneità* di Scheffé in Tabella 6.7. I gruppi che contengono i primi diciassette agenti sono molto ampi, segno che le prestazioni sono piuttosto omogenee (anche se bisogna ricordare che il test è molto conservativo). Inoltre si nota, anche in questo caso, una separazione abbastanza netta tra i gruppi contenenti i primi diciassette agenti e i gruppi che contengono gli ultimi undici agenti.

Dalla Tabella 6.4, relativa al test di *Bonferroni*, si può notare che anche questo test è conservativo sebbene lo sia meno del precedente. Come nel test di Scheffé, si nota la separazione piuttosto netta tra i primi diciassette e gli ultimi undici agenti. Essendo questo test meno conservativo, però, si può notare che l'area grigia è meno estesa rispetto al test di Scheffé. Il test di Bonferroni evidenzia tre sottogruppi non disgiunti, formati da agenti con prestazioni molto simili: un primo gruppo formato dagli agenti classificati dal primo al sesto posto, un secondo formato dagli agenti tra il secondo e il quattordicesimo posto e un terzo che comprende gli agenti tra il settimo e il diciassettesimo posto. Bisogna notare che la differenza di prestazioni tra

questi sottogruppi è comunque molto meno netta della differenza tra i primi diciassette agenti e gli ultimi undici agenti.

Il test di *Tukey* (Tabella 6.5) è leggermente meno conservativo del test precedente. Infatti rispetto al test di Bonferroni abbiamo una maggior estensione della zona di transizione (in grigio chiaro). Tuttavia per il test di *Tukey* valgono le stesse osservazioni fatte per il test di Bonferroni e anche il test di *omogeneità* di *Tukey* riportato in Tabella 6.8 suggerisce le medesime conclusioni.

Infine in Tabella 6.6 è riportato il test di *omogeneità* SNK (*Student-Newman-Keuls*). Si tratta del test meno conservativo di tutti, come si può notare dal fatto che i gruppi che riporta siano meno ampi di quelli evidenziati dagli altri test. Infatti il test SNK è l'unico che pone l'agente primo classificato in un gruppo a parte. È importante notare che anche questo test riporta la netta separazione tra i primi diciassette e gli ultimi undici agenti.

Si noti che l'agente *polimi\_bot* (decimo nel turno eliminatorio) appartiene al gruppo formato dai primi diciassette agenti. Quindi, secondo tutti i test, il nostro agente ha prestazioni comparabili a quelle dei primi classificati. Solo l'agente primo in classifica ha prestazioni superiori a quelle di *polimi\_bot*, secondo i test meno conservativi, come ad esempio il test SNK (Tabella 6.6).

## 6.6 Il turno di selezione

Questo turno della competizione si è svolto dal 24 settembre al 5 ottobre. In questo periodo si sono affrontati diciannove agenti che hanno partecipato mediamente a oltre trecento sessioni ciascuno, per un totale di più di settecentocinquanta sessioni complessivamente giocate.

### 6.6.1 Analisi statistica

In Tabella 6.2 sono riportati i risultati dell'*analisi della varianza* effettuata sui punteggi del turno di selezione. Anche in questo caso, il *p-value* è prossimo a zero. Quando il *p-value* è minore del livello di significatività  $\alpha$ , il risultato del

test è il rifiuto dell'ipotesi nulla  $H_0$  che è vera quando le medie dei punteggi sono simili. Nel nostro caso  $\alpha = 0.05$  e  $p\text{-value} < \alpha$ , quindi si rifiuta l'ipotesi  $H_0$ . Pertanto si può affermare con sufficiente sicurezza che in questo turno ci sono agenti con prestazioni significativamente differenti.

	SS	df	MS	F	p-value
Tra-Gruppi	4846493486.426	18	269249638.135	208.560	.000
Intra-Gruppi	7449024944.695	5770	1290992.191		
Totale	12295518431.121	5788			

Tabella 6.2: ANOVA sui risultati del turno di selezione.

Per analizzare i punteggi del turno di selezione, abbiamo utilizzato le stesse *procedure di confronto multiplo* con cui abbiamo analizzato il turno eliminatorio. I test che effettuano confronti tra tutte le coppie di agenti sono riportati in Tabella 6.10 (test di *Bonferroni*), Tabella 6.11 (test di *Tukey*) e Tabella 6.9 (test di *Scheffé*). I test di omogeneità sono riportati in Tabella 6.12 (test SNK), Tabella 6.14 (test di *Tukey*) e Tabella 6.13 (test di *Scheffé*).

Le tabelle che riportano i risultati dei test su ogni coppia di agenti si consultano per righe: su ogni riga è riportato il nome di un agente che viene confrontato con gli agenti che lo precedono in classifica. Il simbolo  $\simeq$  su fondo grigio indica che l'agente nella riga corrispondente ha presumibilmente prestazioni comparabili all'agente nella corrispondente colonna (l'ipotesi nulla è stata *accettata* o *non rifiutata*). Il simbolo  $<$  indica che le prestazioni dell'agente nella riga corrispondente sono molto probabilmente inferiori a quelle dell'agente nella corrispondente colonna (l'ipotesi nulla è stata *rifiutata*). I simboli su fondo grigio chiaro indicano che il test ha fornito una risposta solo *marginalmente* significativa, ovvero con  $p\text{-value}$  compreso tra 0.01 e 0.10. Inoltre queste tabelle riportano nella prima colonna il tipo di implementazione utilizzata. La maggior parte degli agenti utilizza una strategia euristica (H) o basata su un *solver* (S). Non risulta una netta prevalenza di un metodo sull'altro, anche se nel turno di selezione, ai primi posti si sono classificati più agenti basati su euristiche. Anche in questo caso però, non si può parlare

di raggruppamento di strategie. Si noti inoltre che la classificazione in base all'implementazione della strategia non è netta e di alcuni agenti non sono disponibili sufficienti informazioni.

Le tabelle che riportano i risultati dei test di omogeneità si consultano per colonne. In ogni colonna è riportato un gruppo di agenti le cui prestazioni sono presumibilmente omogenee. Maggiore è il *p-value* corrispondente e più omogeneo è il gruppo. Si noti che in generale i gruppi trovati con i test di omogeneità non sono disgiunti.

### 6.6.2 Osservazioni

A differenza del turno eliminatorio, il turno di selezione non contiene gruppi anomali. Si noti inoltre che prima di questo turno gli agenti avevano già partecipato a centinaia di sessioni e quindi ai partecipanti era stata data la possibilità di correggere la maggior parte degli errori nel codice degli agenti. Perciò i risultati dell'analisi su questo turno sono più significativi, rispetto a quelli relativi al turno preliminare, perché gli agenti sono stati modificati di meno e anche perché si sono giocate più sessioni.

La Tabella 6.9 riporta i risultati del test di *Scheffé* applicato al turno di selezione. Questo test è il più conservativo di tutti e quindi è quello che tende a rilevare meno le differenze fra gli agenti. Ciò è rappresentato dall'elevata estensione dell'area grigia. Il test di Scheffé evidenzia che gli agenti classificati dal quattordicesimo all'ultimo posto hanno prestazioni inferiori a quelle degli altri agenti. Gli agenti classificati dal primo al tredicesimo posto hanno prestazioni confrontabili, perché l'area bianca (e quella grigio chiaro che rappresenta una zona di transizione) all'interno del triangolo che include i primi tredici agenti è abbastanza piccola. Per avere dei sottogruppi più omogenei si potrebbero raggruppare gli agenti dal primo all'undicesimo posto e gli agenti dal quarto al tredicesimo posto. Le stesse considerazioni sono confermate dal test di *omogeneità* di Scheffé in Tabella 6.13. I gruppi che contengono i primi tredici agenti sono ampi, segno che le prestazioni sono piuttosto omogenee (anche se bisogna ricordare che il test è molto conservativo). Inoltre si no-

ta una separazione abbastanza netta rispetto ai gruppi che contengono gli ultimi sei agenti.

Dalla Tabella 6.10, relativa al test di *Bonferroni*, si può notare che anche questo test è conservativo anche se meno del precedente. Anche in questo caso si nota la separazione piuttosto netta tra i primi tredici e gli ultimi sei agenti. Essendo questo test meno conservativo, però, si può notare che l'area grigia è meno estesa rispetto a quella in Tabella 6.9 (test di Scheffé). Il test di Bonferroni evidenzia tre sottogruppi non disgiunti formati da agenti con prestazioni molto simili: un primo gruppo formato dagli agenti classificati dal primo all'ottavo posto, un secondo formato dagli agenti tra il quarto e l'undicesimo posto e un terzo che comprende gli agenti tra il sesto e il tredicesimo posto. Bisogna notare che la differenza di prestazioni tra questi sottogruppi è comunque molto meno netta della differenza tra i primi tredici e gli ultimi sei agenti.

Il test di *Tukey* (Tabella 6.11) è meno conservativo del test di Bonferroni. Infatti rispetto al test precedente notiamo una maggior estensione della zona di transizione (in grigio chiaro) e dell'area bianca. Tuttavia per il test di Tukey valgono all'incirca le stesse osservazioni fatte per il test di Bonferroni. Fa eccezione l'agente classificato al sesto posto che anche il test di *omogeneità* di Tukey riportato in Tabella 6.14 sposterebbe dal primo al secondo sottogruppo.

Infine in Tabella 6.12 è riportato il test di *omogeneità* SNK (*Student-Newman-Keuls*). Si tratta del test meno conservativo di tutti. Infatti i gruppi evidenziati dal test SNK sono meno ampi di quelli riportati dagli altri test. È importante notare che anche questo test evidenzia la netta separazione tra i primi tredici e gli ultimi sei agenti.

Si noti che l'agente *polimi\_bot* (settimo nel turno di selezione) appartiene al gruppo formato dai primi tredici agenti. Tutti i test riportano che il nostro agente ha prestazioni simili a quelle dei primi classificati.

## 6.7 Conclusioni

Le analisi che abbiamo effettuato suggeriscono che nessuno tra gli agenti che ha partecipato alla competizione ha prestazioni nettamente superiori rispetto a tutti gli altri. Tuttavia tutti i test, anche quelli più conservativi, hanno confermato che esiste un gruppo di agenti, di cui fa parte anche *polimi\_bot*, che hanno prestazioni decisamente migliori di quelle degli altri agenti. Questo significa che effettivamente non si può indicare un singolo vincitore della competizione. Probabilmente ciò è dovuto alla difficoltà del problema e all’alta varianza dei punteggi. Tra l’altro bisogna notare che il contributo maggiore alla varianza non deriva dalle strategie ma dalle regole della competizione e dagli errori di programmazione (elementi non controllabili dai partecipanti). Questo implica che probabilmente la TAC è troppo “rumorosa” e non è quindi efficace nell’evidenziare una strategia realmente buona nel campo delle aste elettroniche. Per eliminare l’eccessivo “rumore di fondo” nella competizione sarebbe necessario effettuare più sessioni in un ambiente più controllato, in cui gli agenti non possano essere modificati. Un altro modo di incrementare la selettività della competizione potrebbe essere quello di apportare alcune modifiche al regolamento. Tuttavia riteniamo che la TAC sia comunque un valido strumento per studiare le dinamiche di mercati complessi e dei sistemi multiagente.



S	SouthamptonTAC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	
H	whitebear		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
H	DEFend		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
S	TacsMan		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
S	Retsina		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
?	Mike		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
?	Urtaub01		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
S	ATTac		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
H	umbctac		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
H	polimi_bot		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
H	livingagents		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
H	harami		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
?	harami		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
S	PainInNEC		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
P	dummy_buyer0		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
H	CaisersSose		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
S	RedNEC		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
?	Spartacus		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
H	Typhon		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
H	LongAgent		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
?	UATrader		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
L	jboardw		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
S	RoxyBot		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
H	arc-2k		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
P	tricky		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
?	gekko		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
?	bang		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
R	006		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
G	poligen		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
Impl.	Agente	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27

Tabella 6.4: Test di Bonferroni sul turno preliminare. Il simbolo  $\simeq$  su fondo grigio indica che l'agente nella riga corrispondente ha prestazioni comparabili all'agente nella colonna corrispondente. Il simbolo  $<$  indica che le prestazioni dell'agente nella riga corrispondente sono inferiori a quelle dell'agente nella corrispondente colonna. I simboli su fondo grigio chiaro indicano che il test ha fornito una risposta solo marginalmente significativa. Nella prima colonna è riportato se l'agente è euristico (H), basato su solver (S), parametrico (P), basato sull'apprendimento automatico (L), basato su regole Prolog (R) o selezionato da algoritmi genetici (G).

Imp1.	Agente	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
S	SouthamptonTAC	1																											
H	whitebear	2	<																										
?	DEFend	3	<	<																									
S	TacsMan	4	<	<	<																								
S	Retsina	5	<	<	<	<																							
?	Mike	6	<	<	<	<	<																						
?	Urtanb01	7	<	<	<	<	<	<																					
S	ATTac	8	<	<	<	<	<	<	<																				
H	umbctac	9	<	<	<	<	<	<	<	<																			
H	polimi_bot	10	<	<	<	<	<	<	<	<	<																		
H	Ivimgagents	11	<	<	<	<	<	<	<	<	<	<																	
?	harami	12	<	<	<	<	<	<	<	<	<	<	<																
S	PainInNEC	13	<	<	<	<	<	<	<	<	<	<	<	<															
P	dummy_buyer0	14	<	<	<	<	<	<	<	<	<	<	<	<	<														
H	CaizerSose	15	<	<	<	<	<	<	<	<	<	<	<	<	<	<													
S	RedNEC	16	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<												
?	Spartacus	17	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<											
H	Typhon	18	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<										
?	LongAgent	19	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<									
?	UATrader	20	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<								
L	jboadw	21	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<							
S	RoxyBot	22	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<						
H	arc-2k	23	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<					
P	tricky	24	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<				
?	gekko	25	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<			
?	bang	26	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<		
R	006	27	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	
G	poligen	28	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	

Tabella 6.5: Test di Tukey sul turno preliminare. Il simbolo  $\simeq$  su fondo grigio indica che l'agente nella riga corrispondente ha prestazioni comparabili all'agente nella colonna corrispondente. Il simbolo  $<$  indica che le prestazioni dell'agente nella riga corrispondente sono inferiori a quelle dell'agente nella corrispondente colonna. I simboli su fondo grigio chiaro indicano che il test ha fornito una risposta solo marginalmente significativa. Nella prima colonna è riportato se l'agente è euristico (H), basato su solver (S), parametrico (P), basato sull'apprendimento automatico (L), basato su regole Prolog (R) o selezionato da algoritmi genetici (G).

#	Gruppi omogenei										SouthamptonTAC		
1													
3													
2													
5													
6													
4													
7													
8													
10													
9													
13													
12													
14													
11													
15													
16													
17													
18													
19													
20													
21													
22													
23													
24													
25													
27													
26													
28													
p-value	1.000	.245	1.000	.310	.169	.065	.083	.202	.174	.133	.062	.070	1.000

Tabella 6.6: Test di omogeneità SNK sul turno preliminare. In ogni colonna è riportato un gruppo di agenti le cui prestazioni sono presumibilmente omogenee. Maggiore è il  $p$ -value corrispondente e più omogeneo è il gruppo.

#	Gruppi omogenei									
1										
2							DEFend whitebear	SouthamptonTAC DEFend whitebear		
3							Retsina Mike	Retsina Mike		
4							TacsMan UFlaub01	TacsMan UFlaub01		
5							ATTac	ATTac		
6							polimi_bot umbctac	polimi_bot umbctac		
7							harani PainInNEC	harani PainInNEC		
8							harani dummy_buyer0	harani dummy_buyer0		
9							livingagents CaiserSose	livingagents CaiserSose		
10							RedNEC Spartacus	RedNEC Spartacus		
11							Typhon LongAgent	harani dummy_buyer0		
12							UATrader jboardw	livingagents CaiserSose		
13							arc-2k tricky	RedNEC		
14							jboardw RoxyBot	RedNEC		
15							UATrader jboardw	RedNEC		
16							RoxyBot arc-2k	Spartacus		
17							Tricky gekko	Typhon		
18							arc-2k tricky	LongAgent		
19							gekko 006	LongAgent		
20							tricky gekko	LongAgent		
21							bang 006	LongAgent		
22							bang poligen	LongAgent		
23										
24										
25										
26										
27										
28										
p-value	.342	.255	.324	.067	.070	.118	.075	.078	.661	.173

Tabella 6.7: Test di omogeneità di Scheffé sul turno preliminare. In ogni colonna è riportato un gruppo di agenti le cui prestazioni sono presumibilmente omogenee. Maggiore è il *p-value* corrispondente e più omogeneo è il gruppo.

#	Gruppi omogenei									
1										SouthamptonTAC
3										DEFend
2										whitebear
5										Retsina
6										Mike
4										TacsMan
7										Urlaub01
8										ATTac
10										polimi_bot
9										umbctac
13										PainInNEC
12										harani
14										dummy_buyer0
11										livingagents
15										CaiserSose
16										RedNEC
17										Spartacus
18										Typhon
19										LongAgent
20										UATrader
21										jboardw
22										RoxyBot
23										arc-2k
24										tricky
25										gekko
27										006
26										bang
28										poligen
p-value										1.000
										.070
										.350
										.824
										.214
										.364
										.350
										.058
										.110
										.063
										.098
										.056

Tabella 6.8: Test di omogeneità di Tukey sul turno preliminare. In ogni colonna è riportato un gruppo di agenti le cui prestazioni sono presumibilmente omogenee. Maggiore è il *p-value* corrispondente e più omogeneo è il gruppo.

S	SouthamptonTAC	1																		
H	whitebear	2	2																	
?	Urlaub01	3	2	2																
H	livingagents	4	2	2	2															
S	TacsMan	5	2	2	2	2														
H	CaiserSose	6	2	2	2	2	2													
H	polimi_bot	7	2	2	2	2	2	2												
H	umbctac	8	2	2	2	2	2	2	2											
S	RoxyBot	9	2	2	2	2	2	2	2	2										
S	ATTac	10	2	2	2	2	2	2	2	2	2									
S	Retsina	11	2	2	2	2	2	2	2	2	2	2								
S	PainInNEC	12	2	2	2	2	2	2	2	2	2	2	2							
S	ATTac2000	13	2	2	2	2	2	2	2	2	2	2	2	2						
?	harani	14	2	2	2	2	2	2	2	2	2	2	2	2	2					
P	dummy_buyer0	15	2	2	2	2	2	2	2	2	2	2	2	2	2	2				
L	jboadw	16	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2			
R	006	17	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2		
?	bang	18	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	
H	arc-2k	19	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Impl.	Agente	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

Tabella 6.9: Test di Scheffé sul turno di selezione. Il simbolo  $\simeq$  su fondo grigio indica che l'agente nella riga corrispondente ha prestazioni comparabili all'agente nella colonna corrispondente. Il simbolo  $<$  indica che le prestazioni dell'agente nella riga corrispondente sono inferiori a quelle dell'agente nella corrispondente colonna. I simboli su fondo grigio chiaro indicano che il test ha fornito una risposta solo marginalmente significativa. Nella prima colonna è riportato se l'agente è euristico (H), basato su solver (S), parametrico (P), basato sull'apprendimento automatico (L), basato su regole Prolog (R) o selezionato da algoritmi genetici (G).

S	SouthamptonTAC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
H	whitebear		∩																	
?	Urlaub01		∩	∩																
H	livingagents		∩	∩	∩															
S	TacsMan		∩	∩	∩	∩														
H	CaiserSose		∩	∩	∩	∩	∩													
H	polimi_bot		∩	∩	∩	∩	∩	∩												
H	umbctac		∩	∩	∩	∩	∩	∩	∩											
S	RoxyBot		∩	∩	∩	∩	∩	∩	∩	∩										
S	ATTac		∩	∩	∩	∩	∩	∩	∩	∩	∩									
S	Retsina		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩								
S	PainInNEC		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩							
S	ATTac2000		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩						
?	harami		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩						
P	dummy_buyer0		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩					
L	jboardw		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩				
R	006		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩			
?	bang		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩		
H	arc-2k		∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	
Impl.	Agente	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

Tabella 6.10: Test di Bonferroni sul turno di selezione. Il simbolo  $\simeq$  su fondo grigio indica che l'agente nella riga corrispondente ha prestazioni comparabili all'agente nella colonna corrispondente. Il simbolo  $<$  indica che le prestazioni dell'agente nella riga corrispondente sono inferiori a quelle dell'agente nella corrispondente colonna. I simboli su fondo grigio chiaro indicano che il test ha fornito una risposta solo marginalmente significativa. Nella prima colonna è riportato se l'agente è euristico (H), basato su solver (S), parametrico (P), basato sull'apprendimento automatico (L), basato su regole Prolog (R) o selezionato da algoritmi genetici (G).

S	SouthamptonTAC	1																		
H	whitebear	2	≲																	
?	Urlaub01	3	≲	≲																
H	livingagents	4	≲	≲	≲															
S	TacsMan	5	≲	≲	≲	≲														
H	CaiserSose	6	≲	≲	≲	≲	≲													
H	polimi_bot	7	≲	≲	≲	≲	≲	≲												
H	umbctac	8	≲	≲	≲	≲	≲	≲	≲											
S	RoxyBot	9	≲	≲	≲	≲	≲	≲	≲	≲										
S	ATTac	10	≲	≲	≲	≲	≲	≲	≲	≲	≲									
S	Retsina	11	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲								
S	PainInNEC	12	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲							
S	ATTac2000	13	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲						
?	harami	14	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲						
P	dummy_buyer0	15	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲					
L	jboardw	16	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲				
R	006	17	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲			
?	bang	18	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲		
H	arc-2k	19	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	≲	
Impl.	Agente	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

Tabella 6.11: Test di Tukey sul turno di selezione. Il simbolo  $\simeq$  su fondo grigio indica che l'agente nella riga corrispondente ha prestazioni comparabili all'agente nella colonna corrispondente. Il simbolo  $<$  indica che le prestazioni dell'agente nella riga corrispondente sono inferiori a quelle dell'agente nella corrispondente colonna. I simboli su fondo grigio chiaro indicano che il test ha fornito una risposta solo marginalmente significativa. Nella prima colonna è riportato se l'agente è euristico (H), basato su solver (S), parametrico (P), basato sull'apprendimento automatico (L), basato su regole Prolog (R) o selezionato da algoritmi genetici (G).



#	Gruppi omogenei									
1										
2										
3										
4										
5										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
18										
17										
19	arc-2k									
p-value	1.000	.997	.299	.760	.417	.405	.051	.224	.142	.086

Tabella 6.13: Test di omogeneità di Scheffé sul turno di selezione. In ogni colonna è riportato un gruppo di agenti le cui prestazioni sono presumibilmente omogenee. Maggiore è il *p-value* corrispondente e più omogeneo è il gruppo.

#	Gruppi omogenei											
1										SouthamptonTAC		
2										whitebear		
3										Urlaub01		
4										livingagents		
5										TacsMan		
7										polimi_bot		
8										umbctac		
6										CaiserSose		
9										RoxyBot		
10										ATTac		
11										Retsina		
12										PainInNEC		
13										PainInNEC		
14										ATTac2000		
15										harami		
16										dummy_buyer0		
18										jboardw		
17										bang		
19										arc-2k		
p-value	1.000	.646	1.000	1.000	1.000	1.000	.133	.112	.063	.082	.055	.055

Tabella 6.14: Test di omogeneità di Tukey sul turno di selezione. In ogni colonna è riportato un gruppo di agenti le cui prestazioni sono presumibilmente omogenee. Maggiore è il *p-value* corrispondente e più omogeneo è il gruppo.



# Capitolo 7

## Conclusioni

In questa tesi abbiamo studiato alcune problematiche relative alla progettazione e alla realizzazione di agenti autonomi per aste elettroniche. In particolare abbiamo presentato l'agente *polimi\_bot*, un agente software autonomo che abbiamo sviluppato per partecipare alla competizione internazionale nota come *Trading Agent Competition 2001* (TAC-01). In questa competizione, ogni partecipante è un agente di viaggio (software) che ha come obiettivo l'allestimento di pacchetti di viaggio. Ogni agente agisce per conto di otto clienti che esprimono le loro preferenze riguardo a vari aspetti del viaggio (il volo, il soggiorno e l'intrattenimento). Lo scopo dell'agente è massimizzare la soddisfazione dei suoi clienti, cercando di minimizzare le spese. I partecipanti propongono agenti software per allestire in modo automatico pacchetti di viaggio, interagendo in tre mercati simulati. Ogni mercato vende un tipo di bene: biglietti aerei, prenotazioni per stanze d'albergo e biglietti per assistere ad alcune attrazioni. Per rendere le cose più interessanti, ogni categoria di beni viene scambiata in aste differenti con regole diverse. Una difficoltà ulteriore deriva dall'interdipendenza dei beni: ad esempio la prenotazione delle stanze d'albergo deve corrispondere alle date di arrivo e partenza dei voli di cui si possiedono i biglietti. Alla fine della sessione di gioco, gli agenti ricevono un punteggio che dipende da quanto sono riusciti a rispettare le preferenze di ogni cliente e da quanto hanno speso per i beni. Lo scopo

dell'agente è massimizzare il guadagno che dipende dal livello di soddisfazione degli otto clienti. Per la realizzazione dell'agente *polimi\_bot* abbiamo scelto di utilizzare una strategia euristica. In questo modo, abbiamo voluto privilegiare la semplicità di implementazione, mantenendo comunque un certo grado di flessibilità nei comportamenti, attraverso l'utilizzo di parametri. L'implementazione della strategia è stata realizzata definendo un'opportuna architettura basata su regole che definiscono il comportamento dell'agente in risposta agli eventi che si verificano durante la competizione. Per velocizzare i tempi di realizzazione e per poter verificare la strategia su cui si basa il nostro agente, abbiamo deciso di implementare le regole direttamente in C++, rendendo l'agente meno flessibile ma facilitando il compito di trovare e correggere gli errori di implementazione. Durante la competizione il nostro agente *polimi\_bot* ha superato la fase eliminatoria della gara risultando decimo nel turno di qualificazione e settimo nel successivo turno di selezione. È stato quindi ammesso alla fase finale della competizione che si è tenuta a Tampa, Florida USA, e si è classificato solamente sesto (su otto) nel suo girone delle semifinali, per un problema tecnico. L'agente *polimi\_bot* non ha infatti potuto partecipare alla prima delle undici sessioni di gioco delle semifinali per un problema di collegamento alla rete. Non è quindi riuscito ad accedere alla finale a cui erano ammessi solo i primi quattro agenti dei due gironi delle semifinali. Tuttavia da quello che è emerso dall'analisi statistica dei risultati della competizione, si può affermare che il nostro agente ha prestazioni comparabili a quelle dei primi classificati. Infatti in tutti i test effettuati, anche in quelli più inclini a rilevare differenze, si osserva che l'agente *polimi\_bot* appartiene al gruppo formato dai migliori agenti. Inoltre l'analisi dei risultati dell'intera competizione, effettuata tramite metodi statistici, ci ha permesso di osservare che il tipo di competizioni, di cui fa parte la TAC, probabilmente non è adatto a evidenziare una singola strategia realmente efficace nel campo delle aste elettroniche. Questo tipo di competizioni non ha quindi una granularità sufficientemente fine per poter isolare un solo agente. Tuttavia pensiamo che la TAC rimanga comunque un valido strumento per

studiare le dinamiche di mercati complessi e dei sistemi multiagente. Inoltre riteniamo che sia possibile incrementare la selettività della competizione apportando alcune modifiche al regolamento.



# Appendice A

## I parametri dell'agente *polimi\_bot*

L'agente *polimi\_bot* è parametrico: in questa Appendice verranno descritti i parametri che ne regolano il funzionamento.

```
flightStartBiddingTime = 0
flightSecondBiddingTime = 360
flightBid = 800.0
hotelEstTrigger = 100.0
hotelIncrement = 125.0
hotelLimitThres = 300.0
htSwitchValue = 70.0
htChoiceValue = 30.0
minEntertLimit = 40.0
maxEntertLimit = 160.0
entertGain = 0.35
beta = 0.3
momentum = 0.05
lastIterations = 5
startIterations = 3
```

I primi tre parametri sono relativi al mercato dei biglietti aerei. Il parametro *flightStartBiddingTime* indica quanti secondi l'agente deve inizialmente aspettare prima di acquistare il primo blocco di biglietti aerei. Il parametro *flightSecondBiddingTime* specifica il numero di secondi che l'agente deve attendere prima di comprare i biglietti rimanenti. Il parametro *flightBid* indica a che prezzo l'agente effettua le offerte per i biglietti aerei.

I successivi cinque parametri sono relativi al mercato delle stanze d'albergo. Il parametro *hotelEstTrigger* indica la quotazione del prezzo delle stanze d'albergo sotto la quale l'agente utilizza il predittore a regressione lineare. Sopra la quotazione indicata dal parametro *hotelEstTrigger* l'agente utilizza invece il predittore basato sui Processi di Poisson. I parametri *htChoiceValue* e *htSwitchValue* forniscono rispettivamente la base e la pendenza della retta secondo cui sale la soglia che determina se il cambio di hotel è conveniente.

I successivi cinque parametri sono relativi al mercato dei biglietti per le attrazioni. I parametri *minEntertLimit* e *maxEntertLimit* indicano i prezzi limite che l'agente è disposto a spendere per i biglietti. Il parametro *entertGain* indica il guadagno che l'agente vuole ottenere sui suoi prezzi limite. I parametri *beta* e *momentum* sono relativi alla regola Widrow-Hoff di aggiornamento dei margini di profitto per gli agenti ZIP.

Gli ultimi due parametri regolano il comportamento dell'agente in base all'istante di tempo della sessione in cui si trova. Il parametro *lastIterations* indica quando l'agente si trova in prossimità della fine della sessione di gioco, ovvero quando il numero stimato di iterazioni restanti è inferiore al valore di questo parametro. Infine il parametro *startIterations* specifica quante iterazioni l'agente deve attendere prima di iniziare a inviare le offerte al *server*.

# Bibliografia

- [1] *J. Boyan, A. Greenwald, R. M. Kirby, J. Reiter*, “**Bid Determination in Simultaneous Auctions**”. In *Proceedings of Third ACM Conference on Electronic Commerce*, Ottobre 2001.
- [2] *P. Stone, A. Greenwald*, “**The First International Trading Agent Competition: Autonomous Bidding Agents**”. In *Stampa*, 2001
- [3] *W. Vickrey*, “**Counterspeculation, auctions and competitive sealed tenders**”. In *Journal of Finance*, Marzo 1961.
- [4] *P. Milgrom*, “**Auctions and Bidding: A Primer**”. In *Journal of Economic Perspectives*, Vol. 3 No. 3, 1989, pagg. 3-22.
- [5] *H. R. Varian*, “**Economic Mechanism Design for Computerized Agents**”. In *First USENIX Workshop on Electronic Commerce*, Luglio 11-12 1995, pagg. 13-21.
- [6] *Friedman*, “**The double auction market institution**”. In *Friedman and Rust*, 1992
- [7] *Smith, Williams*, “**An Experimental Study of Alternative Rules for Competitive Market Exchange**”. In *Auctions, Bidding and Contracting: Uses and Theory*, New York University Press, 1983
- [8] *M. H. Rothkopf, A. Pekeč, R. M. Harstad*, “**Computationally manageable combinatorial auctions**”. In *Management Science*, 44(8), 1998

- [9] *T. Sandholm*, “**An Algorithm for Optimal Winner Determination in Combinatorial Auctions**”. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI 99*, pagg. 542-547, Agosto 1999
- [10] *R. Korf*, “**Depth-first iterative-deepening: An optimal admissible tree search**”. In *Artificial Intelligence*, 27(1):97-109
- [11] *T. Sandholm, S. Suri*, “**Improved Algorithms for Optimal Winner Determination in Combinatorial Auctions and Generalizations**”. In *Proceedings of AAAI*, pagg. 90-97, 2000
- [12] *R. Gonen, D. Lehmann*, “**Optimal Solutions for Multi-Unit Combinatorial Auctions: Branch and Bound Heuristics**”. Maggio 2000
- [13] *F. Ygge*, “**Improving the Computational Efficiency of Combinatorial Auction Algorithms**”. *EnerSearch Public Report*, EnS 99:08 v1.2/991007
- [14] *P. Stone, M. L. Littman, S. Singh, M. Kearns*, “**ATTac-2000: An Adaptive Autonomous Bidding Agent**”. In *Fifth International Conference on Autonomous Agents*, 2001
- [15] *D. K. Gode, S. Sunder*, “**Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality**”. *J. Polit. Econ.*, 101(1), 119-137, 1993
- [16] *D. Cliff, J. Bruten*, “**More Than Zero Intelligence Needed for Continuous Double-Auction Trading**”. *HP Technical Report*, HPL-97-157, 1997
- [17] *AA. VV.*, “**Guida al commercio elettronico**”. <http://www.artigianato.marche.it/gecomm.htm>, 2000

- [18] AA. VV., “**PMInternet: un progetto per il commercio elettronico**”. *Iniziativa promossa da Anasin, Assinform, Confindustria*, <http://www.e-commercepmi.it>, 2001
- [19] AA. VV., “**Business to Business E-commerce**”. *Web report submitted as partial fulfilment of the requirements of Business 8205 Memorial University of Newfoundland*, 1998
- [20] W. T. Lasley, “**E-Commerce - Then and Now**”. <http://artsandcrafts.about.com/library/weekly/aa041201.htm>, 2001
- [21] AA. VV., “**Guida al commercio elettronico**”. *Forum Italiano sul Commercio Elettronico con il Patrocinio del Ministero dell’Industria*, [http://sole.ilsole24ore.com/24oreinformatica/biblioteca/e\\_commerce/guidcomm.pdf](http://sole.ilsole24ore.com/24oreinformatica/biblioteca/e_commerce/guidcomm.pdf), 1999
- [22] J. Rifkin, “**L’era dell’accesso. La rivoluzione della new economy**”. Mondadori, Milano, 2000
- [23] R. Pianola, “**Dizionario della New Economy**”. Baldini e Castoldi, 2000
- [24] G. Sciulli, “**Commercio elettronico: modelli B2B e B2C**”. [http://www.html.it/dossier/07\\_ecommerce/03.htm](http://www.html.it/dossier/07_ecommerce/03.htm), 2000
- [25] C. Priest, “**Algorithm Design for Agents Which Participate in Multiple Simultaneous Auctions**”. *HP Technical Report*, HPL-2000-88, 2000
- [26] S. Kessler, “**The E-Commerce Winners**”. In *Business Week*, 03-08-2000
- [27] P. R. Wurman, M. P. Wellman, W. E. Walsh, “**The Michigan Internet AuctionBot: A configurable auction server for human and software agents**”. In *Proceedings of Second International Conference on Autonomous Agents*, pagg. 301-308, Minneapolis, 1998

- [28] *B. Strother*, “**TAC: A Trading Agent Competition**”. In *SIGecom Exchanges*, Agosto 2000
- [29] *C. Boutilier, H. H. Hoos*, “**Solving Combinatorial Auctions using Stochastic Local Search** ”. In *Proceedings of AAAI-2000*.
- [30] *M. Tsvetovatyy, M. Gini, B. Mobasher, Z. Wieckowski*, “**MAGMA: An Agent-Based Virtual Market for Electronic Commerce**”. In *Journal of Applied Artificial Intelligence*, (6), 1997
- [31] *A. Chavez, P. Maes*, “**Kasbah: An agent marketplace for buying and selling goods**”. In *Proceedings of First International Conference on the Pratical Application of the Intelligent Agents and Multi-Agent Technology*, Londra, 1996
- [32] *E. H. Durfee, D. L. Kiskis, W. P. Birmingham*, “**The Agent Architecture of the University of Michigan Digital Library**”. *IEE British Computer Society Proceedings on Software Engineering*, vol 144, 1997
- [33] *R. P. McAfee, J. McMillan*, “**Auctions and bidding**”. In *Journal of Economic Literature*, 25:699-738, 1987
- [34] *R. Pike, D. Presotto, S. Dorward, B. Flandrena, K. Thompson, H. Trickey, P. Winterbottom*, “**Plan 9 from Bell Labs**”. In *Computing Systems*, Vol 8 #3, pagg. 221-254, 1995
- [35] *D. Cliff, J. Bruten*, “**Zero is not enough: On the lower limit of agent intelligence for continuous double auction markets**”. *HP Technical Report*, HPL-97-141, 1997
- [36] *P. W. Jones, P. Smith*, “**Stochastic Processes: An Introduction**”. Arnold Publishers <http://www.arnoldpublishers.com>, 2001
- [37] *B. Widrow, M. Hoff*, “**Adaptive switching circuits**”. In *IRE WESTCON Connection Record*, volume 4, pagg. 96-104, 1960

- [38] *R. M. Neal*, “**Probabilistic Inference Using Markov Chain Monte Carlo Methods**”. *Technical Report*, CRG-TR-93-1, University of Toronto, Dept. of Computer Science, 1993
- [39] *R. H. J. M. Otten, L. P. P. P. Ginneken*, “**The Annealing Algorithm**”. Kluwer Academic, Boston, MA, 1989
- [40] *Y. Azar, A. Z. Broder, A. R. Karlin, N. Linial, S. Phillips*, “**Biased Random Walks**”. In *ACM Symposium on Theory of Computing*, 1992
- [41] *S. A. Glantz, B. K. Slinker* “**Primer of Applied Regression & Analysis of Variance**”. (Seconda edizione), McGraw Hill, 2001
- [42] *W. J. Conover, M. E. Johnson, M. M. Johnson* “**A Comparative Study of Tests for Homogeneity of Variance, with Applications to the Outer Continental Shelf Bidding Data**”. In *Technometrics* 23:351-361, 1981
- [43] *J. H. Zar* “**Biostatistical Analysis**”. (Seconda edizione), Englewood Cliffs, NJ, Prentice-Hall, 1984
- [44] AA. VV., “**Strategy of livingagents during TAC**”. Ottobre 2001
- [45] *P. Stone*, “**ATTac-2001: A Machine Learning, Data-Driven Approach**”. Presentato durante *TAC Workshop*, 14 Ottobre 2001
- [46] *I. A. Vetsikas*, “**White Bear: An Adaptive Agent**”. Presentato durante *TAC Workshop*, 14 Ottobre 2001
- [47] *K. Sycara, Y. Nevmyvaka, Pu Huang, G. Economou*, “**Retsina: Intelligent Trading Agent**”. Presentato durante *TAC Workshop*, 14 Ottobre 2001
- [48] *Minghua He, N. Jennings*, “**SouthamptonTAC**”. Presentato durante *TAC Workshop*, 14 Ottobre 2001

- 
- [49] *M. Fasli, I. Korres, G. Rallidis, A. Zakas*, “**Trading Agent Competition 2001: The Essex Team(s)**”. Presentato durante *TAC Workshop*, 14 Ottobre 2001
- [50] *R. Porter, E. Nudelman, H. Chen, Y. Shoham, R. Powers*, “**TACSMAN**”. Presentato durante *TAC Workshop*, 14 Ottobre 2001
- [51] *T. Mullen, S. Park, E. Glover, D. Pennock*, “**PainInNEC**”. Presentato durante *TAC Workshop*, 14 Ottobre 2001
- [52] *Youyong Zou, Shanshan Liu*, “**UMBCTAC**”. Presentato durante *TAC Workshop*, 14 Ottobre 2001
- [53] *A. Greenwald, J. Boyan*, “**RoxyBot: An Approximately Optimal Agent**”. Presentato durante *TAC Workshop*, 14 Ottobre 2001
- [54] *D. Reeves*, “**Parameterized Dummy Agent**”. Presentato durante *TAC Workshop*, 14 Ottobre 2001
- [55] **Amazon.com**, <http://www.amazon.com>
- [56] **eBay**, <http://www.ebay.com>
- [57] **Auction Beagle**, <http://www.auctionbeagle.com>
- [58] **FreeMarkets**, <http://www.freemarket.com/>
- [59] **Agent-Based Market Space**, <http://www.sics.se/~market/>
- [60] **The Michigan Internet AuctionBot**, <http://auction2.eecs.umich.edu/auction/>
- [61] **FishMarket**, <http://www.iiia.csic.es/Projects/fishmarket/newindex.html>
- [62] **Bazaar**, <http://www.media.mit.edu/~guttman/research/bazaar/bazaar.html>

- [63] **Agent-mediated Electronic Commerce**, <http://ecommerce.media.mit.edu/>