POLITECNICO DI MILANO

Facoltà di Ingegneria dell'Informazione

Corso di Laurea Specialistica in Ingegneria Informatica

Dipartimento di Elettronica e Informazione

Giochi stocastici polinomiali a somma zero con Switching Control

Relatore: Prof. Nicola GATTI

Tesi di Laurea di:
Guido BONOMI
Matricola n. 739755

Anno Accademico 2010-2011

Sommario

La Teoria dei Giochi è un campo della matematica applicata che studia il comportamento strategico di diversi decisori quando si trovano a dover interagire tra loro. Nonostante la ricerca sulla Teoria dei Giochi si sia orientata principalmente sullo studio di giochi finiti, in cui ogni giocatore ha a disposizione un numero finito di strategie pure, grazie ai molteplici e differenti ambiti di applicazione risultano di notevole interesse anche i giochi infiniti, in cui ogni giocatore ha accesso ad un numero infinito di azioni.

In questa tesi vengono studiate alcune classi di giochi infiniti a due giocatori e a somma zero, appartenenti alla classe dei giochi polinomiali, in cui le azioni dei giocatori sono numeri reali e la funzione di payoff è polinomiale nelle azioni dei giocatori. In particolare, oltre ai giochi polinomiali in forma normale viene studiata la classe dei giochi polinomiali stocastici con Switching Control, una classe di giochi infiniti che si sviluppano su grafo, in cui ad ogni stato viene associato un unico giocatore da cui dipendono le probabilità di transizione. Per questa classe di giochi viene presentato un algoritmo in grado di calcolare un ϵ-equilibrio risolvendo iterativamente dei problemi di programmazione semidefinita positiva, discutendone le proprietà di convergenza e valutandone sperimentalmente le prestazioni.

Indice

Elenco delle figure

Capitolo 1

Introduzione

La Teoria dei Giochi è un campo relativamente recente della matematica applicata che analizza i comportamenti strategici di vari decisori, chiamati *giocatori*, quando si trovano a dover interagire tra loro per perseguire obiettivi comuni, diversi o conflittuali. La nascita della Teoria dei Giochi risale al 1928, quando John Von Neumann presentò le basi del Teorema Minimax, ma fu anche grazie ai contributi di John Nash ed Oskar Morgenstern negli anni '40 che si crearono le basi solide di questa disciplina. Nata come applicazione in ambito economico, ben presto venne utilizzata in svariati campi, dalla politica all'informatica, dalla finanza alla psicologica, dalla sociologia all'ambito strategico-militare.

Nonostante molta della ricerca nell'ambito della Teoria dei Giochi si sia concentrata principalmente sul caso dei giochi *finiti*, in cui ai decisori viene proposto un numero limitato di alternative tra cui scegliere, grazie ai moltissimi campi di applicazione è risultato fin da subito di notevole interesse anche il caso dei giochi *infiniti*, in cui ai decisori è offerta una gamma di infinite alternative tra cui scegliere. Un primo approccio a questo caso risale infatti agli anni '50, quando Dresher [5] introdusse i *giochi polinomiali*, in cui i decisori devono scegliere dei numeri reali all'intero di un intervallo. I giochi polinomiali generarono subito grande interesse, come possibile collegamento tra i giochi finiti ed infiniti. Tuttavia, nel 1959 Karlin [13] scrisse:

> *"Ben presto si realizzò che i giochi polinomiali con gradi alti possiedono soluzioni di grande complessità, impossibili da descrivere in termini qualitativi, e a maggior ragione da calcolare."*

Nonostante ciò, i notevoli progressi nella teoria dell'ottimizzazione degli

ultimi anni possono essere ben utilizzati per aggiornare in modo significativo questa valutazione, offrendo quindi l'opportunità di "riconsiderare" la classe dei giochi polinomiali e dei giochi infiniti. Scopo di questa tesi è proprio quello di studiare la classe dei giochi polinomiali, ed in particolare dei *giochi polinomiali stocastici*. In tale classe, il gioco si sviluppa in vari stati, ed il movimento da uno stato ad un altro dipende dalle decisioni dei giocatori. In questa tesi viene riservata particolare attenzione ai giochi stocastici polinomiali in cui ad ogni stato è associato un unico giocatore da cui dipende il movimento verso gli altri stati, con l'obiettivo di proporre un algoritmo in grado di trovare, per ogni stato, le migliori decisioni dei giocatori rispetto ai loro scopi.

1.1 Descrizione del Lavoro

I giochi polinomiali sono una particolare classe dei giochi infiniti in cui le azioni corrispondono a numeri reali, mentre la funzione di payoff è un'espressione polinomiale nelle azioni dei giocatori. La struttura ed il calcolo degli equilibri nei giochi con uno spazio di strategie infinito sono da lungo tempo riconosciuti come complessi [40]. Nel 1950, Dresher [5] mostrò la caratterizzazione e l'esistenza di strategie ottime per i giochi polinomiali, ma la mancanza di metodi computazionali efficienti arrestò la ricerca su questa classe di giochi. In effetti, c'erano buone ragioni per la mancanza di soluzioni soddisfacenti a tale problema: fino allo sviluppo della programmazione semidefinita e della connessione con le tecniche di somma di quadrati, anche la semplice minimizzazione di un polinomio univariato non era trattabile con i metodi convessi. Fintanto che la soluzione di un gioco non poteva essere più semplice dell'ottimizzazione di un polinomio (basti considerare, per esempio, un gioco che non dipende dalle azioni di uno dei due giocatori), questa intrattabilità bloccava effettivamente la nascita di metodi efficienti per il calcolo della soluzione.

I notevoli progressi nel campo della teoria dell'ottimizzazione degli ultimi anni hanno però fatto riaffiorare l'interesse sui giochi polinomiali. Nel 2006 infatti, Parrilo [25] descrisse un algoritmo per risolvere i giochi polinomiali (a due giocatori e somma zero) in forma normale attraverso tecniche di ottimizzazione in somma di quadrati e di programmazione semidefinita. In particolare, egli mostrò come caratterizzare e calcolare la soluzione ottima di questa classe di giochi, ovvero il valore del gioco e la strategia di uno dei due giocatori, ri-

solvendo un unico problema di programmazione semidefinita, ed ottenendo la strategia del giocatore avversario risolvendo un secondo problema di programmazione semidefinita. L'anno successivo, Shah e Parrilo [35] presentarono la classe dei giochi polinomiali stocastici a due giocatori e somma zero con *Single Controller*, estendendo il concetto di gioco polinomiale ai giochi sviluppati su grafo, in cui le transizioni da uno stato ad un altro sono indipendenti dalle azioni di uno dei due giocatori. In particolare, Shah e Parrilo discussero i temi della caratterizzazione, dell'esistenza e dell'unicità della soluzione nella classe dei giochi stocastici polinomiali, dimostrando che in tali giochi esiste sempre una soluzione di equilibrio e presentando un algoritmo in grado di calcolarla nel caso di transizioni dipendenti dall'azione di un unico giocatore. Questi risultati hanno quindi posto una possibile base per lo studio di varie classi di giochi polinomiali, oltre che per lo sviluppo di algoritmi in grado di risolverle.

Lo scopo di questa tesi è quello di studiare le tecniche di risoluzione dei giochi polinomiali, ed in particolare di offrire un metodo di risoluzione per una classe di giochi polinomiali stocastici, ovvero i giochi polinomiali stocastici a due giocatori e somma zero con *Switching Control*, di cui i giochi polinomiali stocastici con Single Controller sono una sottoclasse. In ogni stato di un gioco polinomiale stocastico con Switching Control le transizioni dipendono dalle azioni di un unico giocatore, ma questo giocatore può cambiare da stato a stato. Vengono quindi discusse le problematiche nel calcolo della soluzione di un gioco di questa classe, e viene proposto un algoritmo in grado di calcolare un profilo di strategie dei giocatori "vicino" alla soluzione ottima, ovvero un ϵ-equilibrio, valutandone anche sperimentalmente le prestazioni. In particolare, nello studio dei giochi stocastici polinomiali viene discusso anche il caso dei *processi di decisione di Markov polinomiali*, una particolare sottoclasse dei giochi stocastici in cui si ha un unico giocatore, chiamato *agente*, ed in cui le probabilità di transizione e la funzione di payoff sono polinomiali rispetto alle azioni continue dell'agente: viene presentato un algoritmo in grado di calcolare la politica ottima dell'agente risolvendo un unico problema di programmazione semidefinita.

Per ognuna delle tecniche di risoluzione presentate vengono anche proposti degli esempi sperimentali, in modo da mostrare l'utilizzo dei metodi di risoluzione discussi.

1.2 Organizzazione della Tesi

La tesi è organizzata nel modo seguente.

Nel Capitolo 2 vengono introdotte le basi teoriche dei giochi non cooperativi a due giocatori. Vengono descritti i giochi in forma normale, di cui vengono presentati diversi concetti di soluzione. Vengono poi presentati i giochi in forma estesa, in cui il gioco si sviluppa in un albero, ed i giochi in forma stocastica, in cui il gioco si sviluppa in un grafo, discutendo l'esistenza ed il calcolo degli equilibri in tali giochi.

Nel Capitolo 3 vengono presentati i giochi polinomiali in forma normale, discutendo l'esistenza e l'unicità delle loro soluzioni. In questo capitolo viene descritta tutta la teoria che collega l'ottimizzazione polinomiale alla programmazione semidefinita, mostrando come sia possibile risolvere un unico problema di programmazione semidefinita per ottenere la soluzione del gioco.

Nel Capitolo 4 vengono presentati i giochi polinomiali stocastici con Single Controller, estendendo il concetto di gioco polinomiale ai giochi su grafo in cui le transizioni da stato a stato sono indipendenti dalle azioni di un giocatore. Viene inoltre presentato un algoritmo in grado di calcolarne le soluzioni attraverso un unico problema di programmazione semidefinita.

A partire dal Capitolo 5 vengono proposti i principali contributi originali della tesi: vengono presentati i processi di decisione di Markov polinomiali, in cui la funzione di payoff e le probabilità di transizione sono polinomiali nelle azioni continue dell'agente, mostrando il loro legame con le migliori risposte nei giochi stocastici, e presentando un algoritmo in grado di calcolarne la politica ottima risolvendo un unico problema di programmazione semidefinita.

Nel Capitolo 6 vengono presentati i giochi polinomiali stocastici con Switching Control, un'estensione del caso con Single Controller al caso in cui il giocatore che governa le transizioni cambia da stato a stato. Viene inoltre presentato un algoritmo iterativo in grado di calcolarne le soluzioni attraverso dei problemi di programmazione semidefinita positiva.

Nel Capitolo 7 viene presentata una valutazione sperimentale delle prestazioni dell'algoritmo, applicandolo ad alcuni casi di studio ed analizzandone le prestazioni all'aumentare della dimensione del gioco.

Nel Capitolo 8, infine, viene fatto un breve bilancio del lavoro svolto, discutendo alcune possibili direzioni di ricerca futura.

Capitolo 2

Stato dell'Arte

2.1 Introduzione

L'obiettivo di questo capitolo é di richiamare le nozioni di base della Teoria del Gioco Non Cooperativa. Nella Sezione 2.2 vengono introdotti i giochi in forma normale, mentre nella Sezione 2.3 vengono presentati i principali concetti di soluzione per i giochi in tale forma. Nella Sezione 2.4 vengono introdotti i giochi in forma estesa. Infine, nella Sezione 2.5 vengono presentati i giochi stocastici, e ne viene introdotta una particolare proprietá.

2.2 Giochi in Forma Normale

L'approccio dominante per modellizzare l'interesse di un agente nella Teoria dei Giochi è la *Teoria dell'Utilità*. Questo approccio teoretico porta a quantificare il grado di preferenza di un agente su un insieme di alternative disponibili. La relazione di preferenza di ciascun giocatore sull'insieme delle possibili alternative può essere espressa attraverso una *funzione di utilità*, associata a ciascun giocatore, che consiste in un mappaggio dagli stati del mondo ai numeri reali che fa corrispondere valori più elevati ai risultati più graditi. (Una definizione formale di preferenze, della funzione di utilità e delle loro proprietà si può trovare in [37]).

La *forma normale*, anche conosciuta come *forma strategica*, è la rappresentazione più familiare delle interazioni strategiche nella Teoria dei Giochi: un gioco scritto in questa forma consiste in una rappresentazione dell'utilità di

ogni giocatore per ogni stato del mondo, nel caso speciale in cui gli stati del mondo dipendono solo dalle azioni combinate dei giocatori, che vengono eseguite simultaneamente. Nonostante possa sembrare una situazione piuttosto particolare, può essere dimostrato che i casi in cui gli stati del mondo dipendono anche dalla casualità dell'ambiente, chiamati giochi *Bayesiani*, possono essere ridotti alla forma normale. Grazie all'esistenza di conversioni in forma normale di molte delle rappresentazioni di interesse dei giochi, come i giochi in forma estesa trattati nella Sezione 2.4 che contengono elementi temporali, la forma normale è considerata la rappresentazione canonica di un gioco.

Definizione 1 (**Gioco in forma normale).** Un gioco (finito, ad n-giocatori) in normale forma è una tupla (N, A, μ), dove:

- N è un insieme finito di n giocatori, indicizzato da i;
- $A = A_1 \times \ ... \ \times A_n$, dove A_i è un insieme finito di *azioni* disponibili al giocatore i; ogni vettore $a = (a_1, ..., a_n) \in A$ è chiamato *profilo di azioni*, o *outcome*;
- $\mu = (\mu_1, ..., \mu_n)$ dove $\mu_i : A \to \mathbb{R}$ è una funzione di utilità (o *payoff*) a valori reali per il giocatore i

Un modo naturale per rappresentare i giochi in forma normale è attraverso una matrice n-dimensionale. In generale, ogni riga corrisponde ad una possibile azione per il giocatore 1, mentre ogni colonna corrisponde ad una possibile azione per il giocatore 2, ed ogni cella corrisponde ad un possibile outcome. L'utilità di ogni agente per un certo outcome è scritta nella cella corrispondente a tale outcome, elencando le utilità partendo dal giocatore 1.

In questa tesi verranno considerati i giochi a due giocatori, ovvero giochi con $n = 2$. Un gioco in forma normale a due giocatori è anche chiamato *gioco bimatriciale*, in quanto le utilità sono solitamente date da due matrici.

Definizione 2 (**Gioco Bimatriciale in forma normale).** Un gioco bimatriciale in forma normale è una tupla $(A, B) \in (\mathbb{R}, \mathbb{R})^{n \times m}$, dove le n righe sono le azioni per il primo giocatore, detto *giocatore riga*, mentre le m colonne sono le azioni per il secondo giocatore, detto *giocatore colonna*.

L'esempio in Figura 2.1 rappresenta il gioco *Sasso, Carta e Forbice*. È possibile notare che se ad esempio il giocatore riga gioca *Roccia* ed il giocatore colonna gioca *Forbice*, guadagnano rispettivamente 1 e -1.

	Sasso	Carta	Forbice
Sasso	0, 0	-1, 1	1, -1
Carta	1, -1	-1, 1	-1, 1
Forbice	-1, 1	1, -1	0, 0

Figura 2.1: Esempio di gioco in forma normale (Sasso, Carta e Forbice).

Esistono alcune classi speciali di giochi in forma normale che risultano di particolare interesse. Una di queste è quella dei *giochi a somma zero*, o più propriamente detti *giochi a somma costante*. In questa classe di giochi è richiesto che il numero dei giocatori sia esattamente pari a due.

Definizione 3 (Gioco a somma costante). Un gioco a due giocatori in forma normale è a somma costante se esiste una costante c tale che per ogni profilo di strategie $a \in A_1 \times A_2$ si ha $\mu_1(a) + \mu_2(a) = c$.

I giochi a somma zero sono quindi giochi a somma costante in cui $c = 0$. L'esempio in Figura 2.1 mostra proprio un gioco a somma zero. Questi giochi rappresentano situazioni di pura competizione tra i due giocatori.

Strategie nei giochi in forma normale. Sono state definite le azioni disponibili ad ogni giocatore in un gioco: viene ora definito l'insieme di *strategie* che essi possono scegliere. Un tipo di strategia può essere quello di scegliere una singola azione e giocarla. In questo caso la strategia si dice *pura*, e la scelta di strategie pure per ogni agente forma un *profilo di strategie pure*. I giocatori possono anche seguire un altro tipo di strategie: randomizzare su un insieme di azioni disponibili seguendo una distribuzione di probabilità. Una strategia di questo tipo è detta *strategia mista*. Nonostante possa non essere immediatamente ovvio il perché un giocatore dovrebbe introdurre la casualità nella scelta delle sue azioni, in realtà nei casi di multiagenti il ruolo delle strategie miste è critico. Una strategia mista per un gioco in forma normale è così definita.

Definizione 4 (Strategia Mista per un gioco in forma normale). Sia (N, A, μ) un gioco in forma normale, e per ogni insieme X sia $\prod(X)$ l'insieme di tutte le distribuzioni di probabilità su X. Allora l'insieme delle *strategie miste* per il giocatore i è $S_i = \prod(A_i)$.

Definizione 5 (**Profilo di strategie miste).** L'insieme dei *profili di strategie miste* è semplicemente il prodotto Cartesiano degli insieme di strategie miste individuali, $S_1 \times ... \times S_n$.

Si denoti con $s_i(a_i)$ la probabilità che un'azione venga giocata dal giocatore i sotto la strategia mista s_i. Il sottoinsieme di azioni a cui è assegnata una probabilità positiva nella strategia mista s_i è chiamato supporto di s_i.

Definizione 6 (**Supporto di una strategia mista).** Il *supporto* di una strategia mista s_i per un giocatore i è l'insieme di azioni $\{a_i | s_i(a_i) > 0\}$.

Una strategia pura è quindi un caso speciale di strategia mista in cui il supporto è un'unica azione. Dal lato opposto si hanno le strategie completamente miste, che sono strategie miste che assegnano ad ogni azione una probabilità positiva. Non sono stati ancora definiti i payoffs dei giocatori in un particolare profilo di strategie, dato che la matrice di payoff li definisce direttamente solo per il caso speciale di strategie pure. La generalizzazione alle strategie miste è comunque immediata, e si ottiene mediante la nozione base nella teoria delle decisioni, ovvero l'utilità attesa. Intuitivamente, si calcola prima la probabilità di raggiungere ogni outcome dato un profilo di strategie, e poi viene calcolata la media dei payoffs degli outcomes, pesati dalla probabilità di ogni outcome. Formalmente, si definisce l'utilità attesa come segue.

Definizione 7 (**Utilità attesa di una strategia mista).** Dato un gioco in forma normale (N, A, μ), l'utilità attesa μ_i per il giocatore i del profilo di strategie miste $s = (s_1, ..., s_n)$ è definito come

$$\mu_i(s) = \sum_{a \in A} \mu_i(a) \prod_{j=1}^{n} s_j(a_j).$$

2.3 Concetti di Soluzione dei giochi in forma normale

Nei giochi a singolo agente la nozione chiave nella teoria decisionale è quella di *strategia ottima*, ovvero una strategia che massimizza il payoff atteso dell'agente per il dato ambiente in cui l'agente opera. La situazione è però più complessa nel caso di multiagenti. In questo caso l'ambiente include o addirittura consiste in tutti gli altri agenti, ognuno dei quali tenta di massimizzare

il proprio payoff. Non è quindi possibile definire una strategia ottima per il singolo agente, in quanto la migliore strategia dipende dalle scelte degli altri agenti. Questo problema viene quindi affrontato cercando di identificare un certo sottoinsieme di outcomes, chiamati concetti di soluzione, che sono interessanti in un senso o un altro. In questa sezione quindi vengono introdotti e descritti alcuni dei più fondamentali concetti di soluzione.

2.3.1 Ottimalità di Pareto

Dal punto di vista di un osservatore esterno, la difficoltà nel definire un outcome migliore di altri in un gioco è dovuta all'impossibilità di definire l'interesse di un giocatore più importante dell'interesse degli altri. Il problema è quindi quello di trovare un modo per definire alcuni outcomes migliori di altri conoscendo solo la funzione di utilità dei giocatori. È importante notare che, mentre è solitamente impossibile identificare il miglior outcome, ci sono situazioni in cui si può essere sicuri che un outcome sia migliore di un altro. Per esempio, supponendo un gioco a due giocatori, è meglio l'outcome che prevede al giocatore 1 un payoff di 10 ed al giocatore 2 un payoff di 3 che un outcome che prevede al giocatore 1 un payoff di 9 ed al giocatore 2 un payoff di 3. Questa intuizione viene formalizzata con la definizione seguente.

Definizione 8 (**Dominazione di Pareto).** Un profilo di strategie s è *Pareto dominante* rispetto ad un profilo di strategie s' se per ogni $i \in N, \mu_i(s) \geq \mu_i(s')$, ed esiste un qualche $j \in N$ per il quale $\mu_j(s) > \mu_j(s')$.

In altre parole, in un profilo di strategie Pareto-dominate qualche giocatore può fare meglio senza portare altri giocatori a fare peggio. Si può osservare che viene definita la dominazione di Pareto su un profilo di strategie, non solo su un profilo di azioni. L'ottimalità di Pareto dà quindi un ordinamento parziale dei profili di strategie. In questo modo non si identifica un singolo outcome migliore, ma si può avere un insieme di outcome non ottimi.

Definizione 9 (**Ottimalità di Pareto).** Un profilo di strategie $s \in S$ è *Pareto ottimale*, o *strettamente Pareto efficiente*, se non esiste un altro profilo di strategie $s' \in S$ che sia Pareto dominante rispetto ad s.

Si possono facilmente trarre alcune conclusioni sui profili di strategie Pareto ottimali. Prima di tutto, ogni gioco ha almeno un ottimo di questo tipo, ed

inoltre esiste sempre almeno un ottimo in cui tutti i giocatori adottano strategie pure. Infine, alcuni giochi possono avere degli ottimi multipli. Per esempio, in un gioco a somma zero, tutti i profili di strategie sono strettamente Pareto efficienti.

2.3.2 Risposta Migliore ed Equilibrio di Nash

Vengono ora osservati i giochi dal punto di vista dell'agente individuale, invece che dal punto di vista di un osservatore esterno. Questo porta a considerare il più importante concetto di soluzione della Teoria dei Giochi, l'*equilibrio di Nash*. Si può notare che se un agente conoscesse come gli altri agenti giocano, il suo problema strategico diventerebbe semplice. Specificatamente, si potrebbe ricondurre ad un problema a singolo agente scegliendo la strategia che massimizza l'utilità. Formalmente, si definisce $s_{-i} = (s_1, ..., s_{i-1}, s_{i+1}, ..., s_n)$ come un profilo di strategie senza la strategia dell'agente i. Si può quindi considerare $s = (s_i, s_{-i})$. Se gli agenti diversi da i (denotati con $-i$) stanno per giocare il profilo s_{-i}, un agente i che massimizza l'utilità deve quindi affrontare il problema di determinare la sua *risposta migliore*.

Definizione 10 (**Risposta Migliore).** La *risposta migliore* di un giocatore i al profilo di strategie s_{-i} è una strategia mista $s_i^* \in S_i$ tale che $\mu_i(s_i^*, s_{-i}) \geq \mu_i(s_i, s_{-i})$ per ogni strategia $s_i \in S_i$.

La miglior risposta non è necessariamente unica. Invece, eccetto per il caso particolare in cui esiste un'unica risposta migliore che è una strategia pura, il numero di risposte migliore è sempre infinito. Se vi sono due strategie pure che sono individualmente delle risposte migliori, ogni distribuzione di probabilità sulle due azioni è necessariamente una risposta migliore. Naturalmente, in generale un agente non conosce quale strategia adotteranno gli altri agenti. Quindi la nozione di risposta migliore non è un concetto di soluzione, in quanto non identifica un insieme di outcomes interessanti in questo caso generale. Comunque, si può utilizzare l'idea della risposta migliore definendo la nozione centrale nella Teoria dei Giochi non cooperativi: l'equilibrio di Nash.

Definizione 11 (**Equilibrio di Nash).** Un profilo di strategie $s = (s_1, ..., s_n)$ è un *equilibrio di Nash (NE)* se, per ogni agente i, s_i è una risposta migliore a s_{-i}.

Intuitivamente, un equilibrio di Nash è un profilo di strategie stabile: nessun agente vorrebbe cambiare la sua strategia anche se conoscesse le strategie che seguiranno gli altri giocatori. Gli equilibri di Nash possono essere divisi in due categorie, *stretti* e *deboli*, a seconda che ogni strategia degli agenti costituisca o no un'unica risposta migliore alle strategie degli altri agenti.

Definizione 12 (**Equilibrio Stretto di Nash).** Un profilo di strategie $s = (s_1, ..., s_n)$ è un *equilibrio stretto di Nash* se, per ogni agente i e per ogni strategia $s'_i \neq s_i$, si ha $\mu_i(s_i, s_{-i}) > \mu_i(s'_i, s_{-i})$.

Definizione 13 (**Equilibrio Debole di Nash).** Un profilo di strategie $s = (s_1, ..., s_n)$ è un *equilibrio debole di Nash* se, per ogni agente i e per ogni strategia $s'_i \neq s_i$, si ha $\mu_i(s_i, s_{-i}) \geq \mu_i(s'_i, s_{-i})$, ed s non è un equilibrio stretto di Nash.

Intuitivamente, gli equilibri deboli di Nash sono meno stabili degli equilibri stretti di Nash, in quanto almeno un giocatore ha anche un'altra risposta migliore alle strategie degli altri giocatori che non sia la sua strategia di equilibrio. Gli equilibri di Nash in strategie miste sono necessariamente deboli, mentre gli equilibri di Nash in strategie pure possono essere stretti o deboli, a seconda del gioco. L'aspetto più importante dei NE è relativo all'esistenza degli equilibri stessi, ed è mostrato nel seguente teorema.

Teorema 1 (**Nash, 1951).** Ogni gioco con un numero finito di giocatori e di profili di azioni ha almeno un equilibrio di Nash.

2.3.3 Strategie Maxmin e Minmax

La *strategia maxmin* per il giocatore i in un gioco ad n-giocatori è una (non necessariamente unica, ed in generale mista) strategia che massimizza il payoff di i nel caso peggiore, ovvero nella situazione in cui tutti gli altri giocatori giocano le strategie che causano il maggior danno ad i. Il *valore di maxmin* (o *livello di sicurezza*) del gioco per il giocatore i è il minimo ammontare di payoff garantito dalla strategia maxmin.

Definizione 14 (**Maxmin).** La *strategia maxmin* per il giocatore i è $arg\ max_{s_i}\ min_{s_{-i}}\ \mu_i(s_i, s_{-i})$, ed il *valore maxmin* per il giocatore i è $max_{s_i}\ min_{s_{-i}}\ \mu_i(s_i, s_{-i})$.

La strategia maxmin è la miglior scelta quando dapprima i sceglie una (possibilmente mista) strategia, gli altri agenti $-i$ osservano questa strategia (ma non l'azione scelta da i) e poi scelgono le loro proprie strategie in modo da minimizzare il payoff atteso di i. Nonostante possa sembrare non ragionevole assumere che gli altri agenti siano interessati a minimizzare l'utilità di i, si ha che se i gioca una strategia maxmin e gli altri giocatori giocano arbitrariamente, i continuerà a ricevere un payoff almeno pari al suo valore maxmin. Questo significa che la strategia maxmin può essere una scelta sensata per un agente prudente, che vuole massimizzare la sua utilità attesa senza dover fare nessuna assunzione riguardo agli altri agenti. La strategia minmax ed il valore minmax giocano un ruolo duale rispetto alle controparti maxmin; sono utili nel caso in cui si voglia considerare quanto un giocatore può punire un altro senza badare al suo proprio payoff.

Definizione 15 (**Minmax).** La *strategia minmax* per il giocatore i contro il giocatore $-i$ è $arg\ min_{s_i}\ max_{s_{-i}}\ \mu_{-i}(s_i, s_{-i})$, ed il *valore minmax* per il giocatore $-i$ è $min_{s_i}\ max_{s_{-i}}\ \mu_{-i}(s_i, s_{-i})$.

Poiché né la strategia maxmin né la strategia minmax dipendono dalle strategie scelte dagli altri agenti, le strategie maxmin e minmax sono un concetto di soluzione. Viene detto *profilo di strategie misto maxmin* il profilo $s = (s_1, s_2, ..., s_n)$ di un dato gioco se s_1 è una strategia maxmin per il giocatore 1 ed s_2 è una strategia maxmin per il giocatore 2 (analogamente si può definire un profilo di strategie misto minmax).

In un gioco a due giocatori ed a somma zero, alla luce dei risultati proposti da Nash, è importante la seguente la seguente "rivisitazione" del teorema proposto da von Neumann nel 1928.

Teorema 2 (**Teorema Minimax (von Neumann, 1928)).** In un gioco finito a due giocatori ed a somma zero, in ogni equilibrio di Nash ogni giocatore riceve un payoff che è equivalmente ad entrambi il suo valore maxmin ed il suo valore minmax.

Questo teorema permette quindi di concludere che:

- Per ogni giocatore il suo valore maxmin coincide con il suo valore minmax. Per convenzione il valore maxmin del giocatore 1 è chiamato *valore del gioco*;

- Per entrambi i giocatori, l'insieme di strategie maxmin coincide con l'insieme di strategie minmax;
- Ogni profilo di strategie maxmin (o minmax) è un equilibrio di Nash.

2.3.4 Equilibrio di Nash Approssimato

Il seguente concetto di soluzione riflette l'idea che i giocatori possano non essere interessati a cambiare le proprie strategie in una risposta migliore (rispetto ad esse) quando l'ammontare di utilità che potrebbero guadagnare è molto basso. Questo porta all'idea di un *ϵ-equilibrio di Nash*, detto anche *ϵ-Nash*.

Definizione 16 (**ϵ-Nash).** Si fissi un $\epsilon > 0$. Un profilo di strategie $s = (s_1, ..., s_n)$ è un *ϵ-Nash* (*ϵ-NE*) se, per ogni agente i e per ogni strategia $s'_i \neq s_i$, si ha $\mu_i(s_i, s_{-i}) \geq \mu_i(s'_i, s_{-i}) - \epsilon$.

Gli ϵ-Nash esistono sempre; infatti, ogni equilibrio di Nash è circondato da una regione di ϵ-Nash per ogni $\epsilon > 0$. È però importante notare anche che, nonostante gli equilibri di Nash siano sempre circondati da ϵ-equilibri di Nash, il viceversa non è vero: un dato ϵ-Nash non necessariamente è vicino ad un equilibrio di Nash. È inoltre interessante notare che, se si considera un gioco in cui tutti i payoff sono tra 0 ed 1, allora ogni profilo di strategie è un 1-Nash.

Una nozione più forte di equilibrio approssimato di Nash è stata introdotta in [4]: per ogni $\epsilon > 0$, un *ϵ-Nash ben supportato* (*ϵ-well supported equilibria*) è una combinazione di strategie (pure o miste), una per ogni giocatore, in cui ogni giocatore assegna probabilità positiva solamente alle strategie pure che non portano più di ϵ in meno dell'utilità attesa di un equilibrio di Nash. Per semplicità, viene presentata la definizione nel caso di due giocatori.

Definizione 17 (**ϵ-Nash ben supportato).** Sia e_i il vettore colonna con un 1 alla i-esima posizione e 0 altrove. Per ogni $\epsilon > 0$, un profilo di strategie $s = (x, y)$ è un *ϵ-Nash ben supportato* per il gioco bimatriciale $(A, B) \in (\mathbb{R}, \mathbb{R})^{n \times m}$, se e solo se

- per ogni strategia pura $i \in [n]$, $x(i) > 0 \Rightarrow e_i^T Ay \geq e_k^T Ay - \epsilon \ \forall k \in [n]$,
- per ogni strategia pura $j \in [m]$, $y(j) > 0 \Rightarrow x^T Be_j \geq x^T Be_j - \epsilon \ \forall k \in [m]$.

Ogni ϵ-Nash ben supportato è anche un ϵ-Nash, ma in generale non vale il viceversa. Ovviamente, a maggior ragione, un equilibrio di Nash è uno 0-equilibrio ben supportato. Le precedenti due definizioni si basano su matrici normalizzate dei giochi, ovvero giochi in cui i valori sono compresi in $[0, 1]$. Questa assunzione assicura che ϵ appartenga ad un certo intervallo. Infatti, un ϵ-equilibrio di Nash di un certo gioco bimatriciale (A, B), corrisponde ad un $c\epsilon$-equilibrio di Nash per il gioco (cA, cB), con $c > 0$. Nel resto della tesi, quando ci si riferirà ad equilibri approssimati, si utilizzeranno giochi normalizzati o verrà calcolata una normalizzazione dei parametri per scalare il valore una volta trovato.

2.4 Giochi in Forma Estesa

La rappresentazione in forma normale dei giochi non incorpora nessuna nozione di sequenza, o di tempo, delle azioni dei giocatori. La *forma estesa* (o *albero*) è una forma di rappresentazione alternativa che rende esplicita la struttura temporale. In questa tesi si è interessati solo ai casi di *informazione perfetta*, che verrano descritti in questa sezione; una descrizione dei giochi in forma estesa con *informazione imperfetta* si può trovare in [37]. Informalmente, un gioco in forma estesa con informazione perfetta è un albero, nel senso della teoria dei grafi, in cui ogni nodo rappresenta la scelta di uno dei giocatori, ogni lato rappresenta una possibile azione, e le foglie rappresentano gli outcomes finali sui quali i giocatori hanno una funzione di utilità. Formalmente, vengono definiti come segue.

Definizione 18 (Gioco in forma estesa con informazione perfetta). Un gioco (finito) in *forma estesa con informazione perfetta* (o *gioco con informazione perfetta*) è una tupla $G = (N, A, H, Z, \chi, \rho, \sigma, \mu)$, dove:

- N è un insieme finito di n giocatori, indicizzato da i;
- A è un singolo insieme di azioni;
- H è un insieme di nodi di scelta non terminali;
- Z è un insieme di nodi terminali, disgiunto da H;

- $\chi : H \to 2^A$ è la funzione delle azioni, che assegna ad ogni nodo di scelta un insieme di possibili azioni;
- $\rho : H \to N$ è la funzione dei giocatori, che assegna ad ogni nodo non terminale un giocatore $i \in N$ che sceglie una azione in quel nodo;
- $\sigma : H \times A \to H \cup Z$ è la funzione dei successori, che mappa un nodo di scelta ed una azione ad un nuovo nodo di scelta o nodo terminale, tale che per ogni $h_1, h_2 \in H$ e $a_1, a_2 \in A$, se $\sigma(h_1, a_1) = \sigma(h_2, a_2)$ allora $h_1 = h_2$ e $a_1 = a_2$;
- $\mu = (\mu_1, ..., \mu_n)$ dove $\mu_i : Z \to \mathbb{R}$ è una funzione di utilità (o *payoff*) a valori reali per il giocatore i nei nodi terminali di Z.

Fintanto che i nodi di scelta formano un albero, si può senza ambiguità identificare un nodo con la sua storia, ovvero la sequenza di scelte che hanno portato dalla radice a quel nodo. Si possono anche definire i *discendenti* di un nodo h, quali tutti i nodi di scelta e terminali nel sottoalbero con radice h.

Un esempio di un gioco di questo tipo è il *gioco di Condivisione*. Si immagini un fratello ed una sorella che seguono il seguente protocollo per spartirsi due regali identici ed indivisibili ricevuti dai genitori. Prima il fratello propone una tra tre diverse divisioni, ovvero li tiene entrambi, ne tiene uno e uno va alla sorella, oppure vanno entrambi alla sorella. La sorella poi decide se accettare o rifiutare la proposta. Se accetta, i regali vengono distribuiti secondo la divisione accettata, altrimenti nessuno dei due riceve nessun regalo. Assumendo che valutino i due regali in modo uguale, l'albero di rappresentazione di questo gioco è mostrato in Figura 2.2.

Strategie per un gioco in forma estesa. Una strategia pura per un giocatore in un gioco con informazione perfetta è una specifica completa di quali azioni deve scegliere deterministicamente ad ogni nodo di decisione. Una definizione più formale è la seguente.

Definizione 19 (Strategia pura per un gioco in forma estesa). Sia $G = (N, A, H, Z, \chi, \rho, \sigma, \mu)$ un gioco in forma estesa con informazione perfetta. Allora l'insieme delle strategie pure per il giocatore i consiste nel prodotto cartesiano $\prod_{h \in H, \rho(h)=i} \chi(h)$.

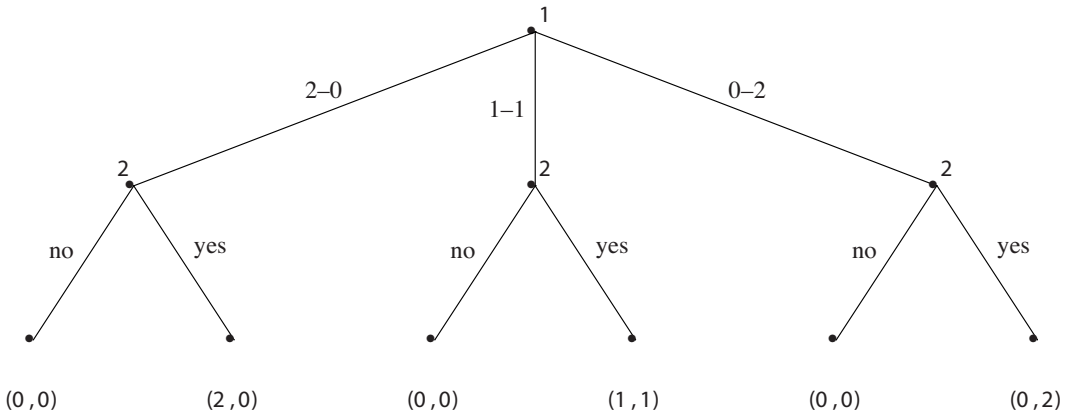

Figura 2.2: Esempio di gioco in forma Estesa: il gioco di Condivisione.

È importante notare che la strategia di un agente richiede una scelta ad ogni nodo di decisione, anche se è impossibile che tale nodo venga raggiunto date le scelte agli altri nodi. Una delle caratteristiche più importanti di questa classe di giochi è che ogni gioco con informazione perfetta può essere convertito in un gioco in forma normale equivalente. Le definizioni di risposta migliore e di equilibrio di Nash in questi giochi sono quindi esattamente le stesse dei giochi in forma normale. In generale, però, non sempre è possibile convertire un gioco in forma normale in un gioco in forma estesa con informazione perfetta. Intuitivamente, il problema che porta in generale all'impossibilità di questa trasformazione è che nei giochi in forma estesa con informazione perfetta non è possibile modellizzare la simultaneità. Il motivo per cui nella definizione precedente sono state considerate sole le strategie pure è nel seguente teorema.

Teorema 3 (**Zermelo, 1913).** Ogni gioco (finito) in forma estesa con informazione perfetta ha un equilibrio di Nash in strategie pure.

Intuitivamente il motivo sembra essere chiaro: fintanto che si arriva al turno di un giocatore, ed ognuno può vedere tutto quello che è successo prima di scegliere quale azione fare, tale giocatore non avrà mai bisogno di introdurre la casualità nella selezione delle sue azioni per trovare un equilibrio.

In ogni caso, l'equilibrio di Nash può rivelarsi una nozione troppo debole per la forma estesa, come mostrato nell'esempio seguente. Si consideri il gioco in forma estesa con informazione perfetta in Figura 2.3.

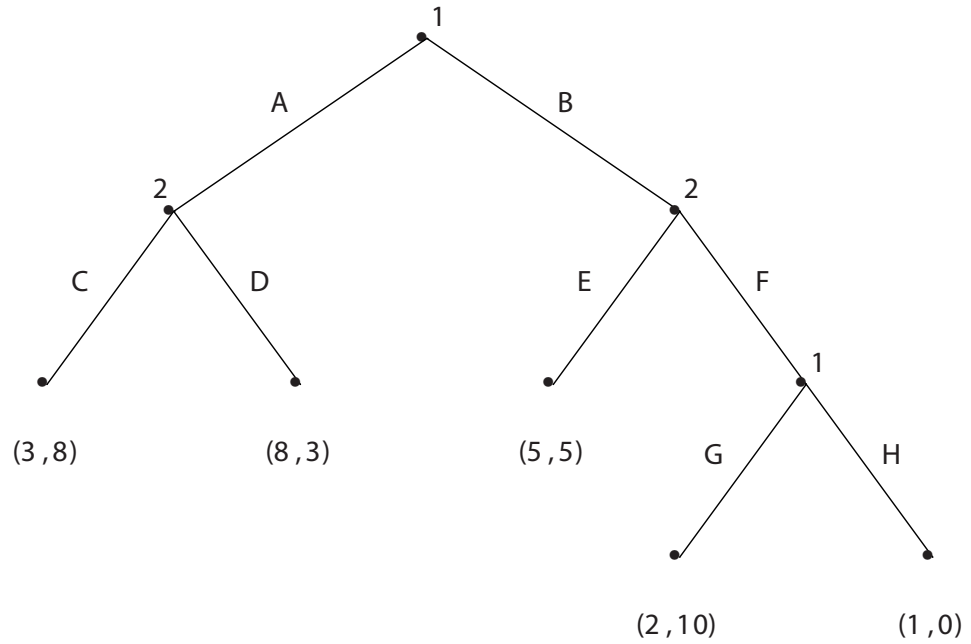

Figura 2.3: Esempio di gioco in forma Estesa.

Si consideri ora la conversione di tale gioco in forma normale, mostrato in Figura 2.4.

	(C, E)	(C, F)	(D, E)	(D, F)
(A, G)	3, 8	**3, 8**	8, 3	8, 3
(A, H)	3, 8	**3, 8**	8, 3	8, 3
(B, G)	5, 5	2, 10	5, 5	2, 10
(B, H)	**5, 5**	1, 0	5, 5	1, 0

Figura 2.4: Conversione in forma normale del gioco in Figura 2.3.

Si consideri il gioco in Figura 2.4. Vi sono tre equilibri di Nash in strategie pure in questo gioco: $\{(A, G), (C, F)\}, \{(A, H), (C, F)\}$, e $\{(B, H), (C, E)\}$, che sono evidenziati in figura. Si consideri l'equilibrio $\{(B, H), (C, E)\}$. Si può notare che in realtà è un equilibrio "non ragionevole" (i.e. non giocato da agenti razionali): infatti il giocatore 2 a seguito dell'azione B del giocatore 1, preferirebbe l'azione E all'azione F solo se a seguito dell'azione F il giocatore 1 scegliesse H. Ma se il giocatore 1 dovesse scegliere la propria seconda azione

nella parte destra dell'albero, sceglierebbe l'azione G che gli porterebbe un payoff pari a 2, invece dell'azione H che gli porterebbe un payoff pari ad 1. Quindi il giocatore 2 non può considerare la minaccia del giocatore 1 di scegliere H dopo B credibile. Per catturare formalmente la ragione per cui l'equilibrio $\{(B, H), (C, E)\}$ è "non ragionevole", e per definire un raffinamento del concetto di equilibrio che non soffra di questo problema, viene prima definita la nozione di *sottogioco*.

Definizione 20 (**Sottogioco).** Dato un gioco in forma estesa con informazione perfetta G, il *sottogioco* di G con radice al nodo h è la restrizione di G ai discendenti di h. L'*insieme dei sottogiochi* di G consiste di tutti i sottogiochi di G con come radice la stessa radice di G.

Si può ora definire la nozione di *equilibrio perfetto nei sottogiochi*, un raffinamento dell'equilibrio di Nash nei giochi in forma estesa con informazione perfetta, che elimina questi non voluti equilibri di Nash.

Definizione 21 (**Equilibrio perfetto nei sottogiochi).** Gli *equilibri perfetti nei sottogiochi (SPE)* di un gioco G sono tutti i profili di strategie s tali che per qualche sottogioco G' di G, le restrizioni ad s di G' sono equilibri di Nash di G'.

Nonostante l'SPE sia un concetto più forte dell'equilibrio di Nash (ovvero ogni SPE è un NE, ma non viceversa), si ha comunque che ogni gioco in forma estesa con informazione perfetta ha almeno un equilibrio perfetto nei sottogiochi. Questa definizione elimina le minacce non credibili come quella illustrata nel gioco in Figura 2.3. In particolare, si può notare che in tale gioco si ha un unico SPE, ovvero $\{(A, G), (C, F)\}$.

2.5 Giochi Stocastici

Un gioco stocastico può essere visto come una collezione di giochi in forma normale: gli agenti giocano ripetitivamente dei giochi da questa collezione, e ad ogni iterazione il particolare gioco giocato in tale iterazione dipende probabilisticamente dal gioco giocato nell'iterazione precedente e dalle azioni prese da tutti gli agenti in tale gioco. I giochi stocastici sono un ampio framework, che generalizza sia i *processi di decisione di Markov (MDP)* sia i *giochi ripetuti*.

Un MDP è semplicemente un gioco stocastico con un unico giocatore, mentre un gioco ripetuto è un gioco stocastico in cui vi la collezione di giochi contiene un gioco solo.

Definizione 22 (**Markov Decision Processes (MDPs)).** Un *processo di decisione di Markov* (*MDP*) è una tupla $(\mathcal{S}, A, P, r)$, dove

- $\mathcal{S}$ è un insieme (finito) di stati,
- A è un insieme (finito) di azioni,
- P è la funzione $p : S \times A \to \mathbb{R}$, tale che $p(s', s, a)$ specifica la probabilità di transizione dallo stato s' allo stato s scegliendo l'azione a,
- $r : \mathcal{S} \times A \to \mathbb{R}$ è la funzione di ricompensa (*reward*) per ogni coppia stato-azione.

Nei processi di decisione di Markov il singolo giocatore (o *agente*) inizia in un qualche stato nell'insieme degli stati, sceglie un'azione nello spazio delle azioni e riceve un qualche reward immediato, basato sullo stato attuale e sull'azione intrapresa in tale stato. A questo punto lo stato cambia secondo alcune transizioni probabilistiche, ed il processo si ripete. Una strategia per un MDP è quindi una funzione $\Pi : \mathcal{S} \to A$ detta *politica*, che mappa ogni stato in un'azione, indicando quale azione verrà scelta dal giocatore quando si troverà in uno stato.

Definizione 23 (**Gioco Stocastico).** Un gioco (finito) *stocastico* (anche conosciuto come *gioco Markoviano*) è una tupla $(\mathcal{S}, N, A, P, r)$, dove:

- $\mathcal{S}$ è un insieme finito di giochi;
- N è un insieme finito di n giocatori, indicizzato da i;
- $A = A_1 \times .... \times A_n$, dove A_i è un insieme finito di azioni disponibili al giocatore i;
- $P = \mathcal{S} \times \mathcal{S} \times A \to [0, 1]$ è la funzione di probabilità di transizione; $P(\hat{q}, q, a)$ è la probabilità di transizione dallo stato q allo stato $\hat{q}$ a seguito del profilo di azioni a;

- $R = r_1, ..., r_n$, dove $r_i : \mathcal{S} \times A \to \mathbb{R}$ è una funzione di payoff a valori reali per il giocatore i.

In questa definizione si assume che lo spazio delle strategie sia lo stesso in tutti i giochi, e che le differenze tra i giochi siano solo nelle funzioni di payoff. Queste assunzioni possono essere rimosse senza difficoltà, ma in questa descrizione non viene fatto per non aggiungere una notazione eccessiva che diminuirebbe la leggibilità. In questa definizione viene specificato il payoff di un giocatore ad ogni stato del gioco, ma non come questi payoffs sono aggregati in un payoff cumulativo. Il payoff cumulativo è però una somma di infiniti termini: per risolvere questo problema si utilizzano principalmente due metodi di aggregazione, ovvero la *ricompensa media* e la *ricompensa futura scontata* (o *discounted).*

Definizione 24 (**Ricompensa media).** Data una sequenza infinita di payoff $r_i^{(1)}, r_i^{(2)}, ...$ per il giocatore i, la *ricompensa media* di i è

$$\lim_{k\to\infty} \frac{\sum_{j=1}^{k} r_i^{(j)}}{k}.$$

La ricompensa futura discounted di un giocatore ad un certo punto del gioco è la somma dei suoi payoff nello stato immediato, più la somma delle ricompense future scontate da un fattore costante.

Definizione 25 (**Ricompensa futura scontata).** Data una sequenza infinita di payoff $r_i^{(1)}, r_i^{(2)}, ...$ per il giocatore i, ed un *fattore di sconto* (o di *discount*) β con $0 \leq \beta \leq 1$, la *ricompensa futura scontata* (o *discounted*) di i è

$$\sum_{j=1}^{\infty} \beta^j \, r_i^{(j)}.$$

Strategie per un gioco stocastico. Viene definito ora lo spazio di strategie di un agente. Sia $h_t = (q^0, a^0, q^1, a^1, ..., a^{t-1}, q^t)$ una storia di t fasi (ovvero t attraversamenti di stati) di un gioco stocastico, e sia H_t l'insieme di tutte le possibili storie di questa lunghezza. L'insieme di strategie deterministiche è il prodotto Cartesiano $\prod_{t,H_t} A_i$, che richiede una scelta per ogni possibile storia ad ogni momento di tempo. Come nelle forme di gioco precedenti, la strategia di un agente può consistere in una distribuzione di probabilità sull'insieme

delle strategie deterministiche. Comunque, ci sono molte classi ristrette di strategie che sono interessanti, ed esse formano la seguente gerarchia. La prima restrizione è alle *strategie comportamentali*, definite come segue.

Definizione 26 (**Strategie Comportamentali).** Una *strategia comportamentale* $s_i(h_t, a_{i_j})$ restituisce la probabilità di giocare l'azione a_{i_j} per la storia h_t.

Una *strategia di Markov* restringe una strategia comportamentale in modo che, ad ogni dato tempo t, la distribuzione sulle azioni dipende solo dallo stato corrente.

Definizione 27 (**Strategie di Markov).** Una *strategia di Markov* s_i è una strategia comportamentale in cui $s_i(h_t, a_{i_j}) = s_i(h'_t, a_{i_j})$ se $q_t = q'_t$, dove q_t e q'_t sono gli stati finali di h_t e h'_t, rispettivamente.

L'ultima restrizione è di rimuovere la possibile dipendenza dal tempo t.

Definizione 28 (**Strategie Stazionarie).** Una *strategia Stazionaria* s_i è una strategia di Markov in cui $s_i(h_{t_1}, a_{i_j}) = s_i(h'_{t_2}, a_{i_j})$ se $q_{t_1} = q'_{t_2}$, dove q_{t_1} e q'_{t_2} sono gli stati finali di h_{t_1} e h'_{t_2}, rispettivamente.

Ora si può considerare l'equilibrio dei giochi stocastici, argomento pieno di sottigliezze. Il caso di ricompensa discounted è il meno problematico: nel 1953 Shapley [36] mostrò che ogni gioco stocastico finito a discounted rewards e a somma zero ha un valore ottimo e delle strategie stazionarie ottime, ovvero che un equilibrio di Nash esiste in ogni gioco stocastico. Questi risultati sono stati anche estesi ai giochi stocastici a somma non zero in [10], [39], [41], ed anche ai giochi stocastici con spazio degli stati infinito e spazio delle azioni infinito [27]. Inoltre, si può definire una proprietà forte: un profilo di strategie è chiamato *equilibrio perfetto di Markov (MPE)* se consiste solo di strategie di Markov, ed un è equilibrio di Nash indipendentemente dallo stato di partenza. In questo senso, l'MPE gioca un ruolo analogo all'equilibrio di sottogioco perfetto nei giochi ad informazione perfetta.

Teorema 4. Ogni gioco stocastico ad n giocatori, a somma zero e con ricompensa discounted, ha un equilibrio perfetto di Markov.

Il caso di ricompensa media presenta difficoltà maggiori. Per prima cosa, il limite medio potrebbe non esistere (ovvero nonostante i payoffs del gioco nella fase siano limitati, la loro media potrebbe ciclare e non convergere). In ogni caso, si può considerare la classe dei giochi *stocastici irriducibili*. Un gioco stocastico è irriducibile se ogni profilo di strategie porta ad una catena di Markov irriducibile sull'insieme dei giochi; questo significa che ogni gioco può essere raggiunto con probabilità positiva senza riguardo della strategia adottata. In tali giochi, il limite medio è ben definito, e vale il seguente teorema.

Teorema 5. Ogni gioco stocastico, irriducibile, a due giocatori e con ricompensa media, ha un equilibrio di Nash.

Nel 1996 [19] venne proposta una prova alternativa dell'esistenza del valore del gioco nei giochi finiti, undiscounted, a somma zero che estendono al caso in cui lo spazio degli stati non è contabile. A questo punto, si può considerare anche un altro teorema: fintanto che ad ogni giocatore viene dato un payoff atteso che è almeno grande quanto il suo valore minmax, ogni coppia di payoff ammissibile può essere raggiunta in un equilibrio attraverso l'uso di minacce.

Teorema 6. Per ogni gioco stocastico, irriducibile, a due giocatori, ed ogni outcome ammissibile con un vettore di payoff r che fornisce ad ogni giocatore almeno il suo valore minmax, esiste un equilibrio di Nash con un vettore di payoff uguale ad r. Questo è vero per giochi con ricompensa media, così come per i giochi con un fattore di discount abbastanza grande (o, con giocatori che sono sufficientemente pazienti).

Calcolo degli equilibri. Calcolare il valore ottimo e le strategie ottime (nel caso di giochi a somma zero), ed un equilibrio di Nash (nel caso di giochi a somma non zero), hanno richiesto grandi sforzi nell'attività di ricerca, da un punto di vista sia computazionale che teoretico. Gli algoritmi ed i risultati per i giochi stocastici dipendono fortemente dal caso in cui si usi la ricompensa media o la ricompensa discounted per la funzione di utilità dell'agente. In questa tesi ci si concentra unicamente sul caso di ricompensa discounted. La prima domanda da porsi sul problema di trovare un equilibrio di Nash è se risulta dispondibile una qualche procedura polinomiale. Di fatto, esiste una formulazione in programmazione lineare per risolvere gli MDPs (sia nel caso

di ricompensa media sia nel caso di ricompensa discounted), che dà una ragione per essere ottimisti, dato che i giochi stocastici sono una generalizzazione degli MDPs. Mentre una tale formulazione non esiste per l'intera classe dei giochi stocastici, esiste per alcune sue sottoclassi. Una di queste sottoclassi è l'insieme dei giochi stocastici a due giocatori con ricompensa discounted, in cui le transizioni sono determinate da un singolo giocatore *(single controller)*. La condizione di single-controller è formalmente definita come segue.

Definizione 29 (Gioco stocastico a single-controller). Un gioco stocastico è a *single controller* se esiste un giocatore i tale che $\forall\ q,\ q' \in \mathcal{S}, \forall\ a \in A, P(q', q, a) = P(q', q, a')$ se $a_i = a'_i$.

Si ritornerà su questa classe di giochi nel Capitolo 6, quando ne verrà trattata un'estensione di particolare importanza per questa tesi.

Quando invece il problema non ricade in una di queste sottoclassi, continuano ad esistere soluzioni pratiche. Una di queste soluzioni, per esempio, è di applicare una versione modificata del metodo di Newton ad una formulazione del problema in programmazione non lineare: questo metodo ha, tra i vari vantaggi, quello della non esistenza dei minimi locali.

Viene ora presentata un'importante proprietà che possiedono alcune classi di giochi stocastici, tra cui anche i giochi stocastici a single-controller.

2.5.1 La Proprietà di Orderfield

Ogni volta che si affronta un problema descritto da un numero finito di parametri in un dato dominio, un'interessante domanda concerne la ricerca di una soluzione del problema che giaccia nel dato dominio. Per esempio, un insieme finito di equazioni lineari che ha una soluzione può essere risolto da un numero finito di operazioni algebriche (addizioni, sottrazioni, moltiplicazioni e divisioni), e quindi ha una soluzione in ogni campo che contiene tutti i parametri del sistema. Un altro esempio è la soluzione di un problema di programmazione lineare. Se tutti i parametri appartengono ad un campo ordinato fissato, il problema ha una soluzione se e solo se ha una soluzione in quel fissato campo ordinato. Una classe di problemi, che sono parametrizzati da un numero finito di elementi di un campo ordinato arbitrario, ha la *proprietà di orderfield* se ha una soluzione nello stesso campo ordinato. In particolare, una classe di

giochi parametrizzati da un numero finito di elementi di un campo ordinato arbitrario ha la proprietà di orderfield se ha una soluzione (valori minmax, strategie ottime, o strategie di equilibrio) nello stesso campo ordinato.

Nel 1950 Weyl [49] dimostrò che i giochi matriciali posseggono la proprietà di orderfield, ovvero che dati dei payoffs di un campo ordinato, esiste una coppia di strategie ottimali le cui coordinate giacciono nello stesso campo ordinato. Segue che il valore ottimo giace anch'esso nello stesso campo ordinato. Anche i giochi bimatriciali hanno la proprietà di orderfield quando ristretti al campo dei razionali. Nash nel 1951 [21] diede un esempio di un gioco a 3 giocatori non cooperativo con payoffs razionale ma un unico equilibrio irrazionale. Diversamente dai giochi matriciali, i giochi stocastici possono non possedere la proprietà di orderfield anche nel caso discounted a somma zero, come mostrato da Shapley nel 1953 [36]. Un esempio esplicito viene mostrato da Parthasarathy e Raghavan in [28], e da Vrieze in [46].

Formalmente, la proprietà di orderfield è definita nel modo seguente.

Definizione 30 (**Gioco Stocastico con proprietà Orderfield).** Un gioco stocastico a somma zero con ingressi in un certo campo ordinato (ovvero con payoffs, probabilità di transizioni e fattore di discount nel caso discounted appartenenti a tale campo ordinato), possiede la *proprietà di orderfield* se ha una coppia di strategie ottimali le cui coordinate sono in tale campo ordinato. Segue che il valore del gioco appartiene anch'esso a tale campo ordinato.

Un gioco stocastico a somma non zero con ingressi razionali possiede la proprietà di orderfield se ha una coppia di strategie in equilibrio di Nash le cui coordinate sono razionali. Segue che anche il corrispondente payoff dell'equilibrio è razionale.

Per alcune classi di giochi stocastici è stato dimostrato che soddisfano la proprietà di orderfield grazie alla loro speciale struttura. In particolare, Raghavan e Filar [28] dimostrarono che i giochi stocastici a single controller possiedono la proprietà di orderfield. Una trattazione approfondita di queste classi di giochi stocastici la si può trovare in [16], [23], [45], mentre una trattazione degli algoritmi per risolvere alcune di queste classi la si può trovare in [31].

Capitolo 3

Giochi Polinomiali in Forma Normale

3.1 Introduzione

In questo capitolo vengono introdotti i giochi a somma zero e a due giocatori, in cui la funzione di payoff é una espressione polinomiale nelle azioni dei giocatori. Questa classe di giochi é stata introdotta in [5] da Dresher, Karlin e Shapley. Nella Sezione 3.2 viene presentata questa classe di giochi, riconducendone la soluzione ad un problema di ottimizzazione peró astratto e quindi non risolvibile. Nella Sezione 3.3 viene presentata la caratterizzazione dei polinomi e dei momenti di una misura in problemi semidefiniti, formulazione necessaria per poi riscrivere, come descritto nella Sezione 3.4, il problema di ottimizzazione astratto in un problema di programmazione semidefinita risolvibile. Infine, nella Sezione 3.5 vengono riportati due esempi di soluzione computazionale di giochi polinomiali in forma normale.

3.2 Giochi Polinomiali in Forma Normale

Mentre molta della ricerca nell'ambito della Teoria dei Giochi si è concentrata, principalmente, nello sviluppo di tecniche computazionali per il calcolo degli equilibri nei giochi finiti (i.e., nei giochi in cui ogni giocatore ha un numero finito di strategie pure), recentemente ha assunto notevole interesse anche la classe dei cosiddetti *giochi infiniti*. In questa importante classe di giochi, i

giocatori hanno accesso ad un numero infinito di strategie pure non equivalenti, ovvero possono scegliere la propria azione tra un insieme infinito o non numerabile di azioni.

Un'importante sottoclasse dei giochi infiniti è quella dei *giochi polinomiali*: in questa classe di giochi, le azioni corrispondono a numeri reali, mentre la funzione di payoff è un'espressione polinomiale nelle azioni dei giocatori.

Definizione 31 (**Giochi Polinomiali).** Un gioco a due giocatori e somma zero è detto *gioco polinomiale* se la sua funzione di payoff è polinomiale nella forma

$$R(a_1, a_2) = \sum_{i=0}^{n} \sum_{j=0}^{m} r_{ij}\, {a_1}^i\, {a_2}^j, \tag{3.1}$$

dove a_1 e a_2, rispettivamente l'azione del giocatore 1 e del giocatore 2, sono elementi di un qualche insieme di strategie X ed Y, con X ed Y sottoinsiemi limitati (solitamente compatti) di spazi euclidei.

I giochi polinomiali hanno generato grande interesse subito dopo la loro introduzione nel 1950, come possibile collegamento tra i giochi finiti ed infiniti. Una proprietà molto importante dei giochi di questa classe, infatti, è che esistono sempre soluzioni di equilibrio con supporto finito, di dimensione proporzionale al grado della funzione di payoffs. Tuttavia, la mancanza di metodi computazionali efficienti portò ad una diminuzione dell'interesse su questa classe di giochi. Nonostante ciò, come viene discusso in questa sezione, i notevoli progressi nella teoria dell'ottimizzazione degli ultimi anni possono essere ben utilizzati per aggiornare in modo significativo questa valutazione.

In questa tesi ci si concentra sui giochi polinomiali a due giocatori, denotati con *Giocatore 1* e *Giocatore 2*, che simultaneamente ed indipendentemente scelgono azioni parametrizzate da numeri reali a_1 ed a_2, rispettivamente, in un intervallo chiuso e limitato (considereremo principalmente gli intervalli chiusi e limitati $[-1, 1]$ e $[0, 1]$).

La caratterizzazione e l'esistenza di strategie ottime per i giochi polinomiali sono state mostrate in [5]. Nel 2006, Parrilo [25] descrisse un algoritmo per risolvere i giochi polinomiali (a due giocatori e somma zero) attraverso tecniche di ottimizzazione in somma di quadrati e di programmazione semidefinita. In particolare, mostrò come caratterizzare e calcolare la soluzione ottima di questa classe di giochi, ovvero il valore del gioco e la strategia di uno dei due giocatori,

risolvendo un unico problema di programmazione semidefinita. Risolvendo poi il problema di programmazione semidefinita duale si ottiene anche la strategia del secondo giocatore. Vengono quindi discussi in questa sezione alcuni dei risultati ottenuti in [5] ed in [25].

3.2.1 Strategie d'equilibrio

Viene ora introdotto il problema del calcolo dell'equilibrio in un gioco a due giocatori e somma zero, con payoff polinomiale. Si consideri la classe di giochi in cui le azioni pure di ogni giocatore sono date dai numeri reali a_1 ed a_2, che appartengono all'intervallo chiuso e limitato $[c, d]$: ci si riferisce perciò ai giochi nel quadrato $\Omega = [c, d] \times [c, d]$. Il payoff è dato dall'equazione polinomiale (3.1), che esprime quanto il giocatore 2 deve "pagare" al giocatore 1. Perciò, il giocatore 1 dovrà scegliere l'azione a_1 che massimizza il payoff $R(a_1, a_2)$, mentre il giocatore 2 tenterà di rendere tale payoff il più piccolo possibile. Poichè si sta considerando la classe dei giochi a somma zero, il concetto di soluzione che verrà adottato è quello del minimax: dal Teorema 2, infatti, è noto che in giochi di questo tipo un profilo di strategie minimax è un equilibrio di Nash. Per il gioco quindi descritto dall'equazione (3.1), si può considerare il limite inferiore e superiore del valore del gioco. Questi possono essere derivati calcolando:

$$\max_{a_1} \min_{a_2} R(a_1, a_2) \quad \text{e} \quad \min_{a_2} \max_{a_1} R(a_1, a_2).$$

Non avendo fatto nessuna assunzione sui payoff $R(a_1, a_2)$, in generale, il valore maxmin sarà differente dal valore minmax, in quanto il gioco non necessariamente ammette una soluzione in strategie pure (come è noto dal caso ad azioni finite). Per ottenere una uguaglianza tra queste espressioni, è quindi necessario permettere l'utilizzo di strategie miste ai giocatori. Le strategie miste di ogni giocatore corrispondono alle misure di probabilità μ, ν sull'insieme di strategie pure, ovvero sull'intervallo $[c, d]$ su cui sono definite a_1 ed a_2. Allargando lo spazio delle strategie dei giocatori da pure a miste, si ottiene la nozione di equilibrio minimax per questi giochi.

Viene illustrato il concetto con un esempio.

Esempio 1 (Guessing Game). Si consideri il gioco (a due giocatori e somma zero) sul quadrato $[-1, 1] \times [-1, 1]$, con funzione di payoff data da

$R(a_1, a_2) = (a_1 - a_2)^2$. Fintanto che il giocatore 2 vuole minimizzare il suo payoff, tenterà di "indovinare" il numero scelto dal giocatore 1 e di scegliere tale numero, in modo da ottenere il payoff minimo, ovvero $R(a_1, a_2) = 0$. Il primo giocatore, quindi, tenterà di massimizzare il suo guadagno rendendo il valore da lui scelto il più difficile possibile da indovinare, scegliendo in modo casuale tra i due valori che gli possono portare il payoff maggiore, ovvero $a_1 = 1$ e $a_1 = -1$. A questo punto, per ridurre la quantità che dovrà "pagare" al giocatore 1, il giocatore 2 sceglierà il valore che minimizza la sua perdita, ovvero $a_2 = 0$. Nessuno dei due giocatori ha convenienza a cambiare la propria strategia (ovvero, nessuno dei due può guadagnare un payoff atteso maggiore cambiando la propria strategia), quindi il profilo di strategie trovato è un equilibrio, ed il valore del gioco è $\gamma = 0.5\ (1-0)^2 + 0.5\ (-1-0)^2 = 1$.

Considerando le strategie miste, similarmente al caso finito, è necessario considerare le espressioni dell'utilità attesa della coppia di strategie μ e ν, rispettivamente per il giocatore 1 e 2. In particolare, si denota con $\mu(a_1)$ la probabilità che il giocatore 1 esegua l'azione a_1 secondo la strategia μ, e con $\nu(a_2)$ la probabilità che il giocatore 2 esegua l'azione a_2 secondo la strategia ν. Si ha quindi:

$$\mathbb{E}_{\mu\times\nu}[R(a_1, a_2)] = \int_c^d \int_c^d \big(R(a_1, a_2)\, \mu(a_1)\, \nu(a_2)\big)\, da_1\, da_2.$$

A questo punto, sostituendo la funzione polinomiale espressa nell'equazione (3.1) si ottiene la seguente equazione:

$$\mathbb{E}_{\mu\times\nu}[R(a_1, a_2)] = \int_c^d \int_c^d \left(\sum_{i=0}^{n}\sum_{j=0}^{m} r_{ij}\, {a_1}^i\, {a_2}^j\, \mu(a_1)\, \nu(a_2)\right) da_1\, da_2.$$

Separando gli integrali, si ottiene la seguente equazione:

$$\mathbb{E}_{\mu\times\nu}[R(a_1, a_2)] = \sum_{i=0}^{n}\sum_{j=0}^{m} \left(r_{ij} \int_c^d {a_1}^i\, \mu(a_1)\, da_1 \int_c^d {a_2}^j\, \nu(a_2)\, da_2\right) \qquad (3.2)$$

Si denoti quindi con μ_i i momenti della misura di probabilità $\mu(a_1)$ di ordine i, e con ν_j i momenti della misura di probabilità $\nu(a_2)$ di ordine j, ovvero:

$$\mu_i = \int_c^d {a_1}^i\, \mu(a_1)\, da_1 \quad \text{e} \quad \nu_j = \int_c^d {a_2}^j\, \nu(a_2)\, da_2.$$

Si può riscrivere infine l'equazione (3.2) utilizzando i momenti delle misure μ_i e ν_j appena introdotti:

$$\mathbb{E}_{\mu\times\nu}[R(a_1,a_2)] = \sum_{i=0}^{n}\sum_{j=0}^{m} r_{ij}\,\mu_i\,\nu_j. \tag{3.3}$$

A questo punto, si consideri l'utilità attesa del profilo di strategie rispettivamente minimax e maxmin:

$$\max_{\mu}\min_{\nu}\ \mathbb{E}_{\mu\times\nu}[R(a_1,a_2)] \quad \text{e} \quad \min_{\nu}\max_{\mu}\ \mathbb{E}_{\mu\times\nu}[R(a_1,a_2)].$$

Si possono riscrivere tali utilità attese come espressioni bilineari, utilizzando la formulazione della utilità attesa ottenuta nella equazione (3.3):

$$\max_{\mu}\min_{\nu}\ \sum_{i=0}^{n}\sum_{j=0}^{m} r_{ij}\,\mu_i\,\nu_j \quad \text{e} \quad \min_{\nu}\max_{\mu}\ \sum_{i=0}^{n}\sum_{j=0}^{m} r_{ij}\,\mu_i\,\nu_j. \tag{3.4}$$

È ben noto che gli spazi dei momenti, ovvero l'immagine delle misure di probabilità sotto la mappa dei momenti data sopra, sono degli insiemi compatti e convessi in $\mathbb{R}^{n+1}$ e $\mathbb{R}^{m+1}$ [14]. Fintanto che la funzione obiettivo nel problema (3.4) è bilineare e gli insiemi ammissibili sono convessi e compatti, si può utilizzare una versione generalizzata del teorema minimax standard, per mostrare che queste due quantità sono esattamente uguali [5]. Inoltre, esistono le misure $\mu^\star, \nu^\star$ che soddisfano la condizione di punto sella (*saddle-point condition*) [25]:

$$\sum_{i=0}^{n}\sum_{j=0}^{m} r_{ij}\,\mu_i\,\nu_j^\star \ \le\ \sum_{i=0}^{n}\sum_{j=0}^{m} r_{ij}\,\mu_i^\star\,\nu_j^\star \ \le\ \sum_{i=0}^{n}\sum_{j=0}^{m} r_{ij}\,\mu_i^\star\,\nu_j. \tag{3.5}$$

Il fattore chiave, è che grazie alla struttura separabile dei payoffs, le strategie ottime possono essere caratterizzate solo in termini dei primi m ed n momenti (rispettivamente per μ e ν). I momenti di ordine maggiore sono irrilevanti, almeno in termini dei payoffs dei giocatori.

Grazie a quanto detto, è quindi possibile esprimere il seguente teorema, contenuto in [5]:

Teorema 7. Si consideri un gioco a due giocatori e somma zero su $[c,d]\times[c,d]$, con payoff dati dall'espressione (3.1). Allora, il valore del gioco è ben definito, ed esistono le strategie ottime miste $\mu^\star, \nu^\star$ che soddisfano la condizione di saddle-point. Inoltre, senza perdita di generalità, i supporti delle misure ottime sono finiti, con al massimo $\min(n,m)+1$ atomi.

3.2.2 Il Valore del Gioco

La derivazione del valore del gioco richiede un procedimento simile a quello svolto nel caso di azioni finite. Si caratterizzano prima le "strategie sicure", che garantiscono di fornire almeno un payoff minimo. Si può poi invocare la dualità convessa per provare che effettivamente questo "minimo payoff garantito" coincide con il valore del gioco. Procedendo in questo modo, per analogia al caso finito, una strategia sicura per il giocatore 2, ovvero il giocatore che vuole minimizzare il valore del gioco, può essere calcolata risolvendo il seguente problema di ottimizzazione:

$$\min_{\nu,\gamma} \gamma, \quad s.t. \begin{cases} \mathbb{E}_\nu[R(a_1, a_2)] & \leq & \gamma & \forall\, a_1 \in [c, d] \\ \int_c^d \nu(a_2)\, d(a_2) & = & 1 \end{cases} \tag{3.6}$$

Si cerca quindi la strategia ν che minimizza il valore massimo dell'utilità attesa che, fissata ν, sarà quindi in funzione dell'azione scelta dal giocatore 1. Infatti, se il giocatore 2 gioca la strategia mista ν ottenuta dalla soluzione di questo problema, la migliore strategia che il giocatore 1 può fare è quella di scegliere il valore di a_1 che massimizza $\mathbb{E}_\nu[R(a_1, a_2)]$, limitando quindi il suo guadagno atteso (e la perdita attesa del giocatore 2) a γ. Ovviamente, nel caso vi sia più di un possibile valore che massimizza l'utilità attesa $\mathbb{E}_\nu[R(a_1, a_2)]$ nell'intervallo $[c, d]$ dello spazio delle azioni del giocatore 1, il giocatore 1 potrà utilizzare una strategia mista il cui supporto sarà dato da tutte e sole queste azioni. Proprio per questo motivo, il giocatore 2 cerca la strategia ν tale da minimizzare il valore dei punti di massimo della funzione di utilità attesa, in quanto il giocatore 1 sceglierà poi, con una strategia mista, un'azione tra quelle che portano il payoff ad uno di questi punti di massimo. In questo modo il valore del gioco coinciderà proprio con il valore γ.

Si consideri ora il primo vincolo del problema di ottimizzazione (3.6): poichè deve valere per ogni a_1 nello spazio delle azioni del giocatore 1, si consideri un qualsiasi $a_1 \in [c, d]$ fissato. Fintanto che $R(a_1, a_2)$ è polinomiale, l'utilità attesa $\mathbb{E}_\nu[R(a_1, a_2)]$ può essere scritta in modo equivalente in termini dei primi n momenti della misura ν, ovvero si ottiene la seguente equazione.

$$\mathbb{E}_\nu[R(a_1, a_2)] = \int_c^d R(a_1, a_2)\, \nu(a_2)\, da_2 \;=\; \int_c^d \sum_{i=0}^{n} \sum_{j=0}^{m} r_{ij}\, {a_1}^i\, {a_2}^j\, \nu(a_2)\, da_2,$$

ovvero si ha:

$$\mathbb{E}_\nu[R(a_1, a_2)] = \sum_{i=0}^{n}\sum_{j=0}^{m} r_{ij}\, {a_1}^i \int_c^d {a_2}^j\, \nu(a_2)\, da_2 \;=\; \sum_{i=0}^{n}\sum_{j=0}^{m} r_{ij}\, \nu_j\, {a_1}^i.$$

Si può notare che l'utilità attesa è quindi data da un polinomio univariato nell'azione a_1 del giocatore 1 (in quanto l'espressione deve valere per ogni $a_1 \in [c, d]$), con coefficienti che dipendono dai momenti ν_j della strategia mista del giocatore 2. Questa proprietà sarà cruciale per il resto del capitolo.

Si consideri ora il problema di ottimizzazione (3.6), ma invece di scriverlo in termini della variabile di decisione ν (che è una misura di probabilità), verrano utilizzati i momenti $\{\nu_j\}_{j=0}^m$. Il problema si riduce quindi alla minimizzazione del valore di minmax γ, soggetto alle condizioni (date dai due vincoli):

- Il polinomio univariato $\gamma - \sum_{i=0}^{n}\sum_{j=0}^{m} r_{ij}\, \nu_j\, {a_1}^i$ è non negativo sull'intervallo $[c, d]$.
- La sequenza $\{\nu_j\}_{j=0}^m$ è una valida sequenza di momenti per una misura di probabilità supportata su $[c, d]$.

A questo punto, si può riscrivere il problema di ottimizzazione (3.6) in una forma più compatta, ottenendo il seguente problema di ottimizzazione:

$$\min_{\nu,\gamma} \gamma, \quad s.t. \begin{cases} \gamma - \sum_{i=0}^{n}\sum_{j=0}^{m} r_{ij}\, \nu_j\, {a_1}^i \;\in\; \mathcal{P}_n \\ \nu \;\in\; \mathcal{M}_m \\ \nu_0 \;=\; 1 \end{cases} \tag{3.7}$$

dove $\mathcal{P}_n$ è l'insieme dei polinomi univariati di grado n non negativi in $[c, d]$, ed $\mathcal{M}_m$ è l'insieme dei primi $m + 1$ momenti di una misura non negativa con supporto sullo stesso intervallo. In particolare, il primo vincolo del problema (3.7) corrisponde al primo vincolo del problema di partenza (3.6), mentre il secondo vincolo del problema (3.7) risulta necessario con l'introduzione della sequenza di momenti. Infine, il terzo vincolo del problema (3.7) corrisponde al secondo vincolo del problema di partenza (3.6), infatti:

$$\nu_0 \;=\; 1 \iff \int_c^d \nu(a_2)\, d(a_2) \;=\; 1.$$

La formulazione del problema di ottimizzazione (3.7) rimane comunque astratta, quindi non risolvibile: per cercare di convertirla in un problema di ottimizzazione concreto che si possa risolvere, è necessaria una rappresentazione

computazionalmente adatta di questi due insiemi ($\mathcal{P}_n$ ed $\mathcal{M}_m$). Questa formulazione verrà presentata nella prossima sezione permettendo, nella Sezione 3.4, di ottenere una rappresentazione del problema (3.7) in un singolo problema di programmazione semidefinita positiva.

3.3 Caratterizzazione di Polinomi e Momenti in Programmazione Semidefinita

Viene introdotta ora una formulazione volta a caratterizzare la non negatività dei polinomi ed alcune proprietà dei momenti di una misura. Esiste un rapporto stretto tra la non negatività dei polinomi e la programmazione semidefinita (SDP), e tale rapporto verrà descritto nella Sezione 3.3.1. Inoltre, nella Sezione 3.3.2 verranno mostrati quali sono i vincoli che si devono porre ad una sequenza di momenti, per garantire l'esistenza di una misura non negativa con esattamente tali momenti; anche in questo caso, vi è un forte rapporto tra tali vincoli e la programmazione semidefinita. L'obiettivo di questa sezione è quindi quello di presentare le caratterizzazioni dei primi due vincoli del problema di ottimizzazione (3.7), esprimendole come vincoli semidefiniti (per un'introduzione alla programmazione semidefinita, si consulti [3]). I risultati presentati in questa sezione derivano principalmente dai risultati ottenuti in [2], [24] e [26].

3.3.1 Polinomi Non Negativi ed SDP

Viene ora presentata la definizione di *polinomio univariato* e viene richiamato il Teorema Fondamentale dell'Algebra, che saranno poi necessari per esprimere la non negatività di un polinomio.

Definizione 32 (Polinomio Univariato). Un polinomio $p(x) \in \mathbb{R}$ di grado n si dice *univariato* se ha la forma:

$$p(x) = p_n x^n + p_{n-1} x^{n-1} + ... + p_1 x^1 + p_0,$$

dove i coefficienti p_k sono reali.

Normalmente si assume $p_n \neq 0$, ed occasionalmente verrà normalizzato a $p_n = 1$, nel cui caso il polinomio $p(x)$ si dice *monico*. Come è noto, il campo $\mathbb{C}$ dei numeri complessi è algebricamente chiuso:

Teorema 8 (**Teorema Fondamentale dell'Algebra).** Ogni polinomio univariato (non zero) di grado n ha esattamente n radici complesse (contate con la loro molteplicità). Inoltre, si ha l'unica fattorizzazione

$$p(x) = p_n \prod_{k=1}^{n} (x - x_k),$$

dove $x_k \in \mathbb{C}$ sono le radici di $p(x)$.

Se tutti i coefficienti p_k sono reali, se x_k è una radice, allora lo è anche il suo complesso coniugato x^*. In altre parole, tutte le k radici complesse appaiono in coppie complesse coniugate.

Una proprietà molto importante di un polinomio riguarda la possibilità o impossibilità che esso assuma valori non negativi. Viene quindi mostrato come caratterizzare questa proprietà.

Definizione 33 (**Polinomio Semidefinito Positivo o Non Negativo).** Un polinomio univariato $p(x)$ è *semidefinito positivo* o *non negativo* se $p(x) \geq 0$ per ogni valore reale di x.

Chiaramente, se $p(x)$ è non negativo, allora il suo grado dovrà essere un numero pari. L'insieme dei polinomi non negativi ha delle proprietà molto interessanti. La più importante per i nostri scopi è la seguente:

Teorema 9. Consideriamo l'insieme $\mathcal{P}_n$ dei polinomi univariati non negativi di grado minore o uguale ad n, con n pari. Allora, identificando un polinomio con i suoi $n+1$ coefficienti $(p_n, ..., p_0)$, l'insieme $\mathcal{P}_n$ è un cono *proprio* (ovvero convesso, solido, chiuso e con vertice) in $\mathbb{R}^{n+1}$.

(Per evitare una notazione troppo pesante, talvolta verrà escluso il grado di $\mathcal{P}_n$ nella notazione, quand'esso sarà comunque chiaro dal contesto.)

Viene ora introdotto il concetto di *somma di quadrati* ed il rapporto con la non negatività di un polinomio. Questo concetto sarà poi fondamentale per mostrare il collegamento tra la non negatività di un polinomio e la programmazione semidefinita.

Definizione 34 (**Polinomio in Somma di Quadrati).** Un polinomio univariato $p(x)$ è in *somma di quadrati (SOS)* se esistono $q_1, ..., q_m \in \mathbb{R}[x]$ tali che

$$p(x) = \sum_{k=1}^{m} q_k^2(x).$$

Se un polinomio $p(x)$ è in somma di quadrati, allora ovviamente soddisfa $p(x) \geq 0$ per ogni $x \in \mathbb{R}$. Quindi, la condizione di SOS è una condizione sufficiente per la non negatività globale. Inoltre, nel caso univariato, è vero anche l'inverso:

Teorema 10. Un polinomio univariato è non negativo se e solo se è in somma di quadrati.

Come verrà mostrato, è possibile verificare se un polinomio è in somma di quadrati (o, equivalentemente, se è non negativo) risolvendo un problema di ottimizzazione semidefinita. Si denoti con $\mathcal{S}^n$ l'insieme delle matrici reali simmetriche ed $n \times n$.

Definizione 35 (Matrice Semidefinita Positiva e Definita Positiva). Una matrice $A \in \mathcal{S}^n$ si dice *semidefinita positiva* se $x^T Ax \geq 0$ per ogni $x \in \mathbb{R}^n$, e si dice *definita positiva* se $x^T Ax > 0$ per ogni $x \in \begin{bmatrix} q_1(x) \\ q_2(x) \\ \vdots \\ q_m(x) \end{bmatrix}$ non nullo.

L'insieme delle matrici (reali, simmetriche, $n \times n$) semidefinite positive è denotato con $\mathcal{S}^n_+$, e l'insieme delle matrici (reali, simmetriche, $n \times n$) definite positive è denotato con $\mathcal{S}^n_{++}$. Il cono $\mathcal{S}^n_+$ è un cono proprio (ovvero convesso, solido, chiuso e con vertice).

Si consideri un polinomio $p(x)$ di grado $2d$ in somma di quadrati. Si può notare che il grado dei polinomi q_k è almeno equivalente a d, poiché il termine più grande di ogni q_k^2 è positivo, e quindi non ci possono essere cancellazioni nella potenza più grande di x. Si può quindi scrivere:

$$\begin{bmatrix} q_1(x) \\ q_2(x) \\ \vdots \\ q_m(x) \end{bmatrix} = V \begin{bmatrix} 1 \\ x \\ \vdots \\ x^d \end{bmatrix} \tag{3.8}$$

dove $V \in \mathbb{R}^{m \times (d+1)}$, e la sua k-esima riga contiene i coefficienti del polinomio q_k. Per riferimenti futuri, si denoti con $[x]_d$ il vettore nel lato destro della equazione (3.8). Si consideri ora la matrice $Q = V^T V$. Si ha quindi: $p(x) = \sum_{k=1}^m q_k^2(x) = (V[x]_d)^T (V[x]_d) = [x]_d^T V^T V [x]_d = [x]_d^T Q [x]_d$. Informalmente, si assume che esista una matrice simmetrica definita positiva

Q, per la quale $p(x) = [x]_d^T Q [x]_d$. Allora fattorizzando $Q = V^T V$, si arriva ad una decomposizione in somma di quadrati di p. Formalmente, questo concetto viene espresso nel seguente lemma, che dà una relazione diretta tra le matrici semidefinite positive e la condizione di somma di quadrati.

Lemma 1. Sia $p(x)$ un polinomio univariato di grado $2d$. Allora, $p(x)$ è non negativo (o in somma di quadrati) se e solo se esiste $Q \in \mathcal{S}_+^{d+1}$ che soddisfa

$$p(x) = [x]_d^T Q [x]_d.$$

Indicizzando le righe e le colonne di Q con $0, ..., d$, si ha:

$$[x]_d^T Q [x]_d = \sum_{j=0}^{d} \sum_{k=0}^{d} Q_{jk} x^{j+k} = \sum_{i=0}^{2d} (\sum_{j+k=i} Q_j k) x^i.$$

Quindi, per rendere questa espressione equivalente a $p(x)$, si deve avere il caso in cui

$$p_i = \sum_{j+k=i} Q_{jk}, \qquad i = 0, ..., 2d. \tag{3.9}$$

Questo è un sistema di $2d + 1$ equazioni lineari tra gli elementi di Q ed i coefficienti di $p(x)$. Quindi, fintanto che Q è simultaneamente vincolata ad essere semidefinita positiva e ad appartenere ad un particolare sottospazio affine, una condizione di SOS è esattamente equivalente ad un problema di programmazione semidefinita.

Lemma 2. Un polinomio $p(x) = \sum_{i=0}^{2d} p_i x^i$ è in somma di quadrati se e soltanto se e soltanto se esiste $Q \in \mathcal{S}_+^{d+1}$ che soddisfa la (3.9). Questo è un problema di programmazione semidefinita.

Si definisce ora l'operatore lineare $\mathcal{H} : \mathbb{R}^{2n-1} \to \mathcal{S}^n$ come:

$$\mathcal{H} : \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{2n-1} \end{bmatrix} \to \begin{bmatrix} a_1 & a_2 & \cdots & a_n \\ a_2 & a_3 & \cdots & a_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ a_n & a_{n+1} & \cdots & a_{2n-1} \end{bmatrix}$$

L'operatore lineare $\mathcal{H}$, semplicemente, riceve un vettore e ne costruisce la matrice di Hankel associata, ovvero una matrice costante lungo le antidiagonali.

Verrà utilizzato frequentemente anche l'operatore lineare $\mathcal{H}^* : \mathcal{S}^n \to \mathbb{R}^{2n-1}$, definito come:

$$\mathcal{H}^* : \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1n} \\ m_{12} & m_{22} & \cdots & m_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{1n} & m_{2n} & \cdots & m_{nn} \end{bmatrix} \to \begin{bmatrix} m_{11} \\ 2m_{12} \\ m_{22} + 2m_{13} \\ \vdots \\ m_{nn} \end{bmatrix}$$

Questo operatore riduce la matrice ad un vettore sommando tutti gli elementi lungo le antidiagonali. Si ottiene quindi il seguente risultato:

Lemma 3. Si consideri il polinomio $p(x) = \sum_{i=0}^{2d} p_i x^i$. Sia $\bar{p} = [p_0, ..., p_{2d}]^T$ il vettore dei suoi coefficienti. Allora $p(x)$ è in somma di quadrati (o non negativo) se e soltanto se e soltanto se esiste $S \in \mathcal{S}_+^{d+1}, \mathcal{S} \succeq 0$ tale che:

$$\bar{p} = \mathcal{H}^*(S).$$

Dimostrazione. Per i polinomi univariati, si sa per il Teorema 10 che la non negatività è equivalente alla condizione di SOS. Sia $[x]_n = [1, x, ..., x^n]^T$. Si ha che per ogni $\mathcal{S} \in \mathcal{S}^{n+1}$,

$$p(x) = \bar{p}^T [x]_{2n} = \mathcal{X}^*(S)^T [x]_{2n} = [x]_n^T S [x]_n.$$

A questo punto, fattorizzando $\mathcal{S} \succeq 0$, si ottiene una decomposizione in somma di quadrati. L'inverso è quindi immediato. □

Si mostra ora come caratterizzare la non negatività di un polinomio in termini di condizioni di SDP, non più nell'intervallo $(-\infty, +\infty)$, ma in un intervallo specifico. Si utilizzano quindi due caratterizzazioni, solitamente associate ai nomi di Pólya-Szegö, Fekete, o Markov-Lukacs. I risultati sono i seguenti:

Teorema 11. Il polinomio $p(x)$ è non negativo in $[0, +\infty)$, se e solo se può essere scritto come

$$p(x) = z(x) + xw(x),$$

dove $z(x), w(x)$ sono in somma di quadrati. Se $\deg(p) = 2d$, allora si ha $\deg(z) \leq 2d$, $\deg(w) \leq 2d-2$, mentre $\deg(p) = 2d+1$, allora $\deg(z) \leq 2d$, $\deg(w) \leq 2d$.

Teorema 12. Sia $a < b$. Allora $p(x)$ è non negativa su $[a, b]$, se e solo se può essere scritta come

$$\begin{cases} p(x) = z(x) + (x-a)(b-x)w(x), & \text{se } \deg(p) \text{ è pari} \\ p(x) = (x-a)z(x) + (b-x)w(x), & \text{se } \deg(p) \text{ è dispari} \end{cases}$$

dove $z(x), w(x)$ sono in SOS. Nel primo caso, si ha $\deg(p) = 2d$, e $\deg(z) \leq 2d$, $\deg(w) \leq 2d-2$. Nel secondo, $\deg(p) = 2d-1$, e $\deg(z) \leq 2d$, $\deg(w) \leq 2d$.

Si definiscono quindi le seguenti matrici:

$$L_1 = \begin{bmatrix} I_{n\times n} \\ 0_{1\times n} \end{bmatrix}, \; L_2 = \begin{bmatrix} 0_{1\times n} \\ I_{n\times n} \end{bmatrix}$$

dove $I_{n\times n}$ indica la matrice identità $n \times n$. Si mostra allora come caratterizzare la non negatività in modo analogo a quanto fatto nel Lemma 3. Poiché in questa tesi si considerano principalmente due intervalli, ovvero gli intervalli $[0, 1]$ e $[-1, 1]$, si può dare una caratterizzazione semidefinita esplicita di $\mathcal{P}([0, 1])$ e $\mathcal{P}([-1, 1])$.

Lemma 4. Il polinomio $p(x) = \sum_{i=0}^{2d} p_i x^i$ è non negativo in $[0, 1]$ se e solo se esistono le matrici $Z \in \mathcal{S}^{d+1}$ e $W \in \mathcal{S}^d$, $Z \succeq 0$, $W \succeq 0$ tali che

$$\begin{bmatrix} p_0 \\ \vdots \\ p_{2d} \end{bmatrix} = \mathcal{H}^*(Z + \frac{1}{2}(L_1 W L_2^T + L_2 W L_1^T) - L_2 W L_2^T).$$

Dimostrazione. La dimostrazione segue dalla caratterizzazione della non negatività di un polinomio su un intervallo. Infatti, per il Teorema 12, sapendo che $\deg(p)$ è pari e che $[a, b] = [0, 1]$, si ha che

$$p(x) \geq 0 \quad \forall x \in [0, 1] \iff p(x) = z(x) + x(1-x)w(x),$$

dove $z(x)$ e $w(x)$ sono in somma di quadrati. Una semplice applicazione del Lemma 3 porta alla condizione richiesta. □

Lemma 5. Il polinomio $p(x) = \sum_{i=0}^{2d} p_i x^i$ è non negativo in $[-1, 1]$ se e solo se esistono le matrici $Z \in \mathcal{S}^{d+1}$ e $W \in \mathcal{S}^d$, $Z \succeq 0$, $W \succeq 0$ tali che

$$\begin{bmatrix} p_0 \\ \vdots \\ p_{2d} \end{bmatrix} = \mathcal{H}^*(Z + L_1 W L_1^T - L_2 W L_2^T).$$

Dimostrazione. Come nella dimostrazione precedente, per il Teorema 12, si sa che

$$p(x) \geq 0 \quad \forall x \in [-1, 1] \iff p(x) = z(x) + (1 - x^2)w(x),$$

dove $z(x)$ e $w(x)$ sono in somma di quadrati. L'applicazione del Lemma 3 porta alla condizione richiesta. □

Si è quindi giunti alla caratterizzazione della non negatività di un polinomio sia nell'intervallo $(-\infty, +\infty)$, sia in un intervallo specifico, attraverso condizioni di programmazione semidefinita. Questo permetterà, come discusso nella Sezione 3.4, di derivare il primo vincolo per problema di ottimizzazione astratto (3.7) in vincoli di programmazione semidefinita. Viene ora analizzato il rapporto tra i momenti e la programmazione semidefinita, con l'obiettivo di arrivare ad una formulazione risolvibile del secondo vincolo del suddetto problema di ottimizzazione (3.7).

3.3.2 Momenti e Programmazione Semidefinita

Si consideri una misura non negativa μ su $\mathbb{R}$ (o se si preferisce, una variabile casuale X a valori reali). Si possono definire i momenti, che sono le medie delle potenze di X.

$$\mu_k := \mathbb{E}[X^k] = \int x^k \, d\mu. \tag{3.10}$$

Sia $\bar{\mu} = [\mu_0, ..., \mu_n]$ un vettore in $\mathbb{R}^{n+1}$. Si può dire che $\bar{\mu}$ è una *sequenza di momenti* di lunghezza $n + 1$ se corrisponde ai primi $n + 1$ momenti di una qualche misura non negativa μ supportata sull'insieme A. Lo *spazio dei momenti*, denotato da $\mathcal{M}(A)$, è il sottoinsieme di $\mathbb{R}^{n+1}$ che corrisponde ai momenti di una misura non negativa supportata sull'insieme A. Si dice che una misura non negativa μ è una *misura di probabilità* se il suo momento di

ordine zero soddisfa $\mu_0 = 1$. L'insieme delle sequenze dei momenti di lunghezza $n+1$ corrispondenti alle misure di probabilità è denotato da $\mathcal{M}_P(A)$.

È normale domandarsi quali vincoli deve soddisfare μ_k, e se per ogni insieme di valori $\bar{\mu}$ esiste sempre una misura non negativa che ha esattamente tali momenti. Fintanto che la misura μ è non negativa, è chiaro che si ha $\mu_k \geq 0$ per ogni $0 \leq k \leq n+1$; ovviamente però devono essere prese maggiori restrizioni. Un semplice vincolo può essere derivato ricordando la relazione tra il primo ed il secondo momento e la varianza di una variabile casuale, ovvero che $\text{var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2 = \mu_2 - \mu_1^2$. Fintanto che la varianza è sempre non negativa, si ha che $\mu_2 - \mu_1^2 \geq 0$. Viene quindi mostrato come derivare sistematicamente condizioni di questo tipo. Si può osservare che la precedente disuguaglianza può essere ottenuta notando che per ogni a e b si ha

$$0 \leq \mathbb{E}[(a+bX)^2] = a^2 + 2ab\mathbb{E}[X] + b^2\mathbb{E}[X^2] = \begin{bmatrix} a \\ b \end{bmatrix}^T \begin{bmatrix} 1 & \mu_1 \\ \mu_1 & \mu_2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix},$$

che implica che la matrice 2×2 sopra deve essere semidefinita positiva. È interessante notare che la disuguaglianza ottenuta precedentemente (con la relazione tra primo e secondo momento e la varianza) è esattamente equivalente al determinante di questa matrice.

La stessa procedura può essere esattamente estesa per i momenti di ordine più alto, considerando la disuguaglianza:

$$\mathbb{E}[(a_0 + a_1 x + a_2 x^2 + ... + a_d x^d)^2] \geq 0.$$

Procedendo in questo modo, si ha che i momenti di ordine maggiore devono soddisfare la seguente condizione:

$$\begin{bmatrix} 1 & \mu_1 & \mu_2 & \cdots & \mu_d \\ \mu_1 & \mu_2 & \mu_3 & \cdots & \mu_{d+1} \\ \mu_2 & \mu_3 & \mu_4 & \cdots & \mu_{d+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mu_d & \mu_{d+1} & \mu_{d+2} & \cdots & \mu_{2d} \end{bmatrix} \succeq 0. \tag{3.11}$$

Si può notare che gli elementi diagonali corrispondono ai momenti di ordine pari, che devono ovviamente essere non negativi.

Proprio come è stato fatto nel caso dei polinomi non negativi su un intervallo, si può similarmente ottenere una caratterizzazione necessaria e sufficiente

per i momenti. Per semplicità, viene presentato ora un particolare caso, quello corrispondente all'intervallo $[-1, 1]$.

Lemma 6. Esiste una misura non negativa in $[-1, 1]$ con momenti $(\mu_0, \mu_1, ..., \mu_{2d+1})$ se e solo se

$$\begin{bmatrix} \mu_0 & \mu_1 & \mu_2 & \cdots & \mu_d \\ \mu_1 & \mu_2 & \mu_3 & \cdots & \mu_{d+1} \\ \mu_2 & \mu_3 & \mu_4 & \cdots & \mu_{d+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mu_d & \mu_{d+1} & \mu_{d+2} & \cdots & \mu_{2d} \end{bmatrix} \pm \begin{bmatrix} \mu_1 & \mu_2 & \mu_3 & \cdots & \mu_{d+1} \\ \mu_2 & \mu_3 & \mu_4 & \cdots & \mu_{d+2} \\ \mu_3 & \mu_4 & \mu_5 & \cdots & \mu_{d+3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mu_{d+1} & \mu_{d+2} & \mu_{d+3} & \cdots & \mu_{2d+1} \end{bmatrix} \succeq 0. \tag{3.12}$$

Si può notare che la necessarietà è chiara, fintanto che segue dalla considerazione della forma quadratica (in a_i):

$$0 \leq \mathbb{E}[(1 \pm X)(\sum_{i=0}^{d} a_i X^i)^2] = \sum_{j=0}^{d} \sum_{k=0}^{d} (\mu_{j+k} \pm \mu_{j+k+1}) a_j a_k,$$

dove la prima disuguaglianza si ha poiché che $1 \pm X$ è sempre non negativo, fintanto che X è supportato su $[-1, 1]$. Si può notare anche la similarità (o dualità) con la condizione di non negatività per i polinomi. Poiché in questa tesi verrà considerato oltre all'intervallo $[-1, 1]$ anche l'intervallo $[0, 1]$, è importante anche il seguente lemma in cui, per semplificare la notazione, viene utilizzato l'operatore $\mathcal{H}$.

Lemma 7. Esiste una misura non negativa in $[0, 1]$ con momenti $(\mu_0, \mu_1, ..., \mu_{2d+1})$ se e solo se

$$\begin{gathered} \mathcal{H}\big([\mu_0, \mu_1, ..., \mu_{2d}]\big) - \mathcal{H}\big([\mu_1, \mu_2, ..., \mu_{2d+1}]\big) \succeq 0, \\ \mathcal{H}\big([\mu_1, \mu_2, ..., \mu_{2d+1}]\big) \succeq 0 \end{gathered} \tag{3.13}$$

Si mostra ora una esplicita caratterizzazione di $\mathcal{M}([-1, 1])$ e $\mathcal{M}_P([-1, 1])$.

Lemma 8. Il vettore $\mu = [\mu_0, \mu_1, ..., \mu_n]^T$ è un insieme dei momenti valido per una misura di probabilità in $[-1, 1]$ se e solo se

$$\mu_0 = 1,$$

$$\mathcal{H}(\mu) \succeq 0,$$

$$L_1^T \mathcal{H}(\mu) L_1 - L_2^T \mathcal{H}(\mu) L_2 \succeq 0.$$

Dimostrazione. La dimostrazione di questo risultato si può trovare in [14]. □

Poiché in questa tesi viene considerato oltre all'intervallo $[-1, 1]$ anche l'intervallo $[0, 1]$, è importante anche il seguente lemma.

Lemma 9. Il vettore $\mu = [\mu_0, \mu_1, ..., \mu_n]^T$ è un insieme dei momenti valido per una misura di probabilità in $[0, 1]$ se e solo se

$$\mu_0 = 1,$$
$$\mathcal{H}(\mu) \succeq 0,$$
$$\tfrac{1}{2}(L_1^T \mathcal{H}(\mu) L_2 + L_2^T \mathcal{H}(\mu) L_1) - L_2^T \mathcal{H}(\mu) L_2 \succeq 0.$$

Dimostrazione. La dimostrazione di questo risultato si può trovare in [14]. □

Per esempio, per $2n = 2$ la sequenza $[\mu_0, \mu_1, \mu_2]$ è una sequenza di momenti corrispondente ad una misura supportata su $[0, 1]$ se e solo se le seguenti disuguaglianze sono vere:

$$\begin{bmatrix} \mu_0 & \mu_1 \\ \mu_1 & \mu_2 \end{bmatrix} \succeq 0,$$
$$\mu_1 - \mu_2 \geq 0,$$

mentre, se la misura è supportata su $[-1, 1]$, la seconda disuguaglianza viene sostituita con

$$1 - \mu_2 \geq 0.$$

Viene ora introdotto un metodo per produrre una misura univariata atomica con un dato insieme dei momenti. La procedura seguente è classica, e può essere trovata in [14], [33] e [38]. Si consideri l'insieme dei momenti $(\mu_0, \mu_1, ..., \mu_{2n-1})$ per il quale si vuole trovare una misura associata non negativa, supportata sull'asse reale. La misura risultante sarà discreta (ovvero, composta da un numero finito di atomi), della forma $\sum_{i=1}^{n} w_i \delta(x - x_i)$, dove

$$\mathbf{Prob}(x = a_i) = w_i, \quad \forall i.$$

Si consideri quindi il sistema lineare:

$$\begin{bmatrix} \mu_0 & \mu_1 & \cdots & \mu_{n-1} \\ \mu_1 & \mu_2 & \cdots & \mu_n \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{n-1} & \mu_n & \cdots & \mu_{2n-2} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix} = - \begin{bmatrix} \mu_n \\ \mu_{n+1} \\ \vdots \\ \mu_{2n-1} \end{bmatrix}. \tag{3.14}$$

La matrice di Hankel nella parte sinistra di questa equazione è quella apparsa prima come condizione sufficiente per la rappresentabilità di una misura negativa da parte dei momenti. Si assuma senza perdita di generalità che tale matrice sia definita positiva. Invertendo tale matrice, si può risolvere il sistema ottenendo $[c_0, ..., c_{n-1}]^T$. Il sistema lineare in (3.14) ha infatti una unica soluzione se la matrice è definita positiva. In questo caso, siano x_i le radici del polinomio univariato

$$x^n + c_{n-1}x^{n-1} + ... + c_1 x + c_0 = 0.$$

Può essere dimostrato che tali radici sono tutte reali e distinte e sono esattamente i punti di supporto della misura discreta. Ora che sono stati ottenuti i supporti, è possibile quindi ottenere i corrispondenti pesi w_i risolvendo il sistema non singolare di Vandermonde dato da

$$\sum_{i=1}^{n} w_i x_i^j = \mu_j \qquad (0 \leq j \leq n-1).$$

3.4 Risolvere un Gioco Polinomiale con SDP

Ora che è stata introdotta la formulazione necessaria per caratterizzare l'insieme $\mathcal{P}_n$, i.e. l'insieme dei polinomi univariati di grado n non negativi in $[c, d]$, e l'insieme $\mathcal{M}_m$, i.e. l'insieme dei primi $m+1$ momenti di una misura non negativa con supporto sullo stesso intervallo, si può tornare al problema di ottimizzazione (3.7), a cui ci si era fermati al termine della Sezione 3.2 per capire come rappresentarlo in una forma concreta e risolvibile. Alcuni dei risultati presentati in questa sezione derivano dai risultati ottenuti in [25].

3.4.1 Dalla Ottimizzazione Polinomiale all'SDP

Per comodità, si riporta qui il problema di ottimizzazione.

$$\min_{\nu,\gamma} \gamma, \quad s.t. \begin{cases} \gamma - \sum_{i=0}^{n}\sum_{j=0}^{m} r_{ij}\,\nu_j\,{a_1}^i & \in & \mathcal{P}_n \\ \nu & \in & \mathcal{M}_m \\ \nu_0 & = & 1 \end{cases} \tag{3.7}$$

Per semplicità di rappresentazione, si considerano in questa sezione i giochi nel quadrato $\Omega = [-1,\, 1] \times [-1,\, 1]$, per mantenere una formulazione uniforme a

quella introdotta nella sezione precedente. È possibile convertire un tipo di gioco in un altro tramite una trasformazione lineare dello spazio di strategie [25]. Vengono ora riscritti i vincoli di questo problema di ottimizzazione esprimendoli come vincoli di un problema di programmazione semidefinita utilizzando i concetti introdotti nella Sezione 3.3.

Si inizi considerando il primo vincolo. Dalla sezione precedente, è noto che le condizioni di non negatività di un polinomio su un intervallo si applicano ai coefficienti di tale polinomio; per questo motivo, si consideri il polinomio $t(a_1) = \gamma - \sum_{i=0}^{n}\sum_{j=0}^{m} r_{ij}\,\nu_j\,{a_1}^i$, e si denoti con $\mathbf{t}$ il vettore dei coefficienti di $t(a_1)$, ovvero $\mathbf{t} = \gamma e_1 - R\nu$, dove $e_1 \in \mathbb{R}^{m+1}$ è il vettore che contiene tutti zeri, eccetto per il primo elemento che equivale ad 1. Si denoti inoltre con $\nu \in \mathbb{R}^{m+1}$ il vettore dei primi $m+1$ momenti di una misura ν, e con $R \in \mathbb{R}^{(n+1)\times(m+1)}$ la matrice che contiene i coefficienti r_{ij} del polinomio $R(a_1, a_2)$. Grazie al Lemma 5, si può concludere che il polinomio univariato $t(a_1)$ è non negativo in $[-1, 1]$ se e solo se esistono le matrici $Z \in \mathcal{S}^{n+1}$ e $W \in \mathcal{S}^n$, $Z \succeq 0$, $W \succeq 0$ tali che:

$$\mathbf{t} = \gamma e_1 - R\nu = \mathcal{H}^*(Z + L_1 W L_1^T - L_2 W L_2^T).$$

Viene quindi mostrato come riscrivere il secondo ed il terzo vincolo del problema (3.7), che riguardano i momenti. Grazie al Lemma 8, si può concludere che $\nu \in \mathcal{M}_m$ e $\nu_0 = 1$ con $[c, d] = [-1, 1]$ se e solo se

$$\begin{gathered}\nu_0 = 1,\\ \mathcal{H}(\nu) \succeq 0,\\ L_1^T \mathcal{H}(\nu) L_1 - L_2^T \mathcal{H}(\nu) L_2 \succeq 0.\end{gathered}$$

Si possono quindi rimettere insieme queste condizioni per formare il singolo problema di programmazione semidefinita positiva che concretizza il problema di ottimizzazione astratto (3.7), ottenendo quindi il seguente SDP.

$$\min_{\nu,\gamma,Z,W} \gamma, \quad s.t. \begin{cases} \mathcal{H}^*(Z + L_1 W L_1^T - L_2 W L_2^T) &= \gamma e_1 - R\nu \\ L_1^T \mathcal{H}(\nu) L_1 - L_2^T \mathcal{H}(\nu) L_2 &\succeq 0 \\ \mathcal{H}(\nu) &\succeq 0 \\ e_1^T \nu &= 1 \\ Z, W &\succeq 0 \end{cases} \tag{3.15}$$

Come sarà poi chiaro dal suo duale, la soluzione del seguente SDP corrisponde esattamente al valore del gioco ed ai momenti della strategia ottima per il giocatore 2.

Osservazione La soluzione dell'SDP (3.15) permette di ottenere il valore del gioco ed i primi $m+1$ momenti delle misure di probabilità. Rimane comunque un problema, che emergerà durante la ricostruzione delle strategie ottime dai loro momenti: a causa di questo problema, sarà necessario aggiungere dei vincoli all'SDP (3.15) per ottenere, oltre al valore del gioco ed ai momenti delle misure di probabilità, anche i supporti ed i pesi delle strategie miste ottime. Quest'osservazione si applica anche al problema duale, che viene ora introdotto.

3.4.2 Dualità

È noto che nei giochi a somma zero c'è una relazione naturale tra il ruolo dei due giocatori e le proprietà di dualità convessa del corrispondente problema di ottimizzazione. Informalmente, utilizzare il problema di ottimizzazione duale è equivalente a scambiare il ruolo dei giocatori. Risulta quindi interessante calcolare il duale dell'SDP (3.15), per ottenere la strategia del giocatore 1, che vuole massimizzare il valore del payoff. Il duale del problema di programmazione semidefinita (3.15) è dato dall'SDP seguente.

$$\max_{\mu,\gamma,A,B} \gamma, \quad s.t. \begin{cases} \mathcal{H}^*(A + L_1 B L_1^T - L_2 B L_2^T) &= R^T\mu - \gamma e_2 \\ L_1^T\mathcal{H}(\mu)L_1 - L_2^T\mathcal{H}(\mu)L_2 &\succeq 0 \\ \mathcal{H}(\mu) &\succeq 0 \\ e_2^T\mu &= 1 \\ A, B &\succeq 0 \end{cases} \tag{3.16}$$

dove $\mu \in \mathbb{R}^{n+1}, e_2 \in \mathbb{R}^{n+1}$. La formulazione di questo problema equivale a quella dell'SDP primale, eccetto che per il cambiamento del segno del valore del gioco (essendo ora una massimizzazione) e dell'uso di $-R^T$ invece di R, dovuto unicamente all'inversione di ruolo del giocatore che si sta considerando. Il gioco infatti rimane lo stesso, ma cambia il punto di vista, che diventa quello del giocatore avversario. Si ha quindi una perfetta corrispondenza tra

il primale ed il duale del gioco, data dal mappaggio $(R, Z, W, \nu, \gamma) \iff (-R^T, A, B, \mu, -\gamma)$. Ricordandosi di come è stato definito il payoff polinomiale nella (3.1), il mappaggio dalla matrice R a $-R^T$ corrisponde a $R(a_1, a_2) \leftrightarrow -R(a_2, a_1)$. Si può scrivere formalmente questo risultato come segue.

Teorema 13 ([25])**.** Si consideri un gioco Ω polinomiale a due giocatori e somma zero, tale che $\Omega = [-1, 1]\times[-1, 1]$, con payoff descritto in (3.1). Il valore del gioco ed i momenti delle strategie miste ottime, possono essere ottenuti risolvendo la coppia di SDP primale e duale data da (3.15) e (3.16).

3.4.3 Ricostruzione delle Strategie Ottime

Le variabili di decisione dei problemi SOS/SDP presentati precedentemente erano i momenti delle strategie miste. Le corrispondenti misure possono essere ricostruite dalle soluzioni ottime degli SDP primali e duali, in particolare dalle matrici $\mathcal{H}(\nu)$ e $\mathcal{H}(\mu)$. La procedura è indicata nella Sezione 3.3.2. Vi è però un problema: si può notare che, mentre sono stati calcolati i primi $m + 1$ momenti, ovvero i momenti sino all'ordine m, la procedura di ricostruzione richiede la presenza di un momento in più rispetto a quelli calcolati risolvendo l'SDP (3.15), richiedendo anche il momento di ordine $m+1$. A causa di questo problema, per poter ricostruire la strategia ottima per il giocatore 2 è necessario calcolare anche il momento "mancante" ν_{m+1}, vincolandolo in qualche modo ai momenti di ordine inferiore, oltre che all'intervallo su cui è definita la misura.

È possibile quindi utilizzare il Lemma 6, ed aggiungere i seguenti vincoli di programmazione semidefinita:

$$\begin{bmatrix} \nu_0 & \nu_1 & \nu_2 & \cdots & \nu_{\frac{m}{2}} \\ \nu_1 & \nu_2 & \nu_3 & \cdots & \nu_{\frac{m}{2}+1} \\ \nu_2 & \nu_3 & \nu_4 & \cdots & \nu_{\frac{m}{2}+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \nu_{\frac{m}{2}} & \nu_{\frac{m}{2}+1} & \nu_{\frac{m}{2}+2} & \cdots & \nu_m \end{bmatrix} + \begin{bmatrix} \nu_1 & \nu_2 & \nu_3 & \cdots & \nu_{\frac{m}{2}+1} \\ \nu_2 & \nu_3 & \nu_4 & \cdots & \nu_{\frac{m}{2}+2} \\ \nu_3 & \nu_4 & \nu_5 & \cdots & \nu_{\frac{m}{2}+3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \nu_{\frac{m}{2}+1} & \nu_{\frac{m}{2}+2} & \nu_{\frac{m}{2}+3} & \cdots & \nu_{m+1} \end{bmatrix} \succeq 0,$$

$$\begin{bmatrix} \nu_0 & \nu_1 & \nu_2 & \cdots & \nu_{\frac{m}{2}} \\ \nu_1 & \nu_2 & \nu_3 & \cdots & \nu_{\frac{m}{2}+1} \\ \nu_2 & \nu_3 & \nu_4 & \cdots & \nu_{\frac{m}{2}+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \nu_{\frac{m}{2}} & \nu_{\frac{m}{2}+1} & \nu_{\frac{m}{2}+2} & \cdots & \nu_m \end{bmatrix} - \begin{bmatrix} \nu_1 & \nu_2 & \nu_3 & \cdots & \nu_{\frac{m}{2}+1} \\ \nu_2 & \nu_3 & \nu_4 & \cdots & \nu_{\frac{m}{2}+2} \\ \nu_3 & \nu_4 & \nu_5 & \cdots & \nu_{\frac{m}{2}+3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \nu_{\frac{m}{2}+1} & \nu_{\frac{m}{2}+2} & \nu_{\frac{m}{2}+3} & \cdots & \nu_{m+1} \end{bmatrix} \succeq 0.$$

Questi vincoli, grazie all'utilizzo dell'operatore $\mathcal{H}$, possono essere riscritti con le seguente formulazione più sintetica.

$$\mathcal{H}([\nu_0,\, \nu_1,\, ...,\, \nu_m]^T) \;+\; \mathcal{H}([\nu_1,\, \nu_2,\, ...,\, \nu_{m+1}]^T) \;\succeq\; 0,$$

$$\mathcal{H}([\nu_0,\, \nu_1,\, ...,\, \nu_m]^T) \;-\; \mathcal{H}([\nu_1,\, \nu_2,\, ...,\, \nu_{m+1}]^T) \;\succeq\; 0.$$

È quindi possibile aggiungere questi due vincoli semidefiniti all'SDP (3.15) per ottenere la sequenza di momenti $(\nu_0, \nu_1, ..., \nu_{m+1})$, sufficiente alla ricostruzione delle misure di probabilità delle strategie miste ottime. Lo stesso procedimento va poi effettuato, ovviamente, anche all'SDP duale (3.16). Si ottiene quindi la seguente coppia primale e duale di SDP.

$$\min_{\nu,\nu_{m+1},\gamma,Z,W} \gamma, \quad \textit{subject to}$$

$$\begin{cases} \mathcal{H}^*(Z + L_1 W L_1^T - L_2 W L_2^T) &= \gamma e_1 - R\nu \\ L_1^T \mathcal{H}(\nu) L_1 - L_2^T \mathcal{H}(\nu) L_2 &\succeq 0 \\ \mathcal{H}(\nu) &\succeq 0 \\ e_1^T \nu &= 1 \\ \mathcal{H}([\nu_0,\, \nu_1,\, ...,\, \nu_m]^T) + \mathcal{H}([\nu_1,\, \nu_2,\, ...,\, \nu_{m+1}]^T) &\succeq 0 \\ \mathcal{H}([\nu_0,\, \nu_1,\, ...,\, \nu_m]^T) - \mathcal{H}([\nu_1,\, \nu_2,\, ...,\, \nu_{m+1}]^T) &\succeq 0 \\ Z, W &\succeq 0 \end{cases} \tag{3.17}$$

$$\max_{\mu,\mu_{m+1},\gamma,A,B} \gamma, \quad \textit{subject to}$$

$$\begin{cases} \mathcal{H}^*(A + L_1 B L_1^T - L_2 B L_2^T) &= R^T\mu - \gamma e_2 \\ L_1^T \mathcal{H}(\mu) L_1 - L_2^T \mathcal{H}(\mu) L_2 &\succeq 0 \\ \mathcal{H}(\mu) &\succeq 0 \\ e_2^T \mu &= 1 \\ \mathcal{H}([\mu_0,\, \mu_1,\, ...,\, \mu_m]^T) + \mathcal{H}([\mu_1,\, \mu_2,\, ...,\, \mu_{m+1}]^T) &\succeq 0 \\ \mathcal{H}([\mu_0,\, \mu_1,\, ...,\, \mu_m]^T) - \mathcal{H}([\mu_1,\, \mu_2,\, ...,\, \mu_{m+1}]^T) &\succeq 0 \\ A, B &\succeq 0 \end{cases} \tag{3.18}$$

dove $e_1 \in \mathbb{R}^{m+1}, e_2 \in \mathbb{R}^{n+1}$ sono i vettori che contengono tutti zeri, eccetto per il primo elemento che equivale ad 1, $\nu \in \mathbb{R}^{m+1}$ è il vettore dei primi $m+1$ momenti di una misura ν, $\mu \in \mathbb{R}^{n+1}$ è il vettore dei primi $n+1$ momenti di una misura μ, μ_{n+1} e ν_{m+1} sono i momenti rispettivamente di μ e di ν di ordine rispettivamente $n+1$ ed $m+1$, ed $R \in \mathbb{R}^{(n+1)\times(m+1)}$ è la matrice che contiene i coefficienti r_{ij} del polinomio $R(a_1, a_2)$.

A questo punto, utilizzando la procedura indicata nella Sezione 3.3.2 è possibile ricostruire sia i supporti sia i pesi delle misure di probabilità μ e ν, ottenendo quindi le strategie miste ottime. È importante notare che il supporto delle misure atomiche sarà dato dalle radici dei polinomi con coefficienti dati da $\gamma e_1 - R\nu$ e $R^T\mu - \gamma e_1$: questo significa che i supporti delle strategie miste ottime sono finiti, ed in particolare il giocatore 1 avrà un supporto composto da non più di n azioni, mentre il giocatore 2 avrà un supporto composto da non più di m azioni.

Questo risultato fornisce una naturale e completa generalizzazione della classica soluzione in programmazione lineare dei giochi finiti a somma zero, e ne condivide le stesse importanti proprietà di dualità.

Poiché in molte applicazioni (ad esempio economiche o di networking) è interessante studiare sotto quali condizioni è garantito che un gioco possegga soluzioni ottime pure, viene ora presentato un risultato noto riguardo ai giochi continui (da [42, Teorema 4.5]).

Teorema 14. Si consideri un gioco polinomiale in $[-1,1] \times [-1,1]$ descritto da $R(a_1, a_2)$. Se $R(a_1, a_2)$ è strettamente concavo in a_1 per ogni $a_2 \in [-1,1]$ e strettamente convesso in a_2 per ogni $a_1 \in [-1,1]$, allora entrambi i giocatori hanno strategie ottime che sono pure.

Questo significa che per giochi di questo tipo la coppia primale-duale di SDP (3.17) e (3.18) ammetterà soluzioni pure, che saranno uniche se il gioco ha un unico equilibrio. Va comunque notato che se è noto che il gioco ha queste proprietà di concavità e convessità, non c'è nessun vantaggio computazionale nel lavorare sullo spazio dei momenti invece che nel naturale spazio di strategie del gioco.

3.5 Esempio di Soluzione

Vengono ora presentati due semplici esempi. Nel primo esempio, il gioco ha solo soluzioni in strategie pure, mentre nel secondo esempio il gioco richiede strategie miste per ottenere il valore ottimo del gioco. Entrambi gli esempi vengono trattati anche da Parrilo in [25].

3.5.1 Esempio di Gioco con Strategie Pure

Si consideri il gioco polinomiale (a due giocatori e somma zero) in forma normale, in cui le azioni a_1 ed a_2 (rispettivamente del primo e del secondo giocatore) sono definite sull'intervallo chiuso e limitato $[-1, 1]$. Il gioco, definito sul quadrato $\Omega = [-1, 1] \times [-1, 1]$, è dato da:

$$R(a_1, a_2) \;=\; 2a_1{a_2}^2 \;-\; {a_1}^2 \;-\; a_2.$$

Si vuole capire quali sono le strategie ottime per i due giocatori e qual è il valore del gioco. Si hanno quindi le seguenti costanti.

$$m = n = 2,\; R = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 2 \\ -1 & 0 & 0 \end{bmatrix},\; e_1 = e_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},\; L_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\; L_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Si hanno poi le seguenti espressioni.

$$\mathcal{H}(\nu) \succeq 0 \iff \begin{bmatrix} \nu_0 & \nu_1 \\ \nu_1 & \nu_2 \end{bmatrix} \succeq 0,\;\; L_1^T\mathcal{H}(\nu)L_1^T - L_2^T\mathcal{H}(\nu)L_2 \succeq 0 \iff 1 - \nu_2 \succeq 0$$

Inoltre, avendo $Z \in \mathcal{S}^2, W \in \mathcal{S}^1$, il primo vincolo dell'SDP (3.17) può essere scritto come segue.

$$\mathcal{H}^*\left(\begin{bmatrix} z_1 & z_2 \\ z_2 & z_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} W[1,0] - \begin{bmatrix} 0 \\ 1 \end{bmatrix} W[0,1] \right) \;=\; [\gamma, 0, 0] - \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 2 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \nu_0 \\ \nu_1 \\ \nu_2 \end{bmatrix}$$

Ovvero, si riconduce alla seguente espressione.

$$\mathcal{H}^*\left(\begin{bmatrix} z_1 + W & z_2 \\ z_2 & z_3 - W \end{bmatrix} \right) = \begin{bmatrix} \gamma + \nu_1 \\ -2\nu_2 \\ \nu_0 \end{bmatrix}$$

Che, tradotto in forma polinomiale, assume il seguente significato.

$$z_1 + 2a_1 z_2 + z_3 {a_1}^2 + (1 - {a_1}^2) W \;=\; \gamma + \nu_0 {a_1}^2 - 2a_1\nu_2 + \nu_1$$

Il quinto ed il sesto vincolo dell'SDP (3.17), che vincolano il momento ν_3 di ordine $m+1=3$, sono quindi i seguenti.

$$\begin{bmatrix} \nu_0 & \nu_1 \\ \nu_1 & \nu_2 \end{bmatrix} + \begin{bmatrix} \nu_1 & \nu_2 \\ \nu_2 & \nu_3 \end{bmatrix} \succeq 0, \quad \begin{bmatrix} \nu_0 & \nu_1 \\ \nu_1 & \nu_2 \end{bmatrix} - \begin{bmatrix} \nu_1 & \nu_2 \\ \nu_2 & \nu_3 \end{bmatrix} \succeq 0.$$

A questo punto, quindi, si può scrivere l'SDP associato a questo problema, la cui soluzione fornisce il valore del gioco ed i primi $m+2$ momenti della misura di probabilità della strategia del giocatore 2, ovvero il giocatore che vuole minimizzare il payoff.

$$\min_{\nu,\nu_{m+1},\gamma,Z,W} \gamma, \quad s.t. \begin{cases} \begin{bmatrix} z_1 + W \\ 2z_2 \\ z_3 - W \end{bmatrix} = \begin{bmatrix} \gamma + \nu_1 \\ -2\nu_2 \\ \nu_0 \end{bmatrix} \\ 1 - \nu_2 \succeq 0 \\ \begin{bmatrix} \nu_0 & \nu_1 \\ \nu_1 & \nu_2 \end{bmatrix} \succeq 0 \\ \nu_0 = 1 \\ \begin{bmatrix} \nu_0 & \nu_1 \\ \nu_1 & \nu_2 \end{bmatrix} + \begin{bmatrix} \nu_1 & \nu_2 \\ \nu_2 & \nu_3 \end{bmatrix} \succeq 0 \\ \begin{bmatrix} \nu_0 & \nu_1 \\ \nu_1 & \nu_2 \end{bmatrix} - \begin{bmatrix} \nu_1 & \nu_2 \\ \nu_2 & \nu_3 \end{bmatrix} \succeq 0 \\ Z, W \succeq 0 \end{cases} \tag{3.19}$$

La soluzione di questo problema di programmazione semidefinita produce i seguenti risultati.

$$\gamma = \alpha^4 - \alpha, \;\; \nu = \begin{bmatrix} 0 \\ \alpha \\ \alpha^2 \end{bmatrix}, \;\; \nu_3 = 0.25, \;\; Z = \begin{bmatrix} \alpha^4 & -\alpha^2 \\ -\alpha^2 & 1 \end{bmatrix}, \;\; W = 0,$$

con $\alpha = 4^{-\frac{1}{3}}$. A questo punto, utilizzando la sequenza di momenti $(\nu_0, \nu_1, \nu_2, \nu_3) = (0, \alpha, \alpha^2, 0.25)$ ed il procedimento descritto nella Sezione 3.3.2,

è possibile ricostruire sia il supporto della strategia del giocatore 2 sia i pesi, ovvero i valori di probabilità associati alla singola azione secondo la misura di probabilità ν. A scopo dimostrativo, viene svolto tale procedimento: si ha quindi la seguente equazione

$$\begin{bmatrix} \nu_0 & \nu_1 \\ \nu_1 & \nu_2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} \nu_2 \\ \nu_3 \end{bmatrix}.$$

Invertendo la matrice immediatamente a sinistra e risolvendo l'equazione, si ottiene il vettore $[c_0, c_1]^T = [0, -0.63]$ dei coefficienti di un polinomio $s(t) = t^2 + c_1 t^1 + c_0$. Calcolando le radici del polinomio $s(t)$ si ottengono i seguenti zeri: $s_1 = \alpha, s_2 = 0$, che coincidono con i punti del supporto della misura di probabilità ν. Risolvendo anche il sistema non singolare di Vandermonde si ottengono i rispettivi pesi: $w_1 = 1, w_2 = 0$.

Risolvendo in modo analogo anche il problema duale si ottiene anche la strategia del giocatore 1, oltre ad ottenere, ovviamente, un valore del gioco che coincide con quello calcolato con il primale. L'equilibrio, quindi, è dato dal profilo di strategie in cui il giocatore 1 sceglie sempre l'azione $a_1 = \alpha^2$, ed il giocatore 2 sceglie sempre l'azione $a_2 = \alpha$. Il fatto che le strategie ottime siano pure non è inatteso, ma è motivato sia dal Teorema 14 sia dal fatto che la funzione di payoff è concava in a_1. L'equilibrio corrisponde all'unico punto di sella della funzione $R(a_1, a_2)$ nel dominio della funzione. Si può anche notare che, poichè in questo caso vi è un unico equilibrio in strategie pure, il punto di sella si sarebbe anche potuto calcolare in un modo più diretto usando il metodo delle "curve di reazione" (si veda [20], [42]).

3.5.2 Esempio di Gioco con Strategie Miste

Si consideri il gioco polinomiale a due giocatori e somma zero che, come nel caso precedente, è sul quadrato $\Omega = [-1, 1] \times [-1, 1]$. La funzione di payoff è data da:

$$R(a_1, a_2) = 5a_1a_2 - 2{a_1}^2 - 2a_1{a_2}^2 - a_1,$$

che non è nè convessa nè concava. Si vuole capire quali sono le strategie ottime per i due giocatori e qual è il valore del gioco.

Avendo $Z \in \mathcal{S}^2, W \in \mathcal{S}^1$, il primo vincolo dell'SDP (3.18) può essere scritto come segue.

$$\mathcal{H}^*\left(\begin{bmatrix} z_1 & z_2 \\ z_2 & z_3 \end{bmatrix}+\begin{bmatrix} 1 \\ 0 \end{bmatrix} W[1,0]-\begin{bmatrix} 0 \\ 1 \end{bmatrix} W[0,1]\right) = [\gamma,0,0]-\begin{bmatrix} 0 & -1 & 0 \\ 0 & 5 & -2 \\ -2 & 0 & 0 \end{bmatrix}\begin{bmatrix} \nu_0 \\ \nu_1 \\ \nu_2 \end{bmatrix}$$

Ovvero, si riconduce alla seguente espressione.

$$\mathcal{H}^*\left(\begin{bmatrix} z_1+W & z_2 \\ z_2 & z_3-W \end{bmatrix}\right) = \begin{bmatrix} \gamma+\nu_1 \\ 2\nu_2-5\nu_1 \\ 2\nu_0 \end{bmatrix}$$

Si ha quindi il seguente problema di programmazione semidefinita.

$$\min_{\nu,\nu_{m+1},\gamma,Z,W} \gamma, \quad s.t. \begin{cases} \begin{bmatrix} z_1+W \\ 2z_2 \\ z_3-W \end{bmatrix} = \begin{bmatrix} \gamma+\nu_1 \\ 2\nu_2-5\nu_1 \\ 2\nu_0 \end{bmatrix} \\ 1-\nu_2 \succeq 0 \\ \begin{bmatrix} \nu_0 & \nu_1 \\ \nu_1 & \nu_2 \end{bmatrix} \succeq 0 \\ \nu_0 = 1 \\ \begin{bmatrix} \nu_0 & \nu_1 \\ \nu_1 & \nu_2 \end{bmatrix} + \begin{bmatrix} \nu_1 & \nu_2 \\ \nu_2 & \nu_3 \end{bmatrix} \succeq 0 \\ \begin{bmatrix} \nu_0 & \nu_1 \\ \nu_1 & \nu_2 \end{bmatrix} - \begin{bmatrix} \nu_1 & \nu_2 \\ \nu_2 & \nu_3 \end{bmatrix} \succeq 0 \\ Z,W \succeq 0 \end{cases} \tag{3.20}$$

Dopo aver risolto l'SDP associato a tale problema e la procedura di ricostruzione del supporto, si ottengono i seguenti valori.

$$\gamma=-0.48, \;\; \nu_1=0.56, \;\; \nu_2=1, \;\; \nu_3=0.56, \;\; Z=\begin{bmatrix} 0.08 & -0.4 \\ -0.4 & 2 \end{bmatrix}, \;\; W=0.$$

Tali valori corrispondono alla strategia mista in cui il giocatore 2 sceglie $a_2=1$ con probabilità 0.78, e $a_2=-1$ con probabilità 0.22. Risolvendo il problema duale, si ottiene la strategia per il giocatore 1, il quale sceglierà l'azione $a_1=0.2$ con probabilità 1.

Capitolo 4

Giochi Polinomiali Stocastici con Single Controller

4.1 Introduzione

Nella Sezione 2.5 sono stati introdotti i giochi stocastici, un'importante classe di problemi che generalizza i processi di decisione di Markov combinandoli alla Teoria dei Giochi. Nel capitolo precedente, invece, sono stati introdotti i giochi polinomiali in forma normale, in cui i giocatori hanno accesso ad un numero infinito di strategie pure. È quindi naturale domandarsi se le tecniche utilizzate nei giochi polinomiali in forma normale possano essere estese ai giochi stocastici con infinite strategie pure. In questo capitolo viene presentata una classe di giochi stocastici polinomiali, in cui le transizioni sono "governate" da un unico giocatore. Nella Sezione 4.2 viene presentata questa classe di giochi, introducendo anche il concetto di *strategie d'equilibrio* per giochi di questo tipo e di *vettore dei valori*. Nella Sezione 4.3 viene discusso il calcolo dell'equilibrio e del vettore dei valori per giochi di questa classe, in cui però ogni giocatore ha accesso ad un numero finito di strategie pure tra cui scegliere. Nella Sezione 4.4, invece, viene mostrato come calcolare le strategie d'equilibrio per giochi di questa classe in cui i giocatori hanno accesso ad un numero infinito di strategie pure. Infine, nella Sezione 4.5 viene presentato un esempio di soluzione di un gioco di questa classe.

4.2 I Giochi Polinomiali Stocastici con Single Controller

Nel 1981 Parthasarathy e Raghavan [28] presentarono la classe dei giochi stocastici a Single Controller, brevemente introdotti nella Sezione 2.5, dimostrando che tale classe di giochi possiede la proprietà di Orderfield. Da allora si sono svolte molte ricerche su questa classe di giochi, sia per studiarne le principali proprietà, come ad esempio in [17] e [46], sia da un punto di vista algoritmico, come ad esempio in [31], sia studiandone alcune particolari estensioni, come ad esempio in [44] e [45]. In questo capitolo viene presentata un'estensione di questa classe di giochi al caso in cui i giocatori possono scegliere tra infinite strategie pure, e la funzione di payoff è polinomiale. I risultati presentati in questa sezione derivano principalmente dai risultati ottenuti da Shah, Parikshit e Parrilo, in [35].

4.2.1 Descrizione del problema

Si consideri un gioco stocastico $G = (\mathcal{S}, N, A, P, r)$, come definito nella Definizione 23, con la proprietà di Single Controller, definita nella Definizione 29. Questo significa che la probabilità di transizione nello stato s' condizionata dallo stato corrente s dipende solamente dalla coppia (s, s') e dall'azione del giocatore 1 per ogni coppia di stati (s, s'). Questa probabilità è quindi indipendente dall'azione del giocatore 2. Come è stato anticipato, la funzione di payoff sarà polinomiale nelle variabili a_1 e a_2, che rappresentano le azioni scelte rispettivamente dal giocatore 1 e 2 e che sono valori reali che appartengono agli spazi delle azioni A_1 e A_2. Per semplicità, si consideri il caso in cui $A_1 = A_2 = [0, 1] \in \mathbb{R}$. I risultati si possono facilmente generalizzare al caso in cui gli spazi delle strategie sono unioni finite di intervalli arbitrari dell'asse reale. Per semplicità, si assume anche che gli spazi delle azioni siano uguali per ogni stato; anche questa assunzione può poi essere facilmente generalizzata. Il numero di stati $|\mathcal{S}| = S$, invece, rimane ancora finito. In analogia quindi con l'equazione del payoff (3.1) presentata nel capitolo precedente, si assume che la funzione di payoff sia polinomiale nelle variabili a_1 e a_2 con coefficienti reali:

$$r(s, a_1, a_2) = \sum_{i=0}^{n_s} \sum_{j=0}^{m_s} r_{ij}(s)\, {a_1}^i\, {a_2}^j. \tag{4.1}$$

Infine, si assume che la probabilità di transizione $p(s', s, a_1)$ (indipendente dall'azione del giocatore 2) sia anch'essa polinomiale nell'azione a_1 con coefficienti reali:

$$p(s', s, a_1) = \sum_{i=0}^{d_{ss'}} p_i(s', s)\, {a_1}^i. \tag{4.2}$$

Per chiarire l'idea, si consideri l'esempio di un gioco stocastico a due stati, ovvero con $\mathcal{S} = \{1, 2\}$. Gli spazi delle azioni dei due giocatori sono $A_1 = A_2 = [0, 1]$. La funzione di payoff nello stato 1 è $r(1, a_1, a_2) = r_1(a_1, a_2)$, e la funzione di payoff nello stato 2 è data da $r(2, a_1, a_2) = r_2(a_1, a_2)$. Si assume che entrambe siano polinomiali in a_1 ed a_2. La matrice di probabilità di transizione è la seguente:

$$P = \begin{bmatrix} p_{11}(a_1, a_2) & p_{12}(a_1, a_2) \\ p_{21}(a_1, a_2) & p_{22}(a_1, a_2) \end{bmatrix}.$$

Si supponga che ogni elemento della matrice sia un polinomio in a_1. Questo gioco stocastico può essere rappresentato graficamente come mostrato in Figura 4.1: le funzioni di payoffs associate agli stati sono denotate da r_1 e r_2, mentre i lati sono marcati dalle corrispondenti probabilità di transizione negli stati.

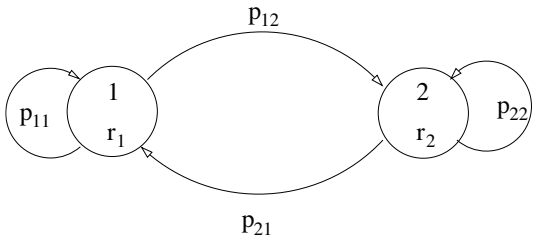

Figura 4.1: Esempio di gioco stocastico con Single Controller.

Si tornerà a questo esempio nella Sezione 4.5, quando verrà studiato come calcolare i valori degli stati e le strategie per entrambi i giocatori.

In questa classe di giochi il processo di decisione opera lungo un orizzonte infinito, quindi risulta naturale restringere l'attenzione alle strategie stazionarie per ogni giocatore, ovvero alle strategie che dipendono solo dallo stato del processo e non dal tempo. Inoltre, fintanto che i processi riguardano due decisori avversari, risulta anche naturale considerare le strategie miste invece delle strategie pure, in modo da ricostruire la nozione di equilibrio minimax, così come mostrato nella Sezione 4.2.2. Una strategia mista per il giocatore 1 è quindi l'insieme finito di misure di probabilità $\mu = [\mu(1), ..., \mu(S)]$, supportate

sull'insieme delle azioni A_1. Ogni misura di probabilità corrisponde ad una strategia mista per il giocatore 1 in un particolare stato: per esempio $\mu(k)$ corrisponde alla strategia mista che il giocatore 1 utilizzerà quando sarà nello stato k. Allo stesso modo, la strategia del giocatore 2 è rappresentata da $\nu = [\nu(1), ..., \nu(S)]$. (Due parole sulla notazione: in questo capitolo gli indici in parentesi saranno usati per denotare lo stato. Le lettere in grassetto saranno utilizzate per indicare le vettorizzazioni rispetto allo stato, ovvero collezioni di oggetti corrispondenti a stati differenti in un vettore con alla posizione i^{esima} l'elemento corrispondente allo stato i. Le lettere greche ξ, μ, ν saranno utilizzate per denotare le misure. I pedici su queste lettere greche saranno utilizzati per denotare i momenti (di vario ordine) delle misure. Una barra sulla lettera greca indica una sequenza (finita) di momenti (la lunghezza della sequenza sarà chiara dal contesto). Per esempio $\xi_j(i)$ denota il j^{esimo} momento della misura ξ corrispondente allo stato i, mentre $\bar{\xi}(i) = [\xi_0(i), ..., \xi_n(i)]$).

Una strategia μ porta ad una matrice di probabilità di transizione $P(\mu)$ tale che $P_{ss'}(\mu) = \int_{A_s} p(s', s, a_1)\, d\mu(s)$. Così, una volta che il giocatore 1 fissa una certa strategia μ, la matrice di probabilità di transizione è fissata, e può essere ottenuta integrando ogni elemento della matrice rispetto alla misura μ. (Fintanto che gli elementi sono polinomi, durante l'integrazione questi elementi dipenderanno in modo affine dai momenti $\mu(s)$). Date le strategie μ e ν, il payoff atteso immediato del giocatore 1 in un qualche stato s è dato dalla seguente espressione.

$$r(s, \mu(s), \nu(s)) = \int_{A_1} \int_{A_2} r(s, a_1, a_2)\, d\mu(s) d\nu(s)$$

Fissato uno stato iniziale s ed una coppia di strategie $\mu(s)$ e $\nu(s)$, il reward collezionato lungo l'orizzonte infinito partendo dallo stato s, ovvero $v_\beta(s, \mu(s), \nu(s))$, è dato dal seguente sistema di equazioni:

$$\begin{aligned} v_\beta\big(s, \mu(s), \nu(s)\big) \;&=\; r\big(s, \mu(s), \nu(s)\big) \;+ \\ &+ \beta \sum_{s' \in S} \left(\int_{A_1} p(s', s, a_1)\, d\mu(s) \right) v_\beta\big(s', \mu(s'), \nu(s')\big) \quad \forall\, s \in \mathcal{S}, \end{aligned}$$

dove β è il fattore di discount. Vettorizzando $v_\beta\big(s, \mu(s), \nu(s)\big)$, si ottiene:

$$\mathbf{v}_\beta(\mu, \nu) \;=\; (I - \beta P(\mu))^{-1} \mathbf{r}(\mu, \nu),$$

dove $\mathbf{r}(\mu, \nu) \;=\; [r(1, \mu(1), \nu(1)), ..., r(S, \mu(S), \nu(S))] \;\in\; \mathbb{R}^S$.

4.2.2 Strategie di Equilibrio

Nel 1953 Shapley [36] generalizzò la nozione di equilibrio di Nash ai giochi stocastici, definendo la nozione di equilibrio stazionario. Per questo motivo, quando ci si concentra sui giochi stocastici e quindi si considerano gli equilibri stazionari, invece di avere degli unici valori (come nei giochi in forma normale), si ha un unico "vettore di valori". Questo vettore è indicizzato dallo stato e l'i^{esimo} componente è interpretato come il valore che il giocatore 1 si aspetta di ricevere all'equilibrio (lungo il processo infinito discounted), condizionato dal fatto che il gioco inizi nello stato i. Ovviamente, stati differenti possono essere favorevoli a giocatori differenti. Fintanto che le azioni condizionano sia il payoff sia le transizioni, i giocatori devono bilanciare le proprie strategie in modo da ricevere dei buoni payoffs in ogni particolare stato mantenendo delle transizioni di stato favorevoli. Vengono ora definiti i concetti di strategie d'equilibrio e di vettore del gioco per questo tipo di giochi.

Definizione 36 (**Strategie di Equilibrio e Vettore dei Valori).** Due vettori di strategie miste (indicizzati dallo stato) μ^0 e ν^0, che soddisfano la proprietà del punto di sella:

$$\mathbf{v}_\beta(\mu, \nu^0) \ \leq \ \mathbf{v}_\beta(\mu^0, \nu^0) \ \leq \ \mathbf{v}_\beta(\mu^0, \nu),$$

per ogni vettore di strategie miste μ, ν, sono chiamati *strategie d'equilibrio.* Il corrispondente vettore $\mathbf{v}_\beta(\mu^0, \nu^0)$ è chiamato *vettore dei valori* del gioco.

Si potrebbe notare che $\mathbf{v}_\beta(\mu, \nu)$ è un vettore in $\mathbb{R}^S$ indicizzato dallo stato iniziale del processo di Markov. Poiché la disuguaglianza appena introdotta è una disuguaglianza di vettori va interpretata componente per componente. Più precisamente, se $\mathcal{A}$ è lo spazio delle azioni, sia $\Delta(\mathcal{A})$ lo spazio delle misure di probabilità supportate su $\mathcal{A}$. Allora la funzione v_β è una funzione della forma:

$$v_\beta : \prod_{i=1}^{S} \Delta(\mathcal{A}) \ \times \prod_{i=1}^{S} \Delta(\mathcal{A}) \to \mathbb{R}^S,$$

e le strategie d'equilibrio corrispondono ai punti di sella di questa funzione. Per ogni stato si ha quindi una coppia di strategie miste, una per giocatore: tali misure di probabilità sono indipendenti tra stato a stato, e sono indipendenti tra i giocatori.

In [36] Shapley mostrò che gli equilibri stazionari esistono sempre (e che i corrispondenti vettori di valori sono unici) per giochi stocastici a due giocatori e somma zero, con stati finiti ed azioni finite. In [35] Shah, Parikshit e Parrilo discutono i temi dell'esistenza e unicità di equilibri stazionari, provando che per ogni gioco stocastico a due giocatori e somma zero, con uno spazio degli stati finito, uno spazio di strategie infinito, e payoffs polinomiali, gli equilibri stazionari esistono sempre, e che il vettore dei valori è unico. Questo risultato è indipendente dalla condizione di Single Controller.

Inoltre, sempre in [35] Shah, Parikshit e Parrilo mostrano un semplice algoritmo per calcolare gli equilibri in ogni gioco di questo tipo. Tale algoritmo è analogo alla *policy-iteration* nella programmazione dinamica e consiste nel risolvere una sequenza di semplici giochi (non stocastici) nei quali i vettori dei valori convergono ai valori del vettore ottimo. Ad ogni iterazione è necessario risolvere un gioco polinomiale in forma normale (che può essere fatto risolvendo un singolo problema di programmazione semidefinita), e risolvendo una sequenza di tali problemi si ottiene una soluzione vicina al vettore dei valori ottimo reale. In ogni caso, la velocità di convergenza di tale algoritmo lo rende poco utilizzabile.

4.3 Caso con Spazi di Strategie Finiti

Viene ora mostrato come calcolare le soluzioni di questo tipo di giochi iniziando dal caso in cui ogni giocatore, per ogni stato, ha un numero finito di strategie pure tra cui scegliere. Viene introdotto prima questo caso per permettere un confronto con il caso di spazi delle azioni infiniti, e per poterne capire meglio le analogie. Una trattazione dettagliata di questo caso la si può trovare in [6] ed in [31].

Quando si ha un numero finito di strategie pure e viene mantenuta la condizione di Single Controller è possibile calcolare una soluzione minimax attraverso la programmazione lineare: viene quindi mostrato in questa sezione come utilizzare la programmazione lineare per risolvere giochi di questa classe. Nella prossima sezione, partendo concettualmente da questo problema di programmazione lineare, verrà mostrato un problema di ottimizzazione dimensionalmente infinito per il caso in cui ogni giocatore può scegliere tra un numero infinito di strategie pure.

Per semplicita, si assuma ancora che gli insiemi di strategie pure disponibili ad ogni giocatore in ogni stato siano identici, ovvero che $A_1 = A_2 = \{1, ..., m\}$, che lo spazio degli stati sia $\mathcal{S} = \{1, ..., S\}$, e che la matrice di probabilità di transizione rispetti la Definizione 29 di gioco stocastico a single controller. Si definisce inoltre con β il fattore di discount. Una strategia mista per il giocatore 1 è una funzione $f : \mathcal{S} \times A_1 \to [0, 1]$ soggetta al vincolo di normalizzazione $\sum_{a_1} f(s, a_1) = 1$ per ogni stato $s \in \mathcal{S}$ (in modo tale che $f(s) = [f(s, 1), ..., f(s, m)]$ diventi una distribuzione di probabilità sullo spazio di strategie A_1). Similarmente, la strategia mista per il giocatore 2 in un particolare stato s è data da $g(s) = [g(s, 1), ..., g(s, m)]$. La collezione di strategie miste (indicizzate dagli stati) è denotata da $\mathbf{f} = [f(1), ..., f(S)]$ (e, rispettivamente, $\mathbf{g} = [g(1), ..., g(S)]$). Una strategia $\mathbf{f}$ conduce ad una matrice di probabilità di transizione $P(\mathbf{f}) = \sum_{a_1 \in A_1} p(s', s, a_1) f(s, a_1)$. Si consideri nuovamente un processo β-discounted su un orizzonte infinito. Date le strategie $\mathbf{f}$ e $\mathbf{g}$, i payoff immediati del giocatore 1 in un qualche stato s sono dati da:

$$r\big(s, f(s), g(s)\big) = \sum_{a_1 \in A_1, a_2 \in A_2} r(s, a_1, a_2) f(s, a_1) g(s, a_2).$$

Il reward collezionato sull'orizzonte infinito partendo dallo stato s, $v_\beta\big(s, f(s), g(s)\big)$, è dato dal sistema di equazioni:

$$\begin{aligned} v_\beta\big(s, f(s), g(s)\big) \;&=\; r\big(s, f(s), g(s)\big) \;+ \\ &+ \beta \sum_{s' \in S} \Bigg(\sum_{a_1 \in A_1} p(s', s, a_1) f(s, a_1) \Bigg) v_\beta\big(s', f(s'), g(s')\big) \quad \forall\, s \in \mathcal{S}, \end{aligned}$$

Si ha quindi:

$$\mathbf{v}_\beta(\mathbf{f}, \mathbf{g}) \;=\; (I - \beta P(\mathbf{f}))^{-1} \mathbf{r}(\mathbf{f}, \mathbf{g}),$$

dove $\mathbf{r}(\mathbf{f}, \mathbf{g}) \;=\; [r(1, f(1), g(1)), ..., r(S, f(S), g(S))] \in \mathbb{R}^S$. L'obiettivo è perciò quello di trovare le strategie di equilibrio $\mathbf{f}^0$ e $\mathbf{g}^0$ che soddisfino la proprietà di equilibrio di Nash:

$$\mathbf{v}_\beta(\mathbf{f}, \mathbf{g}^0) \;\leq\; \mathbf{v}_\beta(\mathbf{f}^0, \mathbf{g}^0) \;\leq\; \mathbf{v}_\beta(\mathbf{f}^0, \mathbf{g}) \tag{4.3}$$

per ogni strategia mista $\mathbf{f}$ e $\mathbf{g}$.

Si consideri allora il seguente problema di programmazione lineare: come sarà poi chiaro dal Teorema 15, tale problema di ottimizzazione permette di calcolare il valore di ogni stato e la strategia ottima per il giocatore 2, ovvero

il giocatore che vuole minimizzare il payoff e le cui azioni non influiscono sulle transizioni di stato.

$$\min_{g(s,a_2),v(s)} \sum_{s=1}^{S} v(s), \quad \textit{subject to}$$

$$\begin{aligned} & v(s) \;\geq\; \textstyle\sum_{a_2 \in A_2} r(s,a_1,a_2)\, g(s,a_2) + \\ & \qquad + \beta \textstyle\sum_{s'=1}^{S} p(s',s,a_1) v(s') \quad \forall\, s \in \mathcal{S},\, a_1 \in A_1 \end{aligned} \tag{LP1}$$

$$\textstyle\sum_{a_2 \in A_2} g(s,a_2) = 1, \quad \forall\, s \in \mathcal{S}$$

$$g(s,a_2) \;\geq\; 0 \quad \forall\, s \in \mathcal{S},\, a_2 \in A_2$$

Si consideri anche il problema di programmazione lineare duale del precedente. Come sarà poi chiaro, sempre dal Teorema 15, tale problema di ottimizzazione permette di calcolare il valore di ogni stato e la strategia ottima per il giocatore 1, ovvero il giocatore che vuole massimizzare il payoff e le cui azioni influiscono sulle transizioni di stato.

$$\max_{f(x,a_1),z(s)} \sum_{s=1}^{S} z(s), \quad \textit{subject to}$$

$$\begin{aligned} & \textstyle\sum_{s=1}^{S} \sum_{a_1 \in A_1} \left[\delta(s,s') - \beta p(s',s,a_1)\right] x(s,a_1) = 1 \quad \forall\, s' \in \mathcal{S} \\ & z(s) \;\leq\; \textstyle\sum_{a_1 \in A_1} x(s,a_1)\, r(s,a_1,a_2) \quad \forall\, s \in \mathcal{S},\, a_2 \in A_2 \\ & x(s,a_1) \;\geq\; 0 \quad \forall\, s \in \mathcal{S},\, a_1 \in A_1 \end{aligned} \tag{DP1}$$

Teorema 15 ([6])**.** Si consideri la coppia primale-duale di problemi di programmazione lineare (LP1) ed (DP1). Sia p^* il valore ottimo di (LP1), e sia d^* il valore ottimo di (DP1). Siano $x^*(s,a_1)$ i valori ottimi delle variabili $x(s,a_1)$ ottenute in (DP1), e siano $g^*(s,a_2)$ i valori ottimi delle variabili $g(s,a_2)$ ottenute in (LP1). Siano

$$f^*(s,a_1) = \frac{x^*(s,a_1)}{\sum_{a_1} x^*(s,a_1)}$$

e $g^*(s,a_2)$ le distribuzioni di probabilità. Allora si può affermare che:

1. $p^* = d^*$

2. Sia $\mathbf{v}^* = [v^*(1), ..., v^*(S)]$ l'ottima soluzione di (LP1). Allora si ha che $\mathbf{v}^* = \mathbf{v}_\beta(\mathbf{f}^*, \mathbf{g}^*)$.

3. $\mathbf{v}_\beta(\mathbf{f}^*, \mathbf{g}^*)$ soddisfa la disuguaglianza del punto di sella (4.3).

Dimostrazione. Si veda ([6, pp.93]). □

Osservazione 1 Si può notare che l'affermazione 2 nel Teorema 15 sostiene che la soluzione dell'LP (LP1) corrisponde al reward scontato sull'orizzonte infinito ottenuto quando i giocatori 1 e 2 giocano secondo le distribuzioni $\mathbf{f}^*$ e $\mathbf{g}^*$. L'affermazione 3 sostiene infatti che queste distribuzioni sono ottime per i giocatori per la definizione di equilibrio di Nash.

Osservazione 2 Si può notare che il problema primale (LP1) ha una interpretazione naturale in termini di strategie minmax. Si considerino i vettori ammissibili $\mathbf{v}$ e $\mathbf{g}$ che soddisfano il primo insieme di disuguaglianze in (LP1): le disuguaglianze possono essere interpretate col significato di garantire che usando la strategia $\mathbf{g}$ il payoff del giocatore 2 sarà almeno $\mathbf{v}$.

4.4 Caso con Spazi di Strategie Infiniti

In questa sezione si considera il caso in cui ogni giocatore possa scegliere tra infinite azioni non numerabili. In particolare, ogni giocatore può scegliere azioni nell'intervallo $[0, 1]$. Il numero di stati $|\mathcal{S}| = S$ rimane ancora finito. La funzione di payoff $r(s, a_1, a_2)$ è come quella indicata in (4.1) per ogni $s \in \mathcal{S}$. Inoltre, poiché si assume che sia soddisfatta la condizione di Single Controller, la probabilità di transizione $p(s', s, a_1)$ è come quella indicata in (4.2). Si considera ancora il caso di gioco a due giocatori e somma zero in cui il giocatore 1 vuole massimizzare il reward atteso lungo l'orizzonte infinito. Viene quindi generalizzato il problema (LP1) a questo caso. Le variabili $\mathbf{f}$ e $\mathbf{g}$, che rappresentavano le distribuzioni sugli insiemi finiti A_1 e A_2, sono rimpiazzate dalle misure di probabilità $\mu(s)$ e $\nu(s)$. Queste misure rappresentano le strategie miste su uno spazio di azioni non numerabile. (Si ricorda che per ogni giocatore vi sono S misure, ogni misura corrispondente ad una strategia mista in un particolare stato. Per esempio $\mu(s)$ corrisponde alla strategia mista che

il giocatore 1 adotterà quando il gioco sarà nello stato s.)
I risultati presentati in questa sezione derivano principalmente dai risultati ottenuti da Shah, Parikshit e Parrilo, in [35].

4.4.1 Da Azioni Finite ed LP ad Azioni Infinite ed SDP

Viene ora mostrato che una generalizzazione del problema di programmazione lineare (LP1) a questo caso porta ad un problema di ottimizzazione, che riguarda la non negatività di un sistema di polinomi univariati con coefficienti che dipendono dai momenti delle misure μ e ν. Vi è quindi, ovviamente, una forte analogia con quanto fatto nel caso dei giochi polinomiali in forma normale, essendo i giochi stocastici una generalizzazione dei giochi in forma normale. L'interpretazione fatta in termini di strategie sicure per il giocatore 2, brevemente introdotta al termine della sezione precedente ed analizzata nel caso dei giochi polinomiali in forma normale, viene anche qui mantenuta: infatti è possibile notare una forte analogia tra il problema di ottimizzazione (3.6) ed il problema di ottimizzazione (4.4), tra poco introdotto. Viene ora mostrato il problema di ottimizzazione che generalizza il problema di programmazione lineare (LP1).

$$\min_{\nu(s),v(s)} \sum_{s=1}^{S} v(s), \quad subject\ to$$

$$\begin{aligned} (a) \quad & v(s) \ \geq \ \textstyle\int_{a_2 \in A_2} r(s,a_1,a_2)\,\nu(s,a_2)\,da_2 + \\ & \qquad + \beta \textstyle\sum_{s'=1}^{S} p(s',s,a_1) v(s') \quad \forall\, s \ \in \ \mathcal{S},\ a_1 \ \in \ A_1 \end{aligned} \tag{4.4}$$

$$(b) \quad \nu(s) \text{ é una misura supportata su } A_2 \text{ per ogni } s \ \in \ \mathcal{S}$$

Poiché si ha $\int_{A_2} r(s,a_1,a_2)\nu(s,a_2)\,da_2 \ = \ t_\nu(s,a_1)$, dove $t_\nu(s,a_1)$ è un polinomio univariato in a_1 per ogni $s \in \mathcal{S}$, fissato un vettore $\nu(s)$, il vincolo (a) è un sistema di disequazioni polinomiali. Si può notare che i coefficienti di t dipendono dalla misura ν solo attraverso un numero finito di momenti. Più concretamente, si hanno le seguenti equazioni.

$$\int_{a_2 \in A_2} r(s,a_1,a_2)\,\nu(s,a_2)\,da_2 \ = \ \int_{a_2 \in A_2} \sum_{i=0}^{n_s} \sum_{j=0}^{m_s} r_{ij}(s)\, {a_1}^i\, {a_2}^j\, \nu(s,a_2)\,da_2 \ =$$
$$= \ \sum_{i=0}^{n_s} \sum_{j=0}^{m_s} r_{ij}(s)\, {a_1}^i \int_{a_2 \in A_2} {a_2}^j\, \nu(s,a_2)\,da_2 \ = \ \sum_{i=0}^{n_s} \sum_{j=0}^{m_s} r_{ij}(s)\, \nu_j(s)\, {a_1}^i$$

Utilizzando questa osservazione, il problema di ottimizzazione (4.4) può essere riscritto nel seguente modo.

$$\min_{\nu(s),v(s)} \sum_{s=1}^{S} v(s), \quad subject\ to$$

$$\begin{aligned} (c) \quad & v(s) - \textstyle\sum_{i=0}^{n_s} \sum_{j=0}^{m_s} r_{ij}(s)\, \nu_j(s)\, {a_1}^i + \\ & - \beta \textstyle\sum_{s'=1}^{S} p(s', s, a_1) v(s') \ \in \mathcal{P}(A_1)\ \forall\, s \in \mathcal{S} \end{aligned} \tag{P'}$$

$$(d) \quad \bar{\nu}(s) \in \mathcal{M}(A_2),\ \nu_0(s) = 1,\ \forall\, s \in \mathcal{S}$$

Si può notare una certa analogia rispetto al problema di ottimizzazione (3.7) nella Sezione 3.2.2 del capitolo precedente. Come in quel caso, la formulazione del problema di ottimizzazione (P′) è astratta e non risolvibile. Anche in questo caso, per cercare di convertirla in un problema di ottimizzazione concreto che si possa risolvere, è necessaria la rappresentazione computazionalmente adatta dei due insiemi $\mathcal{P}(A_1)$ ed $\mathcal{M}(A_2)$ introdotta nella Sezione 3.3. Come è noto dalla Sezione 3.3.1, le condizioni di non negatività di un polinomio su un intervallo si applicano ai coefficienti di tale polinomio. Per questo motivo, si consideri il vincolo (c), che produce un sistema di disuguaglianze polinomiali in a_1, una disuguaglianza per ogni stato: fissato un certo stato $s \in \mathcal{S}$, si cerca di esplicitare i coefficienti del polinomio

$$t_s(a_1) = v(s) - \sum_{i=0}^{n_s} \sum_{j=0}^{m_s} r_{ij}(s)\, \nu_j(s)\, {a_1}^i - \beta \sum_{s'=1}^{S} p(s', s, a_1) v(s').$$

Sia d_s il grado della disuguaglianza per quello stato, e sia inoltre $[a_1]_{d_s} = [1, a_1, {a_1}^2, ..., {a_1}^{d_s}]^T$. Il secondo termine del polinomio $t_s(a_1)$, può essere riscritto in forma vettoriale come:

$$\sum_{i=0}^{n_s} \sum_{j=0}^{m_s} r_{ij}(s)\, \nu_j(s)\, {a_1}^i = \bar{\nu}(s)^T R(s)^T [a_1]_{d_s}. \tag{4.5}$$

dove $R(s)$ è la matrice che contiene i coefficienti del polinomio $r(s, a_1, a_2)$, e $\bar{\nu}(s) \in \mathbb{R}^{d_s+1}$ è il vettore dei primi $d_s + 1$ momenti della misura $\nu(s)$. Similarmente al caso di strategie finite, si può definire il vettore $\mathbf{v}^* = [v^*(1), ..., v^*(S)]^T$, che risulterà essere il vettore dei valori del gioco stocastico (indicizzato dallo stato). Il terzo termine del polinomio $t(a_1)$, che

dipende dalla probabilità di transizione $p(s', s, a_1)$, è anch'esso ovviamente un polinomio in a_1 ed i suoi coefficienti dipendono dai coefficienti di $p(s', s, a_1)$ e $\mathbf{v}$. Specificatamente, si ha:

$$\sum_{s'=1}^{S} p(s', s, a_1)v(s') \;=\; \mathbf{v}^T Q(s)^T [a_1]_{d_s}, \tag{4.6}$$

dove $Q(s)$ è la matrice che contiene i coefficienti di $p(s', s, a_1)$. A questo punto, poiché gli spazi delle azioni sono $A_1 = A_2 = [0, 1]$, dal Lemma 4 nella Sezione 3.3.1, è possibile concludere che il polinomio $t_s(a_1)$ è non negativo in $[0, 1]$ se e solo se esistono le matrici $Z_s \in \mathcal{S}^{d_s+1}$ e $W_s \in \mathcal{S}^{d_s}$, $Z \succeq 0$, $W \succeq 0$ tali che

$$\mathcal{H}^*(Z_s + \frac{1}{2}(L_1 W_s L_2^T + L_2 W_s L_1^T) - L_2 W_s L_2^T) = E_s\mathbf{v} - \beta Q(s)\mathbf{v} - R(s)\bar{\nu}(s) \tag{4.7}$$

dove $E_s \in \mathbb{R}^{d_s \times S}$ è la matrice formata da tutti zeri eccetto un 1 in posizione $(1, s)$.

Si cerca ora di riscrivere il vincolo (d) del problema di ottimizzazione (P′), che riguarda i momenti delle misure di probabilità. Si fissi sempre uno stato $s \in \mathcal{S}$. Grazie al Lemma 9 nella Sezione 3.3.2, è possibile concludere che $\bar{\nu}(s) \in \mathcal{M}(A_2)$ e $\nu_0(s) = 1$ con $[c, d] = [0, 1]$ se e solo se

$$\begin{gathered} {e_1}^T \bar{\nu}(s) = 1, \\ \mathcal{H}(\bar{\nu}(s)) \succeq 0, \\ \tfrac{1}{2}(L_1^T \mathcal{H}(\bar{\nu}(s)) L_2 + L_2^T \mathcal{H}(\bar{\nu}(s)) L_1) - L_2^T \mathcal{H}(\bar{\nu}(s)) L_2 \succeq 0. \end{gathered} \tag{4.8}$$

Si possono quindi rimettere insieme tutte queste condizioni, per formare il singolo problema di programmazione semidefinita positiva che concretizza il problema di ottimizzazione astratto (P′), ottenendo quindi il seguente SDP.

$$\min_{\bar{\nu}(s),v(s),Z_s,W_s} \sum_{s=1}^{S} v(s), \quad subject\ to$$

$$\begin{aligned}
(e) \quad & \mathcal{H}^*(Z_s + \tfrac{1}{2}(L_1 W_s L_2^T + L_2 W_s L_1^T) - L_2 W_s L_2^T = \\
& = E_s\mathbf{v} - R(s)\bar{\nu} - \beta Q(s)\mathbf{v}, \quad \forall\ s \in \mathcal{S} \\
(f) \quad & \mathcal{H}(\bar{\nu}(s)) \succeq 0, \quad \forall\ s \in \mathcal{S} \\
(g) \quad & \tfrac{1}{2}(L_1^T \mathcal{H}(\bar{\nu})(s) L_2 + L_2^T \mathcal{H}(\bar{\nu})(s) L_1) + \\
& - L_2^T \mathcal{H}(\bar{\nu})(s) L_2 \succeq 0, \quad \forall\ s \in \mathcal{S} \\
(h) \quad & e_1^T \bar{\nu}(s) = 1, \quad \forall\ s \in \mathcal{S} \\
(i) \quad & Z_s, W_s \succeq 0, \quad \forall\ s \in \mathcal{S}
\end{aligned} \tag{SP}$$

A questo punto, si può esprimere il seguente Lemma.

Lemma 10. Siano $A_1 = A_2 = [0, 1]$. Il problema di programmazione semidefinita (SP) risolve esattamente il problema di ottimizzazione polinomiale (P′).

Dimostrazione. La disuguaglianza polinomiale (c) ha il vettore dei coefficienti $E_s\mathbf{v} - R(s)\bar{\nu} - \beta Q(s)\mathbf{v}$, come mostrato dalle equazioni (4.5) e (4.6). La dimostrazione si ha quindi per diretta conseguenza del Lemma 4 riguardante la rappresentazione semidefinita dei polinomi non negativi su $[0, 1]$, come mostrato nell'equazione (4.7), ed il Lemma 9 riguardante la rappresentazione semidefinita delle sequenze di momenti di misure non negative supportate su $[0, 1]$, come mostrato nell'espressione (4.8). □

Osservazione La soluzione di questo problema di ottimizzazione, come sarà poi discusso nella sezione successiva, produce il vettore dei valori del gioco ed i momenti delle misure (una per ogni stato) per il giocatore 2, ovvero il giocatore che non controlla le transizioni. Come nel caso dei giochi polinomiali in forma normale rimane comunque un problema, che emergerà durante la ricostruzione delle strategie ottime dai loro momenti: a causa di questo problema, sarà necessario aggiungere dei vincoli all'SDP (SP) per ottenere, oltre al valore

del gioco ed ai momenti delle misure di probabilità (per ogni stato), anche i supporti ed i pesi delle strategie miste ottime.

Si consideri allora il problema duale per cercare le strategie del giocatore 1 che controlla le transizioni. Si consideri il vettore dei primi $d_s + 1$ momenti $\bar{\xi}(s) \in \mathbb{R}^{d_s+1}$, il problema duale è equivalente al seguente problema di programmazione semidefinita:

$$\max_{\bar{\xi}(s),\alpha(s),A_s,B_s} \sum_{s=1}^{S} \alpha(s), \quad \textit{subject to}$$

$$\begin{aligned}
&(j) \quad \mathcal{H}^*(A_s + \tfrac{1}{2}(L_1 B_s L_2^T + L_2 B_s L_1^T) - L_2 B_s L_2^T = \\
&\qquad = R(s)^T \bar{\xi}(s) - \alpha(s) e_1, \quad \forall\, s \in \mathcal{S} \\
&(k) \qquad \mathcal{H}(\bar{\xi}(s)) \succeq 0, \quad \forall\, s \in \mathcal{S} \\
&(l) \qquad \tfrac{1}{2}(L_1^T \mathcal{H}(\bar{\xi})(s) L_2 + L_2^T \mathcal{H}(\bar{\nu})(s) L_1) + \\
&\qquad - L_2^T \mathcal{H}(\bar{\xi})(s) L_2 \succeq 0, \quad \forall\, s \in \mathcal{S} \\
&\qquad \textstyle\sum_s (E_s - \beta Q(s))^T \bar{\xi}(s) = 1 \\
&(m) \qquad A_s, B_s \succeq 0, \quad \forall\, s \in \mathcal{S}
\end{aligned} \tag{SD}$$

che coincide con il seguente problema di ottimizzazione polinomiale.

$$\max_{\xi(s),\alpha(s)} \sum_{s=1}^{S} \alpha(s), \quad \textit{subject to}$$

$$\begin{aligned}
&(n) \quad \textstyle\sum_{i,j} r_{ij}(s) \xi_i(s) {a_2}^j - \alpha(s) \geq 0 \quad \forall\, a_2 \in A_2, \forall\, s \in \mathcal{S} \\
&(o) \quad \bar{\xi}(s) \in \mathcal{M}(A_2) \quad \forall\, s \in \mathcal{S} \\
&(p) \quad \textstyle\sum_s \int_{A_1} (\delta(s, s') - \beta p(s', s, a_1))\, d\xi(s) = 1 \quad \forall\, s' \in \mathcal{S}
\end{aligned} \tag{D$'$}$$

Lemma 11. Il problema di programmazione definita (SD) è equivalente al problema di ottimizzazione polinomiale (D$'$).

Dimostrazione. La dimostrazione si ha ancora per conseguenza diretta del Lemma 4 e del Lemma 9 in modo del tutto analogo a quanto fatto per il Lemma 10. □

Osservazione 1 È importante notare che nel problema duale la sequenza dei momenti non necessariamente corrisponde ad una misura di probabilità. Infatti, per convertirla in una misura di probabilità è necessario normalizzare la misura, dividendo ogni momento per il momento di ordine zero. A seguito della normalizzazione, si hanno le sequenze dei momenti delle misure ottime (una per ogni stato) per il giocatore 1. Allo stesso modo, il vettore dei valori del gioco richiede una qualche normalizzazione per coincidere con il vettore ottimo del gioco reale.

Osservazione 2 La soluzione di questo problema di ottimizzazione, come sarà poi discusso nella sezione successiva, produce il vettore dei valori del gioco ed i momenti delle misure (una per ogni stato) per il giocatore 2 (a seguito della normalizzazione sia dei momenti sia dei valori degli stati). Come per l'SDP (SP), rimane comunque un problema che emergerà durante la ricostruzione delle strategie ottime dai loro momenti: a causa di questo problema, sarà necessario aggiungere dei vincoli all'SDP (SD) per ottenere, oltre al valore del gioco ed ai momenti delle misure di probabilità (per ogni stato), anche i supporti ed i pesi delle strategie miste ottime.

Si conclude questa sezione con il lemma che formalizza il collegamento tra i problemi di ottimizzazione polinomiale (P′) e (D′), e quindi, per il Lemma 10 ed il Lemma 11, anche il collegamento tra l'SDP (SP) e l'SDP (SD).

Lemma 12. I problemi di ottimizzazione polinomiale (P′) e (D′) sono tra loro in rapporto di dualità forte.

Dimostrazione. La dimostrazione si può trovare in [35]. □

4.4.2 Ottimalità della Soluzione

Nella sezione precedente è stato mostrato che il problema (P′) può essere ridotto al sistema di programmazione semidefinita (SP). In questa sezione, invece, viene mostrato che la soluzione del problema (SP) è effettivamente la soluzione d'equilibrio desiderata. Per arrivare a tale risultato, si introduce inizialmente il seguente lemma.

Lemma 13. Siano $\bar{\nu}^*(s)$ e $\bar{\xi}^*(s)$ le sequenze di momenti ottime rispettivamente per (P′) e (D′). Siano $\nu^*(s)$ e $\xi^*(s)$ le corrispondenti misure supportate

rispettivamente su A_1 e A_2. Si ha quindi il seguente risultato complementare per l'ottimo di (P′) e (D′):

$$\begin{aligned} v^*(s)\int_{A_1} d\xi^*(s) \;=\; \int_{A_2}\int_{A_1} r(s,a_1,a_2)\, d\xi^*(s)\, d\nu^*(s) + \\ +\beta \sum_{s'} V^*(s') \int_{A_1} p(s',s,a_1)\, d\xi^*(s) \quad \forall\, s \in \mathcal{S}, \end{aligned} \tag{4.9}$$

$$\alpha^*(s)\int_{A_2} d\nu^*(s) \;=\; \int_{A_2}\int_{A_1} r(s,a_1,a_2)\, d\xi^*(s)\, d\nu^*(s) \quad \forall\, s \in \mathcal{S}. \tag{4.10}$$

Dimostrazione. Il risultato segue dalla dualità forte dell'equivalente rappresentazione semidefinita della coppia primale-duale (P′) e (D′). La funzione Lagrangiana per (P′) è data da:

$$\begin{aligned} \mathcal{L}(\xi,\alpha) \;=\; \inf_{\mathbf{v},\nu}\{\textstyle\sum_{s=1}^{S} v(s) - \int_{A_1}[v(s) - \int_{A_2} r(s,a_1,a_2)\, d\nu(s) + \\ -\beta \sum_{s'} v(s')p(s',s,a_1)]\, d\xi(s) + \sum_s \alpha(s)(1-\nu_0(s))\}. \end{aligned}$$

$\mathcal{L}(\xi,\alpha)$ deve soddisfare la dualità debole, ovvero $d^* \le p^*$. All'ottimo, si ha $p^* = \sum_s v^*(s)$ per qualche vettore $\mathbf{v}^*$. In ogni caso, la dualità forte si mantiene, ovvero si ha $p^* = d^*$. Questo forza la prima relazione complementare. La seconda relazione, invece, è ottenuta similarmente ma considerando la Lagrangiana del problema duale. □

È ora possibile mostrare che la soluzione del problema (P′) è effettivamente la soluzione d'equilibrio desiderata, con il seguente teorema.

Teorema 16. Sia p^* il valore ottimo di (P′), e d^* sia il valore ottimo di (D′). Siano $\nu^*(s)$ e $\xi^*(s)$ le misure ottime ricostruite in (P′) e (D′). Sia

$$\mu^*(s) \;=\; \frac{\xi^*(s)}{\int_{A_1} d\xi^*(s)}$$

in modo che μ^* sia la versione normalizzata di ξ^* (ovvero μ^* è una misura di probabilità). Sia $\mathbf{v}^*$ il vettore ottenuto dalla soluzione ottima di (P′). La soluzione ottima della coppia primale-duale (P′) e (D′) soddisfa le seguenti affermazioni:

1. $p^* = d^*$.
2. $\mathbf{v}^* = v_\beta(\mu^*, \nu^*)$.

3. $\mathbf{v}_\beta(\mu^*, \nu^*)$ soddisfa la seguente disuguaglianza del punto di sella:

$$\mathbf{v}_\beta(\mu, \nu^*) \leq \mathbf{v}_\beta(\mu^*, \nu^*) \leq \mathbf{v}_\beta(\mu^*, \nu)$$

per ogni coppia di strategie miste (μ, ν).

Dimostrazione.

1. Segue dalla dualità forte della coppia primale e duale (P′) e (D′).

2. Usando il Lemma 13 equazione (4.9) in forma normalizzata (ovvero, dividendo ogni elemento per il momento di ordine zero della misura $\xi(s)$, $\xi_0^*(s)$) si ottiene

$$\begin{aligned} v^*(s) = \textstyle\int_{A_2} \int_{A_1} r(s, a_1, a_2)\, d\mu^*(s)\, d\nu^*(s) + \\ \beta \textstyle\sum_{s'} v^*(s') \int_{A_1} p(s', s, a_1)\, d\mu^*(s) \quad \forall\, s \in \mathcal{S} \end{aligned}$$

Semplificando e vettorizzando si ottiene quindi

$$\mathbf{v}^* = r(\mu^*, \nu^*) + \beta P(\mu^*)\mathbf{v}^*.$$

Utilizzando l'equazione di Bellman, o semplicemente iterando questa equazione, è facile vedere che $\mathbf{v}^* = \mathbf{v}_\beta(\mu^*, \nu^*)$.

3. Si consideri la disuguaglianza (c) quando si è all'ottimo. Si ha per ogni stato:

$$\begin{aligned} v^*(s) \;\geq\; \textstyle\int_{a_2 \in A_2} r(s, a_1, a_2)\, d\nu^*(s) + \\ \beta \textstyle\sum_{s'=s}^{S} p(s', s, a_1)\, v^*(s'). \end{aligned}$$

Integrando rispetto a qualche arbitraria misura di probabilità $\mu(s)$ (con supporto su A_1), si ottiene:

$$\begin{aligned} v^*(s) \;\geq\; \textstyle\int_{A_2} \int_{A_1} r(s, a_1, a_2)\, d\mu^*(s)\, d\nu^*(s) + \\ \beta \textstyle\sum_{s'=s}^{S} \int_{A_1} p(s', s, a_1)\, v^*(s')\, d\mu(s), \end{aligned}$$

da cui si ottiene:

$$\begin{aligned} v^*(s) \;\geq\; r(s, \mu(s), \nu^*(s)) + \\ \beta \textstyle\sum_{s'=s}^{S} \int_{A_1} p(s', s, a_1)\, v^*(s')\, d\mu(s). \end{aligned}$$

Integrando questa equazione si ottiene $\mathbf{v}_\beta(\mu^*, \nu^*) - \mathbf{v}^* \geq \mathbf{v}_\beta(\mu, \nu^*)$ per ogni strategia μ. Questo completa un lato della disuguaglianza del punto

di sella.

Utilizzando la versione normalizzata dell'equazione (4.10), si ottiene:

$$\begin{aligned}\tfrac{\alpha^*(s)}{\xi_0^*(s)} &= \int_{A_2}\int_{A_1} r(s,a_1,a_2)\,d\mu^*(s)\,d\nu^*(s)\\ &= r(s,\mu^*(s),\nu^*(s)).\end{aligned}$$

Integrando la disuguaglianza (n) nel problema (D′) rispetto ad una misura di probabilità arbitraria $\nu(s)$ con supporto su A_2, si ottiene

$$\frac{\alpha^*(s)}{\xi_0^*(s)} = r\big(s,\mu^*(s),\nu(s)\big).$$

Quindi $r(s,\mu^*(s),\nu^*(s)) \leq r(s,\mu^*(s),\nu(s))$ per ogni s. Moltiplicando per $(I-\beta P(\mu^*))^{-1}$, si ottiene $\mathbf{v}_\beta(\mu^*,\nu^*) \leq \mathbf{v}_\beta(\mu^*,\nu)$. Questo completa l'altro lato della disuguaglianza del punto di sella.

□

4.4.3 Ottenere le Misure di Probabilità

Le variabili di decisione dei problemi SOS/SDP presentati nella Sezione 4.4.1 erano i momenti delle strategie miste. Come nel caso dei giochi polinomiali in forma normale, le corrispondenti misure possono essere ricostruite dalle soluzioni ottime degli SDP primali e duali, in particolare dai vettori $\bar{\nu}(s)$ e $\bar{\xi}(s)$. Come già indicato, però, i momenti nel vettore $\bar{\xi}(s)$ richiedono una normalizzazione: viene quindi mostrato come realizzarla.

Si fissi un certo stato $s \in \mathcal{S}$. Si consideri la sequenza di momenti $(\xi_0(s),\xi_1(s),...,\xi_{m_s+1}(s))$. La sequenza normalizzata corrispondente a tale sequenza è ottenuta dividendo ogni elemento della sequenza per il momento di ordine zero: si ottiene quindi la sequenza normalizzata

$$\mu(s) = \left(\frac{\xi_0(s)}{\xi_0(s)}, \frac{\xi_1(s)}{\xi_0(s)}, \frac{\xi_2(s)}{\xi_0(s)}, ..., \frac{\xi_{m_s+1}(s)}{\xi_0(s)}\right).$$

Ora che si ha una coppia di sequenze dei momenti $\bar{\nu}(s)$ e $\bar{\mu}(s)$ per ogni stato, per ricostruire le misure di probabilità dei giocatori in ogni stato è necessario utilizzare la procedura indicata nella Sezione 3.3.2. Come nel caso dei giochi polinomiali in forma normale c'è però un problema: la procedura di ricostruzione richiede la presenza di m_s+2 e n_s+2 momenti per ogni stato s, mentre risolvendo i problemi (P′) e (D′) ne sono stati

calcolati $m_s + 1$ e $n_s + 1$. Per poter ricostruire la strategia ottima per i giocatori in ogni stato è quindi necessario calcolare anche la coppia di momenti $\mu_{m_s+1}(s)$ e $\nu_{n_s+1}(s)$ "mancanti" per ogni stato s, vincolandoli in qualche modo ai momenti di ordine inferiore, oltre che all'intervallo su cui è definita la rispettiva misura. Considerando unicamente il giocatore 2 è quindi possibile utilizzare il Lemma 7, ed aggiungere per ogni stato i due vincoli di programmazione semidefinita (che saranno indicati con (e) ed (f)) al problema (SP). Aggiungendo questi vincoli è possibile ottenere per ogni stato s (oltre al valore del gioco ed ai primi $(\nu_0(s), \nu_1(s), ..., \nu_{m_s}(s))$ momenti) anche i momenti $\nu_{m_s+1}(s)$, sufficienti alla ricostruzione della strategia mista ottima del giocatore 2. In questo modo, si ottiene il seguente problema di programmazione semidefinita.

$$\min_{\bar{\nu}(s),\nu_{m_s+1}(s),v(s),Z_s,W_s} \sum_{s=1}^{S} v(s), \quad \textit{subject to}$$

$$(a) \qquad \mathcal{H}^*(Z_s + \tfrac{1}{2}(L_1 W_s L_2^T + L_2 W_s L_1^T) - L_2 W_s L_2^T = \\ = E_s \mathbf{v} - R(s)\bar{\nu} - \beta Q(s)\mathbf{v}, \quad \forall\ s \in \mathcal{S}$$

$$(b) \qquad \mathcal{H}(\bar{\nu}(s)) \succeq 0, \quad \forall\ s \in \mathcal{S}$$

$$(c) \qquad \tfrac{1}{2}(L_1^T \mathcal{H}(\bar{\nu})(s) L_2 + L_2^T \mathcal{H}(\bar{\nu})(s) L_1) + \\ - L_2^T \mathcal{H}(\bar{\nu})(s) L_2 \succeq 0, \quad \forall\ s \in \mathcal{S}$$

$$(d) \qquad e_1^T \bar{\nu}(s) = 1, \quad \forall\ s \in \mathcal{S}$$

$$(e) \qquad \mathcal{H}([\nu_1(s),\ \nu_2(s),\ ...,\ \nu_{m_s+1}(s)]^T) \succeq 0, \quad \forall\ s \in \mathcal{S}$$

$$(f) \quad \mathcal{H}([\nu_0(s),\ \nu_1(s),\ ...,\ \nu_{m_s}(s)]^T) - \mathcal{H}([\nu_1(s),\ \nu_2(s),\ ...,\ \nu_{m_s+1}(s)]^T) \succeq 0, \\ \forall\ s \in \mathcal{S}$$

$$(g) \qquad Z_s, W_s \succeq 0, \quad \forall\ s \in \mathcal{S}$$

(SP′)

Svolgendo lo stesso procedimento anche per il problema duale (SD), si

ottiene il seguente problema di programmazione semidefinita che completa la coppia primale-duale.

$$\max_{\bar{\xi}(s),\xi_{n_s+1}(s),\alpha(s),A_s,B_s} \sum_{s=1}^{S} \alpha(s), \quad subject\ to$$

$$(h) \qquad \mathcal{H}^*(A_s + \tfrac{1}{2}(L_1 B_s L_2^T + L_2 B_s L_1^T) - L_2 B_s L_2^T = \\ = R(s)^T \bar{\xi}(s) - \alpha(s) e_1, \quad \forall\ s \in \mathcal{S}$$

$$(i) \qquad \mathcal{H}(\bar{\xi}(s)) \succeq 0, \quad \forall\ s \in \mathcal{S}$$

$$(j) \qquad \tfrac{1}{2}(L_1^T \mathcal{H}(\bar{\xi})(s) L_2 + L_2^T \mathcal{H}(\bar{\nu})(s) L_1) + \\ - L_2^T \mathcal{H}(\bar{\xi})(s) L_2 \succeq 0, \quad \forall\ s \in \mathcal{S}$$

$$(k) \qquad \textstyle\sum_s (E_s - \beta Q(s))^T \bar{\xi}(s) = 1$$

$$(l) \qquad \mathcal{H}([\xi_1(s),\ \xi_2(s),\ ...,\ \xi_{n_s+1}(s)]^T) \succeq 0, \quad \forall\ s \in \mathcal{S}$$

$$(m) \quad \mathcal{H}([\xi_0(s),\ \xi_1(s),\ ...,\ \xi_{n_s}(s)]^T) - \mathcal{H}([\xi_1(s),\ \xi_2(s),\ ...,\ \xi_{n_s+1}(s)]^T) \succeq 0, \\ \forall\ s \in \mathcal{S}$$

$$(n) \qquad A_s, B_s \succeq 0, \quad \forall\ s \in \mathcal{S} \qquad (\mathrm{SD}')$$

È importante notare che, per coerenza di notazione, $\bar{\nu}(s)$ e $\bar{\xi}(s)$ indicano rispettivamente i vettori dei momenti dall'ordine zero all'ordine m_s ed n_s, senza comprendere quindi i momenti di ordine $m_s + 1$ e $n_s + 1$. A questo punto, normalizzando ed utilizzando la procedura indicata nella Sezione 3.3.2 è possibile ricostruire sia i supporti sia i pesi delle misure di probabilità μ e ν per ogni stato del gioco, ottenendo quindi la coppia di strategie miste ottime per ogni stato. È importante notare che, per ogni stato s, il supporto delle misure atomiche sarà dato dalle radici dei polinomi con coefficienti dati da $E_s\mathbf{v} - R(s)\bar{\nu} - \beta Q(s)\mathbf{v}$ ed ${R_s}^T \bar{\mu}(s) - \alpha(s)e_1$: questo significa che i supporti delle strategie miste ottime sono finiti, ed in particolare il giocatore 1 in un certo stato s avrà un supporto composto da non più di n_s azioni, mentre il giocatore 2 in un certo stato s avrà un supporto composto da non più di m_s

azioni.
Questo risultato fornisce una naturale e completa generalizzazione della classica soluzione in programmazione lineare dei giochi finiti a somma zero, e ne condivide le stesse importanti proprietà di dualità.

4.5 Esempio di Soluzione

Viene presentato in questa sezione un esempio di soluzione di un gioco stocastico a due giocatori e somma zero, in cui la funzione di payoff è polinomiale nelle azioni dei due giocatori e le probabilità di transizione sono anch'esse polinomiali, ma in funzione delle azioni del solo giocatore 1. Questo esempio è lo stesso trattato in [35].

Si consideri il gioco stocastico a due giocatori con valore di discount $\beta = 0.5$, a due stati ($\mathcal{S} = \{1, 2\}$), in cui le azioni dei giocatori 1 e 2 sono rispettivamente negli spazi delle azioni $A_1(s) = A_2(s) = [0, 1]\ \forall s \in \mathcal{S}$. Le funzioni di payoff sono $r(1, a_1, a_2) = (a_1 - a_2)^2$ e $r(2, a_1, a_2) = -(a_1 - a_2)^2$. Le probabilità di transizione sono date da:

$$P(a_1) = \begin{bmatrix} a_1 & 1 - a_1 \\ 1 - a_1^2 & a_1^2 \end{bmatrix}$$

Questo gioco può essere rappresentato graficamente come mostrato in Figura 4.2: i payoffs associati agli stati sono indicati nei nodi corrispondenti, mentre i lati sono marcati dalle corrispondenti probabilità di transizione negli stati.

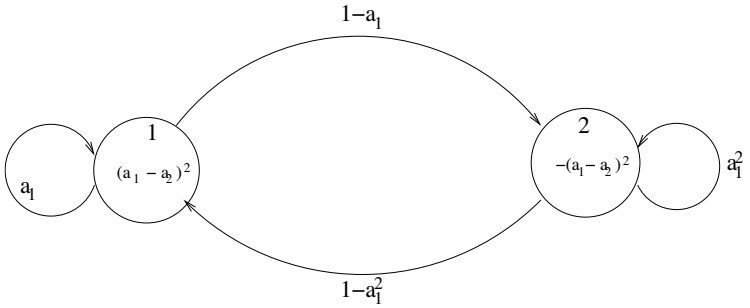

Figura 4.2: Rappresentazione del gioco stocastico a due stati con Single Controller considerato.

Per comprendere questo gioco, si consideri prima il gioco a somma zero

(non stocastico) con funzione di payoff $r(a_1, a_2) = (a_1 - a_2)^2$ sullo spazio di strategie $[0, 1] \times [0, 1]$. Questo gioco è il "guessing game", discusso nella Sezione 4.2.2, con l'unica differenza che lo spazio delle azioni non è più $[-1, 1] \times [-1, 1]$. Il giocatore 1 vorrebbe quindi tentare di "confondere" il giocatore 2 il più possibile, scegliendo in modo casuale tra $a_1 = 0$ ed $a_1 = 1$ con probabilità $\frac{1}{2}$. La migliore risposta del giocatore 2 sarebbe di giocare $a_2 = \frac{1}{2}$ con probabilità 1.

Nel gioco descritto in Figura 4.2, nello stato 1 il giocatore 1 tenta di confondere mentre il giocatore 2 tenta di indovinare; viceversa, nello stato 2, i due giocatori si scambiano gli ruoli. In ogni caso, il gioco è leggermente complicato dal fatto che il giocatore 1, comandando le transizioni, preferirà tentare di rimanere nello stato 1, in quanto in tale stato egli è avvantaggiato, poiché si possono ricevere payoff sempre non negativi. Questo significa che la strategia del giocatore 1 dovrà cercare di fruttargli dei buoni payoffs ed al tempo stesso favorire la probabilità di rigiocare nello stato 1.

Il problema di ottimizzazione polinomiale che calcola la strategia per il giocatore 2 (che minimizza) in ognuno dei due stati è quindi il seguente:

$$\min_{\nu(s), v(s)} v(1) + v(2), \quad \textit{subject to}$$

$$\begin{cases} v(1) \;\geq\; \int_{a_2 \in A_2} (a_1 - a_2)^2 \, \nu(1, a_2) \, da_2 + \\ \qquad + \beta\big(a_1 v(1) + (1 - a_1) v(2)\big) \quad \forall\, a_1 \in A_1 \\ \\ v(2) \;\geq\; - \int_{a_2 \in A_2} (a_1 - a_2)^2 \, \nu(2, a_2) \, da_2 + \\ \qquad + \beta\big((1 - a_1^2) v(1) + a_1^2 v(2)\big) \quad \forall\, a_1 \in A_1 \\ \\ \nu(1), \nu(2) \text{ sono misure di probabilitá supportate su } A_2. \end{cases} \tag{4.11}$$

Questo problema può essere riformulato come segue.

$$\min_{\nu(s),v(s)} v(1)+v(2), \quad \textit{subject to}$$

$$\begin{cases} v(1) \;\geq\; a_1^2 - 2a_1\nu_1(1) + \nu_2(1) + \\ \qquad\quad + \beta\big(a_1 v(1) + (1-a_1)v(2)\big) \quad \forall\, a_1 \,\in\, [0,1] \\ \\ v(2) \;\geq\; -a_1^2 + 2a_1\nu_1(2) - \nu_2(2) + \\ \qquad\quad + \beta\big((1-a_1^2)v(1) + a_1^2 v(2)\big) \quad \forall\, a_1 \,\in\, [0,1] \\ \\ [1, \nu_1(1), \nu_2(1)]^T,\; [2, \nu_1(2), \nu_2(2)]^T \;\in\; \mathcal{M}([0,1]). \end{cases} \tag{4.12}$$

Viene ora mostrato come scrivere il problema di programmazione semidefinita che permettere di risolvere il problema di ottimizzazione polinomiale (4.12). Si hanno quindi le seguenti costanti:

$$R(1) = -R(2) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -2 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad Q(1) = \begin{bmatrix} 0 & 1 \\ 1 & -1 \\ 0 & 0 \end{bmatrix}, \quad Q(2) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ -1 & 1 \end{bmatrix}.$$

Considerando, per comodità di notazione, $Z_s = Z(s)$ ed $W_s = W(s)$, si hanno poi le seguenti espressioni.

$$Z(1) = \begin{bmatrix} z_1(1) & z_2(1) \\ z_2(1) & z_3(1) \end{bmatrix}, \quad Z(2) = \begin{bmatrix} z_1(2) & z_2(2) \\ z_2(2) & z_3(2) \end{bmatrix}, \quad W(1) \in \mathcal{S}^1,\; W(2) \in \mathcal{S}^1.$$

Facendo quindi riferimento all'SDP (SP$'$), si mostra come riscrivere il vincolo (a):

$$\mathcal{H}^*(Z_s + \tfrac{1}{2}(L_1 W_s L_2^T + L_2 W_s L_1^T) - L_2 W_s L_2^T = \begin{bmatrix} z_1(s) \\ 2z_2(s) + W(s) \\ z_3(s) - W(s) \end{bmatrix},$$

$$E_1\mathbf{v} - R(1)\bar{\nu} - \beta Q(1)\mathbf{v} = \begin{bmatrix} v(1) - \frac{1}{2}v(2) - \nu_2(1) \\ 2\nu_1(1) - \frac{1}{2}v(1) + \frac{1}{2}v(2) \\ -\nu_0(1) \end{bmatrix},$$

$$E_2\mathbf{v} - R(2)\bar{\nu} - \beta Q(2)\mathbf{v} = \begin{bmatrix} \nu_2(2) - \frac{1}{2}v(1) + v(2) \\ -2\nu_1(2) \\ \nu_0(2) + \frac{1}{2}v(1) - \frac{1}{2}v(2) \end{bmatrix}.$$

Inoltre, in questo caso, si può riscrivere il vincolo (c) nel seguente modo:

$$\frac{1}{2}(L_1^T\mathcal{H}(\bar{\nu})(2)L_2 + L_2^T\mathcal{H}(\bar{\nu})(2)L_1) \; - \; L_2^T\mathcal{H}(\bar{\nu})(2)L_2 = \nu_1(s) - \nu_2(s).$$

A questo punto, quindi, si possono scrivere gli SDP associati a questo problema:

$$\min_{\bar{\nu}(s),\nu_3(s),v(s),Z_s,W_s} v(1) + v(2), \quad \textit{subject to}$$

$$(a) \quad \begin{bmatrix} z_1(1) \\ 2z_2(1) + W(1) \\ z_3(1) - W(1) \end{bmatrix} = \begin{bmatrix} v(1) - \frac{1}{2}v(2) - \nu_2(1) \\ 2\nu_1(1) - \frac{1}{2}v(1) + \frac{1}{2}v(2) \\ -\nu_0(1) \end{bmatrix}$$

$$\begin{bmatrix} z_1(2) \\ 2z_2(2) + W(2) \\ z_3(2) - W(2) \end{bmatrix} = \begin{bmatrix} \nu_2(2) - \frac{1}{2}v(1) + v(2) \\ -2\nu_1(2) \\ \nu_0(2) + \frac{1}{2}v(1) - \frac{1}{2}v(2) \end{bmatrix}$$

$$(b) \quad \begin{bmatrix} \nu_0(1) & \nu_1(1) \\ \nu_1(1) & \nu_2(1) \end{bmatrix} \succeq 0, \quad \begin{bmatrix} \nu_0(2) & \nu_1(2) \\ \nu_1(2) & \nu_2(2) \end{bmatrix} \succeq 0$$

$$(c) \quad \nu_1(1) - \nu_2(1) \succeq 0, \quad \nu_1(2) - \nu_2(2) \succeq 0$$

$$(d) \quad \nu_0(1) = 1, \qquad \nu_0(2) = 1 \tag{4.13}$$

$$(e) \quad \begin{bmatrix} \nu_1(1) & \nu_2(1) \\ \nu_2(1) & \nu_3(1) \end{bmatrix} \succeq 0,$$

$$\begin{bmatrix} \nu_1(2) & \nu_2(2) \\ \nu_2(2) & \nu_3(2) \end{bmatrix} \succeq 0,$$

$$(f) \quad \begin{bmatrix} \nu_0(1) & \nu_1(1) \\ \nu_1(1) & \nu_2(1) \end{bmatrix} - \begin{bmatrix} \nu_0(1) & \nu_1(1) \\ \nu_1(1) & \nu_2(1) \end{bmatrix} \succeq 0,$$

$$\begin{bmatrix} \nu_0(2) & \nu_1(2) \\ \nu_1(2) & \nu_2(2) \end{bmatrix} - \begin{bmatrix} \nu_0(2) & \nu_1(2) \\ \nu_1(2) & \nu_2(2) \end{bmatrix} \succeq 0$$

$$(g) \quad Z(1) \succeq 0,\ W(1) \succeq 0,\ Z(2) \succeq 0,\ W(2) \succeq 0$$

Il duale di questo SDP, che si ottiene facendo riferimento all'SDP (SD′) con un procedimento del tutto analogo a quello svolto per il primale, è il seguente:

$$\max_{\bar{\xi}(s),\xi_3(s),v(s),A_s,B_s} v(1)+v(2), \quad \textit{subject to}$$

$$(h) \qquad \begin{bmatrix} a_1(1) \\ 2a_2(1)+B(1) \\ a_3(1)-B(1) \end{bmatrix} = \begin{bmatrix} \xi_2(1)-\alpha(1) \\ -2\xi_1(1) \\ \xi_0(1) \end{bmatrix}$$

$$\begin{bmatrix} a_1(2) \\ 2a_2(2)+B(2) \\ a_3(2)-B(2) \end{bmatrix} = \begin{bmatrix} -\xi_2(2)-\alpha(2) \\ 2\xi_1(2) \\ -\xi_0(2) \end{bmatrix}$$

$$(i) \qquad \begin{bmatrix} \xi_0(1) & \xi_1(1) \\ \xi_1(1) & \xi_2(1) \end{bmatrix} \succeq 0, \quad \begin{bmatrix} \xi_0(2) & \xi_1(2) \\ \xi_1(2) & \xi_2(2) \end{bmatrix} \succeq 0$$

$$(j) \qquad \xi_1(1)-\xi_2(1) \succeq 0, \quad \xi_1(2)-\xi_2(2) \succeq 0$$

$$(k) \quad \big((E_1-\beta Q(1)\big)^T \xi(1) + \big((E_2-\beta Q(2)\big)^T \xi(2) = 1 \tag{4.14}$$

$$(l) \qquad \begin{bmatrix} \xi_1(1) & \xi_2(1) \\ \xi_2(1) & \xi_3(1) \end{bmatrix} \succeq 0,$$

$$\begin{bmatrix} \xi_1(2) & \xi_2(2) \\ \xi_2(2) & \xi_3(2) \end{bmatrix} \succeq 0,$$

$$(m) \qquad \begin{bmatrix} \xi_0(1) & \xi_1(1) \\ \xi_1(1) & \xi_2(1) \end{bmatrix} - \begin{bmatrix} \xi_0(1) & \xi_1(1) \\ \xi_1(1) & \xi_2(1) \end{bmatrix} \succeq 0,$$

$$\begin{bmatrix} \xi_0(2) & \xi_1(2) \\ \xi_1(2) & \xi_2(2) \end{bmatrix} - \begin{bmatrix} \nu_0(2) & \nu_1(2) \\ \nu_1(2) & \nu_2(2) \end{bmatrix} \succeq 0$$

$$(n) \qquad Z(1) \succeq 0,\ W(1) \succeq 0,\ Z(2) \succeq 0,\ W(2) \succeq 0$$

La soluzione dell'SDP primale (4.13) fornisce il vettore dei valore del gioco ed i primi m_s+2 momenti delle misure di probabilità (una misura per ogni

stato s) della strategia del giocatore 2. La soluzione del problema duale (4.14) fornisce invece i primi $n_s + 2$ momenti delle misure di probabilità (una misura per ogni stato s) della strategia del giocatore 1. La soluzione della coppia primale-duale produce i seguenti risultati.

$$
\begin{aligned}
&v(1) = 0.298, && v(2) = -0.158,\\
&\nu(1) = [1, 0.614, 0.377, 0.2315], && \nu(2) = [1, 0.614, 0.614, 0.614],\\
&\xi(1) = [2.3408, 1.4373, 1.4373, 1.4373], && \xi(2) = [1.6592, 0.8296, 0.4148, 0.2074],\\
&\mu(1) = [1, 0.614, 0.614, 0.614], && \mu(2) = [1, 0.5, 0.25, 0.125],
\end{aligned}
$$

dove $\mu(s)$ indica il vettore dei momenti dello stato s normalizzati. Le corrispondenti misure, calcolate secondo la procedura indicata nella Sezione 4.4.3, sono date dalle seguenti espressioni:

$$
\begin{aligned}
\mu^*(1) &= 0.386\delta(a_1) + 0.614\delta(a_1 - 1)\\
\mu^*(2) &= \delta(a_1 - 0.5)\\
\nu^*(2) &= \delta(a_2 - 0.614)\\
\nu^*(2) &= 0.386\delta(a_2) + 0.614\delta(a_2 - 1)
\end{aligned}
$$

Si consideri, per esempio, lo stato 1. Se il giocatore 1 giocasse senza curarsi delle transizioni di stato, giocherebbe le azioni $a_1 = 0$ ed $a_1 = 1$ con $\frac{1}{2}$ di probabilità ciascuna. Comunque, considerando anche le transizioni di stato, per incrementare la probabilità di rimanere nello stato 1 giocherà l'azione 1 con una probabilità maggiore, ovvero 0.614. Il giocatore 2 non può influenzare le probabilità di transizione di stato direttamente, quindi giocherà una miglior risposta *miope* (i.e. una miglior risposta guardando solo il gioco nello stato corrente), scegliendo l'azione $a_2 = 0.614$ con probabilità 1. Nello stato 2, invece, la migliore strategia del giocatore 1 è di scegliere l'azione $a_1 = 0.5$. La miglior risposta (miope) per il giocatore 2 è di scegliere una qualsiasi distribuzione di probabilità con supporto sulle azioni $a_2 = 1$ e $a_2 = 0$: ad esempio, sceglierà di giocare $a_2 = 1$ con probabilità 0.614 e $a_2 = 0$ con probabilità 0.386.

Capitolo 5

Processi di Decisione di Markov Polinomiali

5.1 Introduzione

Nella Sezione 2.5 sono stati introdotti i processi di decisione di Markov finiti, che possono essere visti come dei giochi stocastici a singolo giocatore. Gli MDP sono molto importanti, in quanto permettono di modellizzare problemi decisionali in un mondo incerto, ovvero con componenti stocastici. In questo capitolo vengono presentati gli MDP con azioni continue, payoffs polinomiali e probabilità di transizione polinomiali nelle azioni del giocatore, mostrando un algoritmo in grado di calcolare il valore degli stati con un unico problema di programmazione semidefinita. Nella Sezione 5.2 si mostra come gli MDP ad azioni finite vengono risolti mediante la programmazione lineare. Nella Sezione 5.3 viene mostrata invece l'estensione al caso di azioni continue, payoffs polinomiali e probabilità di transizione polinomiali, presentando l'algoritmo in grado di calcolare il valore di ogni stato. Nella Sezione 5.4 viene poi discusso il calcolo della risposta migliore di un giocatore ad un fissato profilo di strategie degli avversari in un gioco stocastico polinomiale, ottenibile risolvendo un MDP ad azioni continue, payoff polinomiali e probabilità di transizione polinomiali. Infine nella Sezione 5.5 viene presentato un esempio di calcolo della risposta migliore di un giocatore ad un noto profilo di strategie dei giocatori avversari nel caso di gioco stocastico polinomiale: in questo modo viene presentato al tempo stesso anche un esempio di soluzione di un MDP con azioni continue.

5.2 MDP ad Azioni Finite

Si consideri un processo di decisione di Markov $M = (\mathcal{S}, A, P, r)$ ad azioni e stati finiti. Essendo un caso particolare di un gioco stocastico (a singolo giocatore), si può definire meglio il concetto di strategia per un MDP. In particolare, una strategia per un processo di decisione di Markov è una *politica* $\Pi : S \to A$ che mappa ogni stato in una azione, ovvero indica quale azione verrà scelta dal giocatore quando si troverà in uno stato. Questo significa che, a differenza dei giochi stocastici, nei processi di decisione di Markov si considerano unicamente strategie stazionarie nelle quali la strategia in ogni singolo stato è una strategia pura, in quanto essendoci un unico giocatore non vi è più la necessità di attuare strategie miste per raggiungere la strategia d'equilibrio. L'utilità attesa di uno stato s (o valore dello stato s) seguendo la politica π è definita come la somma attesa dei discounted rewards a partire dallo stato s seguendo tale politica. Il processo si svolge lungo un orizzonte infinito, perciò come nel caso generale dei giochi stocastici i rewards vengono accumulati utilizzando uno dei due metodi di aggregazione visti nella Sezione 2.5, ovvero la ricompensa media e la ricompensa scontata. In questo capitolo ci si concentra sul caso in cui come metodo di aggregazione si utilizza la ricompensa scontata. Trovare la politica ottima π^* in un MDP significa perciò individuare qual è l'azione $a \in A$ migliore da intraprendere quando ci si trova nello stato $s \in \mathcal{S}$, per ogni possibile stato $s \in \mathcal{S}$, dove per "migliore" si intende l'azione che massimizza l'utilità attesa di ogni stato seguendo la politica π^*.

Come mostrato in [30], il valore di uno stato quando si adotta la politica ottima soddisfa l'equazione di Bellman:

$$V(s) = \max_{a \in A} \left[R(s,a) + \beta \sum_{s' \in S} P(s', s, a) V(s') \right] \tag{5.1}$$

dove $V(s)$ indica il valore dello stato s.

Esistono tre algoritmi "classici" per ottenere le politiche ottime nei processi di decisione di Markov: il value iteration, il policy iteration e la programmazione lineare. Molta della letteratura si focalizza sul value iteration e sul policy iteration, ma ci sono almeno tre buone ragioni per considerare la soluzione di un MDP attraverso l'uso della programmazione lineare. Prima di tutto, un problema di programmazione lineare produce una soluzione esatta senza la necessità di specificare nessun criterio di termine, come invece è necessario nel

caso del policy iteration e del value iteration. Secondo, molti algoritmi basati sull'approssimazione della funzione di valutazione degli stati sono basati sulla programmazione lineare [11] [12] [34]. Terzo, e per altro più importante, la programmazione lineare sembra offrire l'unico modo di considerare problemi in cui la massimizzazione del discounted reward totale atteso è soggetta a vincoli addizionali sui rewards attesi [1] [30]. Oltre a questi validi motivi, la soluzione mediante la programmazione lineare porta ad una formulazione favorevole alla generalizzazione al caso di azioni continue e payoff polinomiali, che verrà introdotta nella Sezione 5.3. Viene ora mostrato come calcolare la politica ottima di un processo di decisione di Markov mediante la programmazione lineare.

Si consideri il seguente problema di programmazione lineare:

$$\begin{aligned} &\min_{v(s)} \sum_{s \in \mathcal{S}} v(s), \quad \textit{subject to} \\ &v(s) - \beta \textstyle\sum_{s' \in S} P(s', s, a) v(s') - R(s, a) \geq 0, \quad \forall\, s \in \mathcal{S}, a \in A \end{aligned} \tag{LPM}$$

Il problema di programmazione lineare (LPM) ammette sempre una soluzione ottima che coincide con il valore degli stati quando si adotta la politica ottima Π^* [30].

5.3 MDP ad Azioni e Payoff Polinomiali

Molti degli attuali algoritmi di pianificazione dei processi di decisione di Markov prevedono degli spazi di azioni discreti e finiti. Risulta quindi interessante l'analisi del caso in cui i domini delle azioni siano continui. Un approccio naturale a questo tipo di giochi può essere quello di discretizzare lo spazio delle azioni ed eseguire una pianificazione sullo spazio delle azioni modificato. Un approccio di questo tipo può per esempio essere trovato in [15] e [48]. Il rischio di questo approccio è che la scelta di discretizzare le azioni può risultare inappropriata per certi domini, con un forte impatto negativo sulla qualità delle soluzioni trovate. Per questo motivo, viene ora presentato un algoritmo per risolvere un processo di decisione di Markov ad azioni continue e payoff polinomiali, basato sulla soluzione di un unico problema di programmazione semidefinita. Risolvendo quindi un unico SDP, sarà possibile ottenere i valori degli stati e, di conseguenza, la politica ottima dell'agente. Viene quindi mostrato come derivare tale problema di programmazione semidefinita

partendo dal problema di programmazione lineare (LPM) che risolve un MDP ad azioni finite.

5.3.1 Da Azioni Finite ed LP ad Azioni Infinite ed SDP

Si consideri il caso in cui l'agente possa scegliere tra infinite azioni non numerabili: in particolare, può scegliere azioni nell'intervallo $A = [c, d]$. Per semplicità, si consideri il caso in cui $A_1(s) = A_2(s) = [0, 1] \in \mathbb{R}$, $\forall\, s \in \mathcal{S}$. I risultati si possono facilmente generalizzare al caso in cui gli spazi delle strategie sono unioni finite di intervalli arbitrari dell'asse reale e/o possono variare da stato a stato. Il numero di stati $|\mathcal{S}| = S$ rimane finito. La funzione di payoff $r(s, a)$, polinomiale nella variabile a, è la seguente:

$$r(s, a) = \sum_{i=0}^{n_s} r_i(s) a^i, \quad \forall s \in \mathcal{S}. \tag{5.2}$$

Inoltre anche la probabilità di transizione $p(s', s, a)$ è polinomiale nell'azione a dell'agente:

$$p(s', s, a) = \sum_{i=0}^{n_{ss'}} p_i(s', s) a^i, \quad \forall s \in \mathcal{S}. \tag{5.3}$$

Si consideri ancora il caso in cui l'agente vuole massimizzare il reward atteso lungo l'orizzonte infinito utilizzando come metodo di aggregazione la ricompensa scontata. Viene ora mostrato che la generalizzazione del problema (LPM) a questo caso porta ad un problema di ottimizzazione che riguarda la non negatività di un sistema di polinomi univariati.

Il seguente problema di ottimizzazione generalizza il problema di programmazione lineare (LPM) al caso di azioni continue con funzione di payoff e probabilità di transizione polinomiale.

$$\begin{aligned} &\min_{v(s)} \sum_{s=1}^{S} v(s), \quad \textit{subject to} \\ &v(s) \geq r(s, a) + \beta \textstyle\sum_{s'=1}^{S} p(s', s, a) v(s') \quad \forall\, s \in \mathcal{S},\, a_1 \in A_1 \end{aligned} \tag{PM}$$

È possibile riscrivere tale problema di ottimizzazione, indicando che il vincolo richieda la non negatività del polinomio lungo l'intervallo $A = [0, 1]$, ottenendo il problema di ottimizzazione polinomiale seguente.

$$\min_{v(s)} \sum_{s=1}^{S} v(s), \quad subject\ to$$

$$v(s) - \sum_{i=0}^{n_s} r_i(s)a^i - \beta \sum_{s'=1}^{S} \sum_{i=0}^{n_{ss'}} p_i(s,s')a^i v(s') \ \in \mathcal{P}(A), \forall\, s \in \mathcal{S} \tag{PM$'$}$$

La formulazione del problema di ottimizzazione (PM$'$) è astratta e quindi non risolvibile. Per cercare di convertirlo in un problema di ottimizzazione concreto che si possa risolvere, risulta necessaria la rappresentazione computazionalmente adatta dell'insieme $\mathcal{P}(A)$ introdotta nella Sezione 3.3. Come è noto dalla Sezione 3.3.1, le condizioni di non negatività di un polinomio su un intervallo si applicano ai coefficienti di tale polinomio. Per questo motivo, si può notare che il vincolo del problema (PM$'$) produce un sistema di disuguaglianze polinomiali in a, una disuguaglianza per ogni stato: fissato un certo stato $s \in \mathcal{S}$ si cerca di esplicitare i coefficienti del polinomio

$$t_s(a) \ = \ v(s) - \sum_{i=0}^{n_s} r_i(s)a^i - \beta \sum_{s'=1}^{S} \sum_{i=0}^{n_{ss'}} p_i(s,s')a^i v(s').$$

Sia d_s il grado della disuguaglianza per lo stato s, e sia inoltre $[a_1]_{d_s} = [1, a_1, {a_1}^2, ..., {a_1}^{d_s}]^T$. Il secondo termine del polinomio $t_s(a)$ può essere riscritto in forma vettoriale come:

$$\sum_{i=0}^{n_s} r_i(s)a^i \ = \ R(s)^T [a]_{d_s} \tag{5.4}$$

dove $R(s) \in \mathbb{R}^{d_s+1}$ è il vettore che contiene i coefficienti del polinomio $r(s,a)$. Si definisce ora il vettore $\mathbf{v}^* = [v^*(1), ..., v^*(S)]^T$, che risulterà essere il vettore del valore degli stati (indicizzato dallo stato). Il terzo termine del polinomio $t_s(a)$, che dipende dalla probabilità di transizione $p(s', s, a)$, è anch'esso ovviamente un polinomio in a ed i suoi coefficienti dipendono dai coefficienti di $p(s', s, a)$ e $\mathbf{v}$. Specificatamente, si ha:

$$\sum_{s'=1}^{S} \sum_{i=0}^{n_{ss'}} p_i(s,s')a^i v(s') \ = \ \mathbf{v}^T Q(s)^T [a]_{d_s}, \tag{5.5}$$

dove $Q(s)$ è la matrice che contiene i coefficienti di $p(s', s, a)$.

A questo punto, poiché gli spazi delle azioni sono $A_1 = A_2 = [0, 1]$, dal Lemma 4 nella Sezione 3.3.1, si può concludere che il polinomio $t_s(a)$ è non

negativo in $[0,1]$ se e solo se esistono le matrici $Z_s \in \mathcal{S}^{d_s+1}$ e $W_s \in \mathcal{S}^{d_s}$, $Z \succeq 0$, $W \succeq 0$ tali che

$$\mathcal{H}^*(Z_s + \frac{1}{2}(L_1 W_s L_2^T + L_2 W_s L_1^T) - L_2 W_s L_2^T) = E_s \mathbf{v} - R(s) - \beta Q(s)\mathbf{v} \quad (5.6)$$

dove $E_s \in \mathbb{R}^{d_s \times S}$ è la matrice con tutti zeri eccetto un 1 in posizione $(1, s)$.

Si possono quindi rimettere insieme tutte queste condizioni, per formare il singolo problema di programmazione semidefinita positiva che concretizza il problema di ottimizzazione astratto (PM′), ottenendo quindi il seguente SDP.

$$\min_{Z_s, W_s, v(s)} \sum_{s=1}^{S} v(s), \quad \textit{subject to}$$

$$\begin{aligned} &\mathcal{H}^*(Z_s + \tfrac{1}{2}(L_1 W_s L_2^T + L_2 W_s L_1^T) - L_2 W_s L_2^T = \\ &= E_s \mathbf{v} - R(s) - \beta Q(s)\mathbf{v}, \quad \forall\ s \in \mathcal{S} \end{aligned} \qquad \text{(SM)}$$

$$Z_s \succeq 0, \quad W_s \succeq 0, \quad \forall\ s \in \mathcal{S}$$

A questo punto, possiamo esprimere il seguente Lemma.

Lemma 14. Siano $A_1 = A_2 = [0,1]$. Il problema di programmazione semidefinita (SM) risolve esattamente il problema di ottimizzazione polinomiale (PM′).

Dimostrazione. La disuguaglianza polinomiale nel vincolo del problema (PM′) ha il vettore dei coefficienti $E_s\mathbf{v} - R(s) - \beta Q(s)\mathbf{v}$, come mostrato dalle equazioni (5.4) e (5.5). La dimostrazione si ha quindi per diretta conseguenza del Lemma 4 riguardante la rappresentazione semidefinita dei polinomi non negativi su $[0,1]$, come risulta evidente dall'equazione (5.6). □

5.3.2 Ottimalità della Soluzione

È stato dimostrato che il problema (PM′) può essere ridotto al problema di programmazione semidefinita (SM). A questo punto, poiché il problema di ottimizzazione polinomiale (PM′) è una semplice estensione del problema di programmazione lineare (LPM), che risolve il processo di decisione di Markov al caso di azioni finite, risulta ovvio che la soluzione ottenuta risolvendo il problema (PM′) è la soluzione ottima del processo di decisione di Markov ad

azioni continue, con funzione di payoff e transizioni di probabilità polinomiali nelle azioni dell'agente.

5.4 Best Response in un Gioco Stocastico Polinomiale

Si consideri un gioco stocastico e si supponga di fissare tutte le strategie dei giocatori tranne uno, chiamato *giocatore* 1. La risposta migliore che il giocatore 1 può realizzare rispetto alle strategie fissate di tutti gli altri giocatori, può essere ottenuta mediante la soluzione di un processo di decisione di Markov. Infatti, nel caso in cui le strategie di tutti i giocatori tranne uno sono fissate, il gioco stocastico si riduce ad un processo di decisione di Markov in cui il giocatore 1 deve massimizzare la sua utilità in ogni stato, in accordo con le strategie fissate degli avversari. In questa sezione si pone il problema di calcolare la risposta migliore del giocatore 1 ad un certo profilo di strategie noto degli avversari, in un gioco stocastico in cui sia la funzione di payoff sia le probabilità di transizione sono polinomiali rispetto alle azioni continue dei giocatori.

5.4.1 Descrizione del Problema

Si consideri il gioco stocastico $G = (\mathcal{S}, N, A, P, r)$. Per semplicità, si supponga di avere due giocatori; i risultati possono essere facilmente generalizzati al caso di N giocatori. Si supponga che la funzione di payoff sia polinomiale nelle variabili a_1 e a_2, che rappresentano le azioni scelte rispettivamente dal giocatore 1 e 2, e che sono valori reali che appartengono agli spazi delle azioni A_1 e A_2. Per semplicità, si consideri il caso in cui $A_1(s) = A_2(s) = [0, 1] \in \mathbb{R}$, $\forall\, s \in \mathcal{S}$. I risultati si possono facilmente generalizzare al caso in cui gli spazi delle strategie sono unioni finite di intervalli arbitrari dell'asse reale e/o possono variare da stato a stato. Il numero di stati $|\mathcal{S}| = S$, è ancora finito. Si supponga che anche la probabilità di transizione $p(s', s, a_1, a_2)$ sia polinomiale nelle azioni dei giocatori con coefficienti reali. Si hanno quindi le due seguenti equazioni:

$$r(s, a_1, a_2) = \sum_{i=0}^{n_s} \sum_{j=0}^{m_s} r_{ij}(s)\, {a_1}^i\, {a_2}^j,$$
$$p(s', s, a_1, a_2) = \sum_{i=0}^{n_{ss'}} \sum_{j=0}^{m_{ss'}} p_{ij}(s', s)\, {a_1}^i\, {a_2}^j.$$

È importante notare che la funzione $r(s, a_1, a_2)$ indica il payoff che riceve il giocatore 1: non si pone nessuna ipotesi sulle proprietà della funzione di payoff (ovvero, non è richiesto che il gioco sia a somma zero). Il processo di decisione opera lungo un orizzonte infinito, quindi risulta naturale restringere l'attenzione alle strategie stazionarie per ogni giocatore, ed utilizzare come metodo di aggregazione la ricompensa scontata. Si supponga infine che i giocatori possano adottare strategie miste. Una strategia mista per il giocatore 2 è quindi l'insieme finito di misure di probabilità $\nu = [\nu(1), ..., \nu(S)]$, supportate sull'insieme delle azioni A_2 ed indicizzate dallo stato. Ogni misura di probabilità corrisponde perciò ad una strategia mista per il giocatore 2 in un particolare stato. Allo stesso modo, la strategia del giocatore 1 è rappresentata da $\mu = [\mu(1), ..., \mu(S)]$.

5.4.2 Calcolo della Best Response

Si fissi ora la strategia mista $\nu = \nu_0$ per il giocatore 2 in ogni stato, ovvero si consideri l'insieme delle misure di probabilità $\nu_0 = [\nu_0(1), ..., \nu_0(S)]$. Si indichi con $\nu_0(s, a_2)$ la probabilità che nello stato s venga scelta l'azione a_2 dal giocatore 2 secondo la strategia $\nu_0(s)$. Qual è la miglior risposta che il giocatore 1 può adottare una volta che conosce la strategia ν_0 del giocatore 2? Come è stato anticipato, la miglior risposta del giocatore 1 ad un fissato profilo di strategie si ottiene mediante la soluzione di un processo di decisione di Markov. Questo risultato si mantiene anche nel caso di payoff e transizioni di probabilità polinomiali rispetto alle azioni continue dei giocatori.

Si può subito notare che, nonostante permettiamo al giocatore 1 l'utilizzo di strategie miste, la risposta migliore sarà una strategia pura in ogni stato del gioco stocastico (in quanto soluzione di un processo di decisione di Markov). Questo risultato non stupisce, in quanto avendo fissato la strategia per il giocatore 2 si perde completamente la necessità per il giocatore 1 di introdurre la casualità nella scelta delle proprie azioni.

Si indichi con $\Psi_{\nu_0} = (\Psi_{\nu_0}(1), ..., \Psi_{\nu_0}(S))$ l'insieme dei supporti della strategia ν_0 del giocatore 2 indicizzato dallo stato. In questo modo, l'insieme $\Psi_{\nu_0}(k)$ con $k \in \mathcal{S}$ conterrà tutte e sole le azioni che vengono giocate dal giocatore 2 nello stato k con probabilità non nulla secondo la strategia ν_0. Per semplicità, si supponga che il numero di azioni che il giocatore 2 gioca con probabilità non

nulla sia finito in ogni stato; questa ipotesi può essere facilmente rimossa, a costo di un aumento di notazione. Il problema di programmazione semidefinita che permette di calcolare la miglior risposta per il giocatore 1 alla strategia ν_0 è quindi la seguente rivisitazione del problema di ottimizzazione (PM′):

$$\min_{v(s)} \sum_{s=1}^{S} v(s), \quad \textit{subject to} \tag{BR}$$

$$v(s) - r(s,a_1) - \beta \sum_{s'=1}^{S} p(s',s,a_1) v(s') \ \in \ \mathcal{P}(A), \forall\, s \ \in \ \mathcal{S}$$

dove per ogni $a_1 \in A$, $s \in \mathcal{S}$ si ha:

$$r(s,a_1) = \sum_{z \in \Psi_{\nu_0}(s)} \sum_{i=0}^{n_s} \sum_{j=0}^{m_s} r_{ij}(s)\, {a_1}^i\, z^j\, \nu_0(s,z), \tag{5.7}$$

$$p(s',s,a_1) = \sum_{z \in \Psi_{\nu_0}(s)} \sum_{i=0}^{n_{ss'}} \sum_{j=0}^{m_{ss'}} p_{ij}(s',s)\, {a_1}^i\, z^j\, \nu_0(s,z). \tag{5.8}$$

Come per il problema (PM′), la formulazione del problema di ottimizzazione (BR) è astratta e non risolvibile. Anche in questo caso, per cercare di convertirla in un problema di ottimizzazione concreto che si possa risolvere, si considera il vincolo del problema (BR) che produce un sistema di disuguaglianze polinomiali in a_1, una disuguaglianza per ogni stato: fissato un certo stato $s \in \mathcal{S}$, si cerca di esplicitare i coefficienti del polinomio

$$t_s(a_1) = v(s) - \textstyle\sum_{z \in \Psi_{\nu_0}(s)} \sum_{i=0}^{n_s} \sum_{j=0}^{m_s} r_{ij}(s)\, {a_1}^i\, z^j\, \nu_0(s,z) + $$
$$- \beta \textstyle\sum_{s'=1}^{S} \sum_{z \in \Psi_{\nu_0}(s)} \sum_{i=0}^{n_s} \sum_{j=0}^{m_s} p_{ij}(s',s)\, {a_1}^i\, z^j\, \nu_0(s,z)\, v(s').$$

Sia d_s il grado della disuguaglianza per quello stato, e sia inoltre $[a_1]_{d_s} = [1, a_1, {a_1}^2, ..., {a_1}^{d_s}]^T$. Si definisca il vettore $\mathbf{v}^* = [v^*(1), ..., v^*(S)]^T$, che risulterà essere il vettore dei valori degli stati. Si indichi con $R(s) \in \mathbb{R}^{d_s+1}$ il vettore che contiene i coefficienti del polinomio $r(s,a)$. Per ogni insieme X, sia $\prod(X)$ l'insieme di tutte le distribuzioni di probabilità su X. Sia $\Phi = \big(\prod A_2(1) \times \prod A_2(2) \times ... \times \prod A_2(S)\big)$ l'insieme delle strategie miste per il giocatore 2 nel gioco G. Si consideri l'operatore lineare $\mathcal{F} : \mathcal{S} \times A_s \times \Phi \to \mathbb{R}^{m_s+1}$ definito come:

$$\mathcal{F}\big(s, a_2, \nu\big) = \nu(s,a_2)[a_2^0,\, a_2^1, ...,\, a_2^{m_s}].$$

Si hanno quindi le seguenti equazioni:

$$\sum_{z \in \Psi_{\nu_0}(s)} \sum_{i=0}^{n_s} \sum_{j=0}^{m_s} r_{ij}(s)\, {a_1}^i\, z^j\, \nu_0(s,z) = R(s) \left(\sum_{z \in \Psi_{\nu_0}(s)} \mathcal{F}(s,z,\nu_0) \right)^T [a_1]_{d_s}, \tag{5.9}$$

$$\sum_{s'=1}^{S} \sum_{z \in \Psi_{\nu_0}(s)} \sum_{i=0}^{n_s} \sum_{j=0}^{m_s} p_{ij}(s', s)\, {a_1}^i\, z^j\, \nu_0(s, z)\, v(s') \;=\; Q(s)\, \mathbf{v}\, [a_1]_{d_s}, \tag{5.10}$$

dove $Q(s)$ è la matrice che contiene i coefficienti del polinomio $p(s', s, a_1)$ indicato in (5.8).

A questo punto, poiché gli spazi delle azioni sono $A_1 = A_2 = [0, 1]$, dal Lemma 4 nella Sezione 3.3.1, è possibile concludere che il polinomio $t_s(a_1)$ è non negativo in $[0, 1]$ se e solo se esistono le matrici $Z_s \in \mathcal{S}^{d_s+1}$ e $W_s \in \mathcal{S}^{d_s}$, $Z \succeq 0$, $W \succeq 0$ tali che

$$\begin{aligned} &\mathcal{H}^*(Z_s + \tfrac{1}{2}(L_1 W_s L_2^T + L_2 W_s L_1^T) - L_2 W_s L_2^T) = \\ &\qquad = E_s\mathbf{v} - R(s) \left(\textstyle\sum_{z \in \Psi_{\nu_0}(s)} \mathcal{F}(s, z, \nu_0) \right)^T - \beta Q(s)\mathbf{v} \end{aligned} \tag{5.11}$$

dove $E_s \in \mathbb{R}^{d_s \times S}$ è la matrice con tutti zeri eccetto un 1 in posizione $(1, s)$.

È quindi possibile scrivere il singolo problema di programmazione semidefinita positiva che concretizza il problema di ottimizzazione astratto (BR).

$$\min_{Z_s, W_s, v(s)} \sum_{s=1}^{S} v(s), \quad \textit{subject to}$$

$$\begin{aligned} &\mathcal{H}^*(Z_s + \tfrac{1}{2}(L_1 W_s L_2^T + L_2 W_s L_1^T) - L_2 W_s L_2^T = \\ &= E_s\mathbf{v} - R(s) \left(\textstyle\sum_{z \in \Psi_{\nu_0}(s)} \mathcal{F}(s, z, \nu_0) \right)^T - \beta Q(s)\mathbf{v}, \quad \forall\, s \in \mathcal{S} \end{aligned} \tag{SBR}$$

$$Z_s \succeq 0, \quad W_s \succeq 0, \quad \forall\, s \in \mathcal{S}$$

A questo punto, è possibile esprimere il seguente Lemma.

Lemma 15. Siano $A_1 = A_2 = [0, 1]$. Il problema di programmazione semidefinita (SBR) risolve esattamente il problema di ottimizzazione polinomiale (BR).

Dimostrazione. La disuguaglianza polinomiale nel vincolo del problema (BR) ha il vettore dei coefficienti $E_s\mathbf{v} - R(s) \left(\sum_{z \in \Psi_{\nu_0}(s)} \mathcal{F}(s, z, \nu_0) \right)^T - \beta Q(s)\mathbf{v}$, come mostrato dalle equazioni (5.9) e (5.10). La dimostrazione si ha quindi per diretta conseguenza del Lemma 4 riguardante la rappresentazione semidefinita dei polinomi non negativi su $[0, 1]$, come risulta evidente dall'equazione (5.11). □

Il problema di programmazione semidefinita (SBR) permette quindi di calcolare il valore degli stati quando un giocatore applica la sua risposta migliore ad un profilo di strategie noto dell'avversario, in un gioco stocastico con azioni continue, funzione di payoff e probabilità di transizioni polinomiali nelle azioni dei giocatori.

5.5 Esempio di Soluzione

In questa sezione viene presentato un esempio di calcolo della risposta migliore di un giocatore ad un profilo di strategie (noto) dei giocatori avversari, nel caso di gioco stocastico ad azioni continue, con funzione di payoff e transizioni di probabilità polinomiali nelle azioni dei giocatori. In questo modo viene presentato al tempo stesso anche un esempio di soluzione di un processo di decisione di Markov con azioni, payoff e probabilità di transizioni di questo tipo.

Si consideri il gioco stocastico a due giocatori e somma zero presentato nella Sezione 4.5, dove il valore di discount è $\beta = 0.5$, vi sono due stati ($\mathcal{S} = \{1, 2\}$), e le azioni dei giocatori 1 e 2 sono rispettivamente negli spazi delle azioni $A_1(s) = A_2(s) = [0, 1]\ \forall s \in \mathcal{S}$. Le funzioni di payoff sono:

- $r(1, a_1, a_2) = (a_1 - a_2)^2$,
- $r(2, a_1, a_2) = -(a_1 - a_2)^2$.

Le probabilità di transizione sono date da:

$$P(a_1) = \begin{bmatrix} a_1 & 1 - a_1 \\ 1 - a_1^2 & a_1^2 \end{bmatrix}.$$

È stato mostrato che per tale gioco la strategia ottima ν_0 del giocatore 2 è di scegliere nello stato 1 l'azione $a_2 = 0.614$ con probabilità 1, e nello stato 2 le azioni $a_2 = 1$ con probabilità 0.614 e $a_2 = 0$ con probabilità 0.386. Per questo motivo si ha:

$$\Psi_{\nu_0}(1) = \{0.614\}, \quad \nu_0(1, 0.614) = 1,$$
$$\Psi_{\nu_0}(2) = \{0, 1\}, \quad \nu_0(2, 0) = 0.386, \ \ \nu_0(2, 1) = 0.614$$

Viene ora mostrato come calcolare la miglior risposta del giocatore 1 quando il giocatore 2 utilizza la strategia ν_0: applicando il problema di programmazione

semidefinita (SBR), si ottiene l'SDP seguente.

$$\min_{Z_1,Z_2,W_1,W_2,v(1),v(2)} v(1)+v(2), \quad \textit{subject to}$$

$$\mathcal{H}^*(Z_1+\tfrac{1}{2}(L_1W_1L_2^T+L_2W_1L_1^T)-L_2W_1L_2^T=$$
$$=E_1\mathbf{v}-R(1)\left(\mathcal{F}(1,0.614,1)\right)^T-\beta Q(1)\mathbf{v}, \quad \forall\ s\in\mathcal{S}$$

$$\mathcal{H}^*(Z_2+\tfrac{1}{2}(L_1W_2L_2^T+L_2W_2L_1^T)-L_2W_2L_2^T=$$
$$=E_2\mathbf{v}-R(2)\left(\mathcal{F}(2,0,0.386)+\mathcal{F}(2,1,0.614)\right)^T-\beta Q(2)\mathbf{v}, \quad \forall\ s\in\mathcal{S}$$

$$Z_1\succeq 0, \quad Z_2\succeq 0, \quad W_1\succeq 0, \quad W_2\succeq 0, \tag{5.12}$$

dove $Z_1\in\mathcal{S}^2, Z_2\in\mathcal{S}^2$, $W_1\in\mathbb{R}, W_2\in\mathbb{R}$, mentre le costanti sono uguali a quelle definite nella sezione 4.5. La soluzione di questo problema di programmazione semidefinita produce i seguenti risultati:

$$v(1)=0.298, \qquad v(2)=-0.158.$$

Si può notare che tali risultati sono gli stessi prodotti dalla soluzione del problema di programmazione (4.13) nella Sezione 4.5: questo non è un risultato inatteso, in quanto mostra come la miglior risposta per il giocatore 1 alla strategia minimax del giocatore 2, porti al giocatore 1 un payoff non maggiore del livello di sicurezza, che il giocatore 2 si è garantito ottenendo la ν_0 dal problema (4.13).

Capitolo 6

Giochi Polinomiali Stocastici con Switching Control

6.1 Introduzione

Nel Capitolo 4 sono stati presentati i giochi stocastici con Single Controller, in cui le transizioni di stato dipendono dalle azioni di un unico giocatore. Una naturale estensione di questa classe di giochi è quella in cui il giocatore che "governa" le transizioni cambia da stato a stato: tale classe di giochi prende il nome di giochi stocastici con *Switching Control*. In questo capitolo viene presentata la classe dei giochi stocastici polinomiali con Switching Control, presentando in particolare un algoritmo in grado di risolvere giochi di questa classe ottenendo un ϵ-equilibrio. In questo capitolo vengono presentati alcuni dei principali contributi originali di questa tesi. Nella Sezione 6.2 viene presentata questa classe di giochi, discutendo l'esistenza di strategie di equilibrio. Nella Sezione 6.3 viene discusso il calcolo dell'equilibrio e del vettore del valore degli stati nei giochi stocastici con Switching Control in cui ogni giocatore ha accesso ad un numero finito di strategie pure tra cui scegliere. Infine nella Sezione 6.4 viene derivato il calcolo delle strategie d'equilibrio nel caso in cui ogni giocatore ha accesso ad un numero infinito di strategie pure, presentando l'algoritmo che permette di calcolare i valori degli stati e le strategie ottime per i giocatori.

6.2 I Giochi Polinomiali Stocastici con Switching Control

Come è stato anticipato nel Capitolo 4, sono state proposte diverse estensioni dei giochi stocastici con Single Controller. In particolare, una delle estensioni più importanti è la classe dei Giochi Stocastici con Switching Control, presentata da Filar [9] nel 1981. In [7] e [9], Filar dimostrò che anche tali giochi posseggono la proprietà di Orderfield. Questo indicò che per il caso di Switching Control potrebbe esistere un algoritmo finito in grado di risolverli, ed un primo tentativo di trovare un algoritmo di questo tipo venne fatto in [8]. Nel 1983 [22] venne proposto un algoritmo per il caso undiscounted, mentre nel 1987 Vrieze [47] propose un algoritmo per il caso discounted, che consiste nel risolvere un numero finito di problemi di programmazione lineare in modo analogo al metodo di policy iteration. Sempre nel 1987, Mohan e Raghavan [32] proposero invece un algoritmo finito per il caso discounted analogo al value iteration. Si svolsero in seguito altre notevoli ricerche su questa classe di giochi, come ad esempio in [16] e [43], ma nonostante queste ricerche tutt'ora non esiste (ancora) un algoritmo efficiente per risolvere i giochi di questa classe [16].

La classe dei giochi stocastici con Switching Control è sicuramente interessante, in quanto è facile immaginare situazioni in cui un giocatore sia tentato di entrare in uno stato del gioco con dei payoff possibilmente alti, ma a costo di perdere l'abilità di controllare le transizioni future. In questo capitolo viene presentata un'estensione di questa classe di giochi al caso in cui i giocatori possono scegliere tra infinite strategie pure, e sia la funzione di payoff sia le probabilità di transizione sono polinomiali.

6.2.1 Descrizione del problema

I giochi stocastici con Switching Control sono giochi in cui le transizioni sono controllate unicamente dal giocatore 1 quando si è in un certo sottoinsieme degli stati, e unicamente dal giocatore 2 quando si è in tutti gli altri stati. La definizione formale è la seguente.

Definizione 37 (Gioco stocastico con Switching Control). Un gioco stocastico $G = (S, F, A, P, r)$ (a due giocatori) ha la proprietà di *Switching*

Control se l'insieme degli stati S può essere partizionato in due insiemi $\mathcal{S}_1$ e $\mathcal{S}_2$, e le probabilità di transizione sono date da

$$\begin{aligned} p(s', s, a_1, a_2) &= p(s', s, a_1), \quad \forall s \in \mathcal{S}_1, s' \in S, a_1 \in A_1(s), a_2 \in A_2(s), \\ p(s', s, a_1, a_2) &= p(s', s, a_2), \quad \forall s \in \mathcal{S}_2, s' \in S, a_1 \in A_1(s), a_2 \in A_2(s). \end{aligned} \tag{6.1}$$

Come risulta ovvio dalla definizione, i giochi stocastici con Switching Control sono una superclasse dei giochi stocastici con Single Controller.

Come è stato anticipato, la funzione di payoff sarà polinomiale nelle variabili a_1 e a_2 con coefficienti reali, con la stessa forma dell'equazione (4.1). Senza perdita di generalità, si assuma che $r(s, a_1, a_2) \geq 0 \, \forall s, a_1, a_2$. Per semplicità, si considera il caso in cui $A_1(s) = A_2(s) = [0, 1] \in \mathbb{R}, \ \forall \, s \in \mathcal{S}$. I risultati si possono facilmente generalizzare al caso in cui gli spazi delle strategie sono unioni finite di intervalli arbitrari dell'asse reale e/o possono variare da stato a stato. Il numero di stati $|S| = N$, rimane finito. Siano $v^*(s)$ i valori degli stati per $s \in \mathcal{S}_1$, e siano $w^*(s)$ i valori degli stati per $s \in \mathcal{S}_2$. Si renumerino gli stati in modo che $\mathcal{S}_1 = \{1, 2, ..., k\}$ e $\mathcal{S}_2 = \{k+1, k+2, ..., N\}$. Inoltre, si assuma che la probabilità di transizione $p(s', s, a_1, a_2)$ sia anch'essa polinomiale nell'azione a_1 in $\mathcal{S}_1$ e nell'azione a_2 in $\mathcal{S}_2$, con coefficienti reali:

$$\begin{aligned} p(s', s, a_1) &= \textstyle\sum_{i=1}^{d_{ss'}} \, p_i(s', s) \, {a_1}^i, \quad \forall s \in \mathcal{S}_1, s' \in S, a_1 \in A_1, a_2 \in A_2, \\ p(s', s, a_2) &= \textstyle\sum_{j=1}^{d_{ss'}} \, p_j(s', s) \, {a_2}^j, \quad \forall s \in \mathcal{S}_2, s' \in S, a_1 \in A_1, a_2 \in A_2. \end{aligned} \tag{6.2}$$

Il processo di decisione opera lungo un orizzonte infinito, quindi risulta naturale restringere l'attenzione alle strategie stazionarie per ogni giocatore, ed utilizzare come metodo di aggregazione la ricompensa scontata. Si supponga infine che i giocatori possano adottare strategie miste, in modo da ricostruire la nozione di equilibrio minimax. La nozione di strategia mista per un giocatore e la rispettiva notazione sono equivalenti a quelle introdotte ed utilizzate nel Capitolo 4. Verranno indicate con μ e ν le strategie rispettivamente del giocatore 1 e 2.

Una strategia μ porta ad una matrice di probabilità di transizione $P(\mu)$ tale che $P_{ss'}(\mu) = \int_{A_1} p(s', s, a_1) \, d\mu(s), \ \forall \, s \in \mathcal{S}_1, s' \in S$. Allo stesso modo, una strategia ν porta ad una matrice di probabilità di transizione $P(\nu)$ tale che $P_{ss'}(\nu) = \int_{A_2} p(s', s, a_2) \, d\nu(s), \ \forall \, s \in \mathcal{S}_2, s' \in S$. In questo modo, una volta che i giocatori fissano una coppia di strategie μ e ν, la matrice di probabilità di transizione $P_{ss'}(\mu, \nu)$ è fissata, e può essere ottenuta integrando ogni elemento

nella matrice rispetto alle misure μ e ν, o più semplicemente, ponendo nelle prime k righe tutte le righe della matrice $P_{ss'}(\mu)$, e nelle righe da $k+1$ ad N tutte le righe della matrice $P_{ss'}(\nu)$.

Date le strategie μ e ν, il payoff atteso immediato del giocatore 1 in un qualche stato s è dato dalla seguente espressione.

$$r(s, \mu(s), \nu(s)) = \int_{A_1} \int_{A_2} r(s, a_1, a_2)\, d\mu(s) d\nu(s)$$

Per alleggerire la notazione, si rinomini $r(s, \mu(s), \nu(s))$ come $r_{s,\mu,\nu}$. Fissati uno stato iniziale s ed una coppia di strategie $\mu(s)$ e $\nu(s)$, il reward collezionato lungo l'orizzonte infinito partendo dallo stato s, ovvero $v_\beta(s, \mu(s), \nu(s))$ se $s \in \mathcal{S}_1$ e $w_\beta(s, \mu(s), \nu(s))$ se $s \in \mathcal{S}_2$, è dato dal sistema di equazioni seguente

$$\begin{aligned} v_\beta\big(s, \mu(s), \nu(s)\big) = r_{s,\mu,\nu} + \beta \textstyle\sum_{s' \in \mathcal{S}_1} \big(\int_{A_1} p(s', s, a_1)\, d\mu(s)\big)\, v_\beta\big(s', \mu(s'), \nu(s')\big) + \\ + \beta \textstyle\sum_{s' \in \mathcal{S}_2} \big(\int_{A_1} p(s', s, a_1)\, d\mu(s)\big)\, w_\beta\big(s', \mu(s'), \nu(s')\big) \quad \forall\, s \in \mathcal{S}_1, \\ w_\beta\big(s, \mu(s), \nu(s)\big) = r_{s,\mu,\nu} + \beta \textstyle\sum_{s' \in \mathcal{S}_1} \big(\int_{A_2} p(s', s, a_2)\, d\nu(s)\big)\, v_\beta\big(s', \mu(s'), \nu(s')\big) + \\ + \beta \textstyle\sum_{s' \in \mathcal{S}_2} \big(\int_{A_2} p(s', s, a_2)\, d\nu(s)\big)\, w_\beta\big(s', \mu(s'), \nu(s')\big) \quad \forall\, s \in \mathcal{S}_2, \end{aligned}$$

dove β è il fattore di discount. Vettorizzando i valori degli stati $v_\beta\big(s, \mu(s), \nu(s)\big)$ e $w_\beta\big(s, \mu(s), \nu(s)\big)$ nel vettore $\mathbf{v}_\beta(\mu, \nu)$, si ottiene:

$$\mathbf{v}_\beta(\mu, \nu) \;=\; (I - \beta P(\mu, \nu))^{-1} \mathbf{r}(\mu, \nu),$$

dove $\mathbf{r}(\mu, \nu) \;=\; [r(1, \mu(1), \nu(1)), ..., r(N, \mu(N), \nu(N))] \;\in\; \mathbb{R}^N$.

6.2.2 Strategie di Equilibrio

Nella Sezione 4.2.2 è stata discussa l'esistenza e l'unicità degli equilibri stazionari nei giochi stocastici a due giocatori e somma zero, con uno spazio degli stati finito, uno spazio di strategie infinito e payoffs polinomiali, mostrando che gli equilibri stazionari esistono sempre e che il vettore dei valori è unico. A maggior ragione, quindi, si sa che gli stessi risultati di esistenza e unicità si hanno anche nei giochi stocastici con Switching Control.

L'obiettivo è quindi quello di trovare la coppia di strategie d'equilibrio (μ^0, ν^0), il cui vettore dei valori del gioco soddisfi la condizione del punto di sella:

$$\mathbf{v}_\beta(\mu, \nu^0) \;\leq\; \mathbf{v}_\beta(\mu^0, \nu^0) \;\leq\; \mathbf{v}_\beta(\mu^0, \nu)$$

per ogni vettore di strategie miste μ e ν.

6.3 Caso con Spazi di Strategie Finiti

Come è stato fatto con i giochi stocastici polinomiali con Single Controller, viene ora mostrato come calcolare le soluzioni di questo tipo di giochi iniziando dal caso in cui ogni giocatore, per ogni stato, ha un numero finito di strategie pure tra cui scegliere. Una trattazione dettagliata di questo caso la si può trovare in [32].

Quando si ha un numero finito di strategie pure e viene mantenuta la condizione di Switching Control, è possibile calcolare una soluzione minimax attraverso un algoritmo che risolve iterativamente due problemi di programmazione lineare. Viene quindi mostrato in questa sezione come utilizzare tale algoritmo per risolvere giochi di questa classe. Nella prossima sezione, partendo concettualmente da questi due problemi di programmazione lineare, verrà mostrato un algoritmo per il caso in cui ogni giocatore possa scegliere tra un numero infinito di strategie pure.

Per semplicita, si assuma ancora che gli insiemi di strategie pure disponibili ad ogni giocatore in ogni stato siano identici ($A_1 = A_2 = \{1, ..., m\}$). Si partizioni lo spazio degli stati $S = \{1, ..., N\}$ in $\mathcal{S}_1 = \{1, ..., k\}$ e $\mathcal{S}_2 = \{k+1, ..., N\}$, in modo che le probabilità di transizione rispettino ancora l'espressione (6.1). Si definisca inoltre con β il fattore di discount. Si considerino le strategie miste $\mathbf{f}$ e $\mathbf{g}$ rispettivamente per i giocatori 1 e 2, come definite nella Sezione 4.3, in modo che il payoff immediato del giocatore 1 in un qualche stato s sia dato da:

$$r\big(s, f(s), g(s)\big) = \sum_{a_1 \in A_1, a_2 \in A_2} r(s, a_1, a_2) f(s, a_1) g(s, a_2).$$

Per alleggerire la notazione, si rinomini $r(s, f(s), g(s))$ come $r_{s,f,g}$. Il reward collezionato sull'orizzonte infinito partendo dallo stato s è dato dal sistema di equazioni:

$$\begin{aligned} v_\beta\big(s, f(s), g(s)\big) = r_{s,f,g} + \beta \textstyle\sum_{s' \in \mathcal{S}_1} \big(\sum_{A_1} p(s', s, a_1) f(s, a_1) \big) v_\beta\big(s', f(s'), g(s')\big) + \\ + \beta \textstyle\sum_{s' \in \mathcal{S}_2} \big(\sum_{A_1} p(s', s, a_1)\, f(s, a_1) \big)\, w_\beta\big(s', f(s'), g(s')\big) \quad \forall\, s \in \mathcal{S}_1, \\ w_\beta\big(s, f(s), g(s)\big) = r_{s,f,g} + \beta \textstyle\sum_{s' \in \mathcal{S}_1} \big(\sum_{A_2} p(s', s, a_2) g(s, a_2) \big) v_\beta\big(s', f(s'), g(s')\big) + \\ + \beta \textstyle\sum_{s' \in \mathcal{S}_2} \big(\sum_{A_2} p(s', s, a_2)\, g(s, a_2) \big)\, w_\beta\big(s', f(s'), g(s')\big) \quad \forall\, s \in \mathcal{S}_2. \end{aligned}$$

L'obiettivo è perciò quello di trovare le strategie di equilibrio $\mathbf{f}^0$ e $\mathbf{g}^0$ che soddisfino la condizione del punto di sella (4.3).

L'algoritmo utilizza quattro problemi di programmazione lineare, di cui due sono primali e due sono i corrispondenti duali. In ognuno dei due LP primali si cercano le strategie di un giocatore negli stati in cui le transizioni sono governate dal giocatore avversario, mentre nei duali si cercano le strategie di un giocatore negli stati in cui egli stesso governa le transizioni.

Si supponga di avere delle stime arbitrarie di $\hat{v}$, ovvero $\hat{v}(1), \hat{v}(2), ..., \hat{v}(k)$. Si consideri quindi il seguente problema di programmazione lineare.

$$\max_{f(s,a_1),w(s)} \sum_{s=k+1}^{N} w(s), \quad \textit{subject to}$$

$$\begin{aligned} &\textstyle\sum_{a_1 \in A_1} r(s,a_1,a_2) f(s,a_1) + \beta \sum_{s'=k+1}^{N} p(s',s,a_2) w(s') + \\ &\quad -w(s) \geq -\beta \textstyle\sum_{s'=1}^{k} p(s',s,a_2) \hat{v}(s'), \quad \forall\, k+1 \leq s \leq N, a_2 \in A_2 \end{aligned} \tag{LP3}$$

$$\textstyle\sum_{a_1 \in A_1} f(s,a_1) = 1, \quad \forall\, k+1 \leq s \leq N,$$

$$f(s,a_1) \geq 0, \qquad \forall\, k+1 \leq s \leq N, a_1 \in A_1$$

Da [32], è noto che tale LP ha una soluzione ottima. Come sarà poi dimostrato, questo problema di programmazione lineare permette di calcolare i valori del gioco e la strategia del giocatore 1 in tutti gli stati in cui le transizioni sono governate dal giocatore 2, supponendo che negli stati in cui le transizioni sono governate dal giocatore 1 i valori del gioco siano fissati, e pari a $\hat{v}(s)$. Si supponga di avere delle stime arbitrarie di $\hat{w}$, ovvero $\hat{w}(k+1), \hat{w}(k+2), ..., \hat{w}(N)$. Si consideri anche il seguente problema di programmazione lineare.

$$\min_{g(s,a_2),v(s)} \sum_{s=1}^{k} v(s), \quad \textit{subject to}$$

$$\begin{aligned} &\textstyle\sum_{a_2 \in A_2} r(s,a_1,a_2) g(s,a_2) + \beta \sum_{s'=1}^{k} p(s',s,a_1) v(s') + \\ &\quad -v(s) \leq -\beta \textstyle\sum_{s'=k+1}^{N} p(s',s,a_1) \hat{w}(s'), \quad \forall\, 1 \leq s \leq k, a_1 \in A_1 \end{aligned} \tag{LP4}$$

$$\textstyle\sum_{a_2 \in A_2} g(s,a_2) = 1, \quad \forall\, 1 \leq s \leq k,$$

$$g(s,a_2) \geq 0, \qquad \forall\, 1 \leq s \leq k, a_2 \in A_2$$

Sempre da [32], è noto che anche tale LP ha una soluzione ottima. In analogia con il problema (LP3), questo problema di programmazione lineare permette di calcolare i valori del gioco e la strategia del giocatore 2 in tutti gli stati in cui le transizioni sono governate dal giocatore 1, supponendo che negli stati in cui le transizioni sono governate dal giocatore 2 i valori del gioco siano fissati, e pari a $\hat{w}(s)$. Risulta quindi di fondamentale importanza il seguente teorema.

Teorema 17 ([32])**.** Si supponga di avere le seguenti due coppie di vettori di valori del gioco e strategie:

$$v(s), g(s, a_2), \quad \forall\, 1 \leq s \leq k, a_2 \in A_2,$$
$$w(s), f(s, a_1), \quad \forall\, k+1 \leq s \leq N, a_1 \in A_1.$$

dove $\big(v(s), g(s, a_2),\ \forall\, 1 \leq s \leq k, a_2 \in A_2\big)$ è ottima per il problema (LP4) quando $\hat{w}(t) = w(t), k+1 \leq t \leq N$, e $\big(w(s), f(s, a_1), \forall\, k+1 \leq s \leq N, a_1 \in A_1\big)$ è ottima per il problema (LP3) quando $\hat{v}(s) = w(s),\ 1 \leq s \leq k$.
Allora $\big(v(1), ..., v(k), w(k+1), ..., w(N)\big)$ è il vettore dei valori del gioco.

A questo punto può essere descritto l'algoritmo che permette di calcolare le strategie ottime ed i valori del gioco per ogni stato e per ogni giocatore. Sia $\hat{w}(t) = 0 = w^0(t),\ \forall\, k+1 \leq t \leq N$ e si consideri il problema (LP4). Sia $(v^0(s), g^0(s))$ la soluzione ottima di questo problema, con G_0 come base ottima. Si risolva quindi il problema (LP3) con $\hat{v}(t) = v^0(t), \forall\, 1 \leq s \leq k$. Sia $(w^1(t), f^1(t))$ la soluzione ottima di tale problema, con F_1 come base corrispondente. Si torni ora al problema (LP4), con $\hat{w}(t) = w^1(t),\ \forall\, k+1 \leq t \leq N$. Si può notare che la base G_0 soddisfa ancora la condizione di ottimalità, ma la soluzione $(v^0(s), g^0(s))$ non è più ammissibile. Si utilizza allora l'algoritmo del simplesso duale per ottenere $(v^1(s), g^1(s))$ e G_1. Similarmente, si riconsidera il problema (LP3) con $\hat{v}(t) = v^1(t), \forall\, 1 \leq s \leq k$, e si esegue ancora l'ottimizzazione utilizzando il simplesso duale per ottenere $(w^2(t), f^2(t)), \forall\, k+1 \leq s \leq N$, e la base ottima F_2. L'algoritmo ripete quindi questi passi, generando una sequenza di basi G_r, F_r con associate le soluzioni (v^r, w^r) di (LP3) e (LP4). Avendo tale sequenza, si ha il seguente risultato.

Teorema 18 ([32])**.**

$$\lim_{r \to \infty} (v^r, w^r) = (v^*, w^*).$$

Questo risultato permette di garantire che l'algoritmo termini in un numero finito di passi. Infatti, ogni volta che una coppia di basi G, F si ripete, si controlla se la coppia è ammissibile per le disuguaglianze lineari di (LP3) e (LP4) messe assieme. Se è ammissibile, allora porta al vettore dei valori del gioco (v^*, w^*) e ad una coppia di strategie ottime. Altrimenti si continua con l'algoritmo. Fintanto esiste un numero finito di coppie di basi, l'algoritmo termina in un numero finito di passi.

Per completezza, vengono mostrati anche i due problemi duali. Il problema di programmazione lineare (DP3) permette di calcolare la strategia del giocatore 2 negli stati $s \in \mathcal{S}_2$, mentre il problema di programmazione lineare (DP4) permette di calcolare la strategia del giocatore 1 negli stati stati $s \in \mathcal{S}_1$.

$$
\begin{aligned}
&\max_{g(s,a_2),z(s)} \sum_{s=k+1}^{N} \big(z(s) - \beta g(s,a_2) \textstyle\sum_{a_2\in A_2} \sum_{s'=1}^{k} p(s',s,a_2)\hat{v}(s')\big), \quad s.t. \\
&\textstyle\sum_{s=k+1}^{N} \sum_{a_2\in A_2} \left[\delta(s,s') - \beta p(s,s',a_2)\right] g(s',a_2) = 1 \quad \forall\, s' \in \mathcal{S}_2 \\
&z(s) \le -\textstyle\sum_{a_2\in A_2} g(s,a_2)\, r(s,a_1,a_2) \quad \forall\, s \in \mathcal{S}_2,\, a_1 \in A_1 \\
&g(s,a_2) \ge 0 \quad \forall\, s \in \mathcal{S}_2,\, a_2 \in A_2
\end{aligned}
\tag{DP3}
$$

$$
\begin{aligned}
&\max_{f(s,a_1),z(s)} \sum_{s=1}^{k} \big(z(s) + \beta f(s,a_1) \textstyle\sum_{a_1\in A_1} \sum_{s'=k+1}^{N} p(s',s,a_1)\hat{w}(s')\big), \quad s.t. \\
&\textstyle\sum_{s=1}^{k} \sum_{a_1\in A_1} \left[\delta(s,s') - \beta p(s,s',a_1)\right] f(s',a_1) = 1 \quad \forall\, s' \in \mathcal{S}_1 \\
&z(s) \le \textstyle\sum_{a_1\in A_1} f(s,a_2)\, r(s,a_1,a_2) \quad \forall\, s \in \mathcal{S}_1,\, a_2 \in A_2 \\
&f(s,a_1) \ge 0 \quad \forall\, s \in \mathcal{S}_1,\, a_1 \in A_1
\end{aligned}
\tag{DP4}
$$

6.4 Caso con Spazi di Strategie Infiniti

In questa sezione si considera il caso in cui ogni giocatore possa scegliere tra infinite azioni non numerabili, presentando un algoritmo per il calcolo del valore

degli stati e delle strategie ottime per i giocatori. Per ottenere tale algoritmo, si inizierà generalizzando i problemi (LP3), (LP4), (DP3) e (DP4) al caso di payoff polinomiali rispetto alle azioni a_1 ed a_2 continue. Questa generalizzazione porterà ad ottenere quattro problemi di programmazione semidefinita. A questo punto verrà descritto l'algoritmo che, iterando tra i quattro SDP ottenuti, converge alla soluzione ottima.

6.4.1 Azioni Infinite ed Ottimizzazione Polinomiale

Si consideri il caso di gioco a due giocatori e somma zero in cui ogni giocatore può, in ogni stato, scegliere azioni continue nell'intervallo $[0, 1]$, mentre il numero di stati $|S| = N$ rimane ancora finito. La funzione di payoff $r(s, a_1, a_2)$ è come quella indicata in (4.1) per ogni $s \in S$. Inoltre, poichè si assume che sia soddisfatta la condizione di Switching Control, la probabilità di transizione $p(s', s, a_1)$ è come quella indicata in (6.1). Le variabili $\mathbf{f}$ e $\mathbf{g}$, che rappresentavano le distribuzioni sugli insiemi finiti A_1 e A_2, sono quindi rimpiazzate dalle misure di probabilità $\mu(s)$ e $\nu(s)$, che rappresentano le strategie miste su uno spazio di azioni non numerabile. L'interpretazione fatta in termini di strategie sicure, che veniva mantenuta nel caso dei giochi stocastici con single controller, viene anche qui mantenuta, ma con un senso più debole. Infatti, mentre nel caso di single controller era possibile con un unico SDP calcolare la strategia sicura ottima per il giocatore 2, in questo caso, dovendo utilizzare un algoritmo iterativo tra due problemi di ottimizzazione, si troveranno nelle varie iterazioni le strategie sicure ipotizzando che il valore di alcuni stati sia noto. È importante anche notare che, in questo caso, risolvendo i problemi primali verranno trovate le strategie per entrambi i giocatori negli stati controllati rispettivamente dall'avversario.

Viene ora mostrato che una generalizzazione dei problemi di programmazione lineare problemi (LP3) e (LP4) a questo caso, porta a due problemi di ottimizzazione che riguardano la non negatività di due sistemi di polinomi univariati, con coefficienti che dipendono dai momenti di queste misure.

Supponendo di avere delle stime arbitrarie di $\hat{v}(s)$ $\forall\, s \in \mathcal{S}_1$, ovvero $\hat{\mathbf{v}} = [\hat{v}(1), \hat{v}(2), ..., \hat{v}(k)]^T$, si consideri il seguente problema di ottimizzazione che generalizza il problema di programmazione lineare (LP3).

$$\max_{\mu(s),w(s)} \sum_{s=k+1}^{N} w(s), \quad \textit{subject to}$$

$$\begin{aligned}
(a) \quad & \int_{a_1 \in A_1(s)} r(s,a_1,a_2)\mu(s,a_1)\,da_1 + \beta \sum_{s'=k+1}^{N} p(s',s,a_2)w(s') + \\
& +\beta \sum_{s'=1}^{k} p(s',s,a_2)\hat{v}(s') \geq \ w(s), \ \ \forall\, s \in \mathcal{S}_2,\ a_2 \in A_2
\end{aligned} \tag{P3}$$

$$(b) \qquad \int_{a_1 \in A_1} \mu(s,a_1)\,da_1 = 1, \qquad \forall\, s \in \mathcal{S}_2$$

$$(c) \qquad \mu_{a_1}(s) \geq 0, \qquad \forall\, s \in \mathcal{S}_2,\ a_1 \in A_1$$

Poichè si ha $\int_{A_1} r(s,a_1,a_2)\mu(s,a_1)\,da_1 \ = \ t_\mu(s,a_2)$, dove $t_\mu(s,a_2)$ è un polinomio univariato in a_2 per ogni $s \in \mathcal{S}_2$, fissato un vettore $\mu(s)$, il vincolo (a) è un sistema di disequazioni polinomiali; inoltre i coefficienti di t dipendono dalla misura μ solo attraverso un numero finito di momenti. Più concretamente, si ha la seguente espressione.

$$\int_{a_1 \in A_1} r(s,a_1,a_2)\,\mu(s,a_1)\,da_1 \ = \ \sum_{i=0}^{n_s}\sum_{j=0}^{m_s} r_{ij}(s)\,\mu_i(s)\,{a_2}^j$$

Utilizzando questa osservazione, il problema di ottimizzazione (P3) può essere riscritto nel seguente modo.

$$\max_{\mu(s),w(s)} \sum_{s=k+1}^{N} w(s), \quad \textit{subject to}$$

$$\begin{aligned}
(a) \quad & \sum_{i}^{n_s}\sum_{j}^{m_s} r_{i,j}(s)\mu_i(s)a_2^j + \beta \sum_{s'=k+1}^{N} p(s',s,a_2)w(s') + \\
& +\beta \sum_{s'=1}^{k} p(s',s,a_2)\hat{v}(s') - w(s) \ \in \mathcal{P}_n(A_2),\ \forall\, s \in \mathcal{S}_2
\end{aligned} \tag{P3$'$}$$

$$(b) \qquad \bar{\mu}(s) \in M(A_1), \quad \forall\, s \in \mathcal{S}_2$$

$$(c) \qquad \mu_0(s) = 1,\ \forall\, s \in \mathcal{S}_2$$

Supponendo ora di avere invece delle stime arbitrarie di $\hat{w}(s)\ \forall\, s\ \in\ \mathcal{S}_2$, ovvero $\hat{\mathbf{w}} = [\hat{w}(k+1),\, \hat{w}(k+2), ..., \hat{w}(N)]^T$, si consideri il seguente problema di ottimizzazione che generalizza il problema di programmazione lineare (LP4).

$$\min_{\nu(s),v(s)} \sum_{s=1}^{k} v(s), \quad \textit{subject to}$$

$$\begin{aligned} (d) \quad & \int_{a_2 \in A_2(s)} r(s,a_1,a_2)\nu(s,a_2)\, da_2 + \beta \sum_{s'=1}^{k} p(s',s,a_1)v(s') + \\ & +\beta \sum_{s'=k+1}^{N} p(s',s,a_1)\hat{w}(s') \leq\ v(s), \ \ \forall\, s \in \mathcal{S}_1,\ a_1 \in A_1 \end{aligned} \tag{P4}$$

$$(e) \qquad \int_{a_2 \in A_2} \nu(s,a_2)\, da_2 = 1, \qquad \forall\, s \in \mathcal{S}_1$$

$$(f) \qquad \nu_{a_2}(s) \geq 0, \qquad \forall\, s \in \mathcal{S}_1,\ a_2 \in A_2$$

In modo analogo a quanto fatto per il problema (P3), considerando il vincolo (d), si ha la seguente espressione.

$$\int_{a_2 \in A_2} r(s,a_1,a_2)\, \nu(s,a_2)\, da_2\ =\ \sum_{i=0}^{n_s} \sum_{j=0}^{m_s} r_{ij}(s)\, \nu_j(s)\, {a_1}^i \tag{6.3}$$

Utilizzando questa espressione, il problema di ottimizzazione (P4) può essere riscritto nel seguente modo.

$$\min_{\nu(s),v(s)} \sum_{s=1}^{k} v(s), \quad \textit{subject to}$$

$$\begin{aligned} (d) \quad & v(s) - \sum_{i}^{n_s} \sum_{j}^{m_s} r_{i,j}(s)\nu_j(s)a_1^i - \beta \sum_{s'=1}^{k} p(s',s,a_1)v(s') + \\ & -\beta \sum_{s'=k+1}^{N} p(s',s,a_1)\hat{w}(s')\ \in \mathcal{P}_n(A_1),\ \forall\, s \in \mathcal{S}_1 \end{aligned} \tag{P4$'$}$$

$$(e) \qquad \bar{\nu}(s) \in M(A_2), \quad \forall\, s \in \mathcal{S}_1$$

$$(f) \qquad \nu_0(s) = 1,\ \forall\, s \in \mathcal{S}_1$$

6.4.2 Da Ottimizzazione Polinomiale ad SDP

La formulazione dei problemi di ottimizzazione (P3′) e (P4′) è astratta e non risolvibile: anche in questo caso, per cercare di convertirla in un problema di ottimizzazione concreto che si possa risolvere, è necessaria la rappresentazione computazionalmente adatta degli insiemi $\mathcal{P}(A_1), \mathcal{P}(A_2), \mathcal{M}(A_1)$, e $\mathcal{M}(A_2)$, introdotta nella Sezione 3.3. Come è noto dalla Sezione 3.3.1, le condizioni di non negatività di un polinomio su un intervallo si applicano ai coefficienti di tale polinomio. Si considerino i vincoli (a) e (d), che producono due sistemi di disuguaglianze polinomiali, uno in a_2 ed uno in a_1, in cui in entrambi i sistemi vi è una disuguaglianza per ogni stato. Fissati uno stato $s \in \mathcal{S}_1$ ed uno stato $t \in \mathcal{S}_2$, si cerca di esplicitare i coefficienti dei due polinomi seguenti:

$$\begin{aligned} y_t(a_2) &= \textstyle\sum_i^{n_t} \sum_j^{m_t} r_{i,j}(t)\mu_i(t)a_2^j + \beta \sum_{s'=k+1}^{N} p(s',t,a_2)w(s') + \\ &\quad + \beta \textstyle\sum_{s'=1}^{k} p(s',t,a_2)\hat{v}(s') - w(t), \\ x_s(a_1) &= v(s) - \textstyle\sum_i^{n_s} \sum_j^{m_s} r_{i,j}(s)\nu_j(s)a_1^i - \beta \sum_{s'=1}^{k} p(s',s,a_1)v(s') + \\ &\quad - \beta \textstyle\sum_{s'=k+1}^{N} p(s',s,a_1)\hat{w}(s'). \end{aligned}$$

Sia d_s il grado della disuguaglianza per lo stato s, e sia d_t il grado della disuguaglianza per lo stato t. Si considerino i due seguenti vettori:

$$[a_1]_{d_t} = [1, a_1, {a_1}^2, ..., {a_1}^{d_t}]^T, \qquad [a_2]_{d_s} = [1, a_2, {a_2}^2, ..., {a_2}^{d_s}]^T$$

Il primo termine del polinomio $y_t(a_2)$ ed il secondo termine del polinomio $x_s(a_1)$ possono essere riscritti in forma vettoriale come:

$$\sum_{i=0}^{n_t} \sum_{j=0}^{m_t} r_{ij}(t)\, \mu_i(t)\, {a_2}^j \;=\; \bar{\mu}(t)^T R(t) [a_2]_{d_t} \tag{6.4}$$

$$\sum_{i=0}^{n_s} \sum_{j=0}^{m_s} r_{ij}(s)\, \nu_j(s)\, {a_1}^i \;=\; \bar{\nu}(s)^T R(s)^T [a_1]_{d_s} \tag{6.5}$$

dove $R(s)$ è la matrice che contiene i coefficienti del polinomio $r(s, a_1, a_2)$, $\bar{\mu}(t) \in \mathbb{R}^{d_t+1}$ è il vettore dei primi $d_t + 1$ momenti della misura $\mu(t)$ e $\bar{\nu}(s) \in \mathbb{R}^{d_s+1}$ è il vettore dei primi $d_s + 1$ momenti della misura $\nu(s)$.

Si considerino $\mathbf{v}^* = [v^*(1), ..., v^*(k)]^T$ e $\mathbf{w}^* = [w^*(k+1), ..., w^*(N)]^T$, che risulteranno essere i vettori dei valori del gioco stocastico (indicizzati dallo stato). Si considerino i termini dei polinomi $y_t(a_2)$ e $x_s(a_1)$ che dipendono dalla probabilità di transizione $p(s', s, a_1, a_2)$: tali termini sono a loro volta dei

polinomi univariati rispettivamente in a_2 e a_1 i cui coefficienti dipendono nel caso di $y_t(a_2)$ dai coefficienti di $p(s', s, a_2)$, $\hat{\mathbf{v}}$ e $\mathbf{w}$, mentre nel caso di $x_s(a_1)$ dai coefficienti di $p(s', s, a_1)$, $\hat{\mathbf{w}}$ e $\mathbf{v}$. Specificatamente, si ha:

$$\begin{aligned}
\textstyle\sum_{s'=k+1}^{N} p(s', t, a_2) w(s') &= \mathbf{w}^T Q_{22}(t)^T [a_2]_{d_s}, \\
\textstyle\sum_{s'=1}^{k} p(s', t, a_2) \hat{v}(s') &= \hat{\mathbf{v}}^T Q_{21}(t)^T [a_2]_{d_s}, \\
\textstyle\sum_{s'=1}^{k} p(s', s, a_1) v(s') &= \mathbf{v}^T Q_{11}(s)^T [a_1]_{d_t}, \\
\textstyle\sum_{s'=k+1}^{N} p(s', s, a_1) \hat{w}(s') &= \hat{\mathbf{w}}^T Q_{12}(s)^T [a_1]_{d_t},
\end{aligned} \tag{6.6}$$

dove

- $Q_{11}(s)$ è la matrice che contiene i coefficienti della probabilità di transizione $p(s', s, a_1)$ con $s \in \mathcal{S}_1$, $\forall\, s' \in \mathcal{S}_1$.
- $Q_{12}(s)$ è la matrice che contiene i coefficienti della probabilità di transizione $p(s', s, a_1)$ con $s \in \mathcal{S}_1$, $\forall\, s' \in \mathcal{S}_2$.
- $Q_{21}(s)$ è la matrice che contiene i coefficienti della probabilità di transizione $p(s', s, a_2)$ con $s \in \mathcal{S}_2$, $\forall\, s' \in \mathcal{S}_1$.
- $Q_{22}(s)$ è la matrice che contiene i coefficienti della probabilità di transizione $p(s', s, a_2)$ con $s \in \mathcal{S}_2$, $\forall\, s' \in \mathcal{S}_2$.

A questo punto è possibile combinare le equazioni (6.4), (6.5) ed (6.6) per riscrivere le disuguaglianze polinomiali che compongono i sistemi di disuguaglianze dei vincoli (a) e (d): poichè gli spazi delle azioni sono $A_1 = A_2 = [0, 1]$, dal Lemma 4 nella Sezione 3.3.1, si può concludere che i polinomi $y_t(a_1)$ e $x_s(a_2)$ sono non negativi in $[0, 1]$ se e solo se esistono le matrici $Z_t \in \mathcal{S}^{d_t+1}, Z_s \in \mathcal{S}^{d_s+1}, W_t \in \mathcal{S}^{d_t}, W_s \in \mathcal{S}^{d_s}$ con $Z_t, Z_s, W_t, W_s \succeq 0$, tali che

$$\begin{aligned}
&\mathcal{H}^*(Z_t + \tfrac{1}{2}(L_1 W_t L_2^T + L_2 W_t L_1^T) - L_2 W_t L_2^T) = \\
&\qquad = \bar{\mu}(t)^T R(t) + \beta\big(\mathbf{w}^T Q_{22}(t) + \hat{\mathbf{v}}^T Q_{21}(t)\big) - e_t \mathbf{w}
\end{aligned} \tag{6.7}$$

$$\begin{aligned}
&\mathcal{H}^*(Z_s + \tfrac{1}{2}(L_1 W_s L_2^T + L_2 W_s L_1^T) - L_2 W_s L_2^T) = \\
&\qquad = e_s \mathbf{v} - \bar{\nu}(s)^T R(s)^T - \beta\big(\mathbf{v}^T Q_{11}(s)^T + \hat{\mathbf{w}}^T Q_{12}(s)^T\big)
\end{aligned} \tag{6.8}$$

dove $e_t \in \mathbb{R}^{d_t+1}, e_s \in \mathbb{R}^{d_s+1}$ sono vettori che contengono tutti elementi a zero eccetto il primo elemento ad uno. Si cerca ora di riscrivere i vincoli (b)

ed (e) dei problemi (P3′) e (P4′), che riguardano i momenti delle misure di probabilità.

Siano ancora fissati uno stato $s \in \mathcal{S}_1$ ed uno stato $t \in \mathcal{S}_2$. Grazie al Lemma 9 nella Sezione 3.3.2, è possibile concludere che $\bar{\mu}(t) \in \mathcal{M}(A_1)$ e $\mu_0(t) = 1$ con $[c,d] = [0,1]$ se e solo se

$$\begin{aligned} {e_t}^T \bar{\mu}(t) &= 1, \\ \mathcal{H}(\bar{\mu}(t)) &\succeq 0, \\ \tfrac{1}{2}(L_1^T \mathcal{H}(\bar{\mu}(t)) L_2 + L_2^T \mathcal{H}(\bar{\mu}(t)) L_1) - L_2^T \mathcal{H}(\bar{\mu}(t)) L_2 &\succeq 0. \end{aligned} \tag{6.9}$$

Sempre grazie al Lemma 9, è possibile concludere che $\bar{\nu}(s) \in \mathcal{M}(A_2)$ e $\nu_0(s) = 1$ con $[c,d] = [0,1]$ se e solo se

$$\begin{aligned} {e_s}^T \bar{\nu}(s) &= 1, \\ \mathcal{H}(\bar{\nu}(s)) &\succeq 0, \\ \tfrac{1}{2}(L_1^T \mathcal{H}(\bar{\nu}(s)) L_2 + L_2^T \mathcal{H}(\bar{\nu}(s)) L_1) - L_2^T \mathcal{H}(\bar{\nu}(s)) L_2 &\succeq 0. \end{aligned} \tag{6.10}$$

Prima di rimettere insieme tutte queste condizioni è importante notare che, come discusso nella Sezione 4.4.3 per il caso dei giochi stocastici con Single Controller, anche nel caso dei giochi a Switching Control risulta necessario aggiungere alcuni vincoli di programmazione semidefinita per calcolare i momenti "mancanti" necessari alla ricostruzione delle strategie dei giocatori. In particolare, grazie al Lemma 7, è noto che sarà sufficiente aggiungere i due vincoli

$$\begin{aligned} \mathcal{H}\big([\mu_0,(t)\mu_1(t),...,\mu_{d_t}(t)]\big) - \mathcal{H}\big([\mu_1(t),\mu_2(t),...,\mu_{d_t+1}(t)]\big) &\succeq 0, \\ \mathcal{H}\big([\mu_1(t),\mu_2(t),...,\mu_{d_t+1}(t)]\big) &\succeq 0 \end{aligned} \tag{6.11}$$

all'SDP che concretizza il problema (P3′), per ottenere una misura non negativa in $[0,1]$ con momenti $(\mu_0(t),\mu_1(t),...,\mu_{d_t+1}(t))$. Analogamente, sarà sufficiente aggiungere i due vincoli

$$\begin{aligned} \mathcal{H}\big([\nu_0,(s)\nu_1(s),...,\nu_{d_s}(s)]\big) - \mathcal{H}\big([\nu_1(s),\nu_2(s),...,\nu_{d_s+1}(s)]\big) &\succeq 0, \\ \mathcal{H}\big([\nu_1(s),\nu_2(s),...,\nu_{d_s+1}(s)]\big) &\succeq 0 \end{aligned} \tag{6.12}$$

all'SDP che concretizza il problema (P4′), per ottenere una misura non negativa in $[0,1]$ con momenti $(\nu_0(s),\nu_1(s),...,\nu_{d_s+1}(s))$.

È quindi possibile utilizzare le espressioni (6.7), (6.9) ed (6.11) per formare il singolo problema di programmazione semidefinita positiva che concretizza il

problema di ottimizzazione astratto (P3′), ottenendo quindi il seguente SDP.

$$\max_{\bar{\mu}(t),\mu_{n_t+1}(t),w(t),Z_t,W_t} \sum_{t=k+1}^{N} w(t), \quad \textit{subject to}$$

$$(a) \qquad \begin{aligned} &\mathcal{H}^*(Z_t + \tfrac{1}{2}(L_1 W_t L_2^T + L_2 W_t L_1^T) - L_2 W_t L_2^T = \\ &= R(t)^T \bar{\mu}(t) + \beta\big(Q_{22}(t)\mathbf{w} + Q_{21}(t)\hat{\mathbf{v}}\big) - e_t \mathbf{w}, \quad \forall\, t \in \mathcal{S}_2 \end{aligned}$$

$$(b) \qquad \mathcal{H}(\bar{\mu}(t)) \succeq 0, \quad \forall\, t \in \mathcal{S}_2$$

$$(c) \qquad \begin{aligned} &\tfrac{1}{2}(L_1^T \mathcal{H}(\bar{\mu})(t) L_2 + L_2^T \mathcal{H}(\bar{\mu})(t) L_1) + \\ &- L_2^T \mathcal{H}(\bar{\mu})(t) L_2 \succeq 0, \quad \forall\, t \in \mathcal{S}_2 \end{aligned}$$

$$(d) \qquad e_t^T \bar{\mu}(t) = 1, \quad \forall\, t \in \mathcal{S}_2$$

$$(e) \qquad \mathcal{H}([\mu_1(t),\, \mu_2(t),\, ...,\, \mu_{n_t+1}(t)]^T) \succeq 0, \quad \forall\, t \in \mathcal{S}_2$$

$$(f) \qquad \begin{aligned} &\mathcal{H}([\mu_0(t),\, \mu_1(t),\, ...,\, \mu_{n_t}(t)]^T) - \mathcal{H}([\mu_1(t),\, \mu_2(t),\, ...,\, \mu_{n_t+1}(t)]^T) \succeq 0, \\ &\forall\, t \in \mathcal{S}_2 \end{aligned}$$

$$(g) \qquad Z_t, W_t \succeq 0, \quad \forall\, t \in \mathcal{S}_2$$

(SP3)

dove si ricorda che $\bar{\mu}(t) \in \mathbb{R}^{n_t+1}$, ovvero $\bar{\mu}(t)$ non contiene il momento di ordine $n_t + 1$. A questo punto, possiamo esprimere il seguente Lemma.

Lemma 16. Siano $A_1 = A_2 = [0, 1]$. Il problema di programmazione semidefinita (SP3) risolve esattamente il problema di ottimizzazione polinomiale (P3′).

Dimostrazione. La disuguaglianza polinomiale (a) nel problema (P3′) ha il vettore dei coefficienti $R(t)^T \bar{\mu}(t) + \beta\big(Q_{22}(t)\mathbf{w} + Q_{21}(t)\hat{\mathbf{v}}\big) - e_t \mathbf{w}$, come mostrato dalle equazioni (6.4) e (6.6). La dimostrazione si ha quindi per diretta conseguenza del Lemma 4 riguardante la rappresentazione semidefinita dei polinomi non negativi su $[0, 1]$, come mostrato nell'equazione (6.7), ed il Lemma 9 riguardante la rappresentazione semidefinita delle sequenze di momenti di misure non negative supportate su $[0, 1]$, come mostrato nell'espressione (6.9). □

La soluzione di questo SDP produce:

- il vettore dei valori del gioco $\mathbf{w}$ degli stati in cui le transizioni sono controllate dal giocatore 2,
- i momenti delle misure $\mu(t)$ per il giocatore 1 (una per ogni stato t), negli stati cui le transizioni sono controllate dal giocatore 2.

Una volta ottenuti i momenti delle misure, utilizzando la procedura indicata nella Sezione 3.3.2 è possibile ricostruire il supporto (finito) ed i pesi della strategia del giocatore 1, ricavando una coppia supporto-pesi per ogni stato in cui le transizioni sono governate dal giocatore 2.

In modo analogo, è possibile utilizzare le espressioni (6.8), (6.10) ed (6.12) per formare il singolo SDP che concretizza il problema (P4′):

$$\min_{\bar{\nu}(s),\nu_{m_s+1}(s),v(s),Z_s,W_s} \sum_{s=1}^{k} v(s), \quad \textit{subject to}$$

$$(h) \qquad \begin{aligned} &\mathcal{H}^*(Z_s + \tfrac{1}{2}(L_1 W_s L_2^T + L_2 W_s L_1^T) - L_2 W_s L_2^T = \\ &= e_s \mathbf{v} - R(s)\bar{\nu}(s) - \beta\big(Q_{11}(s)\mathbf{v} + Q_{12}(s)\hat{\mathbf{w}}\big), \quad \forall\ s \in \mathcal{S}_1 \end{aligned}$$

$$(i) \qquad \mathcal{H}(\bar{\nu}(s)) \succeq 0, \quad \forall\ s \in \mathcal{S}_1$$

$$(j) \qquad \begin{aligned} &\tfrac{1}{2}(L_1^T \mathcal{H}(\bar{\nu})(s) L_2 + L_2^T \mathcal{H}(\bar{\nu})(s) L_1) + \\ &- L_2^T \mathcal{H}(\bar{\nu})(s) L_2 \succeq 0, \quad \forall\ s \in \mathcal{S}_1 \end{aligned}$$

$$(k) \qquad e_s^T \bar{\nu}(s) = 1, \quad \forall\ s \in \mathcal{S}_1$$

$$(l) \qquad \mathcal{H}([\nu_1(s),\, \nu_2(s),\, ...,\, \nu_{m_s+1}(s)]^T) \succeq 0, \quad \forall\ s \in \mathcal{S}_1$$

$$(m) \quad \mathcal{H}([\nu_0(s),\, \nu_1(s),\, ...,\, \nu_{m_s}(s)]^T) - \mathcal{H}([\nu_1(s),\, \nu_2(s),\, ...,\, \nu_{m_s+1}(s)]^T) \succeq 0, \quad \forall\ s \in \mathcal{S}_1$$

$$(n) \qquad Z_s, W_s \succeq 0, \quad \forall\ s \in \mathcal{S}_1 \tag{SP4}$$

dove si ricorda che $\bar{\nu}(s) \in \mathbb{R}^{m_s+1}$, ovvero $\bar{\nu}(s)$ non contiene il momento di ordine $m_s + 1$. A questo punto, è possibile esprimere il seguente lemma.

Lemma 17. Siano $A_1 = A_2 = [0, 1]$. Il problema di programmazione semidefinita (SP4) risolve esattamente il problema di ottimizzazione polinomiale (P4′).

Dimostrazione. La disuguaglianza polinomiale (d) nel problema (P4′) ha il vettore dei coefficienti $e_s\mathbf{v} - R(s)\bar{\nu}(s) - \beta\big(Q_{11}(s)\mathbf{v} + Q_{12}(s)\hat{\mathbf{w}}\big)$, come mostrato dalle equazioni (6.5) e (6.6). La dimostrazione si ha quindi per diretta conseguenza del Lemma 4 riguardante la rappresentazione semidefinita dei polinomi non negativi su $[0, 1]$, come mostrato nell'equazione (6.8), ed il Lemma 9 riguardante la rappresentazione semidefinita delle sequenze di momenti di misure non negative supportate su $[0, 1]$, come mostrato nell'espressione (6.10). □

La soluzione di questo problema di ottimizzazione produce:

- il vettore dei valori del gioco $\mathbf{v}$ degli stati in cui le transizioni sono controllate dal giocatore 1,
- i momenti delle misure $\nu(s)$ per il giocatore 2 (una per ogni stato s), negli stati cui le transizioni sono controllate dal giocatore 1.

Una volta ottenuti i momenti delle misure, utilizzando la procedura indicata nella Sezione 3.3.2 è possibile ricostruire il supporto (finito) ed i pesi della strategia del giocatore 2, ricavando una coppia supporto-pesi per ogni stato in cui le transizioni sono governate dal giocatore 1.

6.4.3 Derivazione degli SDP Duali

Si considerino i duali dei problemi (P3′) e (P4′), che generalizzano i problemi (DP3) e (DP4) al caso di spazi di azioni non numerabili. Siano fissate le stime $\hat{\mathbf{v}} = [\hat{v}(1), \hat{v}(2), ..., \hat{v}(k)]^T$ e $\hat{\mathbf{w}} = [\hat{w}(k+1),\ \hat{w}(k+2), ..., \hat{w}(N)]^T$. Si fissi uno stato $s \in \mathcal{S}_2$. Facendo riferimento alla funzione obiettivo del problema (DP3), nel caso di spazi di azioni non numerabili vale la seguente equazione.

$$\beta \int_{a_2 \in A_2} \sum_{s'=1}^{k} p(s', s, a_2)\hat{v}(s')\nu(s, a_2)\, da_2 =$$

$$= \beta \sum_{s'=1}^{k} \hat{v}(s') \int_{a_2 \in A_2} \sum_{j=0}^{m_s} p_j(s', s) a_2^j \nu(s, a_2)\, da_2 = \beta \sum_{s'=1}^{k} \hat{v}(s') \sum_{j=0}^{m_s} p_j(s', s)\nu_j(s).$$

Utilizzando anche l'equazione (6.3) nella generalizzazione del secondo vincolo del problema (DP3), nel caso di spazi di azioni non numerabili tale problema viene generalizzato con il seguente problema di ottimizzazione:

$$\max_{\nu(s),z(s)} \sum_{s=k+1}^{N} \big(z(s) - \beta \textstyle\sum_{s'=1}^{k} \hat{v}(s') \sum_{j=0}^{m_s} p_j(s',s)\nu_j(s)\big), \quad s.t.$$

$$(a)\ \textstyle\sum_{s=k+1}^{N} \int_{a_2 \in A_2} (\delta(s,s') - \beta p(s',s,a_2))\nu(s,a_2)\, da_2 = 1 \quad \forall\, s' \in \mathcal{S}_2$$

$$(b)\ -z(s) - \textstyle\sum_{i=0}^{n_s} \sum_{j=0}^{m_s} r_{ij}(s) a_1^i \nu_j(s) \in \mathcal{P}_n(A_1) \quad \forall\, s \in \mathcal{S}_2,$$

$$(c)\ \nu(s) \in M(A_2) \quad \forall\, s \in \mathcal{S}_2 \tag{DP3$'$}$$

In modo del tutto analogo, la generalizzazione del problema (DP4) al caso di spazi di azioni non numerabili produce il seguente problema di ottimizzazione:

$$\max_{\mu(s),z(s)} \sum_{s=1}^{k} \big(z(s) + \beta \textstyle\sum_{s'=k+1}^{N} \hat{w}(s') \sum_{i=0}^{n_s} p_i(s',s)\mu_i(s)\big), \quad s.t.$$

$$(d)\ \textstyle\sum_{s=1}^{k} \int_{a_1 \in A_1} (\delta(s,s') - \beta p(s',s,a_1))\mu(s,a_1)\, da_1 = 1 \quad \forall\, s' \in \mathcal{S}_1 \tag{DP4$'$}$$

$$(e)\ \textstyle\sum_{i=0}^{n_s} \sum_{j=0}^{m_s} r_{ij}(s) a_2^j \mu_i(s) - z(s) \in \mathcal{P}_n(A_2) \quad \forall\, s \in \mathcal{S}_1,$$

$$(f)\ \mu(s) \in M(A_1) \quad \forall\, s \in \mathcal{S}_1$$

La formulazione dei problemi di ottimizzazione (DP3$'$) e (DP4$'$) è astratta e non risolvibile: anche in questo caso, per cercare di convertirla in un problema di ottimizzazione concreto che si possa risolvere, si evidenziano i coefficienti dei vincoli polinomiali per poi applicare i risultati presentati nella Sezione 3.3. In particolare, fissato uno stato $s \in \mathcal{S}_2$ ed uno stato $t \in \mathcal{S}_1$ si ha:

$$\beta \sum_{s'=1}^{k} \hat{v}(s') \sum_{j=0}^{m_s} p_j(s',s)\nu_j(s) = \beta\big(\bar{\nu}(s)^T Q_{21}(s)\big)\hat{\mathbf{v}}, \tag{6.13}$$

$$\beta \sum_{s'=k+1}^{N} \hat{w}(s') \sum_{i=0}^{n_t} p_i(s',t)\mu_i(t) = \beta\big(\bar{\mu}(t)^T Q_{12}(t)\big)\hat{\mathbf{w}}. \tag{6.14}$$

Prima quindi di mostrare la versione in programmazione semidefinita di questi problemi di ottimizzazione polinomiale, come nel caso dei problemi primali è importante notare che risulterà necessario aggiungere alcuni vincoli di programmazione semidefinita per calcolare i momenti "mancanti" necessari alla ricostruzione delle strategie dei giocatori. In particolare, grazie al Lemma 7 si aggiungeranno i vincoli (e) (f) ed (l) (m) ai problemi duali per permettere la ricostruzione delle strategie miste dei giocatori.

Il problema di programmazione semidefinita che concretizza il problema (DP3′) è il seguente.

$$\max_{\bar{\nu}(s),\nu_{m_s+1}(s),z(s),A_s,B_s} \sum_{s=k+1}^{N} \big(z(s) - \beta\big(\bar{\nu}(s)^T Q_{21}(s)\big)\hat{\mathbf{v}}\big), \quad s.t.$$

$$(a) \qquad \mathcal{H}^*(A_s + \tfrac{1}{2}(L_1 B_s L_2^T + L_2 B_s L_1^T) - L_2 B_s L_2^T = \\ = -R(s)\bar{\nu}(s) - z(s)e_1, \quad \forall\, s \in \mathcal{S}_2$$

$$(b) \qquad \mathcal{H}(\bar{\nu}(s)) \succeq 0, \quad \forall\, s \in \mathcal{S}_2$$

$$(c) \qquad \tfrac{1}{2}(L_1^T \mathcal{H}(\bar{\nu})(s) L_2 + L_2^T \mathcal{H}(\bar{\nu})(s) L_1) + \\ - L_2^T \mathcal{H}(\bar{\nu})(s) L_2 \succeq 0, \quad \forall\, s \in \mathcal{S}_2$$

$$(d) \qquad \textstyle\sum_{s=k+1}^{N} (E_s - \beta Q_{22}(s))^T \bar{\nu}(s) = 1$$

$$(e) \qquad \mathcal{H}([\nu_1(s),\, \nu_2(s),\, ...,\, \nu_{n_s+1}(s)]^T) \succeq 0, \quad \forall\, s \in \mathcal{S}_2$$

$$(f) \quad \mathcal{H}([\nu_0(s),\, \nu_1(s),\, ...,\, \nu_{n_s}(s)]^T) - \mathcal{H}([\nu_1(s),\, \nu_2(s),\, ...,\, \nu_{n_s+1}(s)]^T) \succeq 0, \\ \forall\, s \in \mathcal{S}_2$$

$$(g) \qquad A_s, B_s \succeq 0, \quad \forall\, s \in \mathcal{S}_2 \tag{DP3}$$

dove si ricorda che $\bar{\nu}(s) \in \mathbb{R}^{m_s+1}$, ovvero $\bar{\nu}(s)$ non contiene il momento di ordine $m_s + 1$. A questo punto, si può esprimere il seguente lemma.

Lemma 18. Siano $A_1 = A_2 = [0, 1]$. Il problema di programmazione semidefinita (DP3) risolve esattamente il problema di ottimizzazione polinomiale (DP3′).

Dimostrazione. La disuguaglianza polinomiale (a) nel problema (DP3′) ha il vettore dei coefficienti $-R(s)\bar{\nu}(s) - z(s)e_1$. La dimostrazione si ha quindi per diretta conseguenza del Lemma 4 riguardante la rappresentazione semidefinita dei polinomi non negativi su $[0,1]$, ed il Lemma 9 riguardante la rappresentazione semidefinita delle sequenze di momenti di misure non negative supportate su $[0,1]$. □

La soluzione di questo SDP produce i momenti delle misure $\nu(s)$ per il giocatore 2, una per ogni stato, negli stati in cui egli controlla le transizioni. Normalizzando come indicato nella Sezione 4.4.3 ed utilizzando poi la procedura indicata nella Sezione 3.3.2, è possibile ricostruire il supporto (finito) ed i pesi della strategia del giocatore 2, ricavando una coppia supporto-pesi per ogni stato in cui egli governa le transizioni.
In modo analogo, è possibile ottenere il seguente problema di programmazione semidefinita che concretizza il problema (DP4′).

$$\max_{\bar{\mu}(s),\mu_{n_s+1}(s),z(s),A_s,B_s} \sum_{s=1}^{k} \big(z(s) + \beta\big(\bar{\mu}(s)^T Q_{12}(s)\big)\hat{\mathbf{w}}\big), \quad s.t.$$

$$(h) \qquad \begin{aligned} &\mathcal{H}^*(A_s + \tfrac{1}{2}(L_1 B_s L_2^T + L_2 B_s L_1^T) - L_2 B_s L_2^T = \\ &= R(s)^T \bar{\mu}(s) - z(s)e_1, \quad \forall\ s \in \mathcal{S}_1 \end{aligned}$$

$$(i) \qquad \mathcal{H}(\bar{\mu}(s)) \succeq 0, \quad \forall\ s \in \mathcal{S}_1$$

$$(j) \qquad \begin{aligned} &\tfrac{1}{2}(L_1^T \mathcal{H}(\bar{\mu})(s) L_2 + L_2^T \mathcal{H}(\bar{\mu})(s) L_1) + \\ &- L_2^T \mathcal{H}(\bar{\mu})(s) L_2 \succeq 0, \quad \forall\ s \in \mathcal{S}_1 \end{aligned}$$

$$(k) \qquad \textstyle\sum_{s=1}^{k} (E_s - \beta Q_{11}(s))^T \bar{\mu}(s) = 1$$

$$(l) \qquad \mathcal{H}([\mu_1(s),\, \mu_2(s),\, ...,\, \mu_{n_s+1}(s)]^T) \succeq 0, \quad \forall\ s \in \mathcal{S}_1$$

$$(m) \qquad \begin{aligned} &\mathcal{H}([\mu_0(s),\, \mu_1(s),\, ...,\, \mu_{n_s}(s)]^T) - \mathcal{H}([\mu_1(s),\, \mu_2(s),\, ...,\, \mu_{n_s+1}(s)]^T) \succeq 0, \\ &\forall\ s \in \mathcal{S}_1 \end{aligned}$$

$$(n) \qquad A_s, B_s \succeq 0, \quad \forall\ s \in \mathcal{S}_1$$

(DP4)

dove si ricorda che $\bar{\mu}(s) \in \mathbb{R}^{n_s+1}$, ovvero $\bar{\mu}(s)$ non contiene il momento di ordine $n_s + 1$. A questo punto, è possibile esprimere il seguente lemma.

Lemma 19. Siano $A_1 = A_2 = [0, 1]$. Il problema di programmazione semidefinita (DP4) risolve esattamente il problema di ottimizzazione polinomiale (DP4′).

Dimostrazione. La disuguaglianza polinomiale (d) nel problema (DP3′) ha il vettore dei coefficienti $R(s)^T \bar{\mu}(s) - z(s)e_1$. La dimostrazione si ha quindi per diretta conseguenza del Lemma 4 riguardante la rappresentazione semidefinita dei polinomi non negativi su $[0, 1]$, ed il Lemma 9 riguardante la rappresentazione semidefinita delle sequenze di momenti di misure non negative supportate su $[0, 1]$. □

La soluzione di questo SDP produce i momenti delle misure $\mu(s)$ per il giocatore 1, una per ogni stato, negli stati in cui egli controlla le transizioni. Normalizzando come indicato nella Sezione 4.4.3 ed utilizzando poi la procedura indicata nella Sezione 3.3.2, è possibile ricostruire il supporto (finito) ed i pesi della strategia del giocatore 1, ricavando una coppia supporto-pesi per ogni stato in cui egli governa le transizioni.

Sono stati quindi derivati i quattro problemi di programmazione semidefinita (SP3), (SP4), (DP3) e (DP4) che generalizzano i problemi di programmazione lineare utilizzati nel caso di spazi di strategie pure finiti. Nella prossima sezione viene presentato l'algoritmo principale che, risolvendo iterativamente questi SDP, converge ad una soluzione di ϵ-equilibrio.

6.4.4 Descrizione dell'Algoritmo

Il seguente algoritmo, che permette di calcolare l'ϵ-equilibrio in un gioco di questa classe, può essere diviso in 4 fasi che vengono ora presentate.

Inizializzazione e calcolo dei valori massimi Come è stato discusso nella Sezione 2.3, la definizione di ϵ-equilibrio richiede che i valori del gioco siano normalizzati nell'intervallo $[0, 1]$. Per questo motivo, una volta fissato l'$\epsilon_0 > 0$, è necessario calcolare il valore che assume uno stato sia quando entrambi i giocatori giocano le strategie che avvantaggiano il più possibile il giocatore 1, sia quando entrambi i giocatori giocano le strategie che avvantaggiano il più

possibile il giocatore 2, ed utilizzare tali valori per eseguire la normalizzazione del valore dello stato. Questi due problemi risultano però essere dei problemi di ottimizzazione polinomiale bivariata, in quanto è necessario massimizzare (nel primo) e minimizzare (nel secondo) il valore degli stati facendo variare le due strategie contemporaneamente. Come mostrato in [17], risolvere un problema di ottimizzazione polinomiale multivariato invece che univariato richiede di risolvere, invece di un unico SDP, una gerarchia di SDP di dimensione incrementale; tale calcolo renderebbe molto pesante l'intero algoritmo. La soluzione a questo problema è nel non calcolare i valori degli stati con tali strategie ma di considerare il massimo ed il minimo valore che può assumere uno stato: in questo modo, fissato uno stato $s \in S$ il fattore di normalizzazione risulta essere il seguente

$$\lambda(s) = \frac{1-\beta}{\max_{a_1,a_2} r(s,a_1,a_2)}.$$

È quindi necessario come prima cosa calcolare $\lambda(s)$ per ogni stato s del gioco. Una volta calcolati tutti i fattori di normalizzazione, è possibile iniziare con le iterazioni dell'algoritmo.

Si supponga di inizializzare il vettore $w^0(t) \equiv 0 \; \forall \, t \in \mathcal{S}_2$.

Calcolo delle soluzioni degli SDP Si risolva il problema (SP4) ed il suo duale (DP4) con $\hat{w}(t) = w^0(t), \forall t \in \mathcal{S}_2$. Sia $(v^0(s), \mu^0(s), \nu^0(s)), \forall s \in \mathcal{S}_1$ la soluzione ottima a tali problemi. Si risolva quindi il problema (SP3) ed il suo duale (DP3) con $\hat{v}(s) = v^0(s), \forall s \in \mathcal{S}_1$, e sia $(w^1(t), \mu^1(t), \nu^1(t)), \forall t \in \mathcal{S}_2$ la soluzione ottima ai due problemi di programmazione semidefinita.

Calcolo del valore degli stati Le soluzioni dei 4 SDP producono un profilo di strategie dei due giocatori su tutto lo spazio degli stati, ovvero una coppia di strategie che chiamiamo μ_0 e ν_0. Tuttavia, i valori degli stati prodotti dalla soluzione di questi SDP sono calcolati a partire dal valore noto di alcuni stati, perciò non rispecchiano i valori reali degli stati quando si gioca quel profilo di strategie. Per questo motivo è necessario calcolare il valore degli stati quando si applica il profilo di strategie calcolato. Conoscendo le strategie dei giocatori, il vettore dei valori degli stati $\mathbf{v}_0$ può essere semplicemente calcolato risolvendo un banale sistema di N equazioni lineari in N incognite.

Verifica della condizione di termine Nel caso di spazi di strategie pure finiti, la condizione di termine dell'algoritmo dipendeva dalle basi corrispondenti alle soluzioni degli LP. Nella programmazione semidefinita, però, non esiste un analogo diretto della soluzione di base ammissibile. Per questo motivo è necessario utilizzare una condizione di termine diversa, che consenta di avere comunque delle garanzie sul risultato finale dell'algoritmo. Si utilizza quindi il concetto di ϵ-equilibrio che, come è stato definito nella Sezione 2.3, è un concetto di soluzione in cui all'equilibrio nessun giocatore riesce a guardagnare più di ϵ cambiando la propria strategia.
Per questo motivo la condizione di termine dell'algoritmo è ottenuta nel seguente modo:

1. si calcola il vettore $\mathbf{v}_1$ dei valori degli stati quando il giocatore 1 esegue la sua miglior risposta alla strategia ν_0 del giocatore 2,

2. si calcola il vettore $\mathbf{v}_2$ dei valori degli stati quando il giocatore 2 esegue la sua miglior risposta alla strategia μ_0 del giocatore 1,

3. si calcolano i vettori differenza $\mathbf{d}_1 = \mathbf{v}_1 - \mathbf{v}_0$ e $\mathbf{d}_2 = \mathbf{v}_0 - \mathbf{v}_2$ e si moltiplica ogni elemento di $\mathbf{d}_1$ e $\mathbf{d}_2$ per il corrispondente fattore di normalizzazione $\lambda(s)$, ottenendo i vettori normalizzati $\mathbf{d}_1^{norm}$ e $\mathbf{d}_2^{norm}$,

4. si considera il valore massimo tra tutti gli elementi dei vettori normalizzati $\mathbf{d}_1^{norm}$ e $\mathbf{d}_2^{norm}$: tale valore è il massimo che può ottenere almeno uno dei due giocatori cambiando la sua strategia nel caso l'avversario mantenga la propria. Se tale valore è inferiore ad ϵ_0, allora il profilo di strategie (μ_0, ν_0) è un ϵ_0-equilibrio e l'algoritmo termina; viceversa, si considera $\hat{w}(t) = w^1(t), \forall t \in \mathcal{S}_2$ e si ritorna al calcolo delle soluzioni dei quattro SDP.

Le migliori risposte, come è stato discusso nella Sezione 5.4, possono essere ottenute risolvendo un SDP per ognuna di esse. Questo significa che ogni iterazione dell'algoritmo richiede la soluzione di 6 SDP.

6.4.5 Convergenza

Viene ora discussa la convergenza di questo algoritmo, introducendo prima di tutto i seguenti lemmi.

Lemma 20. Sia $v(t) \leq \check{v}(t), \forall t \in \mathcal{S}_1$. Sia $(w(t), \mu(t)), \forall t \in \mathcal{S}_2$ la soluzione ottima del problema (SP3) quando si ha $\hat{v}(t) = v(t), \forall t \in \mathcal{S}_1$. Sia $(\check{w}(t), \check{\mu}(t)), \forall t \in \mathcal{S}_2$ la soluzione ottima del problema (SP3) quando si ha $\hat{v}(t) = \check{v}(t), \forall t \in \mathcal{S}_1$. Allora $w(s) \leq \check{w}(s), \forall t \in \mathcal{S}_2$.

Dimostrazione. Si può notare che anche $(w(t), \mu(t)), \forall t \in \mathcal{S}_2$ rispetta tutti i vincoli del problema (SP3) quando si ha $\hat{v}(t) = \check{v}(t), \forall t \in \mathcal{S}_1$, mentre in generale $(\check{w}(t), \check{\mu}(t)), \forall t \in \mathcal{S}_2$ non rispetta tutti i vincoli del problema (SP3) quando si ha $\hat{v}(t) = v(t), \forall t \in \mathcal{S}_1$. Da questo si può concludere che $w(s) \leq \check{w}(s), \forall t \in \mathcal{S}_2$. □

Lemma 21. Sia $w(t) \leq \check{w}(t), \forall t \in \mathcal{S}_2$. Sia $(v(t), \nu(t)), \forall t \in \mathcal{S}_1$ la soluzione ottima del problema (SP4) quando si ha $\hat{w}(t) = w(t), \forall t \in \mathcal{S}_2$. Sia $(\check{v}(t), \check{\nu}(t)), \forall t \in \mathcal{S}_1$ la soluzione ottima del problema (SP4) quando si ha $\hat{w}(t) = \check{w}(t), \forall t \in \mathcal{S}_2$. Allora $v(s) \leq \check{v}(s), \forall t \in \mathcal{S}_1$.

Dimostrazione. Si può notare che anche $(v(t), \nu(t)), \forall t \in \mathcal{S}_1$ rispetta tutti i vincoli del problema (SP4) quando si ha $\hat{w}(t) = \check{w}(t), \forall t \in \mathcal{S}_2$, mentre in generale $(\check{v}(t), \check{\nu}(t)), \forall t \in \mathcal{S}_1$ non rispetta tutti i vincoli del problema (SP4) quando si ha $\hat{w}(t) = w(t), \forall t \in \mathcal{S}_2$. Si conclude quindi che $v(s) \leq \check{v}(s), \forall t \in \mathcal{S}_1$. □

Lemma 22. Si consideri la sequenza di coppie di soluzioni $\big((v^0, w^1), (v^1, w^2), ..., (v^{r-1}, w^r), ...\big)$ dei quattro SDP (SP3), (SP4), (DP3) e (DP4), di cui ogni i^{esima} coppia è calcolata nella i^{esima} iterazione dell'algoritmo. Si ha quindi il seguente risultato:

$$v^r \leq v^{r+1} \leq v^* \; \forall \, r, \qquad w^r \leq w^{r+1} \leq w^* \; \forall \, r,$$

dove v^* e w^* sono i valori ottimi del gioco stocastico.

Dimostrazione. Si può notare che $w^0(t) \equiv 0$, quindi si ha $w^0(t) \leq w^*(t), \; \forall \, t \in \mathcal{S}_2$. Dal Lemma 21 sappiamo che $v^0(s) \leq v^*(s), \; \forall \, s \in \mathcal{S}_1$. Poichè si sa che $w^1(t) \geq 0$, per il Lemma 20 sappiamo che $w^0(t) \leq w^1(t) \leq w^*(t), \; \forall \, t \in \mathcal{S}_2$. Grazie ad una induzione elementare è quindi possibile dimostrare il Lemma. □

Teorema 19. Si supponga di avere le seguenti due coppie di vettori di valori del gioco e strategie:

$$v(s), \nu(s, a_2), \quad \forall \, 1 \leq s \leq k, a_2 \in A_2,$$
$$w(s), \mu(s, a_1), \quad \forall \, k+1 \leq s \leq N, a_1 \in A_1.$$

dove $\big(v(s), \nu(s, a_2),\ \forall 1 \leq s \leq k, a_2 \in A_2\big)$ è ottima per il problema (SP4) quando $\hat{w}(s) = w(s), k+1 \leq s \leq N$, e $\big(w(s), \mu(s, a_1), \forall\, k+1 \leq s \leq N, a_1 \in A_1\big)$ è ottima per il problema (SP3) quando $\hat{v}(s) = w(s),\ 1 \leq s \leq k$.
Allora $\big(v(1), ..., v(k), w(k+1), ..., w(N)\big)$ è il vettore dei valori del gioco.

Dimostrazione. La dimostrazione si ha per diretta conseguenza del Teorema 19. □

Questi risultati permettono di arrivare al seguente teorema.

Teorema 20.

$$\lim_{r \to \infty} (v^r, w^r) = (v^*, w^*).$$

Dimostrazione. Dal Lemma 22 si sa che $v^r \leq v^{r+1} \leq v^*\ \forall\ r$, e che $w^r \leq w^{r+1} \leq w^*\ \forall\, r$.
Dal Teorema 19 si sa invece che se $(v^i, w^{i+1}) = (v^{i+1}, w^{i+2})$ allora $\big(v^i(1), ..., v^i(k), w^{i+1}(k+1), ..., w^{i+1}(N)\big)$ è il vettore dei valori del gioco. Si sa quindi che i valori del gioco nella sequenza $\big((v^0, w^1), (v^1, w^2), ..., (v^{r-1}, w^r), ...\big)$ sono non decrescenti, e risultano uguali solo quando si è raggiunto l'ottimo, da cui si ha $\lim_{r\to\infty}(v^r, w^r) = (v^*, w^*)$. □

6.4.6 Ottimalità della Soluzione

Viene ora mostrato che la soluzione dell'algoritmo descritto è effettivamente la soluzione d'equilibrio desiderata.

Teorema 21. Siano μ^* e ν^* le strategie miste dei giocatori calcolate dall'algoritmo, e sia $\mathbf{v} = \mathbf{v}(\mu^*, \nu^*)$ il vettore del valore degli stati con tali strategie. Allora $\mathbf{v}$ soddisfa la seguente disuguaglianza del punto di sella:

$$\mathbf{v}(\mu, \nu^*) - \epsilon_0 e_s \leq \mathbf{v}(\mu^*, \nu^*) \leq \mathbf{v}(\mu^*, \nu) + \epsilon_0 e_s$$

per ogni coppia di strategie miste (μ, ν), dove $e_s \in \mathbb{R}^S$ è un vettore composto unicamente da 1.

Dimostrazione. Viene ora mostrato per assurdo che $\mathbf{v}(\mu, \nu^*) - \epsilon_0 e_s \leq \mathbf{v}(\mu^*, \nu^*)$. Si supponga che esista una strategia μ tale che

$$\mathbf{v}(\mu, \nu^*) > \mathbf{v}(\mu^*, \nu^*) + \epsilon_0 e_s. \tag{6.15}$$

Allora, indicando con μ_{br} la best response alla strategia ν^* nota, si ha

$$\mathbf{v}(\mu_{br}, \nu^*) \geq \mathbf{v}(\mu, \nu^*). \tag{6.16}$$

L'algoritmo è terminato dando soluzione $\mathbf{v}_0$, e poichè la condizione di termine era soddisfatta si sa che vale la seguente disequazione:

$$\mathbf{v}(\mu_{br}, \nu^*) \leq \mathbf{v}(\mu^*, \nu^*) + \epsilon_0 e_s. \tag{6.17}$$

Dalle disequazioni (6.16) ed (6.17) si ha la disuguaglianza

$$\mathbf{v}(\mu, \nu^*) \leq \mathbf{v}(\mu_{br}, \nu^*) \leq \mathbf{v}_0(\mu^*, \nu^*) + \epsilon_0 e_s \tag{6.18}$$

che dimostra che la disequazione (6.15) è un assurdo.

Viene ora dimostrato per assurdo che $\mathbf{v}(\mu^*, \nu^*) \leq \mathbf{v}(\mu^*, \nu) + \epsilon_0 e_s$.
Si supponga che esista una strategia ν tale che vale la seguente disequazione:

$$\mathbf{v}_0(\mu^*, \nu^*) > \mathbf{v}(\mu^*, \nu) + \epsilon_0 e_s. \tag{6.19}$$

Allora, indicando con ν_{br} la best response alla strategia μ^* nota, si ha

$$\mathbf{v}(\mu^*, \nu_{br}) \leq \mathbf{v}(\mu^*, \nu). \tag{6.20}$$

L'algoritmo è terminato dando soluzione $\mathbf{v}_0$, e poichè la condizione di termine era soddisfatta si sa che vale la seguente disequazione:

$$\mathbf{v}(\mu^*, \nu_{br}) \geq \mathbf{v}(\mu^*, \nu^*) - \epsilon_0 e_s. \tag{6.21}$$

Dalle disequazioni (6.20) ed (6.21) si ha la disuguaglianza

$$\mathbf{v}(\mu^*, \nu) \geq \mathbf{v}(\mu^*, \nu_{br}) \geq \mathbf{v}(\mu^*, \nu^*) - \epsilon_0 e_s. \tag{6.22}$$

che dimostra che la disequazione (6.19) è un assurdo, e conclude la dimostrazione. □

Le analisi fatte in questa sezione e nella Sezione 6.4.5 permettono quindi di concludere che l'algoritmo presentato converge, e permette di ottenere un ϵ-equilibrio (con supporti finiti) in un gioco stocastico con switching control ad azioni continue, con funzione di payoff e transizioni di probabilità polinomiali rispetto alle azioni dei giocatori.

Capitolo 7

Valutazione Sperimentale

7.1 Introduzione

In questo capitolo viene presentata una valutazione sperimentale delle prestazioni dell'algoritmo, al variare della dimensione del gioco (polinomiale stocastico e con Switching Control) in ingresso. In particolare, vengono analizzate le prestazioni all'aumentare del numero di stati del gioco e del grado delle funzioni polinomiali, ovvero la funzione di payoff e le probabilità di transizione. Per poter eseguire questa valutazione, sono stati misurati due parametri prestazionali dell'algoritmo. Il primo parametro è il valore ϵ dell'equilibrio approssimato ottenuto, che indica la massima variazione nel reward che in tale profilo di strategie un giocatore può ottenere cambiando la propria strategia, supponendo che la strategia dell'avversario sia nota e fissata. Il secondo parametro prestazionale è il tempo di computazione richiesto, che indica il variare della velocità d'esecuzione dell'algoritmo al variare della dimensione del problema in ingresso.

Nella Sezione 7.2 vengono descritti i modelli utilizzati nell'analisi sperimentale. Nella Sezione 7.3 vengono considerati i risultati dell'algoritmo applicato ad uno dei modelli analizzati sperimentalmente, mostrando che il risultato ottenuto è effettivamente una soluzione di equilibrio. Nella Sezione 7.4 viene valutata la velocità di convergenza dell'algoritmo basandosi sui risultati sperimentali. Infine, nella Sezione 7.5 vengono valutati i tempi di calcolo richiesti dall'algoritmo, basandosi sui risultati sperimentali ottenuti.

7.2 Descrizione dei Modelli Analizzati

Si considerino tre giochi polinomiali stocastici con Switching Control. Nel primo gioco vi sono solo due stati: nel primo stato le transizioni sono governate dal giocatore 1, mentre nel secondo stato le transizioni sono governate dal giocatore 2. Nel secondo gioco vi sono tre stati: in due di questi le transizioni sono governate dal giocatore 1, mentre nel terzo stato le transizioni sono governate dal giocatore 2. Infine, nel terzo gioco vi sono quattro stati: il giocatore 1 governa le transizioni dei primi due stati, mentre negli altri stati le transizioni sono governate dal giocatore 2.

Poiché in questa analisi sperimentale viene fatto variare anche il grado dei polinomi, per semplicità, una volta fissato il grado delle funzioni polinomiali le funzioni di payoff sono uguali da stato a stato, mentre le probabilità di transizione seguono uno schema fissato. In particolare:

- nel caso di polinomi di grado 4, la funzione di payoff è data da $R(s, a_1, a_2) = (a_1 - a_2)^2$, $\forall\, s \in S$, mentre per ogni $s \in \mathcal{S}_1$ e per ogni $s' \in \mathcal{S}_2$ le probabilità di transizione sono le seguenti:

$$p(t, s, a_1) = \frac{{a_1}^2}{|S|},\ \forall\, t \neq s \in S; \qquad p(s, s, a_1) = 1 - \frac{{a_1}^2}{|S|},$$

$$p(t, s', a_1) = \frac{{a_2}^2}{|S|},\ \forall\, t \neq s' \in S; \qquad p(s', s', a_2) = 1 - \frac{{a_2}^2}{|S|};$$

- nel caso di polinomi di grado 6, la funzione di payoff è data da $R(s, a_1, a_2) = (a_1 - a_2)^2 + 1.025 - a_1^2 + \frac{1}{3}a_1^3 - a_2^3 + 2a_1^3a_2^3$, $\forall\, s \in S$, mentre per ogni $s \in \mathcal{S}_1$ e per ogni $s' \in \mathcal{S}_2$ le probabilità di transizione sono le seguenti:

$$p(t, s, a_1) = \frac{{a_1}^3}{|S|},\ \forall\, t \neq s \in S; \qquad p(s, s, a_1) = 1 - \frac{{a_1}^3}{|S|},$$

$$p(t, s', a_1) = \frac{{a_2}^3}{|S|},\ \forall\, t \neq s' \in S; \qquad p(s', s', a_2) = 1 - \frac{{a_2}^3}{|S|};$$

- nel caso di polinomi di grado 8, la funzione di payoff è data da $R(s, a_1, a_2) = 0.908 + (a_1 - a_2)^2 - a_1^2 + 2a_1^4a_2^4 - a_2^4 + \frac{1}{4}a_1^4$, $\forall\, s \in S$, mentre per ogni $s \in \mathcal{S}_1$ e per ogni $s' \in \mathcal{S}_2$ le probabilità di transizione sono le seguenti:

$$p(t, s, a_1) = \frac{{a_1}^4}{|S|},\ \forall\, t \neq s \in S; \qquad p(s, s, a_1) = 1 - \frac{{a_1}^4}{|S|},$$

$$p(t, s', a_1) = \frac{{a_2}^4}{|S|}, \forall\, t \neq s' \in S; \qquad p(s', s', a_2) = 1 - \frac{{a_2}^4}{|S|}.$$

In queste definizioni il termine $|S|$ indica il numero di stati del gioco. Il discount factor è $\beta = \frac{1}{2}$. Ognuno di questi modelli è stato posto in ingresso all'algoritmo, che ne ha calcolato le soluzioni di equilibrio: viene ora considerata la soluzione di uno di questi modelli calcolata sperimentalmente, e ne viene mostrata la correttezza.

7.3 Esempio di Soluzione

Prima di procedere alla valutazione sperimentale, viene presentata e discussa, a scopo dimostrativo, la soluzione di equilibrio ottenuta sperimentalmente dall'algoritmo applicandolo al modello a due stati, con funzioni polinomiali di grado 4.

Supponendo $\epsilon = 10^{-3}$, il profilo di strategie di ϵ-Nash calcolato dall'algoritmo è il seguente in entrambi gli stati:

- il giocatore 1 esegue l'azione $a_1 = 0$ con probabilità $\mu(s, 0) = 0.5$, e l'azione $a_1 = 1$ con probabilità $\mu(s, 1) = 0.5$, $\forall\, s \in S$;
- il giocatore 2 esegue l'azione $a_2 = 1$ con probabilità $\nu(s, 1) = 1$, $\forall\, s \in S$.

Il valore degli stati calcolato dall'algoritmo è uguale per entrambi: si ha infatti $v(s_1) = 0.5$ e $w(s_2) = 0.5$.

Per dimostrare che tale profilo di strategie è effettivamente un equilibrio, si supponga di fissare la strategia del giocatore 2 in ogni stato, ponendola uguale a quella calcolata dall'algoritmo. A questo punto si misura il massimo incremento di rewards che il giocatore 1 può ottenere variando la propria strategia. Il valore dello stato 2, che dipende dall'azione a_1 e dal valore dello stato $v(s_1)$ (avendo fissato $a_2 = 1$), è calcolabile con la seguente equazione:

$$w(s_2) = a_1^2 + 0.25 - a_1 + \beta v(s_1). \tag{7.1}$$

L'equazione (7.1) riflette il concetto di "strategia miope", in quanto si può notare che, indipendentemente dal valore dello stato 1, per massimizzare il valore dello stato 2 il giocatore 1 cercherà unicamente di massimizzarne il payoff immediato. Per questo motivo, nello stato 2, la miglior risposta del

giocatore 1 alla strategia del giocatore 2 fissata, è di scegliere l'azione $a_1 = 0$ o l'azione $a_1 = 1$, che massimizzano il reward immediato: tale strategia coincide con quella calcolata dall'algoritmo. Il valore dello stato 2 con tali strategie è quindi definito nel modo seguente:

$$w(s_2) = 0.25 + \beta v(s_1).$$

A questo punto si considera lo stato 1. Il valore di tale stato, conoscendo l'azione del giocatore 2, è dato dalla seguente equazione:

$$v(s_1) = a_1^2 + 0.25 - a_1 + (1 - a_1^2)\,\beta v(s_1) + \beta\, a_1^2\, 0.25 + \beta^2\, a_1^2\, v(s_1),$$

che può essere riscritta nel seguente modo:

$$v(s_1) = \frac{\frac{9}{8}a_1^2 - a_1 + \frac{1}{4}}{\frac{1}{4}a_1^2 + \frac{1}{2}}.$$

In Figura 7.1 viene mostrato il grafico di tale funzione nell'intervallo $[0, 1]$.

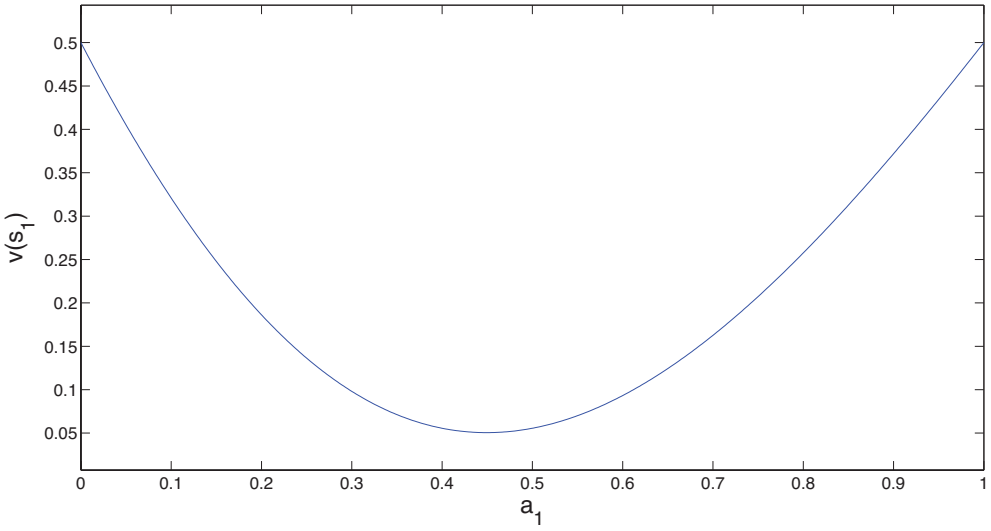

Figura 7.1: Valore dello stato 1 in funzione dell'azione a_1 del giocatore 1, fissata la strategia del giocatore 2.

Come si può notare, per massimizzare il valore dello stato 1 il giocatore sceglierà un'azione tra $a_1 = 0$ ed $a_1 = 1$, portando il valore dello stato a $v(s_1) = 0.5$ ed applicando la stessa strategia calcolata dall'algoritmo. In questo modo, anche il valore dello stato 2 risulterà essere $w(s_2) = 0.25 + \beta v(s_1) = 0.5$.

È stato quindi dimostrato che la strategia del giocatore 1 calcolata dall'algoritmo è effettivamente la miglior risposta alla strategia del giocatore 2: questo

dimostra che il valore di ϵ massimo che può ottenere il giocatore 1 cambiando la propria strategia è zero. In modo del tutto analogo è possibile dimostrare che la strategia del giocatore 2 calcolata dall'algoritmo è la miglior risposta alla strategia fissata del giocatore 1: il profilo di strategie calcolato dall'algoritmo è quindi effettivamente un equilibrio, ed il valore degli stati calcolato coincide con il valore ottimo degli stati del gioco stocastico.

7.4 Valutazione della Convergenza

Viene ora presentata una valutazione della velocità sperimentale di convergenza (i.e. la velocità di diminuzione della variazione massima ϵ del reward), al variare sia del numero degli stati del gioco sia del grado delle funzioni polinomiali: vengono prima presentati i risultati sperimentali ottenuti, facendone poi un'analisi volta a valutare le prestazioni dell'algoritmo.

Per realizzare questa analisi sperimentale è stato utilizzato *YALMIP* [18], un toolbox per *MATLAB* che permette di modellizzare in modo efficiente dei problemi di ottimizzazione, utilizzando poi dei risolutori esterni per calcolare la soluzione. Come risolutore è stato utilizzato *SeDuMi* [29], un pacchetto software per risolvere problemi di programmazione semidefinita.

I risultati sperimentali al variare del numero di stati del gioco sono rappresentati in Figura 7.2: il primo grafico mostra il rapporto tra il valore ϵ ed il numero di iterazioni nel caso di funzioni polinomiali di grado 4, il secondo grafico mostra il rapporto nel caso di funzioni polinomiali di grado 6, ed il terzo grafico mostra il rapporto nel caso di funzioni polinomiali di grado 8.

I risultati sperimentali ottenuti, invece, all'aumentare del grado delle funzioni polinomiali nel modello analizzato sono rappresentati in Figura 7.3: il primo grafico mostra il rapporto tra il valore ϵ ed il numero di iterazioni nel caso di gioco a due stati, il secondo grafico mostra il rapporto nel caso di gioco a tre stati, ed il terzo grafico mostra il rapporto nel caso di gioco a quattro stati.

Da questi risultati è possibile trarre alcune conclusioni sulle prestazioni dell'algoritmo in termini di velocità di convergenza. In primo luogo, si può notare che la velocità di convergenza dell'algoritmo è molto alta, qualsiasi sia il modello analizzato tra quelli considerati. In particolare, già a seguito della seconda iterazione si ottiene un valore di ϵ prossimo allo zero, quindi

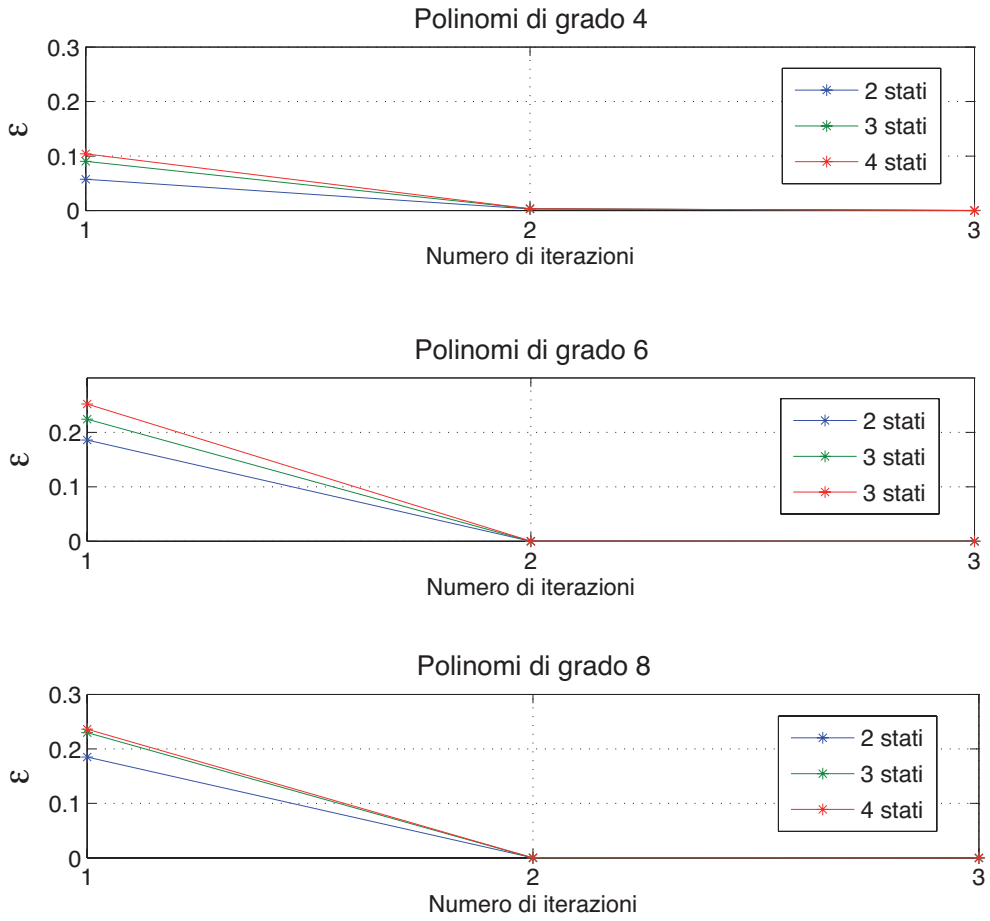

Figura 7.2: Analisi sperimentale della convergenza al variare del numero di stati.

supponendo di voler calcolare un ϵ-Nash con $\epsilon < 10^{-2}$, l'algoritmo impiega solamente due iterazioni per fornire tale risultato.

Inoltre, la velocità di convergenza dell'algoritmo applicato ai modelli analizzati risulta essere indipendente sia dal numero di gradi delle funzioni polinomiali utilizzate, sia dal numero di stati nel modello. Anche questo è un risultato positivo, in quanto indica che, almeno per i modelli analizzati, si ha un'ottima scalabilità rispetto alla dimensione del problema in ingresso all'algoritmo.

Facendo riferimento al terzo grafico della Figura 7.3, si può notare che la velocità di convergenza nel caso si considerino funzioni polinomiali di grado 6 è addirittura inferiore alla velocità di convergenza nel caso si considerino funzioni polinomiali di grado 8. Questo risultato è motivato dal fatto che la velocità di convergenza dell'algoritmo dipende significativamente dai dati del modello analizzato, come i coefficienti delle funzioni polinomiali, oltre che dalla

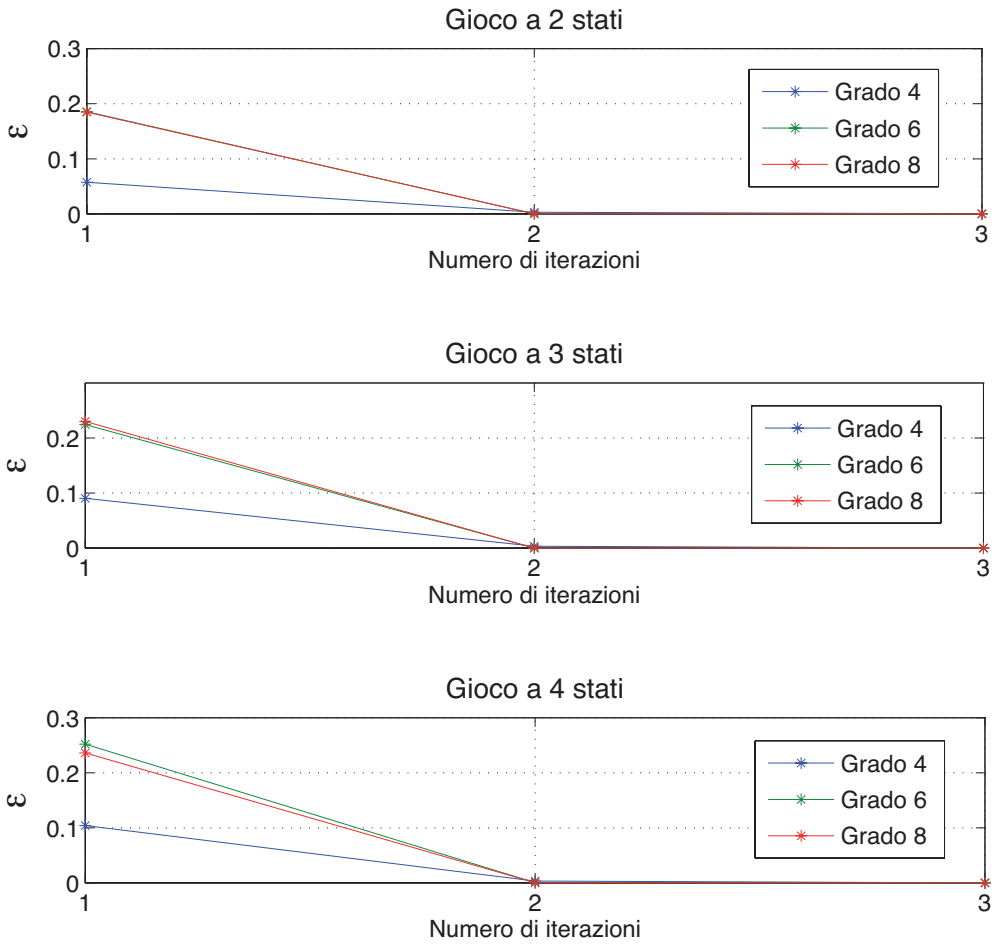

Figura 7.3: Analisi sperimentale della convergenza al variare del grado delle funzioni polinomiali.

dimensione del problema analizzato. A riprova di questa motivazione, durante la sperimentazione sono emersi anche casi in cui, applicando l'algoritmo a funzioni polinomiali di grado crescente, la velocità di convergenza dell'algoritmo è risultata crescente.

Per quanto discusso in questa sezione, la valutazione delle prestazioni dell'algoritmo in termini di velocità di convergenza, almeno per i modelli analizzati, risulta quindi essere positiva.

7.5 Valutazione dei Tempi di Calcolo

Per fornire una maggiore valutazione delle prestazioni dell'algoritmo, oltre alla velocità di convergenza è stato considerato anche il tempo di computazione

richiesto per raggiungere la soluzione d'equilibrio, al variare sia del numero degli stati del gioco sia del grado delle funzioni polinomiali.

Per ogni modello di gioco considerato sono stati misurati 20 campioni di tempi di computazione, dove per ogni singolo modello ed ogni campione sono state eseguite 4 iterazioni dell'algoritmo. La scelta di limitarsi a 4 iterazioni è stata motivata dal fatto che, come mostrato nelle Figure 7.2 e 7.3, la velocità di convergenza dell'algoritmo è tale da ottenere $\epsilon < 10^{-2}$ già alla seconda iterazione.

I risultati sperimentali, ottenuti sia al variare del numero di stati del gioco, sia all'aumentare del grado delle funzioni polinomiali nel modello analizzato, sono rappresentati in Figura 7.4.

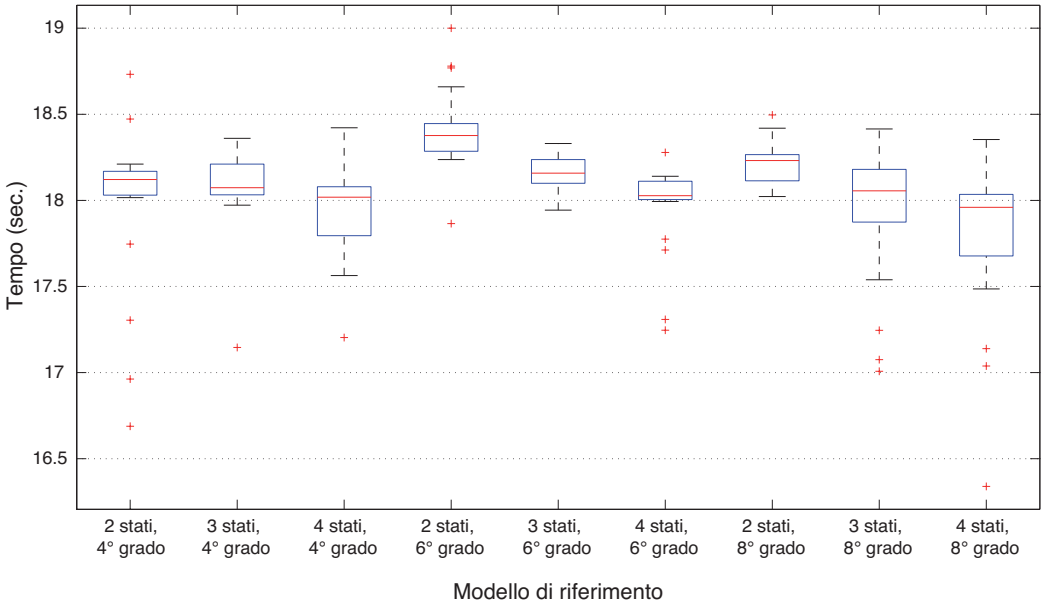

Figura 7.4: Analisi sperimentale del tempo di computazione richiesto, al variare sia del numero di stati del gioco sia del grado delle funzioni polinomiali.

Dal grafico in figura, si può notare che vi è una differenza molto piccola tra i tempi di computazione impiegati dall'algoritmo, qualsiasi sia il modello di gioco considerato. Inoltre, contrariamente a quanto ci si poteva aspettare, all'aumentare della dimensione del problema in ingresso all'algoritmo i tempi di computazione non aumentano, ma in molti casi addirittura decrescono.

Questo risultato è motivato dal fatto che i tempi di computazione richiesti dall'algoritmo sono molto bassi, ed il loro aumento al crescere delle dimensioni del problema è addirittura "mascherato" dal tempo di overhead. Poiché,

quindi, la maggiore componente nel tempo impiegato è dovuta all'overhead computazionale, anche all'aumentare degli stati e del grado dei polinomi nel modello di gioco considerato, il tempo impiegato dall'algoritmo non subisce variazioni.

Ovviamente, è possibile cambiare la configurazione hardware e software utilizzata per cercare di diminuire l'overhead computazionale: questo porterebbe quindi ad una grande riduzione dei tempi di computazione impiegati dall'algoritmo, quando applicato a modelli di gioco della stessa dimensione di quelli che sono stati considerati, ovvero con un massimo di 4 stati e con funzioni polinomiali di grado 8 o inferiore.

Capitolo 8

Conclusioni e Sviluppi Futuri

8.1 Conclusioni

In questa tesi è stata studiata la classe dei giochi polinomiali stocastici a due giocatori e somma zero con Switching Control, estendendo i giochi polinomiali in forma normale e generalizzando la classe dei giochi polinomiali stocastici con Single Controller. Inoltre, sono stati studiati i processi di decisione di Markov polinomiali, estendendo i processi di decisione di Markov al caso in cui lo spazio delle azioni dell'agente è infinito e non numerabile.

Inizialmente sono stati presentati i risultati ottenuti da Parrilo sui giochi polinomiali in forma normale ed i giochi polinomiali stocastici con Single Controller: è stata discussa la caratterizzazione e l'esistenza degli equilibri in queste classi di giochi, e sono stati presentati i rispettivi algoritmi in grado di calcolarne la soluzione.

È stato poi discusso il calcolo della soluzione di un gioco polinomiale stocastico con Switching Control, presentando un algoritmo in grado di calcolare sia il valore degli stati sia un profilo di strategie di ϵ-Nash, risolvendo iterativamente dei problemi di programmazione semidefinita positiva. Per ottenere tali problemi di programmazione semidefinita positiva, sono state considerate due coppie di problemi di ottimizzazione primali e duali di sistemi di polinomi univariati, con vincoli sui momenti delle misure di probabilità, e sono state utilizzate delle tecniche classiche della teoria dei momenti e della riduzione in somma di quadrati per ridurre tali problemi a dei problemi primali e duali di programmazione semidefinita positiva. Inoltre, per determinare la condizione

di termine dell'algoritmo presentato è stato discusso il rapporto tra la miglior risposta di un giocatore ad una strategia dell'avversario, in un gioco di questa classe, e la soluzione di un processo di decisione di Markov polinomiale. È stato perciò discusso il calcolo della politica ottima in un processo di decisione di Markov polinomiale, ed è stato presentato un algoritmo in grado di calcolare il valore ottimo degli stati risolvendo un unico problema di programmazione semidefinita positiva.

Per i giochi polinomiali stocastici con Switching Control è stata dimostrata sia la convergenza dell'algoritmo proposto sia l'ottimalità della soluzione calcolata. Un'importante proprietà dei profili di strategie di ϵ-Nash calcolati dall'algoritmo è quella di avere un supporto finito, di dimensione proporzionale al grado delle funzioni di payoff polinomiali del gioco analizzato.

Nel corso di questo lavoro i principali problemi sono stati riscontrati nella fase di progettazione dell'algoritmo. Per i polinomi multivariati di grado maggiore o uguale a quattro, stabilire la non negatività è un problema NP-hard. Per questo motivo, nella formulazione dei problemi di ottimizzazione polinomiale, ci si è potuti concentrare unicamente sui polinomi univariati. Inoltre, durante la derivazione dal caso di azioni finite ad infinite, si sono dovute risolvere alcune problematiche dovute alla non esistenza del concetto di "soluzione di base" nei problemi di programmazione semidefinita positiva.

È stata infine eseguita una valutazione sperimentale delle prestazioni dell'algoritmo al variare della dimensione del problema in ingresso: per ogni modello di gioco considerato, l'algoritmo ha dimostrato di avere una buona velocità di convergenza. Inoltre, per i modelli di gioco considerati l'algoritmo ha dimostrato una buona scalabilità, mostrando che sia la velocità di convergenza sia i tempi di calcolo impiegati sono indipendenti dall'aumento della dimensione del problema in ingresso.

8.2 Sviluppi Futuri

Una naturale prosecuzione del lavoro qui presentato è quella di eseguire una valutazione sperimentale con un alto numero di modelli di grandi dimensioni, facendo variare i parametri di tali modelli. Questo permetterebbe di ottenere una stima più accurata delle prestazioni dell'algoritmo.

Una possibile prospettiva futura è nell'estensione delle classi di giochi qui presentate al caso dei giochi separabili a somma non zero, una classe di giochi continui in cui i payoffs sono in somma di prodotti e di cui i giochi polinomiali sono una sottoclasse.

Un'altra possibile direzione di ricerca futura potrebbe essere quella di estendere le classi di giochi qui presentate alla classe dei giochi a tempo continuo. Nello sviluppo di questa tesi era stata esaminata anche questa classe di giochi: il problema di ottimizzazione polinomiale nel caso dei giochi a tempo continuo è però multivariato, e non permette quindi l'utilizzo delle tecniche di derivazione dei problemi di programmazione semidefinita positiva che sono state qui utilizzate. Si potrebbe quindi provare ad applicare le tecniche presentate in [17], utilizzando delle gerarchie di SDP per risolvere problemi di ottimizzazione polinomiale multivariati, ottenendo un algoritmo in grado di risolvere i giochi polinomiali stocastici con Switching Control a tempo continuo, e valutando poi l'impatto sulle prestazioni dell'algoritmo di tali gerarchie.

Un ulteriore sviluppo futuro potrebbe essere quello di considerare dei modelli iniziali di giochi non polinomiali, calcolandone quindi un'approssimazione polinomiale delle funzioni di payoff e delle probabilità di transizione, e cercando di valutare lo scostamento tra la soluzione calcolata con il modello polinomiale derivato e la soluzione del modello reale.

Bibliografia

[1] E. Altman. Constrained markov decision processes. In *Chapman & Hall, Boca Raton, Florida*, 1999.

[2] G Blekherman, Pablo A Parrilo, and Thomas R. Polynomial optimization, sums of squares and applications. In *Semidefinite Optimization and Convex Algebraic Geometry*, number x, pages 25–63. 2011.

[3] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. 2004.

[4] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The Complexity of Computing a Nash Equilibrium, 2006.

[5] Samuel Karlin Dresher, Melvin and Lloyd S. Shapley. Polynomial Games. In *Contributions to the Theory of Games*, pages 161–180. 1950.

[6] Filar, Jerzy, Vrieze, and Koos. *Competitive Markov Decision Processes*. 1996.

[7] J. A. Filar. Algorithms for Solving some Undiscounted Stochastic Games. Ph.D. Dissertation, Univeristy of Illinois, Chicago, 1979.

[8] J. A. Filar. Two Remarks Concerning Two Undiscounted Stochastic Games. Tech Rep 392. John Hopkins University, Department of Mathematical Sciences, 1980.

[9] J. a. Filar. Ordered Field Property for Stochastic Games when the Player who Controls Transitions changes from State to State. *Journal of Optimization Theory and Applications*, 34(No 4):503–515, August 1981.

[10] A. M. Fink. Equilibrium in a Stochastic n-Person Game. *J. Sci. Hiroshima Univ. Ser. A-I Math*, 28(No 1):89–93, 1964.

[11] Carlos Guestrin, Daphne Koller, Ronald Parr, and Shobha Venkataraman. Efficient Solution Algorithms for Factored MDPs. *Artificial Intelligence*, 19(c):399–468, 2003.

[12] Milos Hauskrecht and Branislav Kveton. Linear Program Approximations for Factored Continuous-State Markov Decision Processes. In *In Advances in Neural Information Processing Systems 16*, pages 895–902. 2003.

[13] S Karlin. Mathematical Methods and Theory in Games, Programming, and Economics. *Vol. I: Matrix games, programming, and mathematical economics. Vol. II: The Theory of infinite games*, I + II, 1959.

[14] S Karlin and L. S. Shapley. Geometry of Moment Spaces. *Memoirs of the American Mathematical Society*, 12:105, 1953.

[15] Michael Kearns, Yishay Mansour, and Andrew Y. Ng. A Sparse Sampling Algorithm for Near-Optimal Planning in Large Markov Decision Processes. *Machine Learning*, 49:193–208, 2002.

[16] Nagarajan Krishnamurthy, T. Parthasarathy, and G. Ravindran. Orderfield Property of Mixtures of Stochastic Games. *Sankhya A*, 72(No 1):246–275, June 2010.

[17] R. Laraki and J. B. Lasserre. Semidefinite Programming for MinMax Problems and Games. *Mathematical Programming*, (No 1):1–23, May 2010.

[18] J. Lofberg. YALMIP : a toolbox for modeling and optimization in MATLAB. *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, (4):284 – 289.

[19] William D. Maitra, Ashok P., Sudderth. Discrete Gambling and Stochastic Games. *Stochastic Modelling and Applied Probability*, 32, 1996.

[20] R. B. Myerson. *Game Theory: Analysis of Conflict*. 1991.

[21] John Nash. Non-Cooperative Games. *The Annals of Mathematics*, 54(No 2):286–295, 1951.

[22] J. A. Filar O. J. Vrieze, S. H. Tijs, T. E. S. Raghavan. A Finite Algorithm for the Switching Control Stochastic Game. *OR Spektrum 5*, pages 15–24, 1983.

[23] C. A. J. M. Dirven O. J. Vrieze, S. H. Tijs, T. Parthasarathy. A Class of Stochastic Games with Ordered Field Property. *Journal of Optimization Theory and Applications*, 65:519–529, 1990.

[24] Pablo A Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, California Institute of Technology, 2000.

[25] Pablo A Parrilo. Polynomial Games and Sum of Squares Optimization. *Decision and Control, 2006 45th IEEE Conference on*, pages 2855 – 2860, 2006.

[26] Pablo A Parrilo. Materials of the Course: Algebraic Techniques and Semidefinite Optimization. Massachusetts Institute of Technology, 2010.

[27] A. Maitra Parthasarathy and T. On Stochastic Games. *Journal of Optimization Theory and Applications*, 5(No 4):289–300, 1970.

[28] T. Parthasarathy and T. E. S. Raghavan. An Orderfield Property for Stochastic Games when One Player Controls Transition Probabilities. *Journal of Optimization Theory and Applications*, 33(No 3):375–392, March 1981.

[29] Florian Potra, Cornelis Roos, and Tamas Terlaky. Using SeDuMi 1.02, A Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11(1-4):625–653, 1999.

[30] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. 1994.

[31] S Raghavan. Finite-step Algorithms for Single-Controller and Perfect Information Stochastic Games. In *Introduction to Games and Economic Theory Selected contribution in Honor of Robert J. Aumann*, pages 227–251. 1995.

[32] T. E. S. Raghavan S. R. Mohan. An Algorithm for Discounted Switching Control Stochastic Games. *OR Spektrum 9*, pages 41–45, 1987.

[33] Bruce Schmeiser and Luc Devroye. Non-Uniform Random Variate Generation. *Journal of the American Statistical Association*, 83(403):906, September 1988.

[34] P Schweitzer and A Seidmann. Generalized Polynomial Approximations in Markovian Decision Processes. *Journal of Mathematical Analysis and Applications*, 110(2):568–582, September 1985.

[35] Parikshit Shah and Pablo a. Parrilo. Polynomial Stochastic Games via Sum of Squares Optimization. *2007 46th IEEE Conference on Decision and Control*, pages 745–750, 2007.

[36] L S Shapley. Stochastic Games. *Proceedings of the National Academy of Sciences of the United States of America*, 39(No 10):1095–1100, October 1953.

[37] Yoav Shoham and Kevin Leyton-brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. 2010.

[38] J.A. Shohat and J.D. Tamarkin. The Problem of Moments. *American Mathematical Society Mathematical surveys*, 2, 1943.

[39] Matthew J. Sobel. Noncooperative Stochastic Games. *Ann. Math. Statist.*, 42(No 6):1930–1935, 1971.

[40] Noah D. Stein, Asuman Ozdaglar, and Pablo a. Parrilo. Separable and Low-Rank Continuous Games. *International Journal of Game Theory*, 37(No 4):475–504, May 2008.

[41] Masayuki Takahashi. Equilibrium Points of Stochastic Non-Cooperative n-Person Games. *J. Sci. Hiroshima Univ. Ser. A-I Math*, 28(No 1):95–99, 1964.

[42] Geert Jan Olsder Tamer Basar. *Dynamic Noncooperative Game Theory (Classics in Applied Mathematics)*. 1995.

[43] F. Thuijsman and T.E.S Raghavan. Stochastic Games with Switching Control or ARAT Structure. *Technical Report M94-06, University of Limburg, Maastricht, The Netherlands*, 1997.

[44] Frank Thuijsman and Thirukkannamangai E. S. Raghavan. Perfect Information Stochastic Games and Related Classes. *International Journal of Game Theory*, 26(No 3):403–408, October 1997.

[45] S H Tijs and O J Vrieze. On Stochastic Games with Additive Reward and Transition Structure. *Journal of Optimization Theory and Applications*, 47(No 4):451–464, 1985.

[46] O J Vrieze. Stochastic Games, Practical Motivation and the Orderfield Property for Special Classes. In *Stochastic games and applications*. 1999.

[47] O.J. Vrieze. Stochastic Games with Finite State and Action Spaces. *Centrum voor Wiskunde en Informatica*, 1987.

[48] Ari Weinstein, Chris Mansley, and Michael Littman. Sample-based Planning for Continuous Action Markov Decision Processes. In *Processing*, page 4, 2010.

[49] Herman Weyl. Elementary Proof of a Minimax Theorem due to Von Neumann. *Contributions to the Theory of Games*, 1(No 24):19–25, 1950.