بسم الله الرحمن الرحیم

سیستم‌های چندعاملی

درس ۱۷

مذاکره

Negotiation

کاظم فولادی قلعه
دانشکده مهندسی، پردیس فارابی
دانشگاه تهران

http://courses.fouladi.ir/mas

رسیدن به توافق

REACHING AGREEMENTS

عامل‌ها چگونه می‌توانند به توافق برسند،
وقتی که **منفعت‌طلب شخصی** هستند؟

در یک حالت مرزی (بازی مجموع صفر)، هیچ توافقی ممکن نیست.
اما در بیشتر سناریوها، امکان توافق بر روی سود متقابل در مورد منفعت مشترک وجود دارد.

قابلیت‌هایی مثل **مذاکره** و **مباحثه**،
برای اینکه عامل‌ها قادر شوند به چنین توافق‌هایی دست پیدا کنند، اساسی است.

مکانیسم‌ها، پروتکل‌ها و استراتژی‌ها

MECHANISMS, PROTOCOLS, AND STRATEGIES

مذاکره، با یک مکانیسم یا پروتکل خاص اداره می‌شود.

مکانیسم، «قواعد مواجهه» *"rules of encounter"* بین عامل‌ها را تعریف می‌کند.

طراحی مکانیسم
Mechanism Design

طراحی مکانیسم، طراحی مکانیسم‌هایی است که خصوصیات مطلوب بخصوصی را داشته باشند.

طراحی استراتژی پس از طراحی مکانیسم مطرح می‌شود:

با داشتن یک پروتکل/مکانیسم بخصوص،
چگونه می‌توانیم یک استراتژی خاص را طراحی کنیم که
تک تک عامل‌ها بتوانند از آن استفاده کنند؟

حراج در مقابل مذاکره

AUCTIONS VS NEGOTIATION

حراج، **تنها** درگیر تخصیص کالاهاست:
در عمل تکنیک‌های قوی‌تری برای رسیدن به توافق لازم است.

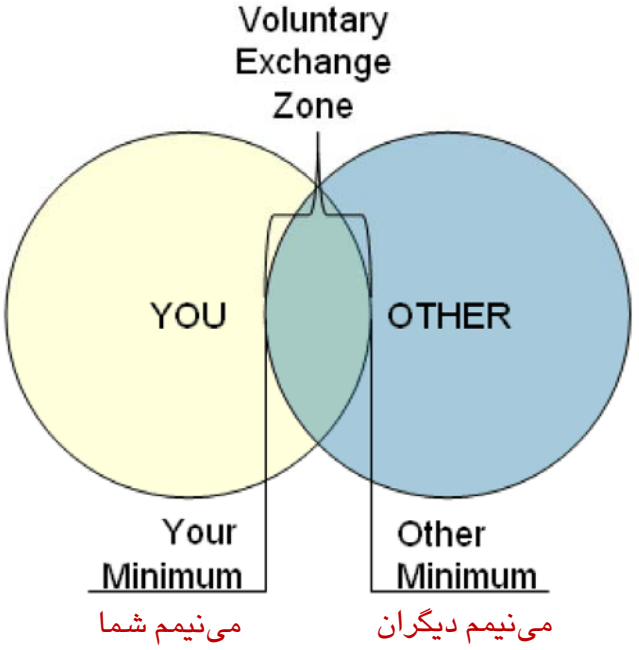
مذاکره، فرآیند رسیدن به توافق بر روی موضوعات **مورد علاقه‌ی مشترک** است.

مذاکره

دو تصویر گویا

NEGOTIATION

منطقه‌ی مبادله‌ی داوطلبانه



مذاکره

مؤلفه‌ها

NEGOTIATION

مؤلفه‌های مذاکره *Negotiation Components*			
مجموعه‌ی مذاکره *Negotiation Set*	**پروتکل** *Protocol*	**استراتژی‌ها** *Strategies*	**قاعده‌ی معامله** *Deal Rule*
پیشنهادهای ممکن که عامل‌ها مطرح می‌کنند.	قراردادهای حاکم بر مذاکره	برای هر عامل یک استراتژی وجود دارد که **خصوصی** است.	قاعده‌ای که تعیین می‌کند **چه زمانی** معامله رخ می‌دهد و **توافق معامله** چیست.

مذاکره معمولاً در یک سری از **دور**ها پیش می‌رود.
هر عامل در هر دور یک پیشنهاد مطرح می‌کند.

مذاکره

دشواری‌ها

NEGOTIATION

دشواری‌های مذاکره
Negotiation Difficulties

تعدد موضوع
Multiple Issues

تعداد معامله‌های ممکن بر حسب تعداد موضوع **نمایی** است.

(مثل تعداد دسته‌ها در حراج ترکیبیاتی)

مقایسه‌ی پیشنهادها در قبال موضوع دشوار است.

(مثل مسئله‌ی فروشنده‌ی خودرو)

تعدد عامل‌ها
Multiple Agents

مذاکره‌ی یک–به–یک

مذاکره‌ی یک–به–چند

مذاکره‌ی چند–به–یک

مذاکره برای تقسیم منبع

مدل پیشنهادهای جایگزین روبینشتاین

NEGOTIATION FOR RESOURCE DIVISION

مدل پیشنهادهای جایگزین روبینشتاین
Rubinstein's Alternating Offers Model

یک پروتکل یک–به–یک

دو عامل ۱ و ۲ داریم.

عامل‌ها طی یک سری از دورها مذاکره می‌کنند:

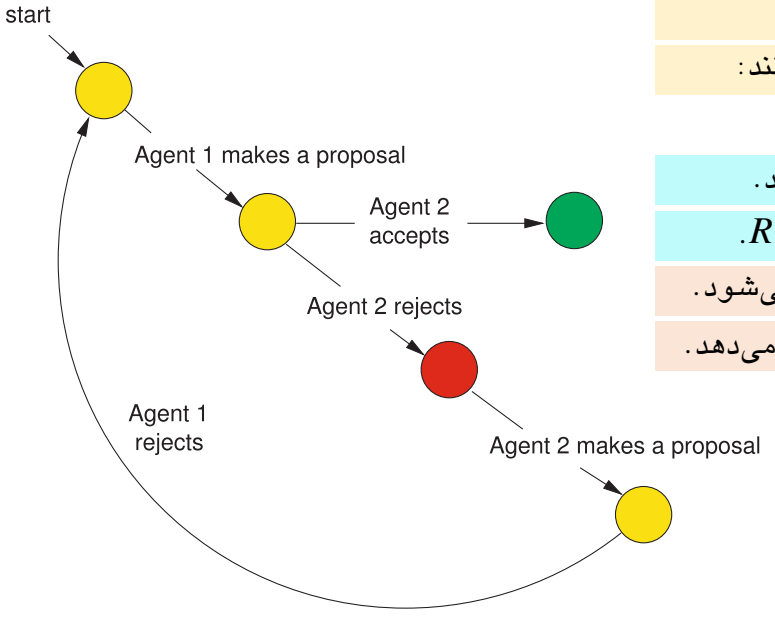
$0, 1, 2, \ldots$

دور 0: عامل ۱ پیشنهاد x^0 را مطرح می‌کند.

عامل ۲ پیشنهاد را می‌پذیرد A یا رد می‌کند R.

اگر پیشنهاد پذیرفته شد: **معامله** پیاده‌سازی می‌شود.

وگرنه، وارد دور 1 می‌شویم و عامل ۲ پیشنهاد می‌دهد.



مذاکره برای تقسیم منبع

NEGOTIATION FOR RESOURCE DIVISION

قواعد این پروتکل به این معنی نیست که همیشه توافق حاصل می‌شود.

عامل‌ها می‌توانند همیشه پیشنهادها را رد کنند ⇐ **درگیری**

درگیری ***Conflict Deal***	وقتی هیچ توافقی وجود ندارد، می‌گوییم نتیجه **درگیری** است.

دو پیش‌فرض زیر را در نظر می‌گیریم:

۱ **عدم توافق بدترین پی‌آمد است.**

هر دو عامل هر توافقی را بر هیچ ترجیح می‌دهند.

۲ **عامل‌ها به‌دنبال ماکزیمم‌سازی سود خود هستند.**

عامل‌ها ترجیح می‌دهند مقادیر سود بیشتری دریافت کنند.

با این مدل پایه، به برخی نتایج غیرعادی می‌رسیم!

تقسیم کردن یک کیک

مدل بر اساس مذاکره برای تقسیم منبع

DIVIDING A PIE

تقسیم منبعی با مقدار 1 به دو بخش
هر بخش بین 0 و 1 است.
مجموع دو بخش برابر با 1 است.

پس: یک **پیشنهاد** به‌صورت $(x, 1-x)$ مطرح می‌شود.

مجموعه‌ی همه‌ی معامله‌های ممکن عبارت است از:

$$\{(x,1-x) : 0 \le x \le 1\}$$

اگر شما عامل ۱ باشید، چه پیشنهادی می‌دهید؟

بازی اولتیماتوم

ULTIMATUM GAME

بازی اولتیماتوم *Ultimatum Game*	یک بازی که فقط یک دور دارد.

تقسیم کردن یک کیک

اگر فقط یک دور داشته باشیم

DIVIDING A PIE

بازی اولتیماتوم

عامل ۱ همه‌ی قدرت را در اختیار دارد.

اگر عامل ۱ پیشنهاد $(1,0)$ را ارائه دهد،
این پیشنهاد برای عامل ۲ هنوز بهتر از **درگیری** است: پذیرش توسط عامل ۲

عامل ۱ کار بهتری نمی‌تواند انجام بدهد.

⇓

$(1,0)$ **تعادل نش** این بازی است!

تقسیم کردن یک کیک

اگر دو دور داشته باشیم

DIVIDING A PIE

اگر **دو** دور داشته باشیم

در دور دوم، قدرت به عامل ۲ منتقل می‌شود.

آنچه عامل ۱ پیشنهاد می‌کند، توسط عامل ۲ رد می‌شود.
اعامل ۱ پیشنهاد $(0,1)$ را ارائه می‌دهد.

مشابه قبل، این پیشنهاد برای عامل ۱ هنوز بهتر از **درگیری** است: پذیرش توسط عامل ۲

عامل ۱ کار بهتری نمی‌تواند انجام بدهد.

⇓

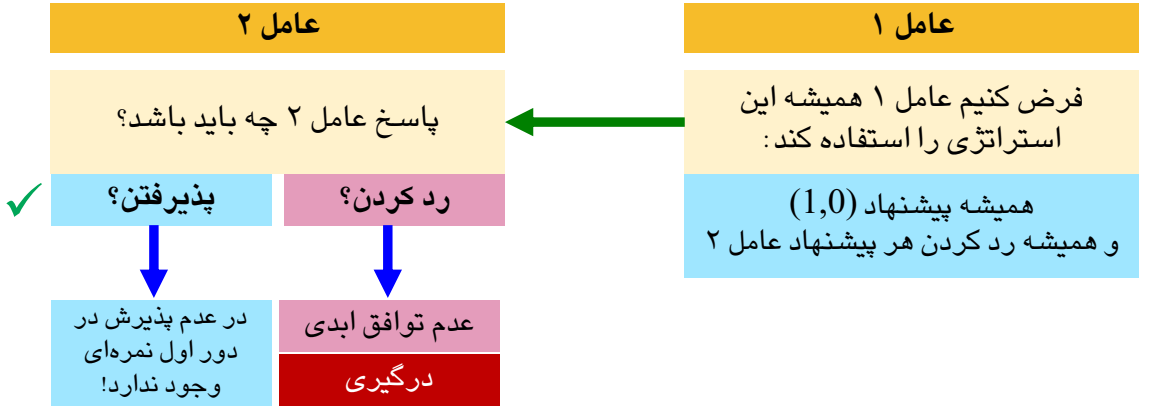
$(0,1)$ **تعادل نش** این بازی است!

در حالت کلی، اگر **تعداد دورها ثابت** باشد، یکی از نتایج $(0,1)$ یا $(1,0)$ حاصل می‌شود.

تقسیم کردن یک کیک

اگر بی‌نهایت دور داشته باشیم

DIVIDING A PIE



در واقع برای هر $(x, 1-x)$ که توسط عامل ۱ پیشنهاد می‌شود،
پذیرش بلافاصله، تعادل نش است؛
مادامی که عامل ۲، **می‌داند** استراتژی عامل ۱ چیست.

پس، تعداد بی‌نهایت تعادل نش وجود دارد که همگی **بهینه‌ی پارتو** هم هستند.

تقسیم کردن یک کیک

عامل‌های نابردبار

IMPATIENT PLAYERS

نظر به اینکه تعداد بی‌نهایت تعادل نش داریم،
مفهوم راه‌حل تعادل نش برای این مسئله خیلی ضعیف‌تر از آن است که کمکی بکند.

برای به‌دست آوردن نتایج یکتا، **زمان** را هم در نظر می‌گیریم:

برای هر برآمد x و زمان‌های $t_2 > t_1$ هر دو عامل x را در زمان t_1 ترجیح می‌دهند.

یک روش استاندارد برای مدل کردن این نابردباری، تخفیف مقادیر برآمد است.

نرخ تخفیف
Discount Rate

برای هر عامل نرخ $0 \leq \delta < 1$ را در نظر می‌گیریم:
هر چه δ به **یک** نزدیک‌تر باشد، عامل **بردبارتر** است.

تقسیم کردن یک کیک

عامل‌های نابردبار

IMPATIENT PLAYERS

هر عامل i که $i \in \{1,2\}$ یک δ_i دارد که $0 \leq \delta_i < 1$.

اگر عامل i پیشنهاد x را بدهد، مقدار برش می‌شود:

time 0	x
time 1	$\delta_i x$
time 2	$\delta_i \delta_i x = \delta_i^2 x$
time 3	$\delta_i \delta_i \delta_i x = \delta_i^3 x$
...	...
time k	$\delta_i \cdots \delta_i \delta_i x = \delta_i^k x$

اینگونه می‌توانیم با تعداد ثابتی دور، پیشرفت داشته باشیم.

تقسیم کردن یک کیک

عامل‌های نابردبار

IMPATIENT PLAYERS

اگر **یک دور** داشته باشیم: همان بازی اولتیماتوم می‌شود.

اگر **دو دور** داشته باشیم

عامل ۱	عامل ۲
$t = 0$	$t = 1$
عامل ۱ پیشنهاد $(1,0)$ را می‌دهد	عامل ۲ پیشنهاد را رد می‌کند. اما فقط δ_2 می‌گیرد.
	عامل ۱ این را در نظر می‌گیرد

عامل ۱	عامل ۲
$t = 0$	$t = 1$
اگر عامل ۱ پیشنهاد $(1-\delta_2,\delta_2)$ را بدهد.	عامل ۲ پیشنهاد را می‌پذیرد. اما و δ_2 می‌گیرد.
	عامل ۲ بهتر از این نمی‌تواند عمل کند!

پس این یک **تعادل نش** است: $(1-\delta_2,\delta_2)$

تقسیم کردن یک کیک

عامل‌های نابردبار

IMPATIENT PLAYERS

در حالت کلی، عامل ۱ پیشنهادی به عامل ۲ می‌دهد که
عامل ۲ می‌خواهد بتواند آن را در دور دوم اعمال کند.

عامل ۱	عامل ۲
آنچه دریافت می‌کند: $\frac{1-\delta_2}{1-\delta_1\delta_2}$	آنچه دریافت می‌کند: $\frac{\delta_2(1-\delta_1)}{1-\delta_1\delta_2}$

هر چه یک عامل **بردبارتر** باشد، **مقدار بیشتری** دریافت می‌کند.

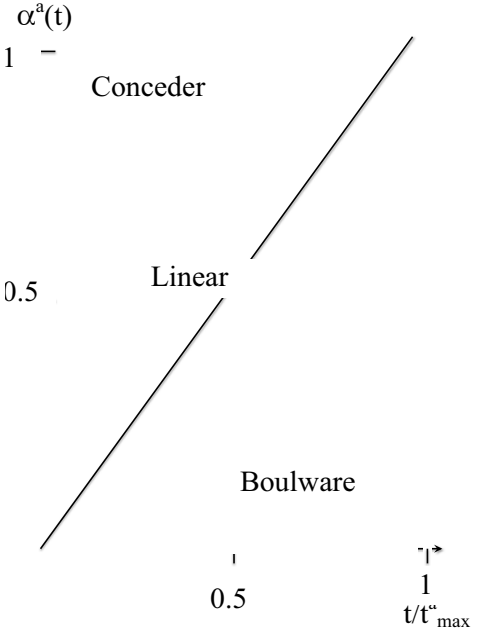
تقسیم کردن یک کیک

روی‌کردهای هیوریستیک

HEURISTIC APPROACH

راه‌حل‌هایی که تاکنون ارائه شد، بر مبنای تفکر استراتژیک در مورد بازیکن مقابل بود.

روی‌کرد ساده‌تر استفاده از برخی تقریب‌های هیوریستیک ساده‌تر **برای چگونگی تغییر تقسیم** است.



سه تقریب (هیوریستیک) ساده [از دید خریدار]

Linear

افزایش خطی از قیمت آغازین در زمان شروع
تا قیمت معکوس در مهلت زمانی

Boulware

افزایش بسیار کند تا رسیدن به نزدیک
مهلت زمانی و سپس افزایش نمایی

Conceder

افزایش نمایی تا رسیدن به نزدیک
قیمت معکوس و سپس افزایش بسیار کم

دامنه‌های مذاکره

NEGOTIATION DOMAINS

دامنه‌های وظیفه‌محور
Task-Oriented Domains (TOD)

دامنه‌های سرمایه‌محور
Wealth-Oriented Domains (WOD)

دامنه‌های مذاکره

دامنه‌های وظیفه‌محور

TASK-ORIENTED DOMAINS (TOD)

دامنه‌های وظیفه‌محور
Task-Oriented Domains (TOD)

وظیفه‌ها صراحتاً در مواجهه تعریف شده‌اند:
به هر عامل مجموعه‌ای از وظایف برای انجام داده می‌شود که به هر یک هزینه‌ای نسبت داده شده است.
هر عامل تلاش می‌کند که هزینه‌ی کل انجام وظیفه‌ها را می‌نیمم کند.

دامنه‌های سرمایه‌محور
Wealth-Oriented Domains (WOD)

مذاکره در دامنه‌های وظیفه‌محور

NEGOTIATION IN TASK-ORIENTED DOMAINS (TOD)

یک **دامنه‌ی وظیفه‌محور** *(TOD)* یک سه‌تایی به‌صورت زیر است:

$$\langle T, Ag, c \rangle$$

T	**مجموعه‌ی وظیفه‌ها** *Tasks Set*	یک مجموعه‌ی متناهی از وظیفه‌های ممکن
Ag	**عامل‌ها** *Agents*	مجموعه‌ای از n عامل شرکت کننده $Ag = \{1, 2, \ldots, n\}$
c	**هزینه‌ی اجرا** *Execution Cost*	هزینه‌ی اجرای هر زیرمجموعه از وظیفه‌ها $c : 2^T \rightarrow \mathbb{R}^+$

مواجهه *Encounter*

یک **مواجهه**، یک گردایه از زیرمجموعه‌های وظیفه‌ها است: (هر زیرمجموعه به یک عامل)

$\langle T_1, T_2, \ldots, T_n \rangle$
$T_i \subseteq T$ for each $i \in Ag$

مذاکره در دامنه‌های وظیفه‌محور

معامله‌ها در دامنه‌های وظیفه‌محور

NEGOTIATION IN TASK-ORIENTED DOMAINS

برای **مواجهه‌ی** *(encounter)*

$$\langle T_1, T_2 \rangle$$

یک **معامله** *(deal)*، یک تخصیص از وظیفه‌های $T_1 \cup T_2$ به عامل‌های ۱ و ۲ است.

هزینه‌ی معامله $\delta = \langle D_1, D_2 \rangle$ برای i عبارت است از $c(D_i)$ و به‌صورت $cost_i(\delta)$ نوشته می‌شود.

سودمندی معامله‌ی δ برای عامل i عبارت است از:

$$u_i(\delta) = c(T_i) - cost_i(\delta)$$

* درگیری *(conflict deal)* Θ: معامله‌ی $\langle T_1, T_2 \rangle$ شامل وظیفه‌های تخصیص‌داده شده در ابتدا است:

$$\forall i \in Ag \quad u_i(\Theta) = 0$$

* معامله‌ی δ **رسیونال انفرادی** است اگر سودمندی مثبت بدهد.

مجموعه‌ی مذاکره

THE NEGOTIATION SET

مجموعه‌ی مذاکره
The Negotiation Set

مجموعه‌ی **معامله‌ها** بر روی عامل‌ها که دارای دو خاصیت زیر باشد:
۱) **رسیونال انفرادی** ۲) **کارآمد پارتو**

رسیونال انفرادی
Individually Rational

عاملی که علاقه‌مند نیست در یک معامله سودمندی منفی به‌دست آورد، و درگیری (سودمندی صفر) را بر آن ترجیح می‌دهد.

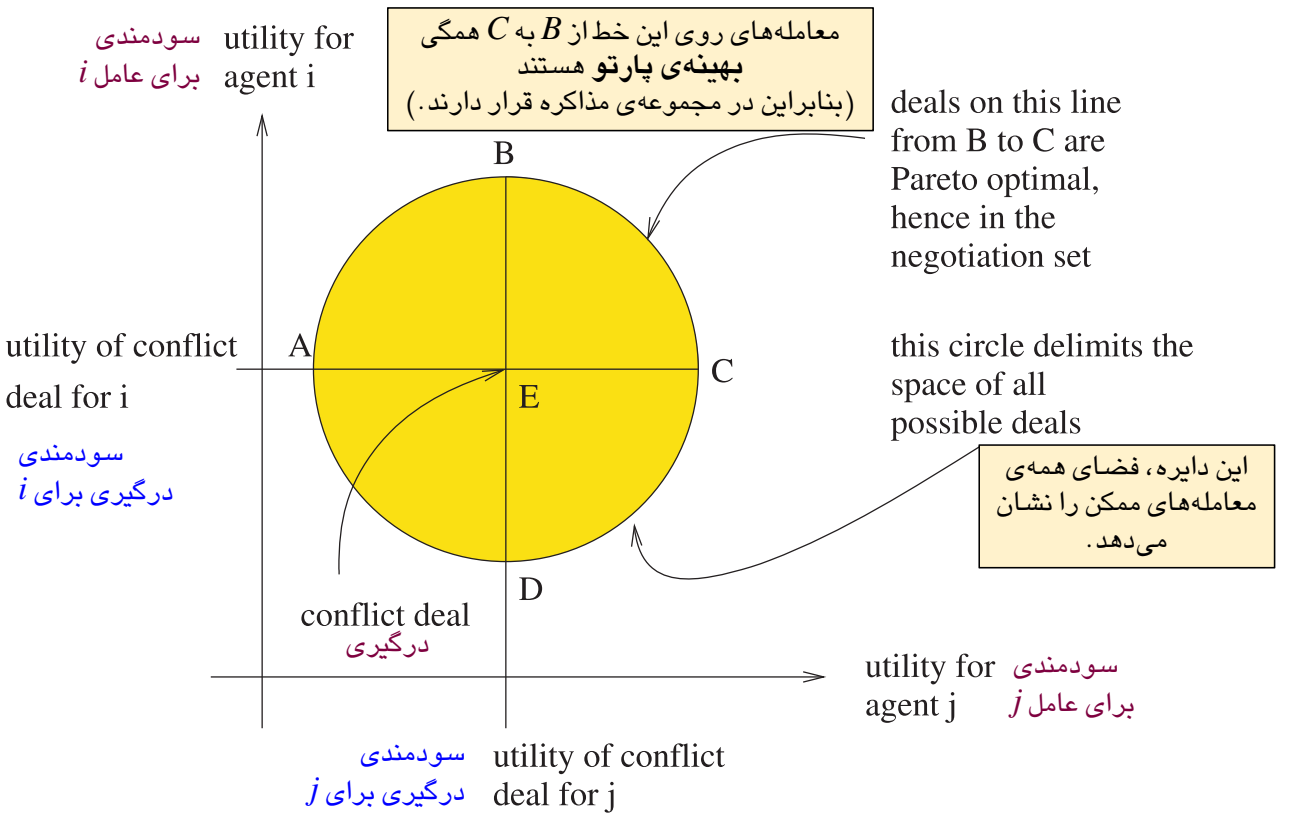
کارآمد پارتو
Pareto Efficient

عامل‌ها همیشه می‌توانند یک معامله‌ی کارآمد غیر پارتو را به معامله‌ی کارآمد پارتو تبدیل کنند: با بهتر کردن سود یک عامل بدون اینکه سود بقیه‌ی عامل‌ها بدتر شود.

مجموعه‌ی مذاکره

نمودار

THE NEGOTIATION SET



Prepared by Kazim Fouladi | Fall 2017 | 3rd Edition
کاظم فولادی

مذاکره در دامنه‌های وظیفه‌محور

پروتکل اعطای یکنوا

THE MONOTONIC CONCESSION PROTOCOL

پروتکل اعطای یکنوا
The Monotonic Concession Protocol

مذاکره در چند دور انجام می‌شود:

دور اول:

عامل‌ها به‌طور همزمان یک معامله را از مجموعه‌ی مذاکره پیشنهاد می‌دهند.

توافق حاصل می‌شود: اگر یکی از عامل‌ها ببیند که
معامله‌ی پیشنهادی توسط دیگری به‌اندازه‌ی پیشنهاد خودش خوب یا بهتر باشد.

اگر توافق حاصل نشود، آن‌گاه
مذاکره در دور بعدی با پیشنهادهای همزمان ادامه می‌یابد.

دور $r + 1$:

عامل‌ها به‌طور همزمان یک معامله را از مجموعه‌ی مذاکره پیشنهاد می‌دهند.

(اعطا) هیچ عاملی اجازه ندارد که پیشنهادی بدهد که
نسبت به معامله‌ی پیشنهاد داده شده در دور r توسط عامل دیگر کمتر ترجیح داده شود.

اگر هیچ‌یک از عامل‌ها در یک دور $r > 0$ اعطایی نداشته باشد، مذاکره با **درگیری** خاتمه می‌یابد.

مذاکره در دامنه‌های وظیفه‌محور

استراتژی زئوثن

THE ZEUTHEN STRATEGY

استراتژی زئوثن، به سه مسئله در پروتکل اعطای یکنوا پاسخ می‌دهد:

۱	۲	۳
اولین پیشنهاد یک عامل چه باید باشد؟	**در هر دور کدام عامل باید اعطا کند؟**	**عاملی که اعطا می‌کند، چه‌قدر باید اعطا کند؟**
مرجح‌ترین معامله برای آن عامل	عاملی که کمترین تمایل را برای ریسک درگیری دارد.	به‌اندازه‌ای که توازن ریسک تغییر کند، کافی است.

عاملی که بخواهد **بسیار** اعطا کند:

- پیشنهادش به درگیری نزدیک می‌شود.
- اگر درگیری رخ بدهد، برای این عامل خیلی بدتر نمی‌شود.
- این عامل، **تمایل بیشتری** به ریسک درگیری دارد.

یک عامل
تمایل بیشتری به ریسک درگیری
دارد، اگر
تفاضل سودمندی او بین پیشنهاد فعلی و معامله‌ی درگیری پایین باشد.

مذاکره در دامنه‌های وظیفه‌محور

استراتژی زئوثن

THE ZEUTHEN STRATEGY

استراتژی زئوثن، در تعادل نش است:
تحت این فرض که یک عامل از یک استراتژی استفاده می‌کند
که دیگری خودش نمی‌تواند بهتر از این عمل کند ...

مذاکره در دامنه‌های وظیفه‌محور

فریب در دامنه‌های وظیفه‌محور

DECEPTION IN TODS

در *TOD*ها، عامل‌ها از دو طریق می‌توانند از فریب دادن سود ببرند:

وظیفه‌های پوچ و تطمیع‌کننده
Phantom and Decoy Tasks

تظاهر می‌کند که وظیفه‌ای به عامل تخصیص داده شده است که در واقع نشده است.

وظیفه‌های پنهان
Hidden Tasks

تظاهر می‌کند که وظیفه‌ای به عامل تخصیص داده نشده است که در واقع شده است.

دامنه‌های مذاکره

دامنه‌های سرمایه‌محور

WEALTH-ORIENTED DOMAINS (WOD)

دامنه‌های وظیفه‌محور
Task-Oriented Domains (TOD)

دامنه‌های سرمایه‌محور
Wealth-Oriented Domains (WOD)

هدف‌های یک عامل توسط تعریف یک تابع سرمایه برای حالت‌های ممکن محیط مشخص می‌شود.
هدف این عامل به‌طور ضمنی ایجاد حالتی از محیط است که دارای بیشترین ارزش است.

برای این‌کار باید، برای عامل‌ها مجموعه‌ای از طرح‌های مشترک *(joint plans)* موجود باشد.

بنابراین عامل‌ها برای طرح‌های مشترک مذاکره می‌کنند (نه وظیفه‌ها)

علاقه‌ی هر عامل به حصول توافقی روی طرحی است که حالت محیط با بیشترین سرمایه را ایجاد می‌کند.

دامنه‌های سرمایه‌محور

WEALTH-ORIENTED DOMAINS (WOD)

یک **دامنه‌ی سرمایه‌محور** *(WOD)* یک چهارتایی به‌صورت زیر است:

$$\langle E, Ag, J, c \rangle$$

E	**حالت‌های محیط** *Environment States*	مجموعه‌ی حالت‌های ممکن محیط
Ag	**عامل‌ها** *Agents*	مجموعه‌ای از n عامل شرکت کننده $Ag = \{1, 2, \ldots, n\}$
J	**طرح‌های مشترک** *Joint Plans*	مجموعه‌ی طرح‌های مشترک ممکن
c	**تابع هزینه** *Cost Function*	هزینه‌ی اجرای طرح j برای عامل i $c : J \times Ag \rightarrow \mathbb{R}$

مواجهه *Encounter*

یک **مواجهه**، یک دوتایی شامل حالت آغازین محیط و یک تابع سرمایه است.

$\langle e, W \rangle$

$e \in E$

$W : E \times Ag \rightarrow \mathbb{R}$

(سرمایه‌ی عامل i در حالت محیط e)

دامنه‌های سرمایه‌محور

طرح مشترک بهینه

WEALTH-ORIENTED DOMAINS (WOD)

یک طرح *(plan)*:
(طرح مشترکی که در حالت e_1 اجرا می‌شود و پس از اجرا در آن، منجر به حالت e_2 می‌شود.)

$$j : e_1 \rightsquigarrow e_2$$

اگر عامل به‌تنهایی عمل کند، نیازی به مذاکره نیست:

$$j^i_{\text{opt}} = \arg \max_{j:e_0 \rightsquigarrow e \in J} (W(i,e) - C(j,i))$$

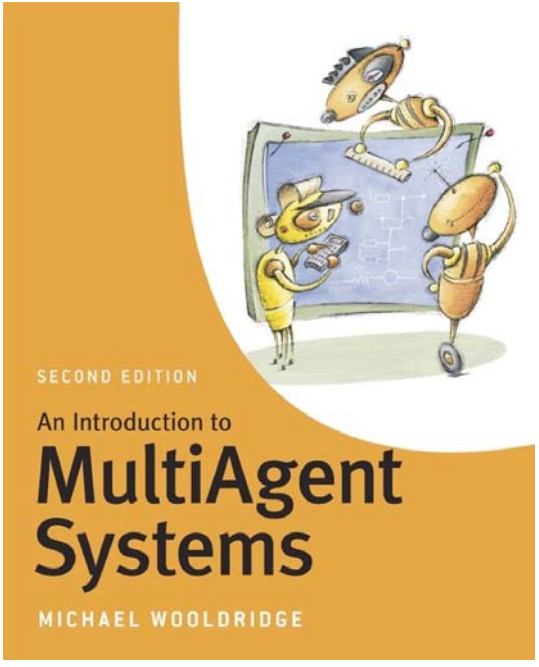
اما در مواجهات چندعاملی، عامل می‌تواند از حضور سایر عامل‌ها سود ببرد: نیاز به مذاکره

سیستم‌های چندعاملی

مذاکره

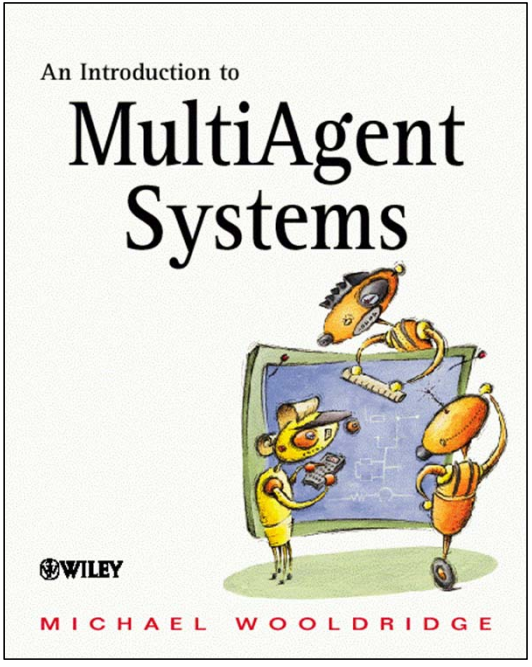
منابع

منبع اصلی

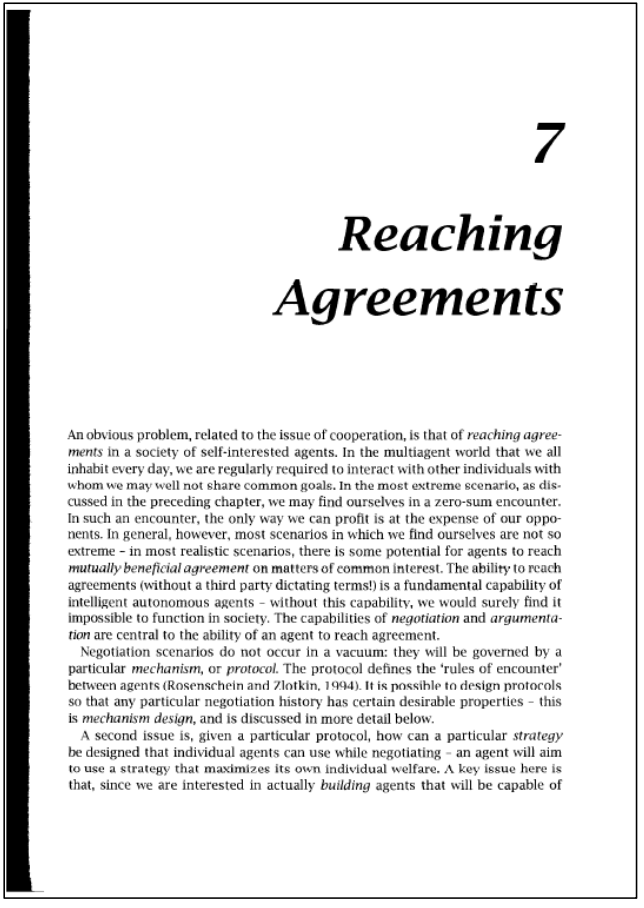


Michael Wooldridge,
An Introduction to Multiagent Systems,
Second Edition,
John Wiley & Sons, 2009.
Chapter 15

منبع کمکی



Michael Wooldridge,
An Introduction to Multiagent Systems,
John Wiley & Sons, 2002.
Chapter 7



7

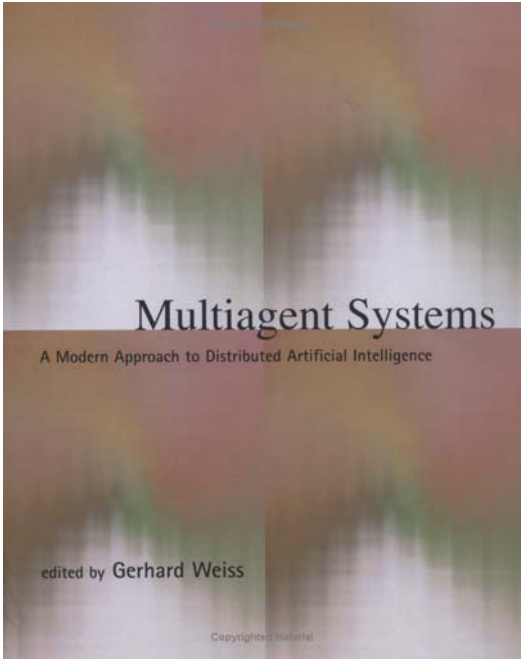
Reaching Agreements

An obvious problem, related to the issue of cooperation, is that of *reaching agreements* in a society of self-interested agents. In the multiagent world that we all inhabit every day, we are regularly required to interact with other individuals with whom we may well not share common goals. In the most extreme scenario, as discussed in the preceding chapter, we may find ourselves in a zero-sum encounter. In such an encounter, the only way we can profit is at the expense of our opponents. In general, however, most scenarios in which we find ourselves are not so extreme – in most realistic scenarios, there is some potential for agents to reach *mutually beneficial agreement* on matters of common interest. The ability to reach agreements (without a third party dictating terms!) is a fundamental capability of intelligent autonomous agents – without this capability, we would surely find it impossible to function in society. The capabilities of *negotiation* and *argumentation* are central to the ability of an agent to reach agreement.

Negotiation scenarios do not occur in a vacuum: they will be governed by a particular *mechanism*, or *protocol*. The protocol defines the 'rules of encounter' between agents (Rosenschein and Zlotkin, 1994). It is possible to design protocols so that any particular negotiation history has certain desirable properties – this is *mechanism design*, and is discussed in more detail below.

A second issue is, given a particular protocol, how can a particular *strategy* be designed that individual agents can use while negotiating – an agent will aim to use a strategy that maximizes its own individual welfare. A key issue here is that, since we are interested in actually *building* agents that will be capable of

منبع کمکی



Gerhard Weiss (ed.),
Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence,
MIT Press, 1999.
Chapter 5 (5.4)

5 Distributed Rational Decision Making

Tuomas W. Sandholm

5.1 Introduction

Automated negotiation systems with self-interested agents are becoming increasingly important. One reason for this is the *technology push* of a growing standardized communication infrastructure—Internet, WWW, NII, EDI, KQML, FIPA, Concordia, Voyager, Odyssey, Telescript, Java, *etc.*—over which separately designed agents belonging to different organizations can interact in an open environment in real-time and safely carry out transactions. The second reason is strong *application pull* for computer support for negotiation at the operative decision making level. For example, we are witnessing the advent of small transaction electronic commerce on the Internet for purchasing goods, information, and communication bandwidth [31]. There is also an industrial trend toward virtual enterprises: dynamic alliances of small, agile enterprises which together can take advantage of economies of scale when available (e.g., respond to more diverse orders than individual agents can), but do not suffer from diseconomies of scale.

Multiagent technology facilitates such negotiation at the operative decision making level. This automation can save labor time of human negotiators, but in addition, other savings are possible because computational agents can be more effective at finding beneficial short-term contracts than humans are in strategically and combinatorially complex settings.

This chapter discusses multiagent negotiation in situations where agents may have different goals, and each agent is trying to maximize its own good without concern for the global good. Such self-interest naturally prevails in negotiations among independent businesses or individuals. In building computer support for negotiation in such settings, the issue of self-interest has to be dealt with. In *cooperative distributed problem solving* [12, 9], the system designer imposes an interaction *protocol*[1] and a *strategy* (a mapping from state history to action; a

1. Here a protocol does not mean a low level communication protocol, but a negotiation protocol which determines the possible actions that agents can take at different points of the interaction. The *sealed-bid first-price auction* is an example protocol where each bidder is free to submit one bid for the item, which is awarded to the highest bidder at the price of his bid.