سیستم‌های چندعاملی

درس ۹

هماهنگی عامل‌ها

همکاری

Coordination of Agents: Cooperation

کاظم فولادی قلعه
دانشکده مهندسی، پردیس فارابی
دانشگاه تهران

http://courses.fouladi.ir/mas

تصمیم‌گیری توزیع شده

نقش هماهنگی

DISTRIBUTED DECISION MAKING

در یک سیستم چندعاملی، تصمیم‌گیری عامل‌ها می‌تواند **توزیع‌شده** باشد.
(نتیجه: کارآمدی و قوام)

اما، مکانیزم‌های **هماهنگی** بیشتری باید طراحی شود.

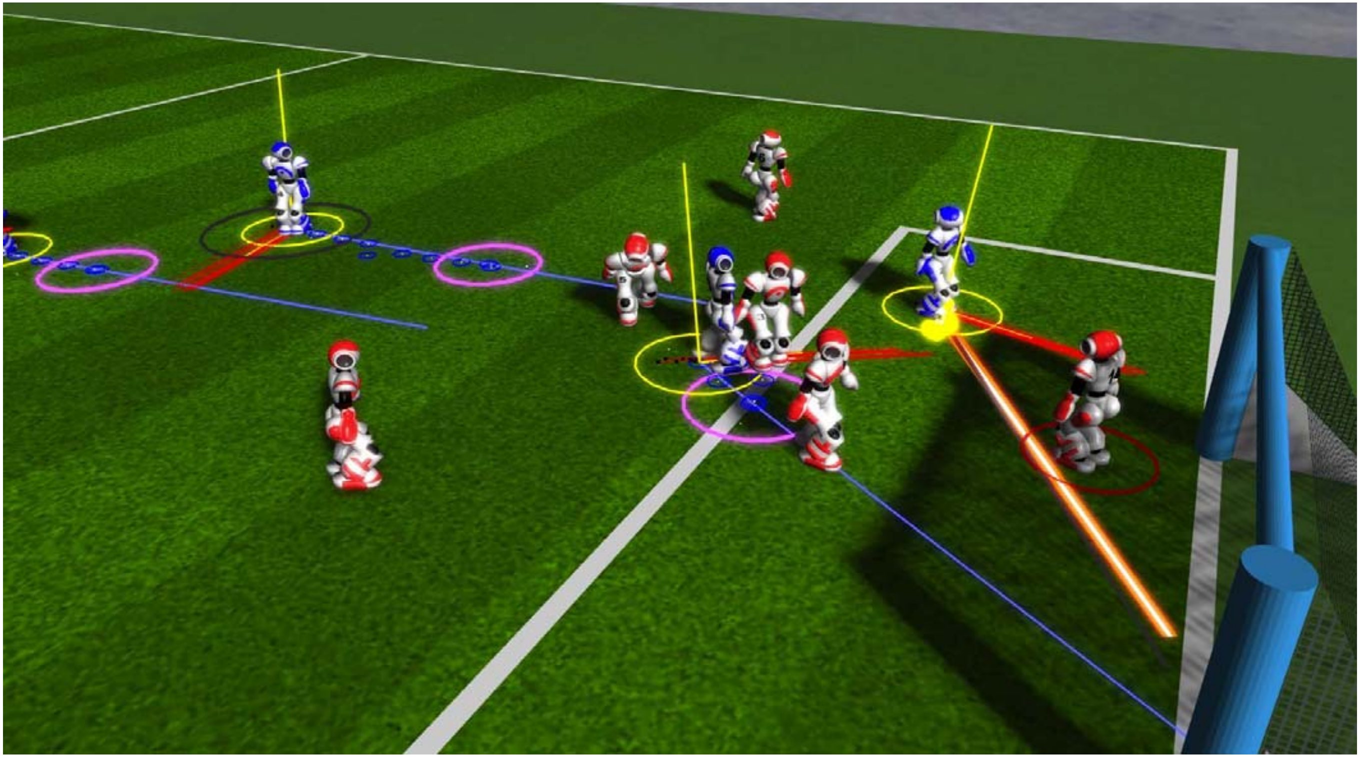
هماهنگی
تضمین می‌کند که
تصمیم‌های انفرادی عامل‌ها
منجر به
تصمیم‌های توأم خوب برای گروه
شود.

یک حالت خاص، عامل‌های **همکار** (*Cooperative Agents*) است.

عامل‌های همکار

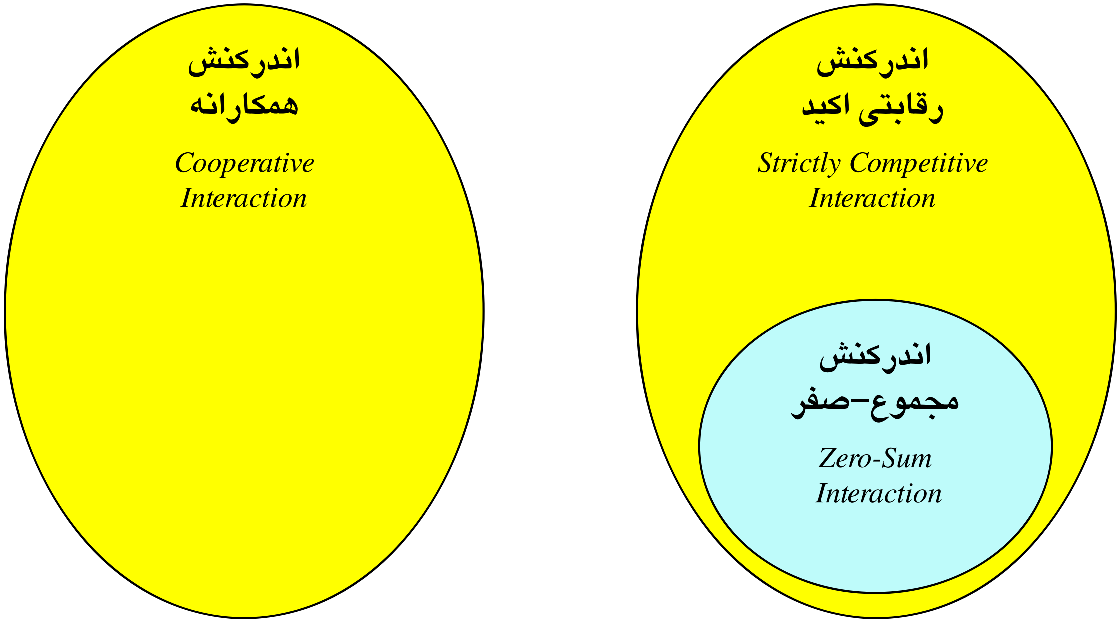
مثال

COOPERATIVE AGENTS



عامل‌های همکار: مانند ربات‌های فوتبالیست در یک تیم

بازی‌ها



	Cross	*Stop*
Cross	$-1, -1$	$1, 0$
Stop	$0, 1$	$0, 0$

Coordination game

	Head	*Tail*
Head	$1, -1$	$-1, 1$
Tail	$-1, 1$	$1, -1$

Zero-sum game

بازی‌های هماهنگی

مثال

Coordination Game

خودروها در تقاطع

Cars in an Intersection

دو خودرو در یک تقاطع به هم می‌رسند. رانندگان چه کاری باید انجام دهند؟

- اگر هر دو عبور کنند، تصادف می‌کنند و هر دو ضرر می‌کنند.
- اگر یکی توقف کند و دیگری عبور کند، فقط عبور کننده سود می‌کند.
- اگر هر دو توقف کنند، هیچ‌کدام سود نمی‌کنند.

	Cross	*Stop*
Cross	$-1, -1$	$1, \quad 0$
Stop	$0, \quad 1$	$0, \quad 0$

این بازی **دو** تعادل نش دارد که هر دو بهینه‌ی پارتو هم هستند:

$$(Cross,\ Stop) \text{ and } (Stop,\ Cross)$$

این عامل‌ها چگونه می‌توانند کنش‌های خود را **هماهنگ** کنند؟

بازی‌های هماهنگی

مثال

Coordination Game

بازی جنسیت‌ها

Battle of the Sexes

دو عامل (زن و شوهر) می‌خواهند با هم به دیدن یک فیلم بروند.

- اگر هر دو بر سر یک نوع فیلم (هیجان‌دار یا خنده‌دار) توافق کنند، می‌برند.
- در صورت عدم توافق، هیچ‌کدام سود نمی‌برند!

	Thriller	*Comedy*
Thriller	1, 1	0, 0
Comedy	0, 0	1, 1

در این بازی دو تعادل نش وجود دارد که بهینه‌ی پارتو هم هستند.

در حالت کلی، اگر n عامل همکار وجود داشته باشد، خواهیم داشت:

$$u_1(a) = \ldots = u_n(a) \equiv u(a)$$

هماهنگی

COORDINATION

هماهنگی *Coordination*	هماهنگی فرآیندی است که در آن عامل‌ها بر روی یک تعادل نش بهینه‌ی پارتو **واحد** توافق می‌کنند.

هماهنگ‌سازی از طریق قراردادهای اجتماعی

Coordination Via Social Conventions

قرارداد اجتماعی
Social Convention

قاعده‌ای که می‌گوید
عامل‌ها **چگونه** باید یک تعادل واحد را در یک بازی انتخاب کنند.

قانون اجتماعی
Social Law

هنجار اجتماعی
Social Norm

مثال:

۱) گروهی از افراد در ایستگاه اتوبوس منتظر هستند.
وقتی اتوبوس سر می‌رسد، چه کسی باید اول سوار اتوبوس شود؟

۲) در هنگام حرکت در پیاده‌رو از کدام طرف باید حرکت کنیم؟

در سیستم‌های چندعاملی، می‌توانیم هنجارهای اجتماعی را طراحی کنیم
و عامل‌ها را به‌گونه‌ای برنامه‌ریزی کنیم که از آن هنجارها پیروی کنند یا اجازه بدهیم که هنجارها تحول پیدا کنند.

هماهنگ‌سازی از طریق قراردادهای اجتماعی

مثال

Coordination Via Social Conventions

قرارداد اجتماعی *Social Convention*	قاعده‌ای که می‌گوید عامل‌ها **چگونه** باید یک تعادل واحد را در یک بازی انتخاب کنند.

یک قرارداد ساده:

عامل‌ها بر روی یک **شمای ترتیب** همه‌ی کنش‌های توأم در یک بازی توافق می‌کنند.

در هر بازی مشخص،
عامل‌ها **اولین** تعادل را در لیست مرتب همه‌ی تعادل‌های بازی انتخاب می‌کنند.

هماهنگ‌سازی از طریق قراردادهای اجتماعی

مثال : شبه‌کد

Coordination Via Social Conventions

```
For each agent i in parallel
    Compute all equilibria of the game.
    Order these equilibria based on a unique ordering scheme.
    Select the first equilibrium a* = (a*_{-i}, a*_i) in the ordered list.
    Choose action a*_i.
End
```

هماهنگ‌سازی از طریق نقش‌ها

COORDINATION VIA ROLES

هماهنگ‌سازی توسط قراردادهای اجتماعی، متکی بر این فرض است که:
یک عامل، می‌تواند همه‌ی تعادل‌های یک بازی را پیش از انتخاب یک تعادل واحد محاسبه کند.

اما محاسبه‌ی تعادل‌ها می‌تواند گران باشد (در صورت بزرگ بودن مجموعه‌های کنش‌های عامل‌ها).

راه‌حل: ابتدا اندازه‌ی مجموعه‌های کنش‌های عامل‌ها را کاهش دهیم.

یک راه برای این کار: **انتساب نقش به عامل‌ها**

یک نقش خاص برخی کنش‌های عامل در یک حالت خاص را غیرفعال می‌کند.

هماهنگ‌سازی از طریق نقش‌ها

انتساب نقش‌ها

Coordination Via Roles

فرض می‌کنیم n نقش وجود داشته باشد.

موارد زیر باید به عنوان **دانایی مشترک** عامل‌ها در نظر گرفته شود:

ترتیب ثابتی از نقش‌ها وجود دارد: $<1, 2, \ldots, n>$

برای هر نقش j تابعی وجود دارد که به هر عامل i **پتانسیل** r_{ij} را نسبت می‌دهد:

«میزان مناسب بودن عامل i برای نقش j در حالت جاری»

به هر عامل می‌توان تنها یک نقش را نسبت داد.

COORDINATION VIA ROLES

برای مثال، در بازی فوتبال رباتیک، می‌خواهیم
یک نقش j را به نزدیک‌ترین ربات به توپ نسبت بدهیم.

پتانسیل ربات i به صورت زیر قابل محاسبه است

$$r_{ij} = \exp(-\lambda||x_i - x_b||^2)$$

که در آن x_i و x_b مکان ربات i و مکان توپ است.

هماهنگ‌سازی از طریق نقش‌ها

الگوریتم انتساب نقش به عامل: شبه‌کد

Roles Assignment

For each agent in parallel
    $I = \{\}$.
    For each role $j = 1, \ldots, n$
        For each agent $i = 1, \ldots, n$ with $i \notin I$
            Compute the potential r_{ij} of agent i for role j.
        End
        Assign role j to agent $i^* = \arg\max_i \{r_{ij}\}$.
        Add i^* to I.
    End
End

هماهنگ‌سازی از طریق نقش‌ها

انتساب نقش‌ها : تابع انتساب پتانسیل : مثال

برای مثال، در بازی فوتبال رباتیک، می‌خواهیم
یک نقش j را به نزدیک‌ترین ربات به توپ نسبت بدهیم.

پتانسیل ربات i به صورت زیر قابل محاسبه است

$$r_{ij} = \exp(-\lambda||x_i - x_b||^2)$$

که در آن x_i و x_b مکان ربات i و مکان توپ است.

گراف‌های هماهنگی

COORDINATION GRAPHS

نقش‌ها اندازه‌ی مجموعه کنش‌های عامل‌ها را پیش از محاسبه‌ی نقاط تعادل کاهش می‌دهند.

با این وجود، محاسبه‌ی نقاط تعادل در صورت بزرگ بودن n، هنوز می‌تواند دشوار باشد.

راه‌حل: تجزیه‌ی بازی به چندین زیربازی کوچک‌تر که حل آنها ساده‌تر است
یک راه برای این کار: **استفاده از گراف‌های هماهنگی**

فرض اساسی

تابع سودمندی سراسری $u(a)$ می‌تواند
به صورت یک ترکیب خطی از k تابع سودمندی محلی f_j نوشته شود، مانند:

$$u(a) = f_1(a_1, a_2) + f_2(a_1, a_3) + f_3(a_3, a_4)$$

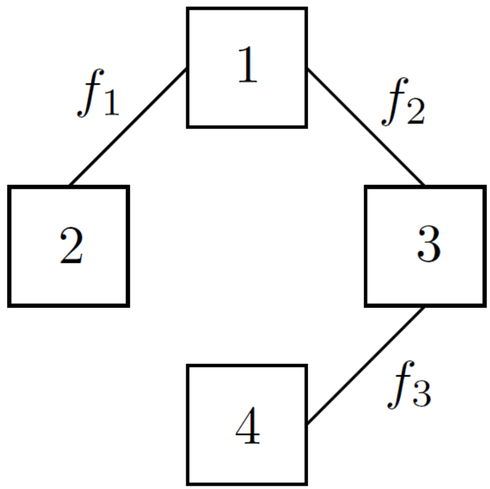
گراف‌های هماهنگی

COORDINATION GRAPHS

بر اساس یک تجزیه از تابع سودمندی، مانند

$$u(a) = f_1(a_1, a_2) + f_2(a_1, a_3) + f_3(a_3, a_4)$$

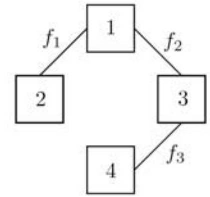
یک بازنمایی با گراف ایجاد می‌کنیم:



گراف‌های هماهنگی

الگوریتم حذف متغیر: مثال (۱ از ۲)

THE VARIABLE ELIMINATION ALGORITHM



$$u(a) = f_1(a_1, a_2) + f_2(a_1, a_3) + f_3(a_3, a_4)$$

هدف یافتن تعادل نش بهینه‌ی پارتو است که u را ماکزیمم می‌کند:

$$\max_a u(a) = \max_{a_2,a_3,a_4} \left\{ f_3(a_3, a_4) + \max_{a_1} \left[f_1(a_1, a_2) + f_2(a_1, a_3) \right] \right\}$$

تابع بهترین پاسخ $B_1(a_2,a_3)$ را برای عامل ۱
در زیربازی تشکیل‌شده از عامل‌های ۱، ۲ و ۳ و سودمندی مجموع $f_1 + f_2$ محاسبه می‌کنیم؛

یک تابع سودمندی جدید تعریف می‌کنیم:

$$f_4(a_2, a_3) = \max_{a_1} [f_1(a_1, a_2) + f_2(a_1, a_3)]$$

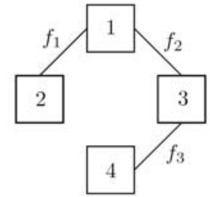
با این کار **عامل ۱** حذف می‌شود:

$$\max_a u(a) = \max_{a_2,a_3,a_4} \left[f_3(a_3, a_4) + f_4(a_2, a_3) \right]$$

گراف‌های هماهنگی

الگوریتم حذف متغیر: مثال (۲ از ۲)

THE VARIABLE ELIMINATION ALGORITHM



پس از حذف عامل ۱، ماکزیمم u می‌شود:

$$\max_a u(a) = \max_{a_2,a_3,a_4} \left[f_3(a_3,a_4) + f_4(a_2,a_3)\right]$$

حال، عامل ۲ را حذف می‌کنیم؛

با محاسبه‌ی $B_2(a_3)$ و تولید یک تابع سودمندی جدید $f_5(a_3)$:

$$\max_a u(a) = \max_{a_3,a_4} \left[f_3(a_3,a_4) + f_5(a_3)\right]$$

به‌طور مشابه، عامل ۳ را حذف می‌کنیم؛

با محاسبه‌ی $B_3(a_4)$ و تولید یک تابع سودمندی جدید $f_6(a_4)$:

$$\max_a u(a) = \max_{a_4} f_6(a_4)$$

و عامل ۴ می‌تواند a_4^* را انتخاب کند.

گراف‌های هماهنگی

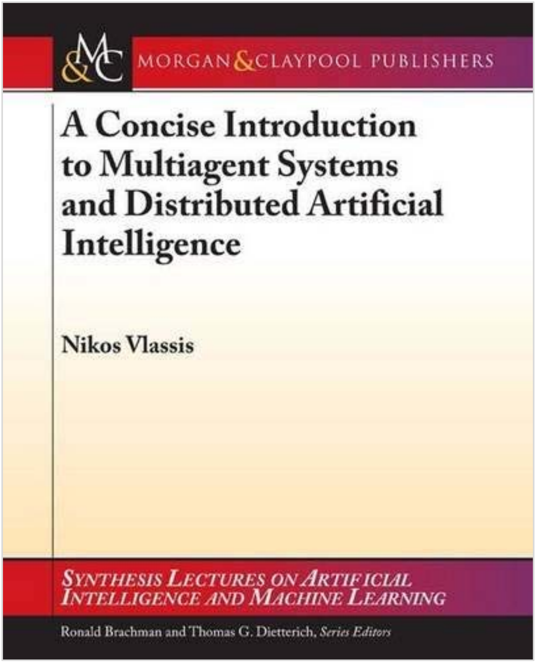
الگوریتم حذف متغیر : شبه‌کد

THE VARIABLE ELIMINATION ALGORITHM

<u>For</u> each agent in parallel
    $F = \{f_1, \ldots, f_k\}$.
    <u>For</u> each agent $i = 1, 2, \ldots, n$
        Find all $f_j(a_{-i}, a_i) \in F$ that involve a_i.
        Compute $B_i(a_{-i}) = \arg\max_{a_i} \sum_j f_j(a_{-i}, a_i)$.
        Compute $f_{k+i}(a_{-i}) = \max_{a_i} \sum_j f_j(a_{-i}, a_i)$.
        Remove all $f_j(a_{-i}, a_i)$ from F and add $f_{k+i}(a_{-i})$ in F.
    <u>End</u>
    <u>For</u> each agent $i = n, n-1, \ldots, 1$
        Choose $a_i^* \in B_i(a_{-i}^*)$ based on a fixed ordering of actions.
    <u>End</u>
<u>End</u>

منابع

منبع اصلی



Nikos Vlassis,
A Concise Introduction to Multiagent Systems and Distributed Artificial Intelligence,
Morgan & Claypool, 2007.
Chapter 4

23

CHAPTER 4

Coordination

In this chapter we address the problem of multiagent **coordination**. We analyze the problem using the framework of strategic games that we studied in Chapter 3, and we describe several practical techniques like social conventions, roles, and coordination graphs.

4.1 COORDINATION GAMES

As we argued in Chapter 1, decision making in a multiagent system should preferably be carried out in a decentralized manner for reasons of efficiency and robustness. This additionally requires developing a coordination mechanism. In the case of **collaborative** agents, coordination ensures that the agents do not obstruct each other when taking actions, and that these actions serve the common goal of the team (for example, two teammate soccer robots must coordinate their actions when deciding who should go for the ball). Informally, coordination can be regarded as the process by which the individual decisions of the agents result in good joint decisions for the group.

Formally, we can model a coordination problem as a **coordination game** using the tools of game theory, and solve it according to some solution concept, for instance Nash equilibrium. We have already seen an example in Fig. 3.2(b) of Chapter 3 of a strategic game where two cars meet at a crossroad and one driver should cross and the other one should stop. That game has two Nash equilibria, (*Cross*, *Stop*) and (*Stop*, *Cross*). In the case of n collaborative agents, all agents in the team share the same payoff function $u_1(a) = \ldots = u_n(a) \equiv u(a)$ in the corresponding coordination game. Figure 4.1 shows an example of a coordination game (played between two agents who want to go to the movies together) that also has two Nash equilibria. Generalizing from these two examples, we can formally define coordination as *the process in which a group of agents choose a single Pareto optimal Nash equilibrium in a game.*

	Thriller	*Comedy*
Thriller	1, 1	0, 0
Comedy	0, 0	1, 1

FIGURE 4.1: A coordination game